Received: 26 October 2023 | Revised: 1 April 2024 | Accepted: 8 May 2024

DOI: 10.1002/fld.5312

RESEARCH ARTICLE

WILEY

Directionally-split volume-of-fluid technique for front propagation under curvature flow

Ali Fakhreddine[1] | **Karim Alamé**[1] | **Krishnan Mahesh**[1,2]

[1]Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota, USA

[2]Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, Michigan, USA

Correspondence
Krishnan Mahesh, Department of Naval Architecture and Marine Engineering, University of Michigan, MI 48109, USA.
Email: krmahesh@umich.edu

Funding information
United States Office of Naval Research (ONR), Grant/Award Number: N00014-17-1-2676

Abstract

A directionally-split volume-of-fluid (VOF) methodology for evolving interfaces under curvature-dependent speed is devised. The interface is reconstructed geometrically and the volume fraction is advected with a technique to incorporate a topological volume conservation constraint. The proposed approach uses the idea that the role of curvature in a speed function $\mathbf{V}$ is analogous to the role of viscosity in the corresponding hyperbolic conservation law to propagate complex interfaces where singularities may exist. The approach has the advantage of simple implementation and straightforward extension to more complex multiphase systems by formulating the interface evolution problem using energy functionals to derive an expression for the interface-advecting velocity. The numerical details of the volume-of-fluid based formulation are discussed with emphasis on the importance of curvature estimation. Finally, canonical curves and surfaces traditionally investigated by the level set (LS) method are tested with the devised approach and the results are compared with existing work in LS.

KEYWORDS

curvature flow, interfaces, level set, volume-of-fluid

1 | INTRODUCTION

Traditional interface-capturing (IC) methods, namely the Volume-of-Fluid (VOF), Level set (LS), and Phase-Field (PF) methods, have been used to study a wide range of multiphase flow problems for over forty years. Some of the applications include bubbly flows, spray atomization, breaking waves, and phase change (i.e., boiling, cavitation, solidification). While all three interface-capturing methods achieve the common task of advecting a fluid interface given a prescribed velocity field, they differ in the level of accuracy, efficiency, and implementation. This consequently makes one method more favorable than the other based on the level of fidelity that the physical problem imposes on the numerical method. IC methods can be categorized by the type of approach used to track the interface. VOF and LS methods fall under sharp-interface approaches while PF methods fall under diffuse-interface approaches. The primary emphasis of this work will be on the former category.

VOF and LS methods have long existed in the two-phase modeling community, starting with the early work of Hirt and Nichols[1] who introduced the fractional VOF technique to study problems involving free-surface boundaries, and Osher and Sethian[2]—the developers of the traditional LS formulation. Further developments in that area gave rise to improved VOF[3,4] and LS formulations that emphasize discrete mass conservation and accurate interface representation.

A hybrid method originally developed by Sussman and Pucket[5] also currently exists and is known as the coupled-level set and volume-of-fluid method (CLSVOF). The hybrid method utilizes level sets to accurately compute curvature and interface normals and uses the VOF framework to improve mass conservation. Although CLSVOF achieves higher accuracy than the use of either VOF or LS separately, its employment is still less common due to higher computational cost and the inherent difficulty in achieving a balanced load between VOF and LS operations which often leads to poor parallel scalability.

At the most fundamental level, both VOF and LS present attractive frameworks for modeling two-phase phenomena. However, it is well established that the LS method yields more precise interface normals and curvature estimation which is owed to the mathematical properties of the level set function. This naturally made the study of propagating fronts with curvature-dependent speed an area that is unique to the LS framework. Formally, curvature-driven motion can be described as the evolution of an interface due to a self-generated velocity field $\mathbf{V}$ that is normal to the interface and proportional to its curvature. An example of this type of motion is shown in Figure 1 where a 6-petal star shape evolves into a circle due to its curvature; the troughs propagate outwards and the peaks propagate inwards with an interfacial speed proportional to $\kappa(\mathbf{x})$. Chopp and Sethian[6] studied hypersurfaces moving under a flow that depends on mean curvature using the LS method of Osher and Sethian.[2] In a later work, Chopp and Sethian[7] presented a discussion on various numerical schemes to model the motion of surfaces under the intrinsic Laplacian of curvature. In their work, they tested the curvature-driven motion of different shapes (convex and non-convex) using their proposed algorithm.

While using the LS method is suitable for the study of curvature-driven motion, the presence of high curvature regions imposes a more stringent requirement to solve the reinitialization equation (see References 8–10 for details on reinitialization). As the level set function evolves in time, its values begin to drift away from the originally signed distance function (SDF); the reinitialization step serves to retain that property of the level set before the proceeding advection. The frequency of the reinitialization step increases when regions of high curvature are present since the deviation of the level set from an SDF is more pronounced in those areas of the interface. This has an adverse effect on computational cost depending on the complexity of interface topology. Initial solutions in that direction evolved in the context of variational LS formulations, starting with the work of Li et al.[11] in the image processing community. They proposed an energy functional that regularizes the distance function as the level set deviates from an SDF. This is more commonly known as the distance regularized level set evolution (DRLSE) where the regularization term penalizes the deviation of the level set from an SDF, therefore, bypassing the need for an explicit reinitialization step. More recently, Alamé et al.[12] devised a robust numerical methodology to predict equilibrium interfaces by utilizing a DRLSE approach. Their variational LS formulation comes in the context of Gibbs free energy minimization where they demonstrate comparisons between analytical solutions of liquid-air interface equilibria and the proposed variational LS method.

In the light of increasing interest in physical phenomena involving the motion of a front whose speed depends on the local curvature (e.g., icing and crystal growth on wings, flame propagation in propulsion systems), a VOF-based front advection is of particular relevance. The work on variational LS provides a stepping stone for a VOF-based formulation that is accurate, efficient, intuitive, and easy to implement. Details of such a formulation do not currently exist in the literature to the best of the author's knowledge, and the prospect of the described approach is to open the door for the simulation and modeling of a wider range of applications that involve interfacial phenomena using VOF (e.g., bubble dynamics, phase change, etc.) with the additional advantage of discrete mass conservation and interface sharpness. The methodology is inherently applicable to geometric and algebraic VOF formulations since the advection equation is invariant with respect to the type of VOF, and since the principles through which the approach is derived do not change based on the method of interface representation. In the current work, we will utilize a conservative geometric volume-of-fluid method

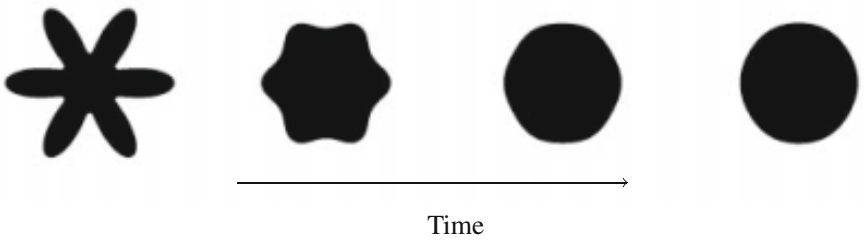

FIGURE 1 Evolution of a star shape under curvature-driven motion.[7]

whose interface reconstruction step is based on the analytic relations of Scardovelli and Zaleski,[13] and whose advection step is based on the work of Weymouth and Yue (WY).[4] We investigate constrained and unconstrained curvature-driven motion, with more emphasis on the mathematical definition of the constraint in the following sections. The evolution of curves and surfaces traditionally presented in the LS community is studied, and qualitative and quantitative comparisons are highlighted.

The paper is organized as follows: §2 will provide an overview of the volume-of-fluid method with canonical validation problems, §3 will present the proposed VOF approach for curvature flow and its details, §4 will demonstrate the results of different geometries evolving under curvature-dependent speed and the relevant discussions and lastly, the key findings of this work will be summarized in §5.

2 | VOLUME-OF-FLUID METHOD

Given a fluid boundary consisting of two phases that share an interface at a particular location $\mathbf{x}$, the phase indicator function $H(\mathbf{x}, t)$ is defined as

$$H(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is in fluid 1,} \\ 0 & \text{if } \mathbf{x} \text{ is in fluid 2.} \end{cases} \tag{1}$$

As the interface moves and deforms, the shape of each fluid changes, however, each fluid parcel is assumed to be immiscible in the absence of phase change so that the material derivative of H remains zero that is,

$$\frac{DH}{Dt} = \frac{\partial H}{\partial t} + \mathbf{u} \cdot \nabla H = 0. \tag{2}$$

This is equivalent to

$$\frac{\partial H}{\partial t} + \nabla \cdot (\mathbf{u}H) = H(\nabla \cdot \mathbf{u}) \tag{3}$$

using $\nabla \cdot (\mathbf{u}H) = \mathbf{u} \cdot \nabla H + H(\nabla \cdot \mathbf{u})$. The volume fraction C, often referred to as the color function, is used to define the spatial average of H in each computational cell such that

$$C_{i,j,k} = \frac{1}{\delta x \delta y \delta z} \int_{\Omega} H(\mathbf{x}, t)\, dV, \tag{4}$$

where Ω represents the computational domain, and the subscripts i, j, and k denote the index locations of the control volume on the Cartesian grid in the x, y, and z directions, respectively. This consequently results in a continuous distribution of volume fraction values that range between $0 < C < 1$. After integrating Equation (3) over the computational cell, and approximating C as a spatial average of H using Equation (4):

$$\delta x \delta y \delta z \frac{\partial C_{i,j,k}(t)}{\partial t} + \oint_{\partial\Omega} (\mathbf{u} \cdot \mathbf{n})\, H(\mathbf{x}, t)\, dA = \int_{\Omega} H(\mathbf{x}, t)(\nabla \cdot \mathbf{u})\, dV, \tag{5}$$

where $\partial\Omega$ is the cell boundary line and $\mathbf{n}$ is the outgoing unit normal. Note that for the case of incompressible flow $\nabla \cdot \mathbf{u}$ is equal to zero and the presence or absence of the right-hand side in Equation (5) is related to the implementation details of the chosen advection algorithm. The primary focus of this work is the geometric VOF method, therefore, the following subsection briefly describes algebraic VOF.

2.1 | Algebraic VOF

In algebraic VOF methods, C is obtained with a numerical approximation of H. Furthermore, the fluxes are computed algebraically without the need to geometrically reconstruct the interface. This makes algebraic VOF easier to implement

on arbitrary meshes, and also more computationally efficient. Amongst the most common approaches to compute the numerical fluxes in algebraic VOF are compressive schemes and THINC schemes.

In compressive schemes, an upwind donor cell, and a downwind acceptor cell are assigned to every cell face with respect to the underlying flow field. Then, based on the angle between the local interface normal and the cell face normal, donor-acceptor schemes switch between compressive downwind and diffusive upwind schemes. The list of common compressive schemes includes the high-resolution interface capturing scheme (HRIC),[14] compressive interface capturing scheme for arbitrary meshes (CICSAM),[15] switching technique for advection and capturing surfaces (STACS),[16] and the high-resolution artificial compressive formulation (HiRAC).[17] On the other hand, the THINC (tangent of hyperbola for interface capturing) scheme[18] involves the piecewise reconstruction of the volume fraction inside each cell by making use of the hyperbolic tangent function which is continuous with a controllable thickness. THINC does not require compressive techniques to eliminate interface smearing or explicit interface reconstruction, however, the cost of its three-dimensional implementation is still a topic of further research.

Although the type of curvature flow investigated in this work is applicable to algebraic VOF, algebraic VOF methods are generally less accurate than their geometric counterpart and have yet to be tested for large-scale simulations as indicated in the literature.

2.2 | Geometric VOF

In geometric VOF methods, the interface is approximated geometrically in a computational cell. This type of VOF is summarized in two fundamental steps: the reconstruction step and the advection step.

I **Reconstruction step**: The most commonly used reconstruction approach is the piecewise linear interface calculation (PLIC) scheme.[19] For PLIC methods, the reconstruction is a two-step procedure: the normal $\mathbf{m}$ (note that $\mathbf{n} = \mathbf{m}/|\mathbf{m}|$) is first determined using the values of C in a given cell and its neighborhood. The equation of the interface segment (a plane in three dimensions) is then written as

$$\mathbf{m} \cdot \mathbf{x} = m_x x + m_y y + m_z z = \alpha, \tag{6}$$

where α is adjusted until the area under the interface is equal to $\delta_x \delta_y \delta_z C_{i,j,k}$ and $\mathbf{x}$ is defined as the position vector extending from the origin of a computational cell to the interface.

II **Advection step**: Given the velocity field of the reconstructed interface, C of the reference phase needs to be shared across neighboring cells in a manner that guarantees volume conservation. Two advection algorithms currently exist to achieve that task: (1) split methods where a series of one-dimensional advections are performed in each spatial direction via operator-splitting of Equation (5), and (2) unsplit methods where a single complex flux calculation is done in cells containing the interface.

2.2.1 | Reconstruction method

The interface reconstruction employed uses the PLIC methodology, and Young's finite difference method to compute the local normal. Young's finite difference method was first developed by Youngs[20] and independently by Li.[21] The normal $\mathbf{m}$ is estimated as a gradient using finite difference approximation such that $\mathbf{m} = -\nabla_{\delta x_i} C$. In 2D, Young's method can be summarized by the following steps:

1. The normal $\mathbf{m}$ is evaluated at the four corners of the central cell (i, j); for example, the x- and y-components of $\mathbf{m}$ on the top-right corner are given by

$$m_{x:i+1/2,j+1/2} = -\frac{1}{2\delta x}(C_{i+1,j+1} + C_{i+1,j} - C_{i,j+1} - C_{i,j}), \tag{7}$$

$$m_{y:i+1/2,j+1/2} = -\frac{1}{2\delta y}(C_{i+1,j+1} - C_{i+1,j} + C_{i,j+1} - C_{i,j}), \tag{8}$$

and similarly for the other three corners. Note that the cell is assumed to be uniform.

2. The cell-centered value of $\mathbf{m}$ is then obtained by averaging the four cell-corners such that

$$\mathbf{m}_{i,j} = \frac{1}{4}(\mathbf{m}_{i+1/2,j+1/2} + \mathbf{m}_{i+1/2,j-1/2} + \mathbf{m}_{i-1/2,j+1/2} + \mathbf{m}_{i-1/2,j-1/2}). \tag{9}$$

After $\mathbf{m}_{i,j}$ is found, we use the analytical relations of Scardovellie and Zaleski[13] to find the slope and intercept of the interface segment.

2.2.2 | Advection method

For advection, the WY method provides an improvement to the traditional split-direction VOF advection algorithm by making it conservative. Consider the advection equation Equation (5) written in discrete form after integrating in space and time such that

$$C^{n+1}_{i,j,k} - C^{n}_{i,j,k} = -\sum_f F_f + \int_{t^n}^{t^{n+1}} \delta t \int_\Omega H(\nabla \cdot \mathbf{u}) dV, \tag{10}$$

where the first term in the RHS is the sum of the flux term F_f over all faces of $\mathbf{u}H$ defined by

$$F_f = \int_{t^n}^{t^{n+1}} \delta t \int_f u_f(\mathbf{x}, t) H(\mathbf{x}, t) dx, \tag{11}$$

where $u_f = \mathbf{u} \cdot \mathbf{n_f}$ and n_f is the unit normal vector pointing out of the face f. Therefore, the general 3D directionally split advection scheme can be written as

$$C^{(n,d+1)}_{i,j,k} - C^{(n,d)}_{i,j,k} = \Delta F_d + \int_{t^n}^{t^{n+1}} \delta t \int_\Omega H \frac{\partial u_d}{\partial x_d} dV = \Delta F_d + \underbrace{c_d \frac{\partial u_d}{\partial x_d}}_{\text{Dilatation term}}, \tag{12}$$

where d is the Cartesian index that indicates the direction of advection, ΔF_d is the net flux in that direction whose sign determines if the faces are donating or receiving regions, and c_d is the compression coefficient. Note that when alternating the directions, the sweeps are rotated cyclically to avoid preferential direction. The improvement to the existing split advection method comes in the context of meeting the following set of concurrent requirements—if for a given algorithm (1) the flux terms are conservative, (2) the divergence term sums to zero, and (3) no clipping or filling of a cell is needed due to the violation of $0 < C < 1$, then the algorithm must conserve C exactly. Previous VOF advection methods estimate the integral in the dilation term using the volume fraction C; this does not necessarily lead to obtaining zero dilatation for incompressible flow since the value of c_d may not be the same along the three Cartesian directions. Alternatively, Weymouth and Yue[4] suggested defining c_d as

$$c_d = H(C^n_{i,j,k} - 1/2) \tag{13}$$

such that $c_d = 0$ if $C^n_{i,j,k} < 1/2$ and $c_d = 1$ if $C^n_{i,j,k} \geq 1/2$—this definition guarantees directional invariance. Hence, as long as c_d is treated fully explicitly (using C from the previous time step), the simple advection method:

$$\Delta C = \frac{\delta t}{\delta x \delta y \delta z}\left(\Delta F_d + c_d \frac{\partial u_d}{\partial x_d}\right) \quad \text{for } d = 1, \ldots, N \tag{14}$$

satisfies flux conservation and zero velocity divergence. As for the requirement of no overfilling or over-emptying, this can be enforced with the following Courant number restriction

$$\delta t \sum_{d=1}^{N} \left| \frac{u_d}{\delta x_d} \right| < \frac{1}{2}. \tag{15}$$

More details can be found in Tryggvason et al.,[22] Alamé,[23] and Mirjalili et al.[24]

2.3 | VOF validation

This subsection presents validation problems for VOF (2D and 3D) under rigid body rotation and periodic vortex flow. These test cases were performed to determine the extent to which interface integrity is maintained before attempting to propagate the interface with curvature-dependent speed where the interface can significantly stretch. This represents a more systematic way to quantify mass errors originating from the derived advecting velocity (Section 3) and those inherent to the VOF method.

2.3.1 | Zalesak disc and star shape

Consider a unit square domain with an initial slotted circular fluid parcel centered at $(0.5, 0.75)$. The fluid parcel has a radius of $r = 0.15$ and its slot has dimensions $l = 0.2$ and $w = 0.06$. The slotted disc is allowed to cycle once inside the computational domain under the following velocity field

$$u(y) = 2\pi(y - y_c); \quad v(x) = -2\pi(x - x_c) \tag{16}$$

which describes rigid body rotation, where x_c and y_c are the coordinates of the center of rotation. Figure 2 shows the interface of the disc ($C \approx 0.5$) at three different instants of the cycle. The *CFL* number was fixed at 0.192, and the time-step chosen was equal to $\Delta t = 0.001, 0.0005, 0.00025$, and 0.000125 for grid sizes 32×32, 64×64, 128×128, and 256×256, respectively. The results in Figure 2 are for a 256×256 grid.

For the second 2D validation, consider a unit square domain with an 8-pointed star shape centered at $(0.5, 0.5)$ (see Section 4 for star-shape parameterization). The star shape is allowed to stretch clockwise until $t = T/2$ where the flow is reversed and the interface un-stretches back to its initial shape at $t = T$. The motion is induced by the following velocity field:

$$u(x,y) = -2\sin^2(\pi x)\ \sin(\pi y)\ \cos(\pi y)\ \cos\left(\frac{\pi}{T}t\right); \quad v(x,y) = 2\sin^2(\pi y)\ \sin(\pi x)\ \cos(\pi x)\ \cos\left(\frac{\pi}{T}t\right). \tag{17}$$

Figure 3 shows interface evolution ($C \approx 0.5$) of the prescribed geometry at $t = 0$, 1.2, and 2.0 where $t = 1.2$ represents the time at which maximum stretching of the star shape petals occurs. The *CFL* number was fixed at 0.064, and the time-step chosen was equal to $\Delta t = 0.001, 0.0005, 0.00025$, and 0.000125 for grid sizes 32×32, 64×64, 128×128, and 256×256, respectively. The results in Figure 3 are for a 256×256 grid.

A comparison between the initial and final interface shape is shown in Figure 4A,B for both 2D geometries and the deviation from the initial shape is measured via the L_1-norm error. The L_1-norm is defined as follows:

$$L_1 = \frac{1}{NxNy}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left|C^f(i,j) - C^0(i,j)\right|, \tag{18}$$

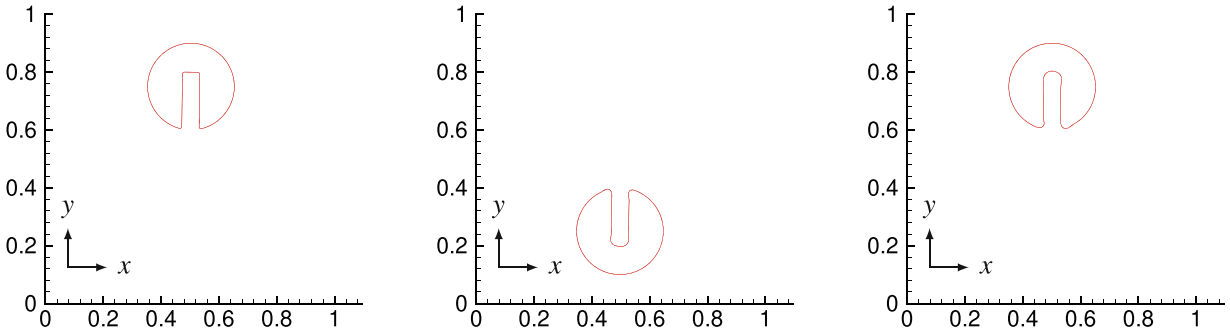

FIGURE 2 Evolution of the interface of the Zalesak disc under rigid body rotation. The images are snapshots of the interface in time from left to right where the disc rotates clockwise starting from its initial position at $(0.5, 0.75)$ and returning to the same position. Snapshots shown are for $t = 0.0$, 0.5, and 1.0. [Colour figure can be viewed at wileyonlinelibrary.com]

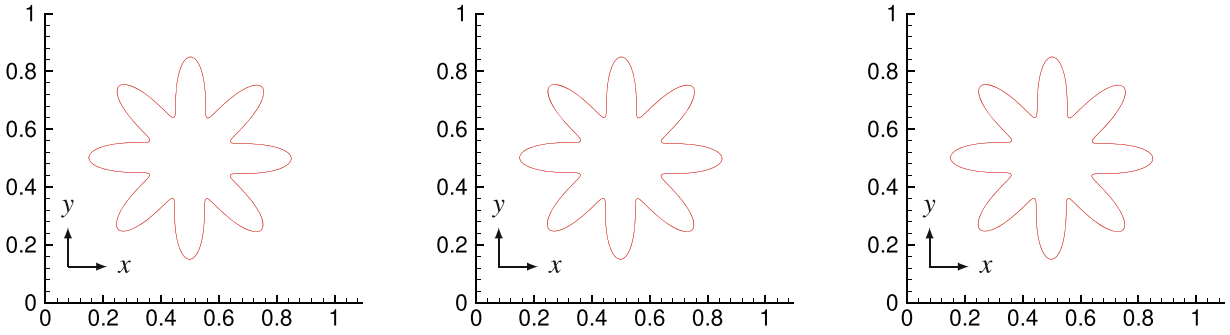

FIGURE 3 Evolution of the interface of a star-shape under vortex flow. The images are snapshots of the interface in time from left to right where the star is stretched clockwise until $T/2$ at which the flow is reversed and the initial shape is recovered. Snapshots shown are for $t = 0.0$, 1.2, and 1.0. [Colour figure can be viewed at wileyonlinelibrary.com]

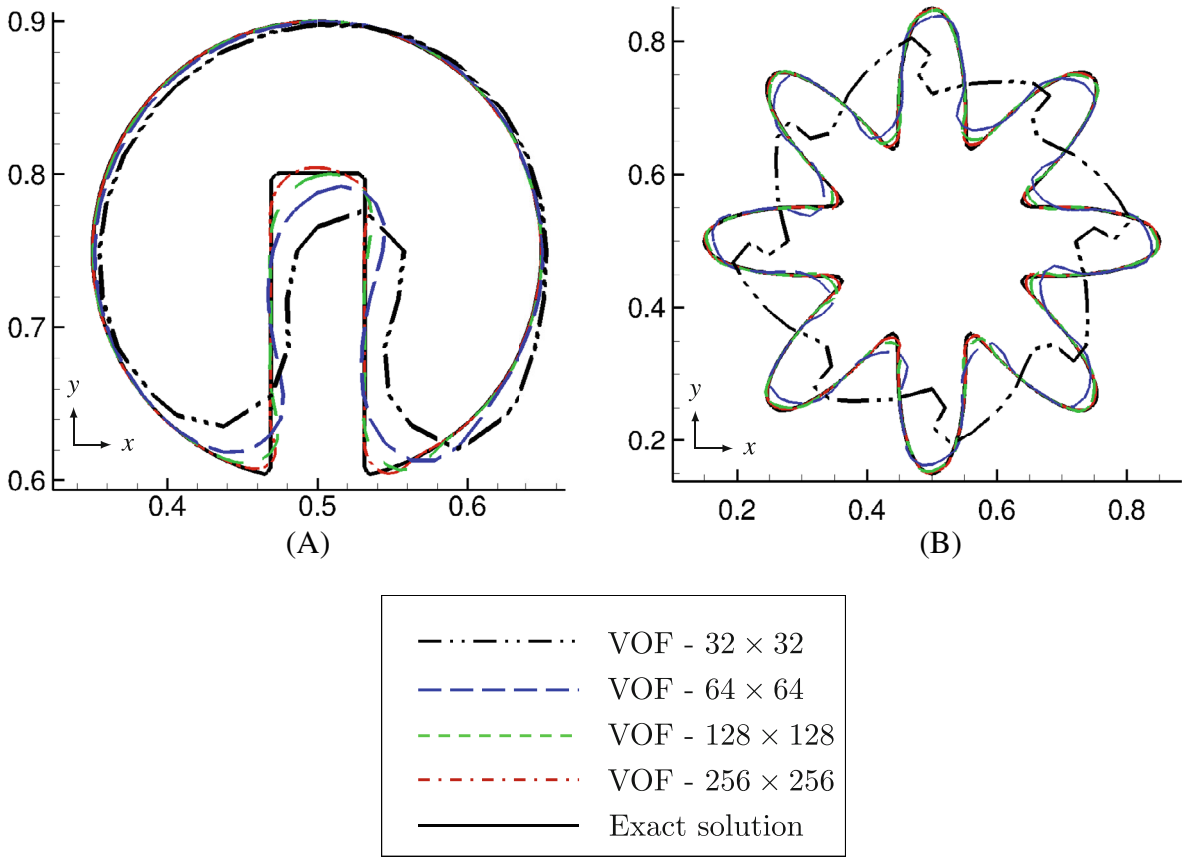

FIGURE 4 Comparison between initial and final interface position for different grid sizes for (A) Zalesak disc and (B) Star shape. [Colour figure can be viewed at wileyonlinelibrary.com]

where the superscripts f and 0 denote final and initial states, respectively. The order of convergence is given by the following relation,

$$n_{L_1} = \frac{log(L_{1,2N}) - log(L_{1,N})}{log(2)}, \tag{19}$$

where $L_{1,N}$ is the L_1 error using an $N_x \times N_y$ grid and $L_{1,2N}$ is the L_1 error for a $2N_x \times 2N_y$ grid. The order of convergence was found to be $n_{L_1} \approx 1.6$ for the Zalesak disc and $n_{L_1} = 2.5$ for the star shape as indicated in Figure 5A,B. The effect of

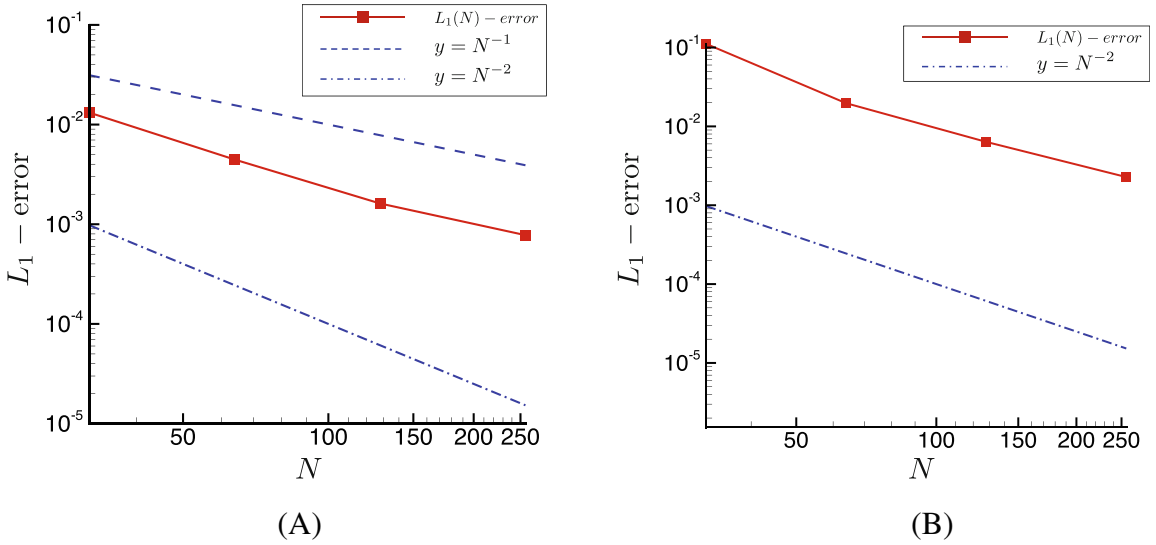

FIGURE 5 Shape error convergence upon grid refinement for (A) Zalesak disc and (B) Star shape. [Colour figure can be viewed at wileyonlinelibrary.com]

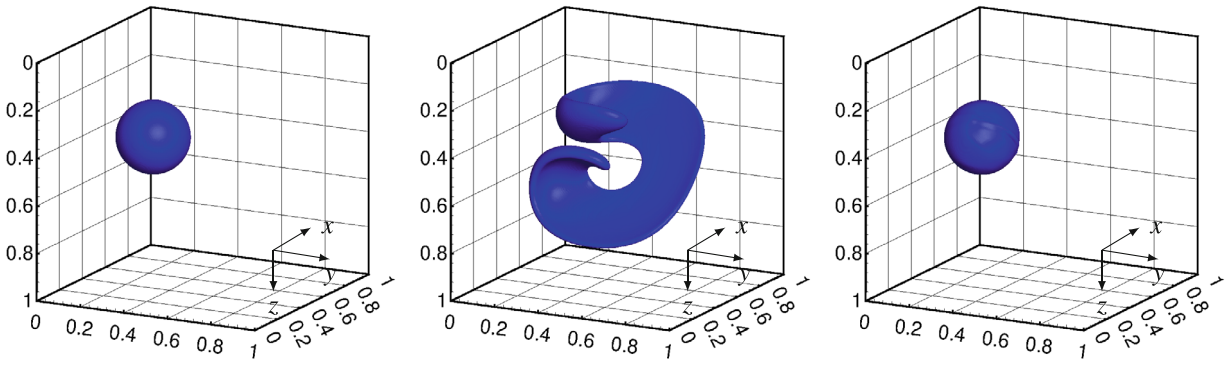

FIGURE 6 Evolution of a 3D sphere in a vortex field for a grid size 256^3 and a period $T = 3$. The images are snapshots of the interface in time going from left to right where the middle image is a snapshot at the instant of maximum strain. [Colour figure can be viewed at wileyonlinelibrary.com]

the first grid refinement is more apparent in Figure 4B where the final shape deviates significantly from the initial shape using the 32×32 grid.

2.4 | Sphere

The evolution of a 3D sphere is examined. A similar setup is investigated using a $256 \times 256 \times 256$ grid where the radius of the sphere is $r = 0.15$, its center is at $(0.35, 0.35, 0.35)$ in a unit cube, and the time period of advection is $T = 3$. The velocity field is directly given by:

$$u(x,y,z) = 2\sin^2(\pi x)\ \sin(2\pi y)\ \sin(2\pi z)\ \cos\left(\frac{\pi}{T}t\right), \tag{20}$$

$$v(x,y,z) = -\sin^2(\pi y)\ \sin(2\pi x)\ \sin(2\pi z)\ \cos\left(\frac{\pi}{T}t\right), \tag{21}$$

$$w(x,y,z) = -\sin^2(\pi z)\ \sin(2\pi x)\ \sin(2\pi y)\ \cos\left(\frac{\pi}{T}t\right). \tag{22}$$

The interface stretches significantly (see Figure 6) before returning to its initial position, however, mass is conserved to machine precision as expected.

3 | CURVATURE FLOW WITH VOF

The goal in this section is to derive an advecting velocity that is normal to the interface and minimizes its energy. Given the discontinuous nature of the color function C the variational analysis is carried out using the level set function $\phi(\mathbf{x}, t)$ which has the appropriate mathematical properties for this type of approach. The result of the analysis holds for VOF nonetheless and will be used accordingly since the advection equation is identical to that of LS with the type of indicator function being the only difference.

From the variational point of view, the minimum energy solution of an interface can be reached by minimizing its interfacial surface energy while respecting specified constraints (if any). If the area of a nonintersecting closed curve is to be minimized while maintaining a constant volume, the problem can be formulated mathematically as follows. Let the auxiliary energy functional $\mathscr{L}(\phi)$ be equal

$$\mathscr{L}(\phi) = \mathcal{E}(\phi) + \lambda G(\phi), \tag{23}$$

where $\phi(\mathbf{x}, t)$ is the level set indicator function, $\mathcal{E}(\phi)$ is the interfacial surface energy, $G(\phi)$ is the volume of the closed nonintersecting curve, and λ is the Lagrange multiplier dictating the strength of the volume conservation constraint. The interfacial surface energy and the volume can be expressed as

$$\mathcal{E}(\phi(\mathbf{x}, t)) = \int_{\Omega} \delta(\phi(\mathbf{x}, t)) |\nabla \phi(\mathbf{x}, t)| d\mathbf{x} \tag{24}$$

and

$$G(\phi(\mathbf{x}, t)) = \int_{\Omega} H(\phi(\mathbf{x}, t)) d\mathbf{x} = V_o, \tag{25}$$

respectively, where H is the Heaviside function and V_o is the initial volume. The evolution equation of ϕ can be written as a gradient flow that minimizes $\mathscr{L}(\phi)$ such that

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \mathscr{L}(\phi)}{\partial \phi}. \tag{26}$$

We will later show that the advection equation of ϕ can be retained using Equation (26). Upon taking the Fréchet derivative in the L^2 norm of $\mathscr{L}(\phi)$,

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon} [\mathscr{L}(\phi + \epsilon \chi) - \mathscr{L}(\phi)] = \left\langle \frac{\partial \mathscr{L}}{\partial \phi}, \chi \right\rangle = \left\langle \frac{\partial \mathcal{E}}{\partial \phi}, \chi \right\rangle + \lambda \left\langle \frac{\partial G}{\partial \phi}, \chi \right\rangle, \tag{27}$$

where χ is a test function in L^2. Therefore, $\frac{\partial \mathscr{L}}{\partial \phi} = \delta(\phi)\left[\lambda - \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right)\right]$ leading to the following form of the advection equation

$$\frac{\partial \phi}{\partial t} = \delta(\phi)\left[\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \lambda\right]. \tag{28}$$

To determine the value of the Lagrange multiplier, we enforce the invariance of the constraint with time evolution which mathematically translates to

$$\frac{dG(\phi)}{dt} = 0 \ \forall \ t \in [0, t_f],$$

where t_f is the final time. For $dG(\phi)/dt = 0$, λ is equal to $\overline{\kappa}$ (see Appendix A.3). In other words, the Lagrange multiplier dictating the applied constraint is the arithmetic mean interface curvature. This in itself is an interesting result; it implies that at every point on the interface, the local curvature balances the arithmetic mean interface curvature to get a minimal energy solution. Grayson[25] showed that any nonintersecting curve evolving under its own curvature must collapse into a circle (sphere in 3D). In the presence of a volume constraint, the interface will stop moving once it reaches a circle (sphere

in 3D) where $\kappa(\mathbf{x})$ is equal to $\overline{\kappa}$. This will later be verified by the numerical experiments carried out for different interface topologies.

In the context of LS, Equation (28) does not guarantee that the motion of the interface due to mean curvature is gradient descent of Equation (23) and requires an alternative definition of the norm used to compute the Frechét derivatives as indicated in Appendix A.1. However, the details of the overall evolution of the indicator function in Equation (28) are sufficient to rewrite the equation in terms of VOF variables and derive the directional velocity components accordingly. Furthermore, both VOF and LS define $\kappa(\mathbf{x})$ as $\nabla \cdot \left(\frac{\nabla I_{1,2}}{|\nabla I_{1,2}|} \right)$ where $I_1 = C$ and $I_2 = \phi$.

The Dirac delta $\delta(C)$ is typically approximated by $|\nabla C|$ in VOF methods which in this case is a requirement in order to ensure that the motion due to mean curvature is gradient descent. Hence, Equation (28) can be re-written in the following form,

$$\frac{\partial C}{\partial t} = (\kappa - \overline{\kappa})\ |\nabla C|. \tag{29}$$

The more general form of Equation (29) is given by

$$\frac{\partial C}{\partial t} + (V_n \mathbf{n} + V_t \mathbf{T}) \cdot \nabla C = 0, \tag{30}$$

where V_n is the component of velocity in the normal direction, otherwise known as the normal velocity, and V_t is the tangential component. Since the tangential component is identically zero for motion in the normal direction, Equation (30) reduces to

$$\frac{\partial C}{\partial t} + V_n |\nabla C| = 0. \tag{31}$$

Determining an expression for V_n is purely dependent on the physical phenomenon under study. For example, V_n will be proportional to interface curvature $\kappa(\mathbf{x})$ for curvature-driven motion, it will depend on Δp or ΔT for multiphase flows, and it can also be a constant a where $a > 0$ or $a < 0$. For the case where fluid motion is determined by pressure (e.g., cavitation), V_n takes the form of a phase change velocity that is a function of the pressure drop across the interface such that $V_n \propto \sqrt{p - p_{vap}}$[26] where p is the local fluid pressure and p_{vap} is vapor pressure.

Since maintaining interface sharpness is paramount, using the form of the advection equation in Equation (31) is disadvantageous as we will later show. Further, the form in Equation (31) does not represent an intuitive starting point for directionally-split advection with VOF. Using the property $\mathbf{n} \cdot \nabla C = (\nabla C / |\nabla C|) \cdot \nabla C = |\nabla C|$, Equation (29) can be written as

$$\frac{\partial C}{\partial t} - (\kappa - \overline{\kappa})\ \mathbf{n} \cdot \nabla C = 0 \tag{32}$$

leading to the original advection equation of C for incompressible flow

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0, \tag{33}$$

where $\mathbf{u} = -(\kappa - \overline{\kappa})\ \mathbf{n}$ and the unit normal to the interface is $\mathbf{n} = \nabla C / |\nabla C|$. For a directionally-split advection algorithm, the unit normal vector can be used to project the normal interface velocity in the three Cartesian directions such that

$$u_\Gamma = -(\kappa - \overline{\kappa}) \frac{\partial C}{\partial x} \frac{1}{|\nabla C|}; \quad v_\Gamma = -(\kappa - \overline{\kappa}) \frac{\partial C}{\partial y} \frac{1}{|\nabla C|}; \quad w_\Gamma = -(\kappa - \overline{\kappa}) \frac{\partial C}{\partial z} \frac{1}{|\nabla C|}, \tag{34}$$

where the subscript Γ designates the interface. The computation of the numerical fluxes uses the velocity vector at the cell face $\mathbf{u}_f$ which is calculated using simple averaging such that $\mathbf{u}_f = (\mathbf{u}_{icv1} + \mathbf{u}_{icv2})/2$. The sign of u_Γ, v_Γ and w_Γ in Equation (34) is based on the convention used to define the normal direction (inwards or outgoing). Therefore, omitting the negative sign that multiplies $(\kappa - \overline{\kappa})$ in each term is also valid as long as it is consistent with the direction of the normal. Furthermore, $\overline{\kappa}$ vanishes ($\lambda = 0$) for all components in Equation (34) in the case of free curvature flow.

3.1 | Curvature estimation

The curvature is defined as $\kappa(\mathbf{x}) = \nabla \cdot \mathbf{n}$ and is estimated using the height function (HF) method in this work. Similar to the reconstruction method of WY, where the normal is estimated using the local height function, the HF method was extended by Reference 27 to compute curvature.

The following is a 2D example to demonstrate a sample calculation. As shown in Figure 7, for $|n_y|$ greater than $|n_x|$, the summation of the height function is computed in the vertical direction. Similar to the definition used in the HF method for estimating normals, the height function is defined as:

$$\bar{y}_{i,j} = \sum_{l=-1}^{1} C_{i,j}\delta y_{i,j+l} \tag{35}$$

such that the curvature is given by

$$\kappa = \frac{n_y}{|n_y|}\frac{\bar{y}_{xx}}{(1+\bar{y}_x^2)^{3/2}}, \tag{36}$$

where $n_y/|n_y|$ gives the sign of curvature. The derivatives $\bar{y}_{xx}$ and $\bar{y}_x$ are computed using a central difference scheme which leads to second-order method convergence.

The height function in three dimensions is a straightforward extension, such that

$$\bar{y}_{i,j,k} = \sum_{l=-1}^{1} C_{i,j+l,k}\delta y_{i,j+l,k}. \tag{37}$$

However, the curvature expression is slightly different whereby

$$\kappa = \frac{n_y}{|n_y|}\left(\frac{\bar{y}_{xx}+\bar{y}_{zz}+\bar{y}_{xx}\bar{y}_z^2+\bar{y}_{zz}\bar{y}_x^2-2\bar{y}_{xz}\bar{y}_x\bar{y}_z}{(1+\bar{y}_x^2+\bar{y}_z)^{3/2}}\right). \tag{38}$$

The procedure for calculating the arithmetic mean interface curvature $\overline{\kappa}$ will be introduced and discussed in Section 4.2.

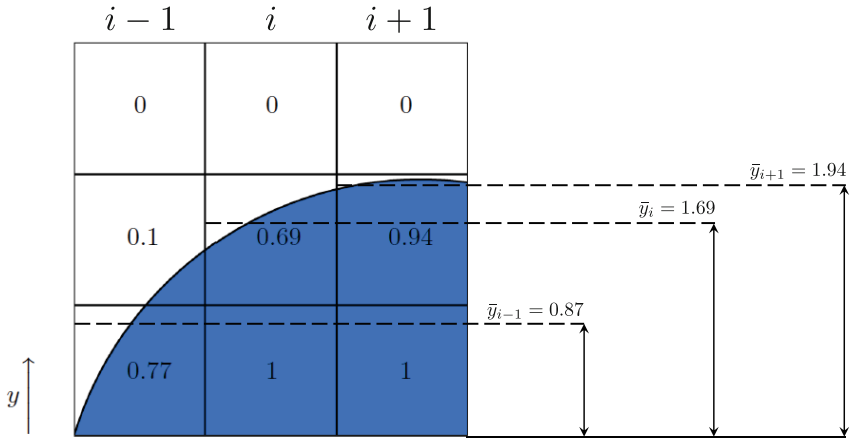

FIGURE 7 Example of the HF method. The normals satisfy $|n_y| > |n_x|$, therefore the summation is in the dominant vertical direction. Note that this example assumes that $\delta x = \delta y = 1$. [Colour figure can be viewed at wileyonlinelibrary.com]

4 | RESULTS AND DISCUSSION

In order to test the validity of the suggested projection in Equation (34), Rayleigh-Plesset (RP) bubble collapse was simulated. The velocity field driving the problem is interfacial in nature and serves as an adequate check given the presence of an analytical solution. In the absence of inertial effects, surface tension, and non-condensable gas (NCG), the RP equation reduces to

$$R\ddot{R} + \frac{3}{2}(\dot{R})^2 = \frac{P_B(t) - P_\infty(t)}{\rho_L}, \tag{39}$$

where $\Delta P = P_B(t) - P_\infty(t)$ and ρ_L is the liquid density. A constant ΔP was chosen and ρ_L was set to 1. Note that the value of ΔP is not of particular importance to the current problem since the comparison with the analytical solution is done on a non-dimensional basis. Furthermore, the solutions R and $\dot{R}$ are obtained from solving Equation (39) using fourth-order Runge-Kutta (RK4) time integration. Since $u(R, t) = \dot{R}$ acts normally to the interface, the velocity field can be written as

$$u_\Gamma = \dot{R}\frac{\partial C}{\partial x}\frac{1}{|\nabla C|}; \quad v_\Gamma = \dot{R}\frac{\partial C}{\partial y}\frac{1}{|\nabla C|}; \quad w_\Gamma = \dot{R}\frac{\partial C}{\partial z}\frac{1}{|\nabla C|} \tag{40}$$

similar to Equation (34).

A bubble of radius $r = 1$ is initialized at the center of a cubic domain with $L = 4$, and C evolves via $\dot{R}$ using two different forms of the advection equation for comparison. The first form of the advection equation is traditional in the LS method, and it represents a special case of the Hamilton-Jacobi equation such that

$$\frac{\partial C}{\partial t} = -\dot{R}|\nabla C|, \tag{41}$$

where the source term acting at the level of the interface is proportional to $\dot{R}$ and is parabolic in nature. The second form of the advection equation is Equation (33) where $\dot{R}$ contributes to the convection term through the velocity components in Equation (40).

The advection of C using Equation (41) shows fair agreement with the RP solution for the finest grid. However, the agreement quickly deteriorates for the coarser grids and is no longer acceptable as seen in Figure 8 (left). On the contrary, the advection of C using Equation (33) shows very good agreement with RP across all three grids with visually indistinguishable difference as seen in Figure 8 (right). The effect of the choice of advection can be further observed by looking at central slices of the color function at different times as the bubble collapses.

Following the snapshots of C in Figure 9 from left to right for the case of non-split VOF, the smearing of the interface becomes more pronounced as the bubble shrinks in size. For the same grid ($100 \times 100 \times 100$) on the other hand, and using directionally-split VOF, interface smearing is minimal compared to the smearing observed in Figure 9, as

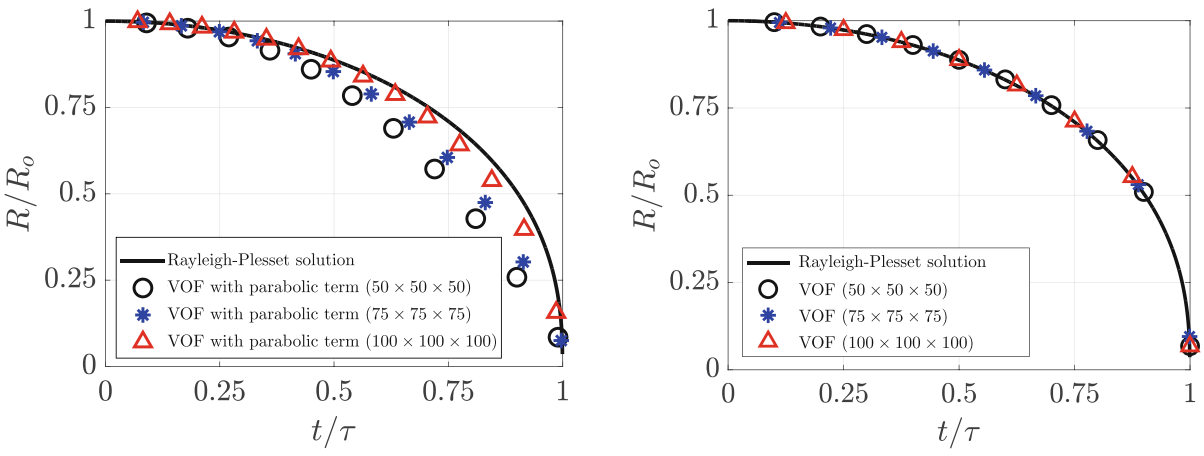

FIGURE 8 Comparison of bubble collapse profile for three different grid sizes using nondirectionally-split VOF (left) and directionally-split VOF (right). [Colour figure can be viewed at wileyonlinelibrary.com]

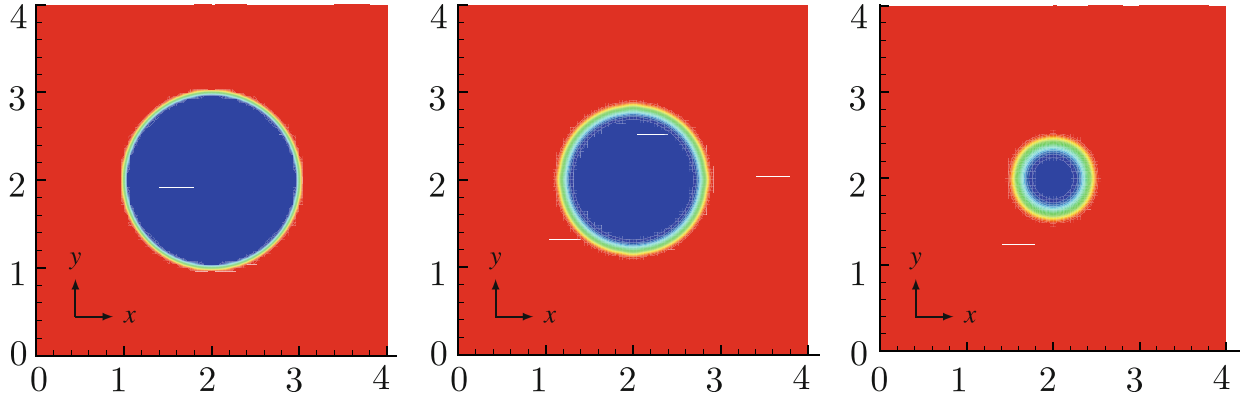

FIGURE 9 Snapshots of the color function as the bubble collapses using nondirectionally-split VOF. [Colour figure can be viewed at wileyonlinelibrary.com]

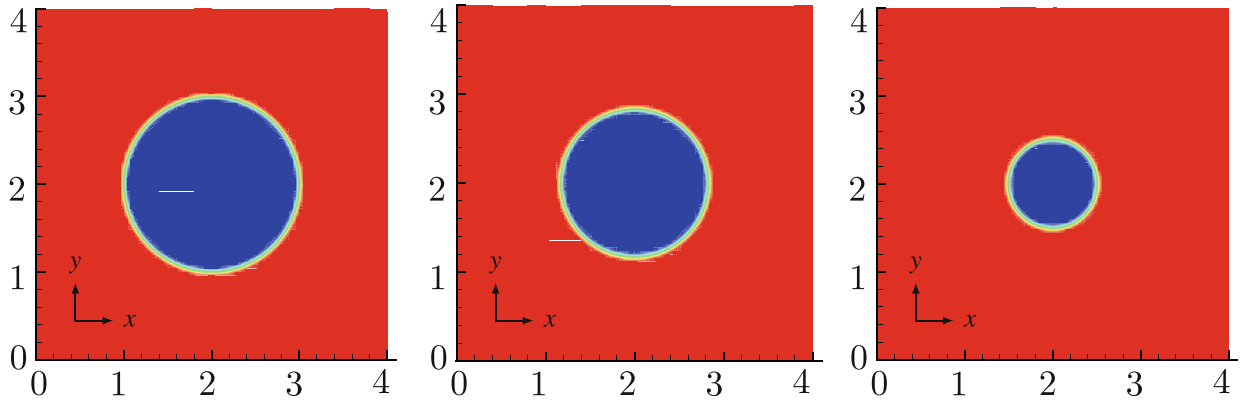

FIGURE 10 Snapshots of the color function as the bubble collapses using directionally-split VOF. [Colour figure can be viewed at wileyonlinelibrary.com]

shown in Figure 10. This comes as a direct effect of having fewer control volumes resolving the bubble as the diameter decreases.

The primary intention behind this exercise is to show that although $\dot{R}$ is identical for both forms of the advection equation, and although Equation (41) is consistent with the physical mechanism governing the motion of the interface in the normal direction, evolving the interface using the demonstrated directional projection of $\dot{R}$ is superior and less sensitive to grid refinement. This provides valuable input for what follows in the context of curvature-driven interface motion. Since the modeled normal velocity can represent different physics (e.g., phase change), the same conclusions can be extrapolated to other problems.

4.1 | Curvature-driven motion

In what follows, we investigate the curvature-driven motion of various geometries using the proposed approach where the interface evolves under the following curvature-dependent velocity components:

$$u_\Gamma = -\kappa \frac{\partial C}{\partial x}\frac{1}{|\nabla C|}; \quad v_\Gamma = -\kappa \frac{\partial C}{\partial y}\frac{1}{|\nabla C|}; \quad w_\Gamma = -\kappa \frac{\partial C}{\partial z}\frac{1}{|\nabla C|}. \tag{42}$$

The interface is defined using the zeroth level set $\phi(\mathbf{x},\mathbf{0})$ which is then used to construct the color function for VOF. We also make use of Boolean operators to initialize more complex geometries which represent a union of multiple level set functions.

4.1.1 | Pointed star

Consider a pointed star given by the following parameterized curve

$$\gamma(s) = [\mathcal{A} + \mathcal{B}\ \cos(\mathcal{K} \cdot 2\pi s)][\cos(2\pi s), \sin(2\pi s)] \quad \text{for} \quad s \in [0, 1], \tag{43}$$

such that the zeroth level set, defined from polar coordinates, is given by

$$\phi(\mathbf{x}, 0) = ||\mathbf{x} - \mathbf{x_c}|| - \left\{ \mathcal{A} + \mathcal{B}\ \cos\left[\mathcal{K} \cdot \arctan\left(\frac{y - y_c}{x - x_c} \right) \right] \right\}, \tag{44}$$

where $\mathcal{A}$ is the shift of the star petals from the center, $\mathcal{B}$ is a scaling factor for the star size, $\mathcal{K}$ is the number of petals, and $\mathbf{x_c}$ is the offset parameter. The petals of the star are shifted from its center by 25 units of length ($\mathcal{A} = 25$) which effectively increases the relative size of the petal shape in comparison to the main body, the scaling factor $\mathcal{B}$ is chosen to be 10, and the number of petals is 8 ($\mathcal{K} = 8$). The computational domain is a square with a side length of 100 units and a grid resolution of 200×200. The time-step is $\Delta t = 0.05$ and the star-shaped level curve is centered at (50, 50). Figure 11 shows

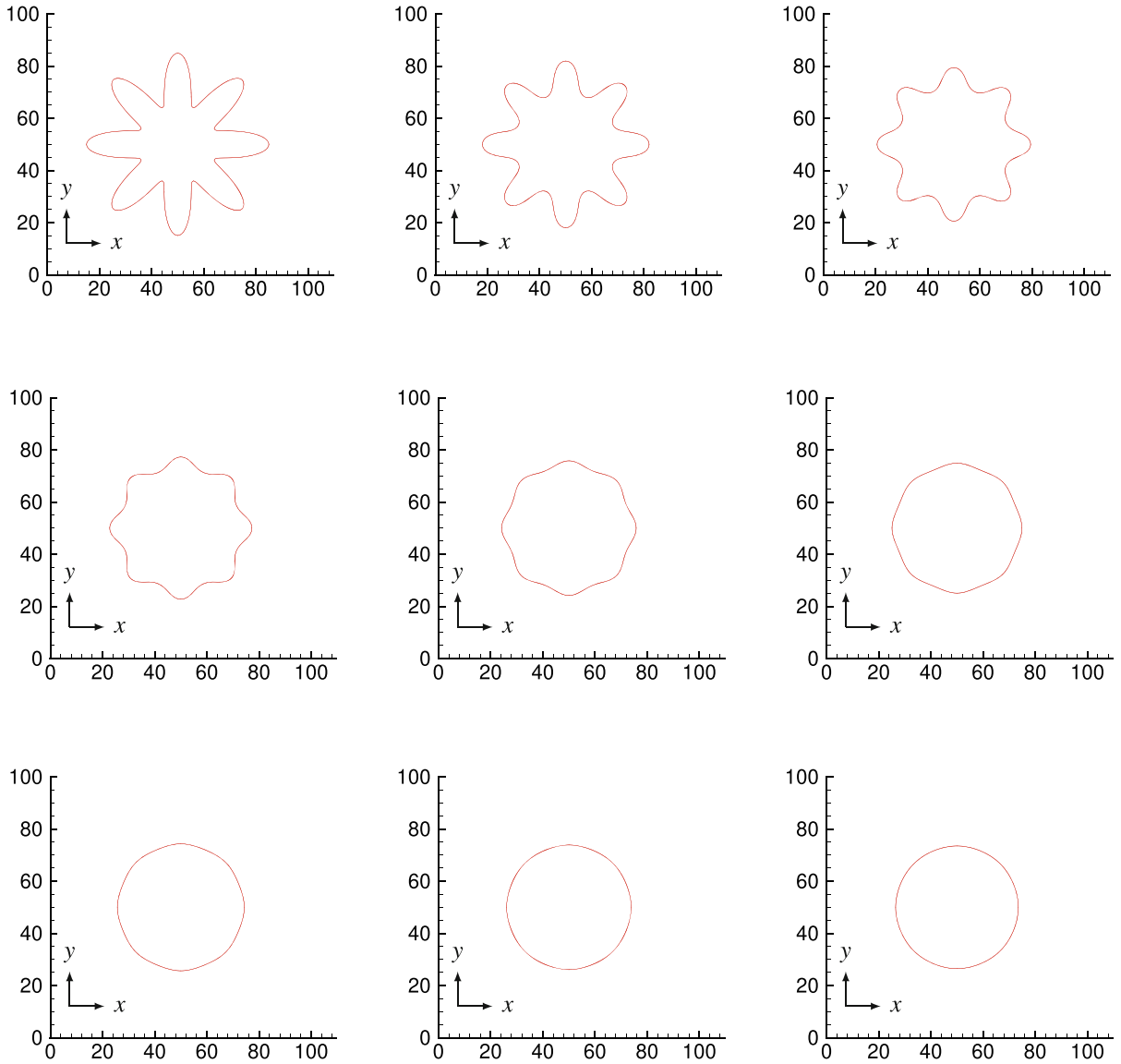

FIGURE 11 Evolution of the interface of a star-shape under curvature-driven flow. The images are snapshots of the interface in time from left to right, top to bottom where the star collapses unto a circle for $t = 0.0$, 7.5, 15.0, 22.5, 30.0, 37.5, 45.0, 52.5, and 60.0. [Colour figure can be viewed at wileyonlinelibrary.com]

the progression of the color function front under curvature-driven flow for $t = 0.0, 7.5, 15, 22.5, 30.0, 37.5, 45.0, 52.5, 60.0$. The interface collapses to a circle under the curvature-driven motion due to the collapse of the peaks $(+\kappa)$ inwards and the propagation of the troughs outwards $(-\kappa)$. Similar results have been reported in References 12 and 28 for level set methods. We note that for geometries involving sharp corners (e.g., pointed star), the smoothness of the interface at those locations is paramount to the physical evolution of the full geometry. This is particularly important in the context of VOF since errors in normal estimation will grow with every directional sweep leading eventually to the rupture of the interface if the corners are not well-resolved.

The results in Figure 11 were compared with standard LS with reinitialization for the same problem parameters. Figure 12 shows a comparison between the interface contour of VOF and LS with and without reinitialization. Reinitialization has no observable effect on the final shape of the level set and the final result is consistent for both interface capturing methods.

Since the pointed star should evolve into a perfect circle under curvature-driven motion, the roundness (sphericity in 3D) of the final contour was calculated for both methods. Roundness is the measure of how closely the shape of an object approaches that of a mathematically perfect circle, and among the most common roundness error definitions is circularity which is often used in digital image processing. Circularity is the ratio of the perimeter squared of shape to its area such that

$$\mathcal{S} = \frac{Perimeter^2}{4\pi \times Area}. \tag{45}$$

The ratio between the $\mathcal{S}_{LS}$ and $\mathcal{S}_{VOF}$ was found to be less than 1 which indicates higher roundness with VOF and a contour shape that more closely approximates a perfect circle. Note that the variable N_{RI} is Figure 12 refers to the number of reinitialization steps.

4.1.2 | Wound spiral

Consider the curvature-driven motion of a wound spiral given by the following parameterization

$$\theta = 2\pi D\sqrt{s} \quad \text{for} \quad s \in [0, 1], \tag{46}$$

such that $s = (k + a)/(n_p + a)$, where k is an integer that loops over the total number of points n_p, and the constant a determines the shape of the spiral head in the center. The location of the points describing the spiral is defined by

$$x_p = x_c + s_f\left[\frac{D\sqrt{s}\,\cos(\theta)}{1+D}\right] \quad \text{and} \quad y_p = y_c + s_f\left[\frac{D\sqrt{s}\,\sin(\theta)}{1+D}\right], \tag{47}$$

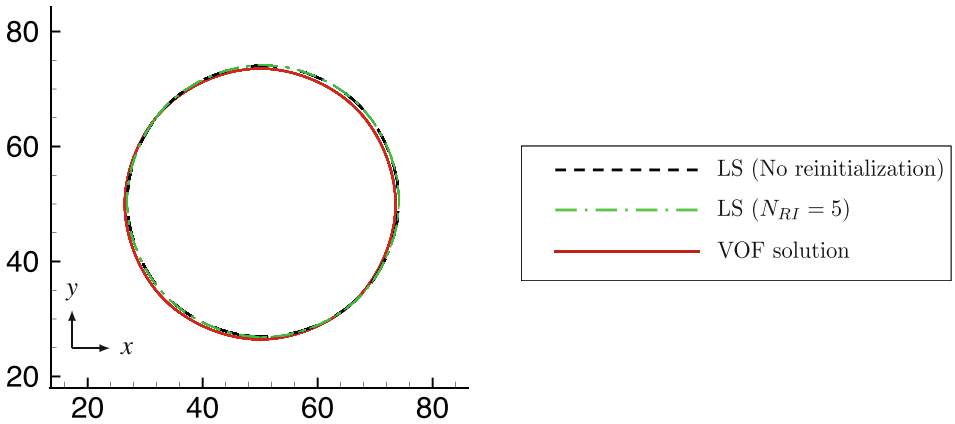

FIGURE 12 Comparison of final contour shape between VOF and LS with and without reinitialization for the case of the pointed star. [Colour figure can be viewed at wileyonlinelibrary.com]

where $\mathbf{x_c}$ is the location of the domain center, s_f is the scaling factor for the spiral size, and D is the number of spirals. We define a distance function $d(\mathbf{x}) = ||\mathbf{x} - \mathbf{x_p}||$ such that the zeroth level set has the following form

$$\phi(\mathbf{x}, 0) = d(\mathbf{x}) - w, \tag{48}$$

where w is the width of the spirals. The computational domain is a square of side length equal to 100 units and a grid size of 100×100. The time-step is $\Delta t = 0.05$, the level curve center $\mathbf{x_c} = (50, 50)$, $n_p = 400$, $D = 2.5$, $a = 3$, and $s_f = 50$. The simulation is stopped after 6000 iterations ($t_{final} = 300.0$) and snapshots of the interface evolution are shown in Figure 13 for $t = 0.0, 37.5, 75.0, 112.5, 150.0, 187.5, 225.0, 262.5$, and 300.0. Problem parameters describing spiral geometry were chosen to match[12] for direct comparison. The spiral unwinds under curvature-driven motion, and its size decreases in the process until it eventually acquires a bean shape that later collapses rapidly. The interface maintains its sharpness even at the last stages of collapse which is consistent with the bubble collapse problem discussed earlier.

Similar to the pointed star case, the shape of the final contour of the wound spiral is compared between VOF and LS as shown in Figure 14. The effect of no reinitialization is immediately observable for the LS method where the contour deviates completely from the VOF solution. As the number of reinitializations increases, the contour obtained from

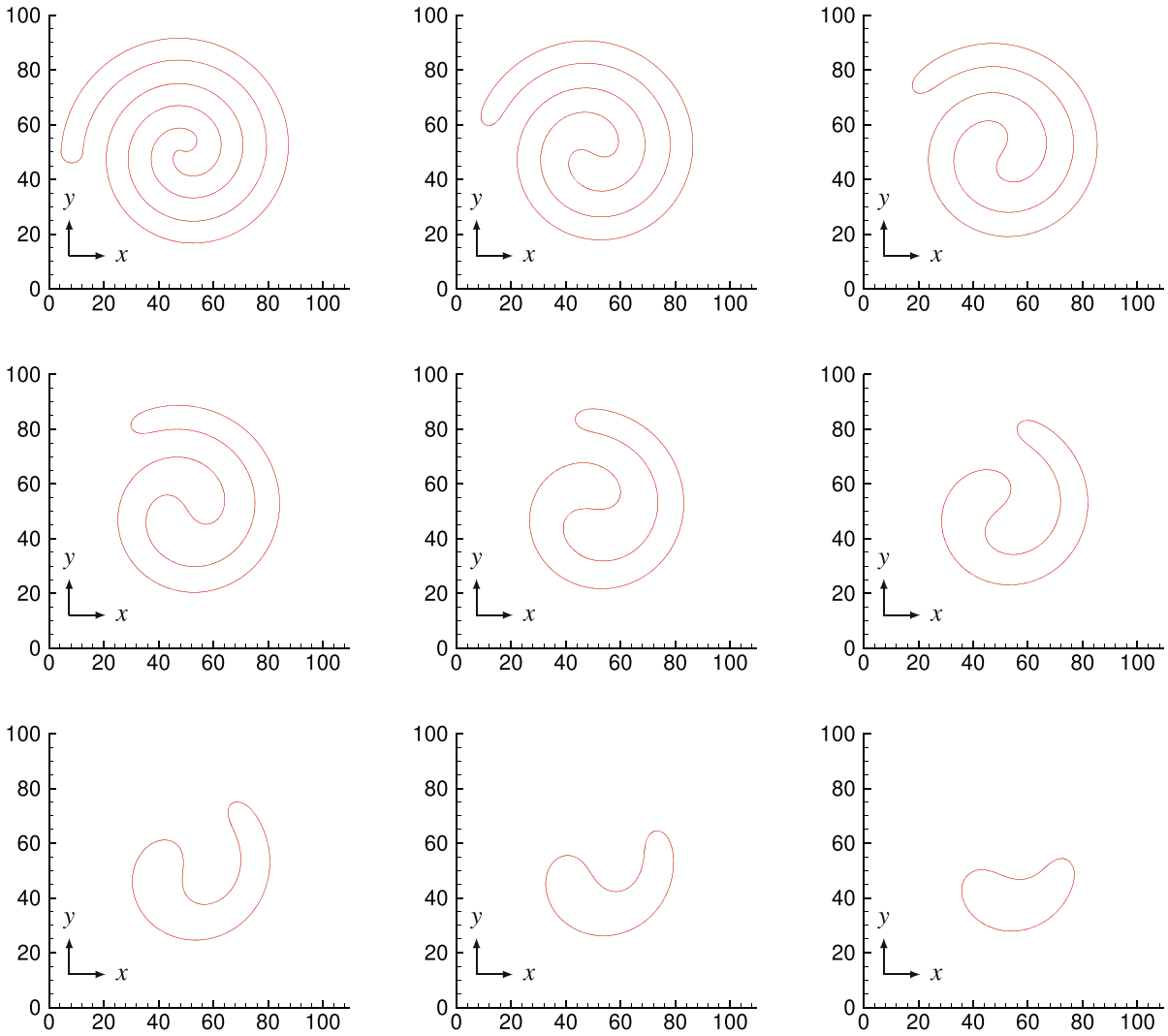

FIGURE 13 Evolution of the interface of a spiral under curvature-driven flow. The images are snapshots of the interface in time from left to right, top to bottom where the spiral unwinds for $t = 0.0, 37.5, 75.0, 112.5, 150.0, 187.5, 225.0, 262.5$, and 300.0. [Colour figure can be viewed at wileyonlinelibrary.com]

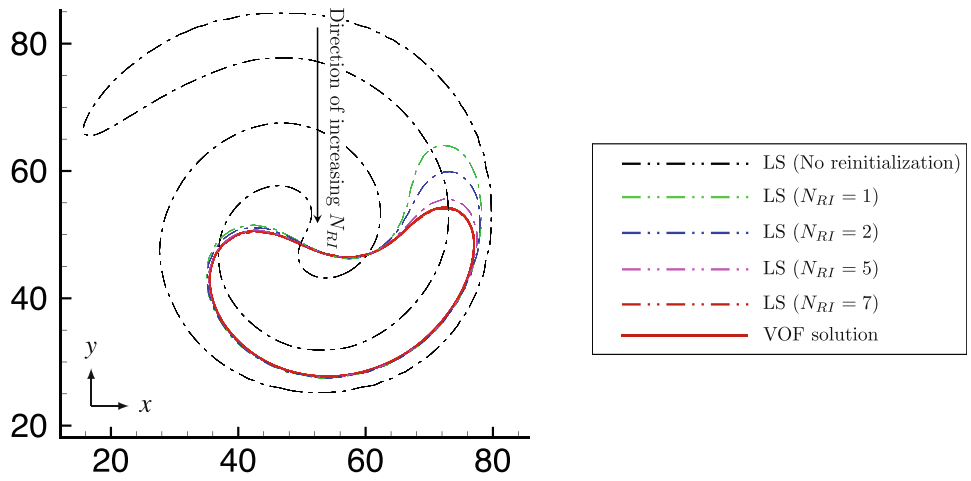

FIGURE 14 Comparison of final contour shape between VOF and LS with and without reinitialization for the case of the wound spiral. [Colour figure can be viewed at wileyonlinelibrary.com]

LS converges to the VOF solution at $N_{RI} = 7$. What is worth noting is that the result obtained with VOF closely agrees with that obtained with DRLSE in Reference 12 for the same problem setup. Moreover, there is no clear criterion that determines the number of reinitialization steps needed to converge to the expected solution, this as a result makes VOF advantageous and more robust for such problems.

4.1.3 | Dumbbell

In this example, three-dimensional geometries are introduced. A dumbbell shape is initialized via the union of the level curves of two spheres and a cylinder such that

$$\begin{cases} \phi_{sphere,R} = \sqrt{(x - x_c + o)^2 + (y - y_c)^2 + (z - z_c)^2} - r, \\ \phi_{sphere,L} = \sqrt{(x - x_c - o)^2 + (y - y_c)^2 + (z - z_c)^2} - r, \\ \phi_{cylinder} = max\left[(|x - x_c| - o), \left(\sqrt{(y - y_c)^2 + (z - z_c)^2} - w\right)\right], \end{cases} \tag{49}$$

and the zeroth level set is given by

$$\phi(\mathbf{x}, 0) = min[\phi_{sphere,R}, \phi_{sphere,L}, \phi_{cylinder}]. \tag{50}$$

The coordinates of the domain center are $\mathbf{x_c}$, o is the center-to-center distance between the cylinder and each of the two spheres, r is the radius of the spherical shells, and w is the radius of the cylinder. The computational domain is a cube of side length equal to 100 units and grid size of $200 \times 200 \times 200$. The time-step Δt is 0.01, the center of the dumbbell level curve is $\mathbf{x_c} = (50, 50, 50)$, $r = 10$, $w = 5$, and $o = 20$. The total number of iterations is 1600 ($t_{final} = 16.0$) and snapshots of interface evolution are shown in Figure 15 for $t = 0.0, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0$, and 16.0. The problem parameters describing dumbbell geometry are again chosen to match the reported values in Reference 12 for better comparison. Figure 15 shows that the VOF methodology presented is capable of capturing all three unique features of the problem (pinching, merging, separation) without any special algorithmic treatment of the interface. The dumbbell handle initially collapses faster than the spheres on both ends due to higher curvature. The higher shrinkage rate at the handle then causes pinch-off whilst the spherical shells are shrinking simultaneously at a slower rate. The end result is two teardrop surfaces that will continue to collapse due to their curvature (see Figure 15).

Upon running the problem with a coarser grid size of $100 \times 100 \times 100$, parasitic "wisps" appear in the pinch-off region. Arrufat et al.[29] define these wisps as cells with tiny values of $1 - C_{i,j,k}$ in fluid 1 or $C_{i,j,k}$ in fluid 2. Hence, their appearance for the dumbbell problem is not surprising. Reference 29 suggest using a higher clipping tolerance to bypass that problem, however, there is still no clear criterion for choosing that tolerance as they had reported. The wisps did not decrease

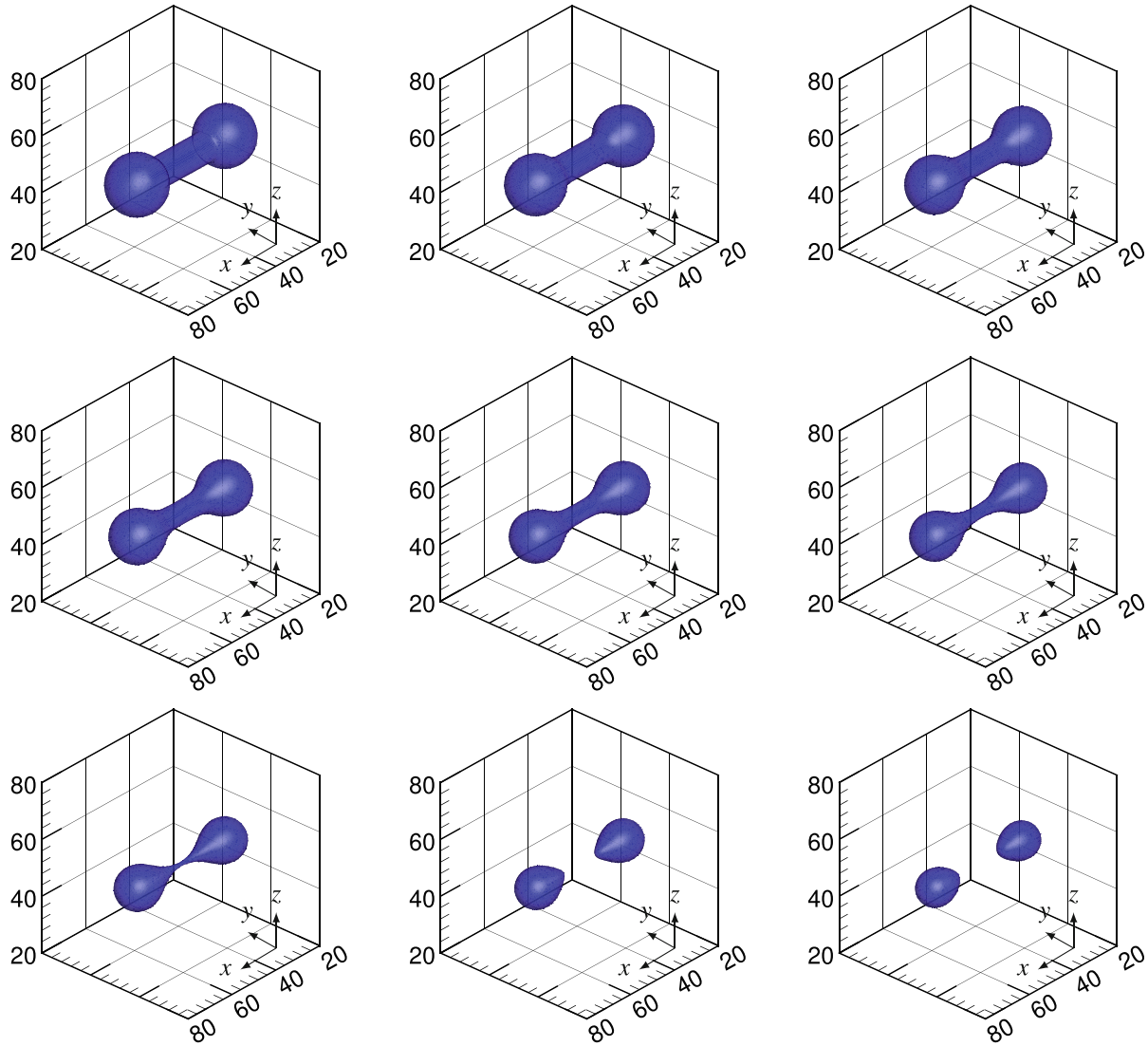

FIGURE 15 Evolution of a dumbbell-shaped surface under curvature flow. The images are snapshots of the interface in time from left to right, top to bottom where the handle of the dumbbell shrinks faster than the spherical shells due to its high curvature leading to a pinch-off for $t = 0.0$, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, and 16.0. [Colour figure can be viewed at wileyonlinelibrary.com]

upon varying the clipping tolerance, nonetheless, they disappeared upon doubling the grid size. Note that our clipping tolerance ranged between $\epsilon = 10^{-13}$ and $\epsilon = 10^{-3}$ for the trial runs.

Before doubling the grid size and varying the clipping tolerance, the advection of the volume fraction was also performed with CIAM (Calcul d'Interface Affine par Morceaux)[21,29] alongside the WY method to compare the pinch-off region. As shown in Figure 16, the pinch-off wisp does not appear for the $100 \times 100 \times 100$ grid using CIAM. This has been found to be the case for problems where interface stretching is excessive as for the case presented here and can be attributed to the algebraic nature of the WY advection. Nonetheless, the WY advection has better conservation properties compared to its counterparts making it the method of choice for this work.

The results for the dumbbell shape were compared with those obtained with the DRLSE method of Reference 12. Figure 17 shows isocontours of the interface as the dumbbell evolves in time using VOF (left) and the variational LS method (right). VOF shows good agreement with variational LS with a minor difference in the shape of the pinch-off region. Iso-surfaces of both methods at the last stages of pinch-off were also compared as shown in Figure 18 for DRLSE and Figure 19 for VOF. The three main topological phenomena that characterize the dumbbell problem are pinching, merging, and separation, as mentioned earlier. VOF is shown to capture all three features with good agreement compared to the variational LS method.

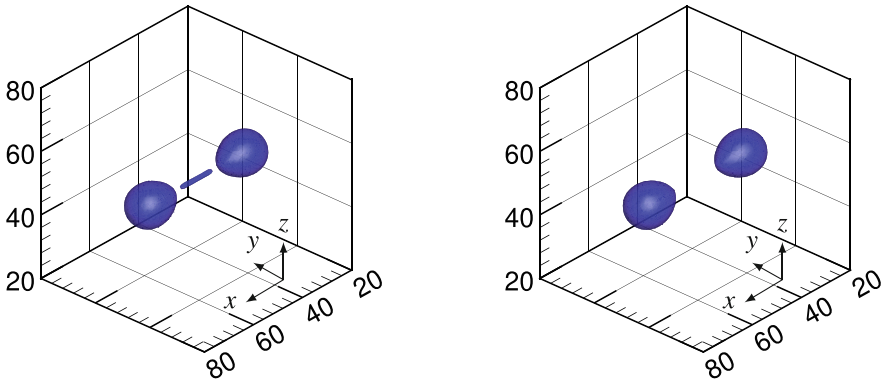

FIGURE 16 Instant of dumbbell pinch-off using WY (left) and CIAM (right). [Colour figure can be viewed at wileyonlinelibrary.com]

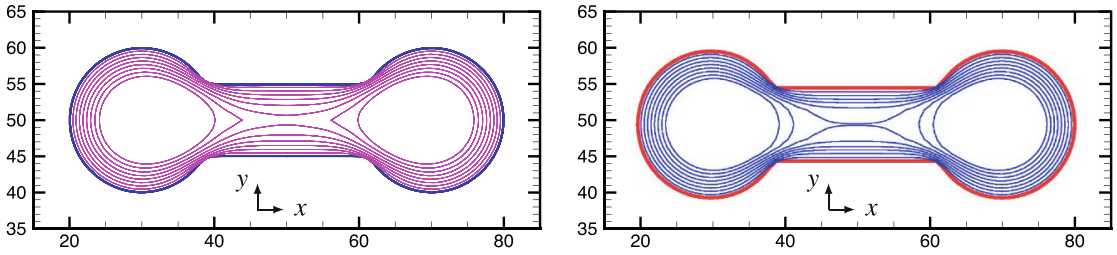

FIGURE 17 Z-plane slice showing the evolution of dumbbell interface in time with VOF (left) and DRLSE[12] (right). [Colour figure can be viewed at wileyonlinelibrary.com]

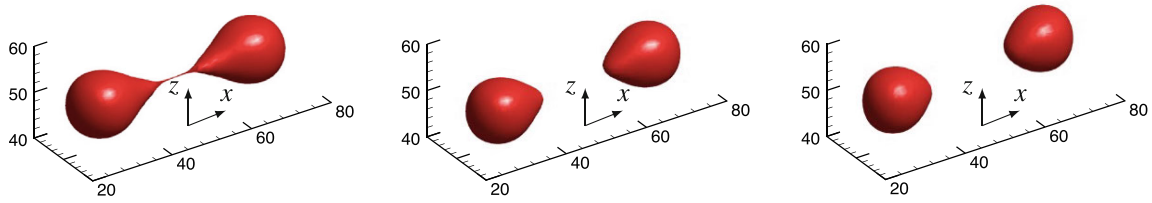

FIGURE 18 Snapshots of final stages of dumbbell pinch-off using DRLSE.[12] [Colour figure can be viewed at wileyonlinelibrary.com]

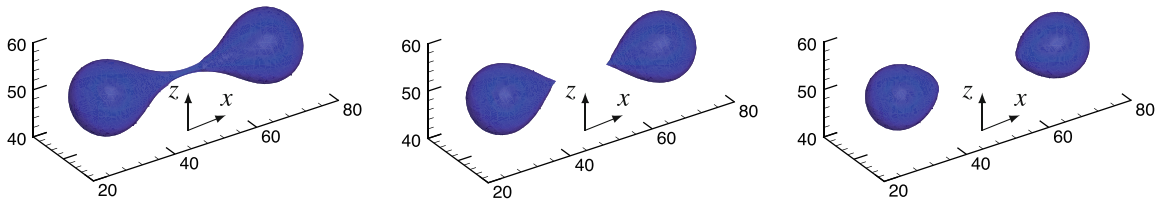

FIGURE 19 Snapshots of final stages of dumbbell pinch-off using VOF. [Colour figure can be viewed at wileyonlinelibrary.com]

4.1.4 | Composite geometry

The shape under consideration in this section is a composite geometry. This makes interface evolution more challenging from the numerical point of view due to the higher chance of developing singularities at junctions where $\kappa(\mathbf{x})$ is ill-defined. The closed geometry is created via the union of the level curves of three prolate spheroids; a main stem spheroid oriented in the z-direction and two secondary spheroids oriented in the x and y-directions, respectively. In general, the level curve of a tri-axial ellipsoid centered at point C with semi-axes a, b, and c aligned along the coordinate axes is given by

$$\phi_{ellipsoid} = \frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} + \frac{(z-z_c)^2}{c^2} - 1. \tag{51}$$

Using the definition in Equation (51), the level curves defining the geometry of this problem can be expressed as

$$\begin{cases} \phi_{spheroid,x} = \frac{(x-x_c)^2}{a_1^2} + \frac{(y-y_c)^2+(z-z_c)^2}{c_1^2} - 1, \\ \phi_{spheroid,y} = \frac{(x-x_c)^2+(z-z_c)^2}{a_2^2} + \frac{(y-y_c)^2}{c_2^2} - 1, \\ \phi_{spheroid,z} = \frac{(x-x_c)^2+(y-y_c)^2}{a_3^2} + \frac{(z-z_c)^2}{c_3^2} - 1, \end{cases} \tag{52}$$

and the zeroth level set is given by

$$\phi(\mathbf{x}, 0) = min[\phi_{spheroid,x}, \phi_{spheroid,y}, \phi_{spheroid,z}], \tag{53}$$

where the subscripts x, y, and z refer to the direction of the symmetry axis of the spheroid, a_i is the equatorial radius of the spheroid, and c_i is the distance from center to pole along the symmetry axis. For $c_i > a_i$ the resulting geometry is a prolate spheroid. The computational domain is a cube of side length equal to 100 units and grid size of $100 \times 100 \times 100$. The time-step $\Delta t = 0.05$, the center of the geometry is at $\mathbf{x_c} = (50, 50, 50)$, $a_1 = 25$, $a_2 = 7.5$, $a_3 = 10$, $c_1 = 7.5$, $c_2 = 25$, and $c_3 = 35$, and the total number of iterations is 1200 (t_final = 60.0). 2D snapshots of interface evolution are shown in Figure 20, and snapshots of the 3D interface evolution are shown in Figure 21 for $t = 0.0, 6.25, 12.5, 18.75, 25.0, 31.25, 37.5, 43.75$, and 50.0. The four arms parallel to the bottom plane propagate faster toward the stem of the geometry due to higher curvature as the junctions at the stem evolve outwards simultaneously. The difference in front propagation rate between the symmetry axes eventually leads to an egg-like shape that continues to shrink until it disappears at $t = 60.0$ (not

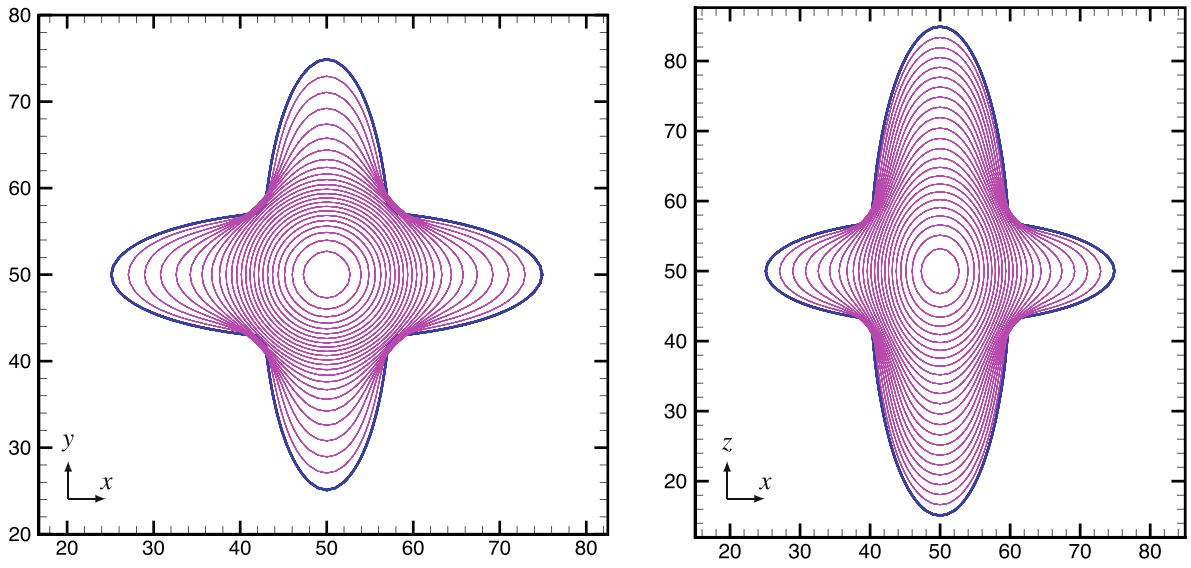

FIGURE 20 Z-plane slice (left) and Y-plane slice (right) showing the evolution of the interface of the composite geometry in time. [Colour figure can be viewed at wileyonlinelibrary.com]

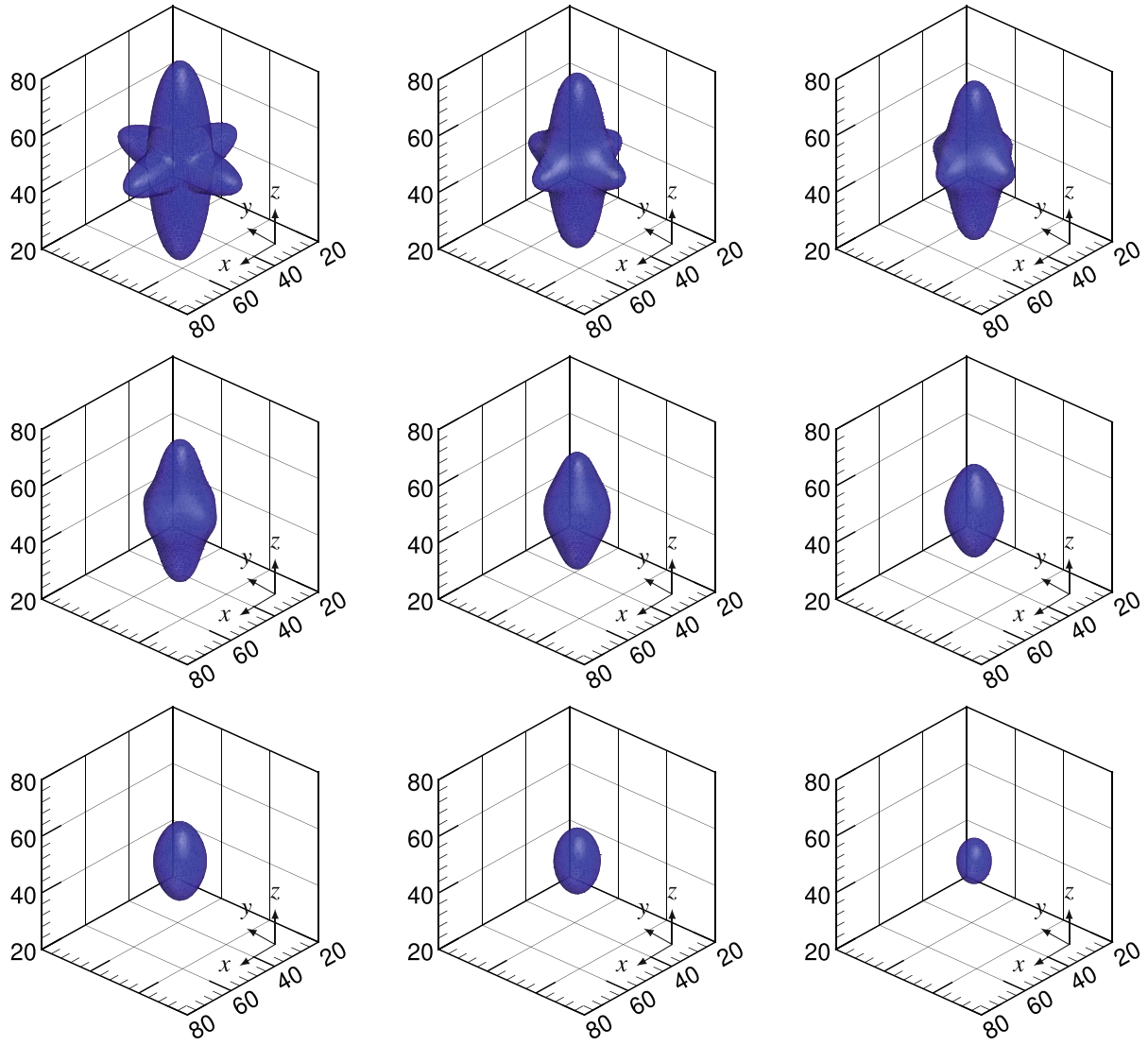

FIGURE 21 Evolution of composite geometry under curvature flow. The images are snapshots of the interface in time from left to right, top to bottom where the horizontal heads shrink faster than the vertical spheroid due to higher curvature for $t = 0.0$, 6.25, 12.5, 18.75, 25.0, 31.25, 37.5, 43.75, and 50.0. [Colour figure can be viewed at wileyonlinelibrary.com]

shown in Figure 21). Since the role of curvature is analogous to the role of viscosity in the corresponding hyperbolic conservation law as explained by Reference 6, the singularities created from the union of three spheroids are partially smoothed out leading to a stable interface evolution until the closed geometry collapses to a singularity beyond $t = 50.0$ (not shown).

4.1.5 | Artificial surface

In this example, the evolution of an artificial open surface mimicking a realistic material interface is simulated. The surface is constructed by summing a series of trigonometric functions such that

$$F_1 = A\ \cos(\Theta x)\ \cos(\beta y) \quad \text{and} \quad F_2 = A\ \cos\left(\frac{\Theta x}{3}\right)\ \sin\left(\frac{\beta y}{3}\right) + L_z, \tag{54}$$

where A is the amplitude, Θ and β are the function phases, and L_z is a shifting length equal to the domain length in the z-direction. The zeroth level set describing the artificial surface is therefore expressed as

$$\phi(\mathbf{x}, 0) = F_1 + F_2 - 2z. \tag{55}$$

The level set is then converted to a color function as in the previous cases. Figure 22 shows the topology of the constructed surface for $A = 5$, $\Theta = 0.25$, $\beta = 0.25$, and $L_z = 100$ and Figure 23 shows the color function representation of the surface viewed from different planes.

The computational grid for the VOF simulation is a cubic domain with $L = 100$ and $100 \times 100 \times 100$ resolution. The time step Δt is 0.2 and the total number of iterations is 4500. When considering the evolution of F_1 and F_2 individually, one should expect the respective surfaces to evolve symmetrically to a flat horizontal plane at $L/2$ since the peaks and troughs are exactly symmetrical in amplitude and wavelength about the center plane. The same argument can be made for the artificial surface since it represents the summation of F_1 and F_2. Since the constructed surface has a component of "randomness" due to the summation of out-of-phase functions, the topological features will be instantaneously different in terms of length scale, however, the final solution will also be a flat plane at $L/2$. This is verified by looking at Figure 24 which shows the iso-surface of C at different time instants until equilibrium.

Since the interfacial velocity is proportional to local curvature, the high-frequency peaks evolve first due to higher curvature as seen in the top center image of Figure 24. This is followed by the evolution of the lower frequency waves shown

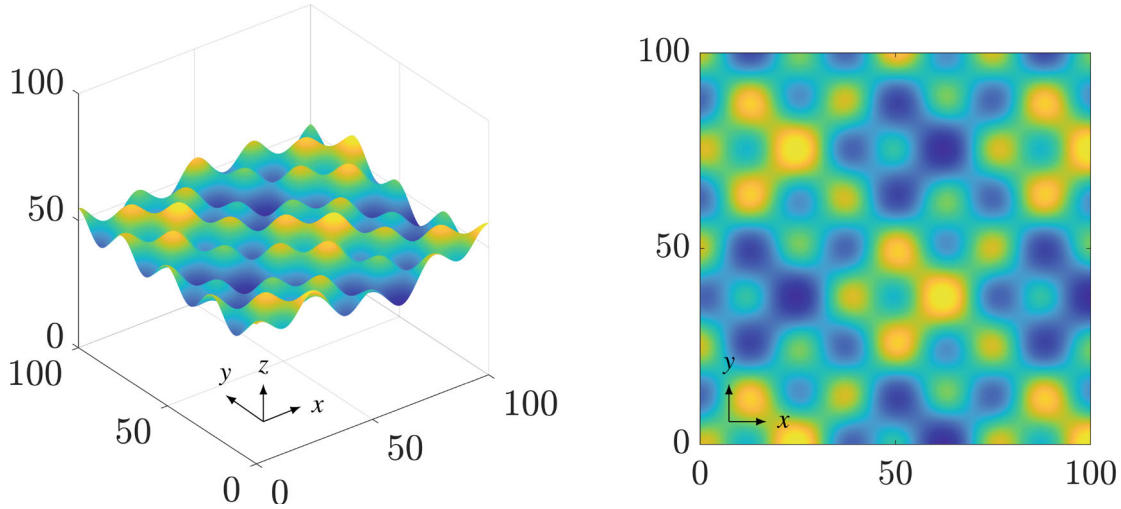

FIGURE 22 3D topology (left) and 2D projected topology (right) of initialized surface. [Colour figure can be viewed at wileyonlinelibrary.com]

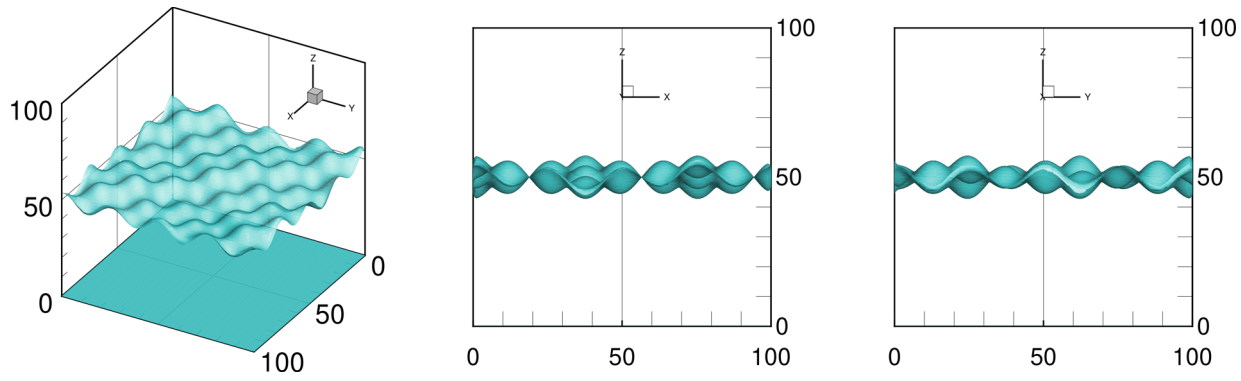

FIGURE 23 Iso-surface of the color function viewed in 3D (left), from the ZX-plane (center), and from the ZY-plane (right). [Colour figure can be viewed at wileyonlinelibrary.com]

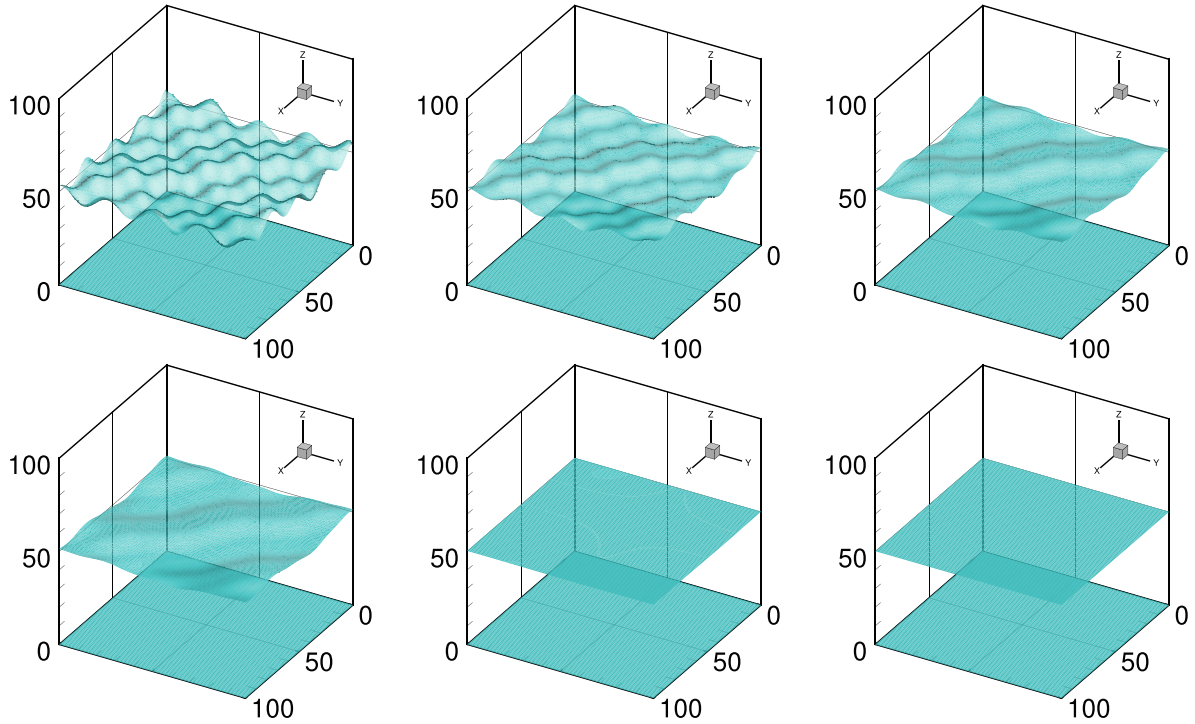

FIGURE 24 Evolution of artificial surface under curvature flow. The images are snapshots of the interface in time from left to right, top to bottom for $t = 0.0$, 6.0, 12.0, 18.0, 450, and 900. [Colour figure can be viewed at wileyonlinelibrary.com]

in the top right image of Figure 24, and so on until the final equilibrium is reached (bottom right image in Figure 24). The simulation was run 2000 iterations beyond equilibrium to check for the sensitivity of the solution to accumulated round-off error with no observable effect on stability.

4.2 | Constrained curvature-driven motion

In the following set of problems, the curvature-constrained velocity field given by Equation (34) is used (see Section 3). The interface evolution of different geometries under constrained curvature flow is examined whereby volume conservation is enforced. Volume conservation in the context of the proposed VOF approach should not be confused with volume conservation in VOF methods in general. The nature of the imposed conservation is purely topological and not related to any alternation at the level of the flux computation algorithm. In other words, the reported errors in volume conservation quantify deviation from an imposed penalty on the volume of the entire initialized geometry. For instance, in the case of pure advection of C via an Eulerian velocity, the total error in the conservation of C is equal to the error in flux conservation plus the deviation of the volume from its initial value due to the constraint.

By examining Equation (34), two questions become apparent,

1. How should $\overline{\kappa}$ be computed?
2. To what extent does the accuracy in the computation of $\overline{\kappa}$ affect the overall solution?

In addressing the first question, the traditional way of computing interfacial properties is by making use of the Dirac delta $\delta(x)$ that is approximated by $|\nabla C|$ in VOF methods. For example, to compute the length of a material interface, the discrete integral has the following expression

$$L = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z} |\nabla C_{i,j,k}|\ \Delta x_i \Delta y_j \Delta z_k. \tag{56}$$

Since $|\nabla C|$ is strictly positive and asymptotically goes to zero away from the interface, the only control volumes that contribute to the summation are those where $|\nabla C_{i,j,k}| \neq 0$. As the computational grid is refined, the band containing the interface where $|\nabla C|$ is non-zero becomes narrower, and $\delta(C) = |\nabla C|$ more closely approximates the Dirac delta locally. The behavior of $\delta(C)$ can be demonstrated by analytically expressing $\delta(x)$ as a zero-centered normal distribution such that

$$\delta_a(x) = \frac{1}{|a|\sqrt{\pi}} e^{-(x/a)^2} \quad \text{where} \quad \delta(x) = \lim_{a \to 0} \delta_a(x) \tag{57}$$

and a is analogous to the computational grid spacing. Figure 25 shows the behavior of $\delta_a(x)$ for decreasing value of a. As stated above, in the limit as a goes to zero, $\delta_a(x)$ tends to $\delta(x)$.

From the numerical point of view, it is unrealistic and rather expensive to increase grid refinement to a point where $|\nabla C|$ overlaps exactly with the interface. Therefore for a non-adaptive grid, one is bound by the resolution imposed by the physical problem under study. To test the accuracy of $|\nabla C|$ in computing $\overline{\kappa}$, we take the simple case of a circle and a sphere whose mean curvatures are given by $1/R$ and $2/R$, respectively. For these simple geometries, the radii of curvature do not change as one traces the curve. Hence, $\overline{\kappa}$ should come out exactly equal to $\kappa(\mathbf{x})$ regardless of the number of differential elements involved in the summation. For a circle and a sphere of radius $r = 25$, the respective mean curvature is 0.04 and 0.08. Table 1 shows $\overline{\kappa}$ obtained from numerical integration using $|\nabla C|$ for the case of a circle and a sphere.

The finest grid in the case of a circle resulted in a 20% error relative to the expected value of 0.04 and an 11.71 error for the sphere relative to the expected value of 0.08. In the context of interface equilibrium, the errors in the computation

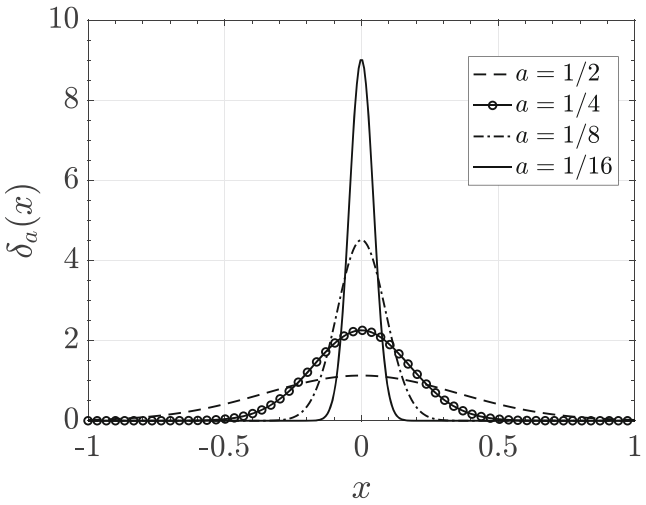

FIGURE 25 $\delta_a(x)$ for different values of a.

TABLE 1 Average curvature estimation of a Circle (Sphere) using $\delta_1(C)$.

Grid size	$\overline{\kappa}$ Circle	$\overline{\kappa}$ Sphere
32×32 (×32)	0.03421	0.06980
64×64 (×64)	0.03363	0.07040
128×128 (×128)	0.03294	0.07049
$192 \times 192 \times 192$	–	0.07049
256×256 (×256)	0.03283	0.07063
512×512	0.03270	–
1024×1024	0.03269	–
1536×1536	0.03268	–

of $\overline{\kappa}$ using $\delta(C) = |\nabla C|$ are unsatisfactory. This motivates the search for a more accurate definition of $\delta(C)$ that results in a better estimation of $\overline{\kappa}$.

The alternative candidate that is considered is $\delta(C) = 4C(1 - C)$. This definition of the Dirac delta is inherently bounded since C is bounded between 0 and 1 by construction. Furthermore, it admits a maximum value of 1 locally at the interface ($C = 0.5$) regardless of the grid size, unlike $|\nabla C|$ where the dependence on grid spacing is explicit via the gradient operator. In what follows, $|\nabla C|$ will be referred to as $\delta_1(C)$ and $4C(1 - C)$ will be referred to as $\delta_2(C)$. In comparing both definitions of $\delta(C)$, the question that remains is: Which definition spans fewer control volumes around the interface resulting in a more accurate $\overline{\kappa}$?

A circle of $r = 25$ was initialized and both definitions of $\delta(C)$ were calculated for various grid sizes. Figure 26 shows the variation of the grid-based width of $\delta_{1,2}(C)$ for increasing resolution. The ratio of W_{δ_1} to W_{δ_2} was found to be approximately equal to 2 for all grids that is, $\delta_1(C)$ is two times wider than $\delta_2(C)$ for any resolution. The same calculation was done in 3D and the ratio $r = W_{\delta_1}/W_{\delta_2}$ was found to be equal to 2 as well. Figure 27 shows a visual comparison of the grid width of $\delta_1(C)$ normalized by its maximum (dashed dot) and $\delta_2(C)$ (solid line) for a coarse grid and a fine one. Note that the reason for normalizing $\delta_1(C)$ in Figure 27 is to enable direct comparison since $max(\delta_1(C)) \neq 1$ locally.

Following the width comparison of $\delta_{1,2}(C)$, $\overline{\kappa}$ was computed again, this time using $\delta_2(C)$ to verify its validity as a superior alternative to $\delta_1(C)$. The results are shown below in Table 2. An immediate improvement in the estimation of

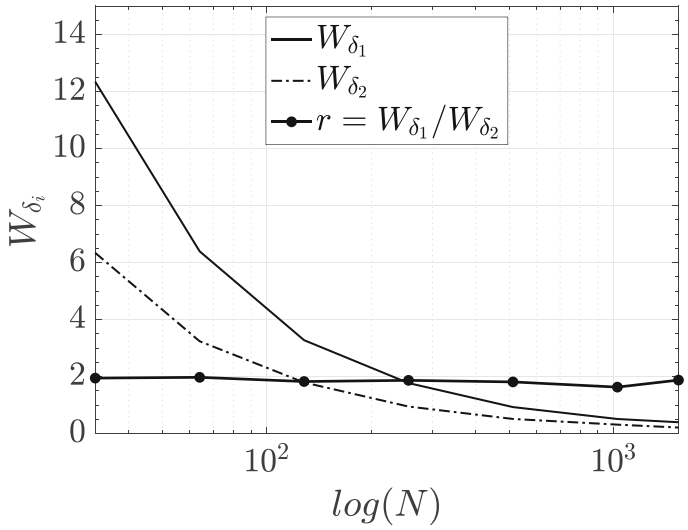

FIGURE 26 Grid based width of $\delta_1(C)$ and $\delta_2(C)$ for different grid resolutions.

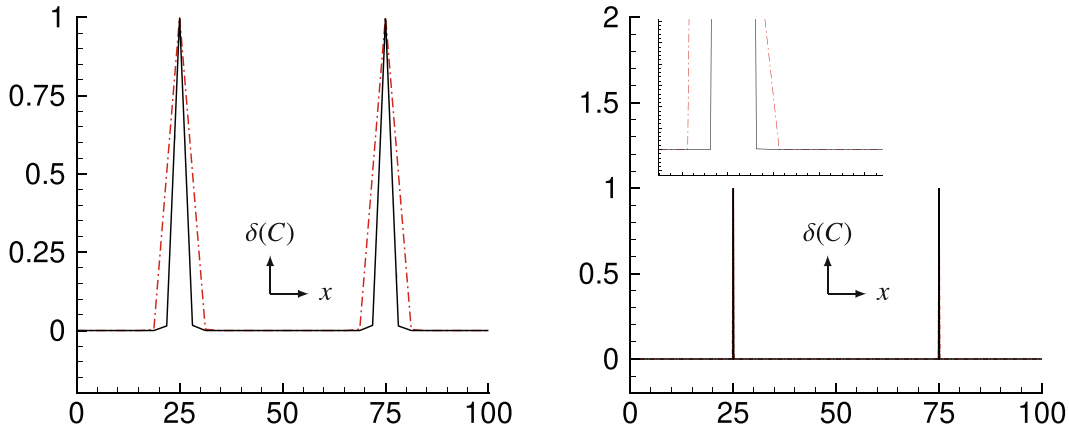

FIGURE 27 Horizontal line extracts of $\delta_1(C)$ (dashed dot) and $\delta_2(C)$ (solid line) calculated across a circular interface for 32×32 (left) and 1536×1536 (right). [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Average curvature estimation of a Circle (Sphere) using $\delta_2(C)$.

Grid size	$\overline{\kappa}$ Circle	$\overline{\kappa}$ Sphere
$32 \times 32(\times 32)$	0.03948	0.07956
$64 \times 64(\times 64)$	0.03932	0.07941
$128 \times 128(\times 128)$	0.03941	0.07943
$192 \times 192 \times 192$	–	0.07942
$256 \times 256(\times 256)$	0.03955	0.07945
512×512	0.03950	–
1024×1024	0.03945	–
1536×1536	0.03945	–

$\overline{\kappa}$ is observed where the error for the finest grid reduces from 20% using $\delta_1(C)$ to 1.375% using $\delta_2(C)$ for the case of the circle, and from 11.71% to 0.6875% for the case of a sphere.

Therefore, $\overline{\kappa}$ in Equation (34) is computed as

$$\overline{\kappa} = \sum_{i=1}^{N_x}\sum_{i=1}^{N_y}\sum_{k=1}^{N_z} 4C(1-C)\ \kappa_{i,j,k}\ \Delta x_i \Delta y_j \Delta z_k \tag{58}$$

in the subsequent problems involving constrained curvature flow, and the second question posed at the being of Section 4.2 will be addressed.

4.2.1 | Ellipsoid

Consider the constrained curvature-driven motion of an ellipsoid whose zeroth level curve is defined as

$$\phi(\mathbf{x}, 0) = \frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} + \frac{(z-z_c)^2}{c^2} - 1, \tag{59}$$

where a, b, and c are the lengths of the semi-axes whose values are taken to be 35.0, 15.625, and 15.625, respectively. This geometry represents a canonical test of the methodology due to the smoothness of the interface and the absence of junctions. The computational domain is a unit cube and three resolutions were chosen for comparison, $50 \times 50 \times 50$, $75 \times 75 \times 75$, and $100 \times 100 \times 100$. The time-step is $\Delta t = 10^{-5}$ and the total number of iterations for both grids is 10000 ($t_{final} = 0.1$). Figure 28 shows snapshots of interface evolution for $t = 0.0, 0.01, 0.02, 0.03, 0.04$, and 0.05.

As expected, the ellipsoid evolves into a sphere with time. When a spherical shape is reached, the volume no longer varies due to the balance between local curvature $\kappa(\mathbf{x})$ and the arithmetic mean curvature $\overline{\kappa}$ of the surface. The simulations were run 5000 iterations beyond the equilibrium point to ensure that built-up round-off errors would not cause the solution to drastically deviate from equilibrium or cause the rupture of the interface at any point in time. Figure 29 shows a contour plot comparison of the two definitions of the Dirac delta for the case of a spheroid (Figure 29A shows $\delta_1(C)$ and Figure 29B shows $\delta_2(C)$).

Figure 30 shows the variation of non-dimensionalized volume with respect to non-dimensional time for the three different grids. Error in volume conservation is approximately 0.7% for $50 \times 50 \times 50$ (Figure 30A), 0.16% for $75 \times 75 \times 75$ (Figure 30B), and 0.138% for $100 \times 100 \times 100$ (Figure 30C) indicating that higher grid resolution reduces errors in constraint violation. Before fixing Δt to 10^{-5} for the three reported resolutions, each grid was fixed while reducing the time step over multiple runs to examine the effect of Δt on the reported errors. No reduction in volume error was observed for any of the grids. Figure 30D shows a scaled comparison between all three grid sizes.

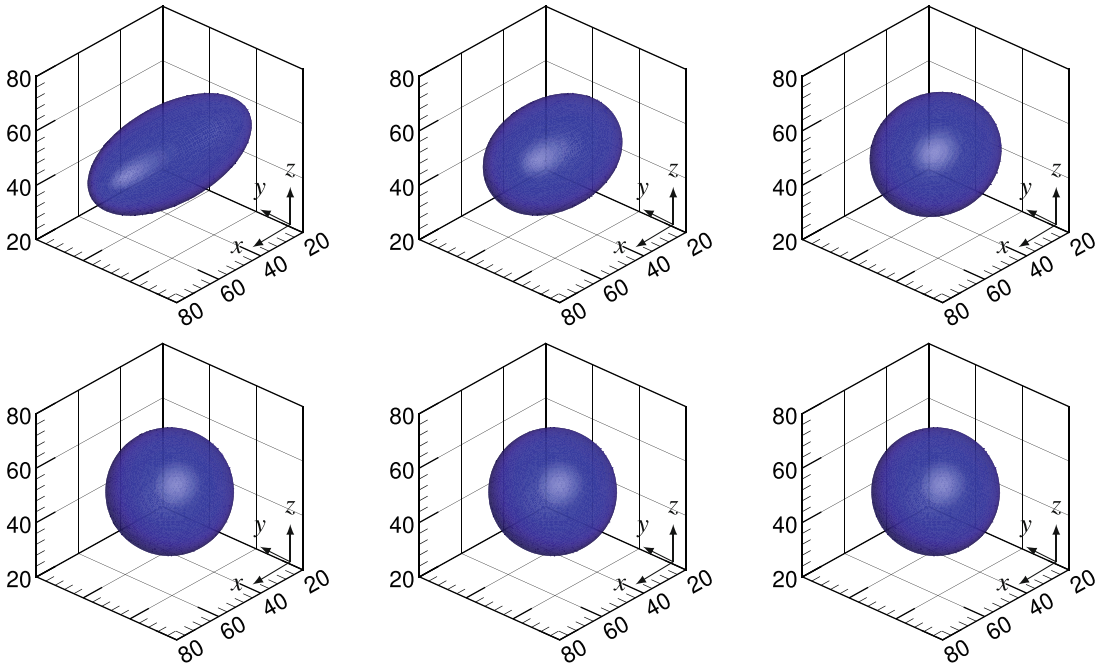

FIGURE 28 Evolution of an ellipsoid under constrained curvature-driven flow. The images are snapshots of the interface in time from left to right, top to bottom $t = 0.0$, 0.01, 0.02, 0.03, 0.04, and 0.05. [Colour figure can be viewed at wileyonlinelibrary.com]

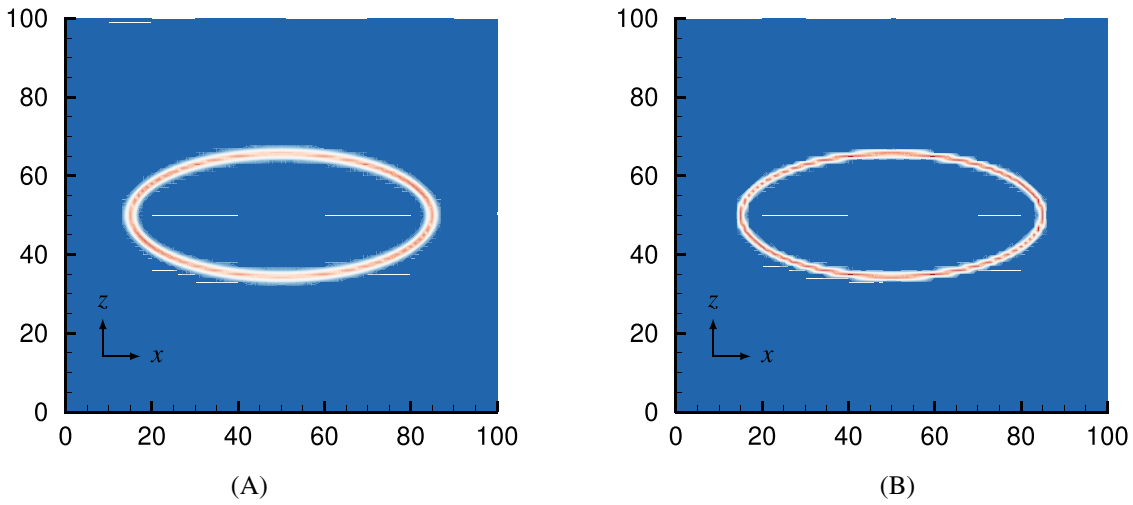

FIGURE 29 Contour plot of (a) $\delta_1(C)$ and (b) $\delta_2(C)$ for the case of an ellipsoid resolved on a $100 \times 100 \times 100$ grid. $\delta_2(C)$ defines the interface more sharply as for the case of a sphere. [Colour figure can be viewed at wileyonlinelibrary.com]

4.2.2 | Superellipsoid

In a Cartesian coordinate system, the superellipsoid is defined by

$$\left|\frac{x-x_c}{a}\right|^n + \left|\frac{y-y_c}{b}\right|^n + \left|\frac{z-z_c}{c}\right|^n = 1, \tag{60}$$

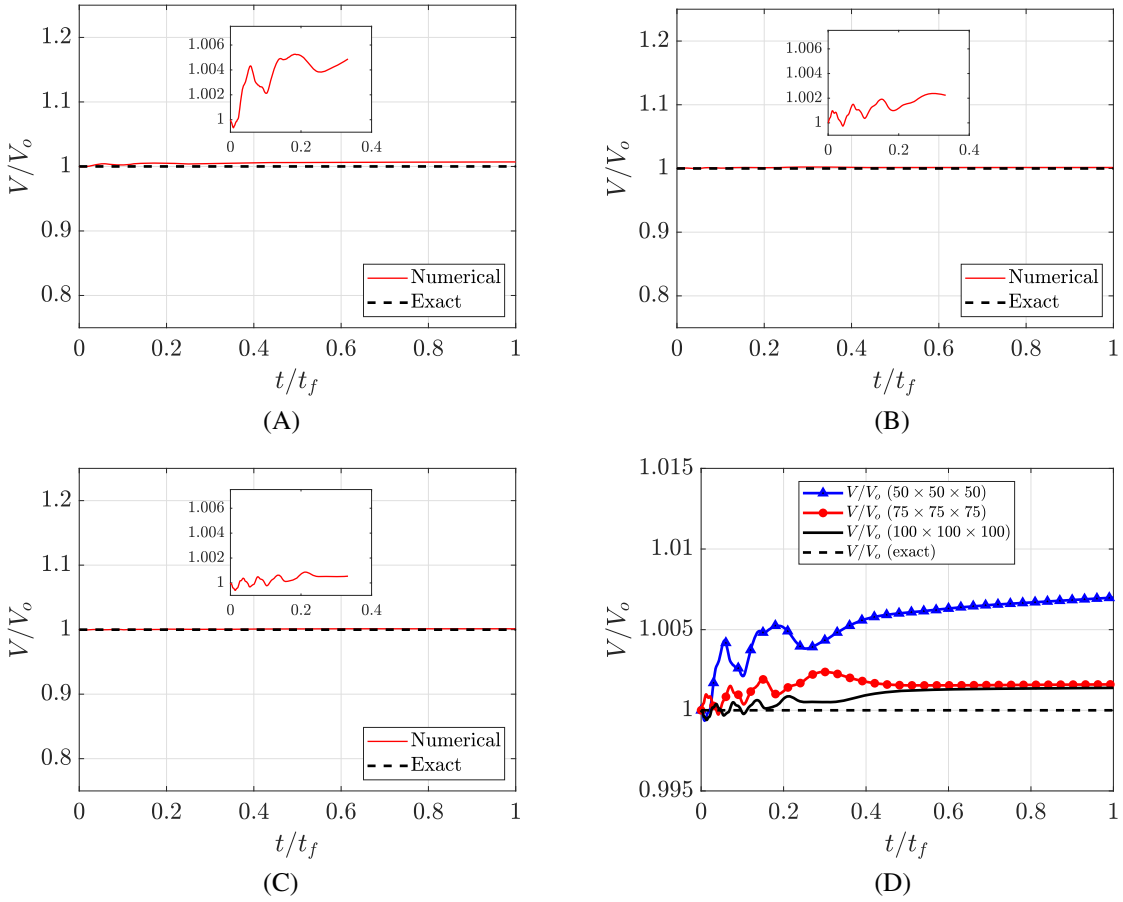

FIGURE 30 Variation of the normalized volume of an ellipsoid with time under constrained curvature-driven motion for (A) $50 \times 50 \times 50$, (B) $75 \times 75 \times 75$, and (C) $100 \times 100 \times 100$. (D) Shows a scaled comparison between all three resolutions. [Colour figure can be viewed at wileyonlinelibrary.com]

where a, b, and c are the lengths of the semi-axes, x_c, y_c and z_c are the coordinates of the center of the ellipsoid, and n is a positive integer. The choice of $n = 4$ and equal semi-axes leads to a Squircle—a circle with rounded edges. As n increases, the resulting geometry attains sharper edges, this in turn helps in approximating a cubic shape whilst controlling the roundedness of the edges and corners. The purpose behind simulating this geometry is to investigate the limitations of curvature flow with VOF where $\kappa(\mathbf{x})$ is not properly defined (curvature is infinite at a point).

Consider the constrained curvature-driven motion of a smooth-edge cube whose zeroth level curve is represented by

$$\phi(\mathbf{x}, 0) = \left(\frac{x - x_c}{r}\right)^n + \left(\frac{y - y_c}{r}\right)^n + \left(\frac{z - z_c}{r}\right)^n - 1, \tag{61}$$

where $n = 12$, $r = 25$, and $\mathbf{x_c}$ is $(50, 50, 50)$. The computational domain is a cube of $L = 100$, and the grid resolutions examined are $75 \times 75 \times 75$, $100 \times 100 \times 100$, and $125 \times 125 \times 125$, respectively. The time-step chosen for the first two grids is $\Delta t = 2 \times 10^{-2}$, $\Delta t = 1 \times 10^{-2}$ for the last grid for stability, and $t_{final} = 100$ units. Figure 31 shows snapshots of interface evolution for $t = 0.0, 20.0, 40.0, 60.0, 120.0$ and 140.0.

The surface minimizes to a sphere as shown in Figure 31 and the corners of the cube evolve normally without "tearing" the interface. This was not the case when $\overline{\kappa}$ was estimated using $\delta_1(C)$ where $\overline{\kappa}$ was found to fluctuate between consecutive time steps due to the development of parasitic velocities at the corners eventually leading to the rupture of the surface. The same phenomenon was also observed for the ellipsoid although the surface is smooth and $\overline{\kappa}$ is well-defined everywhere. This answers the second question posed in Section 4.2 about the effect of $\overline{\kappa}$ accuracy on the stability of numerical simulation. Based on what was previously discussed, $\delta_2(C)$ is superior to $\delta_1(C)$ in terms of the stability of the numerical simulation and the accuracy of $\overline{\kappa}$. Figure 32A,B shows contours of $\delta_{1,2}(C)$.

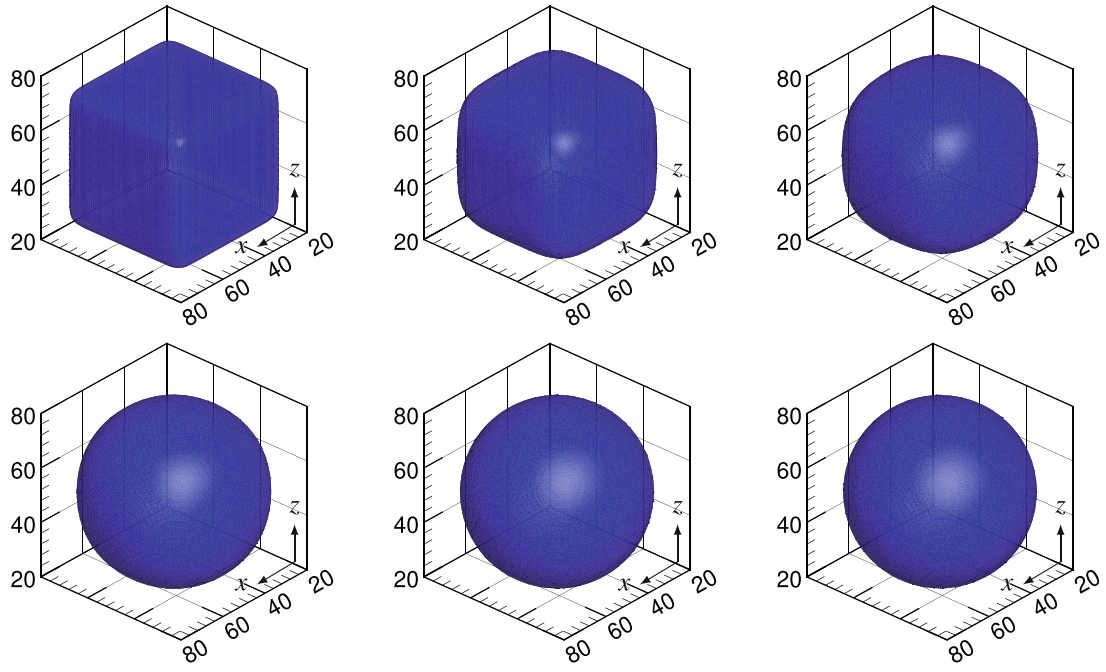

FIGURE 31 Evolution of a round-edge cube under constrained curvature-driven flow. The images are snapshots of the interface in time from left to right, top to bottom where the cubic geometry evolves into a sphere for $t = 0.0$, 20.0, 40.0, 60.0, 120.0, and 140.0. [Colour figure can be viewed at wileyonlinelibrary.com]

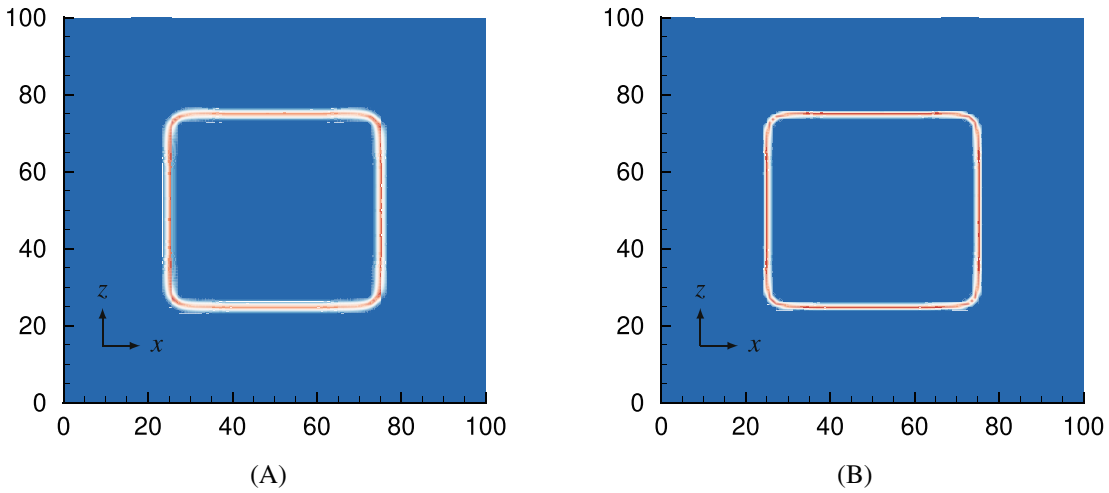

FIGURE 32 Contour plot of (A) $\delta_1(C)$ and (B) $\delta_2(C)$ for the case of a superellipsoid resolved on a $125 \times 125 \times 125$ grid. [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 33 shows the variation of non-dimensionalized volume with respect to non-dimensional time for the three different grids. Error in volume conservation is approximately 1.3% for $75 \times 75 \times 75$ (Figure 33A), 6.34% for $100 \times 100 \times 100$ (Figure 33B), and 0.03% for $125 \times 125 \times 125$ (Figure 33C) indicating that higher grid resolution reduces errors in constraint violation which is consistent with the results in Section 4.2.1. Since the superellipsoid geometry is non-canonical, the grid-based width of $\delta_{1,2}(C)$ was also determined for different grids, and $r = W_{\delta_1}/W_{\delta_2}$ was also found to be equal to 2 as in the case of the sphere (see Section 4.2). Figure 33D shows a scaled comparison between all three grid sizes.

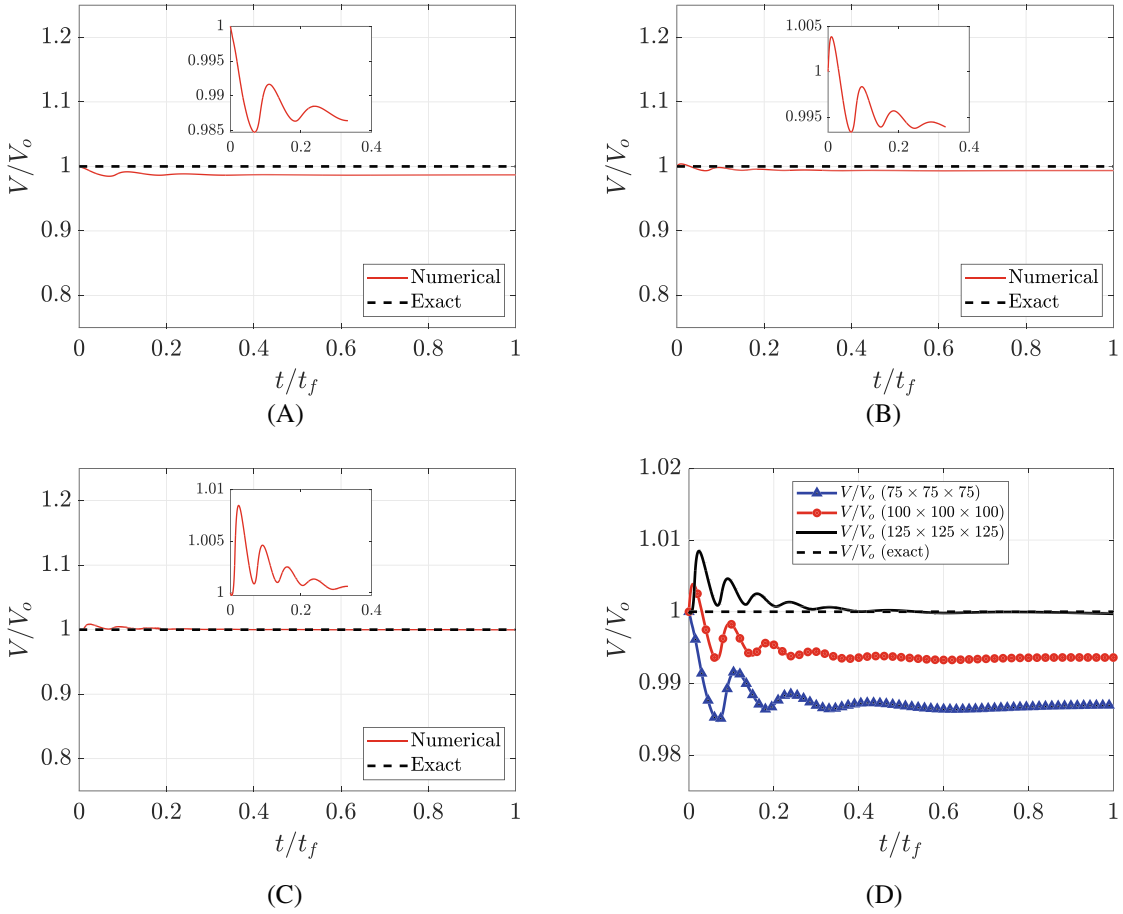

FIGURE 33 Variation of the normalized volume of a superellipsoid with time under constrained curvature-driven motion for (A) $75 \times 75 \times 75$, (B) $100 \times 100 \times 100$, and (C) $125 \times 125 \times 100$. (D) Shows a scaled comparison between all three resolutions. [Colour figure can be viewed at wileyonlinelibrary.com]

4.3 | General velocity field

All the problems simulated in previous sections investigated pure interfacial motion in the normal direction due to free and constrained curvature flow. However, the velocity field in a general fluid flow is rarely exclusively interfacial and is often accompanied by additional contributions. Some of these contributions include the momentum of the background fluid/phase which is Eulerian in nature, gravitational effects or buoyancy in the presence of a density difference between the carrier fluid and the dispersed fluid and so forth. This section looks at two curvature flow problems, the first without the volume constraint and the second with it. An Eulerian velocity component is introduced to the advection in each of the problems and the performance of the proposed approach is assessed.

In the presence of additional velocity contributions, the advection equation can be written as

$$\frac{\partial C}{\partial t} + \mathbf{U} \cdot \nabla C = 0 \quad \text{such that} \quad \mathbf{U} = \mathbf{u} + \mathbf{u}_\Gamma + \ \dots\ , \tag{62}$$

where $\mathbf{U}$ is the total advection velocity, $\mathbf{u}$ is the bulk velocity, and $\mathbf{u}_\Gamma$ is the interfacial velocity that is proportional to curvature. As mentioned earlier, $\mathbf{U}$ can include more terms, however, the following is limited to the first two terms in the expression of $\mathbf{U}$. For the free curvature flow case, consider a one-dimensional pulse function given by

$$P(x) = A_p \ exp\Big(-\big[(x - x_c)/W_p\big]^2\Big), \tag{63}$$

where A_p is the amplitude of the pulse, x_c is location where the pulse is centered, and W_p is the width of the pulse. The zeroth level set defining the 3D surface is therefore

$$\phi(\mathbf{x}, 0) = P(x) - z + \frac{L_z}{2}, \tag{64}$$

where the right-most term of Equation (64) centers ϕ about the z-axis. The computational grid is a cube with $L = 100$ and a resolution $100 \times 100 \times 100$. The time step Δt is 0.1, t_{final} is 100 units, $A_p = 25$, $W_p = 10$, and $\mathbf{u}$ was set to (1,0,0).

The boundary condition is periodic in the x-direction to allow the pulse to return to its initial position after one period. As shown in Figure 34, the pulse flattens due to its curvature and also propagates parallel to itself due to the background velocity. This re-enforces the finding of Reference 6 about the role of curvature being analogous to viscosity in the corresponding hyperbolic conservation law. The pulse propagation problem is a standard test for flux conservation where the observed loss in the conserved quantity is a result of numerical diffusion embedded in the constructed flux computation scheme. The difference in what is presented here is that the spreading of the pulse is due to a self-generated velocity field whose strength can be explicitly controlled by scaling the velocity components in Equation (34) whilst maintaining interface sharpness, that being an implicit feature of geometric VOF. The direct effect of scaling $\mathbf{u}_\Gamma$ in this case is speeding up or slowing down the rate at which the interface goes to equilibrium that is a flat surface.

A two-dimensional view of pulse evolution can be seen in Figure 35A, and a comparison between the initial interface shape and the final interface shape can be seen in Figure 35B.

For the last problem, a sphere of $r = 0.1$ was placed in a helical velocity field and allowed to evolve under constrained curvature flow. The helical velocity field $\mathbf{u}$ is defined as

$$u = 2\pi\ U_{\max}\ (y - y_c), \tag{65}$$

$$v = 2\pi\ V_{\max}\ (x_c - x), \tag{66}$$

$$w = W_{\max}\ \cos^{-1}\zeta \tag{67}$$

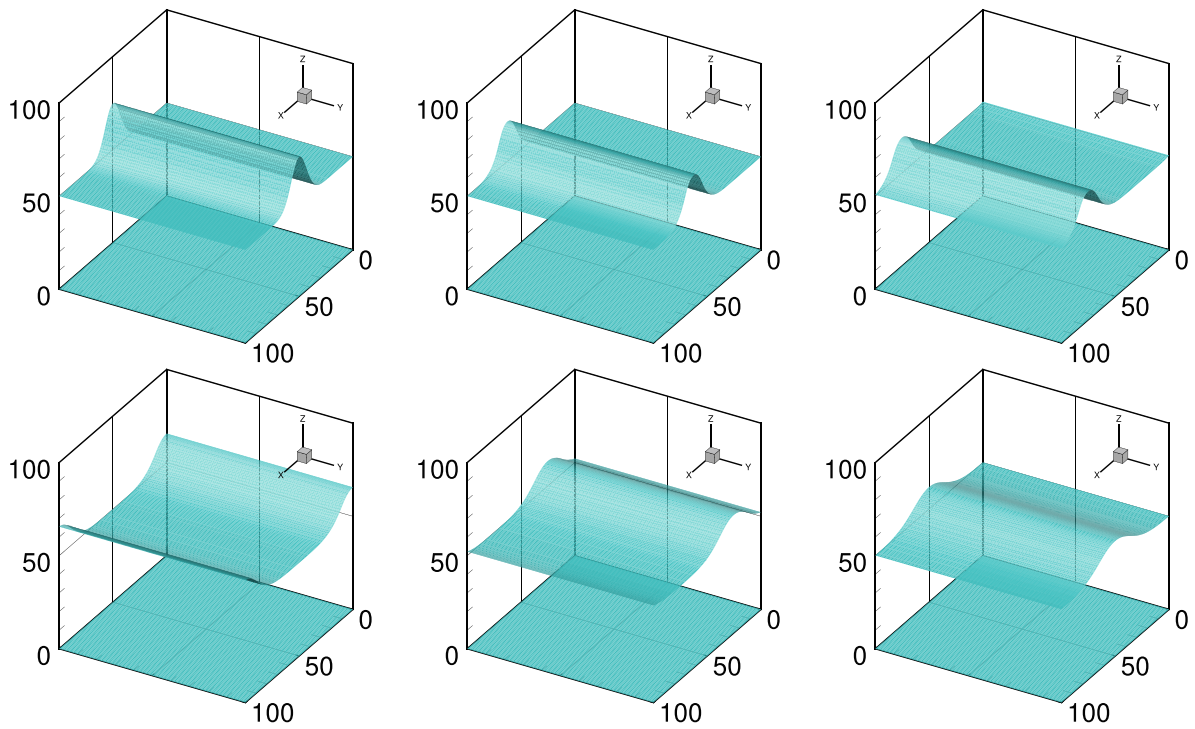

FIGURE 34 Evolution of a 3D pulse under curvature flow and an Eulerian background velocity. The images are snapshots of the interface in time from left to right, top to bottom for $t = 0.0$, 12.5, 25.0, 50.0, 75.0, and 100.0. [Colour figure can be viewed at wileyonlinelibrary.com]

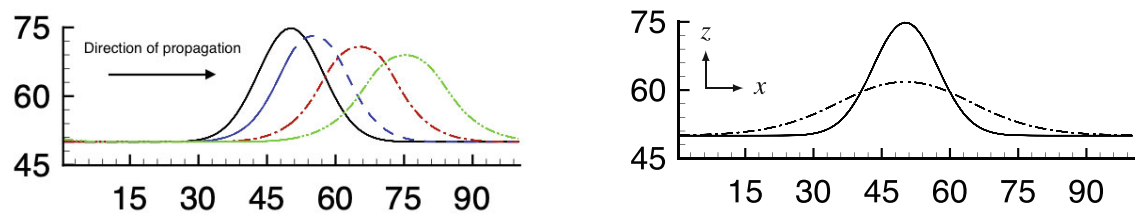

FIGURE 35 (A) Pulse interface ($C \approx 0.5$) in the first half period of propagation. (B) Initial interface contour (solid) and final interface contour (dash dot) after a full period of propagation. [Colour figure can be viewed at wileyonlinelibrary.com]

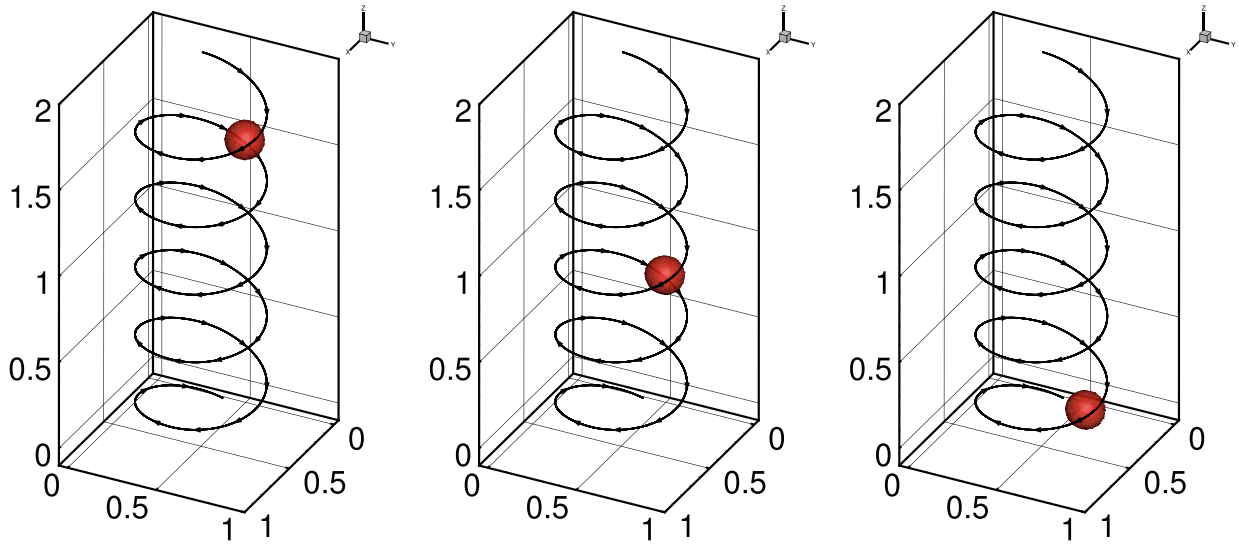

FIGURE 36 Evolution of a sphere under constrained curvature-driven motion and Eulerian advection. The images are snapshots of the interface in time from left to right, top to bottom for $t = 0.0$, 0.0125, and 0.025. [Colour figure can be viewed at wileyonlinelibrary.com]

with $\zeta = (x - x_c)/\sqrt{(x - x_c)^2 + (y - y_c)^2}$. The computational domain is 2 units of length in the z-direction and 1 unit in the other two directions. The time-step is $\Delta t = 0.000025$ for the $75 \times 75 \times 150$ grid and 0.0000125 for $150 \times 150 \times 300$, $U_{\max} = V_{\max} = 160$, $W_{\max} = -40$, $\mathbf{x_c} = (0.5, 0.75, 0.25)$ and the total number of iterations is 1000 for the coarse grid and 2000 for the finer one. The boundary conditions are periodic in all three directions and the sphere is transported by the helical velocity field from its initial position to the bottom of the computational domain. The images in Figure 36 are snapshots of the sphere as it advects to the bottom of the domain for $t = 0.0, 0.0125$, and 0.025 where the surface maintains its sphericity throughout the advection.

The time scale of local interface equilibrium is relatively small compared to the time scale of the Eulerian advection, this is particularly advantageous if the proposed approach is applied to bubble-laden flows where the bubbles are used as tracers without the need for a surface tension model that controls interface deformation. The highest reported mass error is 0.67% for the coarse grid (Figure 37A) and reduces to 0.12% upon doubling the grid size (Figure 37B). Figure 37C shows a scaled comparison of the volume variation with time between the two reported grids.

5 | SUMMARY AND CONCLUDING REMARKS

A directionally-split geometric volume-of-fluid approach for front propagation under curvature flow was presented. The advection velocity for constrained/unconstrained curvature flow was derived using variational analysis applied to the level set function and the results are extended to VOF variables without the loss of generality. The method can reproduce

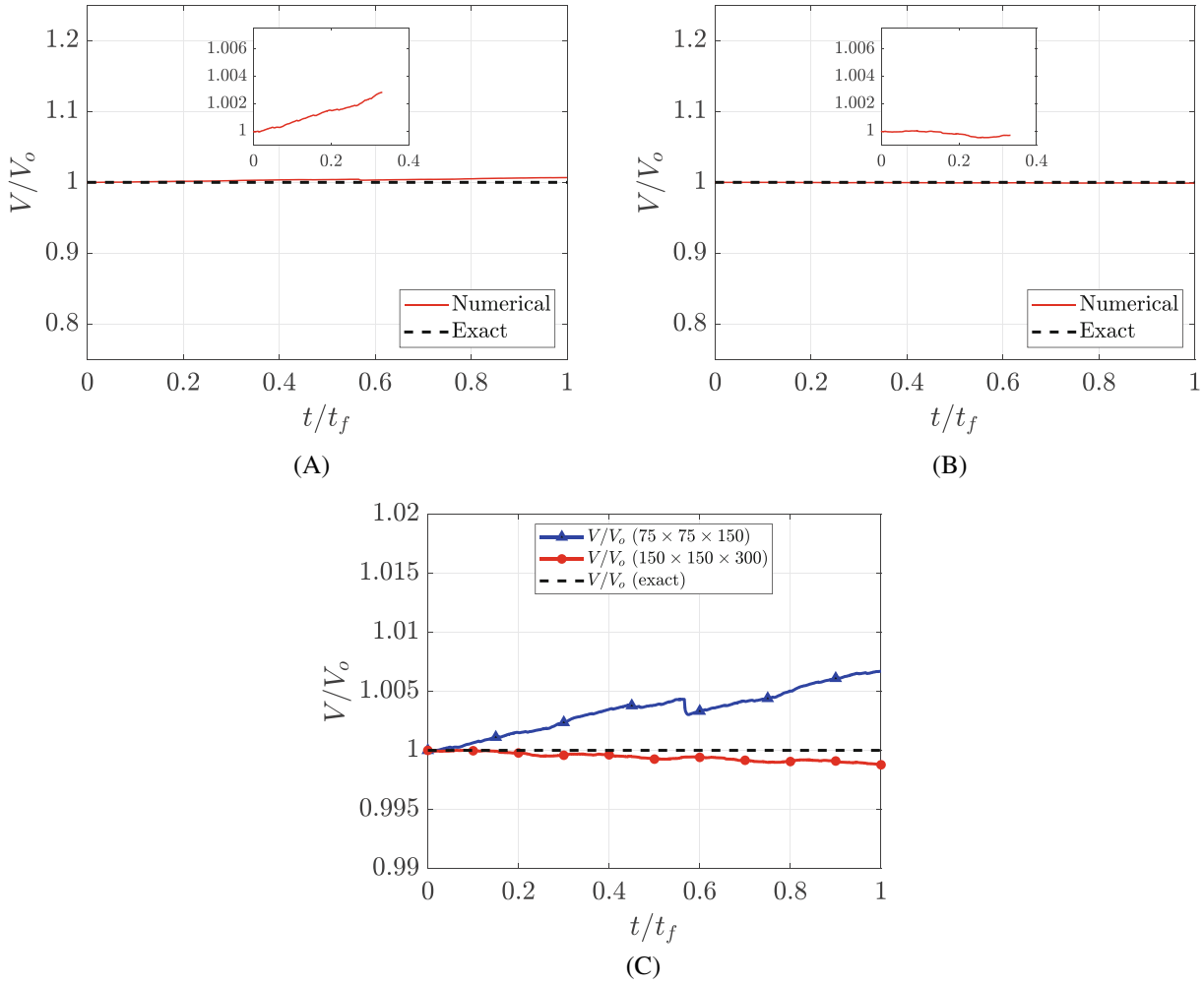

FIGURE 37 Variation of the normalized volume of a sphere due to constrained curvature flow in a general velocity field for (A) $75 \times 75 \times 150$ and (B) $150 \times 150 \times 300$. (C) Shows a scaled comparison between the two resolutions. [Colour figure can be viewed at wileyonlinelibrary.com]

benchmark cases like the RP bubble collapse which is an instance of interface motion in the normal direction. The VOF approach was also able to reproduce canonical level set evolution problems such as the pointed star, the wound spiral, and the dumbbell. For constrained curvature flow, the test cases chosen were designed to investigate the limitations of the method, especially in geometries where point and line singularities may develop. Key findings in this work can be summarized as follows:

- VOF produces consistent results across different grid resolutions for the free curvature flow problems compared to LS with reinitialization. This comes as a direct consequence of the absence of a clear criterion that sets the number of reinitialization steps needed for the LS solution to converge.
- Although algebraic advection algorithms promote better flux conservation, they are not always favorable for problems involving excessive stretching (under curvature flow or otherwise) due to the possible appearance of parasitic "wisps". This was confirmed through the inspection of pinch-off regions in the case of the breaking wave in Reference 4 as well as comparing dumbbell evolution under curvature flow using CIAM and the WY algorithm in Section 4.1.3
- Appropriate approximation of the Dirac delta is paramount to the calculation of $\overline{\kappa}$. This was found to have a direct effect on the stability of the constrained curvature flow problems as well as the accuracy of the value of $\overline{\kappa}$ which dictates the final equilibrium. Using $\delta(C) = |\nabla C|$ promotes relatively large errors in the computation of $\overline{\kappa}$ as compared to using $\delta(C) = 4C(1 - C)$.

- The degree to which a constraint is accurately enforced varies from one geometry to the other. This is primarily dependent on the complexity of interface topology and how well the computational grid resolves the interface in regions where $\kappa(\mathbf{x})$ is ill-defined.

An intuitive extension of the current work comes in the context of the Gibbs free energy framework whereby Gibbs free energy of a system is minimized. This moves the method from a purely numerical framework to a physical model that can be used to solve real equilibrium problems. In that case, the total energy functional is more comprehensive. It can include additional terms such as the bulk energy due to the pressure difference at a phase boundary, a no-penetration constraint between phase boundaries, etc. Details of the extension of the current approach are left for future investigation.

ACKNOWLEDGMENTS

This work is supported by the United States Office of Naval Research (ONR) under ONR Grant N00014-17-1-2676 with Dr. Ki-Han Kim and Dr. Julie Young as the technical monitors. The computations were made possible through the Minnesota Supercomputing Institute (MSI) at the University of Minnesota.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Ali Fakhreddine https://orcid.org/0000-0003-2068-0181
Karim Alamé https://orcid.org/0000-0002-9618-9033
Krishnan Mahesh https://orcid.org/0000-0003-0927-5302

REFERENCES

1. Hirt C, Nichols B. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys*. 1981;39:201-225.
2. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys*. 1988;79:12-49.
3. Aulisa E, Manservisi S, Scardovelli R, Zaleski S. Interface reconstruction with least-squares fit and split advection in three dimensional Cartesian geometry. *J Comput Phys*. 2007;225:2301-2319.
4. Weymouth G, Yue DK-P. Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids. *J Comput Phys*. 2010;229:2853-2865.
5. Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J Comput Phys*. 2000;162:301-337.
6. Chopp DL, Sethian JA. Flow under curvature: singularity formation, minimal surfaces, and geodesics. *Exp Math*. 1993;2(4):235-255.
7. Chopp DL, Sethian JA. Motion by intrinsic Laplacian of curvature. *Interfaces Free Bound*. 1999;1:107-123.
8. Chopp DL. Computing minimal surfaces via level set curvature flow. *J Comput Phys*. 1993;106:77-91.
9. Sussman M, Smereka P, Stanley O. A level set approach for computing solutions to incompressible two-phase flow. *J Comput Phys*. 1994;114:146-159.
10. Peng D, Merriman B, Osher S, Zhao H, Kang M. A PDE-based fast local level set method. *J Comput Phys*. 1999;155:410-438.
11. Li C, Xu C, Gui C, Fox MD. Distance regularized level set evolution and its application to image segmentation. *IEEE Trans Image Process*. 2010;19:3243-3254.
12. Alamé K, Anantharamu S, Mahesh K. A variational level set methodology without reinitialization for the prediction of equilibrium interfaces over arbitrary solid surfaces. *J Comput Phys*. 2020;406:109184.
13. Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J Comput Phys*. 2000;164:228-237.
14. Muzaferija S, Peric M, Sames P, Schelin T. A two-fluid Navier-Stokes solver to simulate water entry. *Proceeding of the 22nd Symposium of Naval Hydrodynamics*. National Academy Press; 1998.
15. Ubbink O, Issa R. A method for capturing sharp fluid interfaces on arbitrary meshes. *J Comput Phys*. 1999;153:26-50.
16. Darwish M, Moukalled F. Convective schemes for capturing interfaces of free-surface flows on unstructured grids. *Numer Heat Tr B-Fund*. 2006;49:19-42.
17. Heyns J, Malan A, Harms T, Oxtoby O. Development of a compressive surface capturing formulation for modelling free-surface flow by using the volume-of-fluid approach. *Int J Numer Methods Fluids*. 2013;71:788-804.

18. Xiao F, Honma Y, Kono T. A simple algebraic interface capturing scheme using hyperbolic tangent function. *Int J Numer Methods Fluids*. 2005;48:1023-1040.
19. Debar R. Fundamentals of the KRAKEN code. LLNL Tech. Rep. UCIR-760 1974.
20. Youngs DL. An interface tracking method for a 3D Eulerian hydrodynamics code. *Atom Weapons Res Establish Tech Rep*. 1984;44(92):35.
21. Li J. Calcul d'interface affine par morceaux. *CR Acad Sci II B*. 1995;320(8):391-396.
22. Tryggvason G, Scardovelli R, Zaleski S. *Direct Numerical Simulations of Gas–Liquid Multiphase Flows*. Cambridge University Press; 2011.
23. Alamé K. *Numerical Prediction of Drag Reduction and Interfacial Shapes Over Superhydrophobic Surfaces*. Ph.D. thesis. University of Minnesota; 2020.
24. Mirjalili S, Jain SS, Dodd MS. Interface-capturing methods for two-phase flows: An overview and recent developments. Technical Report 2017.
25. Grayson MA. The heat equation shrinks embedded plane curves to round points. *J Differ Geom*. 1987;26:285-314.
26. Yuan W, Sauer J, Schnerr GH. Modelling and computation of unsteady cavitation flows in injection nozzles. *J Mech Ind*. 2001;2:383-394.
27. Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. *Comput Struct*. 2005;83:424-425.
28. Smereka P. Semi-implicit level set methods for curvature and surface diffusion motion. *J Sci Comput*. 2003;19:439-456.
29. Arrufat T, Crialesi-Esposito M, Fuster D, et al. A mass-momentum consistent, Volume-of-Fluid method FOT incompressible flow on staggered grids. *J Comput Phys*. 2021;215:104785.

How to cite this article: Fakhreddine A, Alamé K, Mahesh K. Directionally-split volume-of-fluid technique for front propagation under curvature flow. *Int J Numer Meth Fluids*. 2024;1-38. doi: 10.1002/fld.5312

APPENDIX. DERIVATIONS OF THE ENERGY FUNCTIONALS

A.1 General interfacial surface energy functional

$$\mathcal{E}_I(\phi(\mathbf{x},t)) = \int_\Omega \tau(\mathbf{x})\delta(\phi(\mathbf{x},t))|\nabla\phi(\mathbf{x},t)|d\mathbf{x}.$$

Assume that $\tau(\mathbf{x}) = \text{constant} = 1$, therefore

$$\mathcal{E}_I(\phi(\mathbf{x},t)) = \int_\Omega \delta(\phi(\mathbf{x},t))|\nabla\phi(\mathbf{x},t)|d\mathbf{x}.$$

For a general functional $\mathcal{L}$, the Fréchet derivative is defined as

$$\boxed{\left\langle \frac{\partial \mathcal{L}}{\partial \phi}, \chi \right\rangle = \lim_{\epsilon\to 0} \frac{\mathcal{L}(\phi+\epsilon\chi)-\mathcal{L}(\phi)}{\epsilon}}$$

$$\begin{aligned}\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle &= \lim_{\epsilon\to 0} \frac{\mathcal{E}_I(\phi+\epsilon\chi)-\mathcal{E}_I(\phi)}{\epsilon}\\ &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)|\nabla(\phi+\epsilon\chi)|d\mathbf{x} - \int_\Omega \delta(\phi)|\nabla\phi|d\mathbf{x}\right].\end{aligned}$$

Now add and subtract $\left(\int_\Omega \delta(\phi+\epsilon\chi)|\nabla\phi|d\mathbf{x}\right)$ from the left-hand side (LHS)

$$\begin{aligned}\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle &= \lim_{\epsilon\to 0} \frac{\mathcal{E}_I(\phi+\epsilon\chi)-\mathcal{E}_I(\phi)}{\epsilon}\\ &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)|\nabla(\phi+\epsilon\chi)|d\mathbf{x} - \int_\Omega \delta(\phi)|\nabla\phi|d\mathbf{x} + \int_\Omega \delta(\phi+\epsilon\chi)|\nabla\phi|d\mathbf{x} - \int_\Omega \delta(\phi+\epsilon\chi)|\nabla\phi|d\mathbf{x}\right]\\ &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\{|\nabla(\phi+\epsilon\chi)| - |\nabla\phi|\}d\mathbf{x} + \int_\Omega |\nabla\phi|\{\delta(\phi+\epsilon\chi)-\delta(\phi)\}d\mathbf{x}\right].\end{aligned}$$

Continued …

$$\begin{aligned}
\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\{|\nabla(\phi+\epsilon\chi)|-|\nabla\phi|\}d\mathbf{x} + \int_\Omega |\nabla\phi|\{\delta(\phi+\epsilon\chi)-\delta(\phi)\}d\mathbf{x}\right] \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\{|\nabla(\phi+\epsilon\chi)|-|\nabla\phi|\}d\mathbf{x}\right] + \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega |\nabla\phi|\{\delta(\phi+\epsilon\chi)-\delta(\phi)\}d\mathbf{x}\right] \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\{|\nabla(\phi+\epsilon\chi)|-|\nabla\phi|\}d\mathbf{x}\right] + \left[\int_\Omega |\nabla\phi| \underbrace{\lim_{\epsilon\to 0} \frac{1}{\epsilon}\left\{\frac{\delta(\phi+\epsilon\chi)-\delta(\phi)}{\epsilon}\right\}}_{\delta'(\phi)} d\mathbf{x}\right] \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\{|\nabla(\phi+\epsilon\chi)|-|\nabla\phi|\}d\mathbf{x}\right] + \int_\Omega \delta'(\phi)|\nabla\phi|\chi d\mathbf{x} \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \underbrace{\delta(\phi+\epsilon\chi)\left\{[\nabla(\phi+\epsilon\chi)\cdot\nabla(\phi+\epsilon\chi)]^{1/2}-[\nabla\phi\cdot\nabla\phi]^{1/2}\right\}}_{Subterm\ I.} d\mathbf{x}\right] + \underbrace{\int_\Omega \delta'(\phi)|\nabla\phi|\chi d\mathbf{x}}_{Term\ II.}.
\end{aligned}$$

Multiply and divide *Subterm I*. by: $[\nabla(\phi+\epsilon\chi)\cdot\nabla(\phi+\epsilon\chi)]^{1/2}+[\nabla\phi\cdot\nabla\phi]^{1/2}$, therefore

$$\begin{aligned}
\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\left\{\frac{\nabla(\phi+\epsilon\chi)\cdot\nabla(\phi+\epsilon\chi)-\nabla\phi\cdot\nabla\phi}{[\nabla(\phi+\epsilon\chi)\cdot\nabla(\phi+\epsilon\chi)]^{1/2}+[\nabla\phi\cdot\nabla\phi]^{1/2}}\right\}d\mathbf{x}\right] + Term\ II. \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\left\{\frac{\nabla\phi\cdot\nabla\phi+2\epsilon\nabla\phi\cdot\nabla\chi+\epsilon^2\nabla\chi\cdot\nabla\chi-\nabla\phi\cdot\nabla\phi}{[\nabla(\phi+\epsilon\chi)\cdot\nabla(\phi+\epsilon\chi)]^{1/2}+[\nabla\phi\cdot\nabla\phi]^{1/2}}\right\}d\mathbf{x}\right] + Term\ II. \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\left\{\frac{2\epsilon\nabla\phi\cdot\nabla\chi+\epsilon^2\nabla\chi\cdot\nabla\chi}{\sqrt{\nabla(\phi+\epsilon\chi)\cdot\nabla(\phi+\epsilon\chi)}+|\nabla\phi|}\right\}d\mathbf{x}\right] + Term\ II. \\
\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[\int_\Omega \delta(\phi+\epsilon\chi)\left\{\frac{2\epsilon\nabla\phi\cdot\nabla\chi+\epsilon^2\nabla\chi\cdot\nabla\chi}{\sqrt{\nabla\phi\cdot\nabla\phi+2\epsilon\nabla\phi\cdot\nabla\chi+\epsilon^2\nabla\chi\cdot\nabla\chi}+|\nabla\phi|}\right\}d\mathbf{x}\right] + Term\ II.
\end{aligned}$$

Now divide through by ϵ in *Subterm I*. before taking the limit

$$\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle = \lim_{\epsilon\to 0} \left[\int_\Omega \delta(\phi+\epsilon\chi)\left\{\frac{2\nabla\phi\cdot\nabla\chi+\epsilon\nabla\chi\cdot\nabla\chi}{\sqrt{\nabla\phi\cdot\nabla\phi+2\epsilon\nabla\phi\cdot\nabla\chi+\epsilon^2\nabla\chi\cdot\nabla\chi}+|\nabla\phi|}\right\}d\mathbf{x}\right] + Term\ II.$$

Taking the limit as ϵ tends to zero leads to

$$\begin{aligned}
\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle &= \int_\Omega \delta(\phi)\left\{\frac{2\nabla\phi\cdot\nabla\chi}{\sqrt{\nabla\phi\cdot\nabla\phi}+|\nabla\phi|}\right\}d\mathbf{x} + Term\ II. \\
&= \int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} + Term\ II. \\
&= \int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} + \int_\Omega \delta'(\phi)|\nabla\phi|\chi d\mathbf{x} \\
\boxed{\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle} &= \underbrace{\int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x}}_{Term\ I.} + \underbrace{\int_\Omega \delta'(\phi)|\nabla\phi|\chi d\mathbf{x}}_{Term\ II.}
\end{aligned} \tag{A1}$$

Term I. can be expanded such that

$$\int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} = \underbrace{\int_\Omega \nabla\cdot\left(\delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\chi\right)d\mathbf{x}}_{Term\ I.\ (a)} - \underbrace{\int_\Omega \nabla\cdot\left(\delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x}}_{Term\ I.\ (b)}.$$

Apply Gauss's theorem to *Term I*. (a) and expand *Term I*. (b)

$$\int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} = \int_{\partial\Omega}\delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\chi\cdot\mathbf{n}dS - \int_\Omega \underbrace{\nabla\delta(\phi)}_{\delta'(\phi)|\nabla\phi|}\cdot\frac{\nabla\phi}{|\nabla\phi|}\chi d\mathbf{x} - \int_\Omega\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x}$$

$$\int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} = \int_{\partial\Omega}\delta(\phi)\frac{1}{|\nabla\phi|}\frac{\partial\phi}{\partial n}\chi dS - \int_\Omega\delta'(\phi)|\nabla\phi|\chi d\mathbf{x} - \int_\Omega\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x}.$$

Assuming zero Neumann boundary conditions at the boundary of the computational domain

$$\int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} = \underbrace{\cancel{\int_{\partial\Omega}\delta(\phi)\frac{1}{|\nabla\phi|}\frac{\partial\phi}{\partial n}\chi dS}}_{\frac{\partial\phi}{\partial n}=0\ on\ \partial\Omega} - \int_\Omega\delta'(\phi)|\nabla\phi|\chi d\mathbf{x} - \int_\Omega\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x}$$

$$\boxed{\int_\Omega \delta(\phi)\left(\frac{\nabla\phi\cdot\nabla\chi}{|\nabla\phi|}\right)d\mathbf{x} = -\int_\Omega\delta\prime(\phi)|\nabla\phi|\chi d\mathbf{x} - \int_\Omega\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x}.}$$

Finally, Equation (A1) becomes,

$$\left\langle\frac{\partial\mathcal{E}_I}{\partial\phi},\chi\right\rangle = -\int_\Omega\delta'(\phi)|\nabla\phi|\chi d\mathbf{x} - \int_\Omega\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x} + \int_\Omega\delta'(\phi)|\nabla\phi|\chi d\mathbf{x}$$

$$\left\langle\frac{\partial\mathcal{E}_I}{\partial\phi},\chi\right\rangle = -\int_\Omega\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\chi d\mathbf{x}.$$

Since the unit normal at the interface is defined as $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ and mean interface curvature is defined as $\kappa = \nabla\cdot\mathbf{n}$, then

$$\boxed{\left\langle\frac{\partial\mathcal{E}_I}{\partial\phi},\chi\right\rangle = -\int_\Omega\kappa\delta(\phi)\chi d\mathbf{x}.} \tag{A2}$$

A.2 Energy functional associated with volume conservation constraint

The energy functional associated with the volume conservation constraint is given by

$$G(\phi(\mathbf{x},t)) = \int_\Omega H(\phi(\mathbf{x},t))d\mathbf{x} = V_o,$$

where $H(\phi(\mathbf{x},t))$ is the Heaviside function associated with level set $\phi(\mathbf{x},t)$ and V_o is the initial volume of a closed surface. The Fréchet derivative of $G(\phi)$ is equal to

$$\left\langle\frac{\partial G}{\partial\phi},\chi\right\rangle = \lim_{\epsilon\to0}\frac{G(\phi+\epsilon\chi)-G(\phi)}{\epsilon}$$

$$\left\langle\frac{\partial G}{\partial\phi},\chi\right\rangle = \lim_{\epsilon\to0}\frac{1}{\epsilon}\int_\Omega[H(\phi+\epsilon\chi)-H(\phi)]d\mathbf{x} = \int_\Omega\lim_{\epsilon\to0}\frac{H(\phi+\epsilon\chi)-H(\phi)}{\epsilon}d\mathbf{x} = \int_\Omega H'(\phi)\chi d\mathbf{x} = \int_\Omega\delta(\phi)\chi d\mathbf{x}$$

$$\boxed{\left\langle\frac{\partial G}{\partial\phi},\chi\right\rangle = \int_\Omega\delta(\phi)\chi d\mathbf{x}.} \tag{A3}$$

A.3 Total energy functional

The total energy functional has the following form

$$\mathcal{L}(\phi) = \mathcal{E}_I(\phi) + \lambda G(\phi)$$

such that $\frac{\partial \phi}{\partial t} = -\frac{\partial \mathcal{L}}{\partial \phi}$ for gradient flow that minimizes $\mathcal{L}(\phi)$. The Fréchet derivative in the L^2-norm of $\mathcal{L}(\phi)$ is given by

$$\left\langle \frac{\partial \mathcal{L}}{\partial \phi}, \chi \right\rangle = \left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle + \lambda \left\langle \frac{\partial G}{\partial \phi}, \chi \right\rangle.$$

For a gradient flow that minimizes $\mathcal{L}(\phi)$:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \underbrace{\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)}_{\kappa} - \lambda \delta(\phi). \tag{A4}$$

Although Equation (A4) looks similar to a curvature-dependent evolution of $\phi(\mathbf{x}, t)$, the difference is having $\delta(\phi)$ in the equation instead of $|\nabla \phi|$. The action of $\delta(\phi)$ and $|\nabla \phi|$ on ϕ are fundamentally different, since the former only moves the zeroth level set while the latter moves all the level sets by their respective curvatures. Since $\phi(\mathbf{x}, t)$ is a signed distance function, an alternative level set norm can be defined instead of the L^2-norm where

$$\langle f, g \rangle_\phi = \int_\Omega f \frac{g}{|\nabla \phi|} d\mathbf{x}.$$

Hence,

$$\left\langle \frac{\partial \mathcal{E}_I}{\partial \phi}, \chi \right\rangle = \int_\Omega |\nabla \phi| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \frac{\chi}{|\nabla \phi|} d\mathbf{x} = \int_\Omega |\nabla \phi| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \psi d\mathbf{x}$$
$$\left\langle \frac{\partial G}{\partial \phi}, \chi \right\rangle = \int_\Omega |\nabla \phi| \frac{\chi}{|\nabla \phi|} d\mathbf{x} = \int_\Omega |\nabla \phi| \psi d\mathbf{x},$$

where ψ is the modified test function. Therefore,

$$\frac{\partial \phi}{\partial t} - \kappa |\nabla \phi| = -\lambda |\nabla \phi|$$

which retains the traditional evolution equation of $\phi(\mathbf{x}, t)$ under curvature with an applied constraint. To obtain the Lagrange multiplier λ, $\frac{dG(\phi)}{dt} = 0$ which leads to

$$\boxed{\lambda = \frac{\int_\Omega \kappa \delta(\phi(\mathbf{x}, t)) d\mathbf{x}}{\int_\Omega \delta^2(\phi(\mathbf{x}, t)) d\mathbf{x}} = \bar{\kappa}.}$$

Therefore, the constrained time evolution of ϕ is governed by

$$\frac{\partial \phi}{\partial t} - (\kappa - \bar{\kappa}) |\nabla \phi| = 0. \tag{A5}$$