Journal of Computational Physics 380 (2019) 119–142

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A volume-conserving balanced-force level set method on unstructured meshes using a control volume finite element formulation

Stephen Lin [a], Jinhui Yan [b], Dmitriy Kats [a], Gregory J. Wagner [a,*]

[a] Department of Mechanical Engineering , Northwestern University, Evanston, IL, USA
[b] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

ARTICLE INFO

Article history:
Received 22 February 2018
Received in revised form 15 November 2018
Accepted 23 November 2018
Available online 26 November 2018

Keywords:
Level set method
Control volume finite element method
Unstructured grid
Balanced force

ABSTRACT

The level set method for modeling multi-fluid interfaces gives a simple approach for the simulation of multiphase flow problems. However, many of the components of this method, such as the redistancing of the level set function and discretization of the singular surface tension force, become more difficult on unstructured meshes in a control volume framework; furthermore, these operations are shown to lead to errors in conservation of mass of the individual phases, as well as unphysical currents near the interface. This paper presents a novel formulation for the level set method combined with a balanced-force algorithm using a control volume finite element method (CVFEM) for unstructured grids. A diffuse interface description is used, allowing a straightforward inclusion of surface tension forces; we show that by consistently evaluating the pressure gradient and surface tension, we are able to discretely balance these two forces and suppress any parasitic currents given a prescribed curvature. By converting the redistancing equation to a form that is suitable for CVFEM and enforcing a constraint directly linked with the phase fraction in each cell, we are able conserve mass and preserve the shape of the interface to a high degree of accuracy in multiphase flow problems with high density and viscosity ratios. The accuracy of the of our proposed method is assessed by comparing with benchmark results from the literature.

1. Introduction

Multiphase flows are frequently observed in many engineering applications. These may include wave breaking, melting, combustion and liquid jet atomization. Numerical modeling can play a crucial role in understanding the physical behavior of these flows. However, the accurate simulation of a moving interface, along with the complex interface conditions and discontinuous material properties that often characterize such problems, can be very challenging. As a result, a great deal of research effort has been targeted at developing numerical methods for modeling multiphase flows.

Two classes of methods are used for resolving the motion of the interface within numerical modeling: interface tracking or interface capturing methods. Interface tracking methods explicitly track the interface by using a deforming mesh to conform to the interface. Examples include the front tracking method [1,2], arbitrary Lagrangian–Eulerian (ALE) methods [3]

* Corresponding author.
E-mail address: gregory.wagner@northwestern.edu (G.J. Wagner).

https://doi.org/10.1016/j.jcp.2018.11.032
0021-9991/

and boundary integral methods [4]. These types of methods provide high accuracy per degree of freedom at the interface, but may require complex remeshing strategies, especially for problems with large topological changes such as coalescence or breakup.

Conversely, interface capturing methods make use of an additional auxiliary function defined over the entire grid to represent the geometry of the interface. Some examples include the volume of fluid (VOF) [5,6], level set (LS) [7–9] and coupled level set volume of fluid (CLSVOF) methods [10–12]. These types of methods are very robust in their ability to handle complex morphological changes in the interface, and are often easier to implement in practice compared with interface tracking methods, though at the cost of the need for a higher mesh resolution to accurately resolve the interface. As a result, these methods have been applied to a broad range of applications, including include jet atomization [13,14], additive manufacturing [15–18] and wind turbine flow [19,20].

Level set methods are among the most popular interface capturing methods for resolving free surface flows due to the fact that they provide an accurate and straightforward computation of the interface normal and curvature from a smooth function. These quantities are harder to obtain in other interface capturing approaches such as VOF methods, which require complex interface reconstruction techniques to calculate these values. Level set methods utilize a smooth signed distance function to implicitly represent arbitrary interface geometries. The evolution of this function is governed by an advection equation combined with a redistancing algorithm to maintain its signed distance property; this redistancing scheme can either be done with a time-dependent evolution in pseudo time [21] or by solving the Eikonal equation using either a fast marching [22–24] or fast sweeping [25–27] method to restore the signed distance property of the field. The fast marching and fast sweeping methods redistance the level set field by solving $|\nabla\phi| = 1$ over the entire domain. This redistancing operation may lead to a loss of volume conservation, and can be difficult to parallelize efficiently. Parallel implementations of the fast marching [24] and fast sweeping [26,27] methods have been reported; however, challenges remain for unstructured meshes. Alternatively, a pseudo time-dependent evolution equation can be solved that results in a redistanced level set field at steady state [9]. These methods can be parallelized efficiently since they can be treated in like other transport equations; however, these types of methods are still prone to mass conservation error. As a result, much research has been devoted to developing new methods to conserve mass in a level set method framework.

There have been attempts to improve mass conservation during redistancing for the original LS scheme. Sussman et al. [28] proposed a constraint that preserved the volume fraction within each cell during redistancing in pseudo time so that the interface is preserved with a high degree of accuracy. This constraint has been implemented in a higher-order finite difference scheme [28] as well as in a finite element framework [29]. Another method for conserving mass within the original level set framework is the addition of a perturbation to the entire level set field after redistancing. This perturbation can be taken as a global constant [30] or as a spatially varying function [31]. An alternative formulation to the LS method, introduced by Olsson and Kreiss [32] and further developed by Olsson et al. [33], introduces a regularized hyperbolic tangent level set that plays the role of a liquid volume fraction. This conservative level set (CLS) method also includes a conservative resharpening procedure to maintain the correct form of the hyperbolic tangent, which is otherwise distorted by advection. Accurate solution of the conservative forms of advection and resharpening preserves the integral of the CLS field, and thus total volume of the liquid phase.

Incorporation of surface tension forces at the free surface also requires careful consideration. Common discretization methods include the Continuum Surface Force (CSF) approach [34], which smears the surface force over a small number of elements near the interface, or the ghost fluid method [35], which imposes sharper boundary conditions on embedded boundaries. Both of these techniques, however, have the tendency to generate “parasitic currents” near the interface when surface tension forces are dominant. These parasitic currents are best demonstrated for an equilibrium problem of an inviscid static drop without any gravitational forces. The cause for these currents results from a combination of a numerical imbalance between the pressure gradient and the surface tension force and an inaccurate evaluation of the interface curvature. Torres and Brackbill [36] have addressed both of these issues by combining a front-tracking method for accurate computations of the interface curvature with a modified projection scheme to minimize these parasitic currents. Francois et al. [37] have addressed the former by introducing a balanced-force algorithm coupled to a VOF method within a segregated solver; their results show that for a prescribed curvature they are able to eliminate spurious currents up to machine precision zero. This method has been extended to unstructured meshes within a fully-coupled framework [38] using the VOF method. These balanced-force formulations have also been implemented with level set methods [39,40] where the interface is resolved on a Cartesian grid.

It is clear that in the context of a level set framework, the redistancing/resharpening procedure is crucial to providing an accurate depiction of the evolution of the interface; these procedures should both conserve mass and be able to reconstruct geometric quantities with a high degree of accuracy. Moreover, it is desirable to apply these methods on unstructured meshes; this enables meshing of complex domains with a straightforward technique to refine the mesh near the interface. In addition, use of a balanced-force technique improves the accuracy of surface force computations, important in surface tension-dominated flows. Most improvements on mass conservation during redistancing for the original LS scheme have been developed on structured Cartesian grids. Some research efforts have extended these improvements over redistancing to unstructured meshes in the context of a finite element framework [29,31,20], however their implementation in a volume-conservative formulation (such as finite volume methods) is not clear. On the other hand, the CLS method has been implemented on unstructured meshes in a finite-volume formulation by Balcazar et al. [41,42], however the resharpening procedure has its own disadvantages: Desjardins et al. [14] observed oscillatory normal vectors during resharpening,

which ultimately affected the stability of the scheme itself. Owkes and Desjardins [43] addressed this by combining the CLS with a redistancing algorithm that utilized a fast marching method; however, excessive use of this algorithm, especially in areas where the CLS does not require resharpening or redistancing, may still deteriorate the hyperbolic tangent profile [44]. Additionally, a balanced force technique combined with a level set method for purely unstructured meshes is still not available.

There is still a need for a level set formulation for multiphase flows, including level set advection and redistancing, that can be implemented easily on unstructured meshes, conserves mass of each phase, scales efficiently in parallel, and is robust even for problems with large density and viscosity ratios. We aim to address this need in our present work by developing a balanced-force, volume-conserving level set method using a control volume finite element (CVFEM) formulation for unstructured meshes. CVFEM [45,46] combines the flexibility of a finite element interpolation scheme with the robust conservation properties of traditional finite volume techniques. In this way, scalar quantities governed by a volume-conservative form can be accurately transported over unstructured grids while maintaining conservation to a high degree of accuracy. A diffuse interface level set approach is used here, allowing straightforward inclusions of surface tension forces for two-fluid simulations using the continuum surface force model formulated by Brackbill et al. [34]. By converting the pseudo-time dependent redistancing to a form that can be solved using CVFEM, we are able to use higher-order upwinding techniques for unstructured meshes. Additionally, we formulate a new volume-conserving constraint over the redistancing algorithm that is similar in spirit to the formulation developed by Sussman et al. [28] and combine this with the global volume corrector developed by Smolianski [30] to show that we can conserve mass and interface shape to a high degree of accuracy. Lastly, we extend the balanced-force algorithm originally proposed by Francois et al. [37] to purely unstructured meshes in the context of a level set framework; we demonstrate that for the canonical inviscid static droplet problem, we are able to achieve a discrete balance between the pressure gradient and surface tension force under a prescribed curvature.

This paper is organized as follows. The mathematical formulation is described in Section 2. A CVFEM discretization of the method and description of the solution procedure is presented in Section 3. Numerical tests are then performed to assess the robustness and accuracy of our method in Section 4. Conclusions are discussed in Section 5.

2. Mathematical formulation

2.1. The level set field

2.1.1. Level set definition and properties

In the level set method, the phase interface Γ is implicitly represented by the zero level set of smooth function $\phi(\boldsymbol{x}, t)$:

$$\Gamma = \{\boldsymbol{x} \mid \phi(\boldsymbol{x}, t) = 0\} \tag{1}$$

Away from the interface, $\phi(\boldsymbol{x}, t)$ is taken to be a signed distance function to the interface Γ, and thus has the property that $|\nabla\phi| = 1$. In our formulation, we consider the level set function to be positive and negative in the liquid and gas phase, respectively:

$$\phi(\boldsymbol{x}, t) \begin{cases} > 0 & \text{if } \boldsymbol{x} \in \text{liquid phase} \\ = 0 & \text{if } \boldsymbol{x} \in \Gamma \\ < 0 & \text{if } \boldsymbol{x} \in \text{gas phase} \end{cases} \tag{2}$$

The level set $\phi(\boldsymbol{x}, t)$ is advected through the integration of an advection equation:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla\phi = 0. \tag{3}$$

The advective flow field $\boldsymbol{u}$ is obtained from the solution of the Navier–Stokes equations, which will be reviewed in Section 2.2. Geometric parameters related to the interface Γ are also calculated from the level set field ϕ:

$$\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa = -\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \tag{4}$$

where $\boldsymbol{n}$ and κ are respectively the unit normal vector and curvature of the interface Γ.

To evaluate material properties such as density and viscosity, we make use of a diffuse interface formulation; properties associated with each phase (such as density and viscosity) vary smoothly across the interface over a fixed distance. A smoothed Heaviside function H_ε is used to assign physical properties to the liquid and gas phase:

$$\begin{aligned} \rho(\phi) &= \rho_1 H_\varepsilon(\phi) + \rho_2 (1 - H_\varepsilon(\phi)) \\ \mu(\phi) &= \mu_1 H_\varepsilon(\phi) + \mu_2 (1 - H_\varepsilon(\phi)) \end{aligned} \tag{5}$$

where ρ_1 and ρ_2 denote the density of the liquid and gas phases, and μ_1 and μ_2 denote the viscosity of the liquid and gas phase, respectively. The function $H_\varepsilon(\phi)$ is calculated as

$$H_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2}\left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right)\right] & \text{if } |\phi| \le \varepsilon \\ 1 & \text{if } \phi > \varepsilon \end{cases} \tag{6}$$

The interfacial thickness ε represent an interfacial thickness over which the Heaviside function varies. Typically, the parameter ε is set to $\varepsilon = \alpha h$, where $\alpha > 1$ and h is a characteristic mesh size. This smoothed Heaviside function plays the role of a volume fraction of the liquid phase, and can be used for volume conservation constraints as will be seen in Section 3.4.1.

2.1.2. Redistancing of the level set

The solution to Eq. (3) does not in general preserve the signed distance property of the level set function ($|\nabla\phi| = 1$). To re-establish this property, a redistancing operation must be performed that preserves the position of the zero level set surface. A brief overview of this operation will be presented in this section, however unfamiliar readers should refer to the work of Sussman et al. [9] for a detailed introduction into the mathematics and numerics of this operation.

Redistancing of the level set is accomplished by solving to steady state an equation in pseudo-time τ:

$$\frac{\partial\bar{\phi}}{\partial\tau} + S_0(\phi_0)(|\nabla\bar{\phi}| - 1) = 0 \tag{7}$$

where the initial condition for Eq. (7) is the level set field at some physical time t, i.e., $\bar{\phi}(\boldsymbol{x}, \tau = 0) = \phi_0(\boldsymbol{x}) = \phi(\boldsymbol{x}, t)$. It is assumed that $\phi_0(\boldsymbol{x})$ does not satisfy the signed distance property.

The function $S_0(\phi_0)$ is a smoothed sign function that goes to 0 at the $\phi_0 = 0$ interface:

$$S_0(\phi_0) = 2H_\varepsilon(\phi_0) - 1 \tag{8}$$

Eq. (7) can be reformulated as an advection equation. Using $|\nabla\bar{\phi}| = \nabla\bar{\phi}\cdot\frac{\nabla\bar{\phi}}{|\nabla\bar{\phi}|}$ and rearranging results in

$$\frac{\partial\bar{\phi}}{\partial\tau} + \boldsymbol{w}\cdot\nabla\bar{\phi} = S_0(\phi_0) \tag{9}$$

where the redistancing velocity $\boldsymbol{w}$ is defined as:

$$\boldsymbol{w} = S_0(\phi_0)\frac{\nabla\bar{\phi}}{|\nabla\bar{\phi}|} \tag{10}$$

It can be seen from the form of the redistancing equation in Eq. (9) that information is propagated away from the stationary interface to recover the signed distance property of $\bar{\phi}$. At steady state, the solution $\bar{\phi}$ satisfies $|\nabla\bar{\phi}| = 1$, and is used to replace the current level set field $\phi(\boldsymbol{x}, t)$. A numerical discretization of Eq. (7) will introduce mass error in the individual phases, because the zero level set contour does not remain stationary during redistancing. As a result, additional constraints need to be embedded into Eq. (7), which will be discussed in Sections 3.4.1 and 3.4.2.

2.2. Incompressible two-phase flow

Incompressible two-phase flow is governed by the Navier–Stokes equation coupled with a mass conservation constraint:

$$\frac{\partial(\rho(\phi)\boldsymbol{u})}{\partial t} + \nabla\cdot(\rho(\phi)\boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla\cdot\left(\mu(\phi)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)\right) + \rho(\phi)\boldsymbol{g} + \boldsymbol{f}_{sf} \tag{11a}$$

$$\frac{\partial\rho(\phi)}{\partial t} + \nabla\cdot(\rho(\phi)\boldsymbol{u}) = 0 \tag{11b}$$

Applying the product rule to the left hand side of Eq. (11a), the chain rule to Eq. (11b) and utilizing Eq. (3) results in:

$$\rho(\phi)\left(\frac{\partial\boldsymbol{u}}{\partial t} + \nabla\cdot(\boldsymbol{u}\boldsymbol{u})\right) = -\nabla p + \nabla\cdot\left(\mu(\phi)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)\right) + \rho(\phi)\boldsymbol{g} + \boldsymbol{f}_{sf} \tag{12a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{12b}$$

where $\boldsymbol{u}$ is the flow velocity, p is the hydrostatic pressure and $\boldsymbol{g}$ is the gravitational acceleration. The material density $\rho(\phi)$ and viscosity $\mu(\phi)$ are calculated as functions of the level set field, as described in Eq. (5). Additionally, the force term $\boldsymbol{f}$ accounts for surface tension applied over the interface Γ. We utilize the continuum surface force (CSF) method proposed by Brackbill et al. [34] to implicitly apply surface tension effects over Γ. A traditional level set framework utilizes the following form for the surface tension forces:

$$\boldsymbol{f}_{sf} = \sigma\kappa(\phi)\boldsymbol{n}H'(\phi) \tag{13}$$

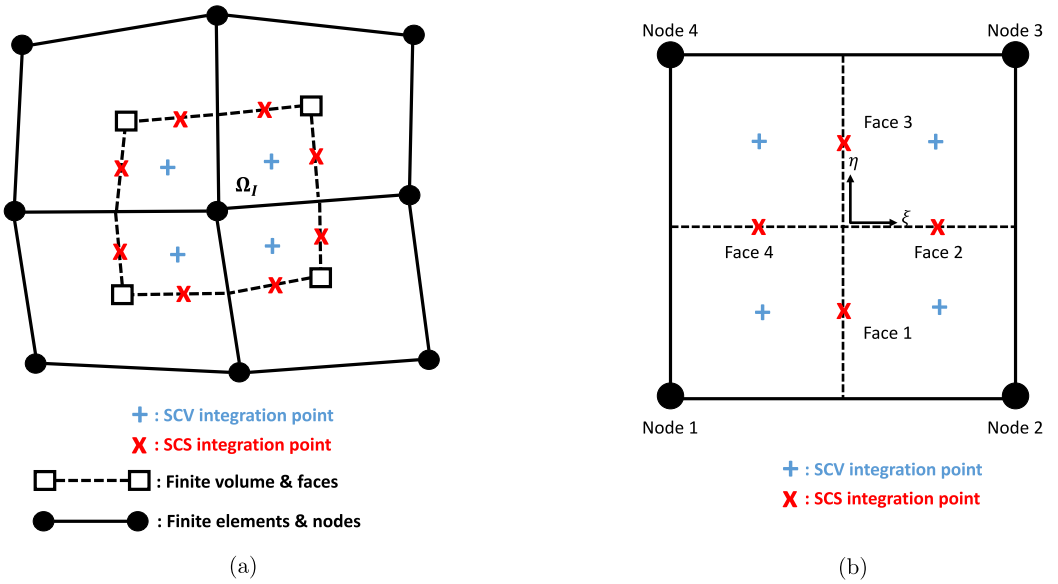


Fig. 1. (a) Control volume centered about a finite element node connected to a patch of 2D quadrilateral elements; (b) mapping for SCS and SCV integration points in the parent coordinate system for a bilinear quadrilateral element.

where σ is the surface tension coefficient and $H'(\phi) = \frac{\partial H}{\partial \phi}$. However, for a proper balanced-force implementation we utilize the gradient of the Heaviside or $\nabla H = \boldsymbol{n} H'(\phi)$. Thus, $\boldsymbol{f}_{sf}$ takes the form:

$$\boldsymbol{f}_{sf} = \sigma \kappa(\phi) \nabla H \tag{14}$$

3. Numerical formulation

In this work, a control volume finite element method (CVFEM) formulation is used to spatially discretize the governing equations reviewed in Section 2. This method combines the conservation properties associated with finite volumes with the geometric flexibility of finite element interpolation functions. The governing equations are posed in a conservative integral form and applied to discrete control volumes. Surface and volumetric integrations are then evaluated using finite element interpolation techniques.

3.1. CVFEM discretization

CVFEM utilizes discrete control volumes constructed around each node within a finite element mesh. Volumes are formed by connecting the centers of elements, edges, and (in 3D) faces of the original finite element mesh, forming a polygonal or polyhedral control volume (CV). For example, Fig. 1a shows a patch of 4 2D elements, and the CV (denoted Ω_I) formed around central node I. An individual element includes a sub-portion of the CVs associated with each of its connected nodes; these sub-portions are referred to as sub-control volumes (SCV), and the faces bounding them as sub-control surfaces (SCS). A canonical 4-node quadrilateral element in 2D includes 4 SCVs and 4 SCSs (Fig. 1b). Integrals over these volumes and surfaces are typically approximated using a single integration point at the center of the surface or volume, as shown in Fig. 1. A complete description of SCS and SCV for 3D hexahedral elements are given by Domino [47].

3.2. Incompressible two-phase flow

The two-phase Navier–Stokes equation posed in Eq. (12) is first posed in an integral form for each control volume Ω_I with a bounding surface Γ_I. Invoking the divergence theorem and assuming density is constant over Ω_I results in:

$$\rho_I \left(\int_{\Omega_I} \frac{\partial \boldsymbol{u}}{\partial t} d\Omega_I + \int_{\Gamma_I} (\boldsymbol{u}\boldsymbol{u}) \cdot \boldsymbol{n} d\Gamma_I \right) = \int_{\Omega_I} (\rho_I \boldsymbol{g} + \boldsymbol{f}_{sf}) d\Omega_I + \int_{\Gamma_I} (\mu(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T) - p\boldsymbol{I}) \cdot \boldsymbol{n} d\Gamma_I \tag{15}$$

$$\int_{\Gamma_I} \boldsymbol{u} d\Gamma_I = 0 \tag{16}$$

where $\boldsymbol{n}$ is the outward pointing normal from surface Γ_I and $\boldsymbol{I}$ is the identity tensor. A CVFEM discretization is then carried out by approximating surface integrals at the SCS integration points and volumetric integrals at the SCV points (with the exception of the time derivative term, which for convenience is evaluated at the node):

$$\rho_I \left(\frac{\partial \boldsymbol{u_I}}{\partial t} \Delta V_I + \sum_{ip=1}^{n_{scs}} (\boldsymbol{u}_{adv,ip}) \dot{v}_{ip} \right) = \rho_I \boldsymbol{g} \Delta V_I + \boldsymbol{f}_{sf,I} \Delta V_I + \sum_{ip=1}^{n_{scs}} (\mu_{ip} (\nabla \boldsymbol{u}_{ip} + \nabla \boldsymbol{u}_{ip}^T) - p_{ip} \boldsymbol{I}) \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \tag{17}$$

$$\sum_{ip}^{n_{scs}} \dot{v}_{ip} = 0 \tag{18}$$

where n_{scs} is the number of SCS integration points, n_{scv} is the number of SCV integration points, ΔA_{ip} is the surface area at a SCS integration point, and ΔV_I is the volume of Ω_I. The volume flow rate $\dot{v}_{ip}$ used in both (17) and (18) approximates $\mathbf{u}_{ip} \cdot \mathbf{n} \Delta A_{ip}$ over the SCS and is designed to satisfy (18) exactly; its form will be presented in section 3.2.1. The advected velocity $\boldsymbol{u}_{adv,ip}$ in (17) is an upwinded value, the form of which will be discussed in section 3.2.2. Other integration points values with subscript "ip" are evaluated using standard finite element shape functions:

$$\boldsymbol{u}_{ip} = \sum_{I=1}^{n_N} N_I^{ip} \boldsymbol{u}_I, \qquad \nabla \boldsymbol{u}_{ip} = \frac{\partial \boldsymbol{u}_{ip}}{\partial x_j} = \sum_{I=1}^{n_N} \frac{\partial N_I^{ip}}{\partial x_j} \boldsymbol{u}_I \tag{19}$$

where n_N is the number of nodes within an element and N_I^{ip} is the finite element shape function at a node I evaluated at an SCS or SCV integration point.

3.2.1. A balanced-force collocated pressure projection scheme and the formulation of $\dot{v}_{ip}$

Our proposed balanced-force formulation requires a pressure projection step in which the pressure gradient and surface tension force $\boldsymbol{f}_{sf}$ are evaluated consistently, allowing a discrete balance between the two for static cases with constant curvature. A backward Euler time integration scheme is used together with a collocated pressure projection method [48,49] to update the velocity and pressure variables. A variable density formulation similar to the one proposed by Almgren et al. [50] is implemented in this work. This fractional step scheme first solves Eq. (17) for an intermediate velocity $\hat{\boldsymbol{u}}$ with an initial pressure guess p^*.

$$\begin{aligned} \rho_I^{n+1} \left(\frac{\hat{\boldsymbol{u}}_I - \boldsymbol{u}_I^n}{\Delta t} \Delta V_I + \sum_{ip=1}^{n_{scs}} \hat{\boldsymbol{u}}_{adv,ip} \dot{v}_{ip}^n \right) &= \rho_I^{n+1} \boldsymbol{g} \Delta V_I + \boldsymbol{f}_{sf,I} \Delta V_I \\ &+ \sum_{ip=1}^{n_{scs}} (\mu_{ip} (\nabla \hat{\boldsymbol{u}}_{ip} + \nabla \hat{\boldsymbol{u}}_{ip}^T) + p_{ip}^* \boldsymbol{I}) \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \end{aligned} \tag{20}$$

After solving for a new pressure field p^{n+1} to enforce continuity (see below), the updated velocity at time t^{n+1} can then be computed:

$$\boldsymbol{u}_I^{n+1} = \hat{\boldsymbol{u}}_I - \frac{\Delta t}{\rho_I^{n+1}} (\nabla p_I^{n+1} - \nabla p_I^*) \tag{21}$$

In this expression, the pressure gradient at the nodes is evaluated according to a lumped L_2 projection based on the divergence theorem:

$$\nabla p_I = \frac{1}{\Delta V_I} \sum_{ip=1}^{n_{scs}} p_{ip} \boldsymbol{n}_{ip} \Delta A_{ip} \tag{22}$$

Eq. (22) can be modified to evaluate nodal gradients for any scalar value. A proper balanced-force implementation requires that the pressure gradients and surface tension forces be evaluated in the same manner; this allows an exact balance between the two under a constant curvature κ. Thus, for a constant surface tension this results in:

$$\boldsymbol{f}_{sf,I} = \frac{\sigma \kappa_I}{\Delta V_I} \sum_{ip=1}^{n_{scs}} H_{ip} \boldsymbol{n}_{ip} \Delta A_{ip} \tag{23}$$

Given σ and a constant curvature κ, and using Eq. (22) and (23) to consistently evaluate the pressure gradient and surface tension force, the exact nodal pressure solution $p_I = \sigma \kappa H_I$ satisfies the discrete, steady state equation and gives the correct analytical pressure jump across the interface. The curvature, which is a second derivative of the level set field Eq. (4), is also calculated through a nodally lumped L_2 projection using the nodal gradient of the level set field:

$$\kappa_I = \frac{1}{\Delta V_I} \sum_{ip=1}^{n_{scs}} \frac{\nabla \phi_{ip}}{|\nabla \phi_{ip}|} \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \tag{24}$$

The volume flow rate $\dot{v}_{ip}$ at the integration point can be computed based on (21), but with pressure gradients evaluated at the integration point to avoid a decoupling between the pressure and velocity fields on collocated meshes [51,52]. In order to discretely balance the pressure gradient, the surface tension force must also be evaluated at the integration point:

$$\dot{v}_{ip}^{n+1} = \left(\hat{\boldsymbol{u}}_{ip} + \Delta t \left[\left(\frac{\nabla p^*}{\rho^{n+1}} \right)_{ip} - \left(\frac{\boldsymbol{f}_{sf}}{\rho^{n+1}} \right)_{ip} - \left(\frac{1}{\rho^{n+1}} \right)_{ip} \left(\nabla p_{ip}^{n+1} - \sigma \kappa_{ip} \nabla H_{ip} \right) \right] \right) \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \tag{25}$$

For large density variations, the stability of the method can be sensitive to the form of interpolation of integration point values in the above expression. In our work, the following forms are used:

$$\nabla p_{ip} = \frac{\partial p}{\partial x_j} = \sum_{I=1}^{n_N} \frac{\partial N_I^{ip}}{\partial x_j} p_I, \quad \nabla H_{ip} = \frac{\partial p}{\partial x_j} = \sum_{I=1}^{n_N} \frac{\partial N_I^{ip}}{\partial x_j} H_I \tag{26a}$$

$$\left(\frac{1}{\rho} \right)_{ip} = \sum_{I=1}^{n_N} N_I^{ip} \left(\frac{1}{\rho_I} \right) \tag{26b}$$

$$\left(\frac{\nabla p}{\rho} \right)_{ip} = \sum_{I=1}^{n_N} N_I^{ip} \frac{\nabla p_I}{\rho_I}, \quad \left(\frac{\boldsymbol{f}}{\rho} \right)_{ip} = \sum_{I=1}^{n_N} N_I^{ip} \frac{\boldsymbol{f}_{sf,I}}{\rho_I} \tag{26c}$$

where ∇p_I and $\boldsymbol{f}_{sf,I}$ are evaluated according to Eq. (22) and (23), respectively.

The pressure at the new time step is determined by enforcing that the volume flow rates satisfy the discrete continuity equation (18), leading to a Poisson equation for p^{n+1}:

$$\begin{aligned} &\sum_{ip=1}^{n_{scs}} \Delta t \left(\frac{1}{\rho^{n+1}} \right)_{ip} \nabla p_{ip}^{n+1} \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \\ &= \left(\hat{\boldsymbol{u}}_{ip} + \Delta t \left[\left(\frac{\nabla p^*}{\rho^{n+1}} \right)_{ip} - \left(\frac{\boldsymbol{f}_{sf}}{\rho^{n+1}} \right)_{ip} + \left(\frac{1}{\rho^{n+1}} \right)_{ip} \left(\sigma \kappa_{ip} \nabla H_{ip} \right) \right] \right) \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \end{aligned} \tag{27}$$

The above algorithm will later be compared to a non-balanced force method that does not include the surface tension contribution in either the expression for volume flow rate or the resulting pressure Poisson equation. This results in the following expressions for $\dot{v}_{ip}$ and p^{n+1}:

$$\dot{v}_{ip}^{n+1} = \left(\hat{\boldsymbol{u}}_{ip} + \Delta t \left[\left(\frac{\nabla p^*}{\rho^{n+1}} \right)_{ip} - \left(\frac{1}{\rho^{n+1}} \right)_{ip} \left(\nabla p_{ip}^{n+1} \right) \right] \right) \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \tag{28a}$$

$$\sum_{ip=1}^{n_{scs}} \Delta t \left(\frac{1}{\rho^{n+1}} \right)_{ip} \nabla p_{ip}^{n+1} \cdot \boldsymbol{n}_{ip} \Delta A_{ip} = \left(\hat{\boldsymbol{u}}_{ip} + \Delta t \left(\frac{\nabla p^*}{\rho^{n+1}} \right)_{ip} \right) \cdot \boldsymbol{n}_{ip} \Delta A_{ip} \tag{28b}$$

3.2.2. Advection stabilization and higher order upwinding

Advective transport requires additional stabilization. In our formulation, stabilization is implemented by including upwinding to evaluate advected quantities at the integration points (e.g., $\mathbf{u}_{adv,ip}$ in Eq. (17)). For some general transported field θ, which may represent for example the velocity or level set field, the advected value θ_{adv} is decomposed into upwinded (θ_{upw}) and generalized central difference (θ_{cd}) contributions:

$$\theta_{adv} = \eta \theta_{upw} + (1 - \eta) \theta_{cd} \tag{29}$$

where η is a local Peclet blending function defined as:

$$\eta = \frac{Pe^2}{5 + Pe^2} \tag{30}$$

Pe is the local Peclet number at an SCS integration point, evaluated as:

$$Pe = \left| \frac{0.5(\boldsymbol{u}_R + \boldsymbol{u}_L) \cdot (\boldsymbol{x}_R - \boldsymbol{x}_L)}{\nu} \right| \tag{31}$$

where ν is the dynamic viscosity. Each SCS integration point has a "left" and "right" node associated with it, subscripted with L and R, respectively. For example, for Face 1 in Fig. 1b, the left and right node are nodes 1 and 2, respectively. The evaluation of θ_{upw} is determined by the sign of $\dot{v}_{ip}$ at the SCS integration point:

$$\theta_{upw,ip} = \begin{cases} \theta_L + \boldsymbol{d}_L \cdot \nabla\theta_L & \text{if } \dot{v}_{ip} < 0 \\ \theta_R + \boldsymbol{d}_R \cdot \nabla\theta_R & \text{if } \dot{v}_{ip} > 0 \end{cases} \tag{32}$$

The normal vector pointing from L toward R is used to determine the sign of the volume flow rate. The vectors $\boldsymbol{d}_L$ and $\boldsymbol{d}_R$ are distance vectors from the integration point to the left and right node, respectively.

A finite element interpolation is used to compute the central difference value in our formulation:

$$\theta_{cd,ip} = \sum_{I=1}^{n_N} N_I^{ip} \theta_I \tag{33}$$

Higher order spatial schemes can be achieved through the choice of θ_{cd}, but are beyond the scope of this work. For more information, interested readers are referred to the work of Domino [47].

3.3. Level set advection

Under a divergence free velocity, the advection of the level set presented in Eq. (3) can be reformulated into a conservative integral form

$$\int_{\Omega_I} \frac{\partial \phi}{\partial t} d\Omega_I + \int_{\Gamma_I} \boldsymbol{u} \cdot \boldsymbol{n} \phi d\Gamma_I = 0 \tag{34}$$

A CVFEM spatial and backward Euler time discretization of Eq. (34) results in

$$\left(\frac{\phi_I^{n+1} - \phi_I^n}{\Delta t} \right) \Delta V_I + \sum_{ip=1}^{n_{scs}} \phi_{upw,ip}^{n+1} \dot{v}_{ip}^n = 0 \tag{35}$$

where the volumetric flow rate $\dot{v}_{ip}$ is defined in Eq. (25) and the upwinded value $\phi_{upw,ip}$ is based on Eq. (32). It should be noted here that we use full upwinding on the level set function since $Pe \to \infty$ for the evolution of the level set field.

3.4. Level set redistancing

Analytically, Eq. (7) preserves the position of the zero level set of $\phi(\boldsymbol{x}, t)$ in pseudo-time. However, as mentioned in Section 2.1.2, a numerical solution of this equation introduces discretization errors that will inevitably perturb the zero level set surface. These errors are small at each pseudo-time step, but accumulation can cause a substantial change in mass over time. As a result, we must enforce additional constraints during the redistancing of the level set.

3.4.1. Interface constraint on redistancing

Sussman et al. [28] introduced a constraint at the zero level set contour during the time integration of Eq. (7). The change in integrated volume fraction over each control volume Ω_I is constrained to zero:

$$\int_{\Omega_I} (H_\varepsilon(\bar{\phi}) - H_\varepsilon(\phi_0)) d\Omega_I = 0 \tag{36}$$

where $\bar{\phi}$ is the redistanced level set field and ϕ_0 is the level set field prior to redistancing. We follow a similar approach and enforce Eq. (36) while integrating Eq. (9) in pseudo-time. First, we note that the CVFEM formulation can be thought of as a special case of a weighted residual method. We take a weighted residual form of Eq. (9):

$$\int_\Omega \delta\bar{\phi} \left(\frac{\partial \bar{\phi}}{\partial \tau} + \boldsymbol{w} \cdot \nabla\bar{\phi} - S_0(\phi_0) \right) d\Omega = \int_\Omega \delta\bar{\phi} R(\bar{\phi}) d\Omega = 0 \tag{37}$$

where $\delta\bar{\phi}$ is a weight function and $R(\bar{\phi})$ is the residual of Eq. (9), defined as:

$$R(\bar{\phi}) = \frac{\partial \bar{\phi}}{\partial \tau} + \boldsymbol{w} \cdot \nabla\bar{\phi} - S_0(\phi_0) \tag{38}$$

If $\delta\bar{\phi}$ is chosen to be a piece-wise constant equal to one on control volume Ω_I and zero elsewhere, we obtain the CVFEM formulation.

The following augmented Lagrangian term is added into Eq. (37) to enforce the constraint posed in Eq. (36)

$$\sum_I \lambda_I \int_{\Omega_I} (H_\varepsilon(\bar{\phi}) - H_\varepsilon(\phi_0)) d\Omega_I \tag{39}$$

where λ_I is a Lagrange multiplier associated with each control volume I. Taking a variation of Eq. (39) and adding it to Eq. (37) we obtain

$$\int_\Omega \delta\bar{\phi} R(\bar{\phi}) d\Omega + \sum_I \delta\lambda_I \int_{\Omega_I} \big(H_\varepsilon(\bar{\phi}) - H_\varepsilon(\phi_0)\big) d\Omega_I + \sum_I \lambda_I \int_{\Omega_I} \delta\bar{\phi} H'_\varepsilon(\bar{\phi}) d\Omega_I \tag{40}$$

where $H'_\varepsilon = \frac{\partial H_\varepsilon}{\partial \phi}$. Gathering $\delta\bar{\phi}$ and $\delta\lambda_I$ terms into separate equations and assuming $\delta\bar{\phi}$ to be piece-wise constant over a volume Ω_I and 0 elsewhere yields

$$\begin{aligned} &\int_{\Omega_I} \left(\frac{\partial \bar{\phi}}{\partial \tau} + \boldsymbol{w} \cdot \nabla\bar{\phi} - S_0(\phi_0) \right) d\Omega_I + \lambda_I \int_{\Omega_I} H'_\varepsilon(\bar{\phi}) d\Omega_I = 0 \\ &\text{Subject to: } \int_{\Omega_I} (H_\varepsilon(\bar{\phi}) - H_\varepsilon(\phi_0)) d\Omega_I = 0 \end{aligned} \tag{41}$$

We approximate the reinitialization velocity $\boldsymbol{w}$ as constant over the control volume, allowing us to formulate the advection term as a surface integral using the divergence theorem so that our CVFEM upwinding scheme discussed in Section 3.2.2 may be used.

$$\int_{\Omega_I} \frac{\partial \bar{\phi}}{\partial \tau} d\Omega_I = -\boldsymbol{w}_I \cdot \int_{\Gamma_I} (\bar{\phi}\boldsymbol{n}) d\Gamma_I + \int_{\Omega_I} S_0(\phi_0) d\Omega_I - \lambda_I \int_{\Omega_I} H'_\varepsilon(\bar{\phi}) d\Omega_I \tag{42}$$

In order to enforce the constraint given by Eq. (36), Eq. (42) is discretized into a predictor–corrector scheme using a backward Euler scheme in time and a CVFEM spatial discretization. The unconstrained advection equation in Eq. (9) is used to predict an intermediate field $\bar{\phi}^*$:

$$\left(\frac{\bar{\phi}_I^* - \bar{\phi}_I^k}{\Delta\tau} \right) \Delta V_I = -\boldsymbol{w}_I \cdot \left(\sum_{ip=1}^{n_{scs}} \boldsymbol{n}_{ip} \bar{\phi}^*_{upw,ip} \Delta A_{ip} \right) + S_0(\phi_0) \Delta V_I \tag{43}$$

where $\Delta\tau$ is the pseudo time step and the superscript k represents a pseudo time level. The parameter λ_I is then calculated for each control volume over a node I based on the intermediate field $\bar{\phi}^*$. Taking a linear expansion and making use of the constraint posed in Eq. (36) results in a Lagrange multiplier of the form

$$\lambda_I = -\frac{\int_{\Omega_I} \big(H_\varepsilon(\bar{\phi}^*) - H_\varepsilon(\phi_0)\big) d\Omega_I}{\Delta\tau \int_{\Omega_I} H'_\varepsilon(\bar{\phi}^*)^2 d\Omega_I} \approx -\frac{\sum_{ip=1}^{n_{scv}} \Big(H_\varepsilon(\bar{\phi}^*_{ip}) - H_\varepsilon(\phi_{0ip})\Big) \Delta V_{ip}}{\Delta\tau \sum_{ip=1}^{n_{scv}} H'_\varepsilon(\bar{\phi}^*_{ip})^2 \Delta V_{ip}} \tag{44}$$

where n_{scv} is the number of SCV integration points within an element. Finally, the intermediate field $\bar{\phi}^*$ is corrected by λ_I to minimize variations around the zero level set contour:

$$\bar{\phi}_I^{k+1} = \bar{\phi}_I^* + \frac{\Delta\tau\lambda_I}{\Delta V_I} \sum_{ip=1}^{n_{scv}} H'_\varepsilon(\bar{\phi}^*_{ip}) \Delta V_{ip} \tag{45}$$

3.4.2. Volume correction for redistancing

The predictor–corrector scheme formulated in Section 3.4.1 is designed to minimize variations in the zero level set contour during redistancing. However, this scheme does not address mass conservation errors incurred during numerical solution of level set advection (Eq. (35)). As a result, an additional procedure proposed by Smolianski [30] is applied wherein the re-distanced level set field $\bar{\phi}$ is perturbed by a global correction factor ζ to restore mass balance. The correction factor ζ is obtained by solving

$$\int_\Omega H_\varepsilon(\bar{\phi} + \zeta) d\Omega - \bar{H} = 0 \tag{46}$$

where H_ε is given by Eq. (6), Ω is the computational domain and $\bar{H}$ is the initial volume fraction integrated over the computational domain. It should be noted that Eq. (46) acts to restore the correct mass of each phase globally after each level set advection time step, but does not necessarily conserve mass locally; the Lagrange multiplier constraint formulated in Eq. (44) is used to conserve mass locally over each pseudo time step during redistancing.

Solving for the global volume corrector ζ requires solving Eq. (46), which is a nonlinear equation that requires a linearization. We implement an incremental formulation in ζ such that $\zeta^{m+1} = \zeta^m + \Delta\zeta^m$, where the superscript m represents an iterative increment. Taking a Newton–Raphson (N–R) formulation of Eq. (46) and linearly expanding around $\bar{\phi} + \zeta$ results in

$$\Delta\zeta^m = \frac{\bar{H} - \int_\Omega H_\varepsilon(\bar{\phi}^m + \zeta^m)d\Omega}{\int_\Omega H'_\varepsilon(\bar{\phi}^m + \zeta^m)d\Omega} \tag{47}$$

The redistanced level set $\bar{\phi}$ is updated after each N–R iteration ($\bar{\phi}^{m+1} = \bar{\phi}^m + \Delta\zeta^m$). Taking a CVFEM discretization of Eq. (47) and integrating over all of the control volumes results in

$$\Delta\zeta^m = \frac{\bar{H} - \sum_{I=1}^{n_{CV}} H_\varepsilon(\bar{\phi}_I^m + \zeta_I^m)\Delta V_I}{\sum_{I=1}^{n_{CV}} H'_\varepsilon(\bar{\phi}_I^m + \zeta_I^m)\Delta V_I} \tag{48}$$

where n_{CV} is the number of control volumes within the computational domain.

3.5. Solution algorithm

The numerical scheme for our proposed methodology is summarized as follows:

1. Advect the level set function ϕ by solving Eq. (35)
2. Solve for the redistanced level set field $\bar{\phi}$ using ϕ^{n+1} as an initial condition
   (a) Solve for an intermediate field $\bar{\phi}^*$ with Eq. (43)
   (b) Calculate the Lagrange multiplier λ_I at each nodes with Eq. (44)
   (c) Correct the intermediate field for $\bar{\phi}^{k+1}$ using λ_I with Eq. (45)
   (d) Repeat steps (a) to (c) until a desired pseudo-time state is reached
3. Correct the redistanced level set field $\bar{\phi}^{k+1}$ through a N–R iteration of Eq. (48)
4. Calculate viscosity and density fields from Eq. (5) and curvatures and normal from Eq. (4)
5. Solve for the velocity field using a fractional step method on the Navier–Stokes equation
   (a) Solve for a provisional velocity field $\hat{\boldsymbol{u}}$ through Eq. (17)
   (b) Solve for the new pressure field p^{n+1} through Eq. (27)
   (c) Correct the velocity field through Eq. (21)
6. Store the conservative volume fluxes calculated from Eq. (25) at the SCS integration points
7. Repeat steps 1 to 6 until a desired time state is reached

In the following examples, the redistancing algorithm listed in step 2 is run until a pseudo-time level that is 5 times the characteristic element length. The advection of the level set and the Navier–Stokes solution algorithm is carried out with second-order upwinding with van Leer slope limiters enforced. This formulation has been implemented in Nalu, an open source, massively parallel, generalized unstructured CVFEM code developed at Sandia National Laboratories for low Mach number flows [47].

4. Numerical results

4.1. 3D sphere advection under uniform flow

We first demonstrate the proposed level set algorithm with a simple test problem of advecting a spherical interface through a rectangular channel. A constant uniform velocity 1 in the x direction is imposed everywhere within the channel. The configuration is shown in Fig. 2. The cross-section of the rectangular channel is a square with a width $w = 4$ and the length of the channel is $L = 20$. The dimensions of the domain are $-2.0 \le x \le 2.0$, $-2.0 \le y \le 2.0$ and $-10.0 \le z \le 10.0$. The diameter of the sphere is $D = 3$ with center initially at $(0.0, 0.0, -7.0)$. The domain is discretized with a structured uniform mesh with cubical elements of size $h = 0.1$, giving 30 elements across the sphere diameter. A time step of $\Delta t = 0.125$ is

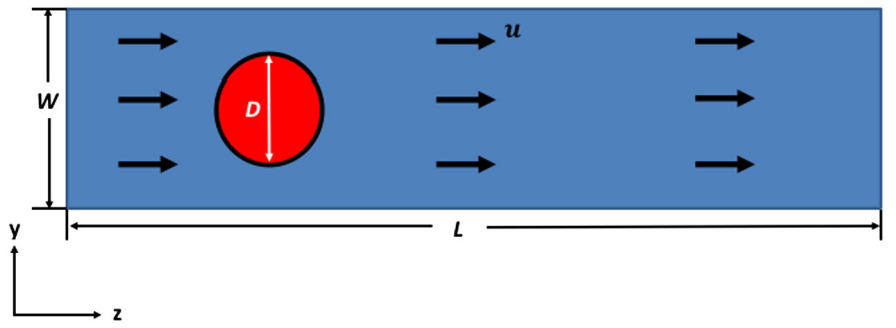


Fig. 2. Setup for sphere advection test.

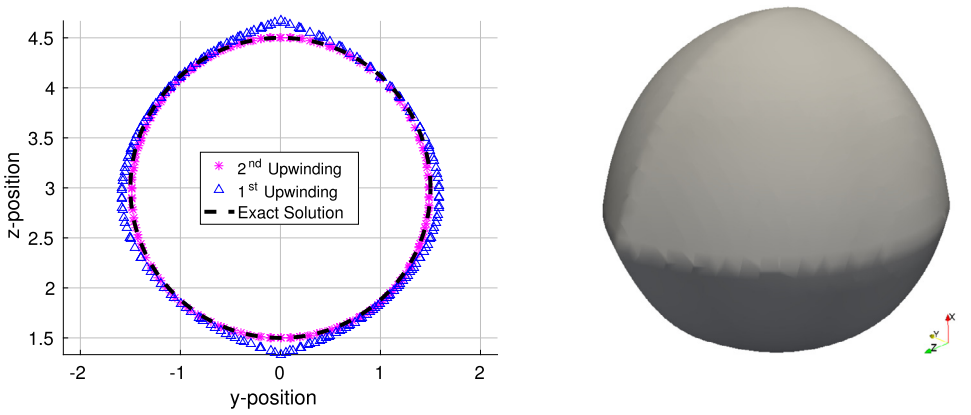


Fig. 3. Final interface contours for the sphere advection test visualized by (a) overlapping 2D contours compared with the exact solution and (b) a 3D surface contour for the case using first order upwinding.

used and the simulation is run until $t = 10$; thus in the exact solution the sphere be translated 10 length units in the x direction.

The interface thickness spans approximately 2 elements ($\varepsilon = 2h = 0.2$). To compare the effects of the upwinding scheme on redistancing, we perform redistancing using both first and second order upwinding; second order upwinding is used on the level set advection itself. The redistancing scheme is performed after every advection step to a final time of $\tau = 0.5$. For this test case, the Lagrange multiplier to enforce the constraint posed in 3.4.1 is not enforced; by doing this we see the effects of upwinding on the level set redistancing formulation.

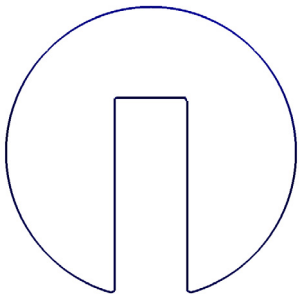
Fig. 3 shows the results for both cases using first and second order upwinding. As seen in Fig. 3b, using only first order upwinding results in a distortion of the sphere in regions where the normal of the interface is oriented diagonally across an element, whereas a second order scheme better preserves the shape. Fig. 3a displays the 2D contours of the interface for the both cases along with the exact solution; the 2D contours are taken from a planar slice in the y–z plane at $x = 0$.

4.2. Zalesak's rotating disk

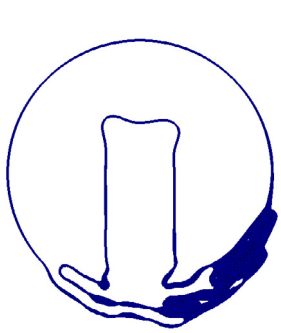
For geometries more complex than the sphere used in Section 4.1, additional constraints as presented in Sections 3.4.1 and 3.4.2 must be added to ensure that both the shape of the interface and the volume it encompasses are preserved, as demonstrated using Zalesak's rotating disk test [53]. A slotted disk is placed at the center of a unit square; the slot is 0.2 units wide and 0.65 units deep. Fig. 4a shows the initial position and Heaviside profile of the disk for a 100×100 quad mesh. The slotted disk rotates under a prescribed divergence-free rotational velocity field given as:

$$\begin{aligned} u_x(y) &= 2\pi y \\ u_y(x) &= -2\pi x \end{aligned} \tag{49}$$

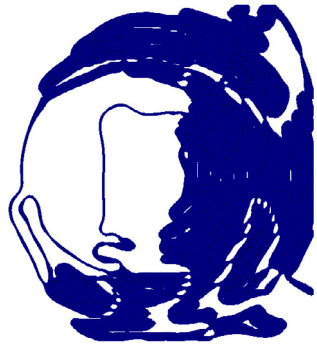
For this test case, the disk is rotated for 2 full revolutions to assess the accuracy of our proposed level set formulation. Second order upwinding is used on both the level set advection and its subsequent redistancing. A physical time step of $\Delta t = 0.0025$ is used along with a pseudo time step of $\Delta\tau = 0.01$. This test case is run with and without the interface constraints and volume correction formulated in Sections 3.4.1 and 3.4.2, respectively.



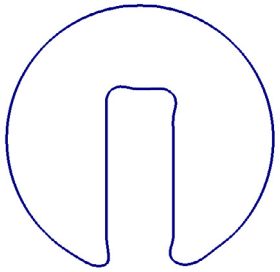
(a) Initial Disk Profile



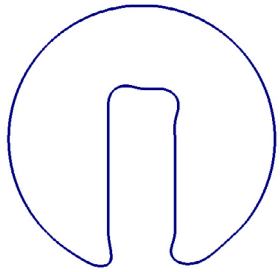
(b) 1 revolution (without constraint)



(c) 2 revolutions (without constraint)



(d) 1 revolution (with constraint)



(e) 2 revolutions (with constraint)

Fig. 4. Time evolution for rotating disk. The Lagrange multiplier constraint is only enforced in (d) and (e).

Figs. 4b and 4c show the profile of the slotted disk without using the interface constraint and volume correction. As seen in Fig. 4b, after 1 revolution the profile of the slotted disk rapidly deteriorates at the sharp corners of the slot. Fig. 4c shows that as the disk continues to rotate, the profile is eventually destroyed and does not retain its original shape.

Figs. 4d and 4e show the slotted disk profile when using both the interface constraint and volume correction. Fig. 4d shows that after 1 revolution, the profile of the slotted disk is maintained well, with the only significant distortion seen at the sharp corners around the slot. Although the Lagrange multiplier constrains error during the redistancing step, the level set advection process is still subject to artificial numerical diffusion, leading to smearing of the corners. The slotted disk profile is still maintained even after a second revolution, as seen in Fig. 4e. In addition, the relative difference between the initial area of the slotted disk profile and after 2 revolutions is summarized in Table 1. As seen in this table, with both the Lagrange multiplier and global volume constraint enforced, area is preserved within an order of 1×10^{-16} over time.

4.3. 3D interface deformation field

A 3D Enright deformation problem [54] is used to assess the performance of the level set and subsequent redistancing algorithm in the presence of complex flows. This problem has been widely-used [54,12,55,56] to assess the accuracy and

Table 1
Area conservation for Zalesak's disk problem.

	Relative difference in area
No Lagrange multiplier constraint	7.456×10^{-3}
Lagrange multiplier constraint enforced	4.409×10^{-16}

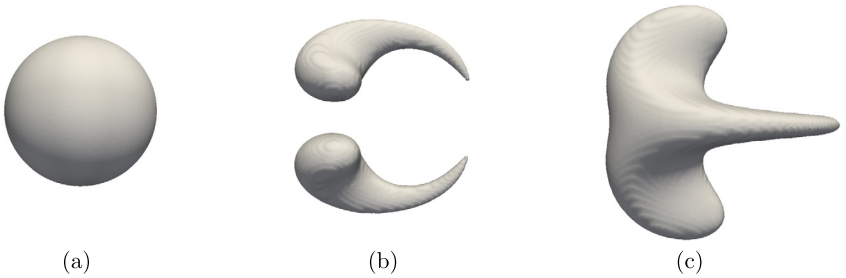


Fig. 5. Enright deformation problem on a structured $150 \times 150 \times 150$ hexahedral mesh with period $T = 3$ using second order upwinding on redistancing without any interface constraints. Isosurface of $\phi = 0$ at (a) $t = 0.0$ (b) $t = 1.5$ and (c) $t = 3.0$.

robustness of interface-capturing solvers under complex flows that dramatically distort the interface over time. The computational domain is a unit cube, with a sphere of radius 0.15 initially centered around (0.35, 0.35, 0.35). A spatially and temporally varying velocity field is prescribed to convect the spherical profile:

$$u_x = 2\sin^2(\pi x)\sin(\pi y)\sin(2\pi z)\cos\left(\frac{\pi t}{T}\right) \tag{50a}$$

$$u_y = -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos\left(\frac{\pi t}{T}\right) \tag{50b}$$

$$u_z = -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos\left(\frac{\pi t}{T}\right) \tag{50c}$$

where T is a time period and t represents the current time of the simulation. This velocity profile is designed to severely deform the sphere until $t = \frac{T}{2}$; after this the velocity profile is reversed and the sphere location and shape are analytically recovered at $t = T$. For this particular problem, a mesh resolution of $150 \times 150 \times 150$ and period of $T = 3.0$ are used. A total of four test cases are performed: three use a structured hexahedral mesh with a resolution of $150 \times 150 \times 150$ (shown in Figs. 5–7), and the last uses an unstructured tetrahedral mesh with an element size equivalent to 1/150 (shown in Fig. 8), with 27,867,795 elements and 5,053,846 nodes. In addition, Table 2 shows the relative difference in volume between the beginning and end of the simulation for each test case.

To illustrate the importance of the interface constraint, we run this problem without the interface constraint and use second order upwinding on the redistancing equation (Fig. 5). Additional test cases are run with the interface constraint while using first (Fig. 6) and second (Fig. 7) order upwinding during redistancing. Lastly, a test case is run using second order upwinding on an unstructured tetrahedral mesh (Fig. 8). The interface constraint plays a crucial role in preserving the interface location during redistancing under highly complex flows; this can be seen at the halfway point $t = 1.5$ in Fig. 5b, where a significant amount of volume has been lost and at the final recovered state $t = 3.0$ in Fig. 5c where the profile is nowhere near spherical. Employing the interface constraint during redistancing yields a much better result; however it can be seen by comparing Figs. 6b and 7b that at the halfway point, the second order upwinding case is able to better resolve the thin membrane region at the center. Moreover, comparing the final shapes in Figs. 6c and 7c shows that the second order upwinding case is able to provide a more accurate recovery.

The second order upwinding results are comparable to similar interface capturing approaches that use a volume of fluid [55] or coupled volume of fluid-level set approach [12,56] on a similar grid resolution. However, these results are not as accurate when compared to approaches such as the hybrid particle level set method [54]; that method is able to recover a spherical profile, but at the cost of introducing additional complexities such as particle tracking which become more difficult to implement in parallel.

The last test case demonstrate that we are able to solve this problem on an unstructured tetrahedral mesh; tetrahedral elements are used to discretize the cubic domain with the same characteristic element length for the $150 \times 150 \times 150$ hexahedral case. It can be seen from Fig. 8 that the method is able to approximately recover the hexahedral first order upwinded case on an unstructured tetrahedral mesh with second order upwinding. For the hexahedral first and second order upwinding case and the unstructured tetrahedral case, the relative difference between the volume of the sphere at $t = 3.0$ and $t = 0.0$ is on the order of 10^{-16} or machine accuracy; this is a result of using the volume correction proposed in Section 3.4.2.

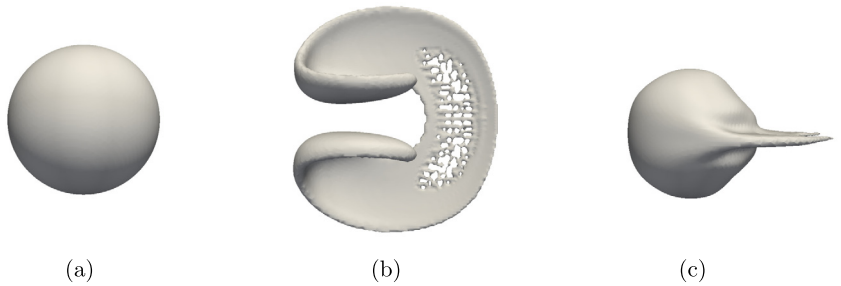


Fig. 6. Enright deformation problem on a structured $150 \times 150 \times 150$ hexahedral mesh with period $T = 3$ using the proposed interface constraint and first order upwinding on redistancing. Isosurface of $\phi = 0$ at (a) $t = 0.0$ (b) $t = 1.5$ and (c) $t = 3.0$.

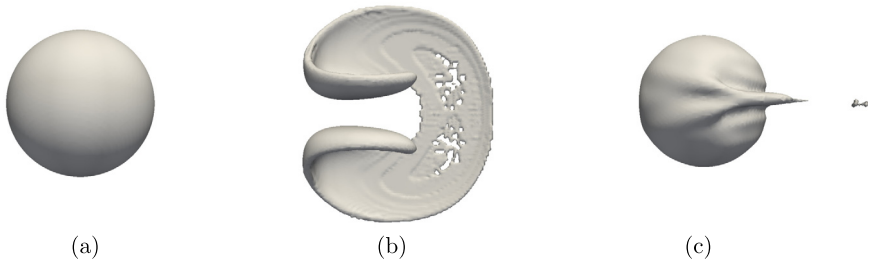


Fig. 7. Enright deformation problem on a structured $150 \times 150 \times 150$ hexahedral mesh with period $T = 3$ using the proposed interface constraint and second order upwinding on redistancing. Isosurface of $\phi = 0$ at (a) $t = 0.0$ (b) $t = 1.5$ and (c) $t = 3.0$.

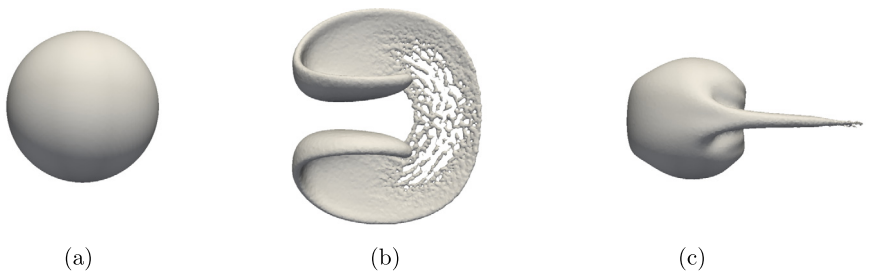


Fig. 8. Enright deformation problem on an unstructured tetrahedral mesh with period $T = 3$ using the proposed interface constraint and second order upwinding on redistancing. Isosurface of $\phi = 0$ at (a) $t = 0.0$ (b) $t = 1.5$ and (c) $t = 3.0$.

Table 2
Volume conservation for 3D Enright interface deformation test.

	Relative difference in volume
Second order upwinding without Lagrange multiplier constraint (hexahedral mesh)	2.269×10^{-3}
First order upwinding with Lagrange multiplier constraint (hexahedral mesh)	2.204×10^{-16}
Second order upwinding with Lagrange multiplier constraint (hexahedral mesh)	1.351×10^{-16}
Second order upwinding with Lagrange multiplier constraint (tetrahedral mesh)	3.134×10^{-16}

4.4. Static pressure test

Section 4.2 and 4.3 show that imposing additional constraints and corrections as well as implementing a second order upwinding method on redistancing can accurately preserve more complex geometries as well as the volume enclosed by the interface. In this section, we analyze an equilibrium inviscid sphere within a cubical domain and demonstrate the effect of the balanced-force algorithm in maintaining a stationary solution. Simulation and domain parameters are those used by Williams et al. [57], Francois et al. [37] and Herrmann et al. [39]. A cubical domain with dimensions $8 \times 8 \times 8$ contains a spherical droplet of radius $r = 2$ located at its center. No-slip conditions are applied at the boundaries, and a surface tension force given in Eq. (13) is applied at the interface. The density inside the sphere is set to $\rho_1 = 1$ and the density outside of the sphere, ρ_2, is varied. In the absence of gravitational forces, the surface tension is balanced by a pressure difference across the interface, resulting in a stationary bubble. The pressure difference can be calculated analytically with the Young–Laplace equation:

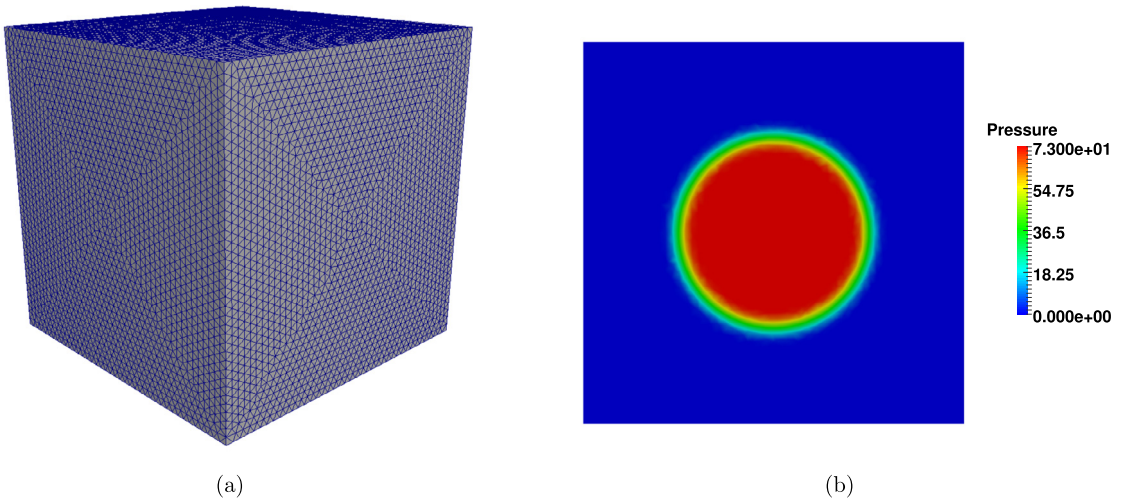


Fig. 9. (a) Unstructured tetrahedral mesh used to solve the inviscid static droplet problem and (b) sample pressure contour on a planar cut through the center of the cubic domain. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 3
Errors in velocity and pressure without the balanced force algorithm after a single time step for varying density ratios in the inviscid, equilibrium sphere test case using a prescribed curvature and tetrahedral mesh.

ρ_1/ρ_2	$L_\infty(\boldsymbol{u})$	$E(\Delta p_{max})$
1	2.666×10^{-5}	0.3229
10^3	9.789×10^{-7}	8.335
10^5	3.762×10^{-7}	8.6122
10^{10}	4.196×10^{-7}	8.6122

$$\Delta p = \sigma\kappa = \frac{2\sigma}{r} \tag{51}$$

Here, Δp is the pressure difference across the interface. The surface tension coefficient for this problem is set to $\sigma = 73$ so that the analytical pressure difference is $\Delta p = 73$. This problem is solved using an unstructured tetrahedral mesh with a characteristic element length of $h = 0.2$ (Fig. 9a). A sample solution of the static pressure field is shown in Fig. 9b. This problem has been extensively investigated [37,57,39] to assess the effects of parasitic currents for the case of a constant σ and a prescribed, constant curvature, it was found that balanced force algorithms were able to eliminate any parasitic currents, whereas non-balanced force algorithms still contained these currents. Quantitative measurements for assessing this problem include the maximum or L_∞ norm of the velocity and the error in the pressure $E(\Delta p_{max})$:

$$E(\Delta p_{max}) = |\max(p_I) - \min(p_I)| - \Delta p_{exact} \tag{52}$$

where $\Delta p_{exact} = 73$ for this test case.

To demonstrate the advantages of the balanced force algorithm, we prescribe the curvature only for these simulations with $\kappa = 1.0$. We prescribe the curvature so that we eliminate any errors from numerical computations of the curvature; this means that any parasitic velocities generated in these simulations are from numerical imbalances between the pressure gradient and surface tension force term. This approach to demonstrating force imbalance error by prescribing the exact curvature follows that of Francois et al. [37]. We solve this problem using the proposed balanced force algorithm described in Section 3.2.1; these results will be compared to a non-balanced force solution that only considers the surface tension in the velocity predictor step with pressure and the volumetric flow rate being calculated through Eqs. (28b) and (28a), respectively. Table 3 shows the results when using the non-balanced force algorithm and Table 4 shows the results with the proposed balanced force algorithm after a single time step size of $\Delta t = 10^{-6}$ for varying density ratios up to 10^{10}. As seen from Table 4, errors are on the order of machine zero, even for extremely large density ratios. It should be noted that values for $L_\infty(\boldsymbol{u})$ and $E(\Delta p_{max})$ lower than machine error 10^{-16} are truncated in the computation; therefore calculations that result in lower values are only reported as $\leq 10^{-16}$. Comparison of Table 3 and (28a) shows that, without the balanced force algorithm, $L_\infty(\boldsymbol{u})$ is nonzero and shows a dependence on the density ratio, while error in the pressure $E(\Delta p_{max})$ is significant, especially at high density ratios; the balanced force algorithm virtually eliminates both parasitic currents and pressure error for this prescribed curvature case.

Table 4
Errors in velocity and pressure with the proposed balanced force algorithm after a single time step for varying density ratios in the inviscid, equilibrium sphere test case using a prescribed curvature and tetrahedral mesh.

ρ_1/ρ_2	$L_\infty(\boldsymbol{u})$	$E(\Delta p_{max})$
1	4.496×10^{-15}	$\leq 10^{-16}$
10^3	$\leq 10^{-16}$	$\leq 10^{-16}$
10^5	$\leq 10^{-16}$	$\leq 10^{-16}$
10^{10}	$\leq 10^{-16}$	$\leq 10^{-16}$

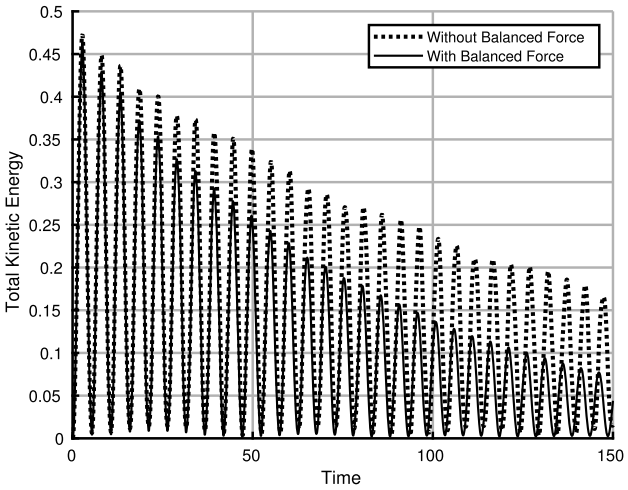


Fig. 10. Plot of the total kinetic energy vs. time for the oscillating droplet case using both a balanced and non-balanced force algorithm.

4.5. Oscillating droplet with surface tension

We perform an oscillating droplet test to assess our proposed balanced force algorithm, using the test case described by Torres et al. [36]. A drop with an initially elliptical interface, given by the equation $x^2/9 + y^2/4 = 1$, is placed in the center of a 20×20 square domain. As in Torres et al. [36] and Francois et al. [37], the domain is partitioned with a 64×64 mesh. There are no gravitational forces, and fluid properties are assigned as $\rho_1 = 1, \rho_2 = 0.01, \mu_2 = 0.01, \mu_2 = 5 \times 10^{-5}$, and $\sigma = 1$. The subscript 1 and 2 refer to properties inside and outside of the droplet, respectively. For this problem, surface tension forces act on the droplet to minimize the free-surface energy, and cause it to oscillate around an equilibrium profile.

Fig. 10 shows the evolution of the total kinetic energy with and without the balanced force algorithm. The total kinetic energy is computed as $\frac{1}{2}\int_\Omega \rho \boldsymbol{u} \cdot \boldsymbol{u} d\Omega$. When using the balanced-force algorithm, the oscillation frequency and initial peak in kinetic energy are similar to those observed by Torres et al. [36] and Francois et al. [37]. In addition, the amplitude decay when using the balanced force algorithm is similar to that presented by Francois et al. [37], who also used a balanced force algorithm. Without the balanced force algorithm, it can be seen in Fig. 10 that the decay rate is smaller than the results when using the balanced force algorithm; moreover, the decay profile is less uniform, and energy minima do not reach as close to zero as for the balanced force case. These observations are attributed to small near-surface flow structures driven by the force imbalance.

4.6. 2D rising bubble simulation and convergence rate

Section 4.4 demonstrates that our proposed coupled level set and momentum solver is able to obtain the correct solution for simple static problems. For this test we run a 2D rising bubble problem and run a subsequent convergence study for the rise velocity. The material parameters for this simulation are listed in Table 5 under Case 1. Properties and values with subscripts 1 and 2 belong to the surrounding fluid and bubble phase, respectively.

This class of problems has been subject to many numerical and physical experiments to characterize the shape and rise velocities of the bubble. The hydrodynamics of this problem is governed by the non-dimensional Reynolds (Re) and Bond (Bo) number:

Table 5
Material properties listed for both rising bubble test cases.

Case	ρ_1	ρ_2	μ_1	μ_2	σ	Re	Bo
1	1000	1	10	0.1	1.96	35	125
2	1000	100	10	1	24.5	35	10

$$Re = \frac{\rho_1 \sqrt{g}(2r_0)^{3/2}}{\mu_1}$$
$$Bo = \frac{4\rho_1 g(r_0)^2}{\sigma} \tag{53}$$

The Bo number measures the ratio of gravitational forces to surface tension forces; as this number increases, gravitational forces become increasingly dominant and will tend to deform the bubble interface. Conversely, for lower Bo numbers surface tension will play a larger role and act to minimize the surface energy and maintain the curvature of the bubble.

A uniform quad mesh is used to discretize the computational domain. This test case is run with 5 separate element sizes; $\Delta x = 0.001$, $\Delta x = 0.005$, $\Delta x = 0.01$, $\Delta x = 0.0125$ and $\Delta x = 0.02$. For all element sizes, a time step of $\Delta t = 0.002$ is used. The rise velocity evolution predicted from using $\Delta x = 0.001$ is used as the reference solution to measure convergence. The rise velocity U for each simulation is calculated as a integral-weighted average:

$$U = \frac{\int_{\Omega_2} u_z d\Omega_2}{\int_{\Omega_2} d\Omega_2} \tag{54}$$

The error in the rise velocity U at a specific time point is then calculated as

$$E_{rise} = \frac{|U_h - U_{0.001}|}{U_{0.001}} \tag{55}$$

Here, the subscript indicates the element size for each grid. This particular rising bubble case has been previously simulated by Hysing et al. [58]. In their tests those authors compared this case among simulations codes, and each demonstrates a filament-like breakup at the edges of the bubble. However, in this regime they observed each code had a different prediction of the terminal bubble shape and the breakup behavior at the edges, rendering the results inconclusive. Thus, we choose to run this simulation to a time where there is still a moderate amount of deformation to the bubble but still within a regime where there is no extreme filament-like breakup within the bubble.

We run this simulation to a time $t = 1.4$ with a fixed time step of $\Delta t = 0.002$ for all mesh sizes. A fixed time step across all meshes is used to demonstrate the spatial convergence of our proposed formulation; in this way we isolate the dependence of error on the mesh size. The time step for this case is chosen such that the Courant number for the highest resolution case ($\Delta x = 0.001$) is below 1.0, and thus well within the stable regime for our implicitly integrated method (which is stable even for Courant numbers above unity). We measure the error in the predicted rises velocities for $\Delta x = 0.005$, $\Delta x = 0.01$, $\Delta x = 0.0125$ and $\Delta x = 0.02$ against the reference rise velocity for $\Delta x = 0.001$ using Eq. (55). Fig. 11a shows the rise velocity evolution for all mesh sizes, and Fig. 11 shows a log–log plot of the errors against the element sizes at time $t = 1.4$. As seen from Fig. 11, we see that the rise velocity of the bubble converges to a solution with a second-order behavior in space.

4.7. 3D rising bubble simulation

We extend the example of the previous section to 3D; an initially stationary spherical bubble of radius 0.25 is initially located in a column of liquid and begins to rise due to buoyancy forces exerted in the negative z-direction. The dimensions of the problem are $-0.5 \leq x \leq 0.5$, $-0.5 \leq y \leq 0.5$ and $-1.0 \leq z \leq 1.0$ and the bubble is placed initially at $(0.0, 0.0, -0.5)$. Fig. 12a shows the setup for this problem and Fig. 12b shows the unstructured mesh used to solve this problem. No-slip boundary conditions are applied at all of the walls and the gravity vector $\mathbf{g}$ is defined as $(0, 0, -0.98)$.

Two test cases are run here, one in a high Bo regime to allow large deformations to the rising bubble and one in a low regime so that surface tension plays a greater role to maintain a spherical bubble shape. The parameters are chosen to match those given by Safi et al. [59], which offers a detailed benchmark comparison for terminal rising velocities U and bubble shape for this problem that is consistent across 3 separate numerical codes. Material properties used for both test cases are given in Table 5. Both cases are run until a final time of $t = 3.0$. In our simulations, 3 separate mesh sizes are used and each case uses a fixed time step of $\Delta t = 0.01$ to assess the convergence of our proposed method. Second-order upwinding is used on the level set advection, redistancing and momentum solve for both cases.

4.7.1. Case 1: low surface tension coefficient

The parameters used for the first test case are listed in Table 5. As seen from this table, the density and viscosity ratio are 1000 and 100, respectively. Here we have $Bo = 125$, thus we expect to see large deformations on the bubble interface

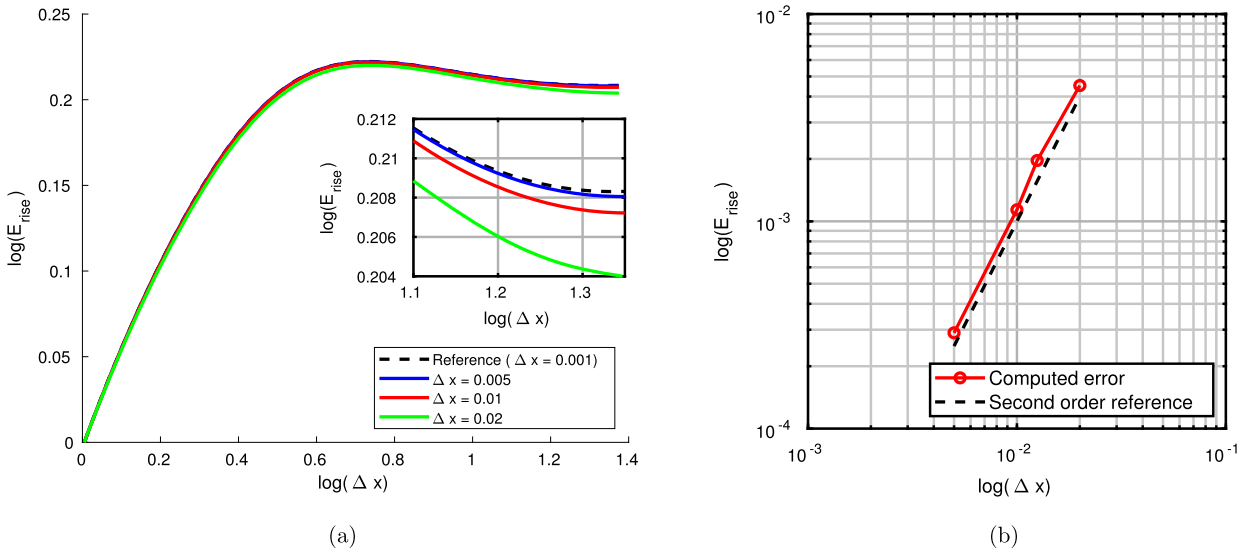


Fig. 11. (a) Time evolution for rise velocity for 3 different element sizes and (b) close up view of rise velocities.

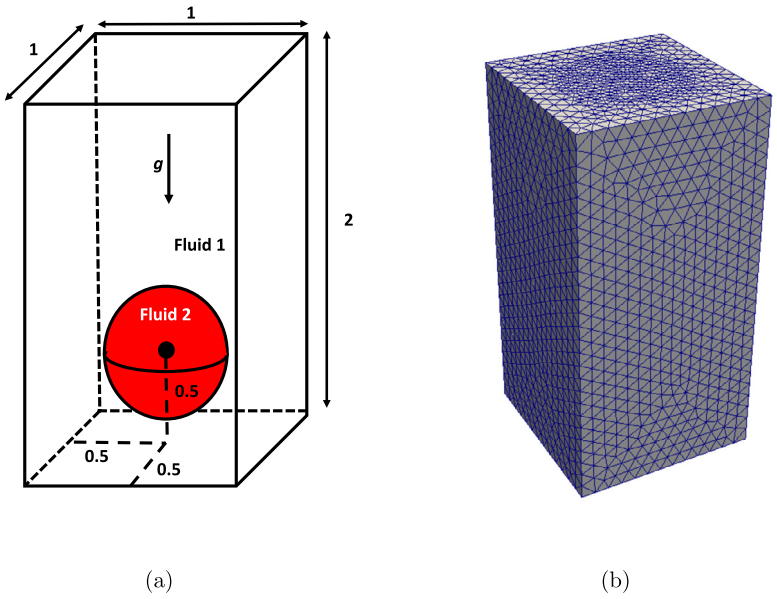


Fig. 12. (a) Schematic for 3D Rising Bubble Problem and (b) sample unstructured tetrahedral mesh which the problem is solved on.

as it rises through the liquid. Fig. 13 shows the 3D contours of the $\phi = 0$ level set at different times. As seen in Fig. 13a, after $t = 1.0$ we see that a dimple forms at the bottom of the bubble due to flow-induced deformation. As the bubble rises even further, the bubble begins to form a spherical cap in Fig. 13b and maintains this shape, as seen in Fig. 13c, until the end of the simulation at $t = 3.0$.

Fig. 14 shows the rise velocity of the bubble vs. time for various grids; Fig. 14b shows a zoomed-in profile of the terminal rise velocities. As seen from this figure, as we refine the mesh size the predicted terminal rise velocity converges to approximately $U = 0.31$.

In addition, Fig. 15 compares the predicted evolution of the rise velocity for our finest mesh with the reference rise velocity evolution from the work of Safi et al. [59]. As seen in this figure, our predicted results are in good agreement with their reference rise velocity evolution. Finally, we look at the terminal shape of the bubble as well as the difference in volume over time. Fig. 16a shows the terminal bubble shape for various element sizes. The terminal bubble shape converges to a profile as the mesh size decreases. The relative difference in the volume of the bubble over time is plotted in Fig. 16b, and it can be seen that the volume enclosed by the interface is preserved up to 1×10^{-16} or on the order of machine accuracy.

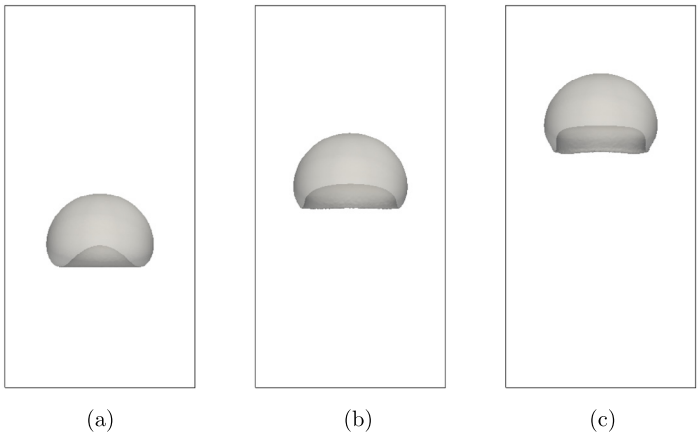


Fig. 13. Time evolution of the bubble for case 1 at time (a) $t = 1.0$ (b) $t = 2.0$ and (c) $t = 3.0$.

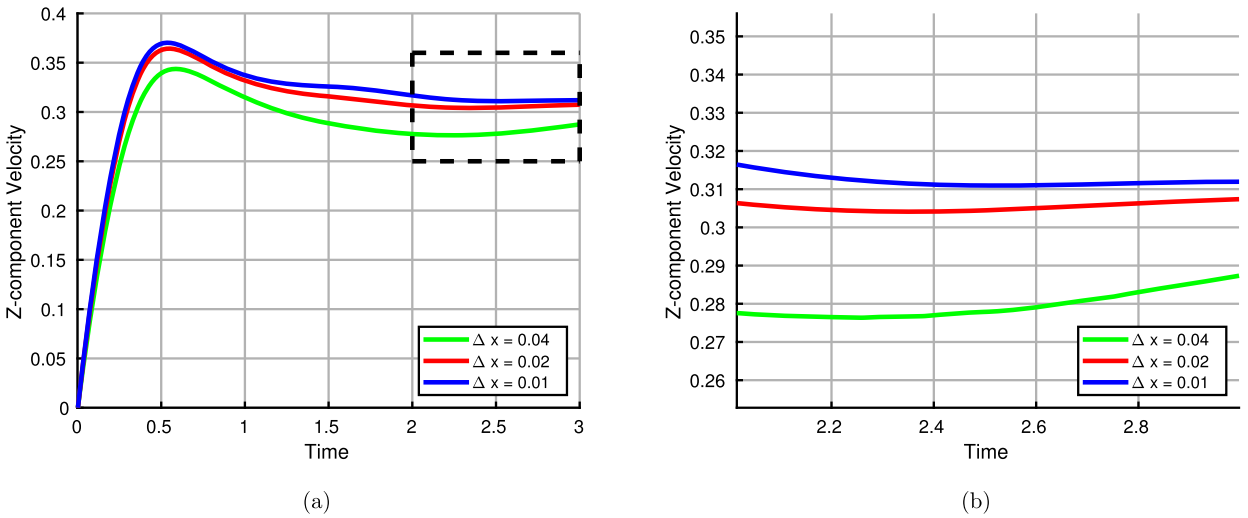


Fig. 14. (a) Time evolution for rise velocity for 3 different element sizes and (b) close up view of rise velocities.

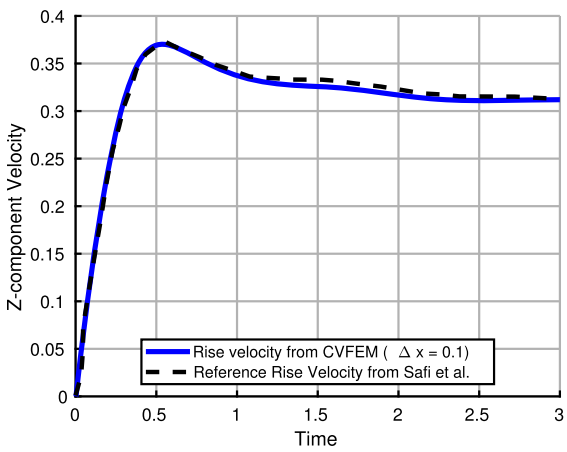


Fig. 15. Comparison between time evolution of rise velocity for 3D rising bubble using a low surface tension coefficient.

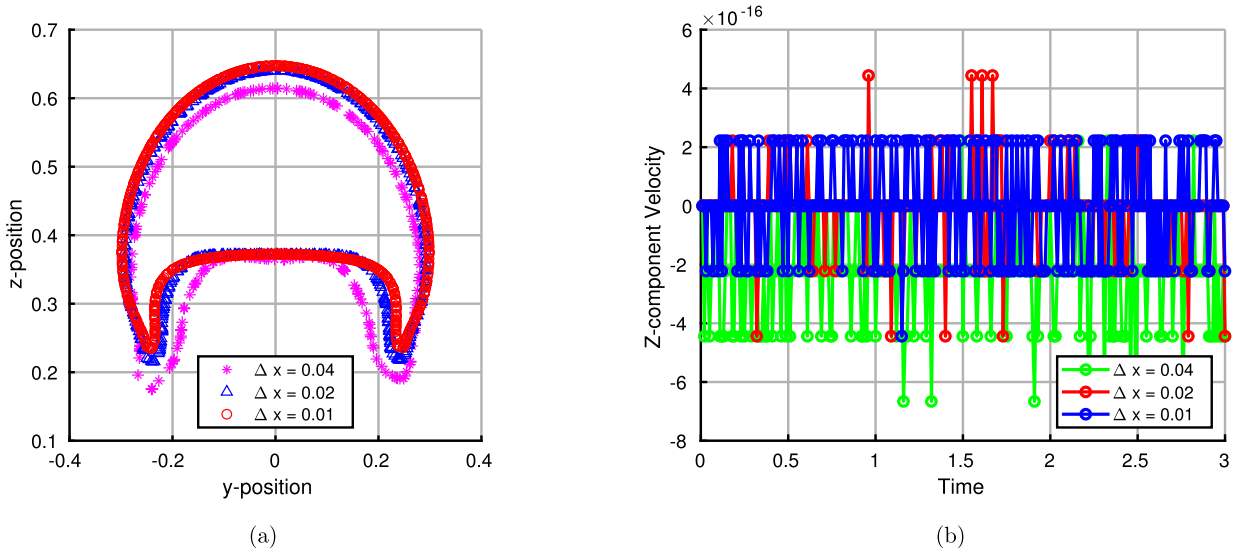


Fig. 16. (a) 2D Slice of the terminal shape for the rising bubble case with a low surface tension coefficient for various grid sizes (b) relative difference in bubble volume over time.

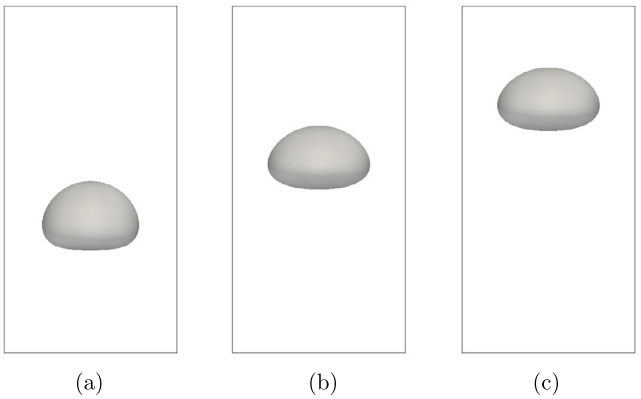


Fig. 17. Time evolution of the bubble for case 2 at time (a) $t = 1.0$ (b) $t = 2.0$ and (c) $t = 3.0$.

4.7.2. Case 2: high surface tension coefficient

The parameters used for the second test case are listed in Table 5. For this case, the density and viscosity ratio are both 10 and $Bo = 10$, thus surface tension plays a larger role in preserving the curvature of the interface. Fig. 17 shows the 3D contours of the $\phi = 0$ level set at different times. In contrast to the first case, we see in Fig. 17a that there is no serious deformation to the bottom of the bubble after $t = 1.0$. This is due to the fact that the surface tension forces are much stronger in this case, and counteract the flow-induced deformation. As the bubble rises even further, it takes the shape seen in Fig. 17b and maintains this shape (as seen in Fig. 17c) until the end of the simulation at $t = 3.0$.

Fig. 18 shows the rise velocity of the bubble vs. time for various grids; Fig. 18b shows a zoomed-in profile of the terminal rise velocities. As the mesh is refined the predicted terminal rise velocity converges to approximately $U = 0.35$.

In addition, Fig. 19 compares the predicted evolution of the rise velocity for the finest mesh with the reference rise velocity evolution from the work of Safi et al. [59] for case 2. Again, our predicted results are in good agreement with their reference rise velocity evolution. The terminal shape of the bubble as well as the difference in volume over time is shown in Fig. 20. Fig. 16a shows that terminal bubble shape converges as the mesh is refined. The relative difference in the volume of the bubble over time is plotted in Fig. 20b, and it can be seen that the volume enclosed by the interface is preserved up to 1×10^{-16} or on the order of machine accuracy.

4.8. Parallel scaling

We demonstrate the parallel scaling of our proposed formulation using Nalu by performing a strong parallel study. The test case used is the low surface tension rising bubble problem presented in Section 4.7.1. This test case uses up to 128 processors, with a total of 3.2 million cubic hexahedral elements. For this particular problem, Nalu solves a total of 6

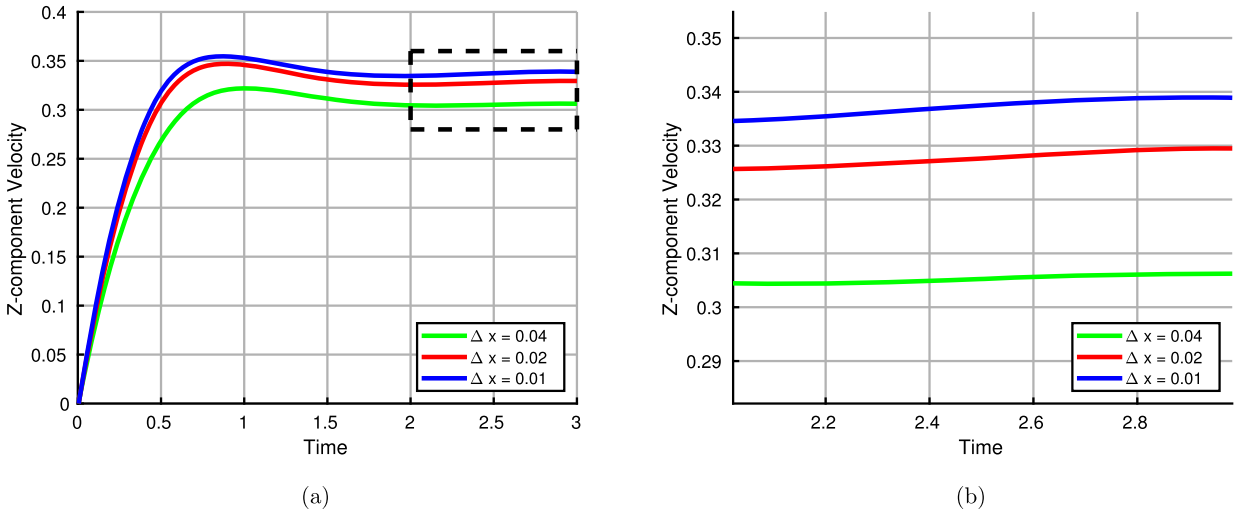


Fig. 18. (a) Time evolution for rise velocity for 3 different element sizes and (b) close up view of rise velocities.

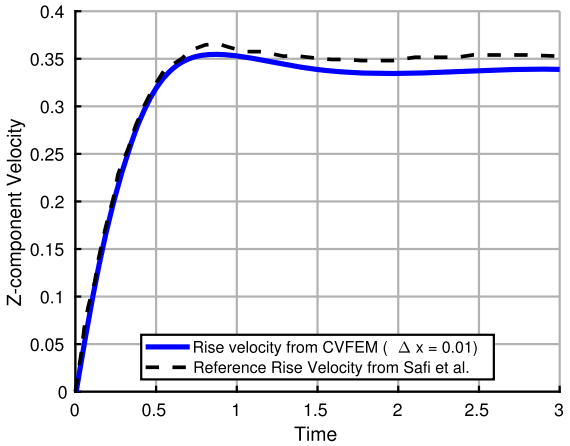


Fig. 19. Comparison between time evolution of rise velocity for 3D rising bubble using a high surface tension coefficient.

equations: level set advection, level set redistancing, the pressure Poisson equation, and coupled momentum system for three component directions. GMRES is used to solve each individual equation; the level set redistancing, advection and momentum equations use a symmetric Gauss–Seidel preconditioner whereas the pressure Poisson equation uses a multigrid preconditioner. The Trilinos linear solver package [60] is used for matrix preconditioning and solve operations. Fig. 21 shows a strong scaling study for assembly and solve times for the coupled system of equations, plotting the parallel speedup against MPI processor count. In all cases, speedup is computed as the factor by which computational time is reduced compared with the single-processor simulation. For each nonlinear step, “Assemble” measures the local matrix assembly time and “Solve” measures the linearized matrix solve time. Fig. 21a shows the total matrix assembly and solve times per nonlinear step, including the solve time for each individual equation system. Fig. 21b plots the solve time per linear iteration during the matrix solve.

As seen in Fig. 21a, the “Assemble” operations scale very well in parallel. Comparison for Figs. 21a and 21b reveals that some of the inefficiencies in the matrix solve parallelization are caused by an increase in the number of linear solver iterations needed on larger numbers of processors; this is due to sensitivity of convergence to the parallelization of the preconditioner. It has been shown by Lin et al. [61] that the base Nalu flow solver scales very well for large-scale problems, showing good weak and strong scaling properties for matrix solve and assembly times using up to 131,072 processors and 9 billion fluid elements. Our current study demonstrates that the additional solution steps (level set advection and redistancing) retain the scaling properties of the base code, and thus are expected to also scale well to massively parallel systems.

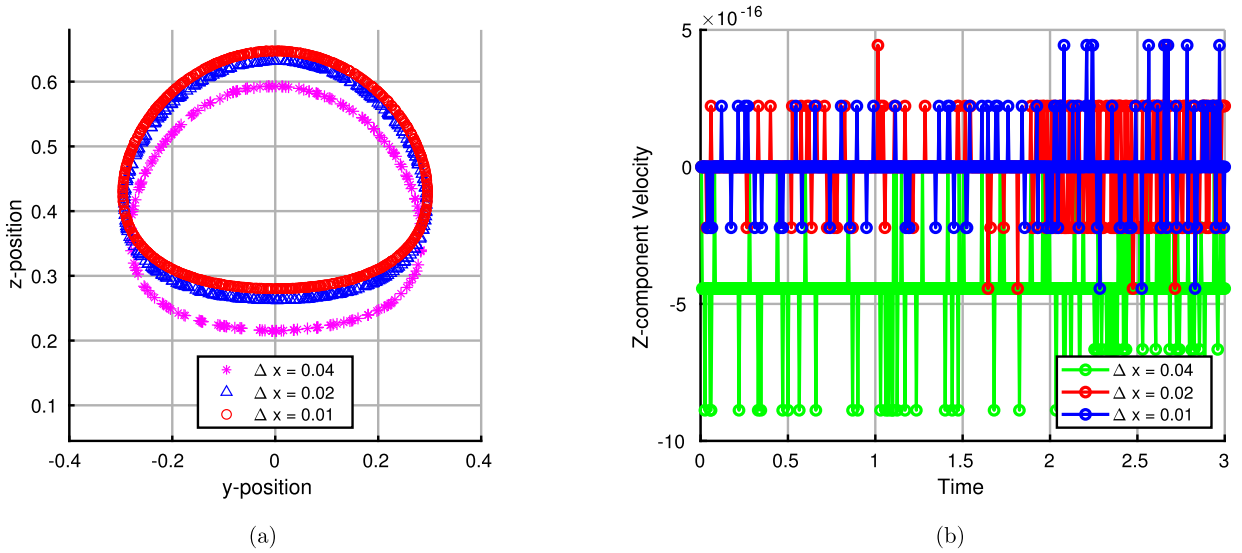


Fig. 20. (a) 2D Slice of the terminal shape for the rising bubble case with a low surface tension coefficient for various grid sizes (b) relative difference in bubble volume over time.

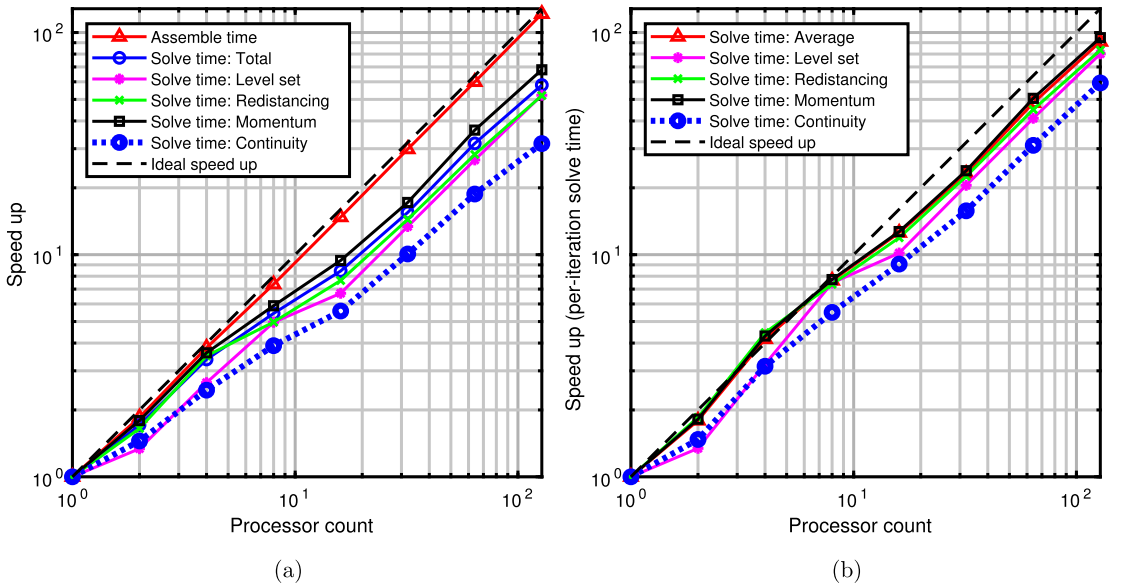


Fig. 21. Strong scaling study for rising bubble case for 3.2 million elements: (a) Assembly with total solve time. (b) Solve times per-iteration.

5. Conclusion

We have presented a novel balanced-force control volume finite element formulation for multiphase flows using a diffuse interface description of the level set method. This formulation is applicable to arbitrary collocated unstructured meshes and is suitable for large-scale parallel simulations for problems involving complex geometries. Moreover, our proposed method is able to handle multi-phase problems with high density and viscosity ratios.

By reformulating the pseudo-time dependent redistancing equation to a form suitable for CVFEM, we are able to take advantage of higher order upwinding schemes on the advection term; examples show that using higher order upwinding has a large effect on the quality of the redistanced level set field. This is demonstrated in Sections 4.1 and 4.3, as we see that second order upwinding techniques are necessary to provide an accurate solution. We have also developed a new formulation to constrain the volume fraction within each cell during redistancing, and enforced this through a Lagrange multiplier approach; Section 4.2 shows that this constraint combined with a global volume corrector is crucial to preserving the shape and volume of the level set profile during redistancing. An Enright deformation problem is solved in Section 4.3; again we see from these results that the second order upwinding scheme is crucial during the redistancing of the level set.

We test our proposed balanced force algorithm on an inviscid static drop using an unstructured mesh with density ratios as high as 10^{10} in Section 4.4, and demonstrate that, for a prescribed constant curvature, the method is able to exactly balance the pressure gradient and surface tension force and eliminate any parasitic velocities; this is not true when a non-balanced force algorithm is used. The coupled level set and momentum solver is shown to handle problems with high density and viscosity ratios in Section 4.7 and shows good agreement with predicted benchmark solutions from the literature. Finally, Section 4.6 illustrates that the predicted rise velocity from 2D rising bubble simulations converge at second order in the spatial grid size, while Section 4.8 demonstrates the parallel scalability of the level set advection solve and redistancing operations.

These presented numerical results demonstrate that our proposed algorithm is able to accurately solve complex problems on unstructured meshes within a control volume framework. In addition to discretely conserving volume, the method enables arbitrary refinements around regions exhibiting highly localized phenomena to achieve a more accurate representation of the underlying physics without sacrificing accuracy. Moreover, our proposed formulation is well-suited for problems that incorporate complex geometries where unstructured meshes are needed. Such a tool is important in engineering applications that involve a high resolution of the free-surface morphology and singular forces, such as surface tension, to drive complex flows. Such applications include but are not limited to metallic additive manufacturing and fuel atomization.

Acknowledgements

This work was supported in part by National Science Foundation (NSF) Cyber-Physical Systems (CPS) under Grant No. CPS/CMMI-1646592 and by the Center for Hierarchical Materials Design (CHiMaD) under grant No. 70NANB14H012. Stephen Lin is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1324585.

References

[1] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[2] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[3] T.J. Hughes, W.K. Liu, T.K. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Eng. 29 (1981) 329–349.
[4] J. Best, The formation of toroidal bubbles upon the collapse of transient cavities, J. Fluid Mech. 251 (1993) 79–107.
[5] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[6] M. van Sint Annaland, N. Deen, J. Kuipers, Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method, Chem. Eng. Sci. 60 (2005) 2999–3011.
[7] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[8] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, NY, 2002.
[9] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[10] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[11] X. Yang, A.J. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, J. Comput. Phys. 217 (2006) 364–394.
[12] Z. Wang, J. Yang, B. Koo, F. Stern, A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves, Int. J. Multiph. Flow 35 (2009) 227–246.
[13] J.M. Gimenez, N.M. Nigro, S.R. Idelsohn, E. Oñate, Surface tension problems solved with the particle finite element method using large time-steps, Comput. Fluids 141 (2016) 90–104.
[14] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (2008) 8395–8416.
[15] W. Yan, W. Ge, Y. Qian, S. Lin, B. Zhou, W.K. Liu, F. Lin, G.J. Wagner, Multi-physics modeling of single/multiple-track defect mechanisms in electron beam selective melting, Acta Mater. 134 (2017) 324–333.
[16] C. Panwisawas, C. Qiu, M.J. Anderson, Y. Sovani, R.P. Turner, M.M. Attallah, J.W. Brooks, H.C. Basoalto, Mesoscale modelling of selective laser melting: thermal fluid dynamics and microstructural evolution, Comput. Mater. Sci. 126 (2017) 479–490.
[17] S.A. Khairallah, A. Anderson, Mesoscopic simulation model of selective laser melting of stainless steel powder, J. Mater. Process. Technol. 214 (2014) 2627–2636.
[18] S.A. Khairallah, A.T. Anderson, A. Rubenchik, W.E. King, Laser powder-bed fusion additive manufacturing: physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones, Acta Mater. 108 (2016) 36–45.
[19] Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, Novel structural modeling and mesh moving techniques for advanced fluid–structure interaction simulation of wind turbines, Int. J. Numer. Methods Eng. 102 (2015) 766–783.
[20] J. Yan, A. Korobenko, X. Deng, Y. Bazilevs, Computational free-surface fluid–structure interaction with application to floating offshore wind turbines, Comput. Fluids 141 (2016) 155–174.
[21] E. Rouy, A. Tourin, A viscosity solutions approach to shape-from-shading, SIAM J. Numer. Anal. 29 (1992) 867–884.
[22] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. 93 (1996) 1591–1595.
[23] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, J. Comput. Phys. 148 (1999) 2–22.
[24] J. Yang, F. Stern, A highly scalable massively parallel fast marching method for the Eikonal equation, J. Comput. Phys. 332 (2017) 333–362.
[25] H. Zhao, A fast sweeping method for Eikonal equations, Math. Comput. 74 (2005) 603–627.
[26] M. Detrixhe, F. Gibou, C. Min, A parallel fast sweeping method for the Eikonal equation, J. Comput. Phys. 237 (2013) 46–55.
[27] M. Detrixhe, F. Gibou, Hybrid massively parallel fast sweeping method for static Hamilton–Jacobi equations, J. Comput. Phys. 322 (2016) 199–223.
[28] M. Sussman, E. Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM J. Sci. Comput. 20 (1999) 1165–1191.
[29] S. Nagrath, K.E. Jansen, R.T. Lahey, Computation of incompressible bubble dynamics with a stabilized finite element level set method, Comput. Methods Appl. Mech. Eng. 194 (2005) 4565–4587.

[30] A. Smolianski, Numerical Modeling of Two-Fluid Interfacial Flows, Ph.D. thesis, University of Jyväskylä, 2001.
[31] C.E. Kees, I. Akkerman, M.W. Farthing, Y. Bazilevs, A conservative level set method suitable for variable-order approximations and unstructured meshes, J. Comput. Phys. 230 (2011) 4536–4558.
[32] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (2005) 225–246.
[33] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow ii, J. Comput. Phys. 225 (2007) 785–807.
[34] J. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[35] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[36] D. Torres, J. Brackbill, The point-set method: front-tracking without connectivity, J. Comput. Phys. 165 (2000) 620–644.
[37] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comput. Phys. 213 (2006) 141–173.
[38] F. Denner, B.G. van Wachem, Fully-coupled balanced-force VOF framework for arbitrary meshes with least-squares curvature evaluation from volume fractions, Numer. Heat Transf., B Fundam. 65 (2014) 218–255.
[39] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, J. Comput. Phys. 227 (2008) 2674–2706.
[40] H. Montazeri, C. Ward, A balanced-force algorithm for two-phase flows, J. Comput. Phys. 257 (2014) 645–669.
[41] N. Balcázar, L. Jofre, O. Lehmkuhl, J. Castro, J. Rigola, A finite-volume/level-set method for simulating two-phase flows on unstructured grids, Int. J. Multiph. Flow 64 (2014) 55–72.
[42] N. Balcázar, A. Oliva, J. Rigola, A level-set method for thermal motion of bubbles and droplets, J. Phys. Conf. Ser. 745 (2016) 032113.
[43] M. Owkes, O. Desjardins, A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows, J. Comput. Phys. 249 (2013) 275–302.
[44] J.O. McCaslin, O. Desjardins, A localized re-initialization equation for the conservative level set method, J. Comput. Phys. 262 (2014) 408–426.
[45] B. Baliga, S. Patankar, A control volume finite-element method for two-dimensional fluid flow and heat transfer, Numer. Heat Transf. 6 (1983) 245–261.
[46] G. Schneider, M. Zedan, Control-volume-based finite element formulation of the heat conduction equation, spacecraft thermal control, design, and operation, Prog. Astronaut. Aeronaut. 86 (1983) 305–327.
[47] S. Domino, Sierra Low Mach Module: Nalu Theory Manual 1.0, SAND2015-3107W, Sandia National Laboratories Unclassified Unlimited Release (UUR), 2015, https://github.com/spdomin/NaluDoc.
[48] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[49] A.J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, Math. Comput. 23 (1969) 341–353.
[50] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.
[51] C. Rhie, W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[52] F.N. Felten, T.S. Lund, Kinetic energy conservation issues associated with the collocated mesh scheme for incompressible flow, J. Comput. Phys. 215 (2006) 465–484.
[53] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.
[54] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (2002) 83–116.
[55] Z. Wang, J. Yang, F. Stern, A new volume-of-fluid method with a constructed distance function on general structured grids, J. Comput. Phys. 231 (2012) 3703–3722.
[56] Y. Zhao, H.-C. Chen, A new coupled level set and volume-of-fluid method to capture free surface on an overset grid system, Int. J. Multiph. Flow 90 (2017) 144–155.
[57] M. Williams, D. Kothe, E. Puckett, Accuracy and convergence of continuum surface tension models, in: Fluid Dynamics at Interfaces, Cambridge University Press, Cambridge, 1998, pp. 294–305.
[58] S.-R. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, Int. J. Numer. Methods Fluids 60 (2009) 1259–1288.
[59] M.A. Safi, N. Prasianakis, S. Turek, Benchmark computations for 3d two-phase flows: a coupled lattice Boltzmann-level set study, Comput. Math. Appl. 73 (2017) 520–536.
[60] M. Heroux, R. Bartlett, V.H.R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, An Overview of Trilinos, Technical Report SAND2003-2927, Sandia National Laboratories, 2003.
[61] P. Lin, M. Bettencourt, S. Domino, T. Fisher, M. Hoemmen, J. Hu, E. Phipps, A. Prokopenko, S. Rajamanickam, C. Siefert, et al., Towards extreme-scale simulations with next-generation Trilinos: a low Mach fluid application case study, in: IEEE International Parallel & Distributed Processing Symposium Workshops (IPDPSW), IEEE, 2014, pp. 1485–1494.