A conservative level-set method based on a posterior mass correction preserving distance property for incompressible multiphase flows simulations

Tian Long, Jinsheng Cai, Shucheng Pan

School of Aeronautics, Northwestern Polytechnical University, Xi’an, 710072, PR China

Abstract

As one of the most popular interface-capturing methods, the level-set method is inherently non-conservative, and its evolution usually leads to unphysical mass gain/loss. In this paper, a novel conservative level set method is developed for incompressible multiphase flows simulations. A posterior mass correction is performed by introducing a small perturbation to the level-set field, which is solved via the Newton method. Unlike in previous researches, the signed distance property of the level-set function is exactly preserved after the present mass correction. Moreover, this method can be easily generalized from 2D to 3D. The influence for the computational efficiency is slight as the correction does not need to be applied at every time step. Various benchmark cases involving pure interface-evolution problems and multiphase flows problems are considered to validate the present method. For all cases, the accuracy and efficiency of the original method and the present method are quantitatively compared. It is observed that, with negligibly extra cost, the conservation error is reduced to the order of machine accuracy by the present method, which indicates its potential applications in complex multiphase flows simulations.

Keywords: Level-set, Posterior mass correction, Incompressible, Multiphase flows

1. Introduction

Over the last decades, many numerical methods have been developed for the simulation of multiphase flows due to its widely practical applications, such as ink jet printing [1], bubble column reactors [2], gas turbine engines [3], etc. In general, these methods can be divided into two classes based on the treatment of the material interface separating different fluids, namely interface tracking methods and interface capturing methods. In the interface tracking methods,
the interface is explicitly tracked by various markers, e.g., the mesh used in the arbitrary Lagrangian Eulerian (ALE) method [4, 5], which evolves and deforms with background flows, and the Lagrangian particles used in the front-tracking method [6]. Through the explicit interface representations, the multiphase flows can be simulated with very high accuracy. Nevertheless, these methods are not suitable for complex problems as the typological changes have to be handled manually, which is challenging and computationally expensive. In contrast, the interface capturing methods, in which the interface is implicitly captured through an auxiliary function, can cope with the interface deformations and topological changes in an automatical way. One such method is the volume of fluid (VOF) method [7–11], which describes the interface through the color function representing the fraction of the liquid volume in each computational cell. Although it is intrinsically mass-conserving, this method suffers from the inaccurate computation of geometrical variables such as normals and curvature, owing to the discontinuous feature of the color function. Another popular interface capturing method is the level-set method proposed by Osher and Sethian [12], which represents the interface with the zero contour of the so-called level-set function defined by the signed distance to the interface. The interface evolution and the related geometrical variables can be computed accurately since the level-set function is Lipschitz-continuous. In recent years, the level-set method is increasingly used in the simulation of multiphase flows [13] due to its simplicity and high efficiency. However, unphysical mass gain/loss will occur when it is applied to incompressible multiphase flows. The main reasons for the missing conservation property are twofold. First, the level-set filed will be evolved via solving an advection equation, whose discretization will unavoidably lead to numerical errors. Second, additional conservation errors will be introduced because of the artificial alteration of the interface in the reinitialization procedure [14], which is needed to maintain the signed distance property of the level-set function.

To achieve mass conservation in the level-set method, many numerical approaches have been proposed, which can be classified into the following five categories. The first strategy is the most straightforward one, i.e., to reduce numerical errors via using high-order discretization schemes [15, 16], minimizing the displacement of the zero level-set during reinitialization [17, 18], or employing a local mesh refinement [19–21]. Nevertheless, these methods can only mitigate (but not eliminate) the conservation errors. The second approach is to modify the definition of the level-set function to reduce the overall mass loss. One such example is the hyperbolic-tangent level-set method [22, 23], which, however, shows a tendency to generate unphysical pieces of fluids called flotsams/jetsams breaking off where the mesh is under-resolved [24]. Thirdly, to improve the mass conservation, some methods have coupled the level-set method with other interface descriptions, e.g., the coupled level-set/volume of fluid method (CLSVOF) [25, 26], the hybrid particle level-set method (HPLS) [27, 28], and the level-set volume constraint method (HLSVC) [29]. Although these hybrid methods show a remarkable improvement on the mass conservation, the simplicity of the original level-set method is also lost. The fourth remedy is enforcing a volume
constraint during the evolution of the interface. For instance, in the method of Yuan et al. [30], the volume constraint is added as a source term in the advection equation. However, they assumed a uniform distribution of the source term in the whole domain, which lacks physical validity. The last approach is the posterior mass correction, in which the absolute mass loss within the computational domain will be redistributed. Examples of such methods include the mass-conserving level-set method (MCLS) [31, 32], the curvature-based mass-redistribution method [33] and the interface-correction level-set method (ICLS) [34]. Since these methods cannot preserve the signed distance property of the level-set function, a reinitialization step is needed after the correction, in which new conservation errors will arise.

In this paper, a novel conservative level-set method for incompressible multiphase flows simulations is proposed. Inspired by the stimulus-response algorithm of Han et al. [35], we have developed a novel posterior mass correction procedure which does not violate the signed distance property of the level-set function. As the correction step is not implemented at every time step, the extra computational cost is negligible. We also emphasize that the present method is strictly conservative, i.e., the conservation error is of the order of machine accuracy. The remainder of this paper is organized as follows. In Section 2, we briefly review the numerical scheme for incompressible multiphase flows with the traditional level-set method. Subsequently in Section 3 we introduce the novel conservative level-set method. A number of numerical tests are carried out to validate the present method in Section 4, followed by the concluding remarks in Section 5.

2. Numerical method

2.1. Governing equations

The non-dimensional governing equations for incompressible multiphase flows are

\[ \nabla \cdot \mathbf{V} = 0, \quad (1a) \]

\[ \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} = \frac{1}{\rho} \left( -\nabla p + \frac{1}{Re} \nabla \cdot (\mu (\nabla \mathbf{V} + (\nabla \mathbf{V})^T)) \right) + \frac{1}{Fr} \mathbf{g}, \quad (1b) \]

where \( \mathbf{V} = (u, v, w) \) is the velocity vector, \( \rho \) the density, \( p \) the pressure, \( \mu \) the dynamic viscosity, and \( \mathbf{g} \) the unit vector aligned with gravity. For viscous flows considered here, the velocity field and its tangential derivatives are continuous across the interface \( \Gamma \) that separates two different fluids (denoted as fluid1 and fluid2). In contrast, the material properties are subject to a jump across the interface, i.e., \( [\rho]_\Gamma = \rho_2 - \rho_1 \) and \( [\mu]_\Gamma = \mu_2 - \mu_1 \), where \([\cdot]_\Gamma\) represents a jump in the variable considered. The pressure jump caused by the viscosity discontinuity and surface tension reads

\[ [p]_\Gamma = \frac{1}{We} \kappa + \frac{2}{Re} [\mu]_\Gamma \mathbf{n}^T \cdot \nabla \mathbf{V} \cdot \mathbf{n}, \quad (2) \]
in which \( \mathbf{n} = (n_x, n_y, n_z) \) is the interface normal, and \( \kappa \) is the curvature. The dimensionless parameters in Eqs. (1) and (2), namely the Reynolds number \( Re \), the Froude number \( Fr \), and the Weber number \( We \), are defined as

\[
Re = \frac{\bar{\rho} \bar{U} \bar{L}}{\mu}, \quad Fr = \frac{\bar{U}^2}{g \bar{L}}, \quad We = \frac{\bar{\rho} \bar{U}^2 \bar{L}}{\bar{\sigma}},
\]

where \( \bar{\rho}, \bar{U}, \bar{L}, \bar{\mu}, \bar{g} \) and \( \bar{\sigma} \) represent the reference density, velocity, length, dynamic viscosity, gravitational acceleration, and surface tension coefficient, respectively. Note that the surface tension coefficient \( \bar{\sigma} \) is assumed to be constant over space and time in this paper.

2.2. Interface representation

In a two fluid system, the flow domain \( \Omega \) is decomposed into two sub-domains \( \Omega_1 \) and \( \Omega_2 \) by an evolving interface \( \Gamma(t) \), which is implicitly defined by the zero contour of the level-set function \( \phi(x,t) \) representing the signed distance from the interface to \( x \). Without loss of generality, let \( \phi > 0 \) in \( \Omega_1 \) (fluid1) and \( \phi < 0 \) in \( \Omega_2 \) (fluid2). The interface movement is captured via solving the level-set equation [36],

\[
\frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0,
\]

where \( \mathbf{V} \) is the background fluid velocity. With a given level-set function, the normal vector \( \mathbf{n} \) and curvature \( \kappa \) can be easily obtained by

\[
\mathbf{n} = \nabla \phi, \quad \text{and} \quad \kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|},
\]

respectively. In practice, a reinitialization procedure is required to enforce the signed distance property \(|\nabla \phi| = 1|\) which will be violated due to the deformations of the interface. Following Sussman et al. [37], the level-set function is reinitialized via iteratively solving a Hamilton-Jacobi equation,

\[
\frac{\partial \phi}{\partial \tau} + S(\phi_0)(|\nabla \phi| - 1) = 0,
\]

where \( \tau \) is the pseudo time, and \( S(\phi_0) \) is the sign function of the original level-set. Note that there is no need to implement the reinitialization at every time step [33]. Following Ref. [34], in the present study, we perform 10 iterations of Eq. (6) for every 20 time steps.

2.3. Numerical scheme

The solver we used employs a staggered uniform grid, in which the pressure \( p \) and the level-set function \( \phi \) are stored at cell centers while the velocity is stored at cell faces. For the solution of the Navier-Stokes equations, spatial discretization is performed by using the second-order accurate finite central difference scheme, and the projection method of Chorin [38] is used for time marching.
Before updating the flow field, $\phi^{n+1}$ is obtained through solving the level-set equation Eq. (4). The fifth-order accurate weighted essentially non-oscillatory (WENO5) scheme [39] and the third-order accurate strong stability preserving (SSP) Runge-Kutta (RK3) scheme [40] are used for the evaluation of $\nabla \phi$ and the temporal discretization, respectively. With $\phi^{n+1}$, the density and velocity fields can be updated by

$$
\rho^{n+1} = \rho_1 H_s(\phi^{n+1}) + \rho_2 (1 - H_s(\phi^{n+1})),
$$

$$
\mu^{n+1} = \mu_1 H_s(\phi^{n+1}) + \mu_2 (1 - H_s(\phi^{n+1})),
$$

where

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

is the smoothed Heaviside function with $\epsilon = 1.5\Delta x$ being the half thickness of the smeared interface.

Let $RU^n$ denotes the right-hand side of the momentum equation Eq. (1b) with the pressure gradient term excluded, which reads

$$
RU^n = -V^n \cdot \nabla V^n + \frac{1}{\rho^{n+1}} \left[ \frac{1}{Re} \nabla \cdot (\mu^{n+1}(\nabla V^n + (\nabla V^n)^T)) \right] + \frac{1}{Fr} g. \quad (9)
$$

Then, by using the second-order accurate Adams-Bashforth scheme (AB2) for the temporal integration, the intermediate velocity $V^*$ can be computed by

$$
V^* = V + \Delta t \left( \frac{3}{2} RU^n - \frac{1}{2} RU^{n-1} \right). \quad (10)
$$

Finally the divergence-free velocity $V^{n+1}$ becomes

$$
V^{n+1} = V^* - \frac{\Delta t}{\rho^{n+1}} \nabla p^{n+1}, \quad (11)
$$

where the pressure field $p^{n+1}$ is obtained by solving a Poisson equation,

$$
\nabla \cdot \left( \frac{1}{\rho^{n+1}} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot V^*. \quad (12)
$$

When discretizing the pressure gradient term in Eq. (12), we treat the pressure jump Eq. (2) via the ghost fluid method (GFM) [34]. Furthermore, the fast pressure-correction method [41] is utilized to improve computational efficiency.

3. Conservative level-set method

When applied to the simulations of multiphase flows, the classic level-set method inherently leads to the non-conservation of mass. The volume enclosed
by the zero level-set will change over time, which implies the unphysical mass gain/loss in the context of incompressible flows. In this section, a novel conservative level-set method is developed to address this issue. We begin by analyzing the sources of volume gain/loss in the original level-set method. Let \( V, V_1 \) and \( V_2 \) represent the volume of \( \Omega, \Omega_1 \) and \( \Omega_2 \), respectively. The conservation of \( V_2 \) is chosen to describe the present method, which can be numerically computed by

\[
V_2 = \int_{\Omega} (1 - H_s(\phi))d\Omega. \tag{13}
\]

Taking the total derivative of Eq. (13) with respect to time \( t \), we obtain

\[
\frac{DV_2}{Dt} = \int_{\Omega} (1 - H_s(\phi))\nabla \cdot V d\Omega - \int_{\Omega} \delta_s(\phi)(\frac{\partial \phi}{\partial t} + V \cdot \nabla \phi)d\Omega \tag{14}
\]

where

\[
\delta_s(\phi) = H'_s(\phi) = \begin{cases} \frac{1}{2\epsilon} \left[ 1 + \cos(\frac{\pi \phi}{\epsilon}) \right] & \text{if } |\phi| \leq \epsilon \\ 0 & \text{elsewise} \end{cases} \tag{15}
\]

is the smoothed Dirac delta function. It can be observed from Eq. (14) that the change rate of \( V_2 \) will be zero when the continuity equation Eq. (1a) and the level-set equation Eq. (4) are solved exactly. However, even with a high-order discretization scheme and a fine mesh, numerical errors are inevitably introduced, which results in the non-conservation \( \delta V_2^{(d)} \). The additional conservation error \( \delta V_2^{(r)} \) will arise from the reinitialization procedure as the zero level-set can be altered due to numerical artifacts. By employing the truly upwind discretization [18] or the subcell reconstruction [42] near the interface, \( \delta V_2^{(r)} \) can be mitigated to some extent, but not eliminated. For the previous conservative level-set methods employing the posterior mass correction, \( \delta V_2^{(r)} \) is difficult to handle since they cannot preserve the signed distance property of \( \phi \). Even though the volume is conserved after being remedied, new conservation error will be introduced in the consequent reinitialization process, see Refs. [33, 34].

In the scale separation algorithm of Han et al. [35], a uniform positive or negative shift referred to as stimulus is applied for the level-set function to determine and separate non-resolvable interface structures. Inspired by this idea, a small perturbation \( \varepsilon \) is used in this study to eliminate the total conservation error \( \delta V = \delta V_2^{(d)} + \delta V_2^{(r)} \). We also emphasize that, as \( |\nabla (\phi + \varepsilon)| = |\nabla \phi| \) holds, this treatment does not violate the signed distance property. Let \( V_2^0 \) be the initial volume of \( \Omega_2 \), thus the target equation reads

\[
\int_{\Omega} (1 - H_s(\phi + \varepsilon))d\Omega = V_2 + \delta V = V_2^0. \tag{16}
\]

This equation can be solved by the Newton’s method. First, we define

\[
f(\varepsilon) = \int_{\Omega} (1 - H_s(\phi + \varepsilon))d\Omega - V_2^0 = 0, \tag{17}
\]
and the formulas for the iteration procedure are given by

\[ \varepsilon_{k+1} = \varepsilon_k - \frac{f(\varepsilon_k)}{f'(\varepsilon_k)} \quad (18a) \]

\[ \varepsilon_0 = \frac{\int_{\Omega} (1 - H_s(\phi)) d\Omega - V_2^0}{\int_{\Omega} \delta_s(\phi) d\Omega} \quad (18b) \]

\[ f'(\varepsilon) = -\int_{\Omega} \delta_s(\phi + \varepsilon) d\Omega \quad (18c) \]

where \( k \) indexes the iteration step. The relative volume error

\[ E_r = \left| \frac{V_2 - V_2^0}{V_2^0} \right| \quad (19) \]

is used to control the iteration process and the termination happens when \( E_r \leq E_t \). In this paper, the threshold \( E_t \) is chosen to \( 1.0 \times 10^{-12} \), and 3-4 iterations are required for all cases we have computed. When considering boundary terms, e.g., fluid2 is the liquid phase flowing into the domain across an inlet boundary, we can perform the correction procedure in the same way by changing the right side of Eq. (16) to \( V_2^0 + U_i A_i t^n \), where \( U_i, A_i, \) and \( t^n \) are the inflow velocity, the area of the inlet, and the current simulation time, respectively.

To elucidate the present method from a geometrical point of view, Eq. (18b) is reformulated as

\[ \varepsilon_0 \int_{\Omega} \delta_s(\phi) d\Omega = -\delta V. \quad (20) \]

When \( \varepsilon_0 \) is small, we have

\[ H(\phi + \varepsilon_0) - H(\phi) \approx \delta_s(\phi) \varepsilon_0, \quad (21) \]

which indicates that the left side of Eq. (20) is the approximation of the volume enclosed by the iso-surfaces \( \phi = 0 \) and \( \phi = -\varepsilon_0 \). The iteration process can be considered as an adjustment to reimburse the approximate error. To improve numerical accuracy, in the present study, the volume integrals in Eqs. (16) - (19) are normalized by \( V \). Moreover, implementing the correction procedure at every time step is superfluous. Additional computational efficiency can be achieved by choosing an appropriate correction frequency, which is discussed in Section 4.1.1.

4. Numerical results

In this section, the proposed conservative level-set method is validated by both the pure interface-evolution problems and the multiphase flows problems. For all cases, the accuracy and efficiency of the present method are quantitatively compared with that of the original level-set method.
Table. 1. Linear advection: the CPU costs of the original method and the present method for different resolutions. For all cases, the simulations are performed on a server with an AMD EPYC 7742 64-core Processor (2.3 GHz).

| Resolution | Without correction | with correction (every 20 time steps) | with correction (every time step) |
|------------|--------------------|--------------------------------------|----------------------------------|
| 16 × 16 (1 core) | 0.1560 | 0.2043 | 0.2424 |
| 32 × 32 (4 cores) | 1.0682 | 1.1136 | 1.3068 |
| 64 × 64 (4 cores) | 6.1711 | 6.2657 | 6.6411 |

4.1. Pure interface-evolution problems

4.1.1. Linear advection

Firstly, as in [31], the linear advection case is considered to demonstrate the conservation. Driven by a constant velocity $u = 1$, a circle of radius $R = 0.1$ is initially placed at the center of a $[0,1] \times [0,1]$ square computational domain with periodic boundary conditions. This domain is discretized by a Cartesian grid, with resolutions increasing from $16 \times 16$ to $64 \times 64$. This circle is advected for 40 periods with the time step $\Delta t$ of $0.8\Delta x$. To investigate the influence of the correction frequency, the correction procedure is applied for every time step and every 20 time steps. As shown in Table. 1, the extra computational cost induced by the present method can be negligible. On the coarsest mesh ($16 \times 16$), the interface indeed disappears due to the numerical dissipation of the original method, while it is preserved well with the present method, see Fig. 1. In addition, the correction frequency does not affect the results. Thus, in the remaining cases, the correction procedure will be performed every 20 time steps for the sake of efficiency. Moreover, for different resolutions, the relative volume error is plotted versus the simulation time in a semi-logarithmic scale. It can be observed that the conservation error is significantly reduced with the present method. Note that the relative volume error obtained will be exactly zero at several time instants, which is set to $10^{-15}$ to be displayed in the logarithmic scale.

4.1.2. Zalesak’s disk - rigid body rotation of a slotted disk

The second benchmark case is the rigid body rotation of the Zalesak’s disk [43], which has been extensively utilized to validate the interface-advection algorithms [33, 34, 44]. Considering a unit square computational domain, with a radius of 0.15, a slot length of 0.25 and a slot width of 0.05, the slotted circle centered at $(0.5, 0.75)$ is advected in a rotating velocity field given by

\[
\begin{align*}
u & = -2\pi(y - 0.5) \\
v & = 2\pi(x - 0.5).
\end{align*}
\]

With $\Delta t = \Delta x/8$, the simulations are carried out on the grid with resolutions increasing from $50 \times 50$ to $200 \times 200$, see Table. 2. The termination time is $t = 1$, after which the disk has recovered to its original position. In Fig. 2(a)-(c), the final interface profiles obtained by the original method and the present method for different resolutions are compared, which indicate that the results of the present method are qualitatively better. As shown in Fig. 2(d), with
Fig. 1. Linear advection: The interfaces at different time instants (the 1st and 2nd columns) and the relative volume errors (the 3rd column) for different resolutions.
the mass correction procedure, the relative volume errors exhibit an order of machine accuracy for all resolutions.

\[ \text{Relative volume error} \]

\[ 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \]

\[ 10^{-17} \quad 10^{-15} \quad 10^{-13} \quad 10^{-11} \quad 10^{-9} \quad 10^{-7} \quad 10^{-5} \quad 10^{-3} \quad 10^{-1} \]

\[ \text{without correction (50)} \quad \text{with correction (50)} \]
\[ \text{without correction (100)} \quad \text{with correction (100)} \]
\[ \text{without correction (200)} \quad \text{with correction (200)} \]

\[ \text{initial interface without correction} \quad \text{with correction} \]

\[ 200 \times 200 \]

\[ \begin{align*}
    u &= -2 \sin(\pi x)^2 \sin(\pi y) \cos(\pi y) \cos(\pi t/T), \\
    v &= 2 \sin(\pi y)^2 \sin(\pi x) \cos(\pi x) \cos(\pi t/T),
\end{align*} \tag{23} \]

where \( T = 8 \) is the period of a reversing vortex flow. Deforming with the background velocity, the circle will be stretched into a thin filament at first.

\[ \text{Fig. 2. Zalesak's disk: (a)-(c) the interfaces after one revolution for different resolutions, and (d) the relative volume errors for different resolutions.} \]

4.1.3. Deformed interface in a vortex flow

Proposed by Bell et al. [45], the 2D reversed single vortex case is considered to assess the ability of the present method for capturing thin filaments. The computational domain is a unit square with a circle of radius \( R = 0.15 \), which is initially located at \((0.5, 0.75)\) and deforms in a vortex flow. As in Refs. [33, 34], the velocity field is given by

\[ \begin{align*}
    u &= -2 \sin(\pi x)^2 \sin(\pi y) \cos(\pi y) \cos(\pi t/T), \\
    v &= 2 \sin(\pi y)^2 \sin(\pi x) \cos(\pi x) \cos(\pi t/T),
\end{align*} \tag{23} \]

where \( T = 8 \) is the period of a reversing vortex flow. Deforming with the background velocity, the circle will be stretched into a thin filament at first.
Table 2. Zalesak’s disk: the CPU costs of the original method and the present method for different resolutions.

| Resolution       | CPU costs (s) | without correction | with correction |
|------------------|---------------|---------------------|-----------------|
| 50 × 50 (4 cores) | 0.6640        | 0.6818              |
| 100 × 100 (16 cores) | 4.3795        | 4.4153              |
| 200 × 200 (16 cores) | 17.271        | 17.407              |

Table 3. Deformed interface in a vortex flow: the CPU costs of the original method and the present method for different resolutions.

| Resolution       | CPU costs (s) | without correction | with correction |
|------------------|---------------|---------------------|-----------------|
| 2D               |               |                     |
| 64 × 64 (4 cores) | 3.0355        | 3.8993              |
| 128 × 128 (4 cores) | 23.822        | 24.786              |
| 256 × 256 (16 cores) | 96.666        | 99.107              |
| 3D               |               |                     |
| 64 × 64 × 64 (4 cores) | 15.234        | 15.787              |
| 128 × 128 × 128 (32 cores) | 56.776        | 57.918              |
| 256 × 256 × 256 (64 cores) | 767.64        | 775.62              |

The largest deformation happens at $t = T/2$, and then the filament is pulled in reverse so that the initial circle is recovered at $t = T$. With $\Delta t = 0.32\Delta x$, the simulations are conducted on the uniform grid with resolutions increasing from 64 × 64 to 256 × 256, see Table 3. The interfaces at $t = T/2$ and $t = T$ obtained by the original method and the present method are compared in Fig. 3. On the coarsest mesh of 64 × 64, the interface obtained with the original method is heavily dissipated, which totally vanishes at $t = T$. In contrast, with the same resolution, satisfying results are obtained by employing the present method. For all resolutions, the thin filament is better captured by the present method. The conservation is confirmed by the time history of the relative volume error, see the third column in Fig. 3.

The 3D version of this case is considered on the grid with increasing resolutions. Following LeVeque et al. [46], a sphere of radius $R = 0.15$ is placed at (0.35, 0.35, 0.35) in a unit cubic domain with the velocity field given by

$$
\begin{align*}
u &= 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos(\pi t/T), \\
v &= -\sin^2(\pi y) \sin(2\pi x) \sin(2\pi z) \cos(\pi t/T), \\
w &= -\sin^2(\pi z) \sin(2\pi x) \sin(2\pi y) \cos(\pi t/T),
\end{align*}
$$

where $T$ is set to 3. The time step is chosen as $\Delta t = \Delta x/4$. It can be observed from Table 3 that the CPU cost of the present method is almost the same as that of the original method. Note that the conservation error is greatly reduced with the present method, see Fig. 4.

4.2. Multiphase flows problems

4.2.1. Falling droplet

We consider a droplet falling in air under gravity effect [47, 48] here to validate the present method for multiphase flows problems. Within a rectangular
Fig. 3. Deformed interface in a vortex flow (2D): The interfaces at different time instants (the 1st and 2nd columns) and the relative volume errors (the 3rd column) for different resolutions.
Fig. 4. Deformed interface in a vortex flow (3D): The interfaces at different time instants (the 1st and 2nd columns) and the relative volume errors (the 3rd column) for different resolutions.
domain of $[0, 5] \times [0, 15]$, a circular liquid drop with a diameter of $D = 1$ is initially placed at $(2.5, 12.5)$ and then falls due to the gravity in the y-direction (see the coordinate axis in Fig. 5(a)). This case can be characterized by the density ratio $\bar{\rho} = \tilde{\rho}_l / \tilde{\rho}_a$, the Bond number $Bo = \bar{g}(\tilde{\rho}_l - \tilde{\rho}_a)\bar{D}^2 / \bar{\sigma}$, and the gravity Reynolds numbers of liquid and air,

\begin{align}
Re_l &= \sqrt{\bar{g}\tilde{\rho}_l(\tilde{\rho}_l - \tilde{\rho}_a)\bar{D}^3 / \tilde{\mu}_l}, \\
Re_a &= \sqrt{\bar{g}\tilde{\rho}_a(\tilde{\rho}_l - \tilde{\rho}_a)\bar{D}^3 / \tilde{\mu}_a}.
\end{align}

Following Refs. [47, 48], we set these characteristic parameters as $\bar{\rho} = 10$, $Bo = 100$, $Re_l = 20$ and $Re_a = 10$, which yields the following initial conditions,

\begin{align}
\rho &= 1, \quad u = 0, \quad v = 0, \quad p = 1, \quad \mu = 1 \quad \text{air}, \\
\rho &= 10, \quad u = 0, \quad v = 0, \quad p = 1, \quad \mu = 2\sqrt{10} \quad \text{liquid drop}, \\
\phi &= -0.5 + \sqrt{(x - 2.5)^2 + (y - 12.5)^2} \quad \text{level set}, \\
Re &= 6.635, \quad We = 10, \quad Fr = 1 \quad \text{dimensionless parameters}.
\end{align}

For this case, the termination time is $t = 9.0$, with the time step being $\Delta t = 5.0 \times 10^{-4}$. The computational domain is discretized on a grid involving $120 \times 360$ cells. The no-slip wall boundary condition is employed for the top and bottom sides while the free-slip wall boundary condition is applied for the left and right sides. As shown in Fig. 5(a), expanding along the x-direction, the droplet gradually deforms into a bag shape similar with that in Fig. 9 of Ref. [47] and Fig. 21 of Ref. [48], and undergoes the bag breakup [49, 50] at $t = 9$. By using eight cores, the CPU costs of the original method and the present method are 194.256 s and 196.332 s, which indicate that the present method is as efficient as the original method. Note that the relative volume error is reduced to the order of machine accuracy via employing the present method, see Fig. 5(b).

\textbf{4.2.2. Rayleigh-Taylor instability}

Involving large deformations of the interface, the Rayleigh-Taylor instability problem with high Reynolds number is considered to assess the present method. In the absence of surface tension, a heavy fluid is located upon a light fluid with a small cosine perturbation along the interface, in which the Rayleigh-Taylor instability is induced. The density difference between two fluids is characterized by the Atwood ratio

\begin{equation}
At = \frac{\bar{\rho}_h - \bar{\rho}_l}{\bar{\rho}_h + \bar{\rho}_l},
\end{equation}
Fig. 5. Falling droplet: (a) the evolving interfaces of the falling droplet obtained by the present method at different time instants, which are corresponding to $t = 0, 2, 3, 4, 5, 6, 7, 9$ from top to bottom, respectively, and (b) the relative volume error over time.
which is set to 0.5 according to Refs. [30, 51, 52]. The flow field is solved in a rectangular domain of $[0, 2] \times [0, 4]$ with the initial conditions given by

$$
\begin{cases}
\rho = 1, \ u = 0, \ v = 0, \ p = 1, \ \mu = 1 & \text{heavy fluid}, \\
\rho = 1/3, \ u = 0, \ v = 0, \ p = 1, \ \mu = 1 & \text{light fluid}, \\
\phi = y - 2.0 - 0.1 \cos(2\pi y) & \text{level set}, \\
Re = 3000, \ Fr = 1 & \text{dimensionless parameters}.
\end{cases}
$$

The time step and the termination time are set to $\Delta t = 2.0 \times 10^{-4}$ and $t = 5$, respectively. With the no-slip wall boundary condition imposed in the x-direction and the periodic boundary condition enforced in the y-direction, the simulations are carried out on a $200 \times 800$ grid using eight cores. The computational time of the original method and the present method are $1279.68$ s and $1281.87$ s, respectively, which confirm the high efficiency of the present method. As shown in Fig. 6, the heavy fluid is falling down due to gravity with two counter-rotating vortices rolled up along the interface. Growing with time, the two vortices are shed and a pair of secondary vortices are formed at the tails of the roll-ups. It can be observed that more details are captured with the present method. Fig. 7(a) depicts the time history of the y-coordinates for the top of the rising fluid and the bottom of the falling fluid. Quantitatively, the numerical results of the present method are in good agreement with the previous simulations [30, 51, 52]. In addition, with the present method, the relative volume error is significantly reduced, see Fig. 7(b).

4.2.3. Bubble rising

A two-dimensional bubble rising problem is chosen here. Within a rectangular domain of $[0, 1] \times [0, 2]$ filled with liquid, a circular gas bubble of diameter $D = 0.5$ is initially placed at $(0.5, 0.5)$ and then rises under buoyancy. The Eotvos number and the gravity Reynolds number can be used to characterize the flow field and are defined as

$$
Eo = \frac{\tilde{\rho} g \tilde{D}^2}{\tilde{\sigma}},
$$

$$
Re = \frac{\tilde{\rho} g \tilde{D}^2}{\tilde{\mu}i},
$$

(29)

With $Eo = 125$ and $Re_g = 35$ [30, 53], the initial conditions are accordingly given by

$$
\begin{cases}
\rho = 1, \ u = 0, \ v = 0, \ p = 1, \ \mu = 1 & \text{liquid}, \\
\rho = 0.001, \ u = 0, \ v = 0, \ p = 1, \ \mu = 0.01 & \text{gas}, \\
\phi = \sqrt{(x-0.5)^2 + (y-0.5)^2} - 0.25 & \text{level set}, \\
Re = 100, \ We = 510.2, \ Fr = 1.02 & \text{dimensionless parameters},
\end{cases}
$$

(30)

and the termination time is $t = 3$. With a time step of $\Delta t = 3.0 \times 10^{-5}$, the simulations are carried out on a $128 \times 256$ grid using 8 cores. The CPU
Fig. 6. Rayleigh-Taylor instability: the evolution of the interface obtained by (a) the original method and (b) the present method.
costs of the original method and the present method are 866.64 s and 871.86 s, respectively. Since surface tension is relatively small, a pair of filaments are formed at the bottom of the bubble and become thinner as time goes on, see Fig. 8(a)-(b). In Fig. 8(c), the y-coordinate of the center of mass, 

\[ y_c = \frac{\int_{\Omega} (1 - H_s(\phi))y \, d\Omega}{V_2}, \]  

(31)

is plotted versus the simulation time. It is observed that the results of the present method agree well with the benchmark solution of Aland and Voigt [53]. The conservation of the present method is validated by the time history of the relative volume error given in Fig. 8(d).

Then a three-dimensional bubble rising case is computed to further validate the capability of the present method for 3D problems. By analyzing a mass of experimental data, Grace [54] concluded that the final shapes of a single gas bubble rising in the quiescent liquid can be grouped into four categories: spherical, ellipsoidal, skirted and dimpled. The governing dimensionless parameters are the Morton number \( M \), the Eotvos number \( Eo \) and the terminal Reynolds number \( Re_t \), which are defined as

\[
M = \frac{\bar{g} \tilde{\rho}_f^4 (\tilde{\rho}_f - \tilde{\rho}_g)}{\tilde{\rho}_f^2 \sigma^3},
\]

\[
Eo = \frac{\bar{g} \tilde{D}_s^2 (\tilde{\rho}_f - \tilde{\rho}_g)}{\sigma},
\]

(32)

\[
Re_t = \frac{\tilde{\rho}_f \bar{U}_\infty \tilde{D}}{\tilde{\mu}_f},
\]

where \( \bar{U}_\infty \) is the terminal velocity of the bubble. Note that the definition of

Fig. 7. Rayleigh-Taylor instability: (a) the y-coordinates of the tips of the falling and rising fluid, and (b) the relative volume error over time.
the Eotvos number is different from that in the 2D situation. We consider
the skirted case here as it is the most challenging one due to the large and
rapid deformation of the bubble, see Refs. [34, 55]. Embedded in the liquid, a
spherical gas bubble of diameter $D = 1$ is initially placed at $(3, 3, 1.5)$ within
a cubic domain of $[0, 6] \times [0, 6] \times [0, 18]$. With $M = 0.971$ and $Eo = 97.1$, the
initial conditions are given by

$$
\begin{align*}
\rho &= 1, \ u = 0, \ v = 0, \ w = 0, \ p = 1, \ \mu = 1 \quad \text{liquid}, \\
\rho &= 0.01, \ u = 0, \ v = 0, \ w = 0, \ p = 1, \ \mu = 0.01 \quad \text{gas}, \\
\phi &= \sqrt{(x-3)^2 + (y-3)^2 + (z-1.5)^2} - 0.5 \quad \text{level set}, \\
Re &= 31.32, \ We = 98.13, \ Fr = 1 \quad \text{dimensionless parameters}.
\end{align*}
$$

The termination time is set as $t = 20$ with a time step $\Delta t = 2 \times 10^{-3}$. With the
free-slip wall boundary condition employed for the vertical walls and the no-slip
wall boundary condition enforced at the horizontal walls, the computational
domain is discretized on a grid involving $120 \times 120 \times 360$ cells. Obtained by
using 64 cores, the CPU time of the original method and the present method
are $1762.68$ s and $1776.78$ s, which are very close. As shown in Fig. 9(a)-(b),
the skirted shape is well predicted by the present method while a wrongly
ellipsoidal shape is obtained by the original method owing to numerical diffusion.
According to Ref. [54], the expected terminal Reynolds number is $20$ and the
computed value of the present method is $18.3$, which is obviously better than
$12.4$ predicted by the original method. The time history of the relative volume
error in Fig. 9 also quantitatively highlights the performance of the present
method.

4.2.4. Binary droplets collision

As a classic benchmark case, the three-dimensional binary droplets collision
problem is considered, which has been extensively investigated [33, 44, 56, 57].
With no-slip wall boundary conditions, the computational domain is a cubic box
of $[0, 2.5D] \times [0, 2.5D] \times [0, 5D]$, where $D = 2.0$ is the diameter of the droplets.
The dynamics of the droplets is governed by the diameter-based Weber number

$$
We_d = \frac{\rho U_r^2 D}{\sigma}.
$$
**Fig. 8.** Bubble rising (2D): (a)-(b) the evolving interfaces at different time instants, (c) the y-coordinate of the center of mass over time and (d) the relative volume error over time.

**Fig. 9.** Bubble rising (3D): (a) the 3D interface at $t = 20$, (b) the corresponding 2D cross section profile and (c) the relative volume error over time.
which is tuned by changing the relative velocity $\tilde{U}_r$. Two situations are computed, namely $We_d = 23$ and $We_d = 40$, with the initial conditions given by

$$
\begin{align*}
\rho &= 63.876, \quad \mu = 815.66, \quad p = 1, \quad u = 0, \quad v = 0, \\
\rho &= 1, \quad \mu = 1, \quad p = 1, \quad u = 0, \quad v = 0, \quad w = 0, \\
\phi &= \sqrt{(x - 2.5)^2 + (y - 2.5)^2 + (z - z_d)^2} - 0.5, \\
\phi &= \sqrt{x^2 + y^2 + (z - z_d)^2} - 0.5, \\
\phi &= \sqrt{(x - 2.5)^2 + (y - 2.5)^2 + (z - z_d)^2} - 0.5, \\
\phi &= \sqrt{(x - 2.5)^2 + (y - 2.5)^2 + (z - z_d)^2} - 0.5, \\
\phi &= \sqrt{(x - 2.5)^2 + (y - 2.5)^2 + (z - z_d)^2} - 0.5,
\end{align*}
$$

(35)

The simulations are conducted on a $160 \times 160 \times 320$ grid and the time step is set to $2.5 \times 10^{-4}$. With 64 cores, the simulation time of the original method and the present method are 17513.28 s and 17570.52 s for $We_d = 23$, and are 31512.96 s and 31633.92 s for $We_d = 40$. As shown in Fig. 10 and Fig. 11, during the early stage, the evolutions of the interface for $We_d = 23$ and $We_d = 40$ are quite similar, in which a flying saucer shape is developed. Then, with the larger inertia force, a more severe bounce is observed for $We_d = 40$, resulting in an additional satellite droplet after the separation of the two droplets. The interfaces obtained by the present method match the experimental results very well. Moreover, Fig. 12 confirms that the present method can conserve the volume even with large deformation involved.

![Fig. 10. Binary droplets collision ($We_d = 23$): (a) the experimental results taken from Ashgriz and Poo [56] with permission of Cambridge University Press, and (b) the numerical results obtained by the present method.](image)
Fig. 11. Binary droplets collision ($\text{We}_d = 40$): (a) the experimental results taken from Ashgriz and Poo \[56\] with permission of Cambridge University Press, and (b) the numerical results obtained by the present method.

Fig. 12. Binary droplets collision: the relative volume error over time.
5. Concluding remarks

In this paper, a novel mass correction procedure has been developed to overcome the unphysical mass gain/loss in the level-set method. Inspired by the scale-separation algorithm of Han et al. [35], we introduce a small perturbation to the level-set field to eliminate the conservation error, which can be solved by using the Newton method. With the signed distance property of the level-set function preserved exactly, the correction procedure can be implemented after the reinitialization step. Thus, unlike in previous researches [30, 33, 34], no additional conservation error will be reintroduced. In addition, the present method is easy to implement in both 2D and 3D simulations since only algebraic operations are needed. To validate the present method, a number of benchmark cases involving large deformations are computed. With the present method, the mass is well-conserved and more details in the flow field are captured. Quantitative study indicates that the conservation error is reduced to the order of machine accuracy by employing the present method and the extra computational cost is negligible. In future work, we plan to add a phase change model in the present method to simulate the icing of droplets.

6. Acknowledgements

References

[1] Jiun Der Yu, Shinri Sakai, and James Sethian. A coupled quadrilateral grid level set projection method applied to ink jet simulation. *Journal of Computational Physics*, 206(1):227–251, 2005.

[2] E. Olmos, C. Gentric, Ch Vial, G. Wild, and N. Midoux. Numerical simulation of multiphase flow in bubble column reactors. Influence of bubble coalescence and break-up. *Chemical Engineering Science*, 56(21):6359–6365, 2001.

[3] Michael Rachner, Julian Becker, Christoph Hassa, and Thomas Doerr. Modelling of the atomization of a plain liquid fuel jet in crossflow at gas turbine conditions. *Aerospace Science and Technology*, 6(7):495–506, 2002.

[4] C. W Hirt, A. A Amsden, and J. L Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[5] Y. Ling, A. Haselbacher, and S. Balachandar. A numerical source of small-scale number-density fluctuations in eulerian–lagrangian simulations of multiphase flows. *Journal of Computational Physics*, 229(5):1828–1851, 2010.

[6] Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, 1992.
[7] Murray Rudman. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24(7):671–691, 1997.

[8] W. J Rider and D. B Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141(2):112–152, 1997.

[9] Denis Gueyffier, Jie Li, Ali Nadim, Ruben Scardovelli, and Stéphane Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152(2):423–456, 1999.

[10] Ruben Scardovelli and Stephane Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics*, 164(1):228–237, 2000.

[11] James Edward Pilliod and Elbridge Gerry Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004.

[12] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[13] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. A review of level-set methods and some recent applications. *Journal of Computational Physics*, 353:82–109, 2018.

[14] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[15] A Salih and S Ghosh Moulic. Some numerical studies of interface advection properties of level set method. *Sadhana*, 34(2):271–298, 2009.

[16] Dongbin Xiu and George Em Karniadakis. A semi-lagrangian high-order method for navier–stokes equations. *Journal of computational physics*, 172(2):658–684, 2001.

[17] Robert R Nourgaliev and Theo G Theofanous. High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set. *Journal of Computational Physics*, 224(2):836–866, 2007.

[18] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *Journal of computational physics*, 163(1):51–67, 2000.

[19] Chohong Min and Frédéric Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *Journal of Computational Physics*, 225(1):300–321, 2007.
[20] Marcus Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of computational physics*, 227(4):2674–2706, 2008.

[21] RR Nourgaliev, S Wiri, NT Dinh, and TG Theofanous. On improving mass conservation of level set by reducing spatial discretization errors. *International journal of multiphase flow*, 31(12):1329–1336, 2005.

[22] Elin Olsson and Gunilla Kreiss. A conservative level set method for two phase flow. *Journal of computational physics*, 210(1):225–246, 2005.

[23] Elin Olsson, Gunilla Kreiss, and Sara Zahedi. A conservative level set method for two phase flow ii. *Journal of Computational Physics*, 225(1):785–807, 2007.

[24] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of computational physics*, 227(18):8395–8416, 2008.

[25] T. Ménard, S. Tanguy, and A. Berlemont. Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary breakup of a liquid jet. *International Journal of Multiphase Flow*, 33(5):510–524, 2007.

[26] Mark Sussman and Elbridge Gerry Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, 2000.

[27] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[28] Zhaoyuan Wang, Jianming Yang, and Frederick Stern. An improved particle correction procedure for the particle level set method. *Journal of Computational Physics*, 228(16):5819–5837, 2009.

[29] Yaohong Wang, S Simakhina, and Mark Sussman. A hybrid level set-volume constraint method for incompressible two-phase flow. *Journal of Computational Physics*, 231(19):6438–6471, 2012.

[30] H-Z Yuan, C Shu, Y Wang, and S Shu. A simple mass-conserved level set method for simulation of multiphase flows. *Physics of Fluids*, 30(4):040908, 2018.

[31] SP Van der Pijl, A Segal, C Vuik, and P Wesseling. A mass-conserving level-set method for modelling of multi-phase flows. *International journal for numerical methods in fluids*, 47(4):339–361, 2005.

[32] SP Van der Pijl, A Segal, C Vuik, and P Wesseling. Computing three-dimensional two-phase flows with a mass-conserving level set method. *Computing and Visualization in Science*, 11(4):221–235, 2008.
[33] Kun Luo, Changxiao Shao, Yue Yang, and Jianren Fan. A mass conserving level set method for detailed numerical simulation of liquid atomization. *Journal of Computational Physics*, 298:495–519, 2015.

[34] Zhouyang Ge, Jean-Christophe Loiseau, Outi Tammissola, and Luca Brandt. An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces. *Journal of Computational Physics*, 353:435–459, 2018.

[35] LH Han, XY Hu, and Nikolaus A Adams. Scale separation for multi-scale modeling of free-surface and two-phase flows with the conservative sharp interface method. *Journal of Computational Physics*, 280:387–403, 2015.

[36] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999.

[37] Mark Sussman, Emad Fatemi, Peter Smereka, and Stanley Osher. An improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5-6):663–680, 1998.

[38] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968.

[39] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.

[40] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[41] Michael S Dodd and Antonino Ferrante. A fast pressure-correction method for incompressible two-fluid flows. *Journal of Computational Physics*, 273:416–434, 2014.

[42] Chohong Min. On reinitializing level set functions. *Journal of Computational Physics*, 229(8):2764–2772, 2010.

[43] Steven T Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

[44] Pao-Hsiung Chiu. A coupled phase field framework for solving incompressible two-phase flows. *Journal of Computational Physics*, 392:115–140, 2019.

[45] John B Bell, Phillip Colella, and Harland M Glaz. A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85(2):257–283, 1989.
[46] Randall J Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33(2):627–665, 1996.

[47] Abbas Fakhari, Martin Geier, and Taehun Lee. A mass-conserving lattice boltzmann method with dynamic grid refinement for immiscible two-phase flows. *Journal of Computational Physics*, 315:434–457, 2016.

[48] Tian Long, Jinsheeng Cai, and Shucheng Pan. An accelerated conservative sharp-interface method for multiphase flows simulations. *Journal of Computational Physics*, 429:110021, 2021.

[49] Jaehoon Han and Grétar Tryggvason. Secondary breakup of axisymmetric liquid drops. i. acceleration by a constant body force. *Physics of Fluids*, 11(12):3650–3667, 1999.

[50] M Jalaal and K Mehravaran. Fragmentation of falling liquid droplets in bag breakup mode. *International Journal of Multiphase Flow*, 47:115–132, 2012.

[51] J-L Guermond and Luigi Quartapelle. A projection fem for variable density incompressible flows. *Journal of Computational Physics*, 165(1):167–188, 2000.

[52] Hang Ding, Peter DM Spelt, and Chang Shu. Diffuse interface model for incompressible two-phase flows with large density ratios. *Journal of Computational Physics*, 226(2):2078–2095, 2007.

[53] S Aland and A Voigt. Benchmark computations of diffuse interface models for two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids*, 69(3):747–761, 2012.

[54] J Grace. Shapes and velocities of bubbles rising in infinite liquid. *Transactions of the Institution of Chemical Engineers*, 51:116–120, 1973.

[55] M van Sint Annaland, NG Deen, and JAM Kuipers. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical engineering science*, 60(11):2999–3011, 2005.

[56] N Ashgriz and JY Poo. Coalescence and separation in binary collisions of liquid drops. *Journal of Fluid Mechanics*, 221:183–204, 1990.

[57] Sébastien Tanguy and Alain Berlemont. Application of a level set method for simulation of droplet collisions. *International journal of multiphase flow*, 31(9):1015–1035, 2005.