Journal of Computational Physics 478 (2023) 111971

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A consistent adaptive level set framework for incompressible two-phase flows with high density ratios and high Reynolds numbers

Yadong Zeng [a], Han Liu [a], Qiang Gao [a], Ann Almgren [b], Amneet Pal Singh Bhalla [c], Lian Shen [a,*]

[a] Department of Mechanical Engineering and Saint Anthony Falls Laboratory, University of Minnesota, Minneapolis, MN 55455, USA
[b] Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[c] Department of Mechanical Engineering, San Diego State University, San Diego, CA 92182, USA

ARTICLE INFO

Article history:
Received 5 February 2022
Received in revised form 11 December 2022
Accepted 21 January 2023
Available online 27 January 2023

Keywords:
Adaptive mesh refinement (AMR)
Level set
Two-phase flow
Consistent transport
High density ratio/High Reynolds number
Subcycling/Non-subcycling

ABSTRACT

We develop a consistent adaptive framework in a multilevel collocated grid layout for simulating two-phase flows with adaptive mesh refinement (AMR). The conservative momentum equations and the mass equation are solved in the present consistent framework. This consistent mass and momentum transport treatment greatly improves the accuracy and robustness for simulating two-phase flows with a high density ratio and high Reynolds number. The interface capturing level set method is coupled with the conservative form of the Navier–Stokes equations, and the multilevel reinitialization technique is applied for mass conservation. This adaptive framework allows us to advance all variables level by level using either the subcycling or the non-subcycling method to decouple the data advancement on each level. The accuracy and robustness of the framework are validated by a variety of canonical two-phase flow problems. We demonstrate that the consistent scheme results in a numerically stable solution in flows with high density ratios (up to 10^6) and high Reynolds numbers (up to 10^6), while the inconsistent scheme exhibits non-physical fluid behaviors in these tests. Furthermore, it is shown that the subcycling and non-subcycling methods provide consistent results and that both of them can accurately resolve the interfaces of the two-phase flows with surface tension effects. Finally, a 3D breaking wave problem is simulated to show the efficiency and significant speedup of the proposed framework using AMR.

1. Introduction

Numerical simulation of immiscible fluids involving liquid and gas is of considerable interest in many environmental problems and engineering applications, such as wind–wave interactions [1,2], ship hydrodynamics [3], bubbly flows [4], and liquid jets [5]. To simulate two-phase flow problems, it is necessary to resort to approaches that can accurately capture the moving interface and resolve surface tension effects [6]. There are many approaches to address these two-phase flow prob-

* Corresponding author.
E-mail address: shen@umn.edu (L. Shen).

https://doi.org/10.1016/j.jcp.2023.111971
0021-9991/

lems, such as the front-tracking method [7], the volume-of-fluid (VOF) method [8–10], and the level set (LS) method [11,12]. The front tracking method solves the fluid part on a background Eulerian mesh while capturing the moving interface using Lagrangian markers [7]. However, it takes more computational resources to remesh when the deformed topology changes significantly [13]. The VOF method is of great interest because of its good conservation property and numerical robustness [8,10]. This method relies on the specific geometric advection scheme considering the discontinuity of the VOF scalar across the interface and is susceptible to a parasitic current problem [14] caused by poor curvature estimation. Alternatively, the LS method can accurately capture the topology change and the curvature of free surfaces [11,12]. Although the LS method suffers from a mass conservation issue due to the deviation from the signed distance LS function and numerical dissipation [15], using the reinitialization algorithms [16–18] can greatly help avoid this issue. In addition to the above methods, Kang et al. [19] extended the ghost fluid method (GFM) in [20] to capture the boundary conditions at a contact discontinuity of the multiphase incompressible flow. Lalanne et al. [21] built a unified framework, including both the level set method and the GFM, to evaluate their efficiency and accuracy. Similarly, Son and Dhir [22] utilized the LS method and GFM approach to match the boundary conditions at the fluid-solid interface, aiming at simulating film boiling on an immersed (or irregularly shaped) surface. Olsson et al. [23] introduced a conservative level set (CLS) approach, in which a conservative scheme with an intermediate step is used to keep the shape and width of the profile across the interface constant. Desjardins et al. [24] combined the accurate CLS method with the GFM approach for simulating two-phase flows, which shows that conservation errors associated with the accurate conservative level set technique remain small even for a complex liquid Diesel jet case. In the present work, we combine the LS method with the reinitialization technique [15,17,25] to mitigate the mass loss of the fluids.

A challenging problem in the multiphase flow community is the treatment of flows with high density ratios [26–28] and high Reynolds numbers [29–31], for which special numerical techniques are needed to ensure numerical stability. Recent studies have begun to address the numerical instabilities associated with flows with density ratios on the order of 10^{3-4} and greater. A variety of numerical approaches has been reported, including the LS method [29], VOF method [26,27], lattice Boltzmann method [27,32], and diffused interface method [33]. For example, Sussman et al. [34] proposed a sharp-interface coupled level-set and volume-of-fluid (CLSVOF) method to solve the liquid and gas system separately. Because the extrapolated velocity from one side is used to advect the LS function, the VOF function, and the velocity, this method can handle flows with a high density ratio. Ding et al. [33] formulated a divergence-free staggered-velocity field from conservation laws, which works well for two-phase flows with a high density ratio in the context of the diffused interface method. Rudman [26] and Bussmann et al. [27] found that using a scheme to consistently transport the momentum equations and VOF scalar can reduce the problem of numerical instability. In the LS context, Raessi [28] and Raessi and Pitsch [35] first introduced geometric mass flux transport for tightly coupling the mass and momentum. However, their methods are limited to one and two dimensions owing to the inherent difficulty of reconstructing the interface using the LS function in three dimensions. Desjardins and Moureau [36] designed a consistent transport scheme for the 3D staggered grid in the context of the LS method, and Ghods and Herrmann [37] extended this scheme to the collocated unstructured grid. However, only the first-order upwind scheme was used for the density and velocity advection in [36,37], and this scheme smears both interface and velocities. An improved third-order Cubic Upwind Interpolation (CUI) interpolation scheme was proposed by Patel and Natarajan [38] in the hybrid staggered/nonstaggered framework for the consistent transport of mass and momentum. The CUI interpolation scheme was also used in [29,30,39,40] in the context of the staggered grid. In the present work, we employ the CUI scheme on a collocated grid to convect mass and momentum for robust simulations of flows with high density ratios. Using the collocated grid significantly simplifies the implementation of the interpolation schemes and the differential operators when multiple levels of grids are involved.

To capture the complex physical processes involved in two-phase flow problems, the grid cells around the free surface are sometimes refined to resolve small flow structures [6]. In addition to improving the resolution of the solution, the refined grid cells also alleviate the mass conservation problem to a great extent when using the LS method [29,41]. However, one does not need (or maybe cannot afford) a fine grid of uniformly high resolution across the whole domain. The need to resolve fine local features near the interface can be addressed by the adaptive mesh refinement (AMR) method. Briefly, the AMR simulations have high resolution at the places of interest and lower resolution elsewhere [42,43]. Based on the grid hierarchy and data structure, AMR methods can be classified into two groups: quadtree/octree-based AMR (TBAMR) [44–46] and block-structured AMR (BSAMR) [42,43,47–51]. In TBAMR, each cell can be split into four cells in two dimensions or eight cells in three dimensions, and the hierarchy of the grid cells is organized using a tree structure [44,45,52]. BSAMR, on the other hand, builds the mesh as nested Cartesian grids [42,43,47–50,53–55]. It is relatively easy to use the multigrid (MG) solver for solving partial differential equations (PDEs) [47,50] and the domain decomposition method for parallelization [56]. In the present work, we choose BSAMR as the basis of our adaptive framework.

Over the past several decades, many researchers have begun to combine AMR with the above interface-tracking methods to simulate two-phase flow problems [17,57–59]. For example, Sussman et al. [17] developed an adaptive LS method for incompressible two-phase flows using the BSAMR framework, and Popinet [58] used the VOF method with the TBAMR framework for surface-tension-driven interfacial flows. Nevertheless, none of them combined AMR with the above consistent transport scheme to simulate two-phase flow problems with high density ratios. Some works [29,39,40] combine the CUI interpolation scheme with BSAMR to obtain stable solutions as well as to reduce the computational cost for high-density-ratio flows. In those studies, the two-phase flow solutions were updated in time using the composite advancement method. For example, discretized equations of velocity and pressure were coupled through coarse-fine boundaries and solved for

multiple levels simultaneously. However, because of this coupling, the time step was restricted by the finest grid spacing for numerical stability. In our previous work [25], we developed a BSAMR framework that can utilize the level-by-level advancement method, especially the subcycling method, for the simulation of two-phase flows. However, that framework cannot handle high-density-ratio two-phase flow problems owing to the use of an inconsistent transport scheme. In the present work, we incorporate the consistent scheme into the BSAMR framework, with both subcycling and non-subcycling methods for level-by-level time advancement.

The main objective of the present work is to develop a consistent and unified adaptive framework to simulate two-phase flows with high density ratios and high Reynolds numbers. The *AMReX* package [60] is utilized to manage the multilevel grids and perform parallel operations. There are three contributions of this paper. First, we present a consistent transport scheme on a collocated grid that can transport mass and momentum consistently, which is specifically designed for the capability of the LS-based two-phase flow solver in simulating flows with high density ratios and high Reynolds numbers. Second, we embed the consistent scheme into an adaptive BSAMR framework. Both the subcycling method and the non-subcycling method can be chosen to advance the solutions level by level. To our knowledge, this is the first framework that utilizes both subcycling and non-subcycling methods to simulate two-phase flows with a consistent transport scheme. A theoretical formula of the speedup is also derived by comparing the computational cost between the subcycling and non-subcycling methods. Third, various canonical cases are tested to validate the robustness and efficacy of the framework for accommodating high-density-ratio and high-Reynolds-number two-phase flows.

The remainder of this paper is organized as follows: we start with the continuous formulation of the Navier–Stokes equations and the LS advection equation for incompressible two-phase flows in Section 2, followed by a description of concepts and definitions of BSAMR in Section 3. Next, both the single-level and multilevel advancement algorithms using the consistent scheme are presented in Section 4. Validation cases that highlight the importance of consistent mass and momentum transport are then given in Section 5. A comparison between the inconsistent scheme and the consistent scheme is also presented. Finally, the conclusions are given in Section 6.

2. Mathematical formulation

We begin with the conservative form of the incompressible Navier–Stokes equations,

$$\begin{aligned}&\frac{\partial\left(\rho\left(\mathbf{x},t\right)\mathbf{u}\left(\mathbf{x},t\right)\right)}{\partial t}+\nabla\cdot\left(\rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)\right)\\&\quad=-\nabla p(\mathbf{x},t)+\nabla\cdot\left[\mu(\mathbf{x},t)\left(\nabla\mathbf{u}(\mathbf{x},t)+\nabla\mathbf{u}(\mathbf{x},t)^{T}\right)\right]+\rho(\mathbf{x},t)\mathbf{g}+\mathbf{f}_{s}(\mathbf{x},t),\end{aligned}\tag{1}$$

$$\nabla\cdot\mathbf{u}(\mathbf{x},t)=0,\tag{2}$$

where $\mathbf{u}(\mathbf{x},t)$, $p(\mathbf{x},t)$, $\rho(\mathbf{x},t)$, and $\mu(\mathbf{x},t)$ are the spatially and temporally varying fluid velocity, pressure, density, and dynamic viscosity, respectively, $\mathbf{g}$ is the vector form of the gravitational acceleration, and $\mathbf{f}_s(\mathbf{x},t)$ is the continuum surface tension force. Note that the continuity constraint Eq. (2) is obtained from the incompressibility nature of the fluid $D\rho(\mathbf{x},t)/Dt=0$, and mass conservation over the computational domain is expressed as

$$\frac{\partial\rho(\mathbf{x},t)}{\partial t}+\nabla\cdot\rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)=\frac{\mathrm{D}\rho(\mathbf{x},t)}{\mathrm{D}t}+\rho(\mathbf{x},t)\nabla\cdot\mathbf{u}(\mathbf{x},t)=0.\tag{3}$$

By combining Eqs. (1), (2), and (3), the incompressible Navier–Stokes equations can also be cast into nonconservative form as

$$\rho(\mathbf{x},t)\left(\frac{\partial\mathbf{u}(\mathbf{x},t)}{\partial t}+\nabla\cdot\mathbf{u}(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)\right)=-\nabla p(\mathbf{x},t)+\nabla\cdot\left[\mu(\mathbf{x},t)\left(\nabla\mathbf{u}(\mathbf{x},t)+\nabla\mathbf{u}(\mathbf{x},t)^{T}\right)\right]+\rho(\mathbf{x},t)\mathbf{g}+\mathbf{f}_{s}(\mathbf{x},t),\tag{4}$$

$$\nabla\cdot\mathbf{u}(\mathbf{x},t)=0,\tag{5}$$

Although the conservative and nonconservative forms of the Navier–Stokes equations are mathematically equivalent, different numerical treatments lead to different behaviors of the simulation results, as shown in later sections.

In this work, we use the LS function $\phi(\mathbf{x},t)$ to explicitly capture the evolution of the two-phase interface [15,61]. As illustrated in the left part of Fig. 1, the densities of the liquid and gas are denoted by ρ_l and ρ_g, respectively, and the corresponding dynamic viscosities are μ_l and μ_g. The right part of Fig. 1 shows that $\phi(\mathbf{x},t)$ is a signed distance from the two-phase interface, with $\phi(\mathbf{x},t)>0$ in phase 1 and $\phi(\mathbf{x},t)<0$ in phase 2. The advection equation of $\phi(\mathbf{x},t)$ is

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t}+\nabla\cdot(\mathbf{u}\phi(\mathbf{x},t))=0\tag{6}$$

and the density and dynamic viscosity can be calculated using $\phi(\mathbf{x},t)$ as

$$\rho(\mathbf{x},t)=\rho_g\left[1-H\left(\phi(\mathbf{x},t)\right)\right]+\rho_l H\left(\phi(\mathbf{x},t)\right),\tag{7}$$

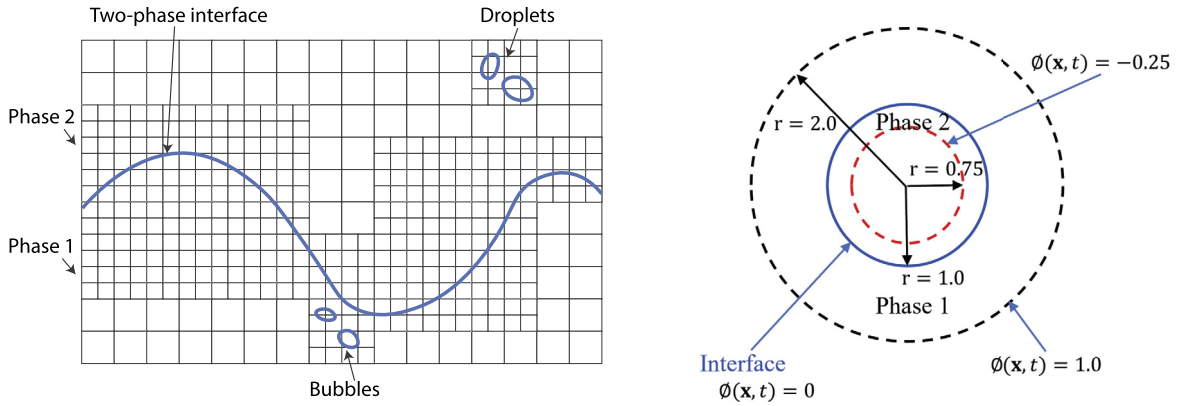


Fig. 1. Left: two-phase flow on a multilevel Cartesian grid. Right: schematic definition of the LS function.

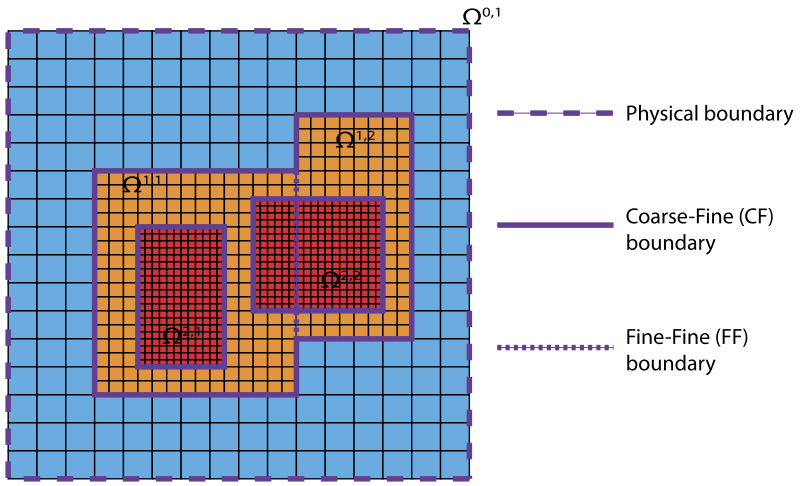


Fig. 2. Diagram of a three-level adaptive grid with three types of boundaries. $\Omega^{i,j}$ represents patch j on level i for all $i \geq 0$, $j \geq 1$.

$$\mu(\mathbf{x},t) = \mu_g\left[1 - H\big(\phi(\mathbf{x},t)\big)\right] + \mu_l H\big(\phi(\mathbf{x},t)\big), \tag{8}$$

Here, $H\big(\phi(\mathbf{x},t)\big)$ is the Heaviside function and can be smoothed around the interface as [15,17]

$$H\big(\phi(\mathbf{x},t)\big) = \begin{cases} 0, & \phi(\mathbf{x},t) < -\epsilon, \\ \frac{1}{2}\left[1 + \frac{x}{\epsilon} - \frac{1}{\pi}\sin\left(\frac{\pi\phi(\mathbf{x},t)}{\epsilon}\right)\right], & |\phi(\mathbf{x},t)| \leq \epsilon, \\ 1, & \phi(\mathbf{x},t) > \epsilon, \end{cases} \tag{9}$$

where ϵ is the smearing width and is usually set to be one or two grid lengths [15,29].

3. Concepts and definitions in BSAMR

This section introduces some important concepts of BSAMR [60]. In the present work, the coarsest level of the grid in the entire computational domain Γ is referred to as level 0. The finest level that the grid can be refined to is denoted as level l_{max}. In other words, the total number of levels is $l_{max} + 1$. Fig. 2 illustrates a three-level adaptive grid with $l_{max} = 2$ as an example.

Grid cells can be dynamically tagged and refined following certain criteria [43]. In BSAMR, although tagging is performed on individual cells, we do not refine or de-refine the cells individually. Instead, these tagged cells are grouped to form a series of rectangular patches for 2D grids or cuboid patches for 3D grids. There can be more than one patch on a specific level, and these patches are refined simultaneously to the next level. For example, in Fig. 2, level 2 consists of two patches. Because the BSAMR uses a nesting hierarchy of rectangular patches, the union patches on level $l+1$ must be contained in the union patches on level l for all $0 \leq l < l_{max}$, i.e., $\Gamma^{l+1} \subset \Gamma^l$, where Γ^l denotes the union of the patches on level l. Because of this nesting property, three types of boundaries exist on the adaptive grid:

- Physical boundary: the boundary that encloses the computational domain, illustrated using the dashed lines in Fig. 2.
- Coarse–fine (CF) boundary: the boundary between the grid cells of different levels. These boundaries are illustrated using the thick solid lines in Fig. 2.
- Fine–fine (FF) boundary: the boundary between two patches at the same level, marked using the dotted lines in Fig. 2.

Ghost cells are defined at all boundaries, and their values are assigned to represent the boundary effects accurately. The ghost cells at the physical boundaries are filled based on the physical boundary conditions. At the CF boundaries, we adopt a conservative interpolation ($\mathcal{I}_{cons}$) method [47,60] to fill the ghost cells of the fine level. Specifically, we reconstruct a continuous functional form, $f(x)$, on the ghost cells by combining the values of the coarse level and the values of the fine level. In addition to satisfying the continuity across the CF boundary, the function $f(x)$ is also subject to the requirement that the average of $f(x)$ over the area of a coarse cell is equal to the original coarse cell value [62]. Owing to the collocated grid, only one set of interpolation and averaging schemes is needed for all cell-centered variables when the multilevel grid is involved. The conservative interpolation scheme maintains the second-order accuracy of the proposed multilevel algorithms, as verified in Section 5. At the FF boundaries, the ghost cell values are copied from neighboring patches.

4. Time advancement

In this work, we use a level-by-level method [48,49] for the time advancement of variables on the multilevel grid. As our multilevel advancement algorithm is based on the single-level advancement method, we first introduce the consistent scheme for the single-level advancement in Section 4.1. The multilevel advancement algorithm is discussed in Section 4.2, which combines the single-level advancement algorithm with two different cycling methods, i.e., the subcycling and non-subcycling methods, in Section 4.2.1. The synchronization operations are discussed in Section 4.2.2. A summary of the multilevel advancement algorithm is given in Section 4.2.3.

4.1. Single-level advancement

This section mainly discusses the temporal and spatial discretizations of equations using the consistent scheme for the single-level advancement. Because our main focus is the consistent scheme, the numerical details of the inconsistent schemes [25] are placed in Appendix A.

4.1.1. Consistent scheme

For the consistent scheme, the discretizations are applied to the conservative form of the Navier–Stokes equations (Eqs. (1) and (2)). At time t^n, we have the velocity $\mathbf{u}^n$, pressure $p^{n-1/2}$, and LS function ϕ^n [17,29]. During the time interval $[t^n, t^{n+1}]$, the time step proceeds as follows.

Step 1: The LS function ϕ, which is used to describe the two-phase interface, is updated by

$$\frac{\phi^{n+1}-\phi^n}{\Delta t} + Q\left(\mathbf{u}_{\text{adv}}^{n+\frac{1}{2}}, \phi^{n+\frac{1}{2}}\right) = 0 \tag{10}$$

where $Q\left(\mathbf{u}_{\text{adv}}^{n+\frac{1}{2}}, \phi^{n+\frac{1}{2}}\right)$ is computed using the second-order Godunov scheme [17,47].

Step 2: The LS function ϕ^{n+1} is then reinitialized to keep its signed distance property and guarantee mass conservation of the two-phase flow. In this step, a temporary LS function $d(\boldsymbol{x}, \tau)$ is updated iteratively using the following pseudoevolution equation:

$$\frac{\partial d}{\partial \tau} = S(\phi)(1 - |\nabla d|), \tag{11}$$

with the initial condition

$$d(\tau = 0) = \phi^{n+1}, \tag{12}$$

where

$$S(\phi) = 2\left(H(\phi) - 1/2\right). \tag{13}$$

Here, τ is the pseudotime for iterations. A second-order Runge–Kutta method is applied for the pseudo time advancement d [15,17]. To ensure mass conservation, the LS function ϕ^{n+1} is corrected by d after pseudotime advancement [15,17,61]. The midpoint value of ϕ is then calculated as $\phi^{n+\frac{1}{2}} = (\phi^{n+1} + \phi^n)/2$.

Step 3: The viscosity μ^{n+1} field is reset through the Heaviside function as

$$\mu^{n+1} = \mu_g + \left(\mu_l - \mu_g\right)\widetilde{H}\left(\phi^{n+1}\right), \tag{14}$$

where μ_l and μ_g are the viscosities of the liquid phase and the gas phase, respectively. The smoothed Heaviside function, which has been regularized over n_{cells} grid cells on either side of the two-phase interface, is defined as

$$\widetilde{H}\left(\phi^{n+1}\right)=\begin{cases}0, & \phi^{n+1}<-n_{\text{cells}}\Delta x\\ \frac{1}{2}\left(1+\frac{1}{n_{\text{cells}}\Delta x}\phi^{n+1}+\frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{cells}}\Delta x}\phi^{n+1}\right)\right), & \left|\phi^{n+1}\right|\le n_{\text{cells}}\Delta x\\ 1, & \text{otherwise}\end{cases} \tag{15}$$

where the uniform grid spacing $\Delta x=\Delta y$ is assumed to be [29] and $n_{cells}=1$ or 2 are applied for all testing cases in Section 5. We have tested $n_{cells}\ge 3$ and found that larger smearing width leads to more diffusion and generate spurious vortices near the interface. Therefore, we choose $n_{cells}=1$ or 2, depending on the specific simulation case. Although $n_{cells}=1$ leads to a sharper interface, $n_{cells}=2$ leads to less iterations and accelerates the convergence of the multigrid solver used in this work. This result is consistent with the numerical experiments in Nangia et al. [29] and Yang et al. [30], which also used $n_{cells}=1$ or $n_{cells}=2$ for their consistent schemes. The midpoint value of μ is then calculated as $\mu^{n+\frac{1}{2}}=(\mu^{n+1}+\mu^{n})/2$.

The density field evolves based on Eq. (3) using the third-order accurate strong stability preserving Runge–Kutta (SSP-RK3) time integrator [29,39] as

$$\begin{aligned}\rho^{(1)}&=\rho^{n}-\Delta t\mathbf{R}\left(\mathbf{u}^{n}_{\text{adv}},\rho^{n}_{\text{lim}}\right),\\ \rho^{(2)}&=\frac{3}{4}\rho^{n}+\frac{1}{4}\rho^{(1)}-\frac{1}{4}\Delta t\mathrm{R}\left(\mathbf{u}^{(1)}_{\text{adv}},\rho^{(1)}_{\text{lim}}\right),\\ \rho^{n+1}&=\frac{1}{3}\rho^{n}+\frac{2}{3}\rho^{(2)}-\frac{2}{3}\Delta t\mathrm{R}\left(\mathbf{u}^{(2)}_{\text{adv}}\cdot\rho^{(2)}_{\text{lim}}\right).\end{aligned} \tag{16}$$

$\mathrm{R}\left(\mathbf{u}_{\text{adv}},\check{\rho}_{\text{lim}}\right)\approx\left(\nabla\cdot\left(\mathbf{u}_{\text{adv}}\check{\rho}_{\text{lim}}\right)\right)_{i,j}$ is an explicit CUI approximation to the cell-centered advection term of the density. The details of the CUI scheme are given in Section 4.1.2. Here, the subscript 'lim' indicates the limited face-centered variables, and these variables are interpolated from the corresponding cell-centered variables using the CUI scheme (Section 4.1.2). In the variables $\mathbf{u}^{n}_{\text{adv}}$, $\mathbf{u}^{1}_{\text{adv}}$, and $\mathbf{u}^{2}_{\text{adv}}$, the subscript 'adv' represents the face-centered advection velocities. To obtain these velocities, we define two auxiliary variables $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ as

$$\mathbf{u}^{(1)}=2\mathbf{u}^{n}-\mathbf{u}^{n-1}, \tag{17}$$

$$\mathbf{u}^{(2)}=\frac{3}{2}\mathbf{u}^{n}-\frac{1}{2}\mathbf{u}^{n-1}. \tag{18}$$

Note that $\mathbf{u}^{n-1}$ and $\mathbf{u}^{n}$ are the cell-centered velocity in the previous time step and the current time step, respectively. As shown in [29,39], it is crucial to use these interpolated and extrapolated velocities to maintain the time accuracy of the consistent scheme while calculating the advection velocities. The cell-centered velocities, including $\mathbf{u}^{n}$, $\mathbf{u}^{(1)}$, and $\mathbf{u}^{(2)}$, are then averaged onto the face of the cell-centered control volume to obtain $\mathbf{u}^{n}_{fc}$, $\mathbf{u}^{(1)}_{fc}$, and $\mathbf{u}^{(2)}_{fc}$. Finally, the marker and cell (MAC) projection is applied to these face-centered velocities to obtain the divergence-free advection velocities $\mathbf{u}^{n}_{\text{adv}}$, $\mathbf{u}^{(1)}_{\text{adv}}$, and $\mathbf{u}^{(2)}_{\text{adv}}$ [17,47]. The midpoint value of the density is calculated by $\rho^{n+\frac{1}{2}}=(\rho^{n+1}+\rho^{n})/2$.

Step 4. The intermediate velocity $\widetilde{\mathbf{u}}^{*,n+1}$ is solved semi-implicitly as

$$\frac{\rho^{n+1}\tilde{\mathbf{u}}^{n+1,*}-\rho^{n}\mathbf{u}^{n}}{\Delta t}+\mathbf{C}^{n+\frac{1}{2}}=-\nabla p^{n-\frac{1}{2}}+\frac{1}{2}(\nabla\cdot(\mu^{n+\frac{1}{2}}\nabla\mathbf{u}^{n})+\nabla\cdot(\mu^{n+\frac{1}{2}}\nabla\tilde{\mathbf{u}}^{n+1,*}))+\rho^{n+\frac{1}{2}}\mathbf{g}+\mathbf{f}_{\text{s}}^{n+\frac{1}{2}}, \tag{19}$$

where the approximation to the convective derivative is given by

$$\mathrm{C}\left(\mathbf{u}^{(2)}_{\text{adv}},\boldsymbol{\rho}^{(2)}_{\text{lim}}\mathbf{u}^{(2)}_{\text{lim}}\right)\approx\left[\left(\nabla\cdot\left(\mathbf{u}^{(2)}_{\text{adv}}\rho^{(2)}_{\text{lim}}u^{(2)}_{\text{lim}}\right)\right)_{i,j},\left(\nabla\cdot\left(\mathbf{u}^{(2)}_{\text{adv}}\rho^{(2)}_{\text{lim}}v^{(2)}_{\text{lim}}\right)\right)_{i,j}\right] \tag{20}$$

using the CUI scheme (Section 4.1.2). Here, we use the same advection velocity $\mathbf{u}_{\text{adv}}$ and the limited density $\rho^{(2)}_{\text{lim}}$ as in Eq. (16). This is the key requirement to ensure consistent mass and moment transport [39]. Finally, we note that the transition region continues to grow wider as the simulation time increases if and only if we use the convected density field to capture the interface. However, in both our consistent and inconsistent schemes, the density field ρ^{n+1} is reset by the LS function ϕ^{n+1} using Eq. (41) after the **Step 4** of each time step. This postprocessing step helps synchronize the density field and LS field to avoid significant distortions in the interface and the growth of the transition regions [29,30]. In Appendix B, a counterexample of a 2D convected droplet is illustrated using the consistent scheme, in which the density field is not reset by the level set function. We can clearly see that the density region grows as the simulation time increases. However, despite the "unbounded" growth of transition region, the simulation remains stable because of the consistent mass-momentum transport formulation.

Step 5: After obtaining the intermediate velocity, a level projection is applied to obtain the updated velocity $\widetilde{\mathbf{u}}^{n+1}$ and pressure $p^{n+1/2}$ fields. An auxiliary variable $\boldsymbol{V}$ is first calculated by

$$\boldsymbol{V} = \frac{\widetilde{\mathbf{u}}^{*,n+1}}{\Delta t} + \frac{1}{\rho^{n+1/2}} \nabla p^{n-\frac{1}{2}}. \tag{21}$$

Then, $\boldsymbol{V}$ is projected on the divergence-free velocity field to obtain the updated pressure $p^{n+1/2}$ via

$$L^{cc,l}_{\rho^{n+1/2}} p^{n+1/2} = \nabla \cdot \boldsymbol{V}, \tag{22}$$

where $L^{cc,l}_{\rho^{n+1/2}} p^{n+1/2}$ is the density-weighted approximation to $\nabla \cdot (1/\rho^{n+1/2} \nabla p^{n+1/2})$ on level l. Finally, the divergence-free velocity $\widetilde{\mathbf{u}}^{n+1}$ on level l is obtained as

$$\widetilde{\mathbf{u}}^{n+1} = \Delta t \left(\boldsymbol{V} - \frac{1}{\rho^{n+1/2}} \nabla p^{n+1/2} \right). \tag{23}$$

As a remark, the level gradient operator ∇ is not the minus transpose of the level divergence operator $\nabla\cdot$, i.e., $\nabla \neq -(\nabla\cdot)^T$ [48,49]. As a result, the idempotency of the approximate projection $\boldsymbol{P} = I - \nabla(\Delta)^{-1}\nabla\cdot$ is not ensured [47], i.e., $\boldsymbol{P}^2 \neq \boldsymbol{P}$. Yet, this nonidempotent approximate projection is stable and appears to be well-behaved in various numerical tests [48,63,64] and practical applications [17,49]. Notably, for a uniform single grid with periodic boundary conditions, Lai [65] theoretically proved that this approximate projection method is stable, in that $\|\boldsymbol{P}\| \leq 1$. We remark that the level projection is applied to the intermediate velocity $\boldsymbol{u}^{*,n+1}$ (Eq. (21)). Compared with the form that projects the increment velocity $\boldsymbol{u}^{*,n+1} - \boldsymbol{u}^{n}$, e.g., as used in Almgren et al. [47], the projection method used here can reduce the accumulation of pressure errors and lead to a more stable algorithm [64,66,67]. In our previous work [25], both the level projection and the synchronization projection methods are tested with a sample problem [44], which confirms the stability of the approximate projection method. We also simulated the Taylor–Green vortex (TGV) case there to show that our numerical schemes can achieve the desired second order of accuracy on a static multi-level mesh for both the subcycling and non-subcycling methods.

4.1.2. Discretization of the convective term and surface tension term

To obtain the convective terms in Eqs. (16) and (20), a third-order accurate Koren's limited CUI scheme is applied, which was first proposed by Roe and Baines [68] and further studied by Waterson and Deconinck [69] and Patel and Natarajan [38] in the multiphase flow simulation. For simplicity, only the 2D discretized formulas of the CUI scheme are given in this section. The 3D formulas can be extended in a straightforward way.

A cell-centered variable ψ_{lim}, which can be ρ_{lim} in Eqs. (16) or $\rho_{\text{lim}}\mathbf{u}_{\text{lim}}$ in Eqs. (20), is advected by $\mathbf{u}_{\text{adv}}$ using

$$\nabla \cdot (\mathbf{u}_{\text{adv}} \psi_{\text{lim}})_{i,j} = \frac{u_e \psi_e - u_w \psi_w}{\Delta x} + \frac{v_n \psi_n - v_s \psi_s}{\Delta y}, \tag{24}$$

where u_e, u_w, v_n, and v_s are edge-centered divergence-free velocities located on the east, west, north, and south sides of cell (i, j). With these associated velocities, the CUI scheme is utilized to obtain ψ_w, ψ_e, ψ_n, and ψ_s on the edges of the control volume, which is marked by the green dashed line in Fig. 3. For a given face of the cell centered control volume $f \in \{e, w, n, s\}$, the upwind ψ_C, the far upwind ψ_U and the downwind ψ_D are labeled according to the direction of the edge-centered advection velocity. The upwinded approximation of ψ_f is then calculated as [29]

$$\widetilde{\psi}_f = \begin{cases} 3\widetilde{\psi}_C, & 0 < \widetilde{\psi}_C \leq \frac{2}{13} \\ \frac{5}{6}\widetilde{\psi}_C + \frac{1}{3}, & \frac{2}{13} < \widetilde{\psi}_C \leq \frac{4}{5} \\ 1, & \frac{4}{5} < \widetilde{\psi}_C \leq 1 \\ \widetilde{\psi}_C, & \text{otherwise} \end{cases} \tag{25}$$

where the normalized value is given by [69]

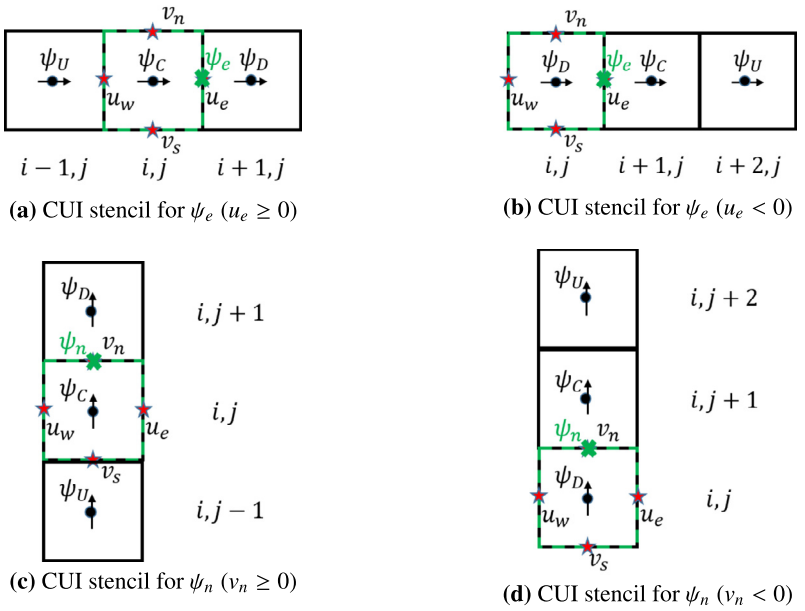
$$\widetilde{\psi} = \frac{\psi - \psi_U}{\psi_D - \psi_U}. \tag{26}$$

After obtaining ψ_w, ψ_e, ψ_n, and ψ_s on the edges of the control volume f in Eq. (25), the convective terms can be calculated using Eq. (24).

We note that the CUI scheme satisfies both a convection-boundedness criterion (CBC) and the total variation diminishing (TVD) property, which are essential to ensure a bounded and monotonic scheme [70]. According to Waterson and Deconinck [69], this method belongs to the class of nonlinear monotonic schemes, which attempts to overcome Godunov's first-order barrier theorem while maintaining the monotonicity of the convected variables [29,71].

Following the continuum surface tension model in Brackbill et al. [72], the cell-centered surface tension force $\mathbf{f}_s$ in Eqs. (43) and (19) is defined as

$$\mathbf{f}_s = \sigma \kappa \nabla H = -\sigma \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \nabla H, \tag{27}$$



(a) CUI stencil for ψ_e ($u_e \geq 0$)

(b) CUI stencil for ψ_e ($u_e < 0$)

(c) CUI stencil for ψ_n ($v_n \geq 0$)

(d) CUI stencil for ψ_n ($v_n < 0$)

Fig. 3. In each diagram, the green dashed line represents the control volume over which the convective term $\nabla \cdot (\mathbf{u}_{\text{adv}}\psi_{\text{lim}})_{i,j}$ is computed. The upwind, centered, and downwind variables are labeled as ψ_U, ψ_C, and ψ_D, respectively. (a) Required CUI stencil to compute ψ_e when $u_e \geq 0$. (b) Required CUI stencil to compute ψ_e when $u_e < 0$. (c) Required CUI stencil to compute ψ_n when $v_n \geq 0$. (d) Required CUI stencil to compute ψ_n when $v_n < 0$.

where σ is the surface tension coefficient and H is the smoothed Heaviside function defined in Eq. (42). This discretization yields a good balance between the surface tension force and the pressure gradient, as indicated by [29,72].

4.2. Multilevel advancement

This section first extends the single-level advancement algorithm to the multilevel advancement algorithm using the subcycling and non-subcycling methods in Section 4.2.1. We then introduce synchronization to make the solutions consistent among different levels (Section 4.2.2). At last, a summary of the multilevel advancement algorithm is given in Section 4.2.3.

4.2.1. Subcycling and non-subcycling methods

Two cycling methods are applied to advance the variables on the multilevel grid. The first is the subcycling method, in which solutions on different levels advance with different time-step sizes. We assume that the CFL number is kept the same among different levels and that the refining ratio r between two consecutive levels [25,55] is a constant. Then, the time step sizes on levels l and $l+1$ satisfy $\Delta t^l = r\Delta t^{l+1}$ for all $0 \leq l < l_{max}$. As shown in the left part of Fig. 4, we have $\Delta t^0 = r\Delta t^1 = r^2\Delta t^2 = \ldots = r^{n-1}\Delta t^{n-1}$ while using the subcycling method on a multilevel grid with $l_{max} = n-1$. Another cycling method is the non-subcycling method, in which all levels have the same time step size as the finest level to avoid instability. As shown in the right part of Fig. 4, we have $\Delta t^0 = \Delta t^1 = \Delta t^2 = \ldots = \Delta t^{n-1}$. To advance the solution from t^n to t^{n+1}, the subcycling method is more efficient than the non-subcycling method because the former takes fewer substeps. On the other hand, the subcycling method needs time interpolation because of the mismatch of the time among different levels [47]. The non-subcycling method does not need this time interpolation because all levels are at the same time instant.

To compare the computational cost between the subcycling and non-subcycling methods, a manufactured case is set up here. As shown in Fig. 5 the domain size is $L_x \times L_y = 1 \times 1$ and the grid number is $N(r/\beta)^l \times N(r/\beta)^l$ on level l, in which β is the shrinking ratio of the domain length between two consecutive levels. Let us assume that the solution needs to be advanced from $t = 0$ to $t = T$ on a multilevel grid, in which $T = n_{\text{step}}\Delta t_0$ and Δt_0 is the time step size on level 0 for the subcycling method. We also assume that only a single CPU is used in this case. For level l, the computational cost C_l, dominated by the solving process of the PDEs [41,47,49], is proportional to the grid number $N(r/\beta)^l \times N(r/\beta)^l$, i.e., $C_l \propto N^2(r/\beta)^{2l}$. For the subcycling method, it takes r^l substeps for the solution on level l to advance from 0 to Δt_0. Thus, the total computational cost in $[0, T]$ for the subcycling method on all levels is

$$C_s = n_{\text{step}} \sum_{l=0}^{n-1} r^l N^2 (r/\beta)^{2l} = \begin{cases} n_{\text{step}} N^2 n, & \beta = r^{3/2}, \\ n_{\text{step}} N^2 \beta^2 \frac{1-(r^3/\beta^2)^n}{\beta^2 - r^3}, & \text{otherwise.} \end{cases} \tag{28}$$

For the non-subcycling method, all levels take r^{n-1} substeps to advance the solution from 0 to Δt_0. Thus, the total computational cost in $[0, T]$ for the non-subcycling method on all levels is

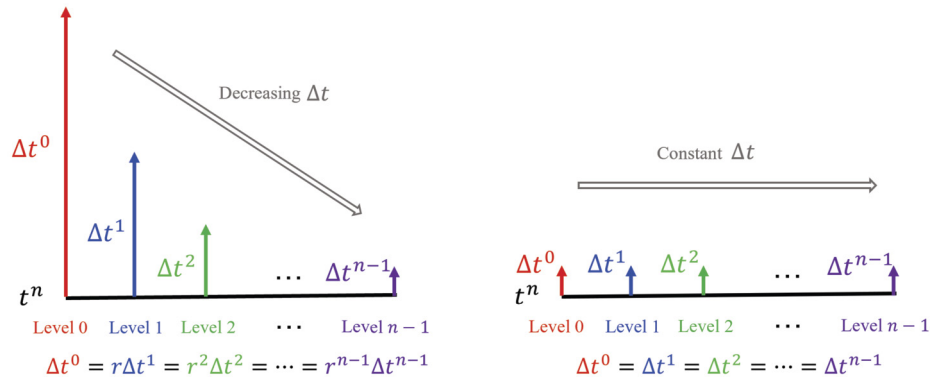


Fig. 4. Schematic of the substeps in the level-by-level advancement method for a n-level grid ($l_{max} = n - 1$). Left: the subcycling method. Right: the non-subcycling method. Parameter r is the refining ratio between two consecutive levels.

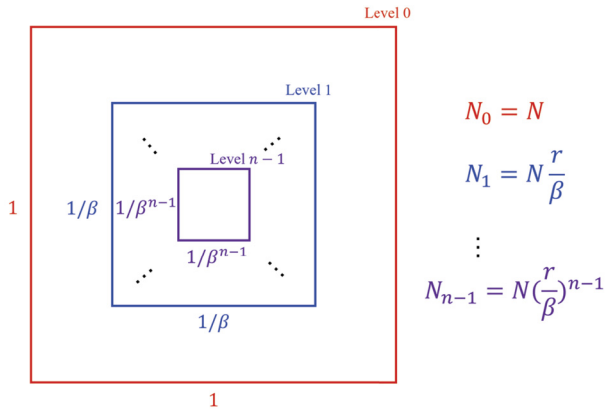


Fig. 5. Schematic of a manufactured case in the level-by-level advancement method for a n-level grid ($l_{max} = n - 1$). On level l ($0 \le l \le l_{max}$), the domain size is $1/\beta^l \times 1/\beta^l$, and the grid number is $N(r/\beta)^l \times N(r/\beta)^l$.

$$C_{ns} = n_{step}\sum_{l=0}^{n-1} r^{n-1}N^2(r/\beta)^{2l} = \begin{cases} n_{step}N^2 n r^{n-1}, & \beta = r, \\ n_{step}N^2 r^{n-1}\beta^2\frac{1-(r^2/\beta^2)^n}{\beta^2 - r^2}, & \text{otherwise.} \end{cases} \tag{29}$$

Based on Eqs. (28) and (29), the speedup, defined as $\gamma = C_{ns}/C_s$, can be calculated as

$$\gamma = \begin{cases} \frac{nr^{n-1}(r-1)}{r^n - 1}, & \beta = r, \\ \frac{r^n - 1}{n(r-1)}, & \beta = r^{3/2}, \\ \frac{r^{n-1}[1-(r^2/\beta^2)^n](\beta^2 - r^3)}{(\beta^2 - r^2)[1-(r^3/\beta^2)^n]}, & \text{otherwise.} \end{cases} \tag{30}$$

Below are several remarks about Eqs. (28)–(30).

- For the single-level case ($n = 1$), there is no speedup ($\gamma = 1$). For the multilevel case ($n > 1$), the speedup $\gamma > 1$ if and only if the refining ratio $r > 1$. Because $r > 1$ is expected in the simulations with AMR [42,43,47,73], the subcycling method is more efficient than the non-subcycling method for the multilevel case.
- When $\beta = r^{3/2}$, the total computational cost on each level is the same in the subcycling method. There are fewer grid cells on the finer levels than on the coarser levels, while coarser-level solutions take fewer substeps for the time advancement.
- When $\beta = r$, the computational cost on each level is the same in the non-subcycling method. The number of grid cells and the substeps for time advancement are the same for each level.
- The contours in Fig. 6 illustrate how the speedup γ varies with the total number of levels n and the refining ratio r when $\beta = r$ or $\beta = r^{3/2}$. The speedup γ increases when n or r increases, which is also true for the cases in which $\beta \neq r$ and $\beta \neq r^{3/2}$.

We note that the above formulas are limited to one CPU core. In practical two-phase flow simulations, more CPUs are utilized in parallel computing. The communication cost among different CPUs and the load-balancing problem are thus

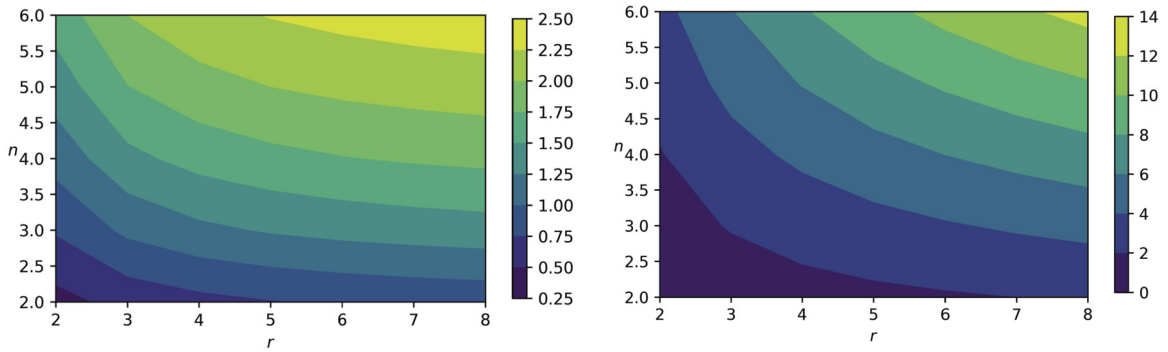


Fig. 6. Contours of the log value of the speedup γ, which varies with the total number of levels n and the refining ratio r. Left: the shrink ratio β and the refining ratio r satisfy $\beta = r$. Right: the shrink ratio β and the refining ratio r satisfy $\beta = r^{3/2}$.

involved [49,60], which makes it difficult to derive a general formula for the speedup γ. At the end of Section 5, a dynamic and complex 3D Stokes wave problem is simulated using multiple CPUs to compare the computational cost among the single-level and multilevel subcycling and non-subcycling cases.

4.2.2. Synchronization

Synchronization is applied to update the data on multiple levels to make them consistent and better represent the composite solution. The synchronization step is necessary for both the subcycling and the non-subcycling methods [47–49]. There are three substeps of the synchronization step. The first substep is conservative averaging, in which all cell-centered variables (e.g., velocity $\boldsymbol{u}$, pressure p, and LS function ϕ) on the coarser levels are replaced by the averaged values from the finer levels because the finer solutions with the mesh refinement are assumed to be more accurate [47,48]. The second substep is refluxing. Because level-by-level time advancement is used in this work, the momentum fluxes are imbalanced at the CF boundaries. We collect these fluxes during the single-level time advancement and redistribute them on the coarser and finer levels at the reflux substep. As found by [25,47], refluxing is necessary and helpful for mass conservation on the multilevel grid. The last substep is synchronization projection, which is used to obtain the divergence-free velocity on all levels. This substep is needed because the level projection used in the single-level time advancement can guarantee divergence-free velocity on that level only, not all levels. The synchronization projection is applied to the velocities on all levels and updates them simultaneously [48,49].

The stability of the projection operation is an important issue. For the non-graded TBAMR, the stability of the discrete projection can be affected by the presence of the large size ratio of the adjacent cells. Min and Gibou [44] proposed a different projection formulation to enforce the orthogonality of the projection. The stability of the synchronization projection method used in the present work has been validated by Martin and Colella [48] using a three-vortex problem. The robustness and stability of the approximate projection are related to the careful interpolations of the ghost cell values [66,74]. The projection is applied to the updated velocity at the new time ($\boldsymbol{u}^{n+1}$) rather than the increment velocity ($\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}$), which helps to stabilize the synchronization projection as found by Martin and Colella [48].

4.2.3. Summary of multilevel advancement

Algorithm 1 summarizes the unified multilevel advancement algorithm for both the subcycling and non-subcycling methods. We first initialize the velocity $\boldsymbol{u}$, pressure p, and LS function ϕ on all levels at the beginning of the simulation based on some refinement criteria. After the initialization, we can use either the subcycling or the non-subcycling method for time advancement. We can also choose either the inconsistent or the consistent scheme for spatial discretization. For the solver part, only the cell-centered multigrid solver is needed because all unknown variables are defined at the cell centers. The synchronization operations are then applied when a coarser level catches up with a finer level. Finally, the grid is either refined or derefined before moving to the next step.

To summarize, the single-level consistent scheme is extended to the multilevel grid so that the multilevel advancement algorithm can accurately simulate the high-density-ratio and high-Reynolds-number two-phase flows using AMR. We emphasize that our multilevel advancement algorithm is a level-by-level advancement method, which differs from the composite advancement method [17,29,39,75]. The level variables are used for time advancement in the level-by-level advancement method. Before the synchronization step, each level can be advanced individually without taking into account the finer levels. The time step restrictions on the coarser levels are alleviated because the time advancements at various levels are decoupled. This is in contrast to the composite advancement technique, in which the multilevel multigrid (MLMG) solver is used to update the velocity and pressure in the valid areas of all levels at the same time. Because of this different treatment, the composite advancement method is not flexible enough to easily integrate both the subcycling and the non-subcycling methods, while the level-by-level method used in this study can easily manage both.

Algorithm 1 Multilevel advancement algorithm.

Initialize $\boldsymbol{u}^0$, ϕ^0, and p^0 on level 0
$l \leftarrow 0$
while refinement criteria are satisfied on level l and $l < l_{max}$ **do**
    Regrid the patch hierarchy to obtain level $l+1$
    Initialize $\boldsymbol{u}^0$, ϕ^0, and p^0 on level $l+1$
    $l \leftarrow l+1$
end while
Determine the time step $\Delta t^{l_{max}}$ on the finest level
if subcycling method is used **then**
    $\Delta t^l = 2^{l_{max}-l}\Delta t^{l_{max}}$ for all $0 \le l < l_{max}$
else
    $\Delta t^l = \Delta t^{l_{max}}$ for all $0 \le l < l_{max}$
end if
for $n = 1, n_{max}$ **do** ▷ n_{max} is the number of time steps to be simulated
    LevelCycling(0, t_n^0, $t_n^0 + \Delta t^0$, Δt^0)
    Regrid the patch hierarchy, and interpolate $\boldsymbol{u}$, ϕ, and p onto new patches
end for

procedure LevelCycling(l, t^l, t^l_{max}, Δt^l)
    while $t^l < t^l_{max}$ **do**
        if consistent scheme is used **then**
            Perform single-level advancement on level l from t^l to $t^l + \Delta t^l$ using Eqs. (10)–(23)
        else
            Perform single-level advancement on level l from t^l to $t^l + \Delta t^l$ using Eqs. (10)–(13), Eqs. (40)–(43), and Eqs. (21)–(23)
        end if
        if $l < l_{max}$ **then**
            LevelCycling($l+1$, t^l, $t^l + \Delta t^l$, Δt^{l+1})
        end if
        $t^l \leftarrow t^l + \Delta t^l$
    end while
    Apply the synchronization step
end procedure

5. Results

This section presents several multiphase flow problems to test the convergence, consistency, and stability of the proposed schemes within the BSAMR framework from various aspects. The consistent and inconsistent schemes are compared to show the importance and necessity of using the consistent scheme in the high-density-ratio and high-Reynolds-number two-phase flow problems. In these problems, either the subcycling or non-subcycling methods are employed for time advancement when the multilevel grid is involved. We first define some common parameters here. For each problem, Δt_0 denotes the time step on level 0. We use Δx_0, Δy_0, and Δz_0 to represent the grid spacings in the x-, y-, and z-directions, respectively, on level 0. For the multilevel grid, the grid spacings on level l satisfy $\Delta x_l = \Delta x_0/2^l$, $\Delta y_l = \Delta y_0/2^l$, and $\Delta z_l = \Delta z_0/2^l$ for all $0 \le l \le l_{max}$. For the time step size, the finest level $\Delta t^{l_{max}}$ is first determined by restrictions of the CFL condition, gravity, viscosity, and surface tension [17,25,76,77]. The time step sizes on the coarser levels, depending on whether the subcycling or the non-subcycling method is used, are then calculated based on Algorithm 1.

5.1. 2D reversed single vortex problem

In this section, we test the order of convergence for the consistent scheme in a 2D reversed single vortex problem [78, 79] on the multilevel grid. In this problem, a 2D circular drop with radius $R = 0.15$ is placed at $(0.5, 0.75)$ in a unit computational domain $[0,1] \times [0,1]$. The velocity field is given by the stream function,

$$\Psi(x, y, t) = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)\cos\left(\frac{\pi t}{T}\right), \tag{31}$$

in which the rotational period is $T = 4.0$. The velocities in the x and y directions are defined as $u(x, y, t) = \partial\Psi/\partial y$ and $v(x, y, t) = -\partial\Psi/\partial x$, respectively. The periodic boundary condition is applied in both x and y directions.

To obtain the pointwise convergence rate on the multilevel grid, we consider four cases here, of which the grid number $N_x \times N_y$ on level 0 is 16×16, 32×32, 64×64, and 128×128. The finest level l_{max} is 2, and the refinement criterion is based on the distance to the drop interface, i.e., grid cells (i, j) on level l $(0 \le l < l_{max})$ are refined to the finer level if $|\phi_{i,j}| < 8.0\max(\Delta x_l, \Delta y_l)$, which ensures that the surface of the drop is always placed on the finest level. The upper part of Fig. 7 shows the state of the drop at $t = T/2$ and $t = T$ for two different resolutions using the subcycling method. As we increase the resolution, the interface at the tail of the stretching vortex becomes sharper. At $t = T$, the interfacial profile of the drop at the high resolution converges to its initial profile, which represents the exact solution. On the other hand,

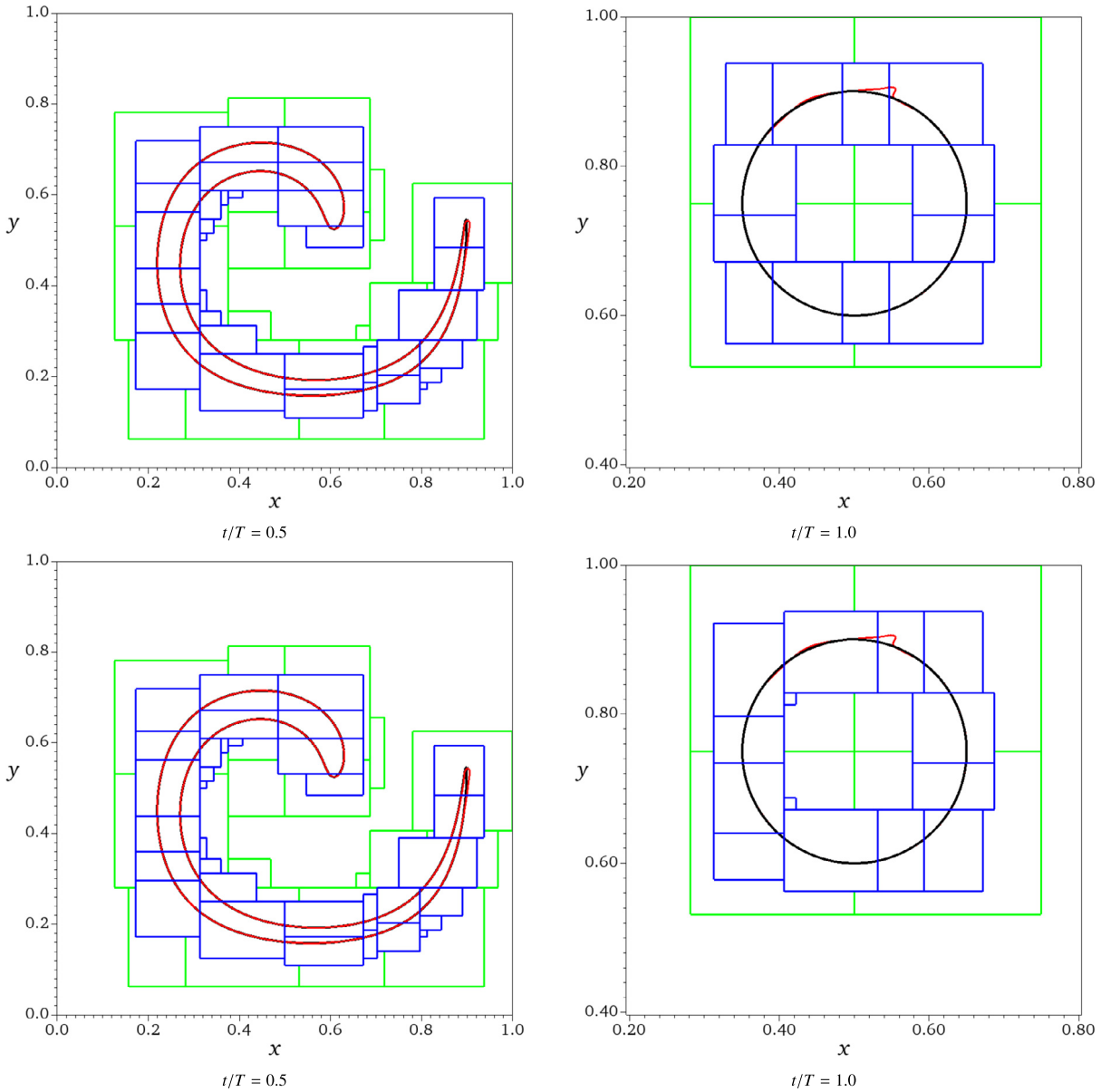


Fig. 7. Profiles of the drop interface in the reverse single vortex problem using the subcycling method (upper) and non-subcycling method (lower) at different time instants. The red and black lines represent the drop interface with the coarsest grid numbers 32×32 and 64×64, respectively. The green and blue lines represent patches on levels 1 and 2, respectively.

the circular shape of the drop is distorted for the simulation with low resolution. The lower part of Fig. 7 shows the grid hierarchy and the shape of the drop using the non-subcycling method. The subcycling and non-subcycling methods produce consistent and accurate results in this simulation. The error of the results is defined as [78,79]

$$E_{\text{shape}} = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |\phi(i, j, t = T) - \phi(i, j, t = 0)| \Delta x \Delta y, \tag{32}$$

where $\phi(i, j, t = 0)$ is the exact reference solution. The corresponding rates of convergence are given in Fig. 8. The errors decrease with an approximately second-order convergence rate for both the subcycling and the non-subcycling methods. These errors are also comparable with the values reported by [79].

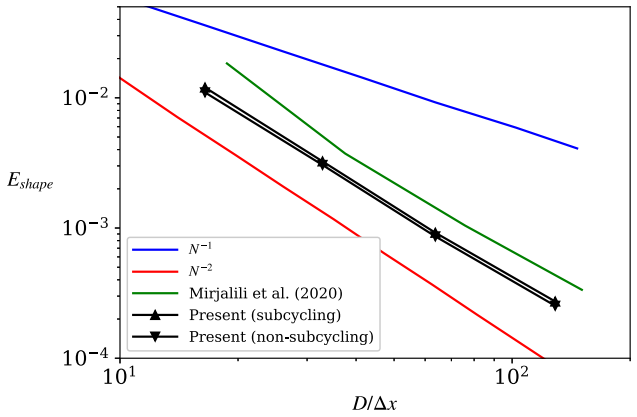


Fig. 8. Spatial–temporal convergence for the 2D reversed single-vortex problem on a multilevel grid.

5.2. 2D convected droplet with a high density ratio

The convected droplet is a canonical problem to demonstrate the importance of consistent mass and momentum transport in flows with a high density ratio. We first consider a single-level 2D test by employing both the consistent and the inconsistent schemes. A dense droplet is placed in a periodic computational domain with size $[0,1]\times[0,1]$. The initial center position of the droplet is $(X_0, Y_0) = (0.25, 0.5)$, and the diameter of the droplet is $D = 0.4$. The density ratio between the inner and outer parts of the droplet is $\rho_i/\rho_o = 10^6$, where $\rho_o = 1.0$. The viscosity, gravity, and surface tension force are all set to zero. The vertical velocity is initialized as zero, and the horizontal velocity is $u_{i,j} = 1 - \widetilde{H}^f_{i,j}$, where $\widetilde{H}^f_{i,j}$ is a smoothed Heaviside function defined as

$$\widetilde{H}^f_{i,j} = \begin{cases} 1, & \phi_{i,j} < -\epsilon, \\ \frac{1}{2}\left(1 + \frac{1}{\epsilon}\phi_{i,j} + \frac{1}{\pi}\sin\left(\frac{\pi}{\epsilon}\phi_{i,j}\right)\right), & \left|\phi_{i,j}\right| \le \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

Here, $\phi(\mathbf{x}, 0) = D/2 - \sqrt{(x - X_0)^2 + (y - Y_0)^2}$ is the initial LS function, and $\epsilon = \min(\Delta x_0, \Delta y_0)$ is the smearing width [29]. After initialization, the level projection is applied to generate a divergence-free initial velocity field [30]. This problem has been investigated by Bussmann et al. [27], Desjardins and Moureau [36], Ghods and Herrmann [37], Patel and Natarajan [38] and Nangia et al. [29]. In the following 2D cases, the grid size is discretized as $N_x \times N_y = 128 \times 128$. Each case is run until $t = 0.5$ with a constant time step size $\Delta t = 1/(31.25N_x)$ [29].

Because of the high density contrast between the droplet and the outer fluid, it is expected that the droplet moves at a constant velocity and maintains its circular shape. As shown in the lower part of Fig. 9, the simulations are stable when using the consistent mass and momentum transport scheme. For the cases using the inconsistent scheme, as plotted in the upper part of Fig. 9, the surface distortion of the original droplet is clearly seen, and small spurious droplets are then generated when the simulation becomes unstable.

5.3. 2D dam breaking

This section investigates a dynamic 2D dam-breaking problem to further validate the robustness and necessity of the consistent scheme when the multilevel grid is involved. As shown in Fig. 10, a square block with side length $a = 0.057$ m is placed at the left-bottom corner, and the computational domain size is $7a \times 1.75a$. A no-slip boundary condition is imposed on the bottom wall, while all other walls are considered free-slip boundaries [80,81]. The dimensionless front is defined as $\tilde{d}_f = d_f/a$, in which d_f refers to the dimensional distance between the front position (point A in Fig. 10) and the origin. Other dimensionless parameters are set as $Re = \rho_l U a/\mu_l = 2950$, $Fr = U/\sqrt{ga} = 1.0$, and $We = \rho_l U^2 a/\sigma = 0.54$. The density ratio and viscosity ratio are set as $\rho_g/\rho_l = 0.0012$ and $\mu_g/\mu_l = 0.016$, respectively [29]. Table 1 gives the parameters of four simulation cases. The refinement criterion is based on the distance to the air–water interface, the same as the 2D reversed vortex problem in Section 5.1.

Fig. 11 compares the dimensionless front $\tilde{d}_f$ of the single-level cases (Cases 1 and 2) and the two-level cases (Cases 3 and 4) with previous experimental and numerical results [29,80,82]. The results of the cases that use the consistent scheme (Cases 1 and 3) agree well with the literature. For cases using the inconsistent scheme (Cases 2 and 4), the magnitude of the dimensionless front $\tilde{d}_f$ is underestimated. As shown in Fig. 11, $\tilde{d}_f$ decreases and then increases at approximately $t/T = 2.4$. This phenomenon is also obtained in [81] because the liquid part of the dam front becomes unstable and breaks into small droplets in the inconsistent scheme cases.

To further compare the differences of the results between the consistent scheme and the inconsistent scheme, the time evolution of the normalized velocity amplitude and normalized fluid mass are plotted in Fig. 12. Here, the normalized

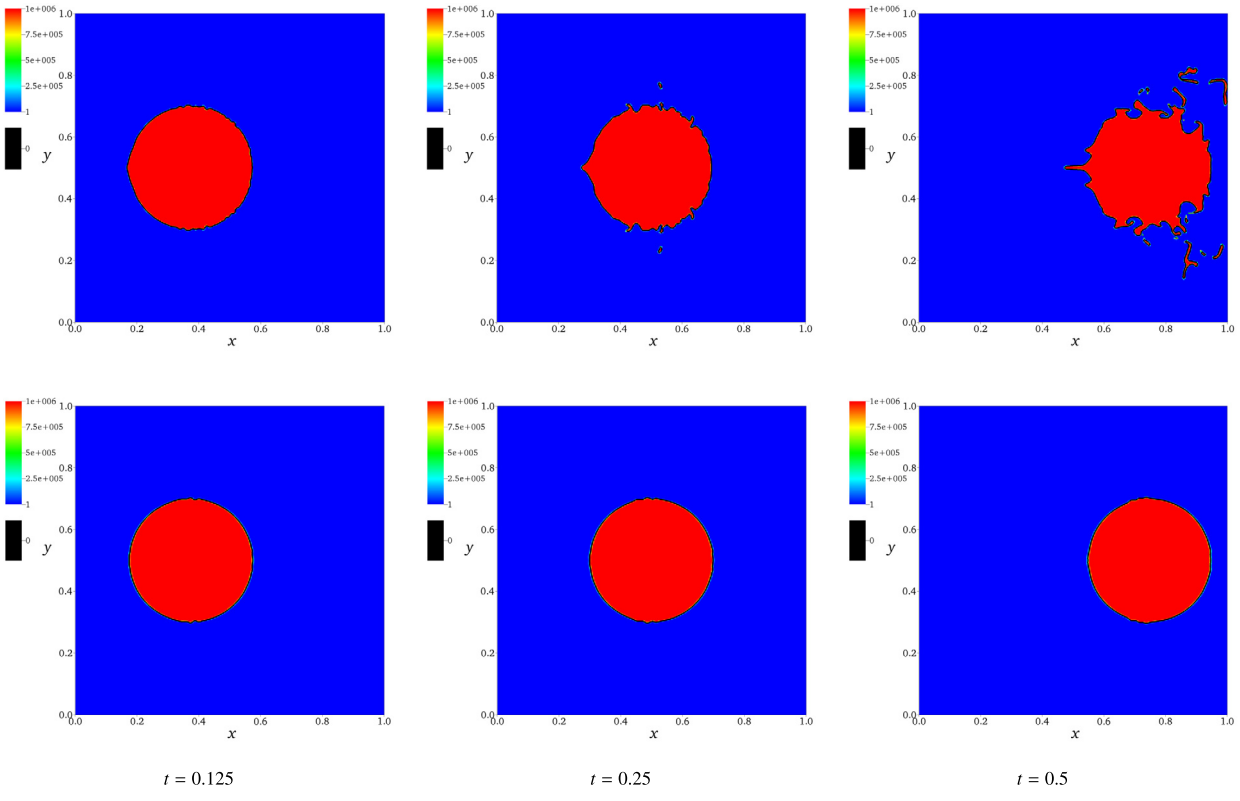


Fig. 9. Geometry and density evolution of the 2D convected droplet problem. The upper and lower parts show the results of the inconsistent and consistent schemes, respectively.

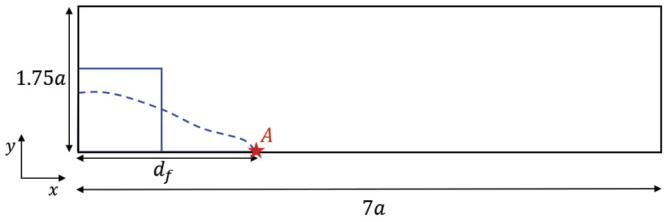


Fig. 10. Sketch of the 2D dam-breaking problem. Solid line: dam interface at $t = 0$. Dashed line: dam interface at $t > 0$. A: dam front position at $t > 0$.

Table 1
Parameters for cases of the dam-breaking problem.

Case No.	Grid number on level 0	l_{max}	Δt^0	Scheme	Cycling method
1	512×128	0	1.25×10^{-4}	Consistent	–
2	512×128	0	1.25×10^{-4}	Inconsistent	–
3	256×64	1	2.50×10^{-4}	Consistent	Subcycling
4	256×64	1	2.50×10^{-4}	Inconsistent	Subcycling

velocity amplitude is defined as the maximum value of $||\mathbf{u}(t)||/||\mathbf{u}(0)||$, where $||\mathbf{u}|| = \sqrt{u^2 + v^2}$ is the L^2 norm of the velocity vector. The normalized fluid mass is defined as $||m(t)||/||m(0)||$, where $m = \int \rho \, dV$ is the total fluid mass. As shown in the left part of Fig. 12, the normalized velocity amplitude agrees well between Cases 3 and 4 when $t/T < 1.0$. Over time, the normalized velocity in the case with the inconsistent scheme becomes unreasonably large, and the simulation quickly becomes unstable. Compared with the consistent scheme, the instability induced by the inconsistent scheme also affects the conservation of the fluid mass. Although the same LS reinitialization method is used for both Cases 3 and 4, the right part of Fig. 12 shows that the mass loss of Case 3 is less than 2%, while the simulation of Case 4 quickly becomes unstable.

Next, the evolution of the breaking dam for the two-level consistent scheme case (Case 3) and the two-level inconsistent scheme case (Case 4) is depicted in Fig. 13, in which patches are dynamically refined around the interface as time evolves. The left part of Fig. 13 shows that the results of the consistent case remain stable and produce a physically reasonable

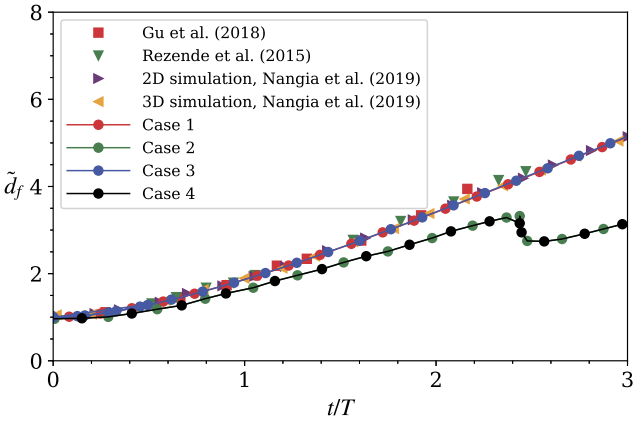


Fig. 11. Comparison of the evolution of the dimensionless front $\tilde{d}_f$ among the single-level cases (Cases 1 and 2), the two-level subcycling cases (Cases 3 and 4), and the literature: (■) Gu et al. [82]; (▼) Rezende et al. [80]; (▶, ◀) Nangia et al. [29]. Cases 1 and 3 use the consistent scheme, while Cases 2 and 4 use the inconsistent scheme.

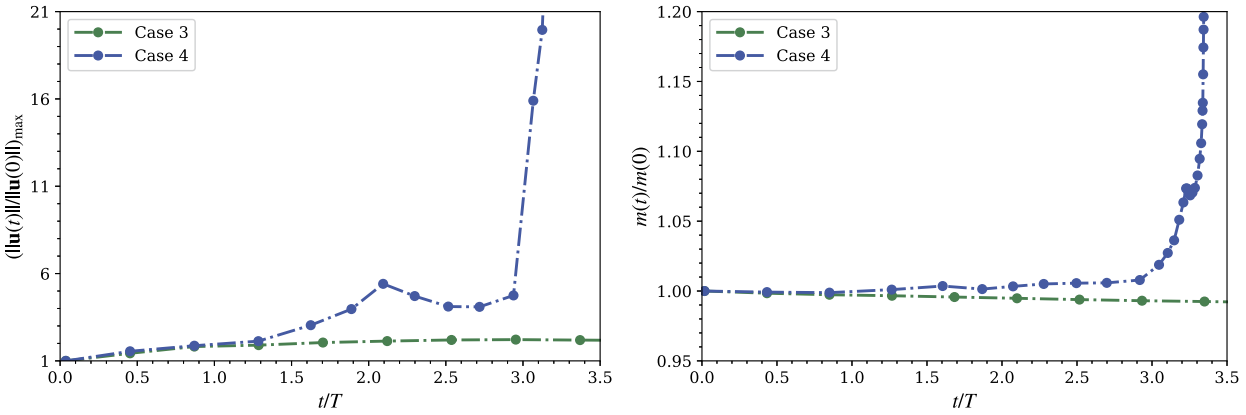


Fig. 12. Comparison of the time evolution of the normalized velocity amplitude (left) and the normalized fluid mass (right) between the case using the consistent scheme (Case 3) and the case using the inconsistent scheme (Case 4) in the 2D dam-breaking problem.

Table 2
Parameters for cases of a 2D droplet splashing on a thin film.

Case No.	Grid number on level 0	l_{max}	Δt^0	Scheme	Cycling method
1	256×64	3	1.2×10^{-3}	Consistent	Subcycling
2	256×64	3	1.5×10^{-4}	Consistent	Non-subcycling

solution. The shape of the dam also compares favorably to the experimental results [83] and the numerical results [29] (not plotted). On the other hand, in the right part of Fig. 13, unphysical deformations appear because of the dissimilar transport of mass and momentum caused by the inconsistent scheme. We again emphasize that consistent mass and momentum transport are important for stable simulations of this dam breaking problem with a large liquid–gas density ratio.

5.4. 2D droplet splashing on a thin liquid film

In this section, we aim to validate the consistency of the results between the subcycling and non-subcycling methods when a multilevel grid is involved. We consider the problem of droplet splashing on a thin liquid film, which has applications in inkjet printing [84] and spray cooling [85]. The computational domain is $8D \times 2D$ with a no-slip boundary condition at all sides. The droplet is initially placed at $(X_0, Y_0) = (4D, 0.75D)$ with a diameter of $D = 1.0$ and an initial downward velocity of $U = 1.0$. The thin film fills one-tenth of the computational domain with an initial height of $0.2D$ [29]. The density ratio and the viscosity ratio between the liquid (or the thin film) and the surrounding gas are $\rho_l/\rho_g = 815$ and $\mu_l/\mu_g = 55$, respectively. Owing to the high impact velocity, the gravitational force does not play an important role and can be neglected [29,81]. The Reynolds number is $Re = \rho_g UD/\mu_g = 66$, and the Weber number is $We = \rho_g U^2 D/\mu_g = 0.126$.

Two multilevel cases are considered in Table 2. For these multilevel cases, the finest level l_{max} is 3, and the refinement criterion is based on the distance to the gas–liquid interface; i.e., grid cells (i, j, k) on level l $(0 \le l < l_{max})$ are refined to the

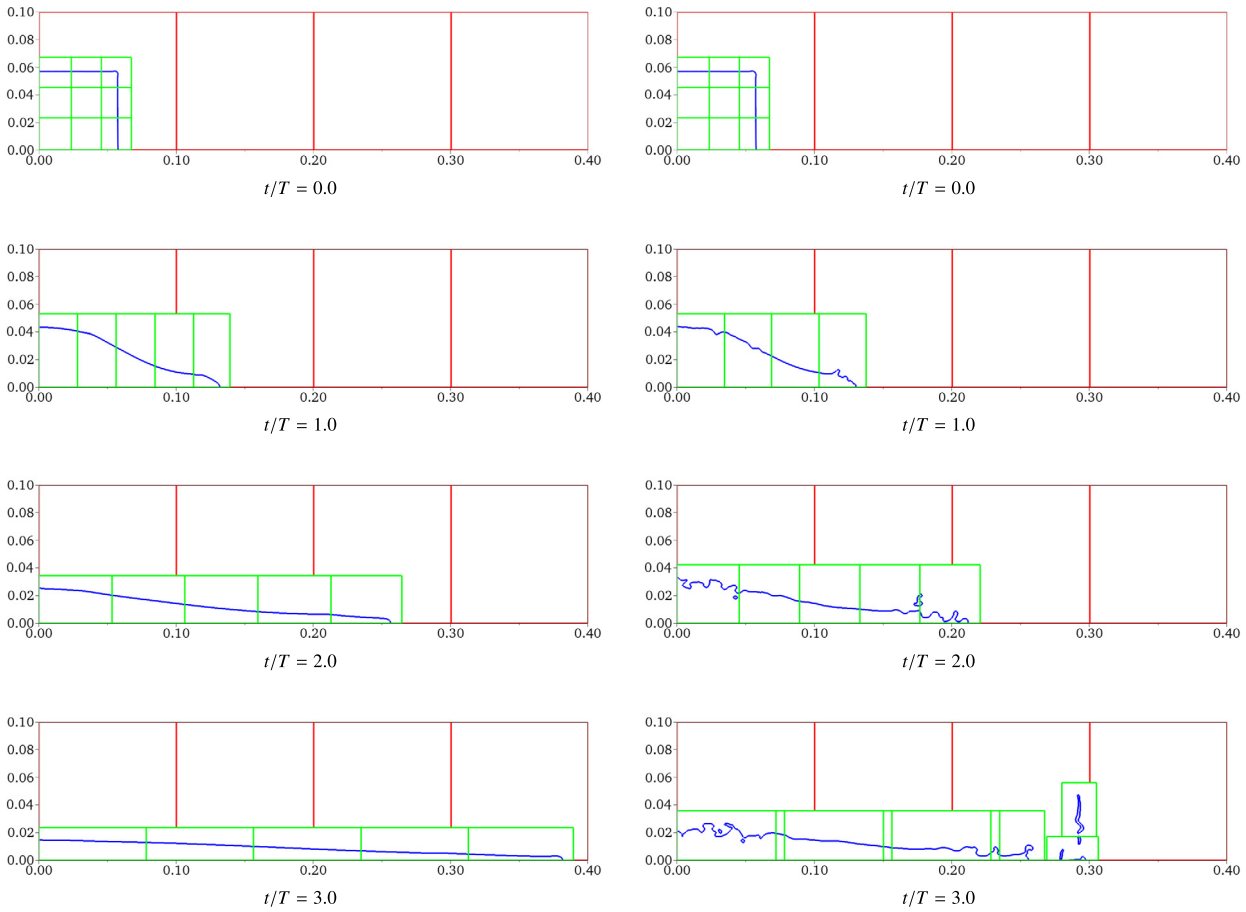


Fig. 13. Profiles of water surface in the breaking dam for the two-level subcycling cases using the consistent scheme (left) and inconsistent scheme (right) at different time instants. The red and green lines represent patches on levels 0 and 1, respectively.

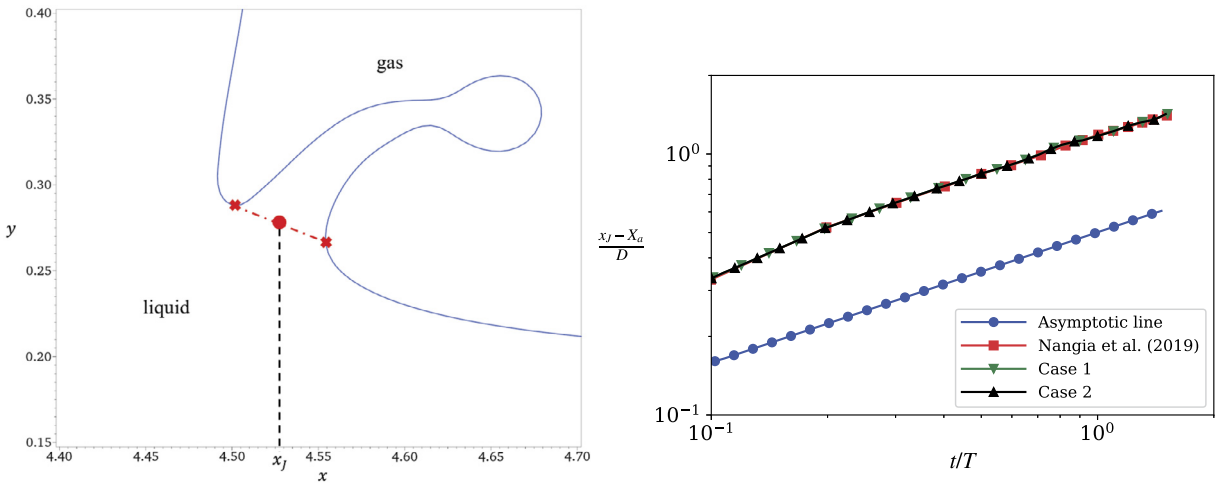


Fig. 14. Left: sketch of the jet base location x_J; right: comparison of the temporal evolution of the dimensionless jet base location among the four-level subcycling case (Case 1), four-level non-subcycling case (Case 2), Nangia et al. [29], and the asymptotic results from theoretical analysis [86–88].

finer level if $|\phi_{i,j,k}| < 8.0\max(\Delta x_l, \Delta y_l, \Delta z_l)$, which ensures that the thin film and the splashing droplet are always placed on the finest level. Simulations of these two cases are carried out until $t/T = 1.5$.

The left part of Fig. 14 shows the jet base location x_J, which is the midpoint of the two neck points of the splashing sheet. The right part shows the temporal evolution of the dimensionless spread factor x_J/D, which follows a power law

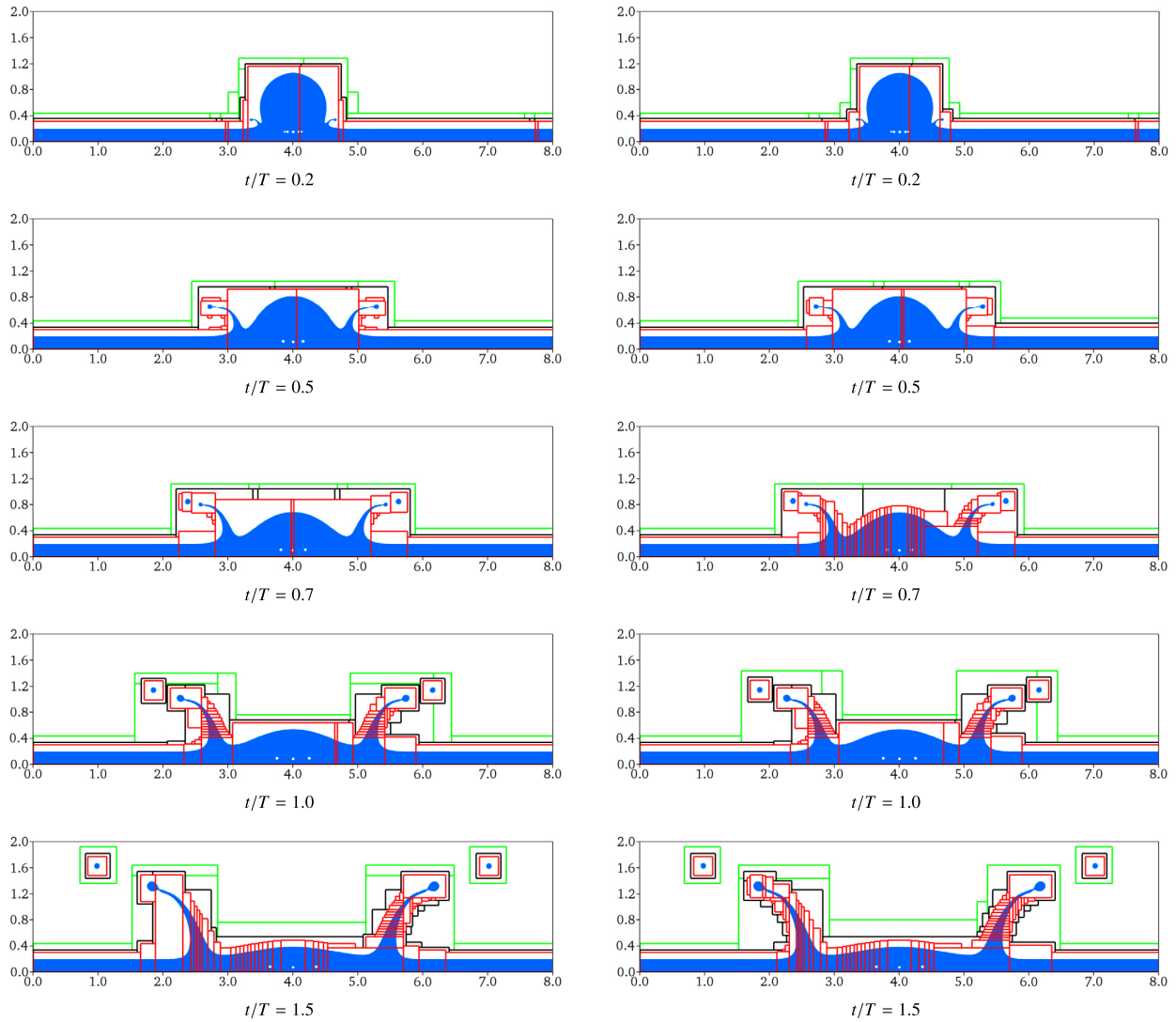


Fig. 15. Grid hierarchy and evolution of droplet splashing on a thin liquid film at five different time instances. Left: subcycling method; right: non-subcycling method. Green patches: level 1; black patches: level 2; red patches: level 3.

defined by $x_J/D \propto (Ut/D)^{1/2}$ [86–88], where U is the characteristic velocity. Our simulation results agree with Nangia et al. [29]. The slope is consistent with the power law, and the results of the subcycling case (Case 1) and non-subcycling case (Case 2) are similar. Fig. 15 plots the time evolution of the splashing at different time instances for both the subcycling and the non-subcycling methods. Although the grid patches for different cycling methods are different, the shapes of the thin film and the splashing droplet are almost identical between these two cycling methods. We note that this problem was also simulated in [89], where a lattice Boltzmann method was used, and in [81], where a VOF method was applied. Although using a different method, our simulations agree favorably with those results.

In summary, the proposed framework produces consistent results between the subcycling and non-subcycling methods when multiple levels of grids are involved in the simulation. The surface tension and droplet splashing dynamics can be accurately captured using the consistent scheme.

5.5. 2D rising bubble

In our previous work [25], we validated the inconsistent scheme by comparing the simulations of a rising bubble at low Reynolds number with the experimental results [90]. In this section, we use our consistent scheme with AMR to simulate the rising bubble. A large computational domain $[-3.3D, 3.3D] \times [-3.2D, 8.0D]$ is chosen to circumvent wall effects [91], in which $D = 1.0$ is the bubble diameter. Free-slip conditions are applied at all boundaries. A spherical bubble of dimensionless radius one is placed at $(x, y) = (0, 0)$ surrounded by stationary fluid as the initial condition. The key

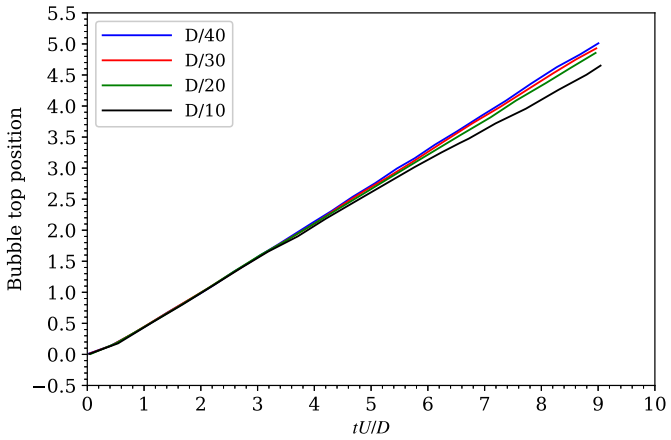


Fig. 16. Time series of the bubble top position under different grid sizes for the rising bubble case with $Re = 18.3$, $Eo = 339$, and $Mo = 43.1$.

Table 3
Comparison of the computed Reynolds number with the experimental result ($Re = 18.3$).

Grid spacing	Re	Relative error (%)
$D/10$	15.25	16.67
$D/20$	17.11	6.50
$D/30$	17.63	3.66
$D/40$	17.92	2.08

dimensionless parameters in this simulation include the Reynolds number $Re = \rho_w DU/\mu_w$, Eotvos number $Eo = \rho_w g D^2/\sigma$, and Morton number $Eo = \mu_w^4 g/(\rho_w \sigma^3)$, where ρ_w is the liquid density, μ_w is the liquid viscosity, U is the bubble terminal velocity, g is the acceleration of gravity, and σ is the surface tension coefficient.

To perform a grid refinement study, we first simulate the rising bubble problem on a uniform single-level grid. The dimensionless parameters are set as $Re = 18.3$, $Eo = 339$, and $Mo = 43.1$, which are the same as those of the skirted bubble in Sussman et al. [34]. The grid spacing in the x and y directions is set as $D/10$, $D/20$, $D/30$, and $D/40$ for the coarse, medium, fine, and finer grid, respectively. The evolution of the bubble top position is plotted in Fig. 16 with different grid sizes. The computed terminal velocities in terms of the Reynolds number Re are shown in Table 3. Our results are close to the experimental results in [92] and the numerical results ($Re = 17.74$) in [91]. Because the relative error of the case with the grid spacing $D/30$ is already less than 4%, we use it as our finest resolution when performing the simulations on the multi-level grid with AMR next.

For the multi-level grid simulation, the subcycling method is used. The finest level l_{max} is 2 and the refinement criterion is based on the distance to the gas–liquid interface. Besides the skirted bubble, the oblate ellipsoidal cap bubble ($Re = 7.77$, $Eo = 243$, and $Mo = 266$) is also considered in the simulation. The computed terminal bubble shapes are shown in Fig. 17, which agree well with the experimental benchmark results (Fig. 2, bubble (d) and bubble (g)) in [92] and the numerical results (Fig. 17 (c)) in [91]. The above result shows that the consistent scheme in the present adaptive framework is stable and robust for the simulation of rising bubbles with different shapes.

5.6. 3D liquid jet

In this section, we compute the growth rate of instability of a co-flowing jet using our consistent scheme. The numerical sketch of this problem is shown in Fig. 18 and the initial coaxial flow is given by [93,94],

$$W_1(r) = -1 + \frac{Nr^2}{N-(1-l^2)}\left\{1 - \frac{1-Q}{4N}R\left[2\ln l + \left(1-l^2\right)\right]\right\}, \tag{34}$$

$$W_2(r) = -\frac{l^2-r^2}{N-(1-l^2)}\left\{1 - \frac{1-Q}{4N}R\left[2\ln l + \left(1-l^2\right)\right]\right\} + \frac{1-Q}{4N}R\left[l^2 - r^2 - 2\ln\left(\frac{l}{r}\right)\right], \tag{35}$$

$$U_1(r) = U_2(r) = V_1(r) = V_2(r) = 0, \tag{36}$$

in which $N = \mu_2/\mu_1$, $l = R_2/R_1$, $Q = \rho_2/\rho_1$, $Re = \rho_1 W_0 R_1/\mu_1$, $Fr = W_0/gR_1$, and $R = Re/Fr$. Here, the subscripts 1 and 2 represent the liquid and gas phase, respectively. The r is the radial distance normalized by R_1, W_0 is the magnitude of the jet velocity on the z-axis, and $W(r)$ is the axial velocity distribution normalized by W_0. The dimensionless parameters N, Q, Re, We, l, and Fr^{-1} are defined case by case.

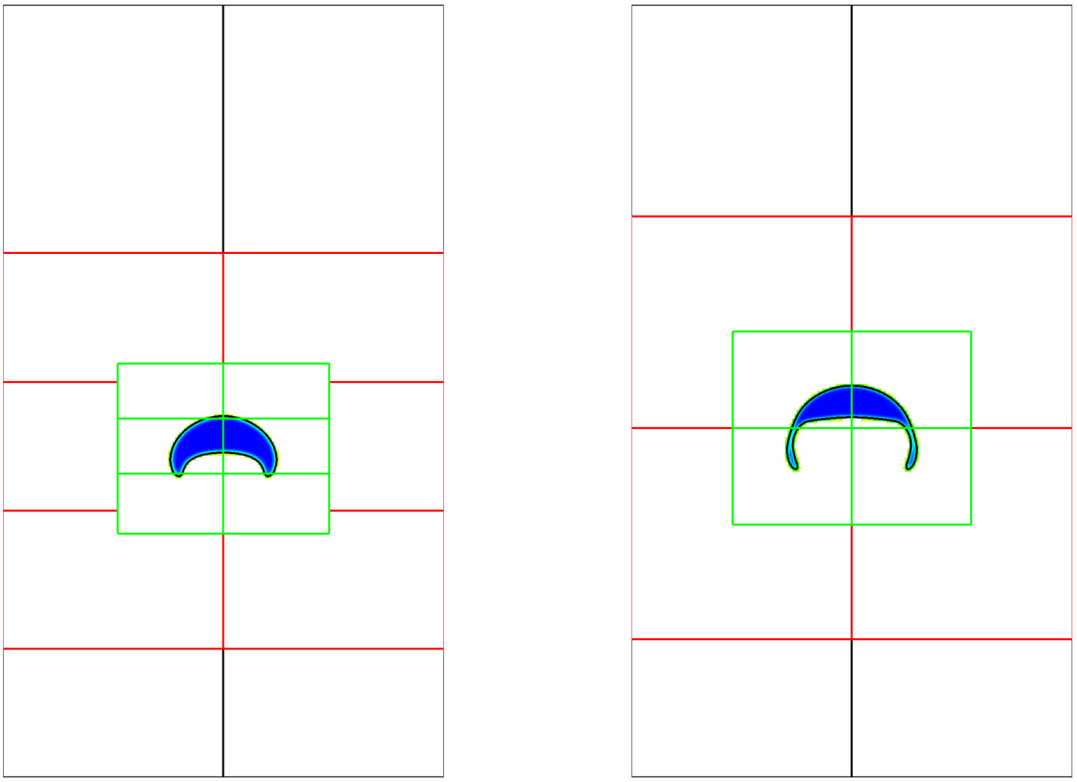


Fig. 17. Computed terminal shapes of the rising bubble. Left: an oblate ellipsoidal bubble with $Re = 7.77$, $Eo = 243$, and $Mo = 266$. Right: a skirted bubble with $Re = 18.3$, $Eo = 339$, and $Mo = 43.1$.

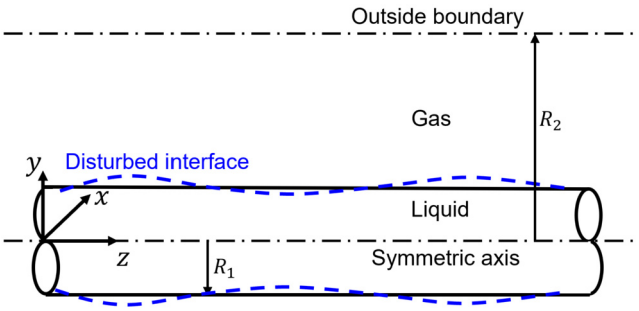


Fig. 18. Sketch of the co-flowing jet model.

The computational domain with a rectangular grid is $(x, y, z) \in [0, 1] \times [0, 1] \times [0, 2]$, same as that in [95]. To perform the grid refinement study, the dimensionless parameters are defined as $N = 0.0018$, $Q = 0.0013$, $Re = 50$, $We = 10$, $l = 10$, and $Fr^{-1} = 0$ [94,96]. The initial level set function is prescribed as

$$\phi(x, y, z) = R_1 + \epsilon \sin(2\pi z/\lambda) - \sqrt{x^2 + y^2}, \tag{37}$$

in which the perturbation of the initial jet is $\epsilon = 0.02$ and the wavelength is $\lambda = 2$. By varying the grid size, the time series of the amplitude of growth rate predicted by the consistent scheme is plotted in Fig. 19. As the grid number increases, it is found that the present results converge to the linear stability analysis (LSA) result in [94–96].

We then compute the growth rate of instability with the realistic density ratio and viscosity ratio, and compare our results with those in Lin and Chen [93]. Due to the limited computational resources, the simulations are conducted on a multi-level grid with the finest grid resolution $128 \times 128 \times 256$ and $l_{max} = 2$. The amplification curves of the Rayleigh mode and the Taylor mode are plotted in Fig. 20. The Rayleigh mode tends to break up the jet into the drop that has the comparable diameter with the jet diameter. In the left part of Fig. 20, Q and We are chosen to satisfy $Q \sim We^{-1}$ to yield the Rayleigh mode [91,93]. The Taylor mode, on the other hand, arises from the pressure and shear fluctuations at the gas-liquid interface, which tends to produce the droplets with diameters smaller than the jet diameter. As shown

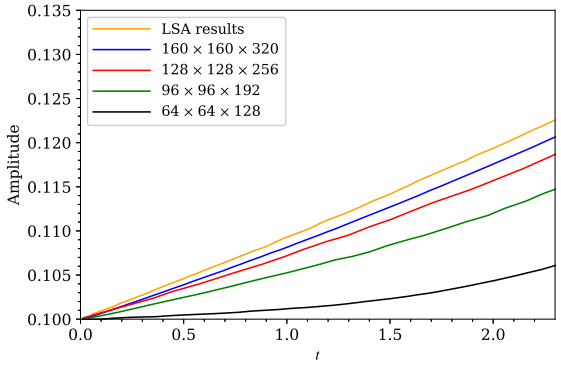


Fig. 19. Comparison of the amplitude of growth rate under different grid sizes for the co-flowing jet problem.

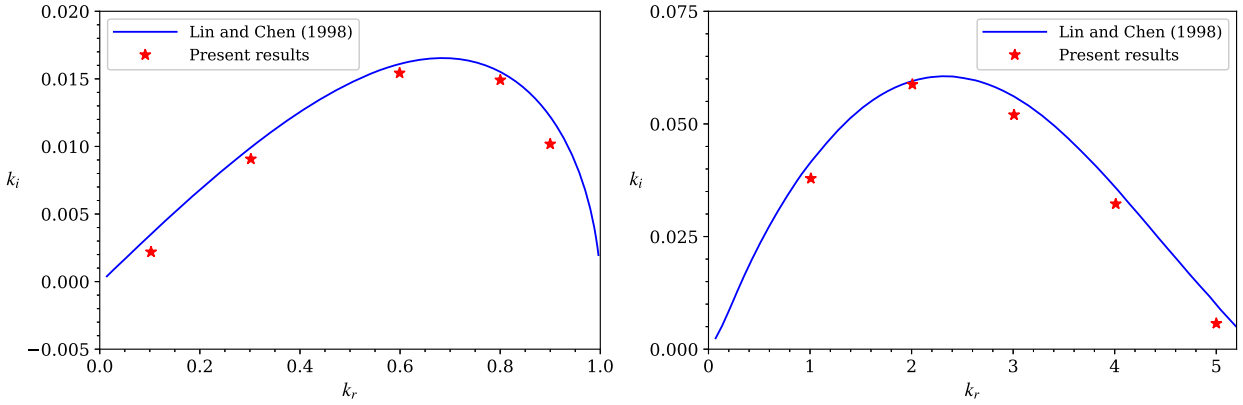


Fig. 20. Left: growth rate curve for the Rayleigh mode with $N = 0.018$, $Q = 0.0013$, $Re = 1000$, $We = 400$, $l = 10$, and $Fr^{-1} = 0$. Right: growth rate curve for the Taylor mode with $N = 0.019$, $Q = 0.013$, $Re = 1000$, $We = 4761.9$, $l = 10$, and $Fr^{-1} = 0.0001$. The blue line represents the linear stability analysis (LSA) results in Lin and Chen [93].

in the right part of Fig. 20, Q and We are chosen such that $Q > We^{-1}$ to yield the Taylor mode [91,93]. In our direct numerical simulations (DNS), the input wavelengths are chosen to capture the growth rate of the instability. It is found that the present results obtained by the consistent scheme can capture the maximum spatial amplification rate k_{im}, and the differences between our scatter points and LSA results in [93] might be due to the nonlinear effect neglected in the LSA. In the next section, a more complex Stokes wave case is considered to further validate the consistent scheme for the 3D problem.

5.7. 3D breaking wave

This section investigates a 3D Stokes breaking wave, which is a dynamic and complex problem that is considered computationally expensive. In addition to validating the consistent scheme for 3D problems, another objective of this test is to compare the computational cost among the single-level and multilevel subcycling and non-subcycling cases. The computational domain is $L_x \times L_y \times L_z = \lambda \times \lambda/2 \times \lambda$, where $\lambda = 0.25\,\mathrm{m}$ is the wavelength. The boundary conditions in the x and z directions are periodic, and the free slip boundary condition is applied in the y direction. The sideview (xy-plane) of the initial wave geometry and the computational domain is given by Fig. 21. The mean water depth is $h = \lambda/2$. The surface profile of a deep-water wave is initialized as

$$\eta(x, y, z) = a\left(\cos(kx) + \frac{1}{2}\varepsilon\cos(2kx) + \frac{3}{8}\varepsilon^2\cos(3kx)\right), \tag{38}$$

where a is the wave amplitude, $k = 2\pi/\lambda$ is the wavenumber, λ is the wavelength, and $\varepsilon = ak$ is the initial wave steepness. The velocities are

$$u(x, y, z) = \Omega a \exp(ky)\cos(kx), \quad v(x, y, z) = \Omega a \exp(ky)\sin(kx), \tag{39}$$

where $\Omega = \sqrt{gk(1+\varepsilon^2)}$ [97]. The free slip boundary condition is imposed in the y direction, and the periodic boundary condition is applied in the x direction.

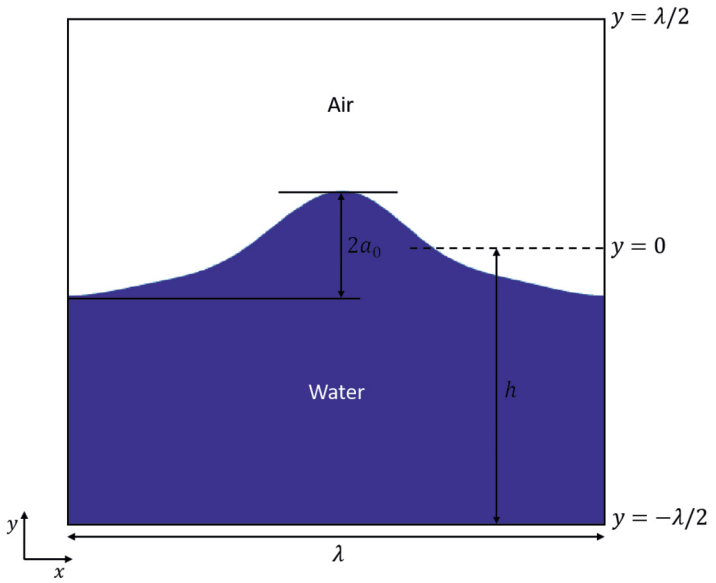


Fig. 21. Sideview (xy-plane) of the initial wave geometry and the computational domain for the 3D breaking wave problem.

Table 4
Parameters of cases for the 3D breaking wave problem.

Case No.	Grid number on level 0	l_{max}	Δt^0	Cycling method
1	$512 \times 256 \times 512$	0	1.6×10^{-5}	–
2	$128 \times 64 \times 128$	2	6.4×10^{-5}	Subcycling
3	$128 \times 64 \times 128$	2	1.6×10^{-5}	Non-subcycling

Table 5
Number of grid cells of the cases in the 3D breaking wave problem at $t/T = 0.5$.

Case No.	Cells on level 0	Cells on level 1	Cells on level 2	Total cells	Total cells normalized by Case 3
1	67,108,864	–	–	67,108,864	4.92
2	1,048,576	4,194,304	8,388,608	13,631,488	1
3	1,048,576	4,194,304	8,388,608	13,631,488	1

Three cases using the consistent scheme are considered, as listed in Table 4. In all these cases, we follow the parameters in [30,97], in which $\lambda = 0.27$ m, $a = 0.0236$ m, and $\varepsilon = 0.55$. Let λ be the characteristic length scale and $T = \sqrt{\lambda/g}$ denote the characteristic time scale. The Reynolds number is $Re = \rho_w \lambda^{3/2} g^{1/2}/\mu_w = 4.0 \times 10^6$. Other dimensionless numbers are the Froude number $Fr = \sqrt{\lambda/g}/T = 1.0$, Weber number $We = \rho_w \lambda^2 g/\sigma = 99.6$, density ratio $\rho_a/\rho_w = 0.0012$, and dynamic viscosity ratio $\mu_a/\mu_w = 0.04$. For the cases on the multilevel grid, grid cells are refined near the air–water interface.

Fig. 22 shows the evolution of the air–water interface obtained from the three-level subcycling case (Case 2). Grid patches are dynamically refined around the interface as time evolves. At the initial stage of the simulation, the 3D wave geometry is similar to that in the 2D simulation. Then, the plunging jets form at the wave crest and strike the front wave face. At approximately $t/T = 2.0$, air bubbles are entrained by the plunging jets, and 3D structures are observed. Next, upward splashes are generated, and more air is entrained at approximately $t/T = 3.0$. Water droplets and bubbles are also generated ($t/T = 4.0$) by the plunging breaker. From $t/T = 8.0$, the bubbles burst out of the surface, the droplets fall into the water, and the wave surface gradually becomes smooth. The breaking process obtained in our 3D simulation is consistent with the results in [30,98]. The left part of Fig. 23 compares the total mechanical energy in the above three cases with results in the literature. The time series of the total mechanical energy obtained from the consistent scheme agrees with the results of Wang et al. [99] and Yang et al. [30].

To compare the computational cost for different cases, we profile each case for $t/T = 0 - 0.5$ using 256 CPUs on a Cray XC40/50 (Onyx) system at the U.S. Army Engineer Research and Development Center, excluding the I/O costs. Table 5 shows the total number of grid cells for different cases at $t/T = 1.0$. Compared with the adaptive cases with $l_{\max} = 2$ (Cases 2 and 3), the single-level case (Case 1) has nearly five times more cells; i.e., the adaptive refinement considerably reduces the total number of grid cells.

The right part of Fig. 23 compares the wall clock time between the single-level case and the multilevel cases for the time range $t/T = 0 - 0.5$. Compared with the single-level case (Case 1), the three-level subcycling case (Case 2) achieves more

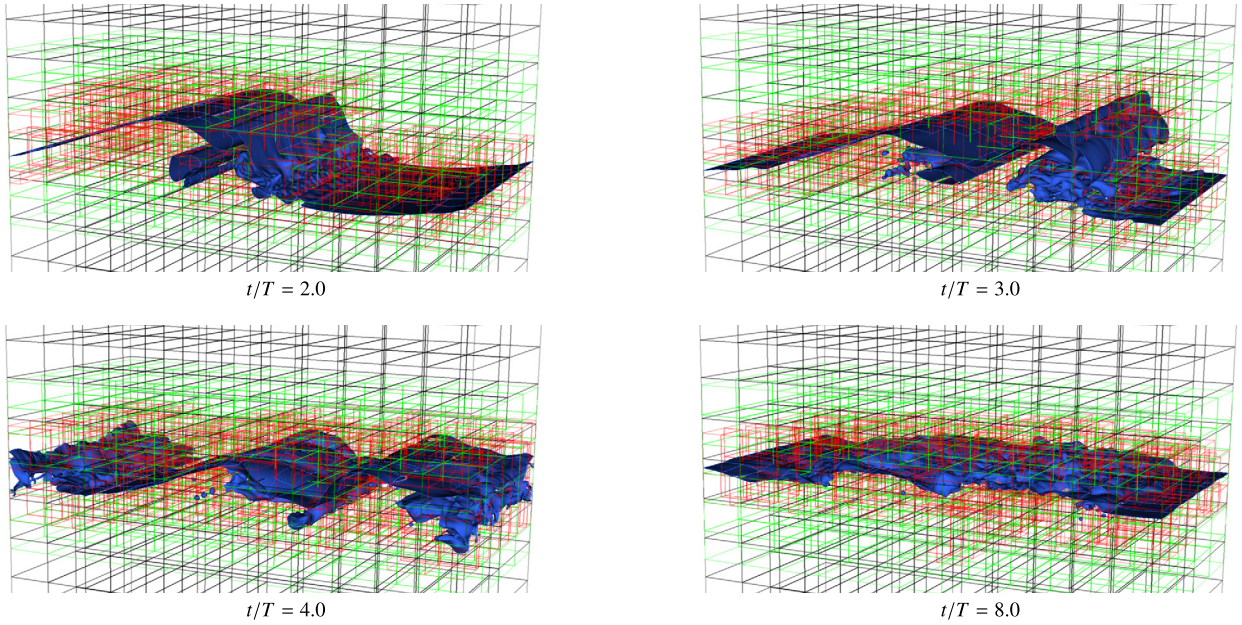


Fig. 22. Evolution of a 3D plunging breaker obtained from the three-level subcycling case (Case 2) using the consistent scheme. The black, green, and red lines represent patches on levels 0, 1, and 2, respectively.

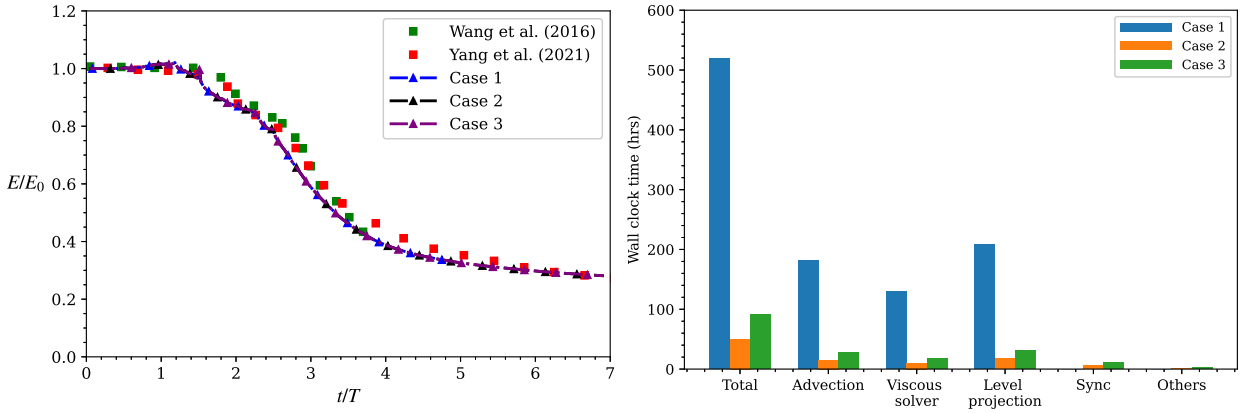


Fig. 23. Left: comparison of the time evolution of the total energy among the single-level case (Case 1), three-level subcycling case (Case 2), three-level non-subcycling case (Case 3), and previous results (Yang et al. [30], Wang et al. [99]) for the 3D breaking wave problem. Right: comparison of the wall clock time of key advancing steps among the single-level case (Case 1), three-level subcycling case (Case 2), and three-level non-subcycling case (Case 3) for the 3D breaking wave problem.

than a 10× speedup in terms of wall clock time, which significantly reduces the computational cost of the 3D simulation. By comparing the nonsubcycled case (Case 3) with the subcycled case (Case 2), we find that the subcycled case further lowers the computational cost by a factor of 1.8. The reason is that, compared to the non-subcycling method, the subcycling method uses a larger time step size for the coarser levels. In addition to the total wall clock time, the wall clock time spent on some key parts of the algorithm is also documented, including advection, viscous solver, level projection, and synchronization. Among them, the level projection takes the most time, followed by the advection step and the viscous solver step. Therefore, optimization of these three parts is desired in future work. The part denoted as the "Others", including the regridding, the interpolation operations, and the reinitialization step, account for only approximately 2% of the total computation time.

Finally, we emphasize that only the cases using the consistent scheme are presented here. For 3D breaking wave cases with a relatively low Reynolds number and a low density ratio, we simulate them using both the inconsistent scheme and the consistent scheme and then compare their computational cost. We find that the computational time in the consistent scheme is approximately the same as that in the inconsistent scheme, which indicates that the consistent scheme has negligible computational overhead compared with the inconsistent scheme.

6. Conclusions

In this work, we have developed a consistent adaptive framework for the simulations of incompressible two-phase flows with high density ratios and high Reynolds numbers. It is found that a consistent discretization of convective terms in mass and momentum equations is important to ensure the stability of the simulations for high-density-ratio flows. This is achieved by employing the bounded and monotonic CUI scheme for the discretization of the convective fluxes. Although the inconsistent scheme, in which a Godunov scheme is applied in place of the CUI scheme for momentum transport, could work well in the rising bubble and Rayleigh–Taylor instability cases [25,38], it does not perform satisfactorily at high density ratios, as evidenced in the convected droplet, dam breaking, and droplet splashing cases, and at high Reynolds numbers, as shown in the breaking wave cases in Section 5.

Different from the previous work [29,81], a purely collocated grid is employed in the present framework, in which all variables are defined at the center of the grid cells. This design eases the implementation of the CUI scheme because the divergence-free advective velocity $\mathbf{u}_{\text{adv}}$ for the cell-centered variables can be directly obtained after the MAC projection. Moreover, only one set of interpolation and averaging schemes is needed when the multilevel grid is involved.

Both subcycling and non-subcycling advancement methods are embedded in this unified framework. One can choose either of these two methods or even combine them for time integration. Note that multilevel advancement (Algorithm 1 in Section 4.2.3) decouples the time advancement for different levels, which relaxes the time step constraint on the coarser levels if the subcycling method is applied. On the other hand, the non-subcycling method avoids time interpolation across the different levels because data on all levels are synchronized to the same time instant during the simulation. Numerical simulations of the droplet splashing problem demonstrate that the subcycling and non-subcycling methods can produce consistent results and accurately capture the complex dynamics when surface tension is involved. A synchronization algorithm is employed to maintain the consistency of variables across multiple levels, and the reinitialization algorithm for the LS function is implemented on the multilevel grid to improve the mass conservation of the two-phase flow.

The consistent scheme provides a numerically stable and reasonably accurate solution to realistic multiphase flows, such as breaking waves with a high Reynolds number. It is found that an unphysical spurious thin sheet is generated at the wave crest when the inconsistent scheme is used. When AMR is applied to locally resolve the complex flow physics near the wave surface, the multilevel cases can achieve the same level of accuracy with fewer total grid cells compared with the single-level fine-grid cases. In particular, for the 3D breaking wave problem, the multilevel simulation can capture the evolution of the total mechanical energy accurately with substantial speedup compared with the single-level simulation. Therefore, the proposed AMR framework is promising for high-fidelity simulations of complex two-phase flows with high density ratios and high Reynolds numbers. In the future, we plan to support a high refining ratio between different levels and incorporate more interface-capturing methods (e.g. the ghost fluid method (GFM) [19,20,100], front-tracking method [7], volume-of-fluid (VOF) method [8–10], conservative level set (CLS) method [16,24,36], and the coupled level set and volume-of-fluid (CLSVOF) method [101]) into this unified AMR framework. More practical 3D cases, including the turbulent liquid jets [24,102–104] and the liquid sheet atomization [105], will also be considered as illustrations to make the consistent scheme more compelling.

CRediT authorship contribution statement

Yadong Zeng: Conceptualization, Methodology, Writing – original draft. **Han Liu:** Formal analysis, Discussion. **Qiang Gao:** Validation, Writing – review & editing. **Ann Almgren:** Discussion, Supervision, Writing – review & editing. **Amneet Pal Singh Bhalla:** Discussion, Writing – review & editing. **Lian Shen:** Funding acquisition, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgements

Y. Z., H. L., Q. G., and L. S. gratefully acknowledge the support of this work by the Office of Naval Research (N00014-17-1-2658, N00014-19-1-2139, and N00014-22-1-2481) and the National Science Foundation (OCE-1924799). A. P. S. B. acknowledges support from NSF award OAC-1931368. Y. Z. and H. L. extend their special thanks to Dr. Nishant Nangia and Dr. Anqing Xuan for discussing the numerical algorithms. Y. Z. also thanks the Center for Computational Sciences and Engineering (CCSE) and the Applied Numerical Algorithms Group (ANAG) in the Berkeley Lab for their help with this study.

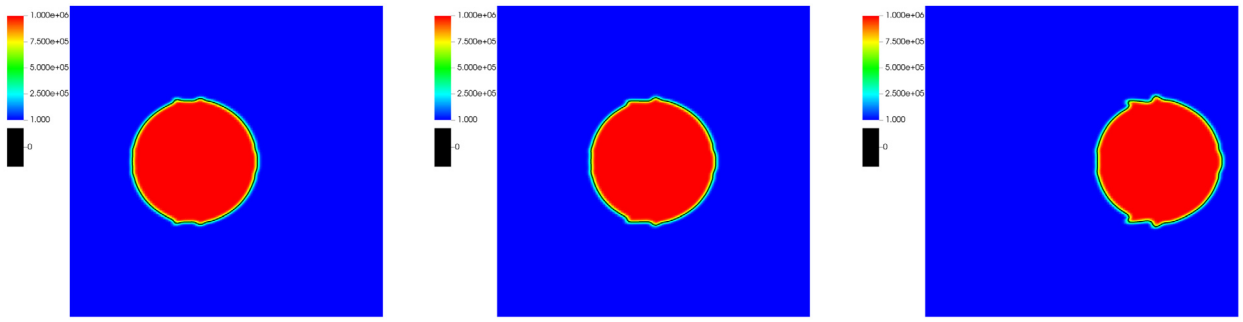


Fig. 24. Geometry and density evolution of the 2D convected droplet problem using the consistent scheme. The density field is not reset by the level set function after each time step.

Appendix A. Inconsistent scheme

For the inconsistent scheme, the discretizations are applied to the nonconservative forms of the Navier–Stokes equations (Eqs. (4) and (5)) [25]. Due to the same advection and reinitialization schemes of the LS function ϕ, **Step 1** and **Step 3** in the inconsistent scheme are the same as those in the consistent scheme (Section 4.1.1). **Step 5**, which is used to obtain the updated velocity $\widetilde{\mathbf{u}}^{n+1}$ and pressure $p^{n+1/2}$, is also the same as **Step 5** in the consistent scheme, assuming the intermediate velocity $\widetilde{\mathbf{u}}^{*,n+1}$ has been calculated. Thus, we write only the details of **Step 3** and **Step 4** for the inconsistent scheme.

Step 3: Both the viscosity μ^{n+1} and the density ρ^{n+1} fields are reset through the Heaviside function as

$$\mu^{n+1} = \mu_g + \left(\mu_l - \mu_g\right) \widetilde{H}\left(\phi^{n+1}\right), \tag{40}$$

$$\rho^{n+1} = \rho_g + \left(\rho_l - \rho_g\right) \widetilde{H}\left(\phi^{n+1}\right), \tag{41}$$

where μ_l and ρ_l are the viscosity and density of the liquid phase and μ_g and ρ_g are those for the gas phase, respectively. The smoothed Heaviside function, which is regularized over n_{cells} grid cells on either side of the two-phase interface, is defined as

$$\widetilde{H}\left(\phi^{n+1}\right) = \begin{cases} 0, & \phi^{n+1} < -n_{\text{cells}}\Delta x \\ \frac{1}{2}\left(1 + \frac{1}{n_{\text{cells}}\Delta x}\phi^{n+1} + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{cells}}\Delta x}\phi^{n+1}\right)\right), & \left|\phi^{n+1}\right| \le n_{\text{cells}}\Delta x \\ 1, & \text{otherwise} \end{cases} \tag{42}$$

where the uniform grid spacing $\Delta x = \Delta y$ is assumed [29] and $n_{cells} = 1$ or 2 is applied for all testing cases in Section 5. The midpoint values of ρ and μ are then calculated as $\rho^{n+\frac{1}{2}} = (\rho^{n+1} + \rho^n)/2$ and $\mu^{n+\frac{1}{2}} = (\mu^{n+1} + \mu^n)/2$, respectively.

Step 4: The intermediate velocity $\widetilde{\mathbf{u}}^{*,n+1}$ is solved semi-implicitly as

$$\rho^{n+\frac{1}{2}}\left(\frac{\widetilde{\mathbf{u}}^{*,n+1} - \mathbf{u}^n}{\Delta t} + \nabla\cdot(\mathbf{u}\mathbf{u})^{n+\frac{1}{2}}\right) = -\nabla p^{n-\frac{1}{2}} + \frac{1}{2}\left(\nabla\cdot\mu(\phi^{n+1})\nabla\widetilde{\mathbf{u}}^{*,n+1} + \nabla\cdot\mu(\phi^n)\nabla\mathbf{u}^n\right) + \rho^{n+\frac{1}{2}}\mathbf{g} + \mathbf{f}_s^{n+\frac{1}{2}}, \tag{43}$$

where the convective term $\nabla\cdot(\mathbf{u}\mathbf{u})^{n+\frac{1}{2}}$ is calculated using the Godunov scheme [17,47]. Same as the consistent scheme, the density field ρ^{n+1} is reset by the LS function ϕ^{n+1} using Eq. (41) after the **Step 4** of each time step.

Appendix B. Necessity of resetting the density field

We provide more evidence for the necessity of resetting the density field by the level set function. Using the same parameters of the 2D convected droplet problem in section 5.2, we preform a simulation without resetting the density field after each time step, which means that the level set function has no influence on the density and there is no information exchange between these two fields. By comparing the lower part of Fig. 9 and Fig. 24, it is evident that the density region grows without resetting it by the level set function.

References

[1] A. Iafrati, F. De Vita, R. Verzicco, Effects of the wind on the breaking of modulated wave trains, Eur. J. Mech. B, Fluids 73 (2019) 6–23.
[2] Z. Yang, B.-Q. Deng, L. Shen, Direct numerical simulation of wind turbulence over breaking waves, J. Fluid Mech. 850 (2018) 120–155.
[3] V. Bertram, Practical Ship Hydrodynamics, Elsevier, 2011.

[4] L. Deike, E. Ghabache, G. Liger-Belair, A.K. Das, S. Zaleski, S. Popinet, T. Séon, Dynamics of jets produced by bursting bubbles, Phys. Rev. Fluids 3 (2018) 013603.
[5] T. Ménard, S. Tanguy, A. Berlemont, Coupling level set/VOF/ghost fluid methods: validation and application to 3D simulation of the primary break-up of a liquid jet, Int. J. Multiph. Flow 33 (2007) 510–524.
[6] A. Prosperetti, G. Tryggvason, Computational Methods for Multiphase Flow, Cambridge University Press, 2009.
[7] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[8] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[9] M. Renardy, Y. Renardy, J. Li, Numerical simulation of moving contact line problems using a volume-of-fluid method, J. Comput. Phys. 171 (2001) 243–263.
[10] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465–502.
[11] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (1994).
[12] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. USA 93 (1996) 1591–1595.
[13] Ž. Tuković, H. Jasak, A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow, Comput. Fluids 55 (2012) 70–84.
[14] S. Popinet, Numerical models of surface tension, Annu. Rev. Fluid Mech. 50 (2018) 49–75.
[15] M. Sussman, P. Smereka, Axisymmetric free boundary problems, J. Fluid Mech. 341 (1997) 269–294.
[16] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (2005) 225–246.
[17] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comput. Phys. 148 (1999) 81–124.
[18] H.-Z. Yuan, C. Shu, Y. Wang, S. Shu, A simple mass-conserved level set method for simulation of multiphase flows, Phys. Fluids 30 (2018) 040908.
[19] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.
[20] 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.
[21] B. Lalanne, L.R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, J. Comput. Phys. 301 (2015) 289–307.
[22] G. Son, V.K. Dhir, A level set method for analysis of film boiling on an immersed solid surface, Numer. Heat Transf., Part B, Fundam. 52 (2007) 153–177.
[23] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, J. Comput. Phys. 225 (2007) 785–807.
[24] 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.
[25] Y. Zeng, A. Xuan, J. Blaschke, L. Shen, A parallel cell-centered adaptive level set framework for efficient simulation of two-phase flows with subcycling and non-subcycling, J. Comput. Phys. 448 (2022) 110740.
[26] M. Rudman, A volume-tracking method for incompressible multifluid flows with large density variations, Int. J. Numer. Methods Fluids 28 (1998) 357–378.
[27] M. Bussmann, D.B. Kothe, J.M. Sicilian, Modeling high density ratio incompressible interfacial flows, in: Fluids Engineering Division Summer Meeting, vol. 36150, 2002, pp. 707–713.
[28] M. Raessi, A level set based method for calculating flux densities in two-phase flows, in: Summer Programm, Center for Turbulent Research, 2008.
[29] N. Nangia, B.E. Griffith, N.A. Patankar, A.P.S. Bhalla, A robust incompressible Navier–Stokes solver for high density ratio multiphase flows, J. Comput. Phys. 390 (2019) 548–594.
[30] Z. Yang, M. Lu, S. Wang, A robust solver for incompressible high-Reynolds-number two-fluid flows with high density contrast, J. Comput. Phys. (2021) 110474.
[31] Q. Gao, G.B. Deane, H. Liu, L. Shen, A robust and accurate technique for Lagrangian tracking of bubbles and detecting fragmentation and coalescence, Int. J. Multiph. Flow 135 (2021) 103523.
[32] T. Inamuro, T. Ogata, S. Tajima, N. Konishi, A lattice Boltzmann method for incompressible two-phase flows with large density differences, J. Comput. Phys. 198 (2004) 628–644.
[33] H. Ding, P.D. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with large density ratios, J. Comput. Phys. 226 (2007) 2078–2095.
[34] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2007) 469–505.
[35] M. Raessi, H. Pitsch, Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method, Comput. Fluids 63 (2012) 70–81.
[36] O. Desjardins, V. Moureau, Methods for multiphase flows with high density ratio, in: Summer Programm 2010, Center for Turbulent Research, 2010, pp. 313–322.
[37] S. Ghods, M. Herrmann, A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods, Phys. Scr. 2013 (2013) 014050.
[38] J.K. Patel, G. Natarajan, A generic framework for design of interface capturing schemes for multi-fluid flows, Comput. Fluids 106 (2015) 108–118.
[39] N. Nangia, N.A. Patankar, A.P.S. Bhalla, A DLM immersed boundary method based wave-structure interaction solver for high density ratio multiphase flows, J. Comput. Phys. 398 (2019) 108804.
[40] A.P.S. Bhalla, N. Nangia, P. Dafnakis, G. Bracco, G. Mattiazzo, Simulating water-entry/exit problems using Eulerian–Lagrangian and fully-Eulerian fictitious domain methods within the open-source IBAMR library, Appl. Ocean Res. 94 (2020) 101932.
[41] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, J. Comput. Phys. 222 (2007) 536–555.
[42] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.
[43] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.
[44] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929.
[45] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321.
[46] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, J. Comput. Phys. 190 (2003) 572–600.
[47] 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.
[48] D.F. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, J. Comput. Phys. 163 (2000) 271–312.
[49] D.F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible Navier–Stokes equations in three dimensions, J. Comput. Phys. 227 (2008) 1863–1886.
[50] M.L. Minion, A projection method for locally refined grids, J. Comput. Phys. 127 (1996) 158–178.

[51] Z. Yao, R. Jambunathan, Y. Zeng, A. Nonaka, A massively parallel time-domain coupled electrodynamics–micromagnetics solver, Int. J. High Perform. Comput. Appl. 36 (2022) 167–181.
[52] C. Burstedde, L.C. Wilcox, O. Ghattas, p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees, SIAM J. Sci. Comput. 33 (2011) 1103–1133.
[53] A. Posa, M. Vanella, E. Balaras, An adaptive reconstruction for Lagrangian, direct-forcing, immersed-boundary methods, J. Comput. Phys. 351 (2017) 422–436.
[54] M. Vanella, P. Rabenold, E. Balaras, A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid–structure interaction problems, J. Comput. Phys. 229 (2010) 6427–6449.
[55] Y. Zeng, L. Shen, A unified AMR framework for multiphase flow and fluid-structure interaction problems with both non-subcycling and subcycling, in: APS Division of Fluid Dynamics Meeting Abstracts, 2019, S19–001.
[56] B.T. Gunney, R.W. Anderson, Advances in patch-based adaptive mesh refinement scalability, J. Parallel Distrib. Comput. 89 (2016) 65–84.
[57] H. Kohno, T. Tanahashi, Numerical analysis of moving interfaces using a level set method coupled with adaptive mesh refinement, Int. J. Numer. Methods Fluids 45 (2004) 921–944.
[58] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (2009) 5838–5866.
[59] M.R. Pivello, M.M. Villar, R. Serfaty, A.M. Roma, A.d. Silveira-Neto, A fully adaptive front tracking method for the simulation of two phase flows, Int. J. Multiph. Flow 58 (2014) 72–82.
[60] W. Zhang, A. Almgren, V. Beckner, J. Bell, J. Blaschke, C. Chan, M. Day, B. Friesen, K. Gott, D. Graves, et al., AMReX: a framework for block-structured adaptive mesh refinement, J. Open Sour. Softw. 4 (2019).
[61] 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.
[62] M. Zingale, Introduction to Computational Astrophysical Hydrodynamics, Open-Astrophysics-Bookshelf, vol. 13, 2014, pp. 129–137.
[63] A.S. Almgren, J.B. Bell, W.G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, SIAM J. Sci. Comput. 17 (1996) 358–369.
[64] W.J. Rider, Approximate Projection Methods for Incompressible Flow: Implementation, Variants and Robustness, LANL Unclassified Report LA-UR-94-2000, Los Alamos National Laboratory, 1995.
[65] M.F. Lai, A projection method for reacting flow in the zero Mach number limit, Ph.D. thesis, University of California, Berkeley, 1993.
[66] R.D. Guy, A.L. Fogelson, Stability of approximate projection methods on cell-centered grids, J. Comput. Phys. 203 (2005) 517–538.
[67] Y. Zeng, A.P. Bhalla, S. He, L. Shen, A subcycling/non-subcycling time advancement scheme-based sharp-interface immersed boundary method framework for solving fluid-structure interaction problems on dynamically adaptive grids, in: APS Division of Fluid Dynamics Meeting Abstracts, 2021, F26–004.
[68] P. Roe, M. Baines, Algorithms for advection and shock problems, in: Numerical Methods in Fluid Mechanics, 1982, pp. 281–290.
[69] N.P. Waterson, H. Deconinck, Design principles for bounded higher-order convection schemes – a unified approach, J. Comput. Phys. 224 (2007) 182–207.
[70] M. Darwish, F. Moukalled, The Finite Volume Method in Computational Fluid Dynamics: an Advanced Introduction with OpenFOAM® and Matlab®, Springer, 2021.
[71] N. Nangia, H. Johansen, N.A. Patankar, A.P.S. Bhalla, A moving control volume approach to computing hydrodynamic forces and torques on immersed bodies, J. Comput. Phys. 347 (2017) 437–462.
[72] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[73] Y. Zeng, L. Shen, Modelling Wave Energy Converter (WEC) Pointer Absorbers Using AMR Techniques with Both Subcycling and Non-subcycling, Earth and Space Science Open Archive, 2020, p. 1.
[74] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[75] A.P.S. Bhalla, R. Bale, B.E. Griffith, N.A. Patankar, A unified mathematical framework and an adaptive numerical method for fluid–structure interaction with rigid, deforming, and elastic bodies, J. Comput. Phys. 250 (2013) 446–476.
[76] Y. Zeng, A.P.S. Bhalla, L. Shen, A subcycling/non-subcycling time advancement scheme-based DLM immersed boundary method framework for solving single and multiphase fluid–structure interaction problems on dynamically adaptive grids, Comput. Fluids 238 (2022) 105358.
[77] Y. Zeng, Numerical Simulations of the Two-phase flow and Fluid-Structure Interaction Problems with Adaptive Mesh Refinement, Ph.D. thesis, University of Minnesota, 2022.
[78] P.-H. Chiu, Y.-T. Lin, A conservative phase field method for solving incompressible two-phase flows, J. Comput. Phys. 230 (2011) 185–204.
[79] S. Mirjalili, C.B. Ivey, A. Mani, A conservative diffuse interface method for two-phase flows with provable boundedness properties, J. Comput. Phys. 401 (2020) 109006.
[80] R.V. Rezende, R.A. Almeida, A.A.U. de Souza, S.M.G.U. Souza, A two-fluid model with a tensor closure model approach for free surface flow simulations, Chem. Eng. Sci. 122 (2015) 596–613.
[81] J.K. Patel, G. Natarajan, A novel consistent and well-balanced algorithm for simulations of multiphase flows on unstructured grids, J. Comput. Phys. 350 (2017) 207–236.
[82] Z. Gu, H. Wen, C. Yu, T.W. Sheu, Interface-preserving level set method for simulating dam-break flows, J. Comput. Phys. 374 (2018) 249–280.
[83] J.C. Martin, W.J. Moyce, J. Martin, W. Moyce, W.G. Penney, A. Price, C. Thornhill, Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane, Philos. Trans. R. Soc. A 244 (1952) 312–324.
[84] M. Singh, H.M. Haverinen, P. Dhagat, G.E. Jabbour, Inkjet printing – process and its applications, Adv. Mater. 22 (2010) 673–685.
[85] J. Kim, Spray cooling heat transfer: the state of the art, Int. J. Heat Fluid Flow 28 (2007) 753–767.
[86] G. Coppola, G. Rocco, L. de Luca, Insights on the impact of a plane drop on a thin liquid film, Phys. Fluids 23 (2011) 022105.
[87] C. Josserand, S. Zaleski, Droplet splashing on a thin liquid film, Phys. Fluids 15 (2003) 1650–1657.
[88] S. Howison, J. Ockendon, J. Oliver, R. Purvis, F. Smith, Droplet impact on a thin fluid layer, J. Fluid Mech. 542 (2005) 1–23.
[89] Q. Li, K. Luo, X. Li, Lattice Boltzmann modeling of multiphase flows at large density ratio with an improved pseudopotential model, Phys. Rev. E 87 (2013) 053301.
[90] J. Hnat, J. Buckmaster, Spherical cap bubbles and skirt formation, Phys. Fluids 19 (1976) 182–194.
[91] 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.
[92] D. Bhaga, M. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, J. Fluid Mech. 105 (1981) 61–85.
[93] S. Lin, J. Chen, Role played by the interfacial shear in the instability mechanism of a viscous liquid jet surrounded by a viscous gas in a pipe, J. Fluid Mech. 376 (1998) 37–51.
[94] Y. Wang, S. Simakhina, M. Sussman, A hybrid level set-volume constraint method for incompressible two-phase flow, J. Comput. Phys. 231 (2012) 6438–6471.
[95] Y. Wang, Numerical methods for two-phase jet flow, Ph.D. thesis, 2010.
[96] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, Y. Wang, A coupled level set-moment of fluid method for incompressible two-phase flows, J. Sci. Comput. 54 (2013) 454–491.

[97] A. Iafrati, Numerical study of the effects of the breaking intensity on wave breaking flows, J. Fluid Mech. 622 (2009) 371–411.
[98] Q. Gao, G.B. Deane, L. Shen, Bubble production by air filament and cavity breakup in plunging breaking wave crests, J. Fluid Mech. 929 (2021).
[99] Z. Wang, J. Yang, F. Stern, High-fidelity simulations of bubble, droplet and spray formation in breaking waves, J. Fluid Mech. 792 (2016) 307–327.
[100] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[101] 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.
[102] M. Natarajan, R. Chiodi, M. Kuhn, O. Desjardins, An all-Mach multiphase flow solver using block-structured AMR, in: ILASS Americas, 30th Annual Conference on Liquid Atomization and Spray Systems, May 12th-15th, Tempe, Arizona, USA, 2019.
[103] M. Herrmann, A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure, J. Comput. Phys. 229 (2010) 745–759.
[104] A.A. Mukundan, T. Ménard, A. Berlemont, J.C.C.B. de Motta, M. Herrmann, Validation and analysis of primary atomization of turbulent liquid jet in crossflow simulations, in: ILASS Americas, 30th Annual Conference on Liquid Atomization and Spray Systems, May 12th-15th, Tempe, Arizona, USA, 2019.
[105] Z. Li, C. Liu, R. Gao, C. Hu, A consistent mass-momentum advection method for the simulation of large-density-ratio two-phase flows, Int. J. Multiph. Flow 156 (2022) 104192.