Inconsistencies in unstructured geometric volume-of-fluid methods for two-phase flows with high density ratios

Jun Liu, Tobias Tolle, Davide Zuzio, Jean-Luc Estivalèzes, Santiago Marquez Damian, Tomislav Marić

Computers and Fluids 281 (2024) 106375

Contents lists available at ScienceDirect

Computers and Fluids

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

Inconsistencies in unstructured geometric volume-of-fluid methods for two-phase flows with high density ratios

Jun Liu [a], Tobias Tolle [b], Davide Zuzio [c], Jean-Luc Estivalèzes [c], Santiago Marquez Damian [d], Tomislav Marić [a,*]

[a] Mathematical Modeling and Analysis, Technische Universität Darmstadt, Germany
[b] Bosch Research, Robert Bosch GmbH; research conducted at MMA, Technische Universität Darmstadt, Germany
[c] ONERA/DMPE, Université de Toulouse, France
[d] Centro de Investigaciones en Mecánica Computacional (CIMEC), UNL/CONICET, Argentina

ARTICLE INFO

Keywords:
Volume-of-fluid
Unstructured
Finite volume
High density ratios

ABSTRACT

Geometric flux-based Volume-of-Fluid (VOF) methods (Marić et al., 2020) are widely considered consistent in handling two-phase flows with high density ratios. However, although the conservation of mass and momentum is consistent for two-phase incompressible single-field Navier–Stokes equations without phase-change (Liu et al., 2023), discretization may easily introduce inconsistencies that result in very large errors or catastrophic failure. We apply the consistency conditions derived for the ρLENT unstructured Level Set/Front Tracking method (Liu et al., 2023) to flux-based geometric VOF methods (Marić et al., 2020), and implement our discretization into the plicRDF-isoAdvector geometrical VOF method (Roenby et al., 2016). We find that computing the mass flux by scaling the geometrically computed fluxed phase-specific volume can ensure equivalence between the mass conservation equation and the phase indicator (volume conservation) if consistent discretization schemes are chosen for the temporal and convective term. Based on the analysis of discretization errors, we suggest a consistent combination of the temporal discretization scheme and the interpolation scheme for the momentum convection term. We confirm the consistency by solving an auxiliary mass conservation equation with a geometrical calculation of the face-centered density (Liu et al., 2023). We prove the equivalence between these two approaches mathematically and verify and validate their numerical stability for density ratios within $[1, 10^6]$ and viscosity ratios within $[10^2, 10^5]$.

1. Introduction

Numerical simulation methods still face challenges when dealing with incompressible two-phase flows that involve fluid phases of significantly different densities (high density ratios). We have provided a detailed review of two-phase flow simulation methods that address the challenges in handling high density ratios recently in [1], and here we only list more current literature contributions.

Huang et al. [2] proposed the mixed Upwind/Central WENO scheme on a staggered structured grid, an extension of the conventional WENO scheme [3] on collocated grids, to spatially discretize the nonlinear convective term in conservative form. The Upwind WENO scheme is used to evaluate velocity at cell faces, while 3 different forms of Central WENO scheme are applied to evaluate mass flux in x-/y-directions as well as density at cell corners. In addition to the mentioned spatial discretization scheme, Huang et al. [2] proposed also a semi-implicit projection scheme to decouple pressure and velocity in momentum transport equation. A backward difference scheme combined with a special treatment for the viscous term is introduced to discretize the momentum equation without the pressure gradient. The viscous term containing the intermediate velocity is split into two parts, one of which comprises the constant arithmetic mean of two phases' viscosity and the intermediate velocity, another one comprises the updated viscosity and the explicit velocity (referring to [4, Apendix A]). Numerous 2D cases with density ratio $[1, 1000]$ are tested, whose results are in good agreement with analytical and experimental results. The interface capturing method in [2] is selected to phase-field. The authors indicated that the mixed Upwind/Central WENO scheme is able to be coupled with any interface capturing/tracking method.

Xie et al. [5] introduced a consistent and balanced-force model with level-set and volume of fluid function (CBLSVOF) on polyhedral

* Corresponding author.
E-mail addresses: liu@mma.tu-darmstadt.de (J. Liu), tolle@mma.tu-darmstadt.de (T. Tolle), Davide.Zuzio@onera.fr (D. Zuzio), Jean-Luc.Estivalezes@onera.fr (J.-L. Estivalèzes), santiagomarquezd@gmail.com (S.M. Damian), maric@mma.tu-darmstadt.de (T. Marić).

https://doi.org/10.1016/j.compfluid.2024.106375
Received 21 December 2023; Received in revised form 4 July 2024; Accepted 17 July 2024
Available online 20 July 2024

unstructured grids. A hybrid algebraic/geometric VOF method based on THINC/QQ [6] is developed to capture interface, while the LS function constructed from volume fraction is exploited to improve the accuracy of interface curvature evaluation. To achieve the consistency, the convective term of both the volume-fraction equation and momentum equation in the conservative form are discretized in the same manner, wherein the velocity and volume fraction are reconstructed by using the identical high-order reconstruction scheme based on a quadratic polynomial function. The balanced-force formulation proposed firstly by Francois et al. [7] is adapted in this work to eliminate the parasitic current at the interface. The interface curvature estimated by the continuous signed-distance function instead of the abruptly changing volume fraction function offers computational simplicity and numerical stability. A 3D single bubble, and two bubbles coalescence cases with the density ratio of magnitude 10^3 are tested. The results agree well with previous experimental results.

Desmons and Coquerelle [8] proposed a generalized approach known as the HOMP (High-Order Momentum Preserving) method. A high-order temporal and spatial scheme, i.e., RK2 and WENO5,3 [9], are used to discretize an auxiliary advection equation of characteristic function χ that is compatible with the mass equation and independent of the underlying interface-representation function. The momentum equation is then discretized using the same high-order schemes as before, and the density is deduced from the characteristic function χ, which is computed from the auxiliary advection equation rather than from the interface transport step. During the discretization, both the auxiliary advection equation and the momentum equation remain in the conservative form. The authors combined several interface representation methods, i.e., Volume Of Fluid, Level Set Method, and Moment Of Fluid, with HOMP and then tested them. The results from various selected validation cases, including water and air, demonstrate good agreement with the literature results. It is worth noting that no theoretical explanation is offered as to why the auxiliary advection equation is required, rather than just maintaining the consistency between the interface transport equation and the momentum equation. Alternatively, the higher-order WENO scheme requires very large stencils of variable width when used with the finite-volume method, making its parallel implementation inefficient using the domain-decomposition / message-passing parallel programming model.

El Ouafa et al. [10] devised a fully coupled solver for simulating incompressible two-phase flows characterized by large density and viscosity ratios on staggered structured meshes. The interface is captured by a Piecewise Linear Interface Calculation (PLIC)-Volume of Fluid (VOF) method. In this solver, the linearized momentum and continuity equations arising from the implicit solution of the fluid velocities and pressure are solved simultaneously to avoid the errors arising from operator splitting. Cases featuring density ratios up to 10^6 and viscosity ratios up to 10^{10} are tested, whose results demonstrate a high level of stability and accuracy.

Yang et al. [11] introduced a robust methodology that integrates the consistent mass-momentum convection approach from Nangia et al. [12] with the CLSVOF interface capturing method by Sussman and Puckett [13] to tackle the challenge posed by high-density ratio problems encountered in high-Reynolds-number flows. At each time step, the interface evolution is initiated using the CLSVOF method to provide an initial value for the mass equation. Subsequently, the conservative form of the momentum equation and mass equation are simultaneously solved employing consistent temporal (second-order Runge–Kutta (RK2)) and spatial (third-order cubic upwind interpolation (CUI)) discretization schemes at all cell faces of the staggered structured mesh. The authors emphasized the importance of employing identical densities in the discretized mass and momentum equations at the two substeps of the RK2 method to ensure robust simulation of high-Reynolds-number two-fluid flows with high density ratios. The proposed method yields accurate predictions for both two-dimensional and three-dimensional wave breaking cases with a density ratio of 10^3 and Reynolds number of 10^8.

Li et al. [14] proposed a straightforward and robust method to simulate high density ratio interfacial flows. Similar to the method in [11], the auxiliary mass equation is solved together with the momentum equation. However, instead of using the consistent discretization schemes in space and time, this method applies the identical cell-center velocity updated by solving the mass equation and the mass flux to the discretized momentum equation to maintain consistency between them. Three different schemes of VOF, i.e, PLIC-VOF, the spatial filtering VOF, and the THINC-VOF, are implemented in the work to transport the interface. The improvements of stability and accuracy for all three schemes in canonical cases like heavy droplet advection and falling droplet show the generality of the consistent method to deal with high density ratios problem.

Zeng et al. [15] transferred the consistent treatment of conservative mass and momentum equations on staggered Cartesian grids introduced by Nangia et al. [12] to multilevel collocated grids. The adapted level set method with a multilevel reinitialization technique is applied to capture the interface. The consistent scheme achieves a numerically stable and reasonably accurate solution to realistic multiphase flows, such as breaking waves with a high Reynolds number.

In this manuscript, we apply the consistency requirements imposed by the single-field Navier–Stokes equations derived in [1] to flux-based geometric Volume-of-Fluid methods [16], using the isoAdvector method to verify and validate our findings [17–19]. The consistency requirements from [1] indicate that geometrical flux-based VOF methods are inherently consistent in handling high density ratios. On the modeling level, an exact solution of the volume fraction equation is equivalent to solving the mass conservation equation. On the discrete level, flux-based VOF methods consistently compute the mass flux needed in the implicitly discretized single-field momentum convection term, by scaling the fluxed phase-specific volume. However, we show that inconsistencies easily arise through the choice of an inconsistent combination of the temporal integration scheme and the interpolation scheme for the two-phase momentum conservation term, leading to significant errors for small density ratios and catastrophic failures for large density ratios.

In the following sections, we briefly review the single-field formulation of incompressible two-phase Navier–Stokes equations and analyze their discretization using the collocated unstructured Finite Volume method [20]. In Section 3.1, we show that a consistent combination of the temporal discretization scheme and the interpolation scheme for the momentum convection term is necessary for ensuring stable solutions with high density ratios. In Sections 3.2 and 3.3, we introduce an auxiliary mass conservation equation with a geometrical calculation of the face-centered density to the geometrical VOF method isoAdvector [17–19]. We prove the equivalence between these two approaches and verify and validate their numerical stability for density ratios of $[1, 10^6]$ and viscosity ratios of $[10^2, 10^5]$ in the results Section 4.

2. Mathematical model

Fig. 1 illustrates an incompressible two-phase flow system without phase change. The flow domain $\Omega \subset \mathbb{R}^3$ with a boundary $\partial\Omega$ is partitioned into two subdomains $\Omega^+(t)$ and $\Omega^-(t)$, occupied by two different fluid phases. The boundary between $\Omega^\pm(t)$, referred to as the fluid interface, is denoted by $\Sigma(t)$ and has the normal $\mathbf{n}_\Sigma$ pointing, say, outwards of the domain $\Omega^-(t)$.

A phase indicator function

$$\chi(\mathbf{x},t) := \begin{cases} 1, & \mathbf{x} \in \Omega^-(t), \\ 0, & \mathbf{x} \in \Omega^+(t), \end{cases} \tag{1}$$

indicates $\Omega^\pm(t)$ and formulates single-field density and dynamic viscosity as

$$\rho(\mathbf{x},t) = (\rho^- - \rho^+)\chi(\mathbf{x},t) + \rho^+, \tag{2}$$

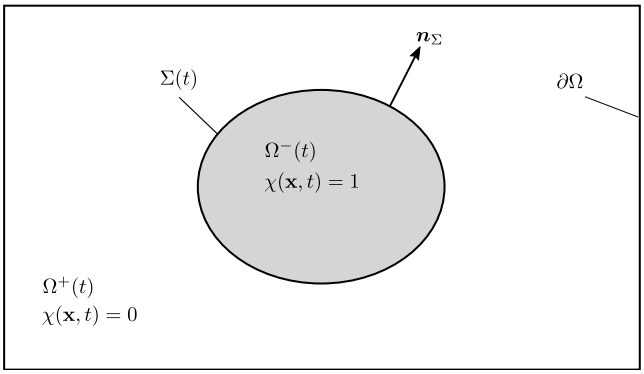


Fig. 1. The domain Ω, split by the fluid interface $\Sigma(t)$ into two sub-domains $\Omega^{\pm}$.

$$\mu(\mathbf{x},t) = (\mu^- - \mu^+)\chi(\mathbf{x},t) + \mu^+, \tag{3}$$

where $\rho^{\pm}$ and $\mu^{\pm}$ are the constant densities and dynamic viscosities in $\Omega^{\pm}(t)$. The one-fluid density and dynamic viscosity are used in the Navier–Stokes equations without phase change, namely

$$\nabla\cdot\mathbf{v} = 0, \tag{4}$$

$$\partial_t(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v}) = -\nabla p - (\mathbf{g}\cdot\mathbf{x})\nabla\rho + \nabla\cdot(\mu\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right)) + \mathbf{f}_\Sigma. \tag{5}$$

where p is the modified pressure defined as the subtraction of hydrostatic pressure from the total pressure, i.e., $p = P - \rho\mathbf{g}\cdot\mathbf{x}$. The surface tension force $\mathbf{f}_\Sigma$ is exerted only on the interface, as

$$\mathbf{f}_\Sigma = \sigma\kappa\mathbf{n}_\Sigma\delta_\Sigma, \tag{6}$$

where σ is the constant surface tension coefficient, κ is twice the local mean curvature of the interface $\Sigma(t)$, and δ_Σ is the interface Dirac distribution.

This single-field formulation of incompressible two-phase Navier–Stokes Equations (NSE) without phase change is especially relevant for engineering applications because it provides a solid modeling basis for two-phase flows with fluid interfaces that can arbitrarily deform, breakup and merge, provided that the method responsible for advecting the phase indicator does not impose its own restrictions regarding fluid interface deformation and topological changes. Single-field NSE also embed strict consistency requirements [1]: an equivalence between the conservation of mass and the phase-indicator (volume) conservation. These requirements translate on the discrete level into requirements for the computation of the mass flux in the discretized two-phase momentum convection term in Eq. (5), analyzed in the following section.

3. Numerical methodology

3.1. Consistent mass, volume, and momentum conservation

The unstructured Finite Volume discretization of the single-field two-phase momentum convection term from Eq. (5) results in

$$\int_{\Omega_c}\nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v})\,dV = \int_{\partial\Omega_c}(\rho\mathbf{v}\otimes\mathbf{v})\cdot\mathbf{n}\,ds = \sum_{f\in F_c}\rho_f F_f\mathbf{v}_f + \mathbf{e}_{\rho\mathbf{v}}(h^2), \tag{7}$$

for the solution domain Ω, discretized by $|C|$ non-overlapping control volumes $\Omega := \cup_{c\in C}\Omega_c$. Each finite volume Ω_c is bounded by a number ($|F_c|$) of non-planar surfaces (faces) S_f, i.e. $\partial\Omega_c = \cup_{f\in F_c}S_f$, oriented outwards of Ω_c. Linearizing the convective term for the velocity $\mathbf{v}$ and using the second-order accurate collocated Unstructured Finite Volume Method (UFVM) [20–22] introduces the face-centered volumetric flux

$$F_f := \mathbf{v}_f\cdot\mathbf{S}_f, \tag{8}$$

where $\mathbf{v}_f$ is the second-order accurate face-centered velocity average and $\mathbf{e}_{\rho\mathbf{v}}(h^2)$ is the discretization error of the momentum convection term. The collocated UFVM denotes with f second-order accurate face-centered area-averaged quantities, and with c second-order accurate cell-centered volume-averaged quantities.

The mass flux $\rho_f F_f$ is constricted in an incompressible two-phase flow by volume conservation, i.e. the transport of the volume fraction, defined as a volumetric average of the phase-indicator function (1), i.e.

$$\alpha_c(t) := \frac{1}{|\Omega_c|}\int_{\Omega_c}\chi(\mathbf{x},t)\,dV. \tag{9}$$

The volume fraction advection equation [16]

$$\frac{d}{dt}\int_{\Omega_c}\chi\,dV = |\Omega_c|\frac{d}{dt}\alpha_c(t) = -\int_{\partial\Omega_c}\chi\mathbf{v}\cdot\mathbf{n}\,dS, \tag{10}$$

is equivalent to the advection equation for the phase $\Omega^+(t)$ indicated by $1-\chi(\mathbf{x},t)$, namely

$$\frac{d}{dt}\int_{\Omega_c}(1-\chi)\,dV = -|\Omega_c|\frac{d}{dt}\alpha_c(t) = -\int_{\partial\Omega_c}(1-\chi)\mathbf{v}\cdot\mathbf{n}\,dS, \tag{11}$$

since $\frac{d}{dt}\int_{\Omega_c}dV = 0$, as $\Omega_c \neq \Omega_c(t)$, and $\int_{\partial\Omega_c}\mathbf{v}\cdot\mathbf{n}\,dS = \int_{\Omega_c}\nabla\cdot\mathbf{v}\,dV = 0$ for incompressible two-phase flows without phase change.

The mass conservation with the single-field density Eq. (2) in a fixed control volume Ω_c states

$$\frac{d}{dt}\int_{\Omega_c}\rho\,dV = -\int_{\partial\Omega_c}\rho\mathbf{v}\cdot\mathbf{n}\,dS. \tag{12}$$

Inserting Eq. (2) into the r.h.s. of Eq. (12) and integrating over the time step $[t^n, t^{n+1}]$ results in

$$\rho_c^{n+1} = \rho_c^n + \frac{1}{|\Omega_c|}\left[-\rho^-\int_{t^n}^{t^{n+1}}\sum_{f\in F_c}\int_{S_f}\chi\mathbf{v}\cdot\mathbf{n}\,dS\,dt - \rho^+\int_{t^n}^{t^{n+1}}\sum_{f\in F_c}\int_{S_f}(1-\chi)\mathbf{v}\cdot\mathbf{n}\,dS\,dt,\right] \tag{13}$$

Inserting Eqs. (10) and (11) in Eq. (13) to replace the sums of surface integrals results in

$$\rho_c^{n+1} = \rho_c^n + \frac{\rho^-}{|\Omega_c|}|\Omega_c|\int_{t^n}^{t^{n+1}}\frac{d}{dt}\alpha_c(t)\,dt + \frac{\rho^+}{|\Omega_c|}|\Omega_c|\int_{t^n}^{t^{n+1}} -\frac{d}{dt}\alpha_c(t)\,dt, \tag{14}$$

leading finally to

$$\rho_c^{n+1} = \rho_c^n + (\rho^- - \rho^+)(\alpha_c^{n+1} - \alpha_c^n). \tag{15}$$

Eq. (15) shows that solving the mass conservation equation (12) for ρ_c^{n+1} is equivalent to solving the volume fraction Eq. (10) scaled with $(\rho^- - \rho^+)$.

Alternatively, integrating the single-field density model (2) over the control volume Ω_c gives

$$\int_{\Omega_c}\rho\,dV = \int_{\Omega_c}\rho^-\chi + \rho^+(1-\chi)\,dV. \tag{16}$$

Dividing Eq. (16) by $|\Omega_c|$ using Eq. (9) results in the discrete single-field density model,

$$\rho_c(t) = \rho^-\alpha_c(t) + \rho^+(1-\alpha_c(t)), \tag{17}$$

which, evaluated at t^{n+1} and t^n and subsequently subtracted, results in

$$\begin{aligned}\rho_c^{n+1} - \rho_c^n &= \rho^-\alpha_c^{n+1} + \rho^+(1-\alpha_c^{n+1}) - \rho^-\alpha_c^n - \rho^+(1-\alpha_c^n)\\ &= (\rho^- - \rho^+)(\alpha_c^{n+1} - \alpha_c^n),\end{aligned}$$

which is Eq. (15). Therefore, solving the mass conservation Eq. (12), solving the volume fraction advection Eq. (10) scaled by $(\rho^- - \rho^+)$, and using the single-field density model to compute the cell centered density from volume fractions by Eq. (2) are equivalent.

Note that all equations Eq. (9)–Eq. (17) are exact, as $\partial\Omega_c := \bigcup_{f\in F_c}S_f$, where S_f are non-linear surfaces that bound the control volume Ω_c, and reformulating the integration in time is exact by the fundamental theorem of calculus.

Ghods and Herrmann [23] introduce the auxiliary mass conservation equation as a means for ensuring the consistency of the two-phase momentum convection, and in Liu et al. [1] we provide the theoretical reasoning for the auxiliary mass conservation equation. Contrary to Ghods and Herrmann [23], Eq. (15) and Eq. (16) both demonstrate, that the solution of the volume fraction equation in the context of the VOF method [16] is exactly equivalent to the solution of the mass conservation equation, rendering an auxiliary mass conservation equation unnecessary.

However, the following question arises: if flux-based algebraic/geometric VOF methods inherently ensure numerical stability for the two-phase momentum convection with high density ratios, where do the numerical inconsistencies reported throughout the literature [24–29] come from?

Although Eqs. (15) and (16) show the inherent consistency between the volume and mass conservation in VOF methods in the mathematical model, the discrete computation of α_c^{n+1}, the approximation of the mass flux $\rho_f F_f$, and the choice of the temporal scheme and the flux limiting scheme, can potentially cause inconsistencies.

We focus first on the consistency between the mass and volume conservation equations on the discrete level. For a second-order accurate flux-based VOF method, the approximations applied to the temporal and surface integrals for the fluxed phase-specific volumes V_f^α when solving Eq. (10) lead to

$$\alpha_c^{n+1} = \alpha_c^n - \frac{1}{|\Omega_c|}\sum_{f\in F_c}\int_{t^n}^{t^{n+1}}\int_{S_f}\chi\mathbf{v}\cdot\mathbf{n}\,dS\,dt = \alpha_c^n - \frac{1}{|\Omega_c|}\sum_{f\in F_c}|V_f^\alpha|_s + e_{\alpha_t}(\Delta t^p) + e_{\alpha_h}(h^2), \tag{18}$$

with $e_{\alpha_t}(\Delta t^p)$ and $e_{\alpha_h}(h^2)$ as temporal and spatial volume fraction discretization errors. Note that

$$|V_f^\alpha|_s := sgn(F_f)|V_f^\alpha| \tag{19}$$

is a signed magnitude of a *phase-specific volume* V_f^α [16], whose sign is determined by the volumetric flux $\rho_f F_f$.

The phase-specific volume $|V_f^\alpha|_s$ can be used to approximate the mass flux $\rho_f F_f$ using Eq. (15) and Eq. (18). Reordering Eq. (18), results in

$$\alpha_c^{n+1} - \alpha_c^n = \frac{1}{|\Omega_c|}\sum_{f\in F_c}|V_f^\alpha|_s + e_{\alpha_t}(\Delta t^p) + e_{\alpha_h}(h^2) \tag{20}$$

Inserting Eq. (20) into Eq. (15) results in

$$\rho_c^{n+1} - \rho_c^n = \frac{\rho^- - \rho^+}{|\Omega_c|}\sum_{f\in F_c}|V_f^\alpha|_s + (\rho^- - \rho^+)[e_{\alpha_t}(\Delta t^p) + e_{\alpha_h}(h^2)]. \tag{21}$$

Equivalently, integrating Eq. (12) over $[t^n, t^{n+1}]$ results in

$$\begin{aligned}\rho_c^{n+1} - \rho_c^n &= \frac{1}{|\Omega_c|}\sum_{f\in F_c}\int_{t^n}^{t^{n+1}}\rho_f F_f\,dt \\ &= \frac{1}{|\Omega_c|}\sum_{f\in F_c}|M_f|_s + e_{\rho_t}(\Delta t^s) + e_{\rho_h}(h^2),\end{aligned} \tag{22}$$

with $e_{\rho_t}(t^s), e_{\rho_h}(h^2)$ as the temporal and spatial discretization errors of the mass conservation equation. The right-hand sides of Eqs. (21) and (22) express the mass fluxed through $\partial\Omega_c$ over $[t^n, t^{n+1}]$, i.e. $|M_f|_s$, as the phase-specific volume $|V_f^\alpha|_s$ scaled by the density difference, thus connecting the fluxed mass with the fluxed phase specific volume.

Consistency of mass conservation and volume conservation on the discrete level requires the equivalence of Eqs. (21) and (22). Eq. (21) and (22) will be exactly the same, only if their errors on the r.h.s. cancel out. Error cancellation is impossible if an auxiliary density equation is actually solved, and Eqs. (21) and (22) are using different numerical schemes when integrating $|V_f^\alpha|_s$ and $|M_f|_s$. In other words, if we use a specific VOF method for $|V_f^\alpha|_s$, we should compute the fluxed mass from $|V_f^\alpha|_s$. This is hypothetical, of course, since there is no need to actually solve two equations that are equivalent, and computing the new cell-centered density ρ_c^{n+1} using Eq. (17) from α_c^{n+1} suffices for VOF methods.

Consistency of the mass and volume conservation can be shown exemplary for a simplified first-order geometrical VOF method, which uses the "Euler" temporal integration (rectangle quadrature) of $|V_f^\alpha|_s$, i.e.

$$\begin{aligned}|V_f^\alpha|_s^{Euler} &= \int_{t^n}^{t^{n+1}}\int_{S_f}\chi\mathbf{v}\cdot\mathbf{n}\,dS\,dt = \frac{F_f^n}{|S_f|} \\ &\quad\times\int_{t^n}^{t^{n+1}}\int_{S_f}\chi\,dS\,dt + e_{\alpha_t}(\Delta t^2) + e_{\alpha_h}(h^2) \\ &= F_f^n\int_{t^n}^{t^{n+1}}\alpha_f(t)\,dt + e_{\alpha_t}(\Delta t^2) + e_{\alpha_h}(h^2) \\ &= F_f^n\alpha_f^n\Delta t + e_{\alpha_t}(\Delta t^2) + e_{\alpha_h}(h^2),\end{aligned} \tag{23}$$

where we define the fraction of the wetted face area $A_f(t)$ as $\alpha_f := A_f(t)/S_f$, and $N_t e_{\alpha_t}(\Delta t^2) = \frac{T}{\Delta t}e_{\alpha_t}(\Delta t^2) \propto \Delta t$ results in a first-order temporal quadrature error over the simulated physical time T. We emphasize that Eq. (23) is a simplified scheme used here only to discuss the consistency of mass and volume conservation on the discrete level, it is not a practically usable scheme for advecting α_c, because it is significantly less accurate than modern geometrical schemes [16]. Multiplying $|V_f^\alpha|_s^{Euler}$ from Eq. (23) with $(\rho^- - \rho^+)$ to obtain $|M_f|_s$ ensures the consistency of Eqs. (21) and (22), if Eq. (22) is integrated using the Euler scheme, namely

$$\frac{(\rho^- - \rho^+)|V_f^\alpha|_s^{Euler}}{\Delta t} = (\rho^- - \rho^+)F_f^n\alpha_f^n + (\rho^- - \rho^+)(e_{\alpha_t}(\Delta t) + \frac{1}{\Delta t}e_{\alpha_h}(h^2)). \tag{24}$$

The temporal accuracy lost by dividing by Δt is recovered when the mass flux is integrated over (multiplied with) Δt, from the temporal term in the momentum conservation equation.

We move on to the approximation of the mass flux in the discretized convective term of the momentum equation.

The unstructured Finite Volume Method linearizes the volumetric flux when discretizing the convective term in Eq. (5), introducing outer iterations $\varepsilon o\varepsilon$ to the equation solution algorithm. We further use an Euler implicit temporal discretization, for example, which leads to

$$\frac{\Delta t}{|\Omega_c|}\left(\int_{\Omega_c}\nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v})\,dV\right)_{t^{n+1}} \approx \frac{\Delta t}{|\Omega_c|}\sum_{f\in F_c}\rho_f^o F_f^o\mathbf{v}_\mathbf{f}^{o+1}, \tag{25}$$

with the factor $\frac{\Delta t}{|\Omega_c|}$ resulting from the finite-difference approximation of the cell-centered average of the temporal derivative term, i.e. $(\partial_t\rho\mathbf{v})_c$, recovering the first-order temporal accuracy in Eq. (24) for the implicit Euler temporal discretization scheme. Note that Eq. (5) is solved (discretized) iteratively in a segregated solution algorithm (e.g., [1,30]), so the linearized mass flux is denoted with the outer iteration index $1 \le o \le N_o$. Once the solution algorithm converges, $\rho_f^o F_f^o = \rho_f^{n+1}F_f^{n+1}$.

The mass flux $\rho_f^o F_f^o$ is uniquely determined by $\chi(\mathbf{x},t)$ and $\mathbf{v}(\mathbf{x},t)$ [1]. Namely, at any time t, omitted here for brevity, from Eq. (2), we get

$$\int_{S_f}\rho\mathbf{v}\cdot\mathbf{n}\,dS =: (\rho_f F_f)^\rho \approx \frac{F_f}{|S_f|}\int_{S_f}[(\rho^- - \rho^+)\chi(\mathbf{x}) + \rho^+]\,dS = (\rho^- - \rho^+)\alpha_f F_f + \rho^+ F_f \tag{26}$$

with superscript ρ in $(\rho_f F_f)^\rho$ denoting the mass flux estimated directly from the model for ρ(2) and the volumetric flux given by Eq. (8).

The contribution $\rho^+ F_f$ to Eq. (26) and its role in the approximation of the mass flux in the momentum equation must be carefully addressed.

The term $\rho^+ F_f$ is a zero-sum contribution to mass conservation (12), and therefore, volume conservation (15). Namely, inserting

$$\rho\mathbf{v} = (\rho^- - \rho^+)\chi\mathbf{v} + \rho^+\mathbf{v} \tag{27}$$

from Eq. (2) multiplied by $\mathbf{v}$, into Eq. (12), and integrating over $[t^n, t^{n+1}]$, and applying divergence-free velocity condition (4) results in

$$\begin{aligned}
\rho_c^{n+1} &= \rho_c^n - \frac{1}{|\Omega_c|}\int_{t^n}^{t^{n+1}}\left[\int_{\partial\Omega_c}(\rho^- - \rho^+)\chi\mathbf{v}\cdot\mathbf{n}\,dS\,dt + \rho^+\int_{\partial\Omega_c}\mathbf{v}\cdot\mathbf{n}\,dS\right]dt \\
&= \rho_c^n - \frac{1}{|\Omega_c|}\int_{t^n}^{t^{n+1}}\left[\int_{\partial\Omega_c}(\rho^- - \rho^+)\chi\mathbf{v}\cdot\mathbf{n}\,dS + \rho^+\int_{\Omega_c}(\nabla\cdot\mathbf{v})dV\right]dt \\
&= \rho_c^n - \frac{1}{|\Omega_c|}\int_{t^n}^{t^{n+1}}\left[\int_{\partial\Omega_c}(\rho^- - \rho^+)\chi\mathbf{v}\cdot\mathbf{n}\,dS\right]dt \\
&= \rho_c^n - \frac{(\rho^- - \rho^+)}{|\Omega_c|}\sum_{f\in F_c}\int_{t^n}^{t^{n+1}}\chi\mathbf{v}\cdot\mathbf{n}\,dS \\
&= \rho_c^n - \frac{(\rho^- - \rho^+)}{|\Omega_c|}\sum_{f\in F_c}|V_f^\alpha|_s.
\end{aligned} \tag{28}$$

The main point of Eq. (28) (and equivalently Eqs. (21) and (22)) is that the additional term $\rho^+\mathbf{v}$, discrete ρ^+F_f in Eq. (26), does not impact consistency of mass and volume conservation given by Eqs. (21) and (22). More importantly, Eq. (28) shows this is true irrespective of the VOF method used to approximate $|V_f^\alpha|_s$, in our case, the plicRDF-isoAdvector method [18], or, in the example from Eq. (23), $|V_f^\alpha|_s^{Euler}$.

However, while ρ^+F_f from Eq. (26) is a zero contribution in mass and volume conservation Eqs. (21), (22) and (28), it is a non-zero contribution to the discretized convective term (25), i.e.

$$\frac{\Delta t}{|\Omega_c|}\sum_{f\in F_c}\rho_f^o F_f^o\mathbf{v}_\mathbf{f}^{n+1} = \frac{\Delta t}{|\Omega_c|}\left(\sum_{f\in F_c}(\rho^- - \rho^+)\alpha_f^o F_f^o\mathbf{v}_f^{n+1} + \rho^+\sum_{f\in F_c}F_f^o\mathbf{v}_f^{n+1}\right), \tag{29}$$

because it models the single-phase momentum convection in the bulk of each phase $\Omega^\pm(t)$.

The final choice of approximating $(\rho^- - \rho^+)\alpha_f^o F_f^o$ is constricted by accuracy and the temporal scheme chosen for Navier–Stokes equations. Concretely, if the Euler implicit scheme is used to discretize Navier–Stokes equations, scaling the fluxed phase-specific volume $|V_f^\alpha|_s$ in Eq. (24) to approximate $(\rho^- - \rho^+)\alpha_f^o F_f^o$ will be consistent, with an Euler-implicit discretization of an equivalent mass conservation equation. The consistency of the mass conservation and volume fraction advection is crucial in the momentum conservation equation, as a deviation from volume conservation increases the source-term $\sum_{f\in F_c}\left(\frac{1}{a_c}\right)_f\left[\mathbf{H}(F^o,\rho^o,\mathbf{v}^{i-1})\right]_f\cdot\mathbf{S}_f$ in the pressure Poisson equation, with i denoting the inner iterations of the pressure equation (cf. [1,30] for details). In other words, errors in the mass flux artificially accelerate or decelerate the fluid as the pressure equation tries to ensure volume (mass) conservation. Examining the errors in Eq. (24), we see that an error in the mass flux scaled from $|V_f^\alpha|_s$ will be multiplied by $(\rho^- - \rho^+)$: small errors in the volumetric flux (fluxed phase-specific volume) are scaled with the density difference, and this leads to large errors in the velocity field and catastrophic failures for large density differences. Therefore, a more accurate $|V_f^\alpha|_s$, e.g., a second-order geometrical $|V_f^\alpha|_s^{isoAdvector}$, results in a more accurately ensured consistency between mass and volume conservation (28).

However, there is a downside in approximating the mass flux $\rho_f^o F_f^o$ by scaling a highly accurate geometrical phase-specific volume, i.e., $|V_f^\alpha|_s^{isoAdvector}$. Consider an alternative temporal integration scheme for Eq. (5), say, Crank–Nicolson scheme, resulting in contributions from the convective term in the form of

$$0.5\frac{\Delta t}{|\Omega_c|}\left(\sum_{f\in F_c}\rho_f^n F_f^n\mathbf{v}_\mathbf{f}^n + \sum_{f\in F_c}\rho_f^o F_f^o\mathbf{v}_\mathbf{f}^{n+1}\right) \tag{30}$$

In this case, it is impossible to ensure consistency by scaling a single value of $|V_f^\alpha|_s$ to obtain two values $\rho_f^n F_f^n$ and $\rho_f^o F_f^o$. The phase-specific volume is integrated in time at t^n over the time step Δt, making it a constant value over the time step, and we obtain $(\rho_f F_f)^{Euler}$ from Eq. (24) as a single average quantity over Δt, so we do not have two mass fluxes for Eq. (30), making Eq. (24) inconsistent with the Crank–Nicolson scheme Eq. (30) already in the first time step.

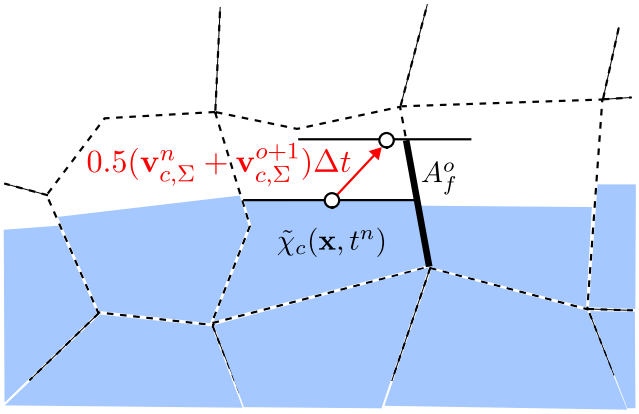


Fig. 2. Geometric upwinding for $|V_f^\alpha|_s^{isoAdvector}$ in plicRDF-isoAdvector [18].

The main takeway point is that computing the mass flux by scaling the fluxed phase-specific volume over Δt limits the temporal discretization to schemes that utilize a single mass flux term within Δt - e.g., Euler explicit or implicit, or 2nd-order backward implicit schemes. Any other temporal discretization scheme (e.g., Runge–Kutta) that utilizes a linear combination of different mass fluxes within Δt are inconsistent with the mass flux scaling given by Eq. (24).

Additionally, any modification of the scaled mass flux causes inconsistencies. Concretely, *limiting the mass flux* in the discretized Eq. (5) causes a hidden inconsistency between mass and volume fraction conservation. The inconsistency is hidden because limiting the mass flux in Eq. (25) creates a mass flux that does not correspond to the mass and volume conservative flux used to update ρ_c^o from α_c^o in Eq. (15), i.e., the consistent approximation of $(\rho^- - \rho^+)\alpha_f^o F_f^o$ as a part of the total mass flux in Eq. (21).

We now turn our attention from the simplified VOF scheme Eq. (23) to a more complex, geometric isoAdvector scheme [17], that computes

$$\begin{aligned}
|V_f^\alpha|_s^{isoAdvector} &= \frac{0.5(F_f^n + F_f^o)}{|\mathbf{S}_f|}\int_{t^n}^{t^o}\int_{S_f}\chi(\mathbf{x},t)\,dS\,dt + e_{\alpha_t}(\Delta t^3) + e_{\alpha_h}(h^2) \\
&= \frac{0.5(F_f^n + F_f^o)}{|\mathbf{S}_f|}\int_{t^n}^{t^o}A_f(t)\,dt + e_{\alpha_t}(\Delta t^3) + e_{\alpha_h}(h^2) \\
&= 0.5(F_f^n + F_f^o)\int_{t^n}^{t^o}\alpha_f(t)\,dt + e_{\alpha_t}(\Delta t^3) + e_{\alpha_h}(h^2)
\end{aligned} \tag{31}$$

with $\int_{t^n}^{t^o}\alpha_f(t)dt$ computed geometrically by kinematically tracking the piecewise-linear interface from the upwind cell, using the interpolated interface velocity $\mathbf{v}_{\Sigma,c}$ associated with the PLIC centroid and the second-order accurate displacement approximation $0.5(\mathbf{v}_{\Sigma,c}^n + \mathbf{v}_{\Sigma,c}^o)\Delta t$, schematically shown in Fig. 2. If the implicit Euler temporal scheme is used for Navier–Stokes equations, the Euler temporal integration of the displacement $\mathbf{v}_c\Delta t$ for the evaluation of $\int_{t^n}^{t^o}\alpha_f(t)dt$ on the r.h.s. of Eq. (31) makes it possible to consistently compute the $(\rho^- - \rho^+)\alpha_f F_f$ contribution to the total mass flux $\rho_f F_f$ by scaling the phase-specific fluxed volume

$$((\rho^- - \rho^+)F_f^o\alpha_f^o)^{isoAdvector} = \frac{(\rho^- - \rho^+)|V_f^\alpha|_s^{isoAdvector}}{\Delta t}, \tag{32}$$

equivalently to Eq. (24). Even though $\int_{t^n}^{t^o}\alpha_f(t)dt$ is evaluated using exact geometric integration (cf. [18] for details), the A_f^o, used as the end-point of the geometric integration is approximated by first-order

accurate displacement when implicit Euler scheme is used for the Navier–Stokes equations, making the average displacement velocity of the fluid interface constant over Δt, and, consequently, the mass flux contribution from Eq. (32) consistent w.r.t. Eq. (15) and Eq. (21).

The same consistency would be ensured for the implicit Euler method, if $|V_f^\alpha|_s$ would be constructed using a flux-based VOF method that maps S_f using the reverse flow-map (e.g., [31]), *if the approximation of the flow-map is first-order accurate*. In other words, if $|V_f^\alpha|_s$ is constructed geometrically using displacements given by $\mathbf{v}(\mathbf{x}, t^n)\Delta_t$, regardless of the geometrical approximation, $|V_f^\alpha|_s$ is constructed using displacements constant over Δt, so dividing the volume V_f^α with Δt results in a volumetric flux consistent with the Euler method, and constant over Δt. In following sections, we describe in detail the consistency and equivalency of the scaled $|V_f^\alpha|_s$ and the solution of an auxiliary density equation, in the context of the solution algorithm for one-field Navier–Stokes equations, for the implicit Euler temporal discretization, and confirm our findings with extensive verification and validation.

3.2. Collocated segregated solution algorithm with the auxiliary density equation

We base our consistent solution algorithms for the unstructured Volume-of-Fluid methods on the plicRDF-isoAdvector method [18] and implement the auxiliary density equation solution into the segregated algorithm (i.e. solver) "interIsoRhoFoam", which is summarized by Algorithm 1.

In [1], we show on the level of the mathematical model why solving an auxiliary mass conservation equation plays a key role in reducing numerical inconsistency caused by high density ratios.

We solve the mass conservation equation in the outer loop of the segregated algorithm Algorithm 1 in the following discrete form

$$\rho_c^{o+1} = \rho_c^o + \frac{\Delta t}{|V_{\Omega_c}|}\sum_f \rho_f^o F_f^o. \tag{33}$$

Eq. (33) is the auxiliary mass conservation (density equation). It is solved after updating the volume fraction in the outer loop to α_c^o by utilizing Eq. (18) with any flux-based VOF method. Interface reconstruction computes $\tilde{\chi}_c(\mathbf{x}, t^o)$ in every finite volume Ω_c intersected by the fluid interface. The piecewise-linear interface approximation $\tilde{\chi}_c(\mathbf{x}, t^o)$, together with Eq. (2), provides ρ_f^o for Eq. (33). Since F_f^o is also available, Eq. (33) can be explicitly evaluated. To compute the mass flux, we utilize the consistency relationship between the phase indicator and the face-centered density ρ_f, derived in [1] for the Level Set/Front Tracking method. It can be applied on any two-phase flow simulation method that is using a phase indicator χ. The surface integral of mass flux at time step o is expressed as

$$\int_{\partial\Omega_c} \rho^o \mathbf{v}^o \cdot \mathbf{n} dS = \int_{\partial\Omega_c} [\rho^- \chi^o + \rho^+(1-\chi^o)]\mathbf{v}^o \cdot \mathbf{n}\, dS, \tag{34}$$

and discretized as

$$\sum_{f\in F_c} \rho_f^o F_f^o = \sum_{f\in F_c} \left[\rho^- \alpha_f^o + \rho^+(1-\alpha_f^o)\right] F_f^o, \tag{35}$$

where

$$\alpha_f^o = \frac{1}{|S_f|}\int_{S_f} \chi(\mathbf{x}, t^o)\, dS = \frac{|A_f(t^o)|}{|S_f|}. \tag{36}$$

is used to define the face-centered density

$$\rho_f^o = \rho^-\alpha_f^o + \rho^+(1-\alpha_f^o) \tag{37}$$

in the mass flux. The momentum equation is discretized as

$$\rho_c^o \mathbf{v}^o - \rho_c^n \mathbf{v}^n + \frac{\Delta t}{|\Omega_c|}\sum_f \rho_f^o F_f^o \mathbf{v}_f^o = \frac{\Delta t}{|\Omega_c|}\mathbf{M}. \tag{38}$$

The source term $\mathbf{M}$ is a shorthand term that contains all the remaining contributions from the discretization, used here for brevity. The volume fractions α_c^o are already solved for using Eq. (18) at the beginning of the "o" outer iteration, and used to approximate the phase indicator $\tilde{\chi}_c(\mathbf{x}, t^o)$ for the mass flux $\rho_f^o F_f^o$ using Eq. (35), used in the same way in the auxiliary density equation Eq. (33), solved for ρ_c^o, as $\rho_f^o F_f^o$ in Eq. (38), *without using flux limiters*. The values computed from the last outer loop are regarded as the new values at the time step. At last, we need to restore the density field with respect to the volume fraction to maintain consistency between them. Algorithm 1 uses a combination of SIMPLE and PISO algorithms in OpenFOAM with a residual-based control to terminate outer iterations, which tests if a maximal number of iterations has been reached, the final domain-maximal residuals of the pressure equation r_p are below absolute tolerance, or the ratio of the final and initial domain-maximal residuals $\frac{r_p}{r_p^i}$ is smaller than a user-prescribed relative tolerance.

Algorithm 1 The solution algorithm interIsoRhoFoam.

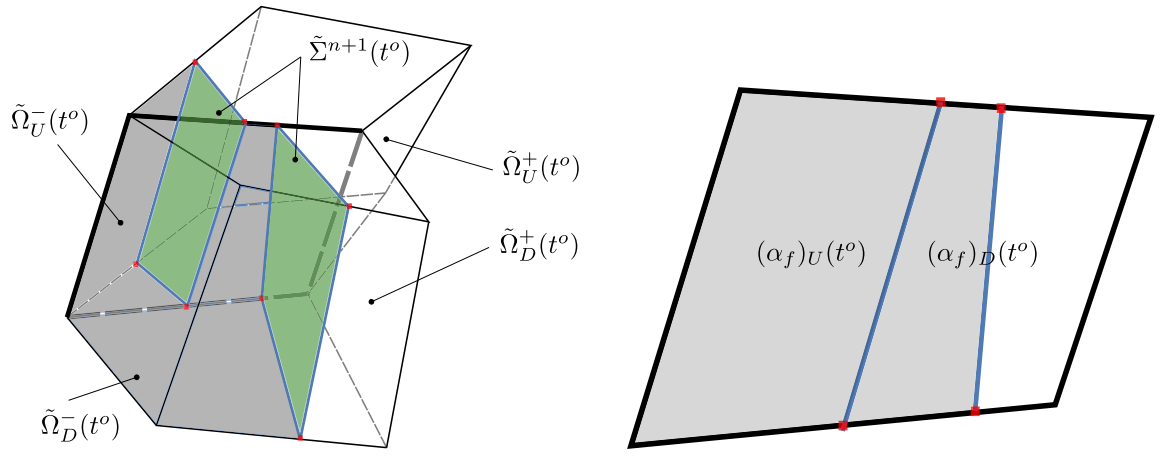
```
while t ≤ t_end do
    t^{n+1} = t^n + Δt
    o = 0
    while o < N_outer or max(r_p) > t_p or r_p/r_p^i > t_p^r do
        o = o + 1
        Solve volume fraction for α_c^o                          ▷ Equation (18)
        Reconstruct the phase indicator χ̃_c(x, t^o)
        Compute ρ_f^o from χ̃_f(x, t^o)                           ▷ Equation (37)
        Compute the mass flux ρ_f^o F_f^o := ρ_f^o(v_f^{o-1} · S_f)
        Solve the density for ρ_c^o with ρ_f^o F_f^o             ▷ Equation (33)
        Discretize momentum equation (Equation (5)) with ρ_c^o and ρ_f^o F_f^o.
        for i = 0; i < N_inner; ++i do
            Solve the pressure equation for p_c^i.               ▷ Cf. [1,30].
            Compute F_f^i and v_c^i from p_c^i.                  ▷ Cf. [1,30].
        end for
    end while
    Restore ρ_c^{n+1} consistent with α_c^{n+1}, i.e. ρ_c^{n+1} = (ρ^- − ρ^+)α_c^{n+1} + ρ^+   ▷ Equation (2).
end while
```

3.3. Phase-specific face area calculation

The integral Eq. (36) leaves room for alternative discretizations, and therefore requires attention. Flux-based geometrical Volume-of-Fluid methods advect volume fractions using geometrical upwind advection schemes, transporting $\tilde{\Omega}^-(t) \approx \Omega^-(t)$ geometrically from the upwind cell, denoted with U in Fig. 3(a), to the downwind cell, denoted with D in Fig. 3(a). Flux-based geometrical VOF methods therefore already provide the intersection points and the intersection line segments denoted as red dots and the blue line segments in Fig. 3, are available, simplifying the area fraction calculation in Eq. (36). Since the geometrical VOF methods approximate the phase indicator as piecewise continuous, with a jump discontinuity not only across the fluid interface $\Sigma(t)$ but also across the finite volume boundary $\partial\Omega_c$, the line segments forming the intersection of the piece-wise continuous interface and a face S_f it intersects (see Fig. 3(a)), do not overlap. Fig. 3(b) shows schematically the two intersection line segments on a cell face. The intersection line segment of the interface and the upwind cell are selected as the submerged area border to calculate the area fraction. The evaluation of α_f and, from it, ρ_f, μ_f, is summarized by Algorithm 2.

Algorithm 2 Sub-algorithm for calculating α_f, ρ_f, μ_f.

```
Initialize ρ_f, α_f with upwind cell-centered values ρ_U, α_U.
α_f = (1/|S_f|) ∫_{S_f} χ̃_U(x) dS = |Ω̃_U^- ∩ S_f| / |S_f|     ▷ Equation (36) with upwind χ.
ρ_f = (ρ^- − ρ^+)α_f + ρ^+.                                    ▷ Equation (2).
μ_f = (ρ^-ν^- − ρ^+ν^+)α_f + ρ^+ν^+.                           ▷ Equation (3).
```



(a) Upwind and downwind cells intersected with piecewise-linear VOF interfaces.

(b) A face S_f intersected with downwind (D) and upwind (U) piecewise-linear VOF interfaces.

Fig. 3. Interface reconstructed as $\tilde{\Sigma}^o_{U,D}$ in upwind (U) and downwind (D) cells. Green polygons are interface polygons $\tilde{\Sigma}^o_{U,D} \cap \Omega_{U,D}$. Blue lines are intersection line segments $\tilde{\Sigma}^o_{U,D} \cap S_f$. Red points are intersection points $\tilde{\Sigma}^o_{U,D} \cap \partial S_f$.

It is important to note that μ_f is geometrically evaluated from a geometrical α_f, and not interpolated, as we found that interpolation leads to large errors in simulations with large differences in dynamic viscosity.

3.4. Consistency of VOF methods for two-phase flows with high density ratios

Here we summarize our findings that lead to an equation discretization with flux-based geometrical VOF methods that remains consistent for very high density ratios:

- Computing the mass flux $\rho_f F_f$ by scaling the fluxed phase-specific volume $|V_f^\alpha|_s$ with $(\rho^- - \rho^+)/\Delta t$ approximates only one constant average mass flux over Δt.
- The constant mass flux $\rho_f F_f$ scaled from $|V_f^\alpha|_s$ with Δt , as an average value over Δt, can only be consistently used in first-order schemes, i.e. use a single mass flux value over Δt.
- Computing the mass flux $\rho_f F_f$ from the density model Eq. (2) disconnects mass conservation from volume fraction advection, which uses geometrical integration, i.e. a geometrically integrated phase-specific volumetric flux. This requires a solution of an additional (auxiliary) density equation [1,23] for cell-centered density, discarded at the end of the time step.
- Upwinding geometric VOF methods that use the Euler temporal scheme to approximate point displacements are equivalent to using $\rho_f F_f$ from Eq. (26) and solving the density equation for ρ_c^o, used further in $p - \mathbf{v}$ coupling to obtain a divergence-free cell-centered velocity. In other words, combining Euler temporal integration scheme with upwind scheme for the momentum equation guarantees numerical consistency for any flux-based VOF method, which uses temporally first-order accurate displacements in its geometrical integration of the fluxed phase-specific volume.

These points we verify and validate in the following section.

4. Verification and validation

Data archives of the implementation of the interIsoRhoFoam algorithm, input data, post-processing software and secondary data are publicly available [32,33]. The method is actively developed in a publicly available git repository [34].

4.1. Time step size

The time step is limited by the CFL condition in the explicit plicRDF-isoAdvector method [18],

$$\Delta t_{CFL} = \frac{CFL\ h}{|\mathbf{v}|}, \tag{39}$$

where h is the discretization length, and we use $CFL = 0.2$ from [18]. Another restriction for the time step size considers the propagation of capillary waves on fluid interfaces,

$$\Delta t_{cw} = \sqrt{\frac{(\rho^+ + \rho^-)h^3}{2\pi\sigma}}. \tag{40}$$

This time step constraint was introduced first by Brackbill et al. [35], and revised by Denner and van Wachem [36]. We restrict the time step using the relation from Tolle et al. [30], namely

$$\Delta t = min\,(k_1 \Delta t_{cw},\ k_2 \Delta t_{CFL}) \tag{41}$$

where k_1 and k_2 are scaling factors. We set $k_1 = 1$, $k_2 = 0.5$ as the default value.

4.2. Translating droplet in ambient flow

A canonical test case, originally introduced by Bussmann et al. [37] in 2D, involves a moving droplet in quiescent ambient flow. Zuzio et al. [25] extended the 2D case to 3D using a density ratio of 10^6. Following the setup in [25], the droplet of radius $R = 0.15$ has the initial velocity of $(0, 0, 10)$. To smooth the velocity field and avoid the perturbation caused by sudden acceleration of still ambient flow, the initial constant velocity is assigned not only for the cells of the droplet and the interface layer but also the interface cell layer adjacent to interface cells. The droplet with initial center location (0.5 0.5 0.5) translates to a distance of $L = 1$ and the simulation time is then $t_{end} = 0.1$. The computational domain has dimensions $L_z = L_x = L_y = 1$. The periodic boundary condition is applied to all boundary patches. During testing with the periodic boundary condition, we found that the plicRDF-isoAdvector method [18] implemented in [38] has an inconsistency at the periodic boundary, which we fixed as described in Appendix A. The mesh setup from [25] is also followed: the mesh resolution is in a range of $N \in (32,\ 48,\ 64)$ per unit side-length of the computational domain, resulting in $\approx$ (10, 15, 20) mesh cells per droplet diameter. Surface tension force is neglected in this case, so only Δt_{CFL} from Eq. (41) is taken into account. The viscosity and gravitational forces are neglected to

highlight the numerically consistent behavior of the mass and momentum convection. We adopted three error norms to evaluate the results quantitatively, for mass, momentum, and sphericity:

$$E_{mass} = \frac{M(t) - M(0)}{M(0)} = \frac{\sum_k m_k(t) - \sum_k m_k(0)}{\sum_k m_k(0)} = \frac{\sum_k \rho_k(t) V_k}{\sum_k \rho_k(0) V_k}, \tag{42}$$

$$E_{mom} = \frac{|\sum_k m_k(t)\mathbf{v}_k(t)| - |\sum_k m_k(0)\mathbf{v}_k(0)|}{|\sum_k m_k(0)\mathbf{v}_k(0)|}, \tag{43}$$

$$E_{sph} = \left| \sum_{c \in C} S_c(t) - \sum_{c \in C} S_c(0) \right|, \tag{44}$$

where the subscript k indicates that the value is extracted from the cell Ω_k and the m_k, V_k denote the mass and volume of Ω_k. In Eq. (44), S_c is the area of the PLIC-VOF interface polygon in the cell Ω_c. Eqs. (42) and (43) represent the time evolution of the normalized error of the global sum of the heavy phase mass and momentum. In this case, E_{mass} is expected to be near zero because there is no source and dissipation for both mass, and flux-based geometrical VOF method have a very high degree of local volume (mass) conservation.

In the absence of force terms on both sides of momentum transport equation, i.e, Eq. (5), assuming the periodic boundary condition is applied to all boundary patches, momentum conservation dictates that the deviation E_{mom} should also theoretically remain at zero over time. Bussmann et al. [37] proposed that a droplet, characterized by a large density ratio (10^6), should undergo translation without deformation in an ambient flow, much like a solid sphere moving through a void. This conclusion has been widely accepted and corroborated by several publications (e.g., [23,39–42]). However, these studies qualitatively assessed droplet deformation based on visual representations of droplet shape. In this work, we employ the sphericity error E_{sph} as a quantitative measure, to characterize the deviation from the initial droplet shape.

As discussed in Section 3.4, when the first-order accurate Euler and Gauss upwind scheme are employed to discretize momentum conservation Eq. (5), the mass flux $(\rho_f F_f)^{isoAdvector}$ from Eq. (32) will be consistent. Since the choice of discretization schemes ensures consistency of the discretization, there is no need to modify the implementation of the numerical method. We have verified the analysis from Section 3 for the "interIsoFoam" solver and compared it with the "interIsoRhoFoam" solution Algorithm 1 that implements the auxiliary density equation. The normalized mass error Eq. (42) is nearing machine epsilon, and is therefore much smaller than than the linear solver tolerance (set for this case to 10^{-12}), for both interIsoFoam and interIsoRhoFoam, showing excellent conservation of mass for both configurations. The consistency of the mass and phase indicator transport, as well as the consistency of the mass flux approximation of the interIsoFoam and interIsoRhoFoam with Euler+upwind schemes is reflected in an equivalent accuracy and stability for the momentum: Eq. (43) remains equally much smaller than the linear solver tolerance, nearing machine epsilon, and remains stable. We therefore verify the consistency and equivalence of the "interIsoFoam" and "interIsoRhoFoam", provided the Euler+upwind schemes are used.

Note that this verification case is extremely challenging, since it is an inviscid case — there is no viscous force available in this case to dampen the errors resulting from inconsistent two-phase mass and momentum transport.

Next, we demonstrate inconsistencies leading to large errors and often to catastrophic failure when more than one mass flux is used over Δt when discretizing Eq. (5), or the mass flux is limited in the discretized Eq. (5).

4.2.1. Comparison of different schemes

Using the Crank–Nicolson scheme to discretize Eq. (5) reveals that the temporal scheme involving implicit mass flux $\rho_f F_f$ results in a mismatch between mass convection and volume fraction convection scaled by $(\rho^- - \rho^+)$, due to the fact that the Navier–Stokes equation using a segregated method is solved iteratively within a time step, and the interface's advection velocity thus cannot be updated simultaneously with $\mathbf{v}^{n+1}$ from the previous $p - v$ coupling iteration. The inconsistency is amplified by the density-ratio. Combinations of schemes listed in Table 1, are tested to verify their effect on the mass flux inconsistency.

Figs. 4 and 5 represent the temporal evolution of E_{mass}, and E_{mom}. It is evident from Fig. 4 that mass conservation is not maintained when utilizing the Crank–Nicolson time discretization scheme, and divergence schemes cubic, Linear-Upwind Stabilized Transport (LUST) and Quadratic Upstream Interpolation for Convective Kinematics (QUICK). These simulations terminate at an early stage with catastrophic failure. However, for cases that can run until the final time, the magnitude of mass errors, as shown in the zoomed subfigure of Fig. 4, is on the order of 10^{-10}, indicating mass conservation. This observation aligns with the inherent characteristic of the volume of fluid method, which is known for its mass conservation property. The vertical lines from the results of cubic, QUICK and Crank–Nicolson can be observed in Fig. 5. Some combinations deliver stable momentum errors. A closer examination of the stable cases in the zoomed view of Fig. 5 confirms the use of the Euler temporal scheme.

The temporal evolution of sphericity error is depicted in Fig. 6 where the steepness of the curves indicates the extent of droplet deformation from its initial shape. The results obtained from four unstable scheme combinations show significantly larger deviations in sphericity. Moreover, the zoomed view reveals that, apart from Euler+ upwind, all other stable scheme combinations display varying degrees of divergence. This suggests that if the simulations were to continue for a longer duration, these cases would likely crash. Fig. 7 presents the final shapes of the droplets simulated using interIsoFoam with two stable scheme combinations and two unstable combinations. For the unstable combinations, a common occurrence is observed: the original droplets disintegrate into small, irregularly scattered pieces. A comparison between the final shapes obtained using the inconsistent but in this verification case, for these parameters, still stable scheme combination Euler + limitedLinearV (Fig. 7(c)) and the consistent scheme combination Euler + upwind (Fig. 7(d)) corroborates the findings from the sphericity errors in Fig. 6: an evident "crown" with irregular bumps forms on the top part of the droplet in Fig. 7(c), leading to a reduction in sphericity. On the contrary, in Fig. 7(d), only a slight shrinkage occurs in the neck region of the droplet, while the top surface remains smooth.

Table 2 summarizes these findings, providing information on early termination of scheme combinations that are inconsistent, as well as mass, momentum and sphericity errors for different mesh resolutions. Two finer resolutions, i.e., $N \in (96, 128)$, are tested for interIsoFoam with Euler+upwind schemes to verify the mesh convergence of the consistent method, reported in Fig. 8. All scheme combinations are tested with interIsoFoam with density ratio 1. As shown in Fig. B.23, all cases with the same densities remain stable.

4.3. Translating sub-millimeter droplet with realistic physical properties

The realistic densities of the mercury droplet and air ambient pair are selected for the verification case of the translating sub-millimeter droplet. The density ratio displayed in Table 3 is around 10^4. The rest setups are same as in [1]. A spherical droplet of radius $R = 0.25\,\text{mm}$ translates a distance of three diameters with velocity $0.01\,\text{m/s}$ in z-direction of the rectangular solution domain ($L_x = L_y = 5R, L_z = 15R$). The ambient flow has the same velocity, that is $\mathbf{v}_a = (0, 0, 0.01)$. Three resolutions are tested in this case: $N \in (16,\ 32,\ 64,\ 96,\ 128)$. The initial centroid position of the droplet is $(2.5R, 2.5R, 2R)$. Surface tension and viscous forces are not considered in this case. Since the droplet translates with the ambient flow and there is no sink or source for the droplet moving, the velocity field should keep unchanged.

The error norm L_∞ is employed to measure the maximal deviation between the numerical velocity and the analytical one among all cells,

Table 1
The combinations of different time and convection schemes used to test the effect of numerical consistency.

Time schemes	Gaussian convection schemes	Order of convection accuracy	Category of convection	Boundedness of convection	Mass flux consistency
Crank–Nicolson [43]	upwind	first-order	NVD/TVD	Bounded	no
Euler	upwind	first-order	NVD/TVD	Bounded	yes
	cubic [44]	second-order	non-NVD/TVD	Unbounded	no
	limitedLinearV	first-/second-order	NVD/TVD	Unbounded	no
	linear	second-order	non-NVD/TVD	Unbounded	no
	LUST	second-order	non-NVD/TVD	Unbounded	no
	MUSCL [45]	second-order	NVD/TVD	Unbounded	no
	QUICK [46]	second-order	NVD/TVD	Unbounded	no
	SuperBee [47]	second-order	NVD/TVD	Unbounded	no
	vanLeer [48]	second-order	NVD/TVD	Unbounded	no

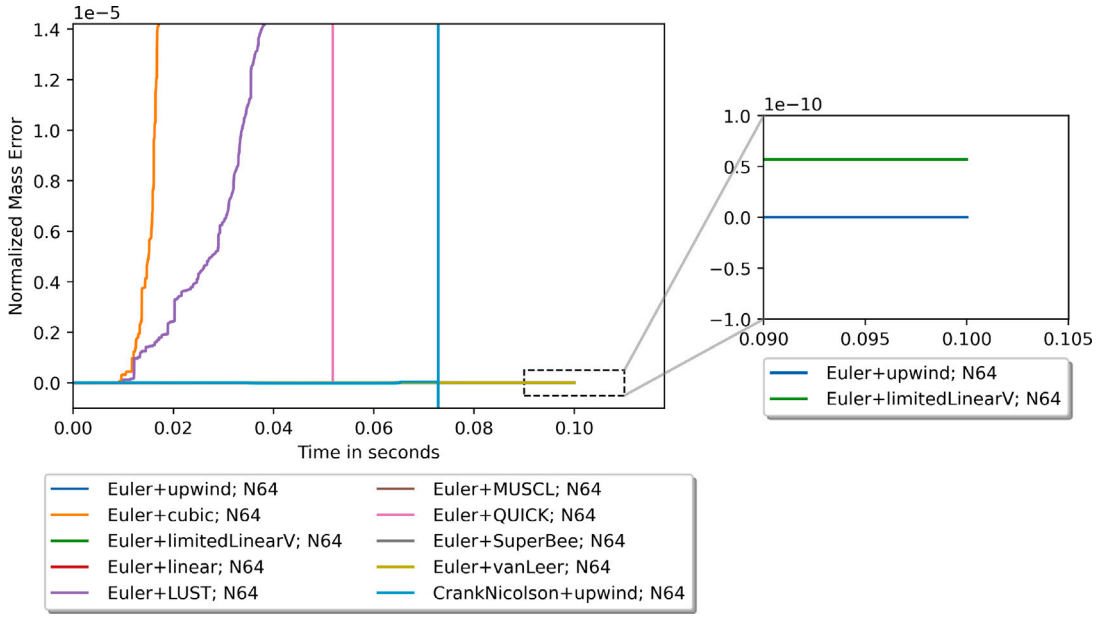


Fig. 4. Temporal evolution of normalized mass conservation error with different schemes: interIsoFoam, $N = 64$.

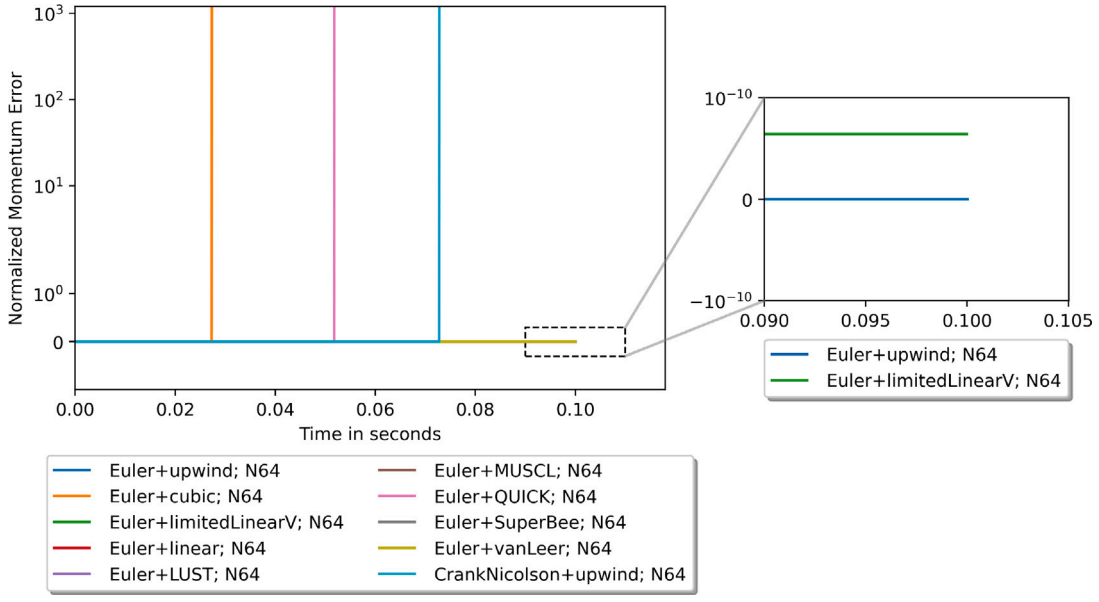


Fig. 5. Temporal evolution of normalized momentum conservation error with different schemes: interIsoFoam, $N = 64$.

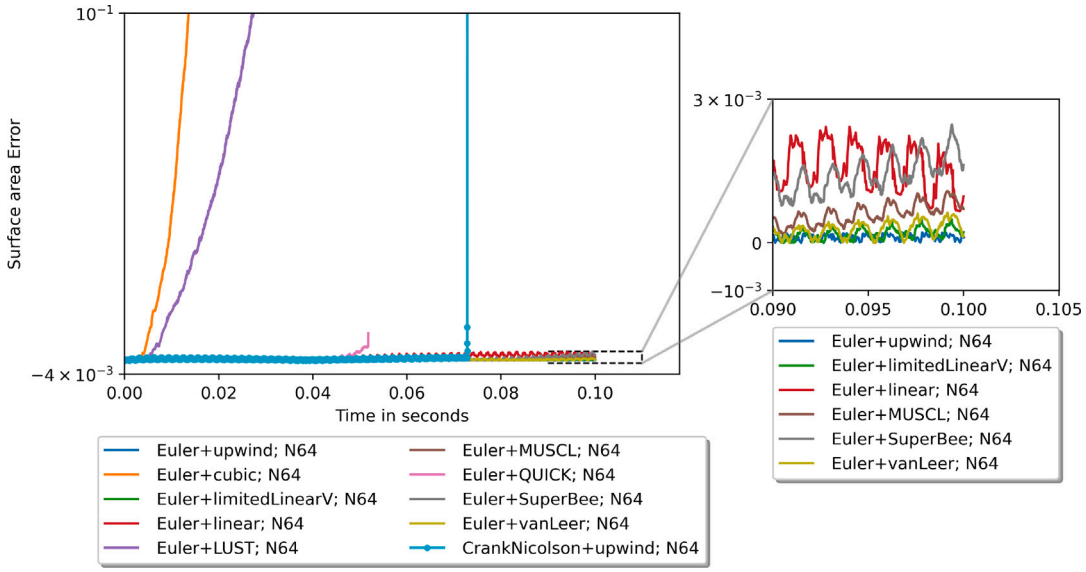
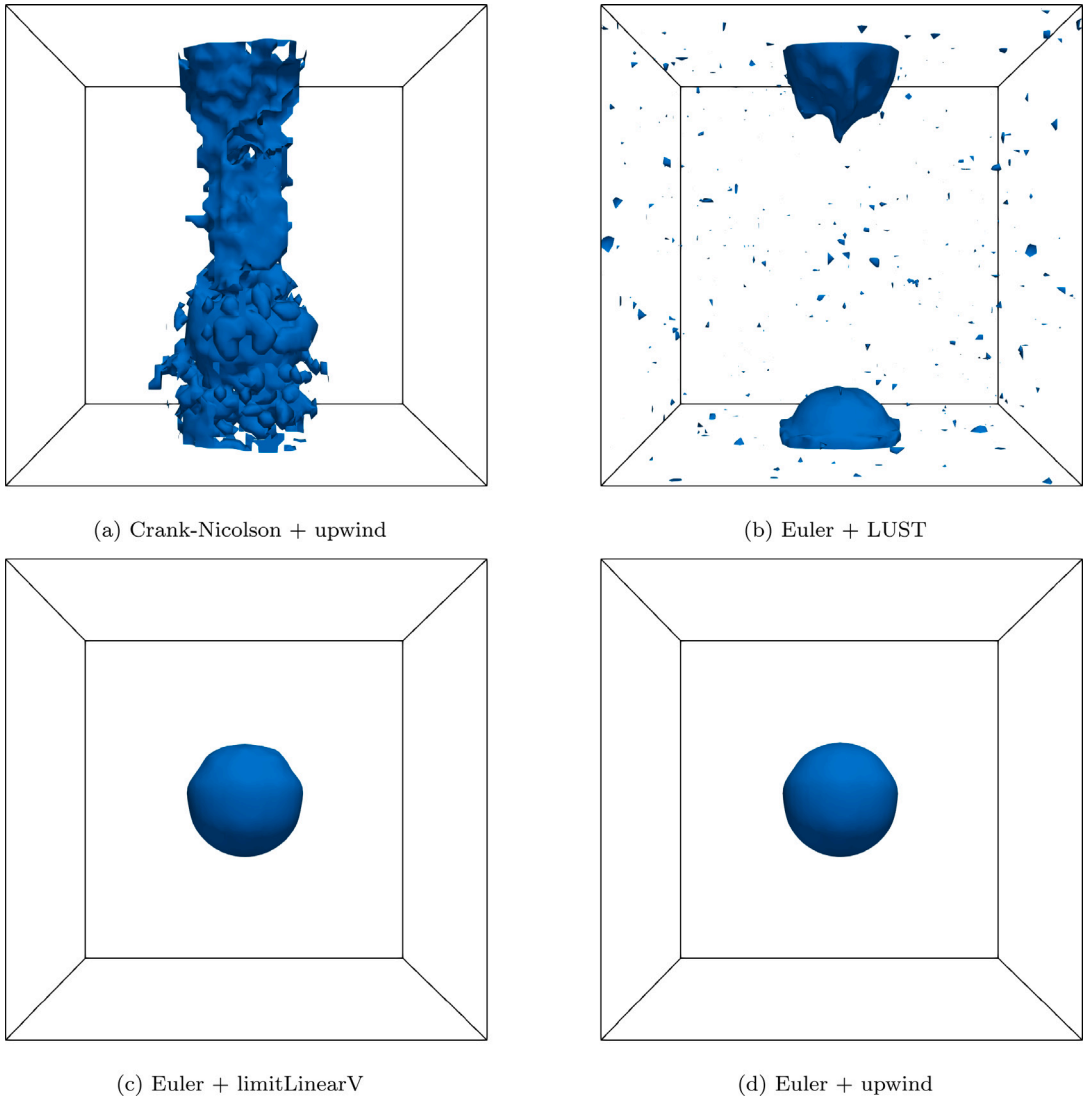


Fig. 6. Temporal evolution of the sphericity error (44) with different schemes: interIsoFoam, $N = 64$.



(a) Crank-Nicolson + upwind

(b) Euler + LUST

(c) Euler + limitLinearV

(d) Euler + upwind

Fig. 7. Final shape of the droplet calculated by interIsoFoam with different schemes: $N = 64$.

Table 2
The terminated time and the final normalized errors of interIsoRhoFoam and interIsoFoam with different scheme combinations and mesh resolutions for translating droplet in ambient flow case.

Scheme combination	Resolution	interIsoRhoFoam				interIsoFoam			
		End time	Mass error	Momentum error	Sphericity Error	End time	Mass error	Momentum error	Sphericity Error
Euler+upwind	32	0.1	0.0	0.0	3.062×10^{-3}	0.1	0.0	0.0	3.062×10^{-3}
	48	0.1	0.0	0.0	1.111×10^{-3}	0.1	0.0	0.0	1.110×10^{-3}
	64	0.1	0.0	0.0	3.484×10^{-4}	0.1	0.0	0.0	3.484×10^{-4}
	96	*	*	*	*	0.1	0.0	0.0	3.197×10^{-4}
	128	*	*	*	*	0.1	0.0	0.0	2.298×10^{-4}
Euler+cubic	32	0.0206	4.840	1.028×10^{18}	−0.1299	0.067	8.807×10^{-3}	1.440×10^{10}	0.3501
	48	0.01	-2.925×10^{-2}	3.581×10^{22}	−0.2341	0.0260	1.6346	2.390×10^{6}	0.1024
	64	0.0065	-2.190×10^{-3}	2.233×10^{22}	−0.2267	0.0273	0.8301	3.142×10^{8}	0.6643
Euler+limitedLinearV	32	0.1	0.0	5.511×10^{-4}	6.773×10^{-3}	0.1	0.0	-2.656×10^{-9}	5.882×10^{-3}
	48	0.1	0.0	4.793×10^{-4}	2.852×10^{-3}	0.1	0.0	-4.995×10^{-9}	3.671×10^{-3}
	64	0.1	0.0	3.290×10^{-4}	2.343×10^{-3}	0.1	5.695×10^{-11}	6.407×10^{-11}	2.216×10^{-4}
Euler+linear	32	0.0384	-4.520×10^{-4}	3.603×10^{21}	−0.1932	0.1	0.0	-2.540×10^{-10}	2.234×10^{-2}
	48	0.0306	−0.1954	4.271×10^{24}	−0.2330	0.1	0.0	2.210×10^{-7}	1.843×10^{-2}
	64	0.0239	−0.0676	2.604×10^{25}	−0.2481	0.1	1.210×10^{-9}	3.915×10^{-10}	9.681×10^{-4}
Euler+LUST	32	0.0221	598.0	3.036×10^{21}	−0.2055	0.0993	0.1097	2.162×10^{19}	0.4242
	48	0.0166	1.026×10^{55}	3.272×10^{120}	0.2428	0.0258	1.047×10^{13}	1.797×10^{39}	−0.1762
	64	0.002	0.0	9.254×10^{-5}	5.042×10^{-4}	0.0485	2.006×10^{-5}	-2.664×10^{-5}	0.2856
Euler+MUSCL	32	0.0046	0.0	4.727×10^{8}	−0.0932	0.1	0.0	1.143×10^{-5}	7.383×10^{-3}
	48	0.0064	0.0	−0.1949	-3.500×10^{-3}	0.0485	2.227×10^{12}	2.566×10^{36}	−0.1434
	64	0.0092	0.0	1.059×10^{-3}	-1.202×10^{-3}	0.1	3.075×10^{-9}	1.780×10^{-7}	7.070×10^{-4}
Euler+QUICK	32	0.0203	0.0	5.455×10^{6}	−0.2096	0.1	0.0	5.177×10^{-4}	9.6670×10^{-3}
	48	0.0066	-3.198×10^{-9}	6.806×10^{82}	−0.2699	0.0185	2.417×10^{-6}	4.164×10^{51}	0.0118
	64	0.0079	-5.097×10^{-7}	1.862×10^{24}	−0.1086	0.0518	0.1564	5.537×10^{21}	7.812×10^{-3}
Euler+SuperBee	32	0.0237	0.0	5.732×10^{15}	−0.1570	0.1	0.0	-2.632×10^{-5}	5.094×10^{-3}
	48	0.0022	0.0	0.2876	-4.630×10^{-4}	0.0631	1.645×10^{-10}	8.588×10^{42}	0.0120
	64	0.0075	0.0	-1.206×10^{-4}	-8.018×10^{-4}	0.1	1.477×10^{-8}	2.970×10^{-7}	1.628×10^{-3}
Euler+vanLeer	32	0.1	0.0	4.190×10^{-4}	-4.514×10^{-4}	0.1	0.0	4.916×10^{-7}	3.495×10^{-3}
	48	0.1	0.0	3.480×10^{-4}	1.417×10^{-3}	0.1	0.0	3.535×10^{-7}	4.868×10^{-3}
	64	0.0896	0.0	-9.273×10^{-5}	0.0107	0.1	3.310×10^{-9}	5.478×10^{-8}	1.388×10^{-4}
CrankNicolson+upwind	32	0.0096	0.0	4.131×10^{-4}	-9.018×10^{-4}	0.0043	−0.2219	7.627×10^{21}	−0.1632
	48	0.0049	0.0	3.238×10^{-3}	-3.139×10^{-4}	0.0037	32.30	2.203×10^{24}	−0.1965
	64	0.0032	6.001×10^{-3}	6.280×10^{-4}	-2.093×10^{-3}	0.0729	9.124×10^{25}	2.576×10^{60}	0.3248

Table 3
Realistic densities of the mercury droplet/air ambient pair.

Materials/properties (25 °C)	Density (kg/m^3)	Kinematic viscosity (m^2/s)	Surface tension (N/m)	Density ratio
air	1.1839	–	–	– [49]
mercury	13.5336×10^{3}	–	– (in air)	11431.37(in air) [49]

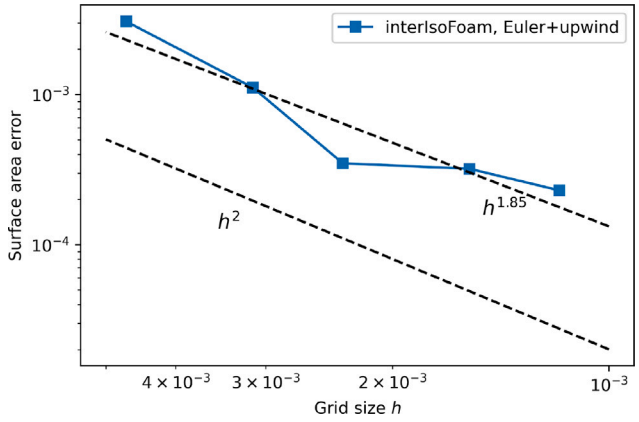


Fig. 8. Mesh convergence study for sphericity error: interIsoFoam, Euler+upwind, $N \in (36, 48, 64, 96, 128)$.

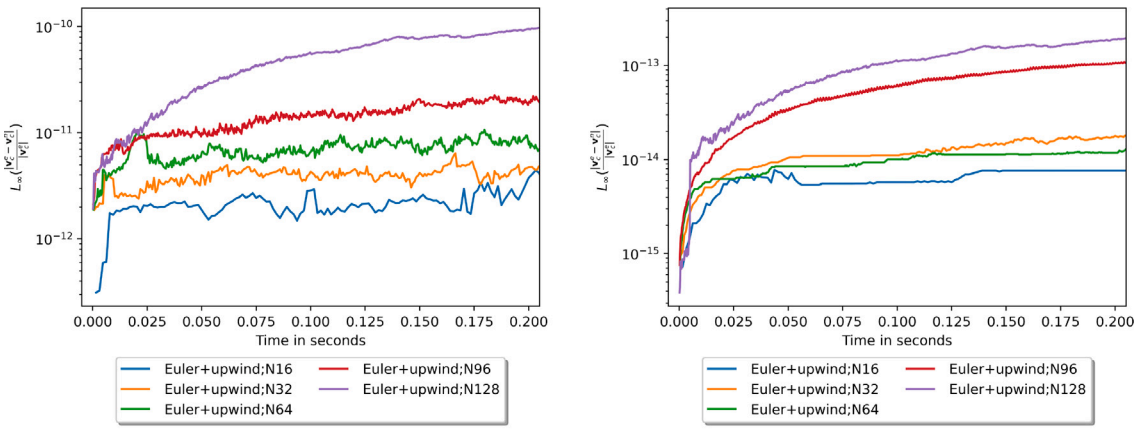
i.e.,

$$L_\infty(\mathbf{v}) = \max_i \left(\frac{\|\mathbf{v}_i - \mathbf{v}_\infty\|}{\|\mathbf{v}_\infty\|} \right), \tag{45}$$

where $\mathbf{v}_i$ denotes the velocity of the cell i, and the analytical velocity value is $\mathbf{v}_\infty = \mathbf{v}_a = (0, 0, 0.01)$. The expected exact value of L_∞ is 0; however, in practice, the absolute accuracy is limited by the absolute tolerance of the linear solver used to solve the pressure Poisson equation.

Fig. 9 presents the temporal evolution of $L_\infty(\mathbf{v})$. The same $L_\infty(\mathbf{v})$ calculated for both solvers reveals a very close numerical equivalence between the volume fraction and mass conservation equation using the Euler+upwind combination of schemes. Errors of both solvers remain stable. Absolute errors of interIsoFoam are somewhat larger; however, they remain in the realm of numerical noise, significantly below the linear tolerance for the pressure Poisson equation, ensuring consistency. A notable outcome from Fig. 9 is the value of the final converged $L_\infty(\mathbf{v})$, which is at the magnitude of 1×10^{-11} and for interIsoRhoFoam almost reaches the machine epsilon, confirming numerical stability and consistency of a very high degree for this challenging verification case.

We also conducted tests on all schemes listed in Table 1 for this particular case setup. The corresponding results are presented in Fig. 10. The results from Crank–Nicolson temporal scheme and SuperBee convection scheme show significant numerical instability at the early stage of the simulation. In contrast, the errors obtained from all other schemes remain stable throughout the entire simulation. Notably, there is a substantial variation in accuracy among these schemes. The velocity errors from Euler + QUICK and upwind remain at magnitudes around 10^{-11}, while the errors from the other schemes initially increase and stabilize at values that are $10^{1}0$ to 10^{11} times larger. Additionally, the same cases are tested with density ratio 1 to investigate the effects of the density ratio on the numerical instability. As shown in Fig. B.25, all cases keep stable with the low density ratio at the final stage. The



(a) interIsoFoam (b) interIsoRhoFoam

Fig. 9. Temporal evolution of the velocity error norm $L_\infty(\mathbf{v})$ with pure advection: Euler, Gauss upwind, density ratio: 10^4, mesh resolution: $N \in (16,\ 32,\ 64,\ 96,\ 128)$.

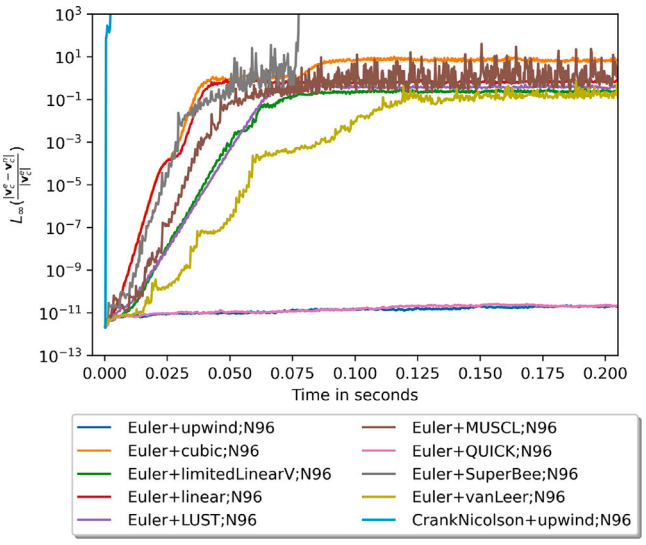


Fig. 10. Temporal evolution of the velocity error norm $L_\infty(\mathbf{v})$ with pure advection — combining 10 schemes, density ratio is 10^4, mesh resolution is $N = 96$. Only Euler and upwind(ing) schemes remain consistent and stable.

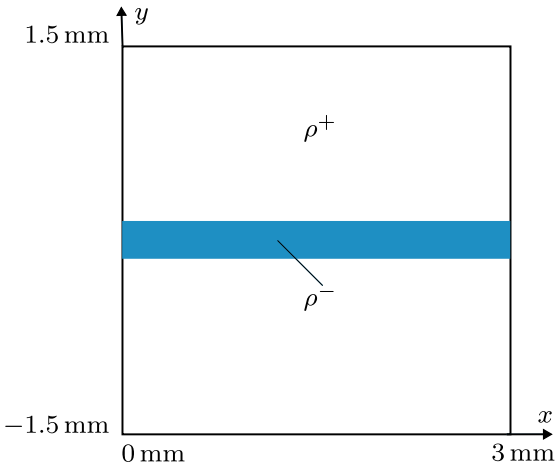


Fig. 11. 2D mixing layer.

velocity errors of all tests are reduced to magnitudes of 10^{-14}, even for the most critical schemes, i.e. Crank–Nicolson + upwind.

4.4. Mixing layer

In this case, a 2D mixing layer case is tested. The 2D computational domain as depicted in Fig. 11 has the same length $L = 3\,\text{mm}$ in both $x-$ and $y-$direction. The liquid with high density $\rho^- = 1000\,\text{kg/m}^3$ flows in the middle region $-0.15\,\text{mm} < y < 0.15\,\text{mm}$ of the square computational domain with relatively low initial velocity $v_x^- = 2\,\text{m/s}$, while the gas with the density $\rho^+ = 1\,\text{kg/m}^3$ flows on both sides of the liquid area with very high initial velocity $v_x^+ = 30\,\text{m/s}$. A spatial velocity perturbation is initialized in the internal field and has the distribution

$$v_y(y) = 0.01 v_x^- \sin 2\pi \frac{x}{L} \exp -(\frac{2y}{h})^2,$$

where h indicates the thickness of the liquid region, i.e. 0.3 mm. The simulations are tested with a resolution of $N_x \times N_y \times N_z = 256 \times 256 \times 1$. The duration of the simulation is set to 0.003 s, allowing for sufficient number of time steps for the inconsistencies to develop. To highlight the dominant impacts of the convection, the surface tension force, gravity, and viscosity are excluded from the simulations. The periodic condition is employed for all boundaries. We compared our results with the results computed by the ONERA DYJEAT codes [25,50–53], which upholds consistent mass-momentum transport through solving the temporary density equations together with momentum equations on staggered meshes.

The Fig. 12 provides a quantitative comparison among multiple schemes in Table 1 and with the DYJEAT code, focusing on the temporal evolution of the normalized momentum error evaluated using Eq. (43). From the plot, it is evident that only two cases using interIsoFoam remain stable, namely, as expected using Euler + upwind, but also Euler + limitedLinearV. The zoomed-in view in Fig. 12 highlights the accuracy of the results for these three stable cases. The errors calculated from DYJEAT are larger than errors from stable cases using interIsoFoam, which are around 4%. As shown in the detail in Fig. 12, the errors from consistent Euler + upwind are minimal, with respect to another combination of schemes.

4.5. Validation of a single rising bubble

In the present study, we investigate the performance of the proposed method by applying it to the simulation of a single bubble rising in a quiescent viscous liquid. To validate our approach, we adopt the configuration presented by Anjos et al. [54], who simplified the rising bubble experiments originally conducted by Bhaga and Weber

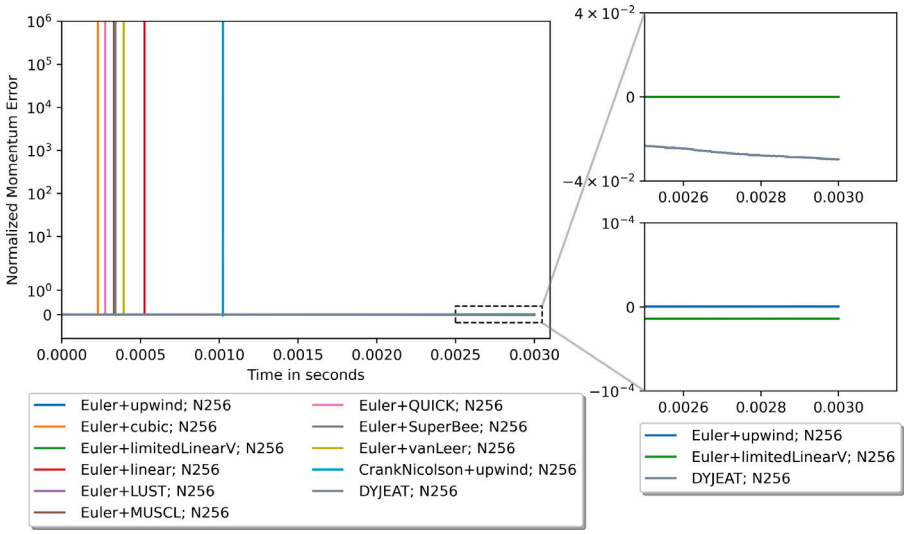


Fig. 12. Time evolution of normalized momentum error of mixing layer with different schemes and DYJEAT codes, density ratio: 10^3, resolution: $N = 256$.

[55]. In their work, Anjos et al. [54] select three distinct viscosities for comparative analysis. Our focus, in particular, lies on the cases, characterized by larger liquid viscosities. The specific cases correspond to a Morton number $Mo = g\nu_l^4/\rho_l\sigma^3 = (848, 41.1, 1.31)$, where g represents the gravitational acceleration, and ν_l, ρ_l, and σ denote the viscosity, density of the ambient liquid, and surface tension coefficient, respectively. Three resolutions are tested: $N \in (64, 96, 128)$.

For our simulations, we initialize the air bubble with an idealized spherical shape, possessing a diameter of $D = 2.61\,\text{cm}$. The air properties are defined by a viscosity of $1.78 \times 10^{-5}\,\text{kg/(m s)}$ and a density of $1.225\,\text{kg/m}^3$, while the liquid properties encompass viscosities of $(2.73, 1.28, 0.54)\,\text{kg/(m s)}$ and a density of $1350\,\text{kg/m}^3$. Furthermore, the surface tension between the air bubble and the liquid is $0.078\,\text{N m}^{-1}$. The computational domain is defined as $(-4D, -4D, -2D)\times(4D, 4D, 6D)$, where the positions of the space diagonal vertices of the computational domain are delineated, with the initial position of the bubble set at the origin, $(0, 0, 0)$. A set of dimensionless normalized variables is introduced as follows:

$$\mathbf{w} = \frac{\mathbf{v}}{\sqrt{gD}}, \quad t = \sqrt{\frac{g}{D}}\tau, \tag{46}$$

where τ indicates the physical time in seconds.

As illustrated in Fig. 13, the utilization of the Euler and upwind schemes ensures the preservation of equivalence between volume fraction and mass advection equation. Consequently, the results obtained from interIsoFoam and interIsoRhoFoam, when employing the identical Morton number, display a substantial level of similarity. When considering Mo = 41.1, as demonstrated in Figs. 13(c) and 13(d), the velocities acquired from both solvers demonstrate an initial increase until approximately $t = 1$, followed by a subsequent decline leading to a stable state. Notably, the differences among the results become evident when dealing with the cases involving Mo = 41.1. Specifically, in instances where a coarse mesh ($N = 64$) is employed, the velocity decline is more pronounced, resulting in a smaller final stable velocity compared to the values presented in prior works [54–56]. The larger final velocity disparity from the previous works, can be observed for Mo = 1.31 with a mesh resolution $N = 64$ in Figs. 13(e) and 13(f), which is resulted from that the rising bubble with lower viscosities has a stronger deformation and has a thinner structure required to be captured. Using a coarse mesh loses this information and subsequently causes a larger error. Conversely, higher resolutions yield stable velocities that agree well with the experimental data documented by Bhaga and Weber [55],Hua and Lou [56]. On the other hand, for cases with Mo = 848, characterized by a larger viscosity, the impact of resolution is less conspicuous, as depicted in Figs. 13(a) and 13(b). The final velocities attained from different resolutions converge similarly to values that fall between the results obtained in the simulation conducted by Hua and Lou [56] and the experimental study conducted by Bhaga and Weber [55].

Furthermore, the profiles obtained by slicing the surfaces of the droplets, passing through their centers, are compared with the experimental profiles from Bhaga and Weber [55], as illustrated in Fig. 14. For both solvers, the droplets with varying viscosities exhibit final shapes that closely align with the experimental visualizations, validating the hypothesis of the equivalence between Euler+upwind discretization of Eq. (5) using the scaled mass flux $|V_f^{\alpha}|_s$ from Eq. (32), and the solution of the auxiliary density equation using Algorithm 1.

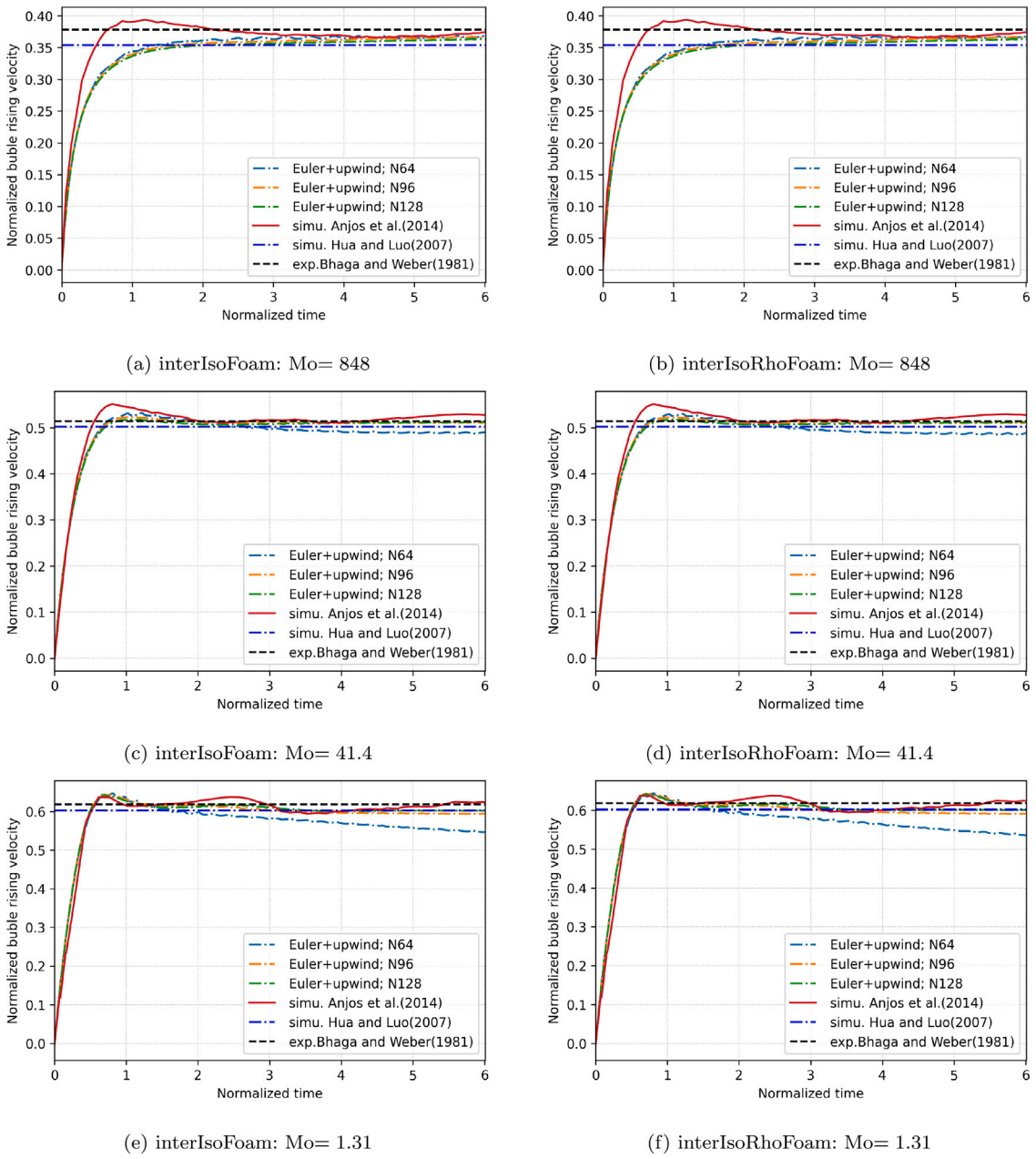
4.6. Liquid jet in high speed gaseous cross-flow

Different from the parallel velocities of the mixing layers in Section 4.4, the liquid flows with a lower velocity is perpendicular to the velocity of the gaseous phase in this case, which is called the injection of a liquid jet in a gaseous cross-flow (LJCF) and is common in many engineering applications. The geometry and the physical properties are configured by referring to Zuzio et al. [25]. The rectangular computational domain $\Omega : [-0.01, 0, -0.01] \times [0.03, 0.02, 0.01]\,\text{m}$ has two inlets. The gas flows in with a velocity $\mathbf{v}^+ = [65, 0, 0]\,\text{m/s}$ from the left boundary x_{min}. Thuillet [57] revealed the impact of the liquid inlet velocity profile on the jet trajectory. He simulated the jet with an uniform liquid inlet velocity profile, and with a velocity profile calculated through simulating the injector. The jet trajectory results from the case with calculated velocity profile showed better agreement with the experiment. We followed the calculated liquid injected velocity profile from [57], which is

$$\mathbf{v}_y^- = -21.434\left(\frac{r}{d}\right)^3 + 15.512\left(\frac{r}{d}\right)^2 + 8.6504,$$

where $\mathbf{v}_y^-$ is the $y-$component of the liquid inlet velocity $\mathbf{v}^-$, r indicates the distance to the nozzle center, and d is the diameter of the nozzle. The x, $z-$component of $\mathbf{v}^-$ are set to zero, whereas $d = 0.002\,\text{m}$. The nozzle's center locates at $[0, 0, 0]$ in the bottom boundary patch y_{min}. To save the computation resource, the uniform Cartesian mesh with a moderate resolution of $[N_x, N_y, N_z] = [128, 64, 64]$ has been adopted. Fig. 15 depicts the flow domain. The physical properties are $\rho^- = 1000\,\text{kg/m}^3$, $\rho^+ = 1.225\,\text{kg/m}^3$, $\mu^- = 1.0 \times 10^{-3}\,\text{kg/(m s)}$, $\mu^+ = 1.78 \times 10^{-5}\,\text{kg/(m s)}$, $\sigma = 7.2 \times 10^{-2}\,\text{N m}^{-1}$, $g = 9.81\,\text{m/s}^2$.

Fig. 17 shows the final state of the injected liquid at $t = 7.1\,\text{ms}$ using interIsoFoam and interIsoRhoFoam with Euler and Gauss upwind



(a) interIsoFoam: Mo= 848

(b) interIsoRhoFoam: Mo= 848

(c) interIsoFoam: Mo= 41.4

(d) interIsoRhoFoam: Mo= 41.4

(e) interIsoFoam: Mo= 1.31

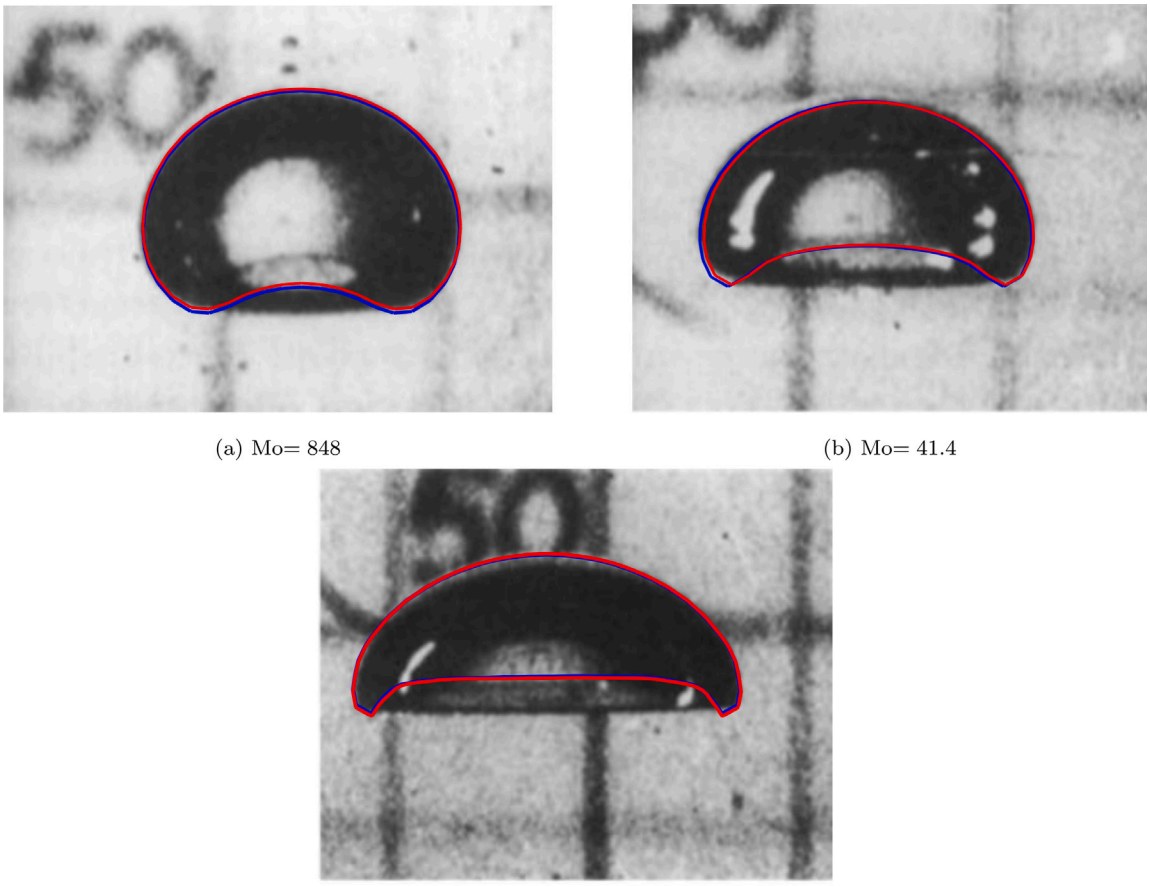
(f) interIsoRhoFoam: Mo= 1.31

Fig. 13. Temporal evolution of rising velocity using interIsoFoam and interIsoFoam: Euler + upwind, $\rho^{-}/\rho^{+} \approx 10^{3}$.

regarding the iso-value of the reconstructed distance function $RDF = 0$, as well as using DYJEAT [25] with a high resolution of $[N_x, N_y, N_z] = [1024, 512, 512]$, whose shape is rendered by the iso-value 0.5 of the volume fraction. Figs. 17(a) and 17(b) illustrate the ruptured liquid jet from the y-side view. The droplets' distributions of interIsoFoam and interIsoRhoFoam display many comparable characteristics. We can observe two strips of droplets and a strip of bag-like liquid structure. The outer liquid segregates into two yz-plane symmetric strips of droplets at an early stage, i.e., at a low penetration height, as shown in the right subfigure of Figs. 17(a) and 17(b). These droplets translate with the gas along the x-direction and spread spanwise in the y-direction. The remaining center liquid has a wavelike detachment and forms the bag-like structure in the middle of the strips. Fig. 17(c) demonstrates the results from DYJEAT codes with a higher resolution. Similar liquid distributions can be observed: the liquid in the center zone of the nozzle propagates like a wave and breaks up into some large packets at a higher position, whereas the liquid in the periphery of the nozzle zone rips at a low penetration height. We also tested interIsoFoam with unstable schemes, and they fail. An example combination of Euler + cubic, is shown in Fig. B.27, with the simulations on both coarser and finer mesh fail catastrophically.

This validation case is a candidate for a benchmark case for validating two-phase flow numerical methods that consistently handle high density ratios, because the experimental form of the jet can be accurately reproduced on coarser mesh resolutions. Using a coarser mesh resolution interIsoFoam and interIsoRhoFoam of course do not capture the small structures such as liquid streaks, sacs and droplets, as shown in DYJEAT's results. However, the solvers accurately predict the jet curve, which can be used as a quantifiable argument for validity of a consistent method against experimental data.

The Fig. 16 displays the final bent shape of liquid jet simulated by three solvers and their comparison with the experimental observation made by ONERA [58,59]. The same case with two different resolutions are tested. We put the liquid jets' shape results in the same parallel



(a) Mo= 848 (b) Mo= 41.4

(c) Mo= 1.31

Fig. 14. Comparisons of final shapes of rising bubbles using interIsoFoam and interIsoRhoFoam with the experimental visualization from Bhaga and Weber [55] (reprinted with permission): Euler + upwind, red line from interIsoRhoFoam, blue line from interIsoFoam, $\rho^-/\rho^+ \approx 10^3$, $N = 128$, $t = 6$.

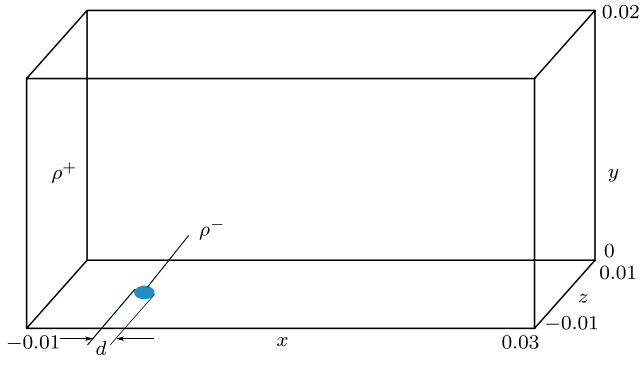


Fig. 15. Liquid in a cross flow.

view to compare them with each other and also with the experimental results marked by the red line. The blue translucent liquid jet represents the results from the case with a higher resolution $N_h = [254, 128, 128]$, while the gray liquid jet comes from the above low-resolution results. It can be seen from each Figs. 16(a)–16(c) that more small droplets and complex structures can be captured when deploying the higher mesh resolution. Despite the different resolutions, there exists a very minor difference between the liquid jets with regard to the bent shape. The windward surfaces of the two liquid jets in each sub-figure almost attach to each other, which highlights that this case is insensitive to the mesh resolution. As to the comparison with the experimental trajectories, both the jets' bent surface in Figs. 16(a) and 16(b) show good correspondence to the experimental shape, i.e. the red line at the low penetration height < 9 mm. The jets reattach to the red line in the upper-right zone. The maximal deviation between the simulated jets and the experiment shape is around 1 mm, which is 5% of the jet height. A more obvious discrepancy between the jets and the experiment is shown in Fig. 16(c). The liquid jet bent less than the experiment in the high-speed flow after the given time, which results in a wrong prediction of the impingement position.

5. Conclusions

We analyze the equivalence between the mass conservation and volume fraction advection equation in the context of flux-based VOF methods. When an average mass flux is computed on the discrete level by scaling the fluxed phase-specific volume with the time step, we find that the equivalence is possible only when the first-order Euler temporal discretization is used for momentum conservation, no flux limiting is applied, and the flux-based VOF scheme uses first-order quadrature for the integration of the fluxed phase-specific volume. For any flux-based VOF method, the integration of the fluxed phase-specific volume V_f^α therefore plays a key role in the stability of two-phase flow simulations with high density ratios, when the mass flux is estimated from it. Altering the mass flux $\rho_f F_f$ in the discrete momentum conservation equation by applying flux limiting schemes or blending schemes causes errors that are proportional to the density difference, and lead either to large errors in the interface shape and topology, or catastrophic failure. In other words, upwinding the two-phase momentum in a flux-based

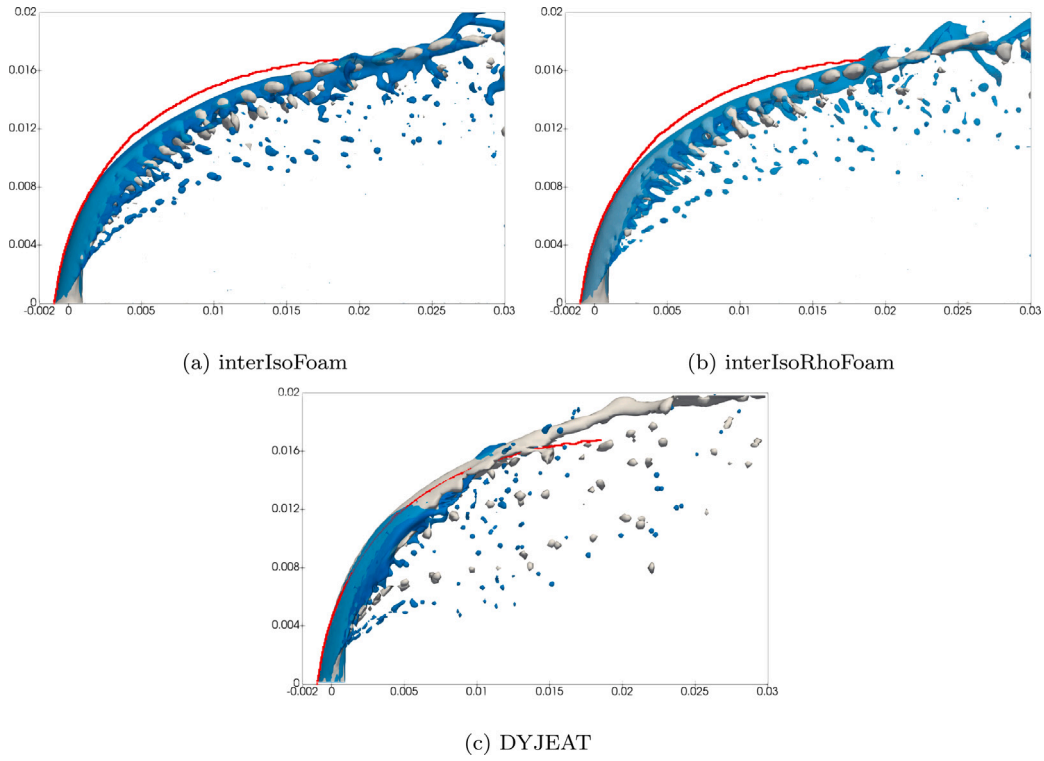


Fig. 16. The instantaneous liquid jet shape at final time $t = 7.1$ ms with different resolutions (the blue translucent jet: $N_h = [256, 128, 128]$; the gray jet: $N_l = [128, 64, 64]$) and its comparison with the experimental results(the red line). This case makes it possible to evaluate performance on coarser meshes as resolving finer structures does not impact the jet trajectory.

VOF method is consistent to the geometrical upwinding of the volume fraction by the flux-based VOF method.

We successfully apply the ρLENT method for high-density ratio flows to the isoAdvector-plicRDF Volume-of-Fluid method [18]. The adaption of the ρLENT method in the context of the geometrical Volume-of-Fluid method is straightforward, requiring only the geometrical calculation of the upwind area fraction α_f^o from the available geometric VOF interface (phase-indicator) approximation.

We demonstrate the equivalence on the discrete level between the mass flux scaled from the phase-specific volume and the solution of an auxiliary density equation. We confirm this hypothesis by means of verification with challenging inviscid cases, as well as by validation against experiments.

CRediT authorship contribution statement

Jun Liu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tobias Tolle:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Davide Zuzio:** Writing – review & editing, Visualization, Validation, Formal analysis. **Jean-Luc Estivalèzes:** Writing – review & editing, Visualization, Validation, Formal analysis. **Santiago Marquez Damian:** Writing – review & editing, Formal analysis. **Tomislav Marić:** Writing – review & editing, Writing – original draft, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

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 research data, the source code archive and the source code repository are publicly available and referenced in the manuscript.

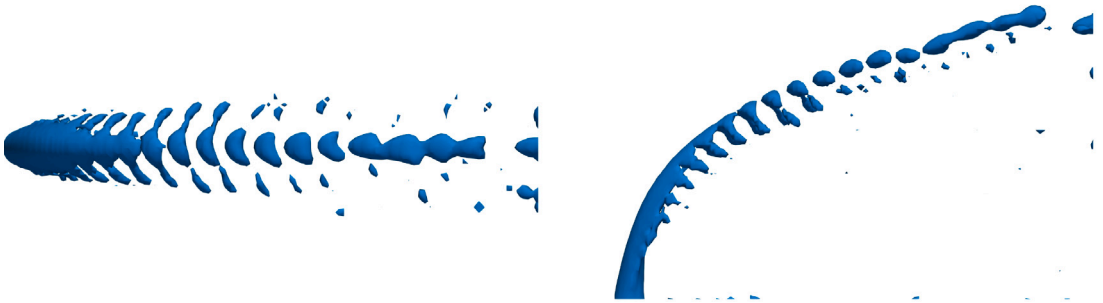
Acknowledgments

The last author acknowledges funding by the German Research Foundation (DFG) – Project-ID 265191195 – SFB 1194. Calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt. The computations with DYJEAT codes were granted access to the HPC resources of CALMIP supercomputing center under the allocation 2023 P18043.

Appendix A. Correct cyclic boundary condition for the plicrdf-isoadvector method

A cyclic boundary condition (BC) treats two boundary patches as if they were physically connected, with their respective cell layers placed next to each other. For the geometrical VOF method such as the plicRDF-isoAdvector, the cyclic boundary condition impacts the interface reconstruction and the volume fraction advection. To achieve this, the cyclic BC performs calculations on the so-called *owner-patch*, and then reflects the result to the so-called *neighbor-patch*, as

(a) interIsoFoam: $t = 7.1\,\text{ms}$



(b) interIsoRhoFoam: $t = 7.1\,\text{ms}$

(c) DYJEAT: $t = 7.07\,\text{ms}$

Fig. 17. The shape of the injected liquid with interIsoFoam and interIsoRhoFoam (Euler and Gauss upwind, density ratio: 816, CFL number: $CFL = 0.2$, resolution: $N_l = [128, 64, 64]$), and with DYJEAT (resolution: $N = [1024, 512, 512]$ [25]).

shown schematically in Fig. A.18 for a geometric VOF method in two dimensions.

For the advection discretization at cyclic BC, we noticed that the fluxed phase-specific volumes V_f^α field is not properly adjusted for the cyclic patches.

In the first step, the cyclic BC initializes V_f^α at every face of both cyclic patches using the upwind scheme, i.e.,

$$V_f^{\alpha,init} = F_f \alpha_U \Delta t, \tag{A.1}$$

where $F_f := \mathbf{v}_f \cdot \mathbf{S}_f$ is the volumetric flux at the centroid of the cell-face S_f, α_U is the volume fraction of the upwind cell U w.r.t the cell face S_f, and Δt is the time step.

In the second step, the cyclic BC computes the geometric V_f^α at the faces that belong to interface cells on both patches

$$V_f^\alpha = \int_{t^n}^{t^{n+1}} \int_{S_f} \chi \mathbf{v} \cdot \mathbf{n}\, dS\, dt = \int_{t^n}^{t^{n+1}} \frac{F_f(t)}{|\mathbf{S}_f|} A_f(t)\, dt \tag{A.2}$$

The existing cyclic BC [60] does not consider cyclic boundary conditions. The discretization in cell layers adjacent to two cyclic boundary patches should handle the cell layers as if they are placed next to each other as shown in Fig. A.18. A phase-specific volume fluxed out of the domain on a face that belongs to the cyclic owner-patch, should be fluxed into the corresponding face in the cyclic-neighbor patch, as described in Algorithm 3.

Algorithm 3 The modified fluxed phase-specific volumes V_f^α update method in interIsoFoam.

Initialize $V_f^{\alpha,owner}$ using the upwind scheme. ▷ Eq. (A.1)
for all boundary patches **do**
  if boundary patch is cyclic **then**
    for all cyclic-patch faces $f \in [1, |P_{cyclic}|]$ **do**
      if $F_f > 0$ **then**
        Geometrically compute the outflow V_f^α.
      else if $F_f < 0$ **then** ▷ The inflow $F_f < 0$ here is outflow $F_f > 0$ of the neighbor.
        $V_f^\alpha = -V_f^{\alpha,neighbor}$
      end if
    end for
  end if
end for

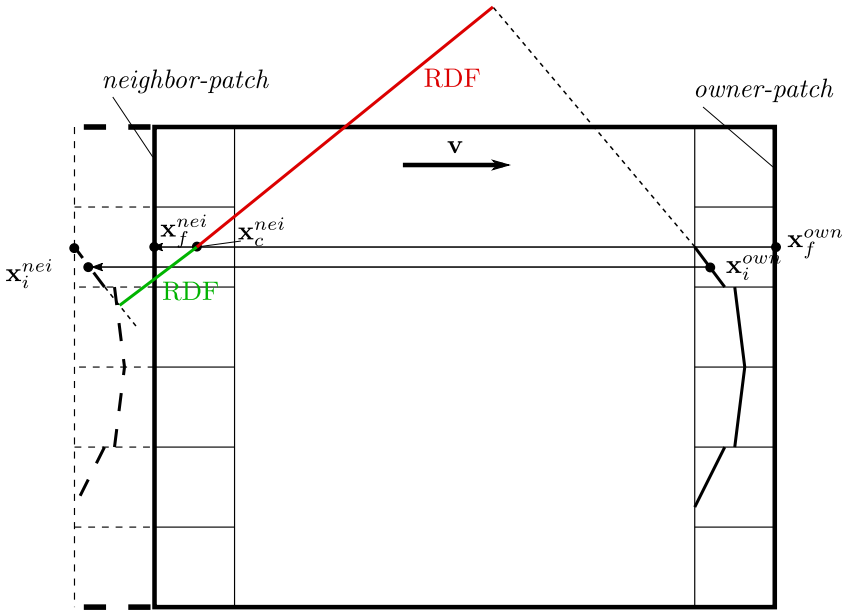


Fig. A.18. The ghost cells layer to correct signed distance: the green line (——) presents fixed RDF from cell centroid $\mathbf{x}_c^{nei}$ to interface in the cyclic neighbor cell, while the red line (——) depicts the original unfixed RDF.

Aside from the consistent calculation of V_f^α in the advection, the cyclic BC also affects the geometric interface reconstruction. The plicRDF reconstruction [18] uses the reconstructed distance function (RDF) and its gradient for improving discrete interface-normal vectors. When considering the cyclic boundary condition, the distance calculation should be treated carefully, especially in the case where two cells sharing a vertex are also attached to two cyclic patches. This issue is illustrated in Fig. A.18, where $\mathbf{x}_i^{own}$ denotes a center of a VOF interface polygon located in a corresponding interface cell Ω_i^{own} that belongs to the cell layer attached to the owner patch of the cyclic BC. In the plicRDF implementation in OpenFOAM-v2306 [60], the cyclic BC falsely uses VOF interface polygon centers $\mathbf{x}_i^{own}$ to compute signed distances at centers $\mathbf{x}_c^{nei}$ in the cell layer adjacent to the cyclic neighbor-patch. A correct implementation of the cyclic BC requires positions $\mathbf{x}_i^{nei}$, as shown in Fig. A.18. The VOF interface centroids from the cell layer adjacent to the cyclic owner-patch $\mathbf{x}_i^{own}$ and the face centers of the cyclic owner-patch can be used to compute $\mathbf{x}_i^{nei}$ for every $\mathbf{x}_i^{own}$ and facilitate a correct cyclic (periodic) computation of the Reconstructed Distance Function (RDF) in the plicRDF-isoAdvector method.

We define a transformation of the position of the interface centers using

$$\mathbf{x}_i^{nei} := \mathbf{x}_i^{own} + (\mathbf{x}_f^{nei} - \mathbf{x}_f^{own}), \tag{A.3}$$

where $(\mathbf{x}_f^{nei} - \mathbf{x}_f^{own})$ is the difference (displacement, or transformation) vector between the face centers of cyclic BC owner and neighbor patch face centers.

Note that there is no need to transform the interface normals $\mathbf{n}_i$, because they are orientation and not position vectors. The transformed PLIC polygon centroids together with the cyclic BC neighbor-patch interface normals $\mathbf{n}_i^{nei} = \mathbf{n}_i^{own}$ build a cyclic ghost-data layer that is shown schematically by dashed cells in Fig. A.18.

Signed distances in the cell layer adjacent to the cyclic owner-patch at cell centers $\mathbf{x}_i^{own}$, are thus computed from the transformed PLIC interface information from the cyclic patch-owner data.

If the cyclic-patch-adjacent cell Ω_c does contain its own PLIC interface with the PLIC centroid, then the signed distance to this PLIC interface is used, if it is closer to $\mathbf{x}_i$ of that cell.

We verify our discretization of the cyclic BC in the plicRDF-isoAdvector method using a constant flow case shown in Fig. A.19, where both the droplet and the ambient flow have the same

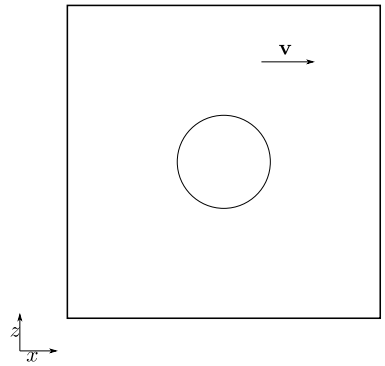


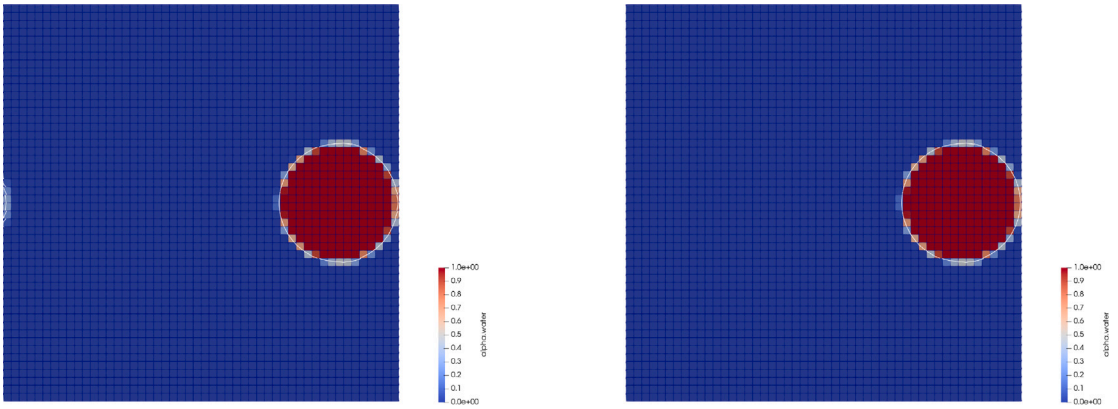
Fig. A.19. A droplet moves with the ambient constant flow.

initial velocity $\mathbf{v} = (v_x, 0)$. The left and right boundaries are set as cyclic (periodic). It is noteworthy that the momentum equation Eq. (5) is not solved in this test case, ensuring exactly constant velocity and pressure over time, aiming to only to test the periodic (cyclic) interface reconstruction and advection. We test two interface reconstruction methods, e.g. isoAlpha and plicRDF from [18].

Fig. A.20 shows the droplet reaching the right cyclic boundary. With the erroneous calculation of the fluxed phase-specific volume V_f^α at the cyclic boundaries, as shown in Fig. A.20(a), the interfaces appear in the patch neighbor cells incorrectly. The Fig. A.20(b) show the accurate result of our modification with a single interface cell. The modification of V_f^α impacts both isoAlpha and plicRDF methods, while adapting the displacement is crucial only for the plicRDF reconstruction. As shown in Fig. A.21(a), without modifying the displacement vector, some liquid remains in the cell layer adjacent to the neighbor-patch after the droplet crosses the cyclic boundary, and reaches a location far from both cyclic boundaries. With applying our modification from Fig. A.21(b), the droplet retains its initial form. The fixed cyclic boundary condition is available in [32].

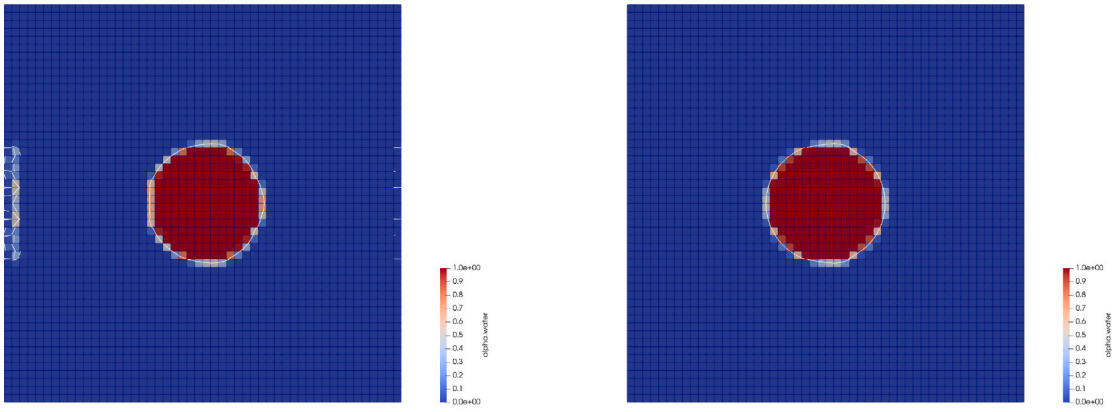
Appendix B. Supplementary results

See Figs. B.22–B.27.



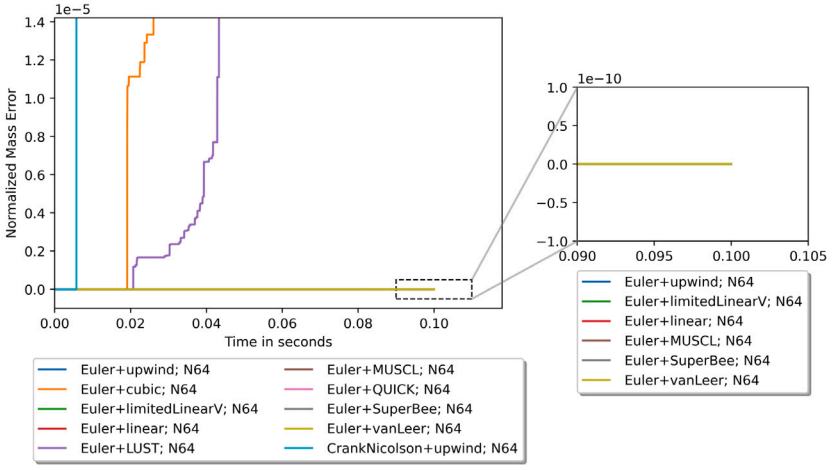
(a) Unfixed (b) Fixed

Fig. A.20. The alpha field and interfaces reconstructed by iso-Alpha method reach to the right cyclic boundary; blue region: the ambient flow; red region: the liquid droplet; white line segments: the PLIC interfaces.

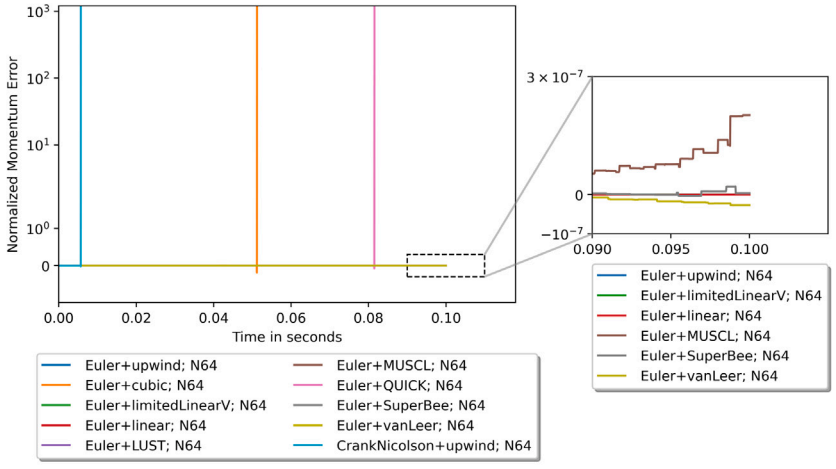


(a) Unfixed (b) Fixed

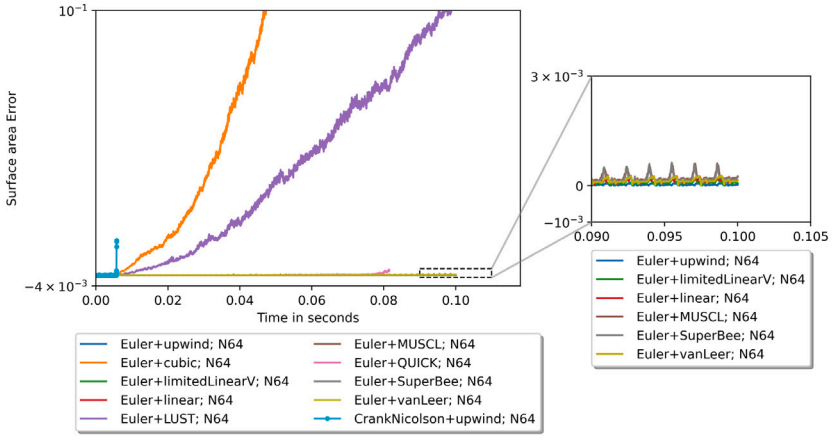
Fig. A.21. The alpha field and interfaces reconstructed by plic-RDF method cross the right cyclic boundary; blue region: the ambient flow; red region: the liquid droplet; white line segments: the PLIC interfaces.



(a) Mass error.

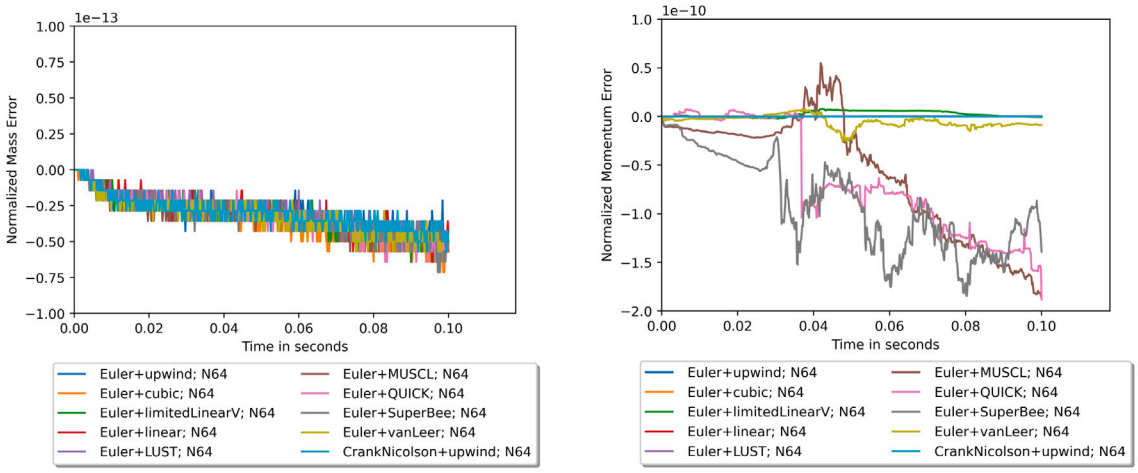


(b) Momentum error.



(c) Sphericity error.

Fig. B.22. Temporal evolution of normalized mass, momentum conservation error and sphericity error with different schemes for the 2D case of Translating droplet in ambient flow: interIsoFoam, $N = 64$, density ratio $= 10^6$.



(a) Mass error. (b) Momentum error.

Fig. B.23. Temporal evolution of normalized mass and momentum conservation error with different schemes for the case of Translating droplet in ambient flow: interIsoFoam, $N = 64$, density ratio = 1.

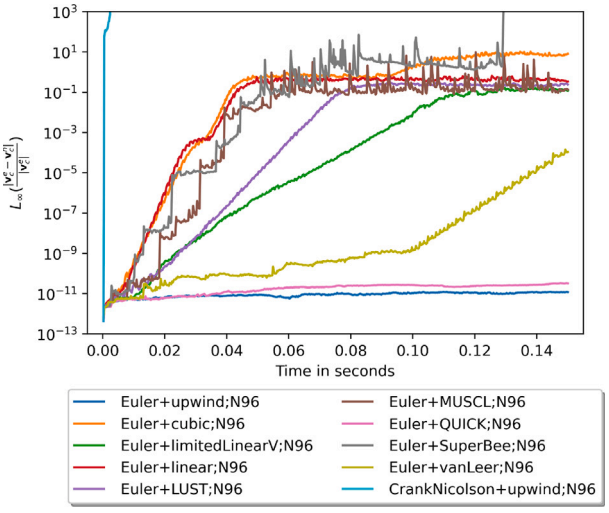


Fig. B.24. Temporal evolution of the velocity error norm $L_\infty(\mathbf{v})$ with 2D pure advection — combining 10 schemes, density ratio is 10^4, mesh resolution is $N = 96$.

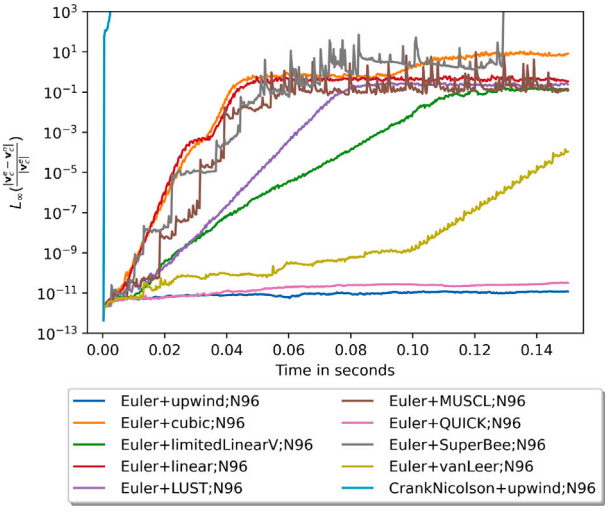
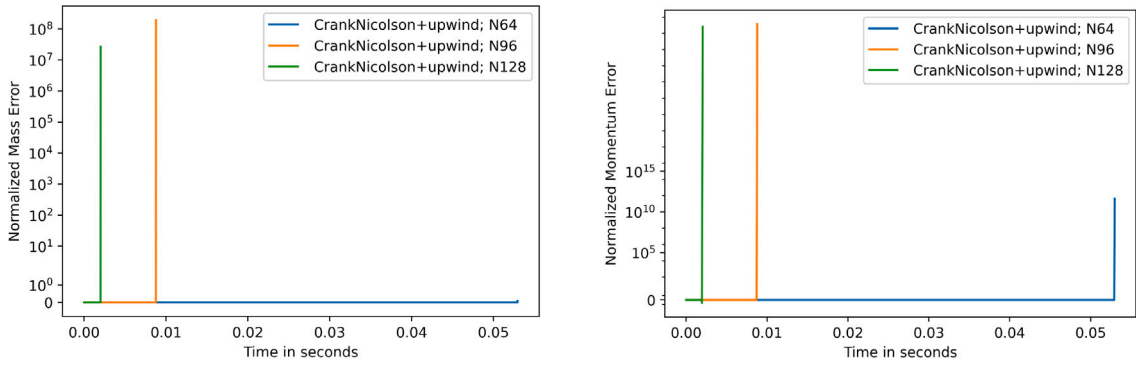
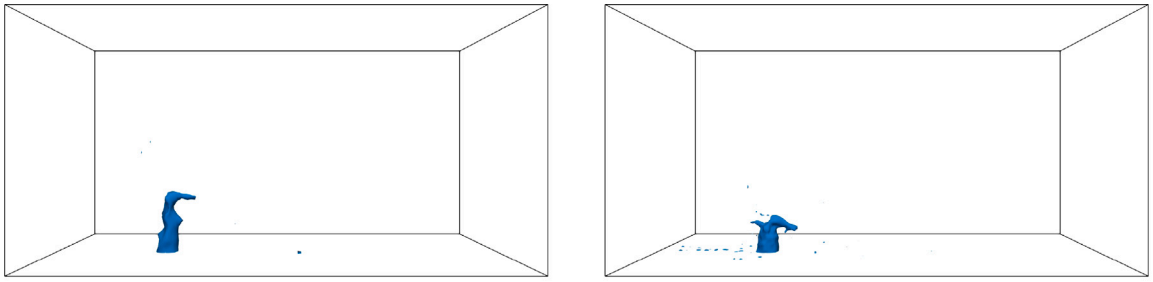


Fig. B.25. Temporal evolution of the velocity error norm $L_\infty(\mathbf{v})$ with pure advection — combining 10 schemes, density ratio is 1, mesh resolution is $N = 96$. All schemes are stable.



(a) Mass error. (b) Momentum error.

Fig. B.26. Temporal evolution of normalized mass and momentum conservation error using CrankNicolson + upwind with different resolutions for the case of Translating droplet in ambient flow: interIsoFoam, $N = 64, 96, 128$, density ratio = 10^6.



(a) Resolution: $N_l = [128, 64, 64]$; final time: $t = 0.7$ ms. (b) Resolution: $N_h = [256, 128, 128]$; final time: $t = 0.4$ ms.

Fig. B.27. The shape of the exploded injected liquid with interIsoFoam (Euler and cubic, density ratio: 816, CFL number: $CFL = 0.2$.

References

[1] Liu J, Tolle T, Bothe D, Marić T. An unstructured finite-volume level set/front tracking method for two-phase flows with large density-ratios. J Comput Phys 2023;493:112426. http://dx.doi.org/10.1016/j.jcp.2023.112426.

[2] Huang Z, Lin G, Ardekani AM. A mixed upwind/central WENO scheme for incompressible two-phase flows. J Comput Phys 2019;387:455–80. http://dx.doi.org/10.1016/j.jcp.2019.02.043.

[3] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126(1):202–28. http://dx.doi.org/10.1006/jcph.1996.0130.

[4] Dodd MS, Ferrante A. A fast pressure-correction method for incompressible two-fluid flows. J Comput Phys 2014;273:416–34. http://dx.doi.org/10.1016/j.jcp.2014.05.024.

[5] Xie B, Jin P, Du Y, Liao S. A consistent and balanced-force model for incompressible multiphase flows on polyhedral unstructured grids. Int J Multiph Flow 2020;122:103125. http://dx.doi.org/10.1016/j.ijmultiphaseflow.2019.103125, URL https://www.sciencedirect.com/science/article/pii/S0301932219304501.

[6] Xie B, Xiao F. Toward efficient and accurate interface capturing on arbitrary hybrid unstructured grids: The THINC method with quadratic surface representation and Gaussian quadrature. J Comput Phys 2017;349:415–40. http://dx.doi.org/10.1016/j.jcp.2017.08.028, URL https://www.sciencedirect.com/science/article/pii/S0021999117305995.

[7] Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J Comput Phys 2006;213(1):141–73. http://dx.doi.org/10.1016/j.jcp.2005.08.004, URL https://www.sciencedirect.com/science/article/pii/S0021999105003748.

[8] Desmons F, Coquerelle M. A generalized high-order momentum preserving (HOMP) method in the one-fluid model for incompressible two phase flows with high density ratio. J Comput Phys 2021;437:110322. http://dx.doi.org/10.1016/j.jcp.2021.110322, URL https://www.sciencedirect.com/science/article/pii/S0021999121002175.

[9] Wang R, Spiteri RJ. Linear instability of the fifth-order WENO method. SIAM J Numer Anal 2007;45(5):1871–901. http://dx.doi.org/10.1137/050637868, URL https://epubs.siam.org/doi/abs/10.1137/050637868.

[10] El Ouafa M, Vincent S, Le Chenadec V. Monolithic solvers for incompressible two-phase flows at large density and viscosity ratios. Fluids 2021;6(1):23. http://dx.doi.org/10.3390/fluids6010023.

[11] Yang Z, Lu M, Wang S. A robust solver for incompressible high-Reynolds-number two-fluid flows with high density contrast. J Comput Phys 2021;110474. http://dx.doi.org/10.1016/j.jcp.2021.110474.

[12] Nangia N, Griffith BE, Patankar NA, Bhalla APS. A robust incompressible Navier-Stokes solver for high density ratio multiphase flows. J Comput Phys 2019;390:548–94. http://dx.doi.org/10.1016/j.jcp.2019.03.042, arXiv:1809.01008.

[13] Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. J Comput Phys 2000;162(2):301–37. http://dx.doi.org/10.1006/jcph.2000.6537.

[14] Li Z, Liu C, Wan D, et al. Numerical simulations of droplet impact onto a pool surface. In: The 30th international ocean and polar engineering conference. International Society of Offshore and Polar Engineers; 2020.

[15] Zeng Y, Liu H, Gao Q, Almgren A, Bhalla APS, Shen L. A consistent adaptive level set framework for incompressible two-phase flows with high density ratios and high Reynolds numbers. J Comput Phys 2023;478:111971. http://dx.doi.org/10.1016/j.jcp.2023.111971.

[16] Marić T, Kothe DB, Bothe D. Unstructured un-split geometrical volume-of-fluid methods–a review. J Comput Phys 2020;420:109695. http://dx.doi.org/10.1016/j.jcp.2020.109695.

[17] Roenby J, Bredmose H, Jasak H. A computational method for sharp interface advection. R Soc Open Sci 2016;3(11). http://dx.doi.org/10.1098/rsos.160405, arXiv:1601.05392.

[18] Scheufler H, Roenby J. Accurate and efficient surface reconstruction from volume fraction data on general meshes. J Comput Phys 2019;383:1–23. http://dx.doi.org/10.1016/j.jcp.2019.01.009.

[19] Scheufler H, Roenby J. TwoPhaseFlow: A framework for developing two phase flow solvers in openfoam. OpenFOAM® J 2023;3:200–24. http://dx.doi.org/10.51560/ofj.v3.80.

[20] Marić T, Höpken J, Mooney KG. The OpenFOAM technology primer. Zenodo; 2021, http://dx.doi.org/10.5281/zenodo.4630596.

[21] Jasak H. Error analysis and estimation for the finite volume method with applications to fluid flows. Imperial College London (University of London); 1996, URL https://spiral.imperial.ac.uk/handle/10044/1/8335.

[22] Moukalled F, Mangani L, Darwish M, et al. In: The finite volume method in computational fluid dynamics:An Advanced Introduction with OpenFOAM® and Matlab®, vol. 6, Springer; 2016, p. 443–50, URL https://link.springer.com/book/10.1007/978-3-319-16874-6.

[23] Ghods S, Herrmann M. A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods. Phys Scr 2013;88(T155). http://dx.doi.org/10.1088/0031-8949/2013/T155/014050, URL https://iopscience.iop.org/article/10.1088/0031-8949/2013/T155/014050.

[24] Fuster D, Arrufat T, Crialesi-Esposito M, Ling Y, Malan L, Pal S, et al. A momentum-conserving, consistent, Volume-of-Fluid method for incompressible flow on staggered grids. Comput & Fluids 2018. arXiv:1811.12327.

[25] Zuzio D, Orazzo A, Estivalèzes JL, Lagrange I. A new efficient momentum preserving Level-Set/VOF method for high density and momentum ratio incompressible two-phase flows. J Comput Phys 2020;410:109342. http://dx.doi.org/10.1016/j.jcp.2020.109342.

[26] Pal S, Fuster D, Zaleski S. A novel momentum-conserving, mass-momentum consistent method for interfacial flows involving large density contrasts. 2021, http://dx.doi.org/10.48550/ARXIV.2101.04142, arXiv. URL https://arxiv.org/abs/2101.04142.

[27] Arrufat T, Crialesi-Esposito M, Fuster D, Ling Y, Malan L, Pal S, et al. A mass-momentum consistent, Volume-of-Fluid method for incompressible flow on staggered grids. Comput & Fluids 2021;215:104785. http://dx.doi.org/10.1016/j.compfluid.2020.104785.

[28] Liu C, Gao R, Hu C. A consistent mass–momentum flux computation method for the simulation of plunging jet. Phys Fluids 2022;34(3):032114. http://dx.doi.org/10.1063/5.0084894.

[29] Jin Q, Hudson D, Price WG. A combined volume of fluid and immersed boundary method for modeling of two-phase flows with high density ratio. J Fluids Eng 2022;144(3). http://dx.doi.org/10.1115/1.4052242.

[30] Tolle T, Bothe D, Marić T. SAAMPLE: A segregated accuracy-driven algorithm for multiphase pressure-linked equations. Comput & Fluids 2020;200:104450. http://dx.doi.org/10.1016/j.compfluid.2020.104450.

[31] Marić T, Marschall H, Bothe D. An enhanced un-split face-vertex flux-based VoF method. J Comput Phys 2018;371:967–93. http://dx.doi.org/10.1016/j.jcp.2018.03.048.

[32] Marić T, Liu J. Inconsistencies in unstructured volume-of-fluid methods for two-phase flows with high density ratios - code. Technical University of Darmstadt; 2023, http://dx.doi.org/10.48328/tudatalib-1276, URL https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/4018.

[33] Marić T, Liu J. Inconsistencies in unstructured volume-of-fluid methods for two-phase flows with high density ratios - data. Technical University of Darmstadt; 2023, http://dx.doi.org/10.48328/tudatalib-1277, URL https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/4019.

[34] Marić T, Liu J. Inconsistencies in unstructured volume-of-fluid methods for two-phase flows with high density ratios - code repository. 2023, URL https://github.com/tmaric/TwoPhaseFlow/tree/feature/density-ratio.

[35] Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. J Comput Phys 1992;100(2):335–54. http://dx.doi.org/10.1016/0021-9991(92)90240-Y.

[36] Denner F, van Wachem BG. Numerical time-step restrictions as a result of capillary waves. J Comput Phys 2015;285:24–40. http://dx.doi.org/10.1016/j.jcp.2015.01.021.

[37] Bussmann M, Kothe DB, Sicilian JM. Modeling high density ratio incompressible interfacial flows. Am Soc Mech Eng Fluids Eng Divis (Public) FED 2002;257(1 B):707–13. http://dx.doi.org/10.1115/FEDSM2002-31125.

[38] Scheufler H, Roenby J. TwoPhaseFlow: An openfoam based framework for development of two phase flow solvers. 2021, http://dx.doi.org/10.48550/arXiv.2103.00870, arXiv preprint.

[39] Desjardins O, Moureau V. Methods for multiphase flows with high density ratio. In: Center for Turbulence Research Proceedings of the Summer Program. 2010, p. 313–22, URL https://web.stanford.edu/group/ctr/Summer/SP10/6_02_desjardins.pdf.

[40] Raessi M, Pitsch H. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. Comput & Fluids 2012;63:70–81. http://dx.doi.org/10.1016/j.compfluid.2012.04.002, URL http://dx.doi.org/10.1016/j.compfluid.2012.04.002.

[41] Chenadec VL, Pitsch H. A 3D unsplit Forward/Backward Volume-of-Fluid approach and coupling to the level set method. J Comput Phys 2013;233(1):10–33. http://dx.doi.org/10.1016/j.jcp.2012.07.019.

[42] Vaudor G, Berlemont A, Ménard T, Doring M. A consistent mass and momentum flux computation method using rudman-type technique with a clsvof solver. In: Fluids engineering division summer meeting, vol. 46230, American Society of Mechanical Engineers; 2014, http://dx.doi.org/10.1115/FEDSM2014-21802, V01CT23A012.

[43] Crank J, Nicolson P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. Math Proc Cambridge Philos Soc 1947;43(1):50–67. http://dx.doi.org/10.1017/S0305004100023197.

[44] Rubin S, Khosla P. Higher-order numerical solutions using cubic splines. AIAA J 1976;14(7):851–8. http://dx.doi.org/10.2514/3.61427.

[45] van Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. J Comput Phys 1979;32(1):101–36. http://dx.doi.org/10.1016/0021-9991(79)90145-1, URL https://www.sciencedirect.com/science/article/pii/0021999179901451.

[46] Leonard B. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. Comput Methods Appl Mech Engrg 1979;19(1):59–98. http://dx.doi.org/10.1016/0045-7825(79)90034-3, URL https://www.sciencedirect.com/science/article/pii/0045782579900343.

[47] Roe PL. Characteristic-based schemes for the Euler equations. Annu Rev Fluid Mech 1986;18(1):337–65. http://dx.doi.org/10.1146/annurev.fl.18.010186.002005.

[48] van Leer B. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. J Comput Phys 1974;14(4):361–70. http://dx.doi.org/10.1016/0021-9991(74)90019-9, URL https://www.sciencedirect.com/science/article/pii/0021999174900199.

[49] Adamson AW, Gast AP, et al. Physical chemistry of surfaces, vol. 150. 6th Ed.. New York: Wiley; 1997, URL https://www.wiley.com/en-us/Physical+Chemistry+of+Surfaces%2C+6th+Edition-p-9780471148739.

[50] Couderc F. Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles. Application à la désintégration assistée d'un jet liquide par un courant gazeux [Ph.D. thesis], Ecole nationale superieure de l'aeronautique et de l'espace; 2007, URL https://theses.hal.science/tel-00143709/document.

[51] Zuzio D, Estivalezes J. An efficient block parallel AMR method for two phase interfacial flow simulations. Comput & Fluids 2011;44(1):339–57. http://dx.doi.org/10.1016/j.compfluid.2011.01.035, URL https://www.sciencedirect.com/science/article/pii/S0045793011000429.

[52] Zuzio D, Estivalèzes J-L, DiPierro B. An improved multiscale Eulerian–Lagrangian method for simulation of atomization process. Comput & Fluids 2018;176:285–301. http://dx.doi.org/10.1016/j.compfluid.2016.12.018, URL https://www.sciencedirect.com/science/article/pii/S0045793016304017.

[53] Xavier T, Zuzio D, Averseng M, Estivalezes J-L. Toward direct numerical simulation of high speed droplet impact. Meccanica 2020;55:387–401, URL https://link.springer.com/article/10.1007/s11012-019-00980-x.

[54] Anjos G, Borhani N, Mangiavacchi N, Thome J. A 3D moving mesh finite element method for two-phase flows. J Comput Phys 2014;270:366–77. http://dx.doi.org/10.1016/j.jcp.2014.03.067, URL https://www.sciencedirect.com/science/article/pii/S0021999114002551.

[55] Bhaga D, Weber ME. Bubbles in viscous liquids: Shapes, wakes and velocities. J Fluid Mech 1981;105:61–85. http://dx.doi.org/10.1017/S002211208100311X.

[56] Hua J, Lou J. Numerical simulation of bubble rising in viscous liquid. J Comput Phys 2007;222(2):769–95. http://dx.doi.org/10.1016/j.jcp.2006.08.008, URL https://linkinghub.elsevier.com/retrieve/pii/S0021999106003949.

[57] Thuillet S. Simulation multi-échelle de l'atomisation d'un jet liquide sous l'effet d'un écoulement gazeux transverse en présence d'une perturbation acoustique [Ph.D. thesis], UNIVERSITE DE TOULOUSE; 2018, URL https://hal.archives-ouvertes.fr/tel-02023561.

[58] Desclaux A, Thuillet S, Zuzio D, Senoner J-M, Sebbane D, Bodoc V, et al. Experimental and numerical characterization of a liquid jet injected into air crossflow with acoustic forcing. Flow Turbul Combust 2020;105(4):1087–117. http://dx.doi.org/10.1007/s10494-020-00126-0, URL https://hal.archives-ouvertes.fr/hal-03104296.

[59] Bodoc V, Desclaux A, Gajan P, Simon F, Illac G. Characterization of confined liquid jet injected into oscillating air crossflow. Flow Turbul Combust 2020;104(1):1–18. http://dx.doi.org/10.1007/s10494-019-00037-9, URL https://hal.archives-ouvertes.fr/hal-02780211.

[60] OpenCFD Ltd. OpenFOAM-v2306. 2023, URL https://develop.openfoam.com/Development/openfoam/-/tree/OpenFOAM-v2306.