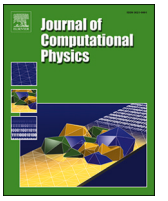
Journal of Computational Physics 566 (2026) 115290

Contents lists available at ScienceDirect

Journal of Computational Physics

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



A particle-grid discontinuous Galerkin method, Part 1: Weakly compressible free-surface and multiphase flows

Tianrun Gao, David Hyde *

Department of Computer Science, Vanderbilt University, 1400 18th Avenue South, Nashville, TN, 37212-2846, USA

ARTICLE INFO

Keywords:
Particle-grid method
Discontinuous Galerkin
Weakly compressible flow
Free-surface flow
Multiphase flow

ABSTRACT

This study proposes a particle-grid discontinuous Galerkin (PG-DG) method for weakly compressible free-surface and multiphase flows. In the particle-to-grid (P2G) step, particles are mapped onto DG cells according to the cell *phase status* determined by the *phase fraction*. The phase fraction is computed directly from particles, eliminating the longstanding issue of mass dissipation. Field variables are then updated on the DG cells, and in the grid-to-particle (G2P) step, the updated values are projected back onto particles at the beginning of a time step using DG test functions. For free-surface flows, numerical fluxes are evaluated based on whether a neighboring cell is vacuum. For multiphase interfacial regions, a post-mixing strategy is introduced within the PG-DG framework. In addition, surface particles can be naturally identified, which helps maintain a uniform particle distribution. Benchmark results show that the PG-DG method is stable and accurate for free-surface flows, high-density-ratio multiphase flows, and hybrid flow problems in both two and three dimensions. The method remains reliable even at coarse resolutions, where conventional meshless methods often struggle to achieve comparable accuracy. Moreover, the proposed approach guarantees mass conservation in the particle domain and exhibits significantly lower energy dissipation than conventional meshless methods in multiphase flow simulations. Overall, the PG-DG method combines the high fidelity of mesh-based methods with the mass-conserving advantage of particle-based approaches.

1. Introduction

Incompressible and weakly compressible free-surface and multiphase flows are ubiquitous in nature and arise in a wide range of engineering applications, such as ship hydrodynamics, propeller flows, and dam-break problems. Accurate simulation of these flows is of great importance for the design, analysis, and optimization of critical systems in ocean, civil, aeronautical, and biomedical engineering, and has long been a major research topic in computational fluid dynamics (CFD). Conventional mesh-based CFD methods have been extensively developed, including the finite-difference method (FDM) [1], finite-volume method (FVM) [2], and finite-element method (FEM) [3]. For incompressible flows without interfaces or free surfaces, these methods are readily applicable [4,5].

However, flows with fluid interfaces—and the ensuing complex evolution of said interfaces—require additional care. Interface-capturing or interface-reconstruction techniques are typically required, such as the volume-of-fluid (VOF) method [6–10] or the level-set method [11–15], in which the interface is represented as either a diffuse or a sharp interface [16]. However, the vanilla level-set method tends to introduce mass dissipation in long-term simulations [17], and VOF methods often suffer from interface smearing

* Corresponding author.
E-mail address: david.hyde.1@vanderbilt.edu (D. Hyde).

https://doi.org/10.1016/j.jcp.2026.115290
Received 6 May 2026; Received in revised form 18 July 2026; Accepted 7 August 2026
Available online 10 August 2026

[6,9]. In addition, interface reconstruction usually involves computationally costly algorithms, especially in three-dimensional cases [18].

The conventional methods for interfacial flows discussed above are all based on the Eulerian framework. By contrast, a Lagrangian description is often more natural for tracking moving interfaces. Based on this idea, methods such as the arbitrary Lagrangian-Eulerian (ALE) method [19] and the front-tracking method [20] have been developed. However, these methods require meshes to track the interface and hence often suffer from mesh distortion [21]. Frequent remeshing is usually needed when the interface undergoes large and complex deformations, such as in the presence of violent jets or splashing [22].

An alternative to mesh-based methods from a Lagrangian perspective is provided by meshless methods, in which the flow is represented solely by numerical particles. Smoothed particle hydrodynamics (SPH), originally proposed in [23] and later extended to fluid simulations [24,25], is one of the earliest and most widely used meshless methods. In recent years, numerous studies have been devoted to SPH for interfacial flows [26–29]. Although SPH has inherent advantages for free-surface and multiphase flows, its accuracy is generally below first order when the particle distribution becomes non-uniform [30]. In addition to the conventional weakly compressible SPH formulation, other particle methods have also been developed, such as the moving particle semi-implicit (MPS) method [31] and incompressible SPH [32]. There are also semi-meshless methods, such as the element-free Galerkin (EFG) method [33] and the reproducing kernel particle method (RKPM) [34], although their applications to interfacial flows have been relatively limited.

Considering that purely mesh-based and purely meshless methods both suffer from limitations, either in interface tracking or in accuracy, a hybrid Eulerian-Lagrangian approach appears to be a promising alternative. Such an approach combines grid-based and particle-based discretizations, and is commonly referred to as a particle-grid (PG) method. The earliest notable PG method is the so-called particle-in-cell (PIC) method proposed in [35], in which field variables are mapped back and forth between particles and grid cells. The major drawback of PIC is its excessive numerical dissipation. To address this issue, the fluid implicit particle (FLIP) method [36] was proposed, in which the time derivatives of variables are transferred to particles for field updates. Based on this idea, the material point method (MPM) was introduced in [37], initially for solid mechanics. More recently, MPM has been extended to fluid simulations, including compressible flows [38,39] and incompressible or weakly compressible flows [40–42]. Nonetheless, simulating fully incompressible flow using MPM can often require implicit or semi-implicit approaches for sufficient stability, which can hamper computational performance [43].

Treating fluid as incompressible or weakly compressible is widely adopted for simulating low-Mach flows. Simulation of totally incompressible flow requires solving a pressure Poisson equation, which is usually expensive. In comparison, weakly compressible flow formulations have attracted considerable attention, in which an equation of state is introduced to relate density to pressure, as is commonly done in SPH [24], although a smaller time step is generally required. Within the MPM framework, however, the simulation of strongly compressible flows is often easier than that of weakly compressible flows, because the so-called volumetric locking issue becomes more pronounced in weakly compressible cases. A number of methods have been proposed to mitigate this artifact [44–46]. Nevertheless, most existing studies focus on weakly compressible single-phase flows, while the performance of these methods for multiphase flows remains unclear, especially for cases with large density ratios. In [47], strongly compressible multiphase flows were studied using MPM. Although an improved formulation was employed, strong spurious pressure oscillations across the interface are still observed. Similar difficulties may also arise in weakly compressible multiphase flows with large density ratios. Overall, although MPM performs well across various simulation tasks, it may not be the most suitable approach for weakly compressible multiphase flows[1].

Apart from MPM, several other particle-grid methods have been developed for incompressible multiphase flows. The marker-and-cell (MAC) method is a typical example of a PG method [49]. However, it is based on a pressure-projection framework, in which the pressure treatment must be strictly consistent with the free-surface boundary conditions. In addition, free-surface treatments such as merging, de-merging, and the removal of surface undulations can be rather complicated. In [50], a hybrid PG method was proposed for incompressible multiphase flows, in which only one phase was represented by particles and the other by the grid. Such a formulation is difficult to extend to flows involving more than two phases. A similar idea has also been adopted in SPH–FVM coupling methods [51,52]. These methods are, in a sense, close to the traditional PIC method, for which the major drawback, namely numerical dissipation caused by repeated data transfer between particles and grid cells, remains unresolved. In addition, there is still a lack of particle-grid methods for weakly compressible multiphase flows. Existing incompressible PG methods cannot be directly extended to such problems, because in weakly compressible formulations the pressure is computed from the density rather than from a velocity-based incompressibility constraint. Under these circumstances, the conventional particle-to-grid interpolation is no longer directly applicable.

The simulation of weakly compressible flows with interfaces is typically carried out using explicit FVM, due to its maturity in handling convection-dominated problems. Although FEM-based methods, relying on the continuous Galerkin (CG) approach, have been developed in recent decades, they do not show clear advantages over FVM for fluid simulations. A similar situation arises in particle-grid methods such as MPM, which are normally CG-based, and this is one reason volumetric locking is prominent in MPM. In this context, a method that combines the favorable features of both CG and FVM would be a desirable choice for weakly compressible interfacial flows. This naturally motivates the use of the discontinuous Galerkin (DG) method [53–56]. Since DG admits

[1] We remark that in [48], a multi-material problem was studied using MPM, where only one phase was represented by MPM particles, while the air phase was treated using a mesh-based method. As a result, that approach was not investigated for genuinely multiphase flows with complex interfaces.

discontinuities across cell interfaces and constructs the solution locally within each cell, it combines key advantages of both FEM and FVM, making it well suited for fluid simulations. In particular, many mature FVM techniques can be naturally incorporated into the DG framework. Several DG methods have been developed for strongly compressible flows [57–59], whereas DG methods for weakly compressible multiphase flows remain very limited. To the best of our knowledge, only [60] addresses this problem, where a volume-fraction formulation is adopted and the interface appears to suffer from severe smearing. In fact, weakly compressible flows are not necessarily easier to simulate than strongly compressible flows. When diffuse-interface methods, such as the VOF approach, are used for cases with large density ratios, numerical dissipation near the interface can become very pronounced, especially in methods that rely on fluxes computed by Riemann solvers. Besides DG, FVM has also been applied to weakly compressible multiphase flows [61,62]. Similar to [60], however, these methods are also based on diffuse-interface formulations and therefore cannot avoid excessive interface smearing. By contrast, particle-grid DG approaches remain largely underexplored. In [63], PIC was combined with DG to simulate plasma dynamics, which actually has little relation with weakly compressible multiphase flows. More recently, DG has been incorporated into MPM [64,65] for impact problems in solid mechanics under a total Lagrangian framework, which is not suitable for fluid descriptions. Moreover, a critical issue of that method is that the volume integration performed at particles is not fully consistent with the surface integration at cell interfaces, which may restrict its application to problems with complex boundaries. Therefore, a more robust and accurate particle-grid DG method for fluid simulation is still needed. One may anticipate that a PG-DG method for weakly compressible flows with interfaces may offer attractive advantages in terms of accuracy, interface sharpness, and mass conservation owing to its combined particle-grid nature. However, extending DG-type formulations to free-surface and multiphase flows remains both challenging and largely unexplored.

In this work, we propose a particle-grid discontinuous Galerkin (PG-DG) method for weakly compressible free-surface and multiphase flows. Different sets of particles are used to represent different fluid phases, serving as both markers and carriers of field variables, while a background DG grid is employed to update the solution fields. Instead of solving an additional advection equation for the phase fraction, the phase fraction is computed directly from particles in neighboring cells, thereby avoiding the mass-dissipation issue associated with conventional treatments. To improve efficiency and robustness in the particle-to-grid (P2G) step, a phase-status-based mapping strategy is introduced for newly occupied cells by taking advantage of the DG framework. In the grid-to-particle (G2P) step, the updated field values are transferred back to particles using DG test functions. Free-surface fluxes are constructed according to whether a neighboring cell is vacuum. For multiphase flows, the field variables of each phase are first updated separately within their respective phase domains. A post-mixing rule is then introduced to enforce consistent pressure and velocity fields across different phases. This treatment significantly reduces numerical dissipation near the interface compared with conventional diffuse-interface methods.

The remainder of this paper is organized as follows. Section 2 presents the governing equations for weakly compressible flow and the DG formulation. Section 3 introduces the particle-grid mapping procedures, the PG-DG discretization, the particle update strategy, and related particle treatments. Section 4 describes the flux construction for free-surface and multiphase interfaces, as well as the post-mixing rule. Section 5 summarizes the overall solution procedures. Section 6 presents a set of challenging numerical examples to validate the proposed method. Finally, concluding remarks are given.

2. Weakly compressible Navier–Stokes equations

2.1. Governing equations

When flow velocity is relatively low, fluid compressibility effects are weak, and thermal effects can thus be neglected. Under this assumption, the weakly compressible Navier–Stokes (NS) equations without the energy equation can be written in hyperbolic form as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}, \tag{1}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \boldsymbol{v} \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \boldsymbol{v}^{\mathrm{T}} \\ \rho \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} + p\mathbf{I} - \boldsymbol{\tau} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \rho \boldsymbol{f} \end{pmatrix}. \tag{2}$$

where ρ, $\boldsymbol{v}$, p and $\boldsymbol{f}$ denote the density, velocity, pressure and body force, respectively, and $\mathbf{I}$ is the identity matrix. The viscous stress tensor $\boldsymbol{\tau}$ is expressed as

$$\boldsymbol{\tau} = \mu\left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}}\right) - \frac{2}{3}\mu(\nabla \cdot \boldsymbol{v})\mathbf{I}, \tag{3}$$

where μ is dynamic viscosity.

For low-speed flow, Tait's equation in barotropic form is frequently employed as the equation of state (EOS) for a fluid [25,66], expressed as

$$p = \frac{\rho_0 c^2}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] = \rho_0 c^2\left[\epsilon + \frac{\gamma - 1}{2}\epsilon^2 + \ldots\right], \tag{4}$$

with $\epsilon = \frac{\rho - \rho_0}{\rho_0}$. Here, ρ_0, c and γ are reference density, artificial sound speed and a material-dependent scalar coefficient, respectively. In weakly compressible flow, the compressibility ϵ is usually less than $1\,\%$ as long as the artificial sound speed satisfies

$$c \geq 10 \max(U_{\max}, \sqrt{p_{\max}/\rho_0}), \tag{5}$$

according to [25,26]. Here, $U_{\max}$ and $p_{\max}$ denote the maximum velocity and pressure in the fluid domain, respectively. Thus, under these flow conditions, the EOS may be simply linearized as

$$p = c^2(\rho - \rho_0), \tag{6}$$

where the error is only $\mathcal{O}(10^{-4})$ based on Eq. (5). This EOS is widely used in weakly compressible flow owing to its simplicity [25,26,60], and we adopt it for our experiments in the present study.

2.2. Discontinuous Galerkin method for weakly compressible flow

Given a Cartesian grid, we index a cell $i \in C$ by (i', j', k'), and resolutions in each direction are denoted by Δx, Δy and Δz (for problems in three spatial dimensions; for 2D problems, we drop k' and Δz). Then a cell's domain may be expressed as $\Omega_i = [x_{i'-\frac{1}{2}}, x_{i'+\frac{1}{2}}] \times [y_{j'-\frac{1}{2}}, y_{j'+\frac{1}{2}}] \times [z_{k'-\frac{1}{2}}, z_{k'+\frac{1}{2}}]$, and the cell volume and cell center are $\Delta x \Delta y \Delta z$ and $\boldsymbol{x}_{c,i} = [(x_{i'-\frac{1}{2}} + x_{i'+\frac{1}{2}})/2, (y_{j'-\frac{1}{2}} + y_{j'+\frac{1}{2}})/2, (z_{k'-\frac{1}{2}} + z_{k'+\frac{1}{2}})/2]$, respectively. Based on the hyperbolic form of the weakly compressible NS equations above, we apply the Galerkin method to Eq. (1) cell-wise, written as

$$\int_{\Omega_i} \delta \mathbf{U}^{\mathrm{T}} \left[\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) - \mathbf{S} \right] dV = 0. \tag{7}$$

In the DG method, the test function space is chosen as $\mathcal{V}_h = \left\{ \Phi : \Phi \in \mathbb{P}^1(\Omega_i) \right\}$, where $\mathbb{P}^1(\Omega_i)$ is linear polynomial space in the cell domain Ω_i. Then, for an arbitrary position $\boldsymbol{x}$ in the cell, i.e., $\forall \boldsymbol{x} \in \Omega_i$, we use the relative position, $\boldsymbol{x} - \boldsymbol{x}_c = (x, y, z)$, and then the cell-wise field variable $\mathbf{U}_h$ is approximated as

$$\mathbf{U}_h = \mathbf{U}^{(0)} + \mathbf{U}^{(1)}x + \mathbf{U}^{(2)}y + \mathbf{U}^{(3)}z = [\mathbf{U}][\mathbf{\Phi}]^{\mathrm{T}} = \mathbf{U}^{(0)} + [\mathbf{U}'][\mathbf{\Phi}']^{\mathrm{T}}, \tag{8}$$

where

$$\begin{aligned} [\mathbf{U}] &= \left(\mathbf{U}^{(0)}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \right), \quad [\mathbf{\Phi}] = (1, x, y, z), \\ [\mathbf{U}'] &= \left(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \right), \quad [\mathbf{\Phi}'] = (x, y, z). \end{aligned} \tag{9}$$

Applying Eq. (8) to Eq. (7) and using the variational principle, the equation corresponding to the I^{th} component of $[\mathbf{U}]$ is derived as

$$\frac{\partial \mathbf{U}^{(J)}}{\partial t} \int_{\Omega_i} \Phi^{(I)} \Phi^{(J)} dV = \int_{\Omega_i} \mathbf{F} \nabla \Phi^{(I)} dV - \int_{\partial \Omega_i} \mathbf{F} \Phi^{(I)} \boldsymbol{n} dS + \int_{\Omega_i} \mathbf{S} \Phi^{(I)} dV, \quad I = 0, 1, 2, 3. \tag{10}$$

This equation shows that the cell-averaged value $\mathbf{U}^{(0)}$ and high-order values $\mathbf{U}^{(1,2,3)}$ should be updated separately. Noting that the employed test function in $\mathcal{V}_h$ space is orthogonal, we have $\int_{\Omega_i} \Phi^{(I)} \Phi^{(J)} dV = M_I \delta_{IJ}$, which makes the time advancement more efficient.

3. Field mapping between particles and grid

In the PG-DG method, particles serve as both markers and carriers of field variables. An overview of the method is illustrated in Fig. 1. Different fluid phases are represented by different sets of particles, as shown in Fig. 1(a). For example, the two particle sets are shown in red and blue in Fig. 1(b) and (c), respectively. In the present method, particle-to-grid mapping is unnecessary over all cells as long as their *phase status* remains unchanged. Here, a cell's status is determined by whether the cell and its neighboring cells contain at least one fluid particle, which requires a grid-status identification procedure. When the cell status changes, the field variables in the corresponding cells are reconstructed from the particles through interpolation. The multiphase interface is treated using the phase fraction, as shown in Fig. 1(d). Details of these procedures are presented in the following sections.

3.1. Particle-to-grid mapping

3.1.1. Phase fraction determination

In the present work, we restrict our attention to structured Cartesian grids, with particles belonging to associated grid cells. Let $\mathcal{P}$ be the set of particles, and each particle $p \in \mathcal{P}$ belong to exactly one cell via a mapping

$$\kappa : \mathcal{P} \to C.$$

Define the set of particles in cell i as

$$\mathcal{P}_i := \{ p \in \mathcal{P} \mid \kappa(p) = i \}. \tag{11}$$

In this way, the particles are visited very easily by searching neighbors of the corresponding cell. Unlike conventional PIC or MPM, the particle resolution $\Delta\bar{x}$ (initial particle spacing) is set equal to the grid resolution, i.e., $\Delta\bar{x} = \sqrt{\Delta x \Delta y}$ or $\sqrt[3]{\Delta x \Delta y \Delta z}$ for 2D or 3D, respectively. This approach greatly reduces the number of particles for a given resolution of the grid. In 2D cases, the cell neighbor set $\mathcal{N}_i$ of a cell i with index (i', j') is shown in Fig. 2, which involves (i', j') and one cell layer around the cell (i', j'). This principle is also used in the 3D scenario, but only the neighbors within radius $1.1\Delta\bar{x}$ of the target cell are used for compactness in 3D cases.

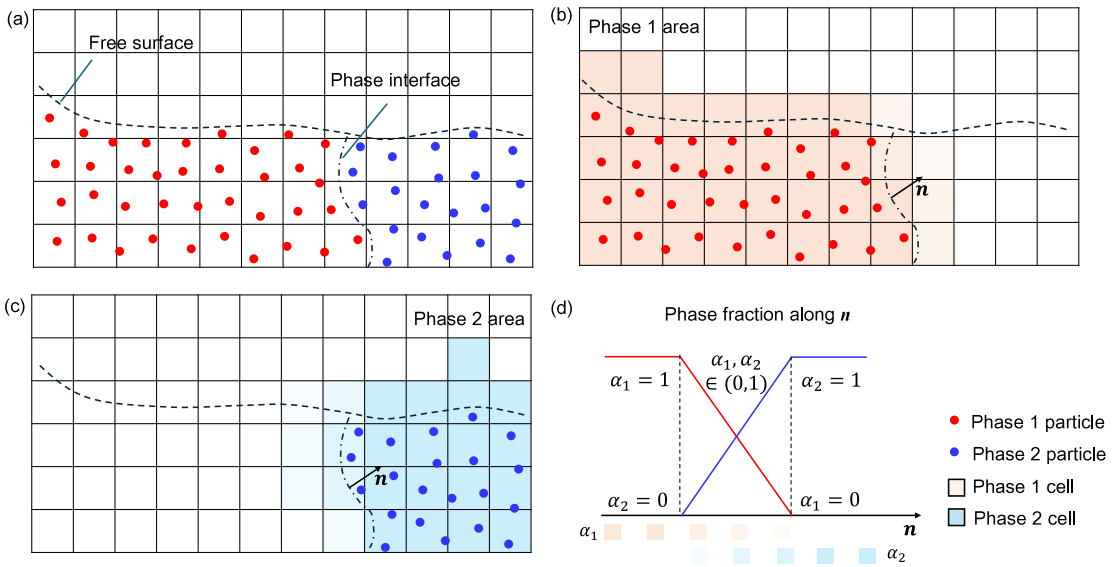


Fig. 1. Particle-grid DG treatment of free-surface and multiphase flows. (a) Red and blue particles represent two different fluids in the background DG domain. (b) and (c) DG cells associated with phase 1 and phase 2, respectively. (d) Phase fraction in the interfacial region, indicated by color shading. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

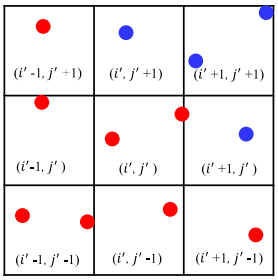


Fig. 2. Neighbor set $\mathcal{N}_i$ of a cell i with index (i', j'), where red and blue dots represent two different phases in the field. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For a particle p, we define the function $\omega(p) = \chi$ to probe the fluid phase χ. To calculate the phase fraction of cell i for phase χ, $\alpha_{i,\chi}$, firstly we need to calculate $\alpha'_{i,\chi}$ defined by

$$\alpha'_{i,\chi} = \sum_{\omega(p)=\chi, p\in\mathcal{P}_j, j\in\mathcal{N}_i} W\left(\left\|\boldsymbol{x}_{ip}\right\|, h\right), \tag{12}$$

where $\boldsymbol{x}_{ip} = \boldsymbol{x}_p - \boldsymbol{x}_p$, which is also used in Eq. (21). Here, a window function such as C^2 Wendland kernel function $W(r,h)$ [67] is used, written as

$$W(r,h) = \begin{cases} C_h\left(1-\frac{r}{2h}\right)^4\left(1+\frac{2r}{h}\right), & \frac{r}{h} \le 2, \\ 0, & \frac{r}{h} > 2, \end{cases} \tag{13}$$

where we choose $h = 1.6\Delta\bar{x}$. $C_h = \frac{7}{4\pi h^2}$ and $\frac{21}{16\pi h^3}$ for 2D and 3D, respectively. In fact, any kind of kernel functions can be used because given $\alpha'_{i,\chi}$, the phase fraction $\alpha_{i,\chi}$ can be normalized as

$$\alpha_{i,\chi} = \frac{\alpha'_{i,\chi}}{\sum_k^C \alpha'_{i,k}}, \tag{14}$$

where C denotes the total number of phases in the computational domain.

Remark 1. We can show that the present calculation of phase fractions is similar to the update based on the advection equation. In continuum space, the calculation of the phase fraction above is equivalent to

$$\alpha_\chi(\boldsymbol{x}) = \frac{\int_{\Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) d\boldsymbol{x}'}{\int_{\sum_\chi^C \Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) d\boldsymbol{x}'}, \tag{15}$$

where $\Omega_\chi(t)$ denotes the occupied domain of phase χ at time t. In a flow field without vacuum phase, $\int_{\sum_\chi^C \Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) d\boldsymbol{x}'$, denoted as m_0, should be constant in space because the neighboring domain is occupied. Then the gradient of $\alpha_\chi(\boldsymbol{x})$ is calculated as

$$\nabla\alpha_\chi(\boldsymbol{x}) = -\frac{1}{m_0}\int_{\partial\Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x})\boldsymbol{n} dS. \tag{16}$$

On the other hand, the time derivative of $\alpha_\chi(\boldsymbol{x})$ can be calculated based on Eq. (15). After reorganization, it may be written as

$$\frac{\partial\alpha_\chi}{\partial t} = \frac{1}{m_0}\int_{\partial\Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x})\boldsymbol{v}\cdot\boldsymbol{n} dS = \frac{\bar{\boldsymbol{v}}}{m_0}\cdot\int_{\partial\Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x})\boldsymbol{n} dS = -\bar{\boldsymbol{v}}\cdot\nabla\alpha_\chi(\boldsymbol{x}). \tag{17}$$

Thus, rearranging, we obtain

$$\frac{\partial\alpha_\chi}{\partial t} + \bar{\boldsymbol{v}}\cdot\nabla\alpha_\chi(\boldsymbol{x}) = 0. \tag{18}$$

This result indicates that the present phase fraction is transported in a manner similar to that in VOF and level-set methods. Although its update resembles the solution of an advection equation, the proposed method has a clear advantage over conventional approaches: it does not suffer from mass loss, because fluid particles are not eliminated during the computation. In traditional interface-capturing methods, basic level-set formulations typically exhibit severe mass dissipation. By contrast, while VOF methods conserve mass, they often suffer from significant interface smearing as the simulation proceeds. In the present method, however, the thickness of the phase-fraction transition region remains unchanged and is always less than $3\Delta\bar{x}$ for the neighbor set $\mathcal{N}_i$ shown in Fig. 2.

3.1.2. Phase and fluid status

In the PG-DG method, mapping from particles to grid cells is not necessary for cells whose *phase status*, denoted with PS, does not change for any phase. We define phase status as a binary variable determined by whether the cell or its neighbors contain at least one particle of a given phase. Specifically, for a cell i with phase χ, if its phase fraction is nonzero, it is called *activated*, denoted as $PS_{i,\chi} = 1$, and otherwise it is 0, i.e.,

$$PS_{i,\chi} = \begin{cases} 1, & \alpha'_{i,\chi} > 0, \\ 0, & \alpha'_{i,\chi} = 0. \end{cases} \tag{19}$$

The *fluid status*, FS, of a grid cell is a binary variable defined by whether the neighbors of the cell contain at least one particle, which is equivalent to determining if the sum of $\alpha'_{i,\chi}$ is non-zero, i.e.,

$$FS_i = \begin{cases} 1, & \sum_\chi \alpha'_{i,\chi} > 0, \\ 0, & \sum_\chi \alpha'_{i,\chi} = 0. \end{cases} \tag{20}$$

When $FS = 0$, we label it a *vacuum cell*. This step is implemented already in the phase fraction determination step introduced in Section 3.1.1. The *fluid status* will be used to identify the surface particles.

3.1.3. Interpolation at activated cells

Unlike the traditional PIC method, which maps field values from particles to each cell, the present method only maps field values from particles to cells for cells i with phase χ whose phase status changes from 0 to 1 between time steps, i.e., $PS^n_{i,\chi} = 0 \Rightarrow PS^{n+1}_{i,\chi} = 1$, where the superscript refers to the index of the time step. When the interpolation is invoked, the DG field values, $\mathbf{U}^{(I)}_{i,\chi}$, should be interpolated from the corresponding particles in the neighboring cells, written as

$$\mathbf{U}^{(I)}_{i,\chi} = \begin{cases} \dfrac{\sum_{\omega(p)=\chi, p\in\mathcal{P}_j, j\in\mathcal{N}_i}\left(\mathbf{U}^{(0)}_p + [\mathbf{U}']_p[\mathbf{\Phi}']^{\mathrm{T}}_{pi}\right)W\left(\left\|\boldsymbol{x}_{ip}\right\|, h\right)}{\alpha'_{i,\chi}}, & I = 0, \\ \dfrac{\sum_{\omega(p)=\chi, p\in\mathcal{P}_j, j\in\mathcal{N}_i}\mathbf{U}^{(I)}_p W\left(\left\|\boldsymbol{x}_{ip}\right\|, h\right)}{\alpha'_{i,\chi}}, & I > 0. \end{cases} \tag{21}$$

where the high-order coefficients $\mathbf{U}^{(I)}_{p,\chi}$ of particles are already inherited from the corresponding cells, which will be introduced in Section 3.3, and $[\mathbf{\Phi}']_{pi}$ is calculated based on $\boldsymbol{x}_{pi}$. After this step, the field values $\mathbf{U}^{(I)}_{i,\chi}$ at grid cells will be updated using the DG approach as introduced in the next section.

Remark 2. The interpolation procedure described above accounts for higher-order terms through the DG test functions. The affine particle-in-cell (APIC) method [68], which is widely used in MPM, introduces an angular-momentum compensation term to interpolate the velocity at node i, i.e., $\boldsymbol{v}_i = \boldsymbol{v}_p + \mathbb{C}_p(\boldsymbol{x}_i - \boldsymbol{x}_p)$. This idea is also adopted in a related method known as Taylor particle-in-cell (TPIC) [69]. Such a treatment can significantly reduce kinetic-energy dissipation. In fact, the APIC term corresponds to $\mathbf{U}_p + [\mathbf{U}']_p[\mathbf{\Phi}']_{pi}^{\mathrm{T}}$ in Eq. (21) employed in the present study. This indicates that the present formulation provides a more general interpolation framework: APIC and TPIC can be viewed as first-central-moment corrections, whereas the present method can achieve higher-central-moment corrections when higher-order DG test functions are employed.

3.2. Field updates on the grid

Based on Eq. (10), the corresponding discretized form for phase χ can be written as

$$\frac{\partial \mathbf{U}_{i,\chi}^{(I)}}{\partial t} = \frac{1}{M_{Ii}}\left[\sum_{s}^{G_1}\left(\mathbf{F}_{is,\chi}\nabla\Phi_{is}^{(I)} + \mathbf{S}_{is,\chi}\Phi_{is}^{(I)}\right)w_s V_i - \sum_{j}^{N_i}\sum_{t}^{G_2}\hat{\mathbf{F}}_{ijt,\chi}\Phi_{ijt}^{(I)}\cdot\boldsymbol{n}_j S_j w_t\right], \quad I = 0,1,2,3. \tag{22}$$

Here, G_1 and G_2 denote the number of Gaussian quadrature points on the cell volume Ω_i and on the cell interface connecting with cell j, respectively. N_i denotes the number of face-adjacent neighbors of cell i, and is 4 and 6 in 2D and 3D for the Cartesian grid, respectively. We remark that the Gaussian integral is easy to implement for a structured Cartesian grid. For the viscous term, the gradient $\nabla\boldsymbol{v}$ required in the viscosity flux of Eq. (3) is estimated using a second-order central difference approximation.

On the right-hand side of Eq. (22), the first term involves computing a cell-wise volume integral, which is performed using Gaussian quadrature points s (weight w_s), which is similar to CG. For the second term, the flux on cell interface ij is estimated on the Gaussian quadrature point t (weight w_t). In advection-dominated problems, a proper Riemann solver is supposed to enhance numerical stability, and by using a Riemann solver, the flux can be expressed as

$$\hat{\mathbf{F}}_{ijt,\chi} = \frac{1}{2}\left(\mathbf{F}_{it,\chi} + \mathbf{F}_{jt,\chi}\right) - \frac{1}{2}\mathbf{A}_{ijt,\chi}\left(\mathbf{U}_{jt,\chi} - \mathbf{U}_{it,\chi}\right), \tag{23}$$

with the diagonal matrix $\mathbf{A}_{ijt,\chi}$ in 3D space as

$$\mathbf{A}_{ijt,\chi} = \diagdown(c_{ijt,\chi}, \tilde{a}_{ijt,\chi}, \tilde{a}_{ijt,\chi}, \tilde{a}_{ijt,\chi}). \tag{24}$$

Here, the diagonal entries are chosen as

$$\begin{aligned} c_{ijt,\chi} &= \xi_{ij,\chi}\frac{\bar{c}_{it} + \bar{c}_{jt}}{2}, \\ \tilde{a}_{ijt,\chi} &= \xi_{ij,\chi}\max\left(\left|\boldsymbol{v}_{it,\chi}\cdot\boldsymbol{n}_j\right|, \left|\boldsymbol{v}_{jt,\chi}\cdot\boldsymbol{n}_j\right|\right), \\ \xi_{ij,\chi} &= \frac{\alpha_{i,\chi} + \alpha_{j,\chi}}{2\alpha_{i,\chi}}, \end{aligned} \tag{25}$$

where $\bar{c}$ is provided by Eq. (53) in Section 4.2. The low-speed Riemann solver above leverages a local velocity norm for $\tilde{a}_{ijt,\chi}$ that is far less than the artificial sound speed; unlike the magnitude of sound speed used in strongly compressible flow, this smaller value can significantly reduce excessive kinetic dissipation [70].

Eq. (22) usually leads to numerical instability without a proper limiter, particularly in cases with complex interfaces. In Cockburn's work [54], a total variation bounded limiter is used to suppress instability. In this work, the B-J limiter [71] is used to enhance numerical stability. Integrating it into DG, each component $\tilde{U}_i$ of the limited field $\tilde{\mathbf{U}}_i$ is expressed as

$$\tilde{U}_{i,\chi} = U_{i,\chi}^{(0)} + \psi_{i,\chi}[U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}, \tag{26}$$

with ψ standing for the flux limiter, and it should satisfy

$$U_{i,\chi}^{(0),\min} \le \tilde{U}_{i,\chi} \le U_{i,\chi}^{(0),\max}, \tag{27}$$

where $U_{i,\chi}^{(0),\max} = \max_{j\in N_i} U_{j,\chi}^{(0)}$ and $U_{i,\chi}^{(0),\min} = \min_{j\in N_i} U_{j,\chi}^{(0)}$. Then the limiter $\psi_{i,\chi} = \min_{j\in N_i}\psi_{ij,\chi}$ for phase χ can be calculated, with $\psi_{ij,\chi}$ estimated on the cell centers, written as

$$\psi_{ij,\chi} = \begin{cases} \min\left(1, \dfrac{U_{i,\chi}^{(0),\max} - U_{i,\chi}^{(0)}}{[U']_{i,\chi}[\mathbf{\Phi}']_{ij}^{\mathrm{T}}}\right), & [U']_{i,\chi}[\mathbf{\Phi}']_{ij}^{\mathrm{T}} > 0, \\ \min\left(1, \dfrac{U_{i,\chi}^{(0),\min} - U_{i,\chi}^{(0)}}{[U']_{i,\chi}[\mathbf{\Phi}']_{ij}^{\mathrm{T}}}\right), & [U']_{i,\chi}[\mathbf{\Phi}']_{ij}^{\mathrm{T}} < 0, \\ 1, & [U']_{i,\chi}[\mathbf{\Phi}']_{ij}^{\mathrm{T}} = 0. \end{cases} \tag{28}$$

Regarding time integration, we leverage the second-order TVD Runge-Kutta scheme [72], where the high-order terms are enhanced with the limiter in two sub-time steps, expressed as

$$\begin{aligned}\mathbf{U}_{i,\chi}^{(I),n+1,*} &= \psi_{i,\chi}^{(I)}\mathbf{U}_{i,\chi}^{(I),n} + \psi_{i,\chi}^{(I)}\left(\frac{\partial \mathbf{U}_{i,\chi}^{(I)}}{\partial t}\right)^{n}\Delta t,\\ \mathbf{U}_{i,\chi}^{(I),n+1} &= \frac{1}{2}\mathbf{U}_{i,\chi}^{(I),n} + \frac{1}{2}\psi_{i,\chi}^{(I)}\mathbf{U}_{i,\chi}^{(I),n+1,*} + \frac{1}{2}\psi_{i,\chi}^{(I)}\left(\frac{\partial \mathbf{U}_{i,\chi}^{(I)}}{\partial t}\right)^{n+1,*}\Delta t,\end{aligned} \tag{29}$$

where

$$\psi_{i,\chi}^{(I)} = \begin{cases} 1, & I = 0,\\ \psi_{i,\chi}, & I > 0.\end{cases} \tag{30}$$

Here, $\psi_{i,\chi}^{(I)}$ is equal to 1 in most regions, since the field values are generally smooth in weakly compressible flows. Therefore, the term $\psi_{i,\chi}^{(I)}\mathbf{U}_{i,\chi}^{(I),n}$ is not excessively smeared. At the same time, this treatment helps to ensure high robustness in complex flows.

The time step can be determined using

$$\begin{aligned}&\Delta t = \min(\Delta t_1, \Delta t_2),\\ &\Delta t_1 = \mathrm{CFL}\cdot\frac{\Delta x}{\max(c_1,\ldots,c_C) + |\boldsymbol{v}|_{\max}}, \quad \Delta t_2 = 0.1\sqrt{\frac{\Delta x}{|\boldsymbol{a}|_{\max}}},\end{aligned} \tag{31}$$

where $|\boldsymbol{a}|_{\max}$ is the maximum acceleration in the computational domain. DG typically requires CFL $< 1/(1+2p')$, with p' the highest polynomial order of DG test functions. After the grid update, the post-mixing rule will be implemented in the interface areas to ensure consistency of pressure and velocity fields, which will be introduced in Section 4.2.

Remark 3. In this method, a CFL number of 0.3 is found to be robust for most cases, although a smaller value may be required for multiphase flows with a large density ratio. In addition, in the context of weakly compressible flow, the artificial sound speed ratio between different phases, c_1/c_2, is generally expected to satisfy a certain, albeit approximate, relation, especially when the density ratio is large [26,73]. In [74], however, it was reported that this relation can be relaxed even for large density ratios while still yielding satisfactory results in practice. A more detailed investigation of the effects of the density ratio and sound speed ratio on the CFL number was reported in [75]. It was shown that, when the density ratio is large, the CFL number should be reduced to maintain numerical stability. Moreover, for a fixed density ratio, an increase in the sound speed ratio requires a corresponding decrease in the CFL number.

In this work, CFL = 0.3 is chosen for cases with low density ratios $\rho_{0,1}/\rho_{0,2} < 10$, and for large density ratios, $\rho_{0,1}/\rho_{0,2} > 1000$, CFL = 0.15 is used in this paper. In order to ensure robustness, CFL = 0.15 is also recommended for $10 < \rho_{0,1}/\rho_{0,2} < 1000$. For more detailed prescription of the CFL number selection for weakly compressible flows, the readers are referred to [75].

3.3. Grid-to-particle mapping and field update of particles

The field values of particles are interpolated cell-wise at the beginning of each time step, i.e.,

$$\mathbf{U}_p^{(0),n} = \mathbf{U}_{i,\chi}^{(0),n} + [\mathbf{U}']_{i,\chi}^{n}[\mathbf{\Phi}']_{ip}^{\mathrm{T}}, \text{ for } \omega(p) = \chi, \tag{32}$$

where the relation between i and p is expressed in the mapping in Eq. (11). This interpolation is used to provide an initial field value at the beginning of the time step, after which the field value is updated as follows.

In accordance with the second-order TVD Runge-Kutta scheme used in the grid domain, the same time integration is adopted for the particle update, and two sub-time steps are involved. In the first sub-time step, for particle p with phase χ, $\omega(p) = \chi$, the field values are given by

$$\begin{aligned}\mathbf{U}_p^{(0),n+1,*} &= \mathbf{U}_p^{(0),n} + \left(\frac{\partial \mathbf{U}_p}{\partial t}\right)^{n}\Delta t,\\ \mathbf{U}_p^{(I),n+1,*} &= \mathbf{U}_{i,\chi}^{(I),n+1,*}, \quad I > 0,\\ \boldsymbol{x}_p^{n+1,*} &= \boldsymbol{x}_p^{n} + \boldsymbol{w}_p^{n}\Delta t,\end{aligned} \tag{33}$$

and in the second sub-time step,

$$\begin{aligned}\mathbf{U}_p^{(0),n+1} &= \frac{1}{2}\mathbf{U}_p^{(0),n} + \frac{1}{2}\mathbf{U}_p^{(0),n+1,*} + \frac{1}{2}\left(\frac{\partial \mathbf{U}_p}{\partial t}\right)^{n+1,*}\Delta t,\\ \mathbf{U}_p^{(I),n+1} &= \mathbf{U}_{i,\chi}^{(I),n+1}, \quad I > 0,\\ \boldsymbol{x}_p^{n+1} &= \frac{1}{2}\boldsymbol{x}_p^{n} + \frac{1}{2}\boldsymbol{x}_p^{n+1,*} + \frac{1}{2}\boldsymbol{w}_p^{n+1,*}\Delta t,\end{aligned} \tag{34}$$

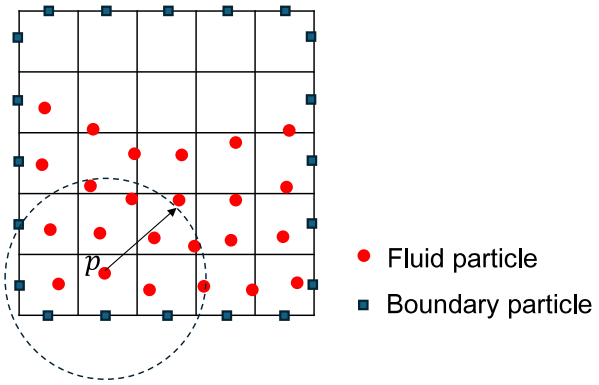


Fig. 3. Neighboring particles of a fluid particle p include fluid particles and boundary particles as well.

where for each component $U_{p,\chi}$ of $\mathbf{U}_{p,\chi}$, its time derivative is calculated as

$$\frac{\partial U_p}{\partial t} = \sum_I \psi_{i,\chi}^{(I)} \left(\frac{\partial U_{i,\chi}^{(I)}}{\partial t} \right) \Phi_{ip}^{(I)} + \nabla U_{i,\chi} \cdot \boldsymbol{w}_p. \tag{35}$$

The second term on the right-hand side takes into account the ALE feature of particles, because the grid is in an Eulerian description but the particle moves according to the transport velocity, $\boldsymbol{w}_p$, in this equation. The calculation of $\boldsymbol{w}_p$ will be introduced in Section 3.4.3.

3.4. Other issues related to particles

3.4.1. Identification of surface particles

The identification of surface particles has recently been investigated in many studies using various approaches [76–79]. In the present method, however, the identification of surface particles is relatively straightforward. It is only necessary to determine whether a non-vacuum cell is adjacent to a vacuum cell. We use $SI'_i = 1$ to mark such cells, which can be expressed mathematically as

$$SI'_i = \left\{ FS_i = 1 \text{ and } \exists\, j \in \mathcal{N}_i \text{ s.t. } FS_j = 0 \right\}. \tag{36}$$

This procedure is already determined when a cell interacts with its neighbors when calculating the interfacial fluxes. Then for a particle p belonging to surface cell $SI'_i = 1$, the particle p is marked as surface particle $SI'_p = 1$, expressed as

$$SI'_p = \left\{ \kappa(p) = i \text{ and } SI'_i = 1 \right\}. \tag{37}$$

The identification of surface particles plays an important role in the uniform distribution of particles, which will be introduced in Section 3.4.3.

3.4.2. Wall non-penetration

As shown in Fig. 3, when a particle p approaches the domain boundary, it may penetrate the boundary. This phenomenon arises because the flow near the boundary is usually not well-resolved when coarse grids and large time steps are used. To avoid this artificial behavior, a wall non-penetration strategy is adopted.

In Fig. 3, boundary particles are deployed at the face centers of boundary elements. For a particle p, it is easy to find the corresponding cell i, i.e., $\kappa(p) = i$. Then, for all boundary interfaces b in cell i, the non-penetration velocity shifting can be summed over b whose normal distance to p is less than $0.4\Delta\bar{x}$,

$$\delta \boldsymbol{v}_p^{(\mathrm{np})} = \frac{\Delta t}{2} \sum_b \frac{\varepsilon c_p^2}{\boldsymbol{x}_{pb} \cdot \boldsymbol{n}_b} \boldsymbol{n}_b, \quad \text{for } \left| \boldsymbol{x}_{pb} \cdot \boldsymbol{n}_b \right| < 0.4\Delta\bar{x}. \tag{38}$$

In this equation, $\boldsymbol{n}_b$ is the normal direction pointing outward, and ε is a coefficient selected as $\varepsilon = 0.01$.

3.4.3. Particle distribution

Among many ways to render particle distributions uniform, the kernel function-based particle shifting technique (PST) is an easy and effective choice widely used in the SPH method. This approach needs to build a neighbor set $\mathfrak{N}_p$ of a particle p, and it is defined as

$$\mathfrak{N}_p = \left\{ s \in \mathcal{P} \mid \left\| \boldsymbol{x}_{ps} \right\| < 3.2\Delta\bar{x} \right\}. \tag{39}$$

The neighbor set can be constructed very easily by searching in surrounding cells using $\kappa(p) = i$. As shown in Fig. 3, when the particle is adjacent to the boundary, the neighbor set includes fluid and boundary particles. Considering the two kinds of particles, the shifted velocity generated by its neighboring particles is expressed as

$$\delta \boldsymbol{v}_p' = -\frac{\Delta t \tilde{c}^2}{2}\left[\sum_s \left(1 + 0.2\mathcal{W}_{ps}^4\right)\nabla W_{ps} V_s + \sum_b \left(1 + 0.2\mathcal{W}_{pb}^4\right) W_{pb} \boldsymbol{n}_b S_b\right], \tag{40}$$

where $\mathcal{W}_{ps} = W\left(\left\|\boldsymbol{x}_{ps}\right\|, h\right)/W(\Delta\bar{x}, h)$, and $\tilde{c}$ denotes the arithmetic mean of the artificial sound speeds of all phases. In this step, the neighbors of surface particles, marked with SI_p, can be identified using

$$SI_p = \left\{\exists\, s \in \mathfrak{N}_p \text{ s.t. } SI_s' = 1\right\}, \tag{41}$$

and $SI_p = 1$ denotes that a particle is within the kernel radius of a surface particle. This step is similar to the method proposed in [76], and it ensures that particles with insufficient kernel support are not shifted.

The amplitude of shifting is limited using

$$\delta \boldsymbol{v}_p^{(\mathrm{pst})} = \delta \boldsymbol{v}_p' \min\left(\left\|\delta \boldsymbol{v}_p'\right\|, \frac{\Delta\bar{x}}{10\Delta t}\right)/\left\|\delta \boldsymbol{v}_p'\right\|. \tag{42}$$

In immiscible flow, the interface should be sharp between different phases to prevent spurious mixing. Numerically, there are usually penetrations into the opposite phase. Similar to the method in [80], the shifting induced by different phases is written as

$$\delta \boldsymbol{v}_p'' = -\frac{\Delta t \tilde{c}^2}{20}\sum_{s'} \nabla W\left(\left\|\boldsymbol{x}_{ps'}\right\|, h\right) V_{s'}, \tag{43}$$

where s' denotes the particles of the different phase, and then the shifting is limited by

$$\delta \boldsymbol{v}_p^{(\mathrm{s})} = \delta \boldsymbol{v}_p'' \min\left(\left\|\delta \boldsymbol{v}_p''\right\|, \frac{\Delta\bar{x}}{10\Delta t}\right)/\left\|\delta \boldsymbol{v}_p''\right\|, \tag{44}$$

Finally, combining Eqs. (38), (42) and (44), the shifted velocity $\delta \boldsymbol{v}_p$ is calculated according to SI_p as

$$\delta \boldsymbol{v}_p = \begin{cases} \delta \boldsymbol{v}_p^{(\mathrm{np})} + \delta \boldsymbol{v}_p^{(\mathrm{pst})} + \delta \boldsymbol{v}_p^{(\mathrm{s})}, & SI_p = 0, \\ \delta \boldsymbol{v}_p^{(\mathrm{np})}, & SI_p = 1. \end{cases} \tag{45}$$

After this, the shifted velocity should be added to the real velocity, giving

$$\boldsymbol{w}_p = \boldsymbol{v}_p + \delta \boldsymbol{v}_p. \tag{46}$$

This is called the transport velocity and is used in the G2P step in Eqs. (33) and (34).

4. Interface treatment for free-surface and multiphase flows

4.1. Fluxes on free surface and multiphase interface

A typical scenario covering the free surface and multiphase interface is illustrated in Fig. 4(a). For example, the cell i lies in the mixing area of the two phases, and for cell i, the statuses for the two phases are both activated, i.e., $PS_{i,1} = PS_{i,2} = 1$. If we focus on phase 1, then we can observe that cell i neighbors with a vacuum cell j and a cell k of different phase. To derive $\hat{\mathbf{F}}_{ijt,\chi}$ and $\hat{\mathbf{F}}_{ikt,\chi}$ on the free surface or multiphase interface as shown in Fig. 4(b), one should obtain $\mathbf{F}_{jt,\chi}(\rho_{jt,\chi}, \boldsymbol{v}_{jt,\chi})$ and $\mathbf{F}_{kt,\chi}(\rho_{kt,\chi}, \boldsymbol{v}_{kt,\chi})$ according to Eq. (23). For the ij interface, which links cell i with a vacuum cell j, the required field values $(\rho_{jt,\chi}, \boldsymbol{v}_{jt,\chi})$ on the interfaces at the Gaussian point t should be

$$\rho_{jt,\chi} = \rho_{0,\chi}, \quad \boldsymbol{v}_{jt,\chi} = \boldsymbol{v}_{it,\chi}, \tag{47}$$

which imposes a zero-pressure state on the vacuum side of the interface.

For the ik interface, as shown in Fig. 4(b), if we focus on phase 1, cell i links with a cell k of phase 2. The flux is constructed under the assumption that pressure and velocity field should be shared within the same cell for different phases. For example, for cell i of phase 1, the field values $(\rho_{kt,1}, \boldsymbol{v}_{kt,1})$ on the interface ik linking its neighbor k of different phase 2, should be estimated based on the field values $(\rho_{kt,2}, \boldsymbol{v}_{kt,2})$ of cell k of phase 2, expressed as

$$\rho_{kt,1} = p_{kt,2}/c_1^2 + \rho_{0,1}, \quad \boldsymbol{v}_{kt,1} = \boldsymbol{v}_{kt,2}, \tag{48}$$

where

$$p_{kt,2} = c_2^2(\rho_{kt,2} - \rho_{0,2}). \tag{49}$$

Then, $(\rho_{kt,1}, \boldsymbol{v}_{kt,1})$ is used to construct $\hat{\mathbf{F}}_{ikt,1}$, and the DG field values for phase 1 can be updated correspondingly thereafter.

Remark 4. We do not directly use the field values of phase 2 to construct the fluxes for $\hat{\mathbf{F}}_{ikt,\chi}$, although this is feasible in the VOF method. First, in the PG-DG method, the phase fraction is computed directly from the particle distribution, which implies that the field values should remain consistent with the phase fraction. If the VOF method is used for physical evolution, inconsistencies may arise between the field values and the phase fraction, which may further lead to numerical blow-up. Second, in VOF-based DG methods,

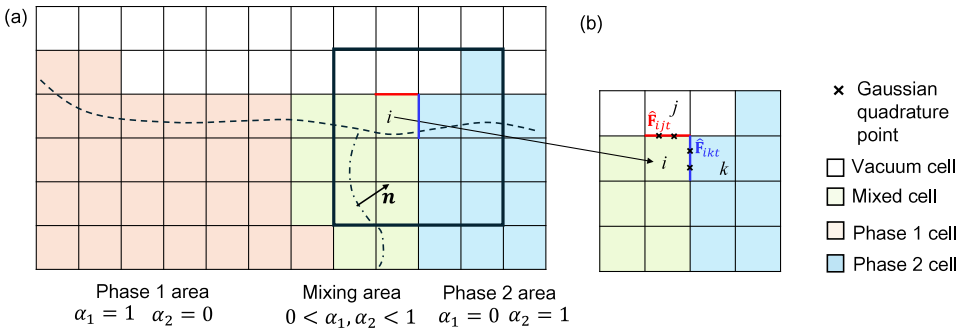


Fig. 4. Fluxes on the free surface and multiphase interfaces: cell i has a vacuum cell neighbor j and a cell neighbor k of different phase. Eqs. (47) and (48) describe appropriate flux calculations for these interfaces.

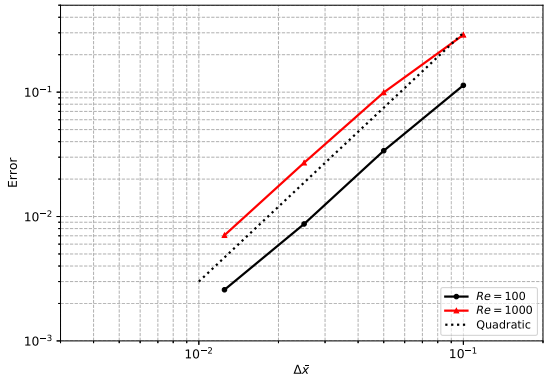


Fig. 5. Taylor-Green Vortex: L_2 error versus simulation resolution $\Delta\bar{x}$ for $Re = 100$ and 1000.

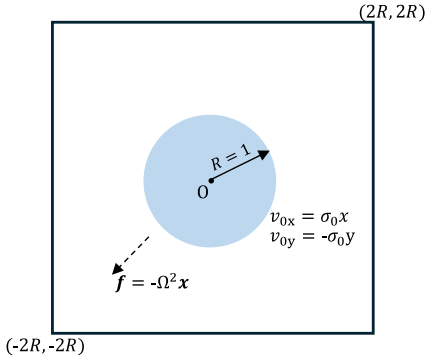
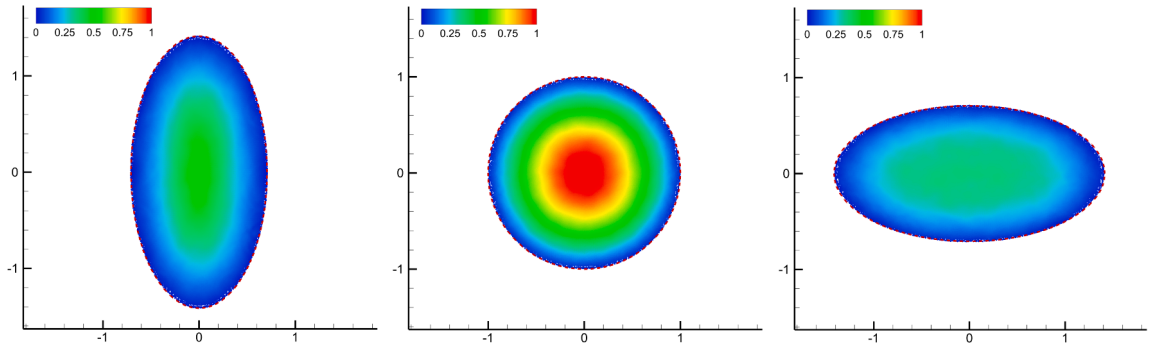


Fig. 6. Schematic of oscillating droplet and dimensions.

the higher-order terms of $\alpha\rho$ or $\bar{\rho}\boldsymbol{v}$ become extremely steep near the interface, which may threaten numerical stability, especially in multiphase flows with a large density ratio. Although strong stabilization can be applied in this region, it tends to smear the interface progressively and thus degrades the sharpness of the solution. In contrast, the present method updates different phases separately and, in fact, behaves more like a sharp-interface method in the evolution of the field values across the multiphase interface. Although a post-mixing treatment is required in the subsequent step, the low-dissipation property is still preserved.

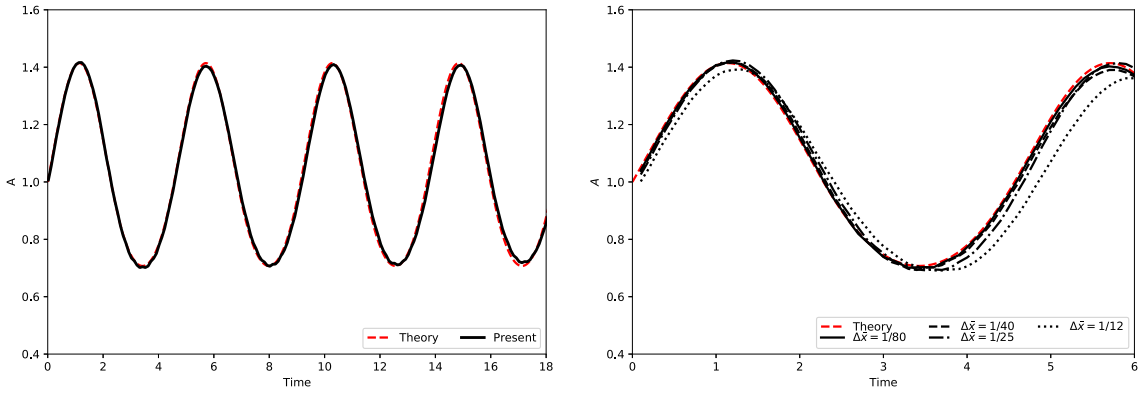
4.2. Post-mixing rule for DG field values

In the PG-DG method, different phases update their DG field values separately, and the multiple field values in the interfacial region are treated after the separate updates. Since more than one phase may be present in the computational domain, a cell can contain multiple field values at each time step. Therefore, the DG field values $\mathbf{U}_\chi^{(I)}$ of different phases should be treated to ensure consistency after each time integration step on the grid. In this study, a post-mixing rule is proposed for weakly compressible multiphase flows. As the name suggests, the mixing procedure is applied after the field values of each fluid phase have been updated separately.



(a) $t = 8.050$ (b) $t = 9.175$ (c) $t = 10.250$

Fig. 7. Oscillating droplet: droplet deformations at different times simulated using resolution $\Delta\bar{x} = 1/80$, where contour indicates pressure and red dashed lines represent the theoretical shapes derived using the theory in [82]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Semi-axis length (b) Resolution effect

Fig. 8. Oscillating droplet: (a) semi-axis length in x direction versus time with $\Delta\bar{x} = 1/80$; (b) semi-axis length recorded using various resolutions. The theoretical solutions are referred to [82].

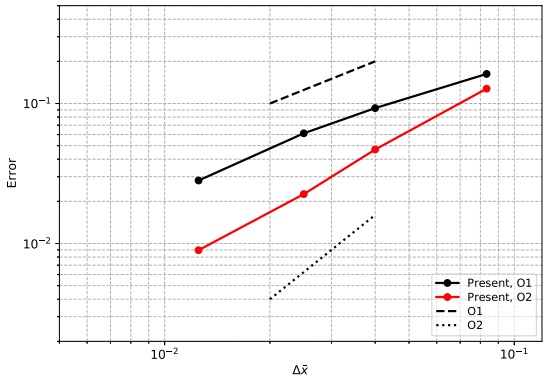


Fig. 9. Oscillating droplet: L_2 error versus simulation resolution $\Delta\bar{x}$ for two different orders of the present method.

4.2.1. *Mixing of* $\mathbf{U}_\chi^{(0)}$

The zeroth-order values of DG field value $\mathbf{U}_\chi^{(0)}$, i.e., density and momentum, play a key role in the fluid field update, and they should be treated with care. The post-mixing rule is similar to the VOF approach, where the pressure and velocity fields should be shared in the mixing area. It is worth noting that the mixing is a "post" implementation after the field value update, and it thereby avoids excessive dissipation, which is a key drawback of the conventional VOF. The key idea of the post-mixing rule is that the pressure and velocity field values should be shared within one cell. Based on this idea, if we only consider two phases in the field,

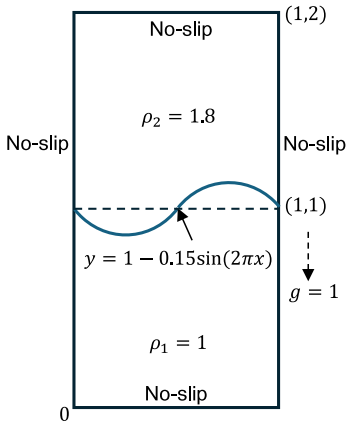
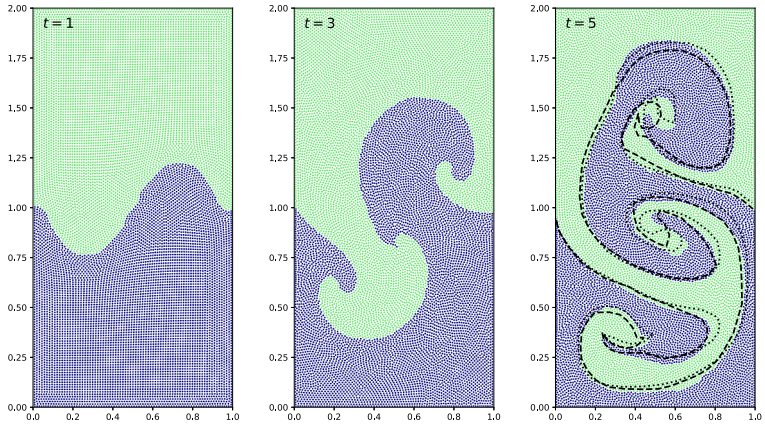


Fig. 10. Schematic of Rayleigh-Taylor instability and dimensions.



(a) $t = 1$ (b) $t = 3$ (c) $t = 5$

Fig. 11. Rayleigh-Taylor instability: snapshots at (a) $t = 1$, (b) $t = 3$ and (c) $t = 5$ simulated using resolution $\Delta\bar{x} = 1/80$, where at $t = 5$, the results simulated using level-set (dotted line) and SPH (dashed line) methods are referred for comparison [80].

the modified density $\hat{\rho}_\chi$ of phase χ after mixing should satisfy

$$p = c_1^2(\hat{\rho}_1^{(0)} - \rho_{0,1}) = c_2^2(\hat{\rho}_2^{(0)} - \rho_{0,2}) = \bar{c}^2(\bar{\rho}^{(0)} - \bar{\rho}_0), \tag{50}$$

to share one pressure field value, where

$$\begin{aligned} \bar{\rho}^{(0)} &= \alpha_1\rho_1^{(0)} + \alpha_2\rho_2^{(0)}, \\ \bar{\rho}_0 &= \alpha_1\rho_{0,1} + \alpha_2\rho_{0,2}. \end{aligned} \tag{51}$$

Note that mass conservation should be ensured after the mixing. Then we have

$$\alpha_1\hat{\rho}_1^{(0)} + \alpha_2\hat{\rho}_2^{(0)} = \bar{\rho}^{(0)}. \tag{52}$$

Then the mixed sound speed $\bar{c}$ satisfies

$$\frac{1}{\bar{c}^2} = \frac{\alpha_1}{c_1^2} + \frac{\alpha_2}{c_2^2}. \tag{53}$$

Based on the equations above, the pressure in the mixing area can be derived.

Likewise, the averaged momentum $\overline{\rho\boldsymbol{v}}^{(0)}$ and velocity $\bar{\boldsymbol{v}}^{(0)}$ can be estimated using phase fraction as

$$\overline{\rho\boldsymbol{v}}^{(0)} = \alpha_1(\rho\boldsymbol{v})_1^{(0)} + \alpha_2(\rho\boldsymbol{v})_2^{(0)}, \quad \bar{\boldsymbol{v}}^{(0)} = \overline{\rho\boldsymbol{v}}^{(0)}/\bar{\rho}^{(0)}, \tag{54}$$

and based on the average values, the treated momentum $\widehat{(\rho\boldsymbol{v})}_\chi$ of phase χ after the mixing can be constructed as

$$\widehat{(\rho\boldsymbol{v})}_1^{(0)} = \hat{\rho}_1^{(0)}\bar{\boldsymbol{v}}^{(0)}, \quad \widehat{(\rho\boldsymbol{v})}_2^{(0)} = \hat{\rho}_2^{(0)}\bar{\boldsymbol{v}}^{(0)}. \tag{55}$$

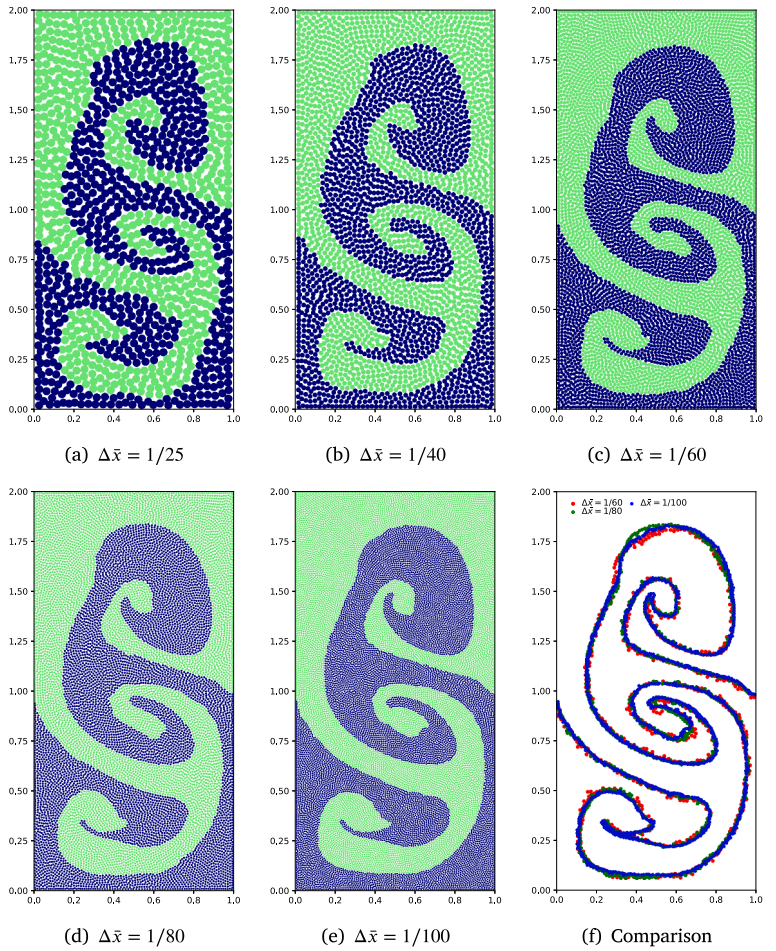


Fig. 12. Rayleigh-Taylor instability: snapshots at $t = 5$ simulated (a)~(e) using various resolutions, and (f) comparison among three results.

It is easy to show that the averaged values of conservative field values remain the same in the mixing area, i.e.,

$$\begin{aligned}
&\alpha_1(\rho\boldsymbol{v})_1^{(0)} + \alpha_2(\rho\boldsymbol{v})_2^{(0)} = \alpha_1\widehat{(\rho\boldsymbol{v})}_1^{(0)} + \alpha_2\widehat{(\rho\boldsymbol{v})}_2^{(0)},\\
&\alpha_1\rho_1^{(0)} + \alpha_2\rho_2^{(0)} = \alpha_1\hat{\rho}_1^{(0)} + \alpha_2\hat{\rho}_2^{(0)}.
\end{aligned} \tag{56}$$

4.2.2. *Mixing of* $\mathbf{U}_\chi^{(I)}, I > 0$

For the high-order terms of $\mathbf{U}_\chi^{(I)}$, the corresponding averaged field values can be estimated using phase fraction as

$$\begin{aligned}
&\bar{\rho}^{(I)} = \alpha_1\rho_1^{(I)} + \alpha_2\rho_2^{(I)}, \quad I = 1, 2, 3,\\
&\overline{\rho\boldsymbol{v}}^{(I)} = \alpha_1(\rho\boldsymbol{v})_1^{(I)} + \alpha_2(\rho\boldsymbol{v})_2^{(I)}, \quad I = 1, 2, 3,
\end{aligned} \tag{57}$$

Considering that the pressure in a cell is shared by different materials, for an arbitrary position $d\boldsymbol{x}$ in the cell, we should have

$$c_1^2\left(\hat{\rho}_1^{(I)}dx_I + \hat{\rho}_1^{(0)} - \rho_{0,1}\right) = c_2^2\left(\hat{\rho}_2^{(I)}dx_I + \hat{\rho}_2^{(0)} - \rho_{0,2}\right) = \bar{c}^2\left(\bar{\rho}^{(I)}dx_I + \bar{\rho}^{(0)} - \bar{\rho}_0\right), \text{ sum over } I. \tag{58}$$

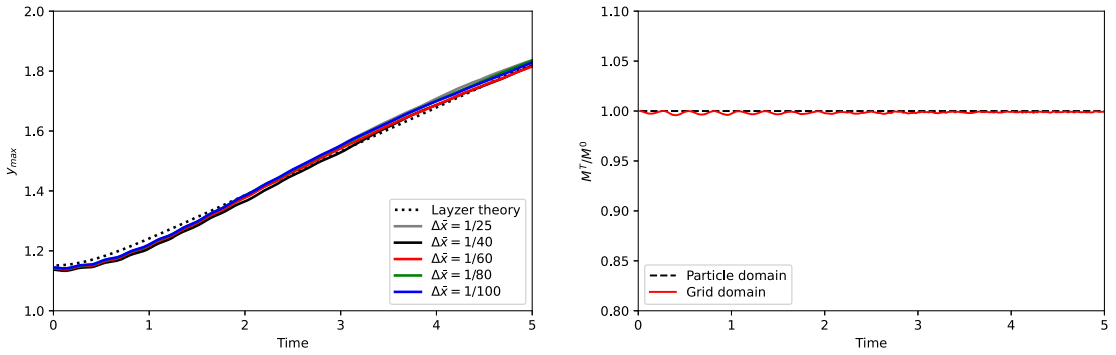
Noting Eq. (50), the above leads to

$$c_1^2\hat{\rho}_1^{(I)}dx_I = c_2^2\hat{\rho}_2^{(I)}dx_I = \bar{c}^2\bar{\rho}^{(I)}dx_I. \tag{59}$$

Considering that dx_I is arbitrary, we may conclude that

$$c_1^2\hat{\rho}_1^{(I)} = c_2^2\hat{\rho}_2^{(I)} = \bar{c}^2\bar{\rho}^{(I)}, \tag{60}$$

and then the treated $\hat{\rho}_\chi^{(I)}$ is derived for phase χ, i.e., $\hat{\rho}_\chi^{(I)} = \bar{c}^2\bar{\rho}^{(I)}/c_\chi^2$.



(a) Maximum height $y_{\max}$ (b) Mass variation

Fig. 13. Rayleigh-Taylor instability: (a) maximum height $y_{\max}$ of the light fluid versus time, simulated using various resolutions and comparison to Layzer theory [82,86]; (b) the total mass M^T relative to the initial mass M^0 of the light phase in the particle and grid domains, simulated using $\Delta\bar{x} = 1/100$.

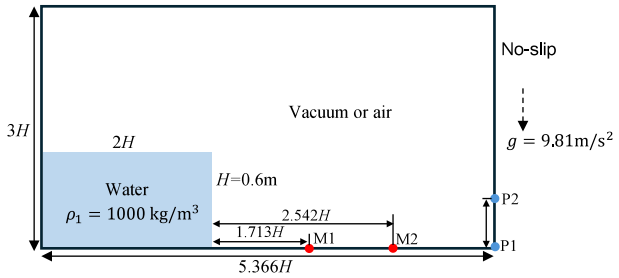


Fig. 14. Schematic of dry dam-break and dimensions.

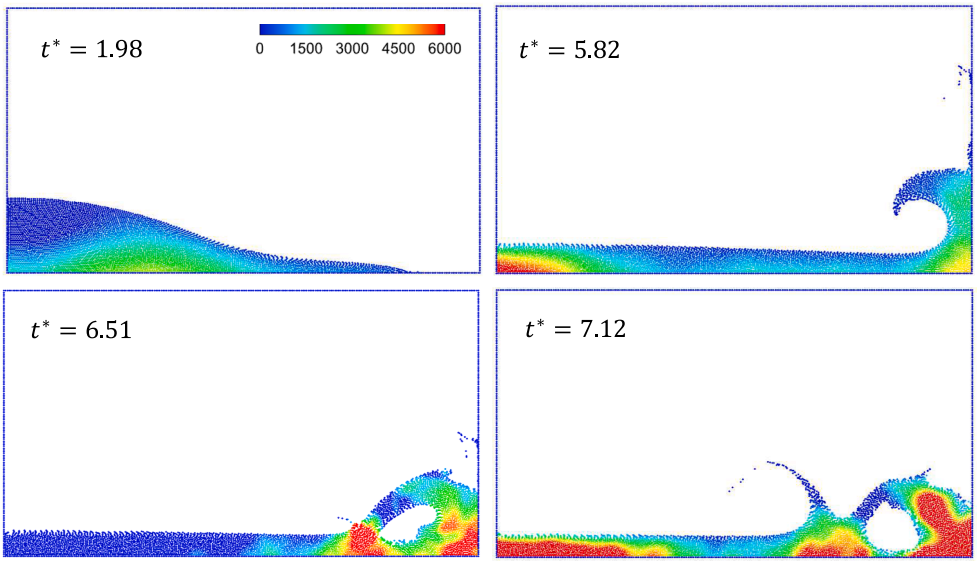


Fig. 15. Dry dam-break (single phase): snapshots simulated with resolution $\Delta\bar{x} = H/50$, where the pressure field is indicated with contour.

As for the momentum, the averaged momentum should be $\overline{\rho\boldsymbol{v}}^{(0)} + \overline{\rho\boldsymbol{v}}^{(I)} dx_I$ at an arbitrary position $d\boldsymbol{x}$ in the cell. For $\overline{\rho\boldsymbol{v}}^{(I)}$, based on the assumption that the velocity is shared within the grid cell and $\frac{\partial \rho\boldsymbol{v}}{\partial x_I} = \rho\frac{\partial \boldsymbol{v}}{\partial x_I} + \boldsymbol{v}\frac{\partial \rho}{\partial x_I}$ in the I direction, we have

$$\overline{\rho\boldsymbol{v}}^{(I)} = \bar{\rho}^{(0)}\bar{\boldsymbol{v}}^{(I)} + \bar{\rho}^{(I)}\bar{\boldsymbol{v}}^{(0)}. \tag{61}$$

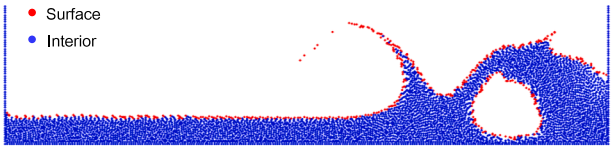
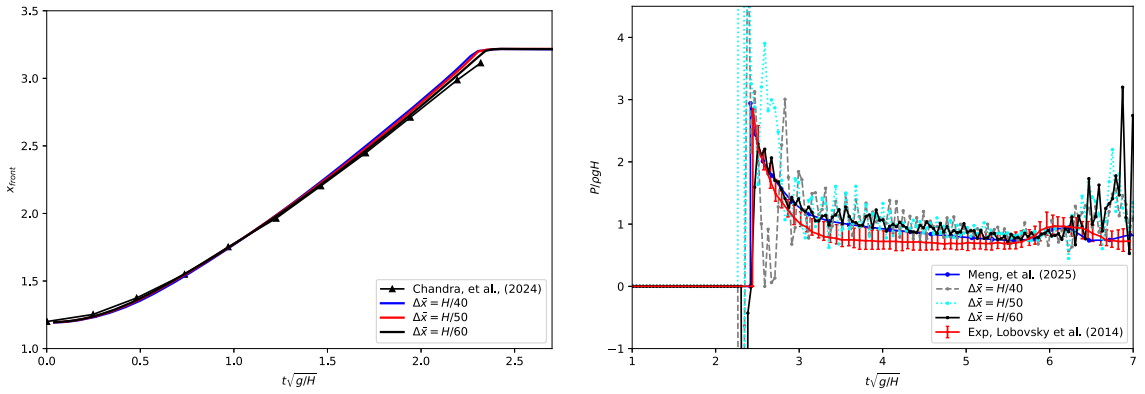


Fig. 16. Dry dam-break (single phase): detected surface and interior particles marked in red and blue, respectively, at $t^* = 7.12$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Water front (b) Pressure at P1

Fig. 17. Dry dam-break (single phase): (a) water front and (b) pressure measured at P1 versus t^*.

Then it is easy to have

$$\bar{\boldsymbol{v}}^{(I)} = \left(\overline{\rho \boldsymbol{v}}^{(I)} - \bar{\rho}^{(I)} \bar{\boldsymbol{v}}^{(0)}\right) / \bar{\rho}^{(0)}. \tag{62}$$

Given $\bar{\boldsymbol{v}}^{(I)}$ and $\rho_\chi^{(I)}$, the high-order DG field values $\rho \boldsymbol{v}^{(I)}$ can be reconstructed as

$$(\widehat{\rho \boldsymbol{v}})_1^{(I)} = \hat{\rho}_1^{(0)} \bar{\boldsymbol{v}}^{(I)} + \hat{\rho}_1^{(I)} \bar{\boldsymbol{v}}^{(0)}, \quad (\widehat{\rho \boldsymbol{v}})_2^{(I)} = \hat{\rho}_2^{(0)} \bar{\boldsymbol{v}}^{(I)} + \hat{\rho}_2^{(I)} \bar{\boldsymbol{v}}^{(0)}. \tag{63}$$

After the treatments of $\mathbf{U}_\chi^{(I)}$, two phases share the same velocity and pressure in the test function space.

Remark 5. The so-called uniform-pressure–velocity (UPV) criterion [60,81] states that uniform pressure and velocity fields in the interface region should be maintained in time integrations. The criterion can be written as

$$\text{if } p_i^n = p_0 \text{ and } \boldsymbol{v}_i^n = \boldsymbol{v}_0, \text{ then } p_i^{n+1} = p_0 \text{ and } \boldsymbol{v}_i^{n+1} = \boldsymbol{v}_0. \tag{64}$$

This is an important criterion for preventing artificial oscillations across the phase interface. Given uniform pressure and velocity fields, it follows from Eqs. (22) and (48) that the flux terms on the right-hand side of Eq. (22) cancel themselves, and these zero values are preserved after the post-mixing treatment. This demonstrates that the proposed post-mixing treatment in PG-DG inherently satisfies the UPV property, since the two phases are treated separately.

5. Summary of solution procedure

To clearly present the solution procedure of our proposed PG-DG method, the implementation of a single sub-time step is summarized as follows:

(1) Determine the phase fraction $\alpha_{i,\chi}$ for different phases [Eq. (14)].
(2) Determine the *phase status* $PS_{i,\chi}$ and *fluid status* FS_i based on the phase fraction [Eqs. (19) and (20)].
(3) P2G: Map the DG field values carried by particles to the grid [Eq. (21)] according to the $PS_{i,\chi}$, only in the first sub-time step.
(4) G2P: Map the DG field values on the grid back to particles [Eq. (32)], only in the first sub-time step.
(5) Compute the limiter $\psi_{i,\chi}$ [Eq. (28)] and the flux $\hat{\mathbf{F}}_{ijt,\chi}$ on the grid [Eqs. (23, 47, 48)].
(6) Update the grid field values using Runge–Kutta time integration [Eq. (29)].
(7) Apply the post-mixing rule to the DG field values $\mathbf{U}_{i,\chi}^{(I)}$ of different phases [Eqs. (50 ∼ 63)].
(8) Update the particle field values [Eqs. (33, 34)], including the particle redistribution treatment associated with $\boldsymbol{w}_p$ [Eq. (45)].
The procedures above are repeated in a loop every time step until the final time is reached.

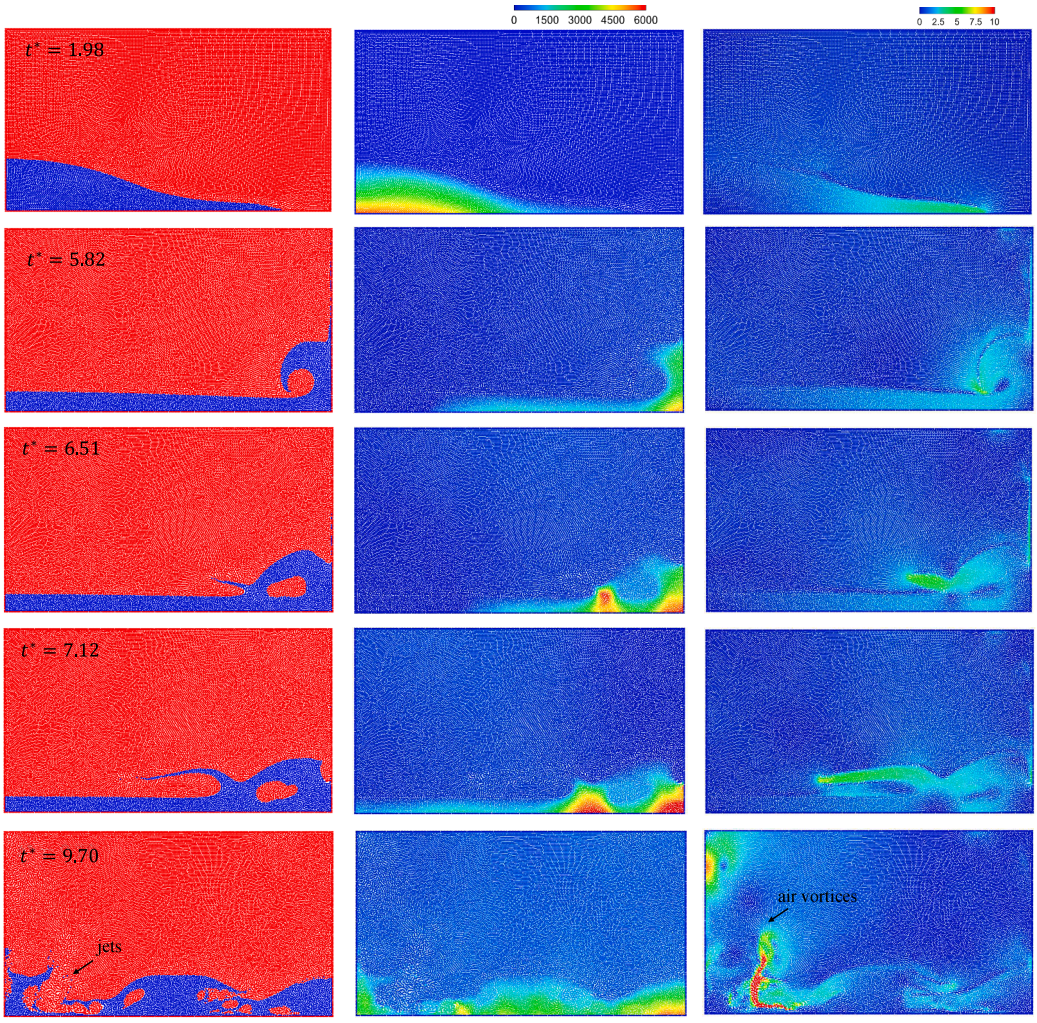


Fig. 18. Dry dam-break (multiphase): snapshots simulated using $\Delta\bar{x} = H/40$ at different times, where phase (left), pressure (middle) and velocity (right) are displayed.

6. Numerical examples

6.1. Convergence order tests

6.1.1. Taylor-Green vortex

The Taylor–Green vortex is a standard benchmark case for assessing the accuracy of numerical methods. Firstly, we check the accuracy of the PG-DG scheme without any free surfaces or multiphase interfaces, and under this circumstance, PG-DG actually reduces to DG. In a square with edge $L = 1$, the initial velocity field is $v_{0x} = -U\cos(2\pi x)\sin(2\pi y)$ and $v_{0y} = U\sin(2\pi x)\cos(2\pi y)$. Two Reynolds numbers $Re = \rho U L/\mu$ are chosen in this test, i.e., 100 and 1000. For simplicity, we select $U = 1$ and $\rho = 1$. All boundaries of the square are set as periodic. Then the analytical solution of the velocity fields is

$$\begin{aligned} v_x &= -U e^{\zeta t}\cos(2\pi x)\sin(2\pi y), \\ v_y &= U e^{\zeta t}\sin(2\pi x)\cos(2\pi y), \end{aligned} \tag{65}$$

with $\zeta = -8\pi^2/Re$. The case is simulated until $t = 1$, and then the results are compared to the analytical solution to calculate the L_2 error. Fig. 5 shows the L_2 error versus simulation resolution $\Delta\bar{x}$. It shows that the two curves are almost parallel to the quadratic convergence line, which means that the present PG-DG scheme achieves the expected order.

6.1.2. Oscillating droplet

We evaluate the oscillating droplet problem from [82], which introduces the problem to test the accuracy of a method for simulating free-surface flow, with a corresponding analytical solution being derived. In this case, a droplet with radius $R = 1$ is placed

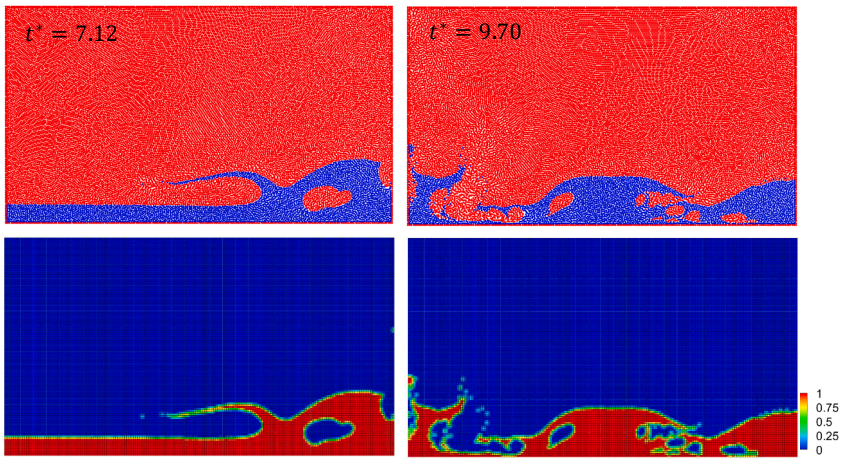


Fig. 19. Dry dam-break (multiphase): the first row indicates the particle phase fields, and their corresponding phase fraction fields of water at grid cells are indicated in the second row, where the results are from Fig. 18.

in the center of the domain, as shown in Fig. 6. The droplet has an initial velocity of $v_{0x} = \sigma_0 x$ and $v_{0y} = -\sigma_0 y$, and is subject to a conservative force field with the acceleration given as $\boldsymbol{f} = -\Omega^2 \boldsymbol{x}$ with $\boldsymbol{x} = (x, y)$. In this case, we select $\sigma_0 = 0.5$ and $\Omega = 1$. The artificial sound speed is chosen as $c = 8$.

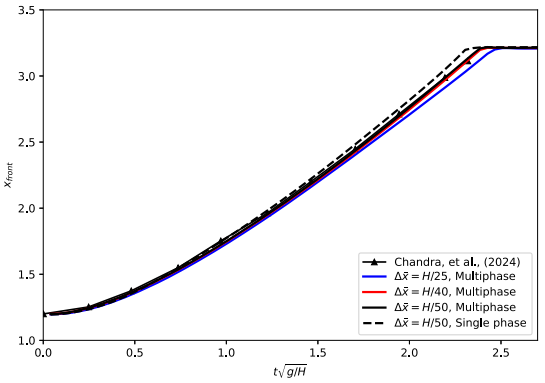
Fig. 7 showcases the droplet deformations at three typical times in the third oscillation period, and this case is simulated using resolution $\Delta\bar{x} = 1/80$. The figure illustrates that the pressure field is quite smooth, and the deformation shapes at three times agree very well with the theoretical shapes derived using the partial differential equations given in [82]. A clearer plot of the semi-axis length A versus time is presented in Fig. 8(a), where the present results are compared with the theoretical solution. Good agreement is observed between the present results and the theory. Fig. 8(b) shows that, as the resolution is increased, the present method gradually approaches the reference solution.

To examine the convergence rate of the present method in simulating free-surface flows, the L_2 error of A is plotted against the resolution $\Delta\bar{x}$, as shown in Fig. 9. Two versions of the present method are employed in the simulations. Here, "present O2" denotes the use of the test functions given in Eq. (8), whereas "present O1" denotes the degraded first-order version, which is equivalent to a conventional first-order finite-volume method. The results show that the convergence order of present O2 is higher than first-order but lower than second-order, which may be attributed to the free-surface treatment adopted in the present method. In the present method, grid cell interfaces are used to represent the phase interface, which is less smooth than the actual free surface. By contrast, conventional approaches typically employ piecewise linear interface calculation (PLIC) [83] to reconstruct the interface with linear segments, thereby achieving second-order accuracy. This explains why the present method exhibits an order of accuracy lower than second-order in free-surface simulations. Although the present method cannot achieve second-order accuracy near the free surface, it is straightforward to implement, and its convergence order remains higher than first-order. The multiphase flow examples presented below further show that the present method remains quite accurate even at very low resolutions when compared with widely used meshless methods.

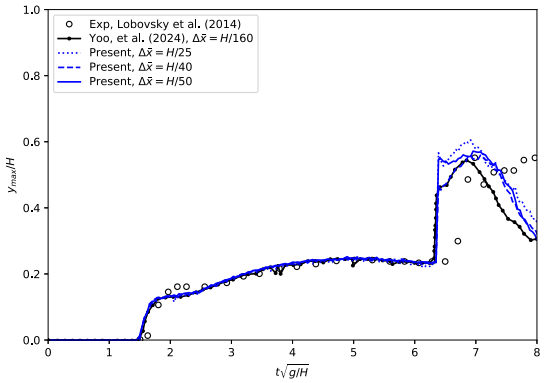
6.2. Rayleigh–Taylor instability

The Rayleigh–Taylor instability is a typical case to validate a method for multiphase flow. This case requires a proper interface treatment between different phases. As shown in Fig. 10, the computational domain is $[L \times 2L]$, with the heavy and light phases placed on the top and bottom, respectively, where $L = 1$ is chosen in this study. The two phase domains are separated with a curve $y = 1 - 0.15\sin(2\pi x)$ and are subject to a gravitational acceleration of $g = 1$ pointing vertically downwards, and the densities of light and heavy fluids are $\rho_1 = 1$ and $\rho_2 = 1.8$, respectively. The Reynolds number is chosen as $Re = \sqrt{L^3 g}/\nu = 420$, where ν is kinematic viscosity, which is the same for the two fluids. In this case, the walls are set as having no-slip boundary conditions, and the artificial sound speeds are chosen as 8 and 12 for the heavy and light fluid, respectively. For the resolution of $\Delta\bar{x} = 1/80$, the computational cost is shown in Table 1.

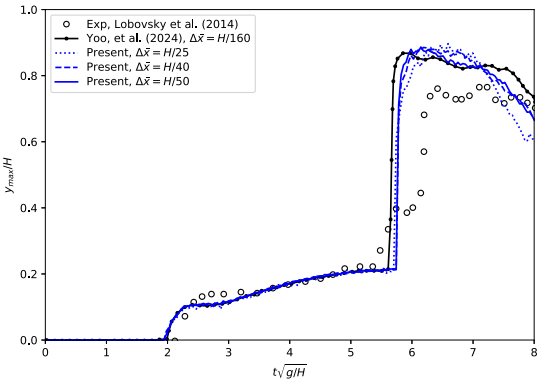
The water-front trajectories obtained at various resolutions are plotted against t^* in Fig. 17(a). It can be seen that the curves almost collapse onto one another and agree well with the reference results reported in [88], indicating that a relatively low resolution, $\Delta\bar{x} = H/40$, is already sufficient to achieve accurate simulation results. Similarly, the pressure history measured at P1 is shown in Fig. 17(b). It can be clearly observed that the present results agree well with the experimental data reported in [87] as the resolution increases, although some temporal oscillations are present. In fact, such pressure oscillations in time are expected. They are a typical feature of simulations based on a weakly compressible EOS, i.e., Eq. (6). In reality, water is nearly incompressible, whereas a much lower artificial sound speed is adopted in the numerical model, which leads to stronger pressure oscillations, particularly at the impact instant near $t^* = 2.4$. This phenomenon has been widely reported in many previous studies, such as [26,89]. It was also reported in



(a) Water front

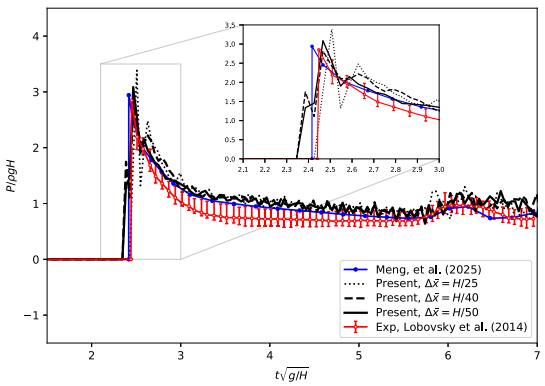


(b) Water Height at M1

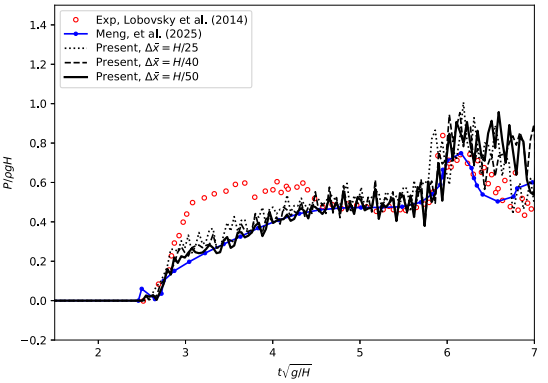


(c) Water Height at M2

Fig. 20. Dry dam-break (multiphase): (a) water front, water height at (b) M1 and (c) M2 versus t^* simulated using different resolutions, and comparison to other references.



(a) Pressure at P1

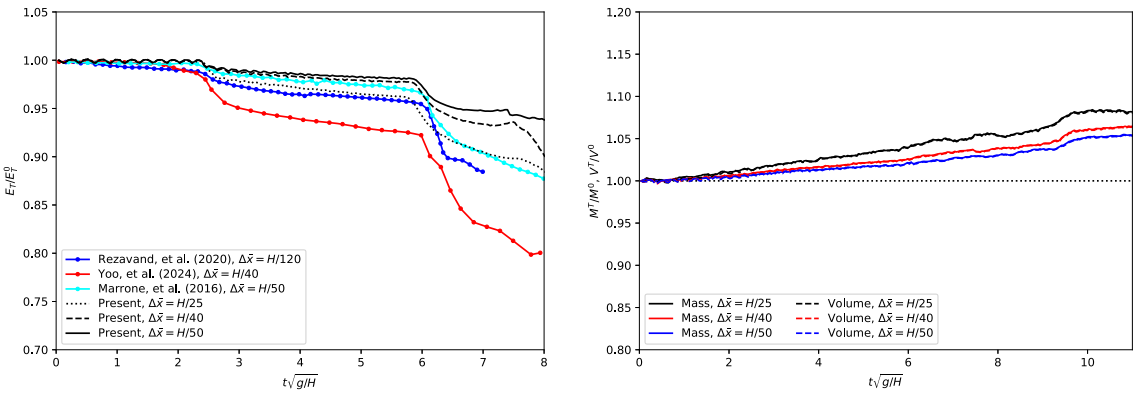


(b) Pressure at P2

Fig. 21. Dry dam-break (multiphase): measured pressure at (a) P1 and (b) P2 versus t^* simulated with different resolutions, and comparison to other references.

[90] that a so-called acoustic damper can be used to alleviate the pressure oscillations associated with the weakly compressible EOS. In addition, some studies model water as a fully incompressible fluid and solve a pressure Poisson equation, in which case no obvious oscillations are observed, as in [88,91]. This is beyond the scope of the present study.

The flow snapshots at different times obtained with a resolution of $\Delta\bar{x} = 1/80$ are shown in Fig. 11. The results show the development of the characteristic mushroom-shaped structure as the heavier fluid descends, and the interface between the two fluids is



(a) Total energy dissipation (b) Mass and volume variation

Fig. 22. Dry dam-break (multiphase): (a) the total energy dissipation versus time, simulated using various resolutions; (b) the total mass and volume variations M^T and V^T relative to their initial values M^0 and V^0 of the water phase in the grid domain, and the dotted line represents the particle mass and volume variation.

Table 1
Computational cost of Rayleigh–Taylor instability at $\Delta\bar{x} = 1/80$, where wall time is that required to reach a simulation time of 5. The computation is performed on two AMD EPYC 75F3 32-core processors.

Particle count	Cell count	Time step	Wall time (s)
12,800	12,800	0.00018	917

clearly captured. The present simulation results are compared with the reference results reported in [80], which were obtained using both mesh-based level-set and meshless SPH methods. Reasonable agreement with the reference results is observed at $t = 5$. Different resolutions are used to simulate this case, and the flow snapshots at $t = 5$ are displayed in Fig. 12. It shows that even at very low resolutions, i.e., $\Delta\bar{x} = 1/25$ and $1/40$, the present method can provide reliable results close to the convergent one. The interfaces of multiple results are put together for further comparison in Fig. 12(f), and it shows that the multiple interfaces almost overlap among the different resolutions. In [84] and [85], where SPH is employed for simulation, the results by resolution $\Delta\bar{x} = 1/50$ are still far away from the reference; in contrast, our result by $\Delta\bar{x} = 1/25$ is almost close to the convergent solution. For a more quantitative comparison, Fig. 13(a) presents the time histories of the maximum height $y_{\max}$ of the light fluid obtained at various resolutions, showing that the results at all resolutions are consistent with one another. The present results are also compared with the reference solution reported in [82], which was derived from a so-called toy model based on Layzer theory [86], and a good agreement is observed. Fig. 13(b) presents the total mass M^T relative to the initial mass M^0 of the light phase in the particle and grid domains. It shows that, while mass conservation is strictly ensured in the particle domain, it is not in the grid domain. In fact, strict mass conservation in the grid domain is not necessary, since the grid domain is used to solve the mass and momentum equations, and particles carry all persistent states. This case shows that the present method can provide a reliable result even with a very low resolution, which is hard to achieve using other meshless methods such as SPH.

6.3. Dry dam-break

The dam-break case is a typical benchmark case to inspect the performance of a numerical method for simulating flows with complex interfaces. This case has both experimental measurements given in [87] and a variety of numerical references, for which a typical setup is shown in Fig. 14. A water column with height $H = 0.6$ m and length $2H$ is placed at the left of the water tank with the length and height being 3.2196 m and 1.8 m, respectively. For water, the density and dynamic viscosity are selected as 1000 $\mathrm{kg/m^3}$ and 1×10^{-3} Pa · s, respectively, and for air, we select 1.29 $\mathrm{kg/m^3}$ and 1.8×10^{-5} Pa · s. The local gravitational acceleration is chosen as $g = 9.81$ $\mathrm{m/s^2}$. The artificial sound speeds for the water and air are set as $10\sqrt{2gH}$ and 150 m/s, respectively. The walls are assigned no-slip boundary conditions in this case. Two pressure probe points (as shown in Fig. 14) are selected for measurement. Using real time t, the non-dimensionalized time is defined as $t^* = t\sqrt{g/H}$.

6.3.1. Single-phase case

First, only the water phase is considered, while the air phase is treated as vacuum, which is a typical treatment in many meshless methods. The present free-surface formulation is employed for this scenario. Fig. 15 shows the simulation results obtained with a resolution of $\Delta\bar{x} = H/50$, where the pressure field is presented in contours. The present results show that the pressure field remains quite smooth, which is an advantage over the conventional SPH method, in which pressure noise often appears in space during violent

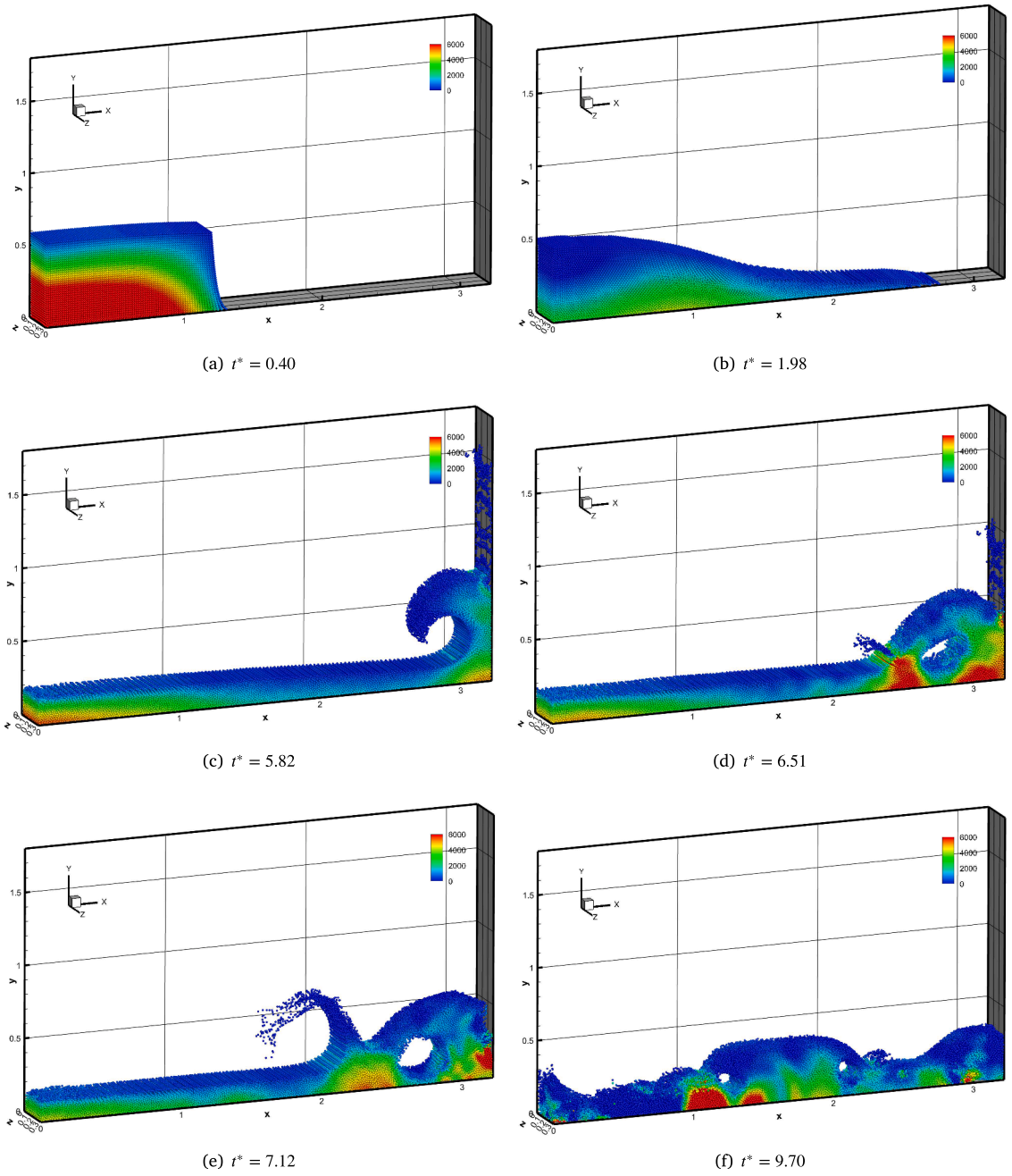


Fig. 23. Dry dam-break (3D single-phase): 3D flow snapshots simulated with resolution $\Delta\bar{x} = H/40$ at different times, where contour indicates pressure.

free-surface flows. In addition, the present method can accurately capture the free-surface particles, as shown in Fig. 16, where the surface particles are marked in red at $t^* = 7.12$.

6.3.2. Multiphase case

Numerous studies have shown that neglecting the air phase leads to inaccurate simulations in the late stage of dam-break flow, because the air-cushion effect becomes non-negligible at that stage. In light of this, multiphase dam-break flow—including the air phase—is considered. Compared with the previous free-surface case, the only difference is that the vacuum phase is replaced by air. In this case, the CFL number is set to 0.15 in view of the large density ratio. For the resolution of $\Delta\bar{x} = H/40$, the computational cost is shown in Table 2.

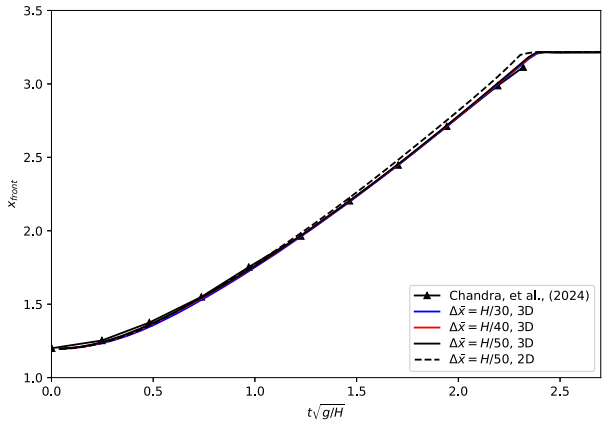


Fig. 24. Dry dam-break (3D single phase): flow front versus time using different resolutions.

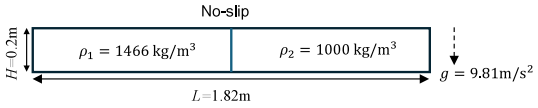


Fig. 25. Schematic of Non-Boussinesq gravity currents and dimensions.

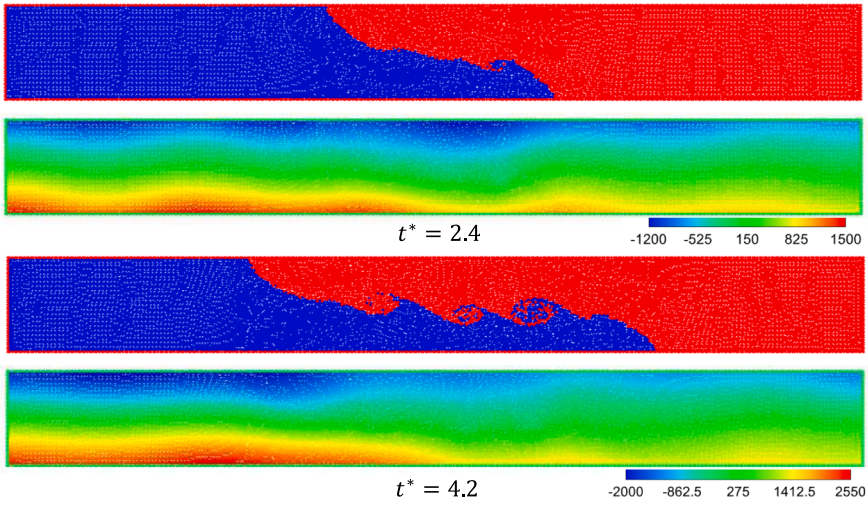


Fig. 26. Non-Boussinesq gravity currents: snapshots simulated using resolution $\Delta\bar{x} = H/40$ showing phase and pressure fields at different times.

Table 2
Computational cost of dam-break at $\Delta\bar{x} = H/40$, where wall time is that required to reach a simulation time of 2.8 s. The computation is performed on two AMD EPYC 75F3 32-core processors.

Particle count	Cell count	Time step (s)	Wall time (s)
25,347	25,466	1.41×10^{-5}	14,475

The simulation results obtained with a resolution of $\Delta\bar{x} = H/40$ at different times are shown in Fig. 18, where the phase distribution (left), pressure field (middle), and velocity field (right) are presented. Before approximately $t^* = 6$, the water motion appears nearly identical to that in the free-surface case. After that, the water front plunges into the water, trapping air as shown at $t^* = 6.51$. The trapped air then begins to exert a cushioning effect, and the air pressure rises slightly. By contrast, in the free-surface case, the air is assumed to remain at a constant ambient pressure; consequently, the free-surface solution starts to deviate from the physical one. A similar behavior is observed at $t^* = 7.12$. At a later stage, e.g., at $t^* = 9.70$, the flow becomes highly violent and breaks into many jets and small fragments in the air, which strongly disturb the surrounding air and generate vortical structures over a wide range of

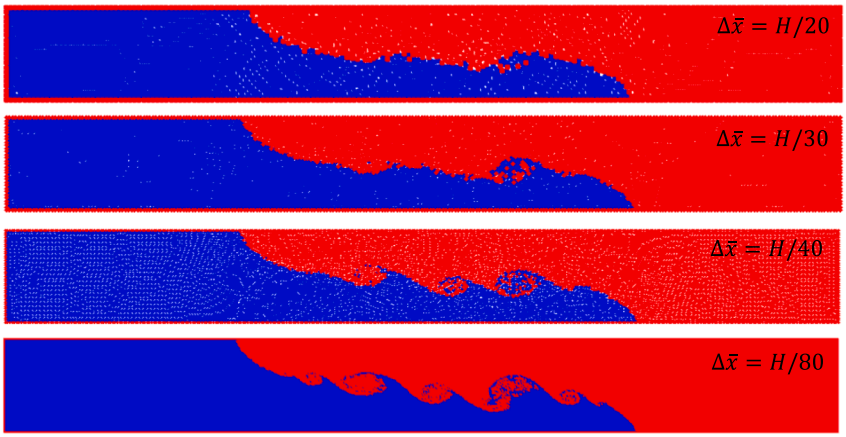
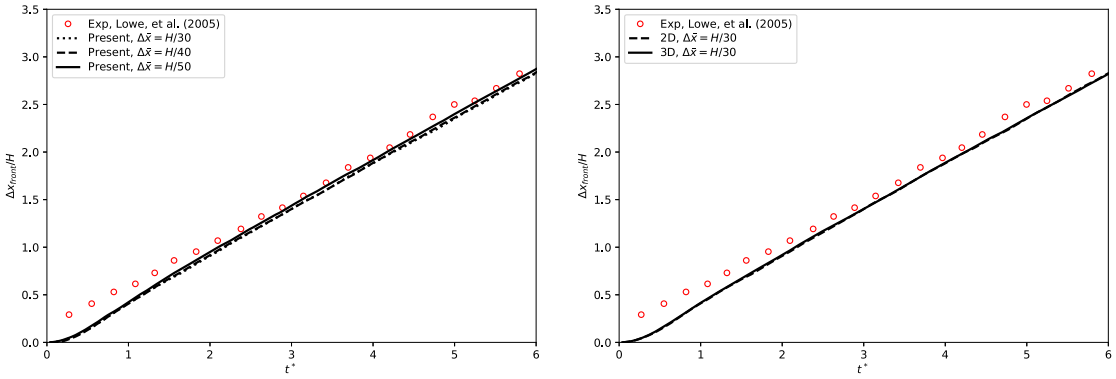


Fig. 27. Non-Boussinesq gravity currents: flow profiles at $t^* = 4.2$ simulated using various resolutions.



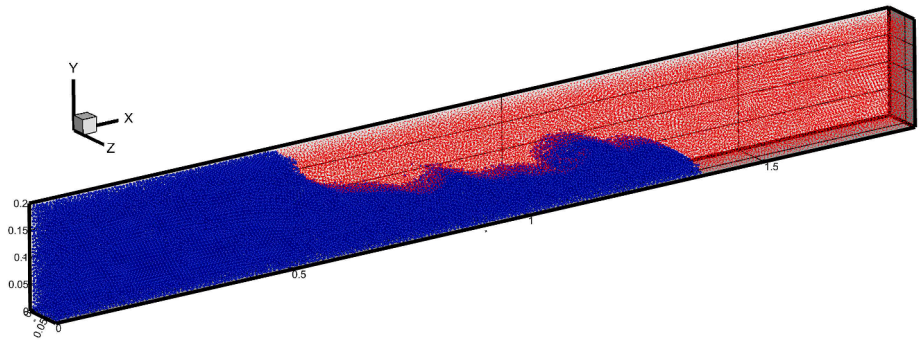
(a) Different resolutions (b) 2D and 3D

Fig. 28. Non-Boussinesq gravity currents: (a) displacement of the light fluid front versus t^*, where multiple resolutions are used and compared to the experimental data from [101]; (b) comparison between 2D and 3D results using resolution $\Delta\bar{x} = H/30$.

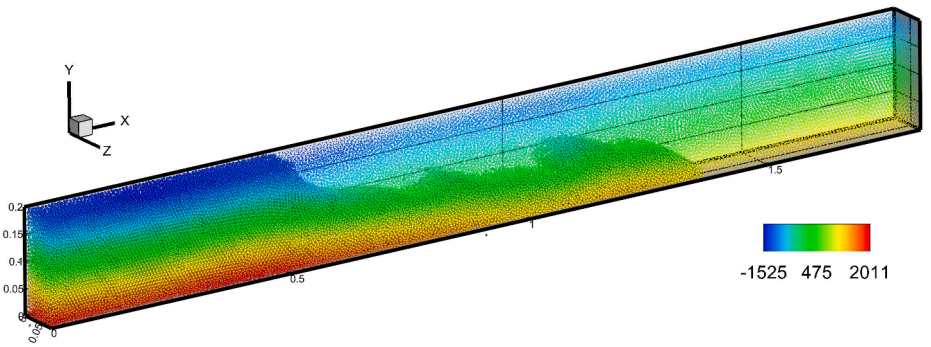
scales. Fig. 19 indicates the corresponding phase fraction fields α_{water} at times $t^* = 7.12$ and 9.70 in the second row. The phase fraction field α_{water} shows consistency with the particle distribution given in the first row. Although the phase fraction is diffuse in the grid domain, the interface width does not grow during the simulation because α_{water} is calculated directly using particle distribution, as remarked in Section 3.1.1. The phase fraction is only used to mix the field values updated in different phases, and in the separate update, it is a sharp-interface treatment, which does not require handling discontinuity at the interface, and this avoids possible excessive numerical dissipation. Thanks to the no-mass-dissipation property of the present method, the small jets can be captured well. By contrast, methods based on the conventional phase advection equation, such as that in [92], smear out most of the violent jets in dam-break flow. A similar phenomenon is also observed in [60], where the flow interface appears to broaden as the simulation progresses. Some other studies employ interface-sharpening techniques [93]; however, such sharp-interface treatments are relatively complicated.

Furthermore, the water-front trajectories and water height in the multiphase dam-break case are shown in Fig. 20. Fig. 20(a) presents the water-front trajectories obtained at various resolutions, and good agreement is observed with the reference results reported in [88]. It can also be clearly seen that the simulation is nearly converged at a resolution of $\Delta\bar{x} = H/40$. Compared with the single-phase case, the water front advances slightly more slowly because of the effect of the air phase. Fig. 20(b) and (c) show the water heights measured at two observation points, M1 and M2. The results indicate that the present simulations achieve convergence at $\Delta\bar{x} = H/40$ and agree well with the SPH results reported in [94], which were obtained using a much higher resolution of $\Delta\bar{x} = H/160$. By contrast, the present method, even with a relatively coarse resolution of $\Delta\bar{x} = H/25$, already provides an accurate prediction. The present simulation results also show reasonable agreement with the experimental data reported in [87], particularly in the early stage of the simulation. The deviation observed at later times is also reported in other numerical studies and may be attributed to the fact that the highly chaotic and violent flow in the experiment is sensitive to external factors, such as tank-wall roughness and geometric imperfections.

The pressure histories probed at two different locations, P1 and P2, are plotted in Fig. 21. At P1, the present results agree quite well with the experimental measurement, especially at the peak around $t^* = 2.45$ and the plunging instant around $t^* = 6$. At the pressure



(a) Phase



(b) Pressure

Fig. 29. Non-Boussinesq gravity currents: 3D (a) phase and (b) pressure snapshots at $t^* = 4.2$, simulated using resolution $\Delta\bar{x} = H/30$.

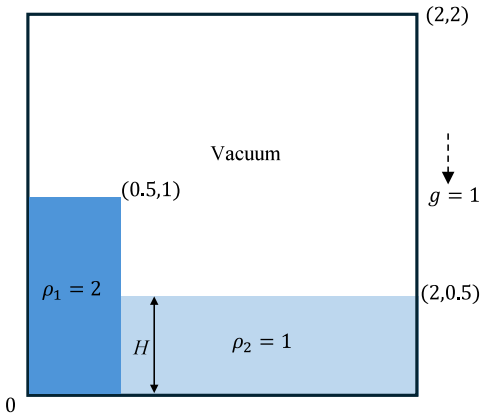


Fig. 30. Schematic of gravity currents of hybrid phases and dimensions.

peak, the present simulation with a relatively coarse resolution of $\Delta\bar{x} = H/25$ already yields a value very close to the experimental result, around $p/\rho g H = 3$. As the resolution is increased, the predicted peak pressure becomes even closer to the experimental value. Around $t^* = 6$, the slight pressure rise caused by the plunging water is accurately captured by the present method. This is noteworthy because many meshless methods tend to predict this event with a delay [26,75,94–96]. In addition, the present results agree very well with the high-resolution results obtained using a purely mesh-based method, in which a weighted essentially non-oscillatory (WENO) scheme is employed to reduce numerical dissipation near the interface [97]. Similarly, good agreement with [97] is also observed in Fig. 21(b). The pressure oscillations are caused by the weakly compressible EOS employed here, as explained in the single-phase case. Compared with the single-phase case, the oscillations are much weaker, which may be attributed to the air-cushion effect. They also become less pronounced as the grid resolution is increased. The reference results shown in this figure are taken from [97], in which a fully incompressible formulation is used. This explains why no pressure oscillations are observed in the reference solution. By contrast, in [61], a weakly compressible EOS is also employed in a purely mesh-based method with a high-order WENO scheme,

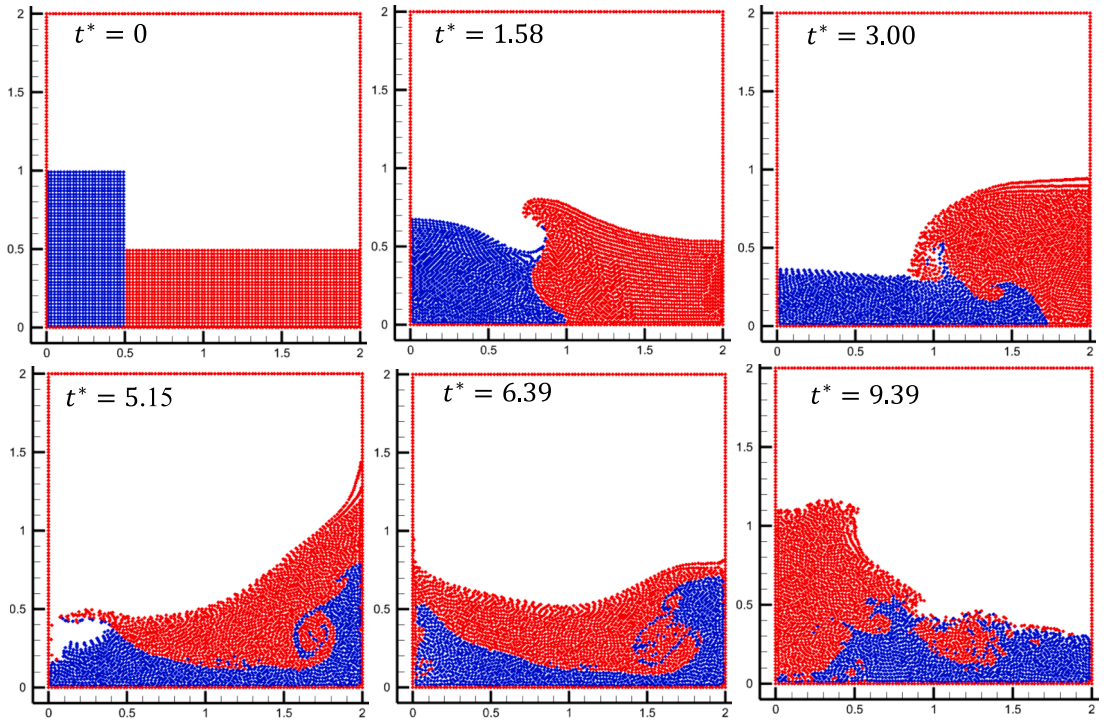


Fig. 31. Gravity currents: flow snapshots of phase field at different times simulated using resolution $\Delta\bar{x} = H/25$.

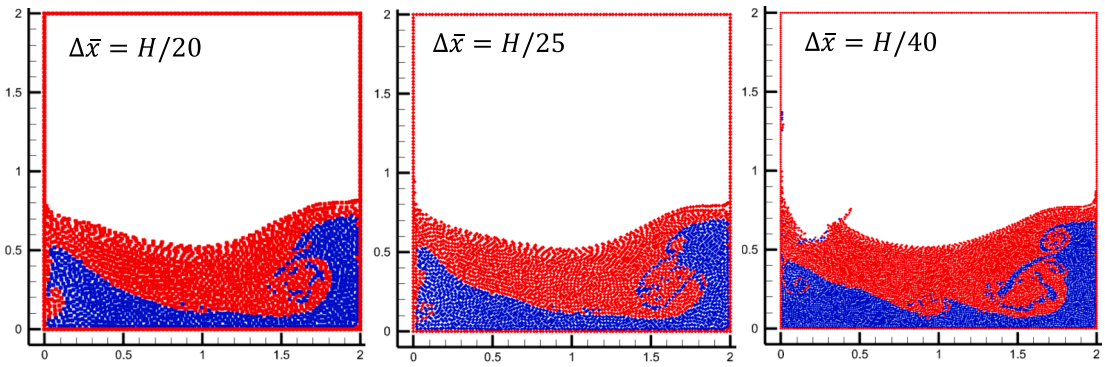


Fig. 32. Gravity currents: phase fields at $t^* = 5.15$ simulated using different resolutions.

and much stronger oscillations than those in the present results are observed. That issue was reported to be alleviated by replacing the conservative variables with primitive variables [62]. In contrast, although conservative variables are also used in the present method, the pressure oscillations in the present results do not appear to be pronounced.

Moreover, the variation of the total energy E_T over time is investigated by comparing it with the initial total energy E_T^0. To this end, the ratio E_T/E_T^0 is plotted in Fig. 22(a) for different resolutions. The present results are compared with the SPH results reported in [94,95,98]. It can be seen that even the coarsest resolution of the present method, $\Delta\bar{x} = H/25$, exhibits lower energy dissipation than that reported in [94] with $\Delta\bar{x} = H/40$ and in [95] with the much finer resolution $\Delta\bar{x} = H/120$. This comparison demonstrates that the present method introduces very low energy dissipation in multiphase flows with a large density ratio. The relative mass and volume variations M^T/M^0 and V^T/V^0 of the water phase in the grid domain are depicted in Fig. 22(b), where the particle mass variation is plotted using the dotted line. It shows that in the particle domain, mass conservation is ensured perfectly; however, the grid domain shows a slight increase, which implies that the conservation property is not met in the grid domain. One can also find that the mass increases entirely overlap with the volume increases, and this means that the grid mass increase is due to an expansion of the region occupied by the water particles. In fact, this is a common issue in simulating violent flows when applying PST to particle-based methods, since the PST cannot strictly ensure the volume consistency, which sometimes leads to volume expansion in violent flows, as reported in previous studies [77,99,100]. Because the occupied region is expanded, the volume of the grid domain will also expand correspondingly, and this leads to a slight mass increase in the weakly compressible flow. Employing an advanced version

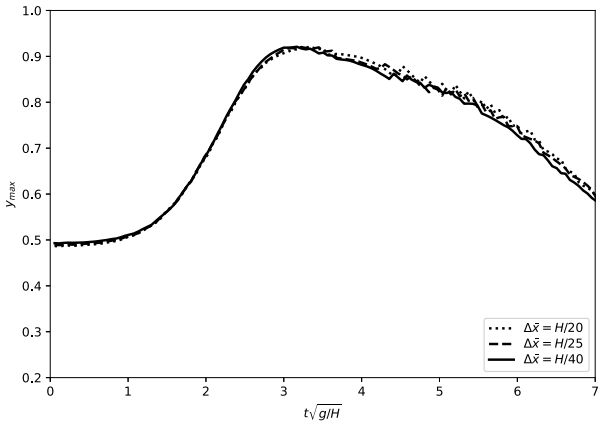


Fig. 33. Gravity currents: flow height measured at $x = 1.5$ simulated using various resolutions.

of the PST can solve this problem. Nonetheless, the present method focuses on particles, where particles carry all the necessary field information. Thus, as long as the mass does not reduce for the particles, they will be mapped back to the grid, which will increase the simulation accuracy in cases with violent flows.

Overall, the multiphase dam-break case demonstrates that, in addition to its no-mass-dissipation property in the particle domain, the present method achieves high accuracy for multiphase flows with a large density ratio, and reliable predictions can be obtained even at low resolutions. Owing to the high accuracy and low-dissipation properties of the PG-DG method, the present approach can achieve accuracy comparable to, or even better than, that of purely mesh-based methods, while accurately preserving mass in the particle domain.

6.3.3. 3D single-phase case

A 3D case is further considered to demonstrate the performance of the present method. All parameters and dimensions are the same as those shown in Fig. 14, except that an additional dimension is introduced such that the flow width in the z-direction is $0.5H$. The flow snapshots obtained with a resolution of $\Delta\bar{x} = H/40$ are shown in Fig. 23. The results show that the pressure field remains smooth, and the overall flow profiles are very similar to the 2D results shown in Fig. 15. The water fronts simulated using different resolutions are recorded in Fig. 24. It turns out that the results of different resolutions almost overlap, and the agreement is observed even at resolution $\Delta\bar{x} = H/30$. The present 3D results show very little difference from the 2D ones, and a good agreement is observed with the reference from [88].

6.4. Gravity currents

6.4.1. Non-Boussinesq gravity currents

The case of non-Boussinesq gravity currents was investigated experimentally in [101] and is adopted here as a numerical benchmark to assess the performance of the present method in simulating multiphase flows. The configuration of this case is shown in Fig. 25. Two fluids with densities $\rho_1 = 1466\ \mathrm{kg/m^3}$ and $\rho_2 = 1000\ \mathrm{kg/m^3}$ are initially placed in the left and right halves of the container, respectively. The container has a length of $L = 1.82$ m and a height of $H = 0.2$ m. The kinematic viscosity is set to $1 \times 10^{-6}\ \mathrm{m^2/s}$ for both fluids. The gravitational acceleration is taken as $g = 9.81\ \mathrm{m/s^2}$. The artificial sound speeds of the two fluids are set to the same value, namely $10\sqrt{2gH}$. No-slip boundary conditions are imposed on the walls in this case. The nondimensional time is defined as $t^* = t\sqrt{g(1-r)/H}$, where $r = \rho_2/\rho_1$. Meanwhile, a 3D case is also considered, where the added dimension is the spanwise width of 0.08 m, and all the settings are the same otherwise.

Fig. 26 shows the simulation snapshots obtained with a resolution of $\Delta\bar{x} = H/40$, where the phase and pressure fields are presented in contours. From the phase field, it can be clearly seen that the heavier fluid propagates to the right along the bottom of the container, and, due to the shear stress acting at the interface, many vortical structures develop. The pressure fields remain quite smooth throughout this process. Fig. 27 shows the flow profiles at $t^* = 4.2$ obtained with various resolutions. The results indicate that the overall flow profiles are consistent across all resolutions. The main difference is that the small vortical structures become increasingly complex as the resolution is refined. At a fixed resolution of $\Delta\bar{x} = H/80$, the present results exhibit many more small-scale vortical structures than those reported in either [29] or [74]. In contrast, even at the relatively low resolution of $\Delta\bar{x} = H/40$, such vortical structures have already developed clearly. As for purely mesh-based methods, these small-scale structures are usually subject to strong numerical dissipation, especially in weakly compressible flows. The displacement histories of the light-fluid front obtained with various resolutions are shown in Fig. 28(a) and compared with the experimental data reported in [101]. The results show that the present simulations are consistent across different resolutions and agree well with the experimental data. Even at the relatively low resolution of $\Delta\bar{x} = H/30$, the result is already very close to the converged solution. The 3D simulation is shown in Fig. 29 at $t^* = 4.2$, where the same profile is observed compared with the 2D result. In Fig. 28(b), the light fluid front shows quite

good agreement between 2D and 3D results. Overall, this case demonstrates that the present method can provide accurate results at very low resolutions while maintaining very low numerical dissipation near the interface. In addition, the 3D capability of PG-DG for multiphase flow is again validated.

6.4.2. Gravity currents of hybrid phases

In the final case, we demonstrate the performance of the present method in simulating flows involving both multiphase interfaces and free surfaces. The setup of this case is adopted from [80]. In this case, two fluids with densities $\rho_1 = 2$ and $\rho_2 = 1$ are placed in the container, as shown in Fig. 30, while the remaining part of the container is treated as vacuum. According to [80], the dynamic viscosities are set to 1.41×10^{-6} and 1×10^{-6}, respectively, and no-slip boundary conditions are imposed on the walls. The gravitational acceleration is set to $g = 1$ in the vertical downward direction. The artificial sound speed $10\sqrt{2gH}$ is found to be sufficient for both fluids. The nondimensional time is defined as $t^* = t\sqrt{g/H}$.

Fig. 31 shows the phase fields at various times obtained with a resolution of $\Delta\bar{x} = H/25$. The results indicate that both the multiphase interface and the free surfaces are well captured simultaneously. Comparison with the reference results reported in [80] shows good agreement at different times. In particular, some small vortical structures are captured, consistent with the reference results. Notably, the present simulation appears to resolve more small-scale vortical structures than the reference results, even though the adopted resolution is lower than those used in the references. In order to show the low dissipation property of the present method, multiple resolutions are employed for simulation, for which the phase fields at $t^* = 5.15$ are shown in Fig. 32. One can clearly observe that the vortical structures and small fragments emerge when the resolution is increased. In contrast, the small structures are usually dissipated using conventional pure mesh-based methods based on the advection equation. To quantitatively assess the performance of the present method, the flow height measured at $x = 1.5$ is shown in Fig. 33, where results obtained at multiple resolutions are compared. The results show clear convergence among the simulations, and even a relatively low resolution of $\Delta\bar{x} = H/20$ already yields an accurate prediction. This further demonstrates that the present method remains accurate even at low resolutions.

7. Concluding remarks

This study proposes a particle-grid DG method for simulating weakly compressible free-surface and multiphase flows. Different sets of particles are employed in the fluid domain to represent different fluid phases, and these particles are embedded in background DG grid cells. In the P2G step, particles are mapped to DG grid cells according to the *phase status* of each cell, which is determined by the phase fraction. The phase fraction of a cell is computed using particles located in its neighboring cells, thereby ensuring mass conservation in the particle domain. The field values are then updated on the DG grid. In the G2P step, the grid field values are mapped back to particles using the DG test functions. Surface particles can be readily identified with the present method, and the particle distribution is correspondingly regularized using a kernel-based technique. Free-surface fluxes are determined according to whether a neighboring cell is vacuum. In multiphase interfacial regions, a post-mixing rule is introduced for the PG-DG method, which avoids the excessive numerical dissipation commonly encountered in conventional VOF or level-set methods for weakly compressible flows.

The proposed PG-DG method is validated through a series of benchmark cases. The results show that the present method achieves second-order convergence in cases without free surfaces or multiphase interfaces, while the convergence rate lies between first and second order in flows involving interfaces. Across all test cases, the proposed method demonstrates high robustness and accuracy in simulating free-surface flows, multiphase flows with large density ratios, and combined free-surface and multiphase flows in both 2D and 3D configurations. Nearly all cases show that the present method can provide reliable results even at low resolutions, where conventional SPH methods generally struggle to achieve comparable accuracy. In fact, the method can achieve accuracy comparable to, or even better than, that of mesh-based methods. A particularly notable advantage is that the proposed method does not suffer from mass dissipation, which remains a longstanding challenge for purely mesh-based methods. In addition, the method exhibits very low energy dissipation in multiphase flow simulations. There are also, at present, several limitations of the PG-DG method: the method is based on structured Cartesian grids, which means that treating complex geometry may not be easy in this version; the method is based on a weakly compressible EOS; and accuracy near free surfaces is lower than the expected second order. Future work aims to address these limitations, including extending the PG-DG method to the important case of strongly compressible multiphase flows.

CRediT authorship contribution statement

Tianrun Gao: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization; **David Hyde:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis.

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

Data will be made available on request.

References

[1] R.J. LeVeque, Finite Difference Methods For Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, SIAM, 2007. https://doi.org/10.1137/1.9780898717839
[2] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Vol. 31, Cambridge University Press, 2002. https://doi.org/10.1017/cbo9780511791253
[3] A.N. Brooks, T.J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1-3) (1982) 199–259. https://doi.org/10.1016/0045-7825(82)90071-8
[4] R. Bridson, Fluid Simulation For Computer Graphics, AK Peters/CRC Press, 2015. https://doi.org/10.1201/9781315266008
[5] D. Roychowdhury, Computational Fluid Dynamics for Incompressible Flows, CRC Press, 2020. https://doi.org/10.1201/9780367809171
[6] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225. https://doi.org/10.1016/0021-9991(81)90145-5
[7] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246. https://doi.org/10.1016/j.jcp.2005.04.007
[8] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (2) (1999) 423–456. https://doi.org/10.1006/jcph.1998.6168
[9] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (2) (1998) 112–152. https://doi.org/10.1006/jcph.1998.5906
[10] J.E. Pilliod Jr, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2) (2004) 465–502. https://doi.org/10.1016/j.jcp.2003.12.023
[11] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (2) (1999) 457–492. https://doi.org/10.1006/jcph.1999.6236
[12] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2) (2000) 301–337. https://doi.org/10.1006/jcph.2000.6537
[13] S. Osher, R. Fedkiw, K. Piechor, Level set methods and dynamic implicit surfaces, Appl. Mech. Rev. 57 (3) (2004) B15–B15.
[14] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2) (2001) 463–502. https://doi.org/10.1006/jcph.2000.6636
[15] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, J. Comput. Phys. 353 (2018) 82–109. https://doi.org/10.1016/j.jcp.2017.10.006
[16] F. Gibou, D. Hyde, R. Fedkiw, Sharp interface approaches and deep learning techniques for multiphase flows, J. Comput. Phys. 380 (2019) 442–463. https://doi.org/10.1016/j.jcp.2018.05.031
[17] D.A. Di Pietro, S.L. Forte, N. Parolini, Mass preserving finite element implementations of the level set method, Appl. Numer. Math. 56 (9) (2006) 1179–1195. https://doi.org/10.1016/j.apnum.2006.03.003
[18] M. Ataei, M. Bussmann, V. Shaayegan, F. Costa, S. Han, C.B. Park, NPLIC: A machine learning approach to piecewise linear interface construction, Comput. Fluids 223 (2021) 104950. https://doi.org/10.1016/j.compfluid.2021.104950
[19] C.W. Hirt, A.A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253. https://doi.org/10.1016/0021-9991(74)90051-5
[20] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1) (1992) 25–37. https://doi.org/10.1016/0021-9991(92)90307-k
[21] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian methods, in: Encyclopedia of Computational Mechanics, 2004. https://doi.org/10.1002/0470091355.ecm009
[22] J. Baiges, R. Codina, A. Pont, E. Castillo, An adaptive fixed-mesh ALE method for free surface flows, Comput. Methods Appl. Mech. Eng. 313 (2017) 159–188. https://doi.org/10.1016/j.cma.2016.09.041
[23] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. R. Astron. Soc. 181 (3) (1977) 375–389. https://doi.org/10.1093/mnras/181.3.375
[24] J.J. Monaghan, Smoothed particle hydrodynamics, Annu. Rev. Astron. Astrophys. 30 (1) (1992) 543–574. https://doi.org/10.1146/annurev.aa.30.090192.002551
[25] J.J. Monaghan, Simulating free surface flows with SPH, J. Comput. Phys. 110 (2) (1994) 399–406. https://doi.org/10.1006/jcph.1994.1034
[26] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, J. Comput. Phys. 191 (2) (2003) 448–475. https://doi.org/10.1016/S0021-9991(03)00324-3
[27] X.Y. Hu, N.A. Adams, A multi-phase SPH method for macroscopic and mesoscopic flows, J. Comput. Phys. 213 (2) (2006) 844–861. https://doi.org/10.1016/j.jcp.2005.09.001
[28] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, G. Graziani, δ-SPH model for simulating violent impact flows, Comput. Methods Appl. Mech. Eng. 200 (13-16) (2011) 1526–1542. https://doi.org/10.1016/j.cma.2010.12.016
[29] Z. Chen, Z. Zong, M. Liu, L. Zou, H. Li, C. Shu, An SPH model for multiphase flows with complex interfaces and large density differences, J. Comput. Phys. 283 (2015) 169–188. https://doi.org/10.1016/j.jcp.2014.11.037
[30] M. Liu, G. Liu, Smoothed particle hydrodynamics (SPH): an overview and recent developments, Arch. Comput. Methods Eng. 17 (2010) 25–76. https://doi.org/10.1007/s11831-010-9040-7
[31] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nucl. Sci. Eng. 123 (3) (1996) 421–434. https://doi.org/10.13182/nse96-a24205
[32] S. Shao, E.Y. Lo, Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface, Adv. Water Resour. 26 (7) (2003) 787–800. https://doi.org/10.1016/s0309-1708(03)00030-7
[33] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, Int. J. Numer. Methods Eng. 37 (2) (1994) 229–256. https://doi.org/10.1002/nme.1620370205
[34] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Int. J. Numer. Methods Fluids 20 (8-9) (1995) 1081–1106. https://doi.org/10.1002/fld.1650200824
[35] F.H. Harlow, The particle-in-cell computing method for fluid dynamics, Methods Comput. Phys. 3 (1964) 319–343.
[36] J.U. Brackbill, H.M. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, J. Comput. Phys. 65 (2) (1986) 314–343. https://doi.org/10.1016/0021-9991(86)90211-1
[37] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Comput. Methods Appl. Mech. Eng. 118 (1–2) (1994) 179–196. https://doi.org/10.1016/0045-7825(94)90112-0
[38] L. Tran, J. Kim, M. Berzins, Solving time-dependent PDEs using the material point method, a case study from gas dynamics, Int. J. Numer. Methods Fluids 62 (7) (2010) 709–732. https://doi.org/10.1002/fld.2031
[39] Y. Cao, Y. Chen, M. Li, Y. Yang, X. Zhang, M. Aanjaneya, C. Jiang, An efficient B-spline Lagrangian/Eulerian method for compressible flow, shock waves, and fracturing solids, ACM Trans. Graph. 41 (5) (2022) 1–13. https://doi.org/10.1145/3519595
[40] S. Kularathna, K. Soga, Implicit formulation of material point method for analysis of incompressible materials, Comput. Methods Appl. Mech. Eng. 313 (2017) 673–686. https://doi.org/10.1016/j.cma.2016.10.013

[41] J. Li, Y. Hamamoto, Y. Liu, X. Zhang, Sloshing impact simulation with material point method and its experimental validations, Comput. Fluids 103 (2014) 86–99. https://doi.org/10.1016/j.compfluid.2014.07.025
[42] R.M. Telikicherla, G. Moutsanidis, A displacement-based material point method for weakly compressible free-surface flows, Comput. Mech., 75 (1) (2025) 389–405. https://doi.org/10.1007/s00466-024-02510-3
[43] J.-H. Wang, W.-J. Xu, Y.-Y. Zhang, A hybrid finite volume and material point method (FVMPM) for simulating multiphase flow in and around deformable porous media, Comput. Geotech. 191 (2026) 107831. https://doi.org/10.1016/j.compgeo.2025.107831
[44] C. Mast, P. Mackenzie-Helnwein, P. Arduino, G.R. Miller, W. Shin, Mitigating kinematic locking in the material point method, J. Comput. Phys. 231 (16) (2012) 5351–5373. https://doi.org/10.1016/j.jcp.2012.04.032
[45] Z.-P. Chen, X. Zhang, K.Y. Sze, L. Kan, X.-M. Qiu, VP material point method for weakly compressible problems, Comput. Fluids 176 (2018) 170–181. https://doi.org/10.1016/j.compfluid.2018.09.005
[46] Y. Zhao, C. Jiang, J. Choo, Circumventing volumetric locking in explicit material point methods: a simple, efficient, and general approach, Int. J. Numer. Methods Eng. 124 (23) (2023) 5334–5355. https://doi.org/10.1002/nme.7347
[47] S. Peddavarapu, T.-H. Huang, Reproducing kernel enhanced material point method with improved artificial viscosity formulations for shock wave and vortical compressible flows, Comput. Mech. 76 (3) (2025) 797–828. https://doi.org/10.1007/s00466-025-02627-z
[48] D.Z. Zhang, Q. Zou, W.B. VanderHeyden, X. Ma, Material point method applied to multiphase flows, J. Comput. Phys. 227 (6) (2008) 3159–3173. https://doi.org/10.1016/j.jcp.2007.11.021
[49] S. McKee, M.F. Tomé, V.G. Ferreira, J.A. Cuminato, A. Castelo, F. Sousa, N. Mangiavacchi, The MAC method, Comput. Fluids 37 (8) (2008) 907–930. https://doi.org/10.1016/j.compfluid.2007.10.006
[50] J. Liu, S. Koshizuka, Y. Oka, A hybrid particle-mesh method for viscous, incompressible, multiphase flows, J. Comput. Phys. 202 (1) (2005) 65–93. https://doi.org/10.1016/j.jcp.2004.07.002
[51] Y. Xu, G. Yang, Y. Zhu, D. Hu, A coupled SPH–FVM method for simulating incompressible interfacial flows with large density difference, Eng. Anal. Bound. Elem. 128 (2021) 227–243. https://doi.org/10.1016/j.enganabound.2021.04.005
[52] S. Liu, G. Yang, Y. Xu, D. Hu, A SPH-FVM coupling method based on triangular mesh for the simulation of two-phase flows, Eng. Anal. Bound. Elem. 159 (2024) 237–258. https://doi.org/10.1016/j.enganabound.2023.12.009
[53] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Tech. rep., Los Alamos Scientific Lab., N. Mex. (USA), 1973.
[54] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework, Math. Comput. 52 (186) (1989) 411–435. https://doi.org/10.1090/s0025-5718-1989-0983311-4
[55] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1) (1989) 90–113. https://doi.org/10.1016/0021-9991(89)90183-6
[56] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case, Math. Comput. 54 (190) (1990) 545–581. https://doi.org/10.1090/s0025-5718-1990-1010597-0
[57] M.T.H. de Frahan, S. Varadan, E. Johnsen, A new limiting procedure for discontinuous Galerkin methods applied to compressible multiphase flows with shocks and interfaces, J. Comput. Phys. 280 (2015) 489–509. https://doi.org/10.1016/j.jcp.2014.09.030
[58] Q. Kong, Y.-L. Liu, Y.-K. Li, S. Ma, Q.-H. Hao, A. Zhang, A-based compressible multiphase model with cavitation based on discontinuous galerkin method, Phys. Fluids 37 (1) (2025). https://doi.org/10.1063/5.0246222
[59] X. Zhao, Z. Li, X. Yu, F. Qing, S. Zou, A physical-constraints-preserving arbitrary Lagrangian–Eulerian discontinuous Galerkin scheme on adaptive unstructured meshes for two-component flows, Phys. Fluids 37 (7) (2025). https://doi.org/10.1063/5.0275778
[60] F. Zhang, J. Cheng, T. Liu, A physical-constraint-preserving discontinuous Galerkin method for weakly compressible two-phase flows, J. Sci. Comput. 96 (3) (2023) 84. https://doi.org/10.1007/s10915-023-02306-2
[61] Z. Li, G. Oger, D. Le Touzé, A finite volume WENO scheme for immiscible inviscid two-phase flows, J. Comput. Phys. 418 (2020) 109601. https://doi.org/10.1016/j.jcp.2020.109601
[62] Z. Li, L. Vittoz, G. Oger, D. Le Touzé, A simplified and efficient weakly-compressible FV-WENO scheme for immiscible two-phase flows, Comput. Fluids 244 (2022) 105555. https://doi.org/10.1016/j.compfluid.2022.105555
[63] G.B. Jacobs, J.S. Hesthaven, High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids, J. Comput. Phys. 214 (1) (2006) 96–121. https://doi.org/10.1016/j.jcp.2005.09.008
[64] A. Renaud, T. Heuzé, L. Stainier, A discontinuous galerkin material point method for the solution of impact problems in solid dynamics, J. Comput. Phys. 369 (2018) 80–102. https://doi.org/10.1016/j.jcp.2018.05.001
[65] A. Renaud, T. Heuzé, L. Stainier, The discontinuous Galerkin material point method for variational hyperelastic–plastic solids, Comput. Methods Appl. Mech. Eng. 365 (2020) 112987. https://doi.org/10.1016/j.cma.2020.112987
[66] G.K. Batchelor, An Introduction to Fluid Dynamics, Cambridge University Press, 2000. https://doi.org/10.1017/cbo9780511800955
[67] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1) (1995) 389–396. https://doi.org/10.1007/bf02123482
[68] C. Jiang, C. Schroeder, A. Selle, J. Teran, A. Stomakhin, The affine particle-in-cell method, ACM Trans. Graph. 34 (4) (2015) 1–10. https://doi.org/10.1145/2766996
[69] K. Nakamura, S. Matsumura, T. Mizutani, Taylor particle-in-cell transfer and kernel correction for material point method, Comput. Methods Appl. Mech. Eng. 403 (2023) 115720. https://doi.org/10.1016/j.cma.2022.115720
[70] X.-s. Li, C.w. Gu, Mechanism of Roe-type schemes for all-speed flows and its application, Comput. Fluids 86 (2013) 56–70. https://doi.org/10.1016/j.compfluid.2013.07.004
[71] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: 27th Aerospace Sciences Meeting, 1989, p. 366. https://doi.org/10.2514/6.1989-366
[72] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Math. Comput. 67 (221) (1998) 73–85. https://doi.org/10.1090/s0025-5718-98-00913-2
[73] J.J. Monaghan, Smoothed particle hydrodynamics, Rep. Prog. Phys. 68 (8) (2005) 1703. https://doi.org/10.1088/0034-4885/68/8/r01
[74] Z.-F. Meng, P.-P. Wang, A.-M. Zhang, F.-R. Ming, P.-N. Sun, A multiphase SPH model based on Roe's approximate Riemann solver for hydraulic flows with complex interface, Comput. Methods Appl. Mech. Eng. 365 (2020) 112999. https://doi.org/10.1016/j.cma.2020.112999
[75] I. Hammani, S. Marrone, A. Colagrossi, G. Oger, D. Le Touzé, Detailed study on the extension of the δ-SPH model to multi-phase flow, Comput. Methods Appl. Mech. Eng. 368 (2020) 113189. https://doi.org/10.1016/j.cma.2020.113189
[76] S. Marrone, A. Colagrossi, D. Le Touzé, G. Graziani, Fast free-surface detection and level-set function definition in SPH solvers, J. Comput. Phys. 229 (10) (2010) 3652–3663. https://doi.org/10.1016/j.jcp.2010.01.019
[77] T. Gao, L. Fu, A new particle shifting technique for SPH methods based on Voronoi diagram and volume compensation, Comput. Methods Appl. Mech. Eng. 404 (2023) 115788. https://doi.org/10.1016/j.cma.2022.115788
[78] D.A.B. Hyde, S.W. Gagniere, A. Marquez-Razon, J. Teran, An implicit updated Lagrangian formulation for liquids with large surface energy, ACM Trans. Graph. 39 (6) (2020) 1–13. https://doi.org/10.1145/3414685.3417845
[79] J. Chen, V. Kala, A. Marquez-Razon, E. Gueidon, D.A.B. Hyde, J. Teran, A momentum-conserving implicit material point method for surface tension with contact angles and spatial gradients, ACM Trans. Graph. 40 (4) (2021) 1–16. https://doi.org/10.1145/3450626.3459874
[80] N. Grenier, M. Antuono, A. Colagrossi, D.L. Touzé, B. Alessandrini, An Hamiltonian interface SPH formulation for multi-fluid and free surface flows, J. Comput. Phys. 228 (22) (2009) 8380–8393. https://doi.org/10.1016/j.jcp.2009.08.009
[81] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach, J. Comput. Phys. 125 (1) (1996) 150–160. https://doi.org/10.1006/jcph.1996.0085

[82] J.J. Monaghan, A. Rafiee, A simple SPH algorithm for multi-fluid flow with high density ratios, Int. J. Numer. Methods Fluids 71 (5) (2013) 537–561. https://doi.org/10.1002/fld.3671
[83] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: Numerical Methods for Fluid Dynamics, 1982.
[84] S.J. Cummins, M. Rudman, An SPH projection method, J. Comput. Phys. 152 (2) (1999) 584–607. https://doi.org/10.1006/jcph.1999.6246
[85] S. Adami, X. Hu, N.A. Adams, A transport-velocity formulation for smoothed particle hydrodynamics, J. Comput. Phys. 241 (2013) 292–307. https://doi.org/10.1016/j.jcp.2013.01.043
[86] S. Dalziel, Toy models for Rayleigh Taylor instability, in: 8th International Workshop on the Physics of Compressible Turbulent Mixing, Vol. 10, 2001.
[87] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, A. Souto-Iglesias, Experimental investigation of dynamic pressure loads during dam break, J. Fluids Struct. 48 (2014) 407–434. https://doi.org/10.1016/j.jfluidstructs.2014.03.009
[88] B. Chandra, R. Hashimoto, S. Matsumi, K. Kamrin, K. Soga, Stabilized mixed material point method for incompressible fluid flow analysis, Comput. Methods Appl. Mech. Eng. 419 (2024) 116644. https://doi.org/10.1016/j.cma.2023.116644
[89] S. Adami, X.Y. Hu, N.A. Adams, A generalized wall boundary condition for smoothed particle hydrodynamics, J. Comput. Phys. 231 (21) (2012) 7057–7075. https://doi.org/10.1016/j.jcp.2012.05.005
[90] P.-N. Sun, C. Pilloton, M. Antuono, A. Colagrossi, Inclusion of an acoustic damper term in weakly-compressible SPH models, J. Comput. Phys. 483 (2023) 112056. https://doi.org/10.1016/j.jcp.2023.112056
[91] A. Khayyer, Y. Shimizu, T. Gotoh, H. Gotoh, Enhanced resolution of the continuity equation in explicit weakly compressible SPH simulations of incompressible free-surface fluid flows, Appl. Math. Model. 116 (2023) 84–121. https://doi.org/10.1016/j.apm.2022.10.037
[92] A. Melvin, J. Mandal, Development of a weakly compressible solver for incompressible two-phase flows, J. Comput. Phys. 538 (2025) 114195. https://doi.org/10.1016/j.jcp.2025.114195
[93] F. Garoosi, K. Hooman, Numerical simulation of multiphase flows using an enhanced Volume-of-Fluid (VOF) method, Int. J. Mech. Sci. 215 (2022) 106956. https://doi.org/10.1016/j.ijmecsci.2021.106956
[94] H.S. Yoo, Y.B. Jo, E.S. Kim, Comparative study of WCSPH, EISPH and explicit incompressible-compressible SPH (EICSPH) for multi-phase flow with high density difference, J. Comput. Phys. 506 (2024) 112930. https://doi.org/10.1016/j.jcp.2024.112930
[95] M. Rezavand, C. Zhang, X. Hu, A weakly compressible SPH method for violent multi-phase flows with high density ratio, J. Comput. Phys. 402 (2020) 109092. https://doi.org/10.1016/j.jcp.2019.109092
[96] Z. Zhou, P. Han, Y. Sun, Z. Sun, G. Xi, Incompressible-compressible moving particle and moving Voronoi mesh method for multiphase flow, J. Comput. Phys.,537, 2025 114062. https://doi.org/10.1016/j.jcp.2025.114062
[97] W. Meng, C.-h. Yu, J. Li, Z. Wu, R. An, Modeling of interfacial two-phase flows via VOF-based LS method with WENO scheme in the finite volume method, Comput. Fluids 291 (2025) 106580. https://doi.org/10.1016/j.compfluid.2025.106580
[98] S. Marrone, A. Colagrossi, A. Di Mascio, D. Le Touzé, Analysis of free-surface flows through energy considerations: single-phase versus two-phase modeling, Phys. Rev. E 93 (2016) 053113. https://doi.org/10.1103/PhysRevE.93.053113
[99] A. Mokos, B.D. Rogers, P.K. Stansby, A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles, J. Hydraul. Res. 55 (2) (2017) 143–162. https://doi.org/10.1080/00221686.2016.1212944
[100] H.-G. Lyu, P.-N. Sun, Further enhancement of the particle shifting technique: towards better volume conservation and particle distribution in SPH simulations of violent free-surface flows, Appl. Math. Model. 101 (2022) 214–238. https://doi.org/10.1016/j.apm.2021.08.014
[101] R.J. Lowe, J.W. Rottman, P.F. Linden, The non-Boussinesq lock-exchange problem. Part 1. Theory and experiments, J. Fluid Mech. 537 (2005) 101–124. https://doi.org/10.1017/s0022112005005069