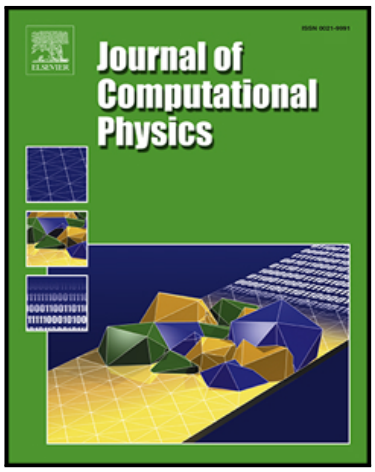
Journal Pre-proof



A Particle-Grid Discontinuous Galerkin Method, Part 2: Strongly Compressible Multiphase Flows

Tianrun Gao, David Hyde

PII: S0021-9991(26)00956-3
DOI: https://doi.org/10.1016/j.jcp.2026.115608
Reference: YJCPH 115608

To appear in: *Journal of Computational Physics*

Received date: 4 August 2026
Revised date: 17 November 2026
Accepted date: 17 November 2026

Please cite this article as: Tianrun Gao, David Hyde, A Particle-Grid Discontinuous Galerkin Method, Part 2: Strongly Compressible Multiphase Flows, *Journal of Computational Physics* (2026), doi: https://doi.org/10.1016/j.jcp.2026.115608

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2026 Published by Elsevier Inc.

Journal Pre-proof

A Particle-Grid Discontinuous Galerkin Method, Part 2: Strongly Compressible Multiphase Flows

Tianrun Gao[a], David Hyde[a,*]

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

Abstract

This study proposes a particle-grid discontinuous Galerkin (PG-DG) method for strongly compressible multiphase flows, which naturally extends the PG-DG framework previously developed for weakly compressible free-surface and multiphase flows. In the PG-DG method, both particles and a background grid are employed, where different sets of particles carrying all field variables are used to represent fluids of different phases, while the governing equations are solved on the grid to update the particle field variables. The PG-DG method allows one-particle-per-cell resolution. The particle-to-grid (P2G) mapping is performed according to whether the *phase status*, determined by the *phase fraction*, changes within a grid cell. The *phase fraction*, which replaces the conventional advection equation, is computed directly from the particle distribution. The grid-to-particle (G2P) mapping is naturally implemented using DG test functions. A flux-mixing method is specifically designed for strongly compressible multiphase flows, enabling the unified treatment of flows involving more than two phases while maintaining a sharp interface. A mixed limiter is developed accordingly for the flux-mixing method. A uniform particle distribution is maintained using particle splitting and merging (PSM) and the particle shifting technique (PST). A set of 2D and 3D cases demonstrates that the proposed PG-DG method is accurate and robust, with its performance maintained even for flows with high Mach numbers and large density ratios. Moreover, multi-component flows involving more than two phases can be handled naturally, while sharp multiphase interfaces are well preserved. Overall, the PG-DG method is fundamentally different from the widely used volume-of-fluid and level-set methods, and provides a new effective tool for simulating complex strongly compressible multiphase flows.

Keywords: Particle-grid method, Discontinuous Galerkin, Strongly compressible flow, Multiphase flow

1. Introduction

Strongly compressible multiphase flows are widely encountered in aerospace engineering, ocean engineering, energy and power systems, and national defense applications. Typical examples include cavitation bubble collapse around ship propellers [1], supersonic and hypersonic combustion [2], and underwater explosions [3]. The interfacial dynamics in such flows are highly complex due to strong compressibility effects, making analytical solutions generally intractable. Moreover, experimental investigations are often challenged by the difficulty of accurately capturing the rapidly evolving interfaces. Consequently, numerical simulation has become a cost-effective and powerful alternative for investigating the complex interfacial phenomena in strongly compressible multiphase flows.

Conventional mesh-based methods have long served as powerful tools for simulating strongly compressible multiphase flows. These methods typically solve additional interface-capturing or interface-tracking equations alongside the governing fluid equations. Interface evolution is commonly described using the volume-of-fluid (VOF) method [4, 5, 6, 7] or the level-set method [8, 9, 10, 11, 12]. According to the treatment of the material interface, these methods can generally be classified into two categories: diffuse interface methods (DIMs) and sharp interface methods (SIMs) [13]. In DIMs, the interface is represented as a finite-thickness mixing region between different phases, allowing all phases to be treated within a unified single-fluid framework. The primary challenge of DIMs lies in constructing an equation of state (EOS)

*Corresponding author.
Email address: david.hyde.1@vanderbilt.edu (David Hyde)

Journal Pre-proof

for the mixing region that is both physically and mathematically consistent. Depending on the equilibrium assumptions imposed in the diffuse region, several governing models have been proposed, including the seven-equation model [14], the six-equation model [15], and the five-equation model [16]. Among them, the five-equation model, which assumes mechanical and thermal equilibrium, is the simplest and has been widely adopted in recent simulations of compressible multiphase flows [17, 18, 19, 20]. Despite their simplicity and robustness, DIMs inevitably suffer from numerical diffusion across the interface, particularly in problems involving large density ratios. To alleviate this issue, numerous high-order interface-capturing schemes have been developed, such as deploying weighted/targeted essentially non-oscillatory (WENO/TENO) schemes [21, 22] with the tangent of hyperbola for interface capturing (THINC) method [23], thereby achieving significantly sharper interface resolution [19, 20]. Nevertheless, the intrinsic numerical diffusion associated with the diffuse-interface formulation cannot be completely eliminated. In contrast, SIMs keep the interface sharp. A representative class of SIMs employs the level-set method for interface tracking, including the Osher-Sethian level-set method [24] and the ghost fluid method (GFM) [8]. A well-known limitation of the classical level-set method is the loss of mass during interface evolution, and to solve this problem, a conservative improvement is proposed in [25]. Another classic work [9] combines level-set and VOF to ensure mass conservation while using the geometric calculation capability of the level-set method. Another widely used SIM is the front-tracking method [26], which employs Lagrangian marker points to explicitly track interface motion. However, it becomes increasingly difficult to handle severe interface deformation, topological changes, and three-dimensional problems. Similar challenges are encountered by moving-mesh methods based on Lagrangian [27] or arbitrary Lagrangian-Eulerian (ALE) [28] frameworks. In addition, interface reconstruction techniques, such as piecewise linear interface construction (PLIC) [29] and cut-cell methods [30, 31, 32], can achieve high interface fidelity but require sophisticated geometric reconstruction algorithms, especially in 3D simulations.

Apart from pure mesh-based methods, meshless methods have also been increasingly explored for the simulation of strongly compressible flows. Smoothed particle hydrodynamics (SPH) is a typical meshless method for fluid simulation, and its application has recently been extended from weakly compressible flows to strongly compressible flows [33, 34, 35, 36]. However, SPH usually suffers from either spurious oscillations or low accuracy, and it is difficult to address these two issues simultaneously. Thus, both pure mesh-based and meshless methods have their own limitations in simulating multiphase flows.

In light of this, combining the two methods appears to be a feasible approach, which is generally referred to as the particle-grid (PG) method. The PG method was first proposed in [37] as the particle-in-cell (PIC) method, in which field variables are mapped back and forth between particles and grid cells, resulting in excessive numerical dissipation. This method was later improved by the fluid implicit particle (FLIP) method [38]. Although numerical dissipation is significantly reduced, FLIP requires a large number of particles within each cell to maintain simulation robustness. In recent decades, the PG approach has been extended to solid mechanics in the form of the material point method (MPM) [39]. There have been a few studies applying MPM to strongly compressible flows [40, 41, 42]; however, they are limited to relatively simple cases and exhibit significant spurious oscillations near discontinuities. These oscillations arise because the continuous Galerkin (CG) formulation adopted in MPM is not well-suited for convection-dominated problems. The marker-and-cell (MAC) method was originally proposed for incompressible flows [43]. However, it cannot be directly applied to compressible multiphase flows and still requires multiple particles within each grid cell.

Compared with the CG approach adopted in conventional PG methods, flux-based methods, such as the finite volume method (FVM) [44], are more suitable for convection-dominated problems. However, the FVM stores only cell averages and needs neighbor-based reconstruction, which complicates particle–grid mapping. As a compromise between FVM and CG, the discontinuous Galerkin (DG) method [45, 46, 47, 48] combines the use of test functions with a flux-based formulation, making it well suited for achieving high-order accuracy and facilitating particle-grid mapping. Recently, DG has been extended to the simulation of strongly compressible multiphase flows [49, 50, 51, 52, 31, 53, 54, 55, 56, 57, 58], where most interface-tracking methods are based on the level-set approach together with multi-resolution techniques [53, 51] or cut-cell methods [56, 31], while some adopt the DIM approach similar to the VOF method [50, 49, 57, 58]. However, these methods either assume a diffuse interface or suffer from numerical dissipation across the interface, and therefore still rely on conventional interface treatment techniques.

A particle-grid discontinuous Galerkin (PG-DG) method was proposed by the authors for weakly compressible free-surface and multiphase flows in Part 1 of this series [59]. The PG-DG method combines the

Journal Pre-proof

Table 1: Comparison of the PG-DG formulations in Parts 1 and 2.

Component	Part 1	Part 2
Applications	Weakly compressible free-surface and multiphase flows	Strongly compressible multiphase flows
Phase fraction	Different particle sets represent different phases; the phase fraction is evaluated directly from particles	Inherited
Particle-to-grid mapping	Phase-status-based selective P2G reconstruction	Inherited
DG field evolution	Phase-wise DG update	Inherited
Multiphase interface coupling	Post-mixing	Flux mixing
Limiter	B-J slope limiter	Mixed limiter accounting for phase fraction
Grid-to-particle mapping	DG-based G2P mapping and particle update	Inherited
Particle regularization	PST	Inherited, plus PSM & particle-volume evolution

Note: "Inherited" denotes components directly adopted from Part 1, whereas the other entries in the Part 2 column indicate newly introduced treatments.

advantages of both DIM and SIM while maintaining high accuracy and extremely low numerical dissipation. Unlike conventional PIC, FLIP, or MPM, the PG-DG method is capable of achieving accurate simulation results with a one-particle-per-cell resolution. In the present study, i.e., Part 2, the PG-DG method is extended to strongly compressible multiphase flows. A flux-mixing method is specially designed in this study, which is able to treat flows of more than two phases in a unified way while maintaining a sharp interface. A mixed limiter is devised correspondingly for the flux-mixing method. The uniform particle distribution is ensured using particle splitting and merging (PSM) and particle shifting technique (PST) approaches. Benchmark cases show that PG-DG is accurate and robust across all the cases, and performance is still maintained even for flows with high Mach numbers and large density ratios. Moreover, multi-component flows with more than two phases can be treated naturally, and the multiphase interface can be captured sharply. This PG-DG method distinguishes itself from the widely used VOF or level-set methods. This work further demonstrates that PG-DG is extensible to various kinds of flows. To further highlight the differences between Part 1 and Part 2, a comparison of the components is provided in Table 1.

The remainder of this paper is organized as follows. Section 2 presents the Euler equations and the DG formulation. Section 3 presents the phase fraction concept and particle-grid mapping methods. Section 4 introduces the limiter and flux-mixing approaches. Section 5 describes the grid-to-particle mapping, PSM and PST treatments. Section 6 summarizes the procedure of the PG-DG method. Section 7 shows various challenging cases for the validation of the PG-DG method. To facilitate reproducibility, source codes for a set of representative numerical examples are made publicly available at https://github.com/gaotr3306/PGDG_p2.

2. Euler equations and discontinuous Galerkin

2.1. Governing equations

The Euler equations in hyperbolic form are written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \tag{1}$$

Journal Pre-proof

with

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho \boldsymbol{v} \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho \boldsymbol{v}^{\mathrm{T}} \\ \rho \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} + p\mathbf{I} \\ (E+p)\boldsymbol{v}^{\mathrm{T}} \end{bmatrix}. \tag{2}$$

Here, ρ, $\boldsymbol{v}$, p, E denote the density, velocity, pressure and total energy density, respectively, and $\mathbf{I}$ is the identity matrix, where $E = \rho e + \frac{1}{2}\rho \boldsymbol{v}^2$, with e specific internal energy. In this study, a stiffened gas equation of state (EOS) is deployed for strongly compressible multiphase flows,

$$p = (\gamma - 1)\,\rho e - \gamma p_{\infty}. \tag{3}$$

2.2. Discontinuous Galerkin method for Euler equations

In this study, the PG-DG method is applied to structured Cartesian grids, where all cells constitute a set $\mathcal{C}$, and the resolutions of the grid are represented with Δx, Δy, and (in three spatial dimensions) Δz. If a cell is indexed with $i(i', j', k') \in \mathcal{C}$, then the cell's element domain is $\Omega_i = [x_{i'-\frac{1}{2}}, x_{i'+\frac{1}{2}}] \times [y_{j'-\frac{1}{2}}, y_{j'+\frac{1}{2}}] \times [z_{k'-\frac{1}{2}}, z_{k'+\frac{1}{2}}]$, 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 this grid setting, the Galerkin principle can be applied to Equation (1) cell-wise, written as

$$\int_{\Omega_i} \delta \mathbf{U}^{\mathrm{T}} \left[\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) \right] dV = 0. \tag{4}$$

The test function Φ in the present DG method is chosen from the linear polynomial space $\mathbb{P}^1(\Omega_i)$ over the cell domain, i.e.,

$$\mathcal{V}_h = \left\{ \Phi : \Phi \in \mathbb{P}^1(\Omega_i) \right\}. \tag{5}$$

Based on the above test function, the cell-wise conservative variables $\mathbf{U}_h$ of the Euler equations are approximated as

$$\mathbf{U}_h = \mathbf{U}^{(0)} + \mathbf{U}^{(1)}x + \mathbf{U}^{(2)}y + \mathbf{U}^{(3)}z = [\mathbf{U}][\mathbf{\Phi}]^{\mathrm{T}} = \mathbf{U}^{(0)} + [\mathbf{U}'][\mathbf{\Phi}']^{\mathrm{T}}, \tag{6}$$

where $\boldsymbol{x} - \boldsymbol{x}_c = (x, y, z)$ is the relative position in the cell, with $\forall \boldsymbol{x} \in \Omega_i$, and

$$[\mathbf{U}] = \left(\mathbf{U}^{(0)}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \right), \quad [\mathbf{\Phi}] = (1, x, y, z). \tag{7}$$

Rearranging Equation (4) based on the approximated $\mathbf{U}_h$ (Equation (6)), we then have the I-th component of $[\mathbf{U}]$ expressed as

$$\frac{\partial \mathbf{U}^{(J)}}{\partial t} \int_{\Omega_i} \Phi^{(I)} \Phi^{(J)} dV = \int_{\Omega_i} \mathbf{F} \nabla \Phi^{(I)} dV - \int_{\partial \Omega_i} \mathbf{F} \Phi^{(I)} \boldsymbol{n} dS, \quad I = 0, 1, 2, 3. \tag{8}$$

Since the employed linear-space test functions ensure orthogonality, we have $\int_{\Omega_i} \Phi^{(I)} \Phi^{(J)} dV = M_I \delta_{IJ}$, which is analogous to the mass-lumping treatment commonly used in dynamic systems.

3. Particle-to-grid mapping

The most distinctive feature of the PG-DG method is that, in addition to the computational grid, different sets of particles are also employed within the computational domain. Unlike previous PG-based methods, the particles in the PG-DG method not only serve as markers but also carry the field variables. This feature makes the PG-DG method similar to the material point method (MPM), while enabling it to better handle simulations of complex strongly compressible flows. As shown in Figure 1(a), three sets of particles are represented by different colors, and each occupies its own grid region, as illustrated in Figure 1(b∼d). These three regions overlap in part of the computational domain, which will be discussed in Section 4. The occupied grid region of each particle set is determined by its *phase status*, which is further characterized by the key concept of the *phase fraction*. Particle-to-grid mapping is unnecessary for all cells as long as their *phase status* remains unchanged. This avoids unnecessary remapping, thereby preventing the degradation of grid-based solution accuracy while preserving the advantages of the particle representation. This section introduces the key concept of the phase fraction and presents the corresponding particle-to-grid mapping method.

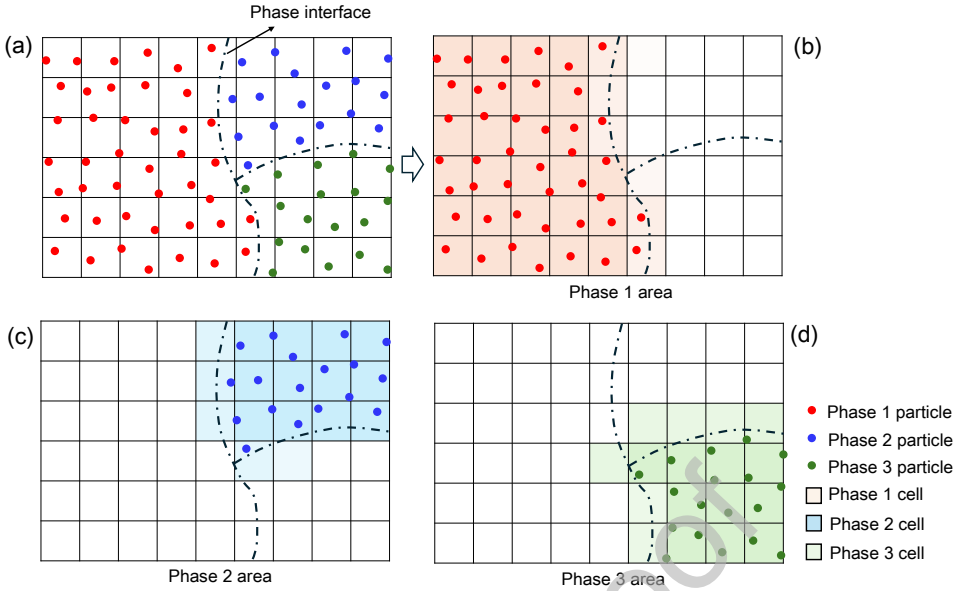


Figure 1. PG-DG treatment of multiphase flow. (a) Red, blue and green particles represent different fluids in the grid domain, and their occupied grid regions in (b), (c) and (d).

3.1. Phase fraction

In this work, we restrict our attention to Cartesian grids, which enables straightforward mapping from particles to grid cells. Let $\mathcal{P}$ and $\mathcal{C}$ be the set of particles and grid cells, respectively, and the mapping between each particle $p \in \mathcal{P}$ and its associated cell is

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

Then, all the particles in the cell i form a set

$$\mathcal{P}_i := \{\, p \in \mathcal{P} \mid \kappa(p) = i \,\}. \tag{9}$$

The particles can be searched efficiently using this method. In this method, the initial resolution $\Delta\bar{x}$ of particles is set as the same as the grid, i.e., $\Delta\bar{x} = \sqrt{\Delta x \Delta y}$ or $\sqrt[3]{\Delta x \Delta y \Delta z}$ in 2D or 3D, respectively, which significantly reduces the number of particles used compared to a traditional method such as PIC.

To calculate the phase fraction of cell i, its neighboring particles need to be identified. An efficient approach is to search within its neighboring cell set $\mathcal{N}_i$. As shown in Figure 2(a), $\mathcal{N}_i$ consists of cell i itself together with the surrounding one-cell-thick layer. Within $\mathcal{N}_i$, only the particles located within a radius of $1.5\Delta\bar{x}$ from the center of cell i are selected, i.e., $\|\boldsymbol{x}_p - \boldsymbol{x}_i\| < 1.5\Delta\bar{x}$.

For a particle p, we define the function

$$\omega(p) = \chi \tag{10}$$

to probe the fluid phase χ. To calculate the phase fraction $\alpha_{i,\chi}$ of phase χ in cell i, an intermediate variable, $\alpha'_{i,\chi}$, is first defined as

$$\alpha'_{i,\chi} = \sum_{\substack{\omega(p)=\chi, p\in\mathcal{P}_j, j\in\mathcal{N}_i \\ \|\boldsymbol{x}_{ip}\|<1.5\Delta\bar{x}}} W(\|\boldsymbol{x}_{ip}\|, h_i) V_p. \tag{11}$$

Here, V_p is the particle volume updated in Section 5.1, and h is the smoothing length. $W(r,h)$ is the window function chosen as a C^2 Wendland kernel function $W(r,h)$ [60], written as

$$W(r,h) = \begin{cases} C_h(1-\dfrac{r}{2h})^4(1+\dfrac{2r}{h}), & \frac{r}{h} \le 2, \\ 0, & \frac{r}{h} > 2, \end{cases} \tag{12}$$

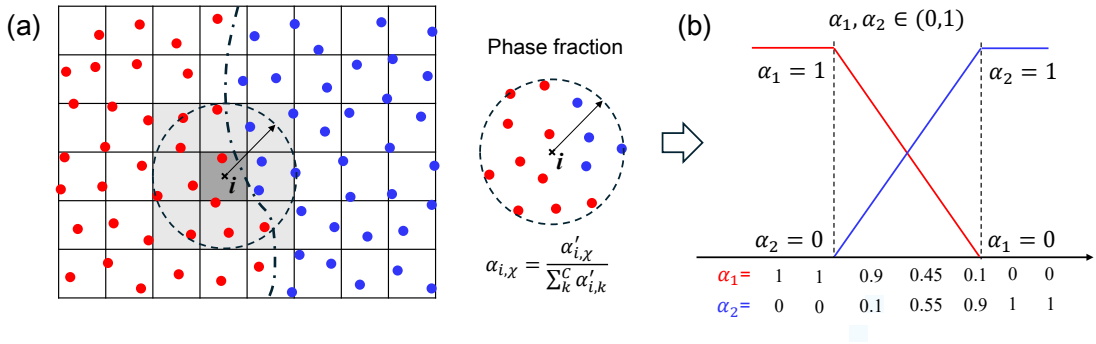


Figure 2. PG-DG calculation of the phase fraction across an interface. (a) The phase fraction of cell i is calculated using the neighboring particles within a radius of $1.5\Delta\bar{x}$ from the cell set $\mathcal{N}_i$, which consists of cell i itself and the surrounding one-cell-thick layer. (b) Across the interface, the phase fraction distribution is illustrated along the interface normal direction.

where $h = 1.6\Delta\bar{x}$ is initially assigned to both particles and grid cells. For grid cell i, h_i remains constant throughout the simulation, whereas for particle p, h_p is updated according to its volume in Section 5.1, where $h_p = 1.6\sqrt[Dim]{V_p}$, with Dim the spatial dimension. $C_h = \frac{7}{4\pi h^2}$ and $\frac{21}{16\pi h^3}$ in 2D and 3D space, respectively. After normalization, the phase fraction $\alpha_{i,\chi}$ is derived as

$$\alpha_{i,\chi} = \frac{\alpha'_{i,\chi}}{\sum_k^C \alpha'_{i,k}}, \tag{13}$$

where C is the total number of phases in the computational domain. We emphasize that this treatment can handle cases with more than two phases.

***Remark* 1.** In continuum space, the phase fraction defined above is calculated as

$$\alpha_\chi(\boldsymbol{x}) = \frac{\int_{\Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) dV'}{\int_{\Sigma_\chi^C \Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) dV'}, \tag{14}$$

where $\Omega_\chi(t)$ denotes the occupied domain of phase χ at time t. In this work, the computational domain has no vacuum area, and then $\int_{\Sigma_\chi^C \Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) d\boldsymbol{x}'$, denoted as m_0, should be invariant in the grid domain because $\Sigma_\chi^C \Omega_\chi(t)$ is always the window function area. Then the gradient of $\alpha_\chi(\boldsymbol{x})$ is calculated as

$$\nabla\alpha_\chi(\boldsymbol{x}) = -\frac{1}{m_0} \int_{\partial\Omega_\chi(t)} W(\boldsymbol{x}' - \boldsymbol{x}) \boldsymbol{n} dS. \tag{15}$$

Regarding the time derivative of $\alpha_\chi(\boldsymbol{x})$, it is calculated based on Equation (14) as

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

where $\bar{\boldsymbol{v}}$ is locally averaged velocity. One readily obtains

$$\frac{\partial\alpha_\chi}{\partial t} + \bar{\boldsymbol{v}} \cdot \nabla\alpha_\chi(\boldsymbol{x}) = 0. \tag{17}$$

This formula implies that the present definition of phase fraction is similar to the advection equation used in VOF or level-set approaches. One advantage of the present particle-based approach is that the phase interface is transported directly by particles, avoiding the numerical diffusion associated with solving an additional phase-advection equation. In addition, the direct calculation of the phase fraction can ensure monotonicity across the interface in the grid domain, which can achieve higher robustness. Moreover, the phase fraction has a fixed interface width according to its definition, and this interface is sharp in the particle field.

Journal Pre-proof

Journal Pre-proof

3.2. Phase status

The mapping from particles to grid cells is not globally required in the present method, since most of the regions occupied by different phases remain unchanged from time step n to $n+1$. Hence, the mapping is conducted only in the regions whose *phase status*, denoted using PS, changes from time step n to $n+1$. The *phase status* is determined by the phase fraction. For a cell i and phase χ, the phase is defined as *activated* if its phase fraction is greater than zero, denoted by $PS_{i,\chi} = 1$; otherwise, it is defined as *inactive*, with $PS_{i,\chi} = 0$, i.e.,

$$PS_{i,\chi} = \begin{cases} 1, & \alpha'_{i,\chi} > 0, \\ 0, & \alpha'_{i,\chi} = 0. \end{cases} \tag{18}$$

Note that each grid cell stores C values, corresponding to the C fluid phases present in the computational domain.

3.3. Particle-to-grid reconstruction at activated cells

As mentioned above, the P2G reconstruction is not required for every grid cell. Instead, it is performed only for cells whose phase status changes from 0 at time step n to 1 at time step $n+1$, i.e.,

$$PS^n_{i,\chi} = 0 \Rightarrow PS^{n+1}_{i,\chi} = 1.$$

When this condition is satisfied, the DG field values $\mathbf{U}^{(I)}_{i,\chi}$ are reconstructed from the corresponding particles of phase χ. The reconstruction employs the same particle set as that used in Equation (11), and is expressed as

$$\mathbf{U}^{(I)}_{i,\chi} = \begin{cases} \displaystyle\sum_{\substack{\omega(p)=\chi,\, p\in P_j,\, j\in\mathcal{N}_i \\ \|\boldsymbol{x}_{ip}\|<1.5\Delta\bar{x}}} \left(\mathbf{U}^{(0)}_p + [\mathbf{U}']_p[\boldsymbol{\Phi}']^{\mathrm{T}}_{pi}\right) W\left(\|\boldsymbol{x}_{ip}\|, h_i\right) V_p/\alpha'_{i,\chi}, & I = 0, \\ \displaystyle\sum_{\substack{\omega(p)=\chi,\, p\in P_j,\, j\in\mathcal{N}_i \\ \|\boldsymbol{x}_{ip}\|<1.5\Delta\bar{x}}} \mathbf{U}^{(I)}_p W\left(\|\boldsymbol{x}_{ip}\|, h_i\right) V_p/\alpha'_{i,\chi}, & I > 0. \end{cases} \tag{19}$$

where the particle values $\mathbf{U}^{(I)}_p$ are inherited from their corresponding grid cells through the G2P mapping procedure introduced in Section 5.

***Remark* 2.** The reconstruction of the cell-centered value $\mathbf{U}^{(0)}_i$ takes the higher-order particle terms into account. There are also other reconstruction methods developed for strongly compressible flows, such as the advection-equation-based reconstruction method [61]. This approach is typically solved using an upwind scheme associated with the corresponding fluid. Similar to the upwind scheme, the present method also reconstructs the values of newly activated cells from their corresponding particles. Other methods employ non-oscillatory reconstruction schemes, such as WENO [61, 62, 63]. In the PG-DG method, the particles inherit the higher-order terms that are constrained by the slope limiters introduced in Section 4, thereby ensuring reconstruction robustness in a manner similar to that of WENO-based reconstruction methods.

4. Field updates on the grid

After the P2G step, all field values are available at the grid cells, and the grid solver is then invoked to update each DG term $\mathbf{U}^{(I)}_{i,\chi}$. The integral terms in Equation (8) are further discretized using Gaussian quadrature over the cell domain and cell interfaces, yielding

$$\frac{\partial \mathbf{U}^{(I)}_{i,\chi}}{\partial t} = \frac{1}{M_{Ii}}\left[\sum_{s}^{G_1} \mathbf{F}_{is,\chi} \nabla\Phi^{(I)}_{is} w_s V_i - \sum_{j}^{N_i}\sum_{t}^{G_2} \hat{\mathbf{F}}_{ijt,\chi}\Phi^{(I)}_{ijt} \cdot \boldsymbol{n}_j S_j w_t\right], \quad I = 0,1,2,3. \tag{20}$$

The number of face-adjacent neighbors of cell i is $N_i = 4$ and 6 for the Cartesian grid in 2D and 3D space, respectively, and $\boldsymbol{n}_j$ is the interface normal connecting cell j, pointing outwards from cell i. G_1 and G_2 denote the numbers of Gaussian quadrature points in the cell volume Ω_i and on the cell interface shared

with cell j, respectively. The corresponding quadrature points are indexed by the subscripts s (with weights w_s) and t (with weights w_t), respectively. In the present work, only linear test functions are employed for simplicity, and two Gaussian quadrature points are used in each coordinate direction, corresponding to $G_1 = 4$ and $G_2 = 2$ for 2D, $G_1 = 8$ and $G_2 = 4$ for 3D, respectively.

4.1. Limiter for multiphase flows

A proper slope limiter is essential for ensuring the robustness of the DG method, particularly in strongly compressible flows. In the original work of Cockburn [46], a total variation bounded (TVB) limiter was adopted to suppress numerical instability. In the present work, the main framework of the limiter is based on the Barth–Jespersen (B-J) limiter [64], while the mixing characteristics near multiphase interfaces are also taken into account. The component $\tilde{U}_i$ of the limited $\tilde{\mathbf{U}}_i$ is expressed as

$$\tilde{U}_{i,\chi} = U^{(0)}_{i,\chi} + \psi_{i,\chi}[U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}, \tag{21}$$

with ψ standing for the slope limiter, and it should make $\tilde{U}_{i,\chi}$ satisfy

$$U^{(0),\min}_{i,\chi} \le \tilde{U}_{i,\chi} \le U^{(0),\max}_{i,\chi}, \tag{22}$$

with $U^{(0),\max}_{i,\chi} = \max\limits_{i\cup(j\in N_i)} \hat{U}^{(0)}_{ij,\chi}$ and $U^{(0),\min}_{i,\chi} = \min\limits_{i\cup(j\in N_i)} \hat{U}^{(0)}_{ij,\chi}$, where $\hat{U}^{(0)}_{ij,\chi}$ is the averaged value of different phases with their phase fraction $\alpha_{j,\hat{\chi}}$, i.e.,

$$\hat{U}^{(0)}_{ij,\chi} = \sum_{\hat{\chi}} \alpha_{j,\hat{\chi}} U^{(0)}_{ij,\chi\hat{\chi}}, \tag{23}$$

where the component $U^{(0)}_{ij,\chi\hat{\chi}}$ of $\mathbf{U}^{(0)}_{ij,\chi\hat{\chi}}$ is estimated as

$$\mathbf{U}^{(0)}_{ij,\chi\hat{\chi}} = \left(\hat{\rho}^{(0)}_{ij,\hat{\chi}},\ \hat{\rho}^{(0)}_{ij,\hat{\chi}} \boldsymbol{v}^{(0)}_{j,\hat{\chi}},\ \frac{p^{(0)}_{j,\hat{\chi}} + \gamma_\chi p_{\infty,\chi}}{\gamma_\chi - 1} + \frac{1}{2}\hat{\rho}^{(0)}_{ij,\hat{\chi}} \left\| \boldsymbol{v}^{(0)}_{j,\hat{\chi}} \right\|^2 \right), \tag{24}$$

where $\hat{\rho}^{(0)}_{ij,\hat{\chi}}$ is chosen as

$$\hat{\rho}^{(0)}_{ij,\hat{\chi}} = \begin{cases} \rho^{(0)}_{j,\chi}, & \chi = \hat{\chi}, \\ \rho^{(0)}_{i,\chi}, & \chi \neq \hat{\chi}. \end{cases} \tag{25}$$

Then the limiter $\tilde{\psi}_{i,\chi} = \min\limits_{j\in N_i} \tilde{\psi}_{ij,\chi}$ for phase χ can be calculated with $\tilde{\psi}_{ij,\chi}$ expressed by

$$\tilde{\psi}_{ij,\chi} = \begin{cases} g\left(\dfrac{U^{(0),\max}_{i,\chi} - U^{(0)}_{i,\chi}}{[U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}_{ij}} \right), & [U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}_{ij} > 0, \\ g\left(\dfrac{U^{(0),\min}_{i,\chi} - U^{(0)}_{i,\chi}}{[U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}_{ij}} \right), & [U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}_{ij} < 0, \\ 1, & [U']_{i,\chi}[\mathbf{\Phi}']^{\mathrm{T}}_{ij} = 0. \end{cases} \tag{26}$$

Here, the function $g(x)$ is chosen as

$$g(x) = \begin{cases} \min\left(\frac{|x|+x}{|x|+1}, 1 \right), & \alpha_{i,\chi} < 1 \text{ or boundary cells}, \\ \min\left(2x, 1\right), & \alpha_{i,\chi} = 1, \end{cases} \tag{27}$$

which means deploying the van Leer function [65] in the interface region while maintaining high accuracy in the rest of the domains with the Superbee function [66]. For higher robustness, the boundary cells use the van Leer limiter.

To further improve the robustness of the limiter, its positivity is checked at a set of test points, for which the cell vertices are adopted in the present work. These test points constitute the set $\mathcal{T}_{i,\chi}$ for cell i. The

positivity condition must be satisfied at every test point in this set; otherwise, the limiter is set to zero. This can be written as

$$\psi_{i,\chi} = \begin{cases} \tilde{\psi}_{i,\chi}, & \forall t \in \mathcal{T}_{i,\chi},\ p_{\mathrm{EOS}}(\tilde{\mathbf{U}}_{t,\chi}) + \varepsilon p_{\infty,\chi} > 0 \text{ and } \tilde{\rho}_{t,\chi} > 0, \\ 0, & \text{else}, \end{cases} \tag{28}$$

and $\psi_{i,\chi}$ will be used as the final limiter. Here, the coefficient ε is chosen as 0.05 in this study, which allows slight negative pressure in the field to avoid over-limiting the high-order terms.

4.2. Flux mixing treatment

In Part 1, the post-mixing treatment is adopted for weakly compressible flows with a barotropic EOS, where the density of each phase can be uniquely determined from the mixed pressure. However, this treatment cannot be directly extended to strongly compressible flows governed by the stiffened-gas EOS, where the pressure determines only ρe rather than ρ and e separately. An additional thermodynamic mixing rule would therefore be required to construct a unique post-mixed state. Moreover, direct mixing of field variables may introduce additional numerical dissipation near strong discontinuities. For these reasons, the present work adopts flux mixing based on interfacial Riemann problems, rather than directly mixing the field variables after their update.

On the right-hand side of Equation (20), the first term should evaluate a cell-wise volume integral, which is computed using Gaussian quadrature points indexed by s, similar to the CG method. The second term requires calculating the numerical flux across the interface ij between cells i and j, where the flux is computed at the Gaussian quadrature point t. Considering that the neighboring cell j sharing the interface ij with cell i may contain field values of different phases, a flux-mixing approach is proposed in the PG-DG method, i.e.,

$$\hat{\mathbf{F}}_{ijt,\chi} = \sum_{\hat{\chi}} \alpha_{j,\hat{\chi}} \mathbf{F}_{ijt,\chi\hat{\chi}}. \tag{29}$$

As shown in Figure 3, cell i contains the field values of two phases, i.e., $\mathbf{U}_{L,1}$ and $\mathbf{U}_{L,2}$, and interacts with both phases 1 and 2 in the neighboring cell j. The interaction is realized through the numerical flux $\mathbf{F}_{ijt,\chi\hat{\chi}}$, solved by a Riemann problem with the neighboring ghost state $\mathbf{U}^g_{L,\chi\hat{\chi}}$, where the ghost state is estimated using a double-shock Riemann solution.

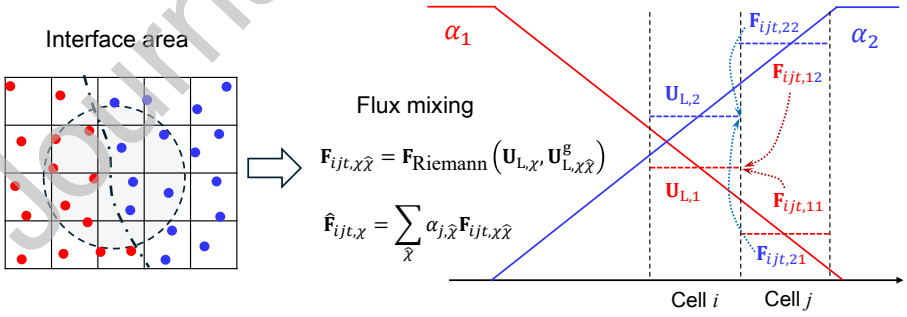


Figure 3. PG-DG treatment of flux mixing across the interface. Cell i contains the field values of two phases, i.e., $\mathbf{U}_{L,1}$ and $\mathbf{U}_{L,2}$,and interacts with both phases 1 and 2 in the neighboring cell j. The interaction is realized through the numerical flux $\mathbf{F}_{ijt,\chi\hat{\chi}}$, which is computed by a Riemann solver using the neighboring ghost state $\mathbf{U}^g_{L,\chi\hat{\chi}}$, which is estimated using a double-shock Riemann solution.

The main task in Equation (29) is to compute $\mathbf{F}_{ijt,\chi\hat{\chi}}$, i.e., the numerical flux between phases χ and $\hat{\chi}$. For single-phase flows, the flux can be computed straightforwardly. However, in strongly compressible multiphase flows, the field values from the neighboring cell cannot be directly used to construct the interfacial flux. The ghost fluid method (GFM) [8] introduces ghost values for different phases, where the original formulation employs entropy extrapolation. This method was later improved by estimating the ghost values through the solution of a Riemann problem [67, 68]. In the present work, the ghost values are obtained using the double-shock Riemann solution proposed in [67].

Given the left state $(\rho_{L,\chi}, u_{L,\chi}, p_{L,\chi})$ and right state $(\rho_{R,\hat{\chi}}, u_{R,\hat{\chi}}, p_{R,\hat{\chi}})$, where L and R denote the values at subscript 'it' and 'jt' at interface 'ijt', we have $u_{L,\chi} = \boldsymbol{v}_{L,\chi} \cdot \boldsymbol{n}_{ij}$ and $u_{R,\hat{\chi}} = \boldsymbol{v}_{R,\hat{\chi}} \cdot \boldsymbol{n}_{ij}$. Then, considering a double shock Riemann problem, we have

$$\begin{cases} u_{L,\chi} - u_{R,\hat{\chi}} = \dfrac{p^* - p_{L,\chi}}{W_{L,\chi}} + \dfrac{p^* - p_{R,\hat{\chi}}}{W_{R,\hat{\chi}}}, \\ W_{L,\chi} = \sqrt{\dfrac{1}{2}\rho_{L,\chi}\left[(\gamma_{L,\chi}+1)p^* + (\gamma_{L,\chi}-1)p_{L,\chi} + 2\gamma_{L,\chi}p_{\infty,L,\chi}\right]}, \\ W_{R,\hat{\chi}} = \sqrt{\dfrac{1}{2}\rho_{R,\hat{\chi}}\left[(\gamma_{R,\hat{\chi}}+1)p^* + (\gamma_{R,\hat{\chi}}-1)p_{R,\hat{\chi}} + 2\gamma_{R,\hat{\chi}}p_{\infty,R,\hat{\chi}}\right]}. \end{cases} \tag{30}$$

For details, readers are referred to [67]. Here, p^* is obtained iteratively using Newton's method, which typically converges within five iterations in the present tests. Once p^* has been obtained, we have

$$\begin{cases} u^* = u_{L,\chi} - \dfrac{p^* - p_{L,\chi}}{W_{L,\chi}} = u_{R,\hat{\chi}} + \dfrac{p^* - p_{R,\hat{\chi}}}{W_{R,\hat{\chi}}}, \\ \rho_L^* = \rho_{L,\chi}\dfrac{(\gamma_{L,\chi}+1)p^* + (\gamma_{L,\chi}-1)p_{L,\chi} + 2\gamma_{L,\chi}p_{\infty,L,\chi}}{(\gamma_{L,\chi}-1)p^* + (\gamma_{L,\chi}+1)p_{L,\chi} + 2\gamma_{L,\chi}p_{\infty,L,\chi}}, \\ \rho_R^* = \rho_{R,\hat{\chi}}\dfrac{(\gamma_{R,\hat{\chi}}+1)p^* + (\gamma_{R,\hat{\chi}}-1)p_{R,\hat{\chi}} + 2\gamma_{R,\hat{\chi}}p_{\infty,R,\hat{\chi}}}{(\gamma_{R,\hat{\chi}}-1)p^* + (\gamma_{R,\hat{\chi}}+1)p_{R,\hat{\chi}} + 2\gamma_{R,\hat{\chi}}p_{\infty,R,\hat{\chi}}}. \end{cases} \tag{31}$$

Then we can reconstruct the ghost values for the left state $\mathbf{U}^g_{L,\chi\hat{\chi}} = \left(\rho^g_{L,\chi\hat{\chi}}, \rho^g_{L,\chi\hat{\chi}}\boldsymbol{v}^g_{L,\chi\hat{\chi}}, E^g_{L,\chi\hat{\chi}}\right)$, where

$$\begin{cases} \rho^g_{L,\chi\hat{\chi}} = \rho^*_L, \\ \boldsymbol{v}^g_{L,\chi\hat{\chi}} = \boldsymbol{v}_{L,\chi} - (\boldsymbol{v}_{L,\chi}\cdot\boldsymbol{n}_{ij})\boldsymbol{n}_{ij} + u^*\boldsymbol{n}_{ij}, \\ E^g_{L,\chi\hat{\chi}} = \dfrac{p^* + \gamma_{L,\chi}p_{\infty,L,\chi}}{\gamma_{L,\chi}-1} + \dfrac{1}{2}\rho^g_{L,\chi\hat{\chi}}\left\|\boldsymbol{v}^g_{L,\chi\hat{\chi}}\right\|^2. \end{cases} \tag{32}$$

The procedure is the same for the derivation of the ghost value for the right state $\mathbf{U}^g_R$. Given the ghost values, then we can use an approximate Riemann solver to derive the flux $\mathbf{F}_{ijt,\chi\hat{\chi}}$, written as

$$\mathbf{F}_{ijt,\chi\hat{\chi}} = \mathbf{F}_{\text{Riemann}}(\mathbf{U}_{L,\chi}, \mathbf{U}^g_{L,\chi\hat{\chi}}), \tag{33}$$

where either the Harten-Lax-van Leer (HLL) [69] or the Harten-Lax-van Leer-Contact (HLLC) [70] Riemann solver is employed in the present work. It is found that the HLLC solver performs well for flows with low density ratios, whereas the HLL solver is recommended for cases requiring additional robustness, such as involving high density ratios. If not specified otherwise, HLLC is deployed in the present study.

It is worth noting that each phase's DG polynomial is defined over the whole cell and all integrals in Equation (20) use the full-cell geometric quadrature regardless of α (the phase-wise fields overlap in interface cells, as in GFM); Furthermore, α does not act as a denominator in the present method. Therefore, a small phase fraction does not directly introduce numerical ill-conditioning through division in the flux-mixing procedure and thus does not affect the robustness of the method.

***Remark* 3.** The flux exchange on the interface between cells i and j can be estimated as follows. For cell i, the numerical flux exchange is

$$\tilde{\mathbf{F}}^i_{ijt} = \sum_{\chi_1}\sum_{\chi_2}\alpha_{i,\chi_1}\left(\alpha_{j,\chi_2}\mathbf{F}_{ijt,\chi_1\chi_2}\right), \tag{34}$$

and for cell j,

$$\tilde{\mathbf{F}}^j_{ijt} = \sum_{\chi_1}\sum_{\chi_2}\alpha_{j,\chi_1}\left(\alpha_{i,\chi_2}\mathbf{F}_{ijt,\chi_1\chi_2}\right), \tag{35}$$

Then the flux difference at the interface between it and jt is

$$\Delta\tilde{\mathbf{F}}_{ijt} = \tilde{\mathbf{F}}^i_{ijt} - \tilde{\mathbf{F}}^j_{ijt} = \sum_{\chi_1}\sum_{\chi_2}\alpha_{i,\chi_1}\alpha_{j,\chi_2}\left(\mathbf{F}_{ijt,\chi_1\chi_2} - \mathbf{F}_{ijt,\chi_2\chi_1}\right). \tag{36}$$

Journal Pre-proof

Journal Pre-proof

This equation shows that the present method ensures an anti-symmetric flux structure by $\mathbf{F}_{ijt,\chi_1\chi_2} - \mathbf{F}_{ijt,\chi_2\chi_1}$. In the single fluid domain, it reduces to $\Delta\tilde{\mathbf{F}}_{ijt} = \sum_{\chi_1}\sum_{\chi_2}\alpha_{i,\chi_1}\alpha_{j,\chi_2}(\mathbf{F}_{ijt} - \mathbf{F}_{ijt}) = 0$. In the interface area, the terms with $\chi_1 = \chi_2$ can also cancel each other perfectly, but for $\chi_1 \neq \chi_2$, the two terms cannot cancel each other, since the flux $\mathbf{F}_{ijt,\chi_1\chi_2}$ is estimated based on the phase χ_1, and $\mathbf{F}_{ijt,\chi_2\chi_1}$ is based on the phase χ_2. Though the star region of the Riemann fan ensures the same values for velocity and pressure, the densities ρ_L^* and ρ_R^* derived in the star region serve as different values for the left and right ghost states. Therefore, the present flux exchange is not strictly conservative in the interface area. Although it is not strictly conservative, the ghost values derived with a double-shock Riemann solver can help mitigate conservation errors as discussed in [71]. In our numerical experiments, the conservation errors will be discussed in Sections 7.3 and 7.4.

Although the flux-mixing approach is developed for the PG-DG method based on the concept of the *phase fraction*, we speculate that it may also be applicable to VOF methods, in which the concept of the *volume fraction* is analogous to that of the *phase fraction*. Therefore, the proposed flux-mixing approach may also be incorporated into conventional VOF methods to achieve sharper interface resolution.

4.3. Time integration

The DG grid cells are advanced in time using the second-order TVD Runge-Kutta scheme [72], which provides a good balance between robustness and computational efficiency, given by

$$\begin{aligned}
\mathbf{U}_{i,\chi}^{(I),n+1,*} &= \psi_{i,\chi}^{(I)}\mathbf{U}_{i,\chi}^{(I),n} + \psi_{i,\chi}^{(I)}\left(\frac{\partial \mathbf{U}_{i,\chi}^{(I)}}{\partial t}\right)^{n}\Delta t, \\
\mathbf{U}_{i,\chi}^{(I),n+1} &= \frac{1}{2}\psi_{i,\chi}^{(I)}\mathbf{U}_{i,\chi}^{(I),n} + \frac{1}{2}\psi_{i,\chi}^{(I)}\mathbf{U}_{i,\chi}^{(I),n+1,*} + \frac{1}{2}\psi_{i,\chi}^{(I)}\left(\frac{\partial \mathbf{U}_{i,\chi}^{(I)}}{\partial t}\right)^{n+1,*}\Delta t,
\end{aligned} \tag{37}$$

where

$$\psi_{i,\chi}^{(I)} = \begin{cases} 1, & I = 0, \\ \psi_{i,\chi}, & I > 0. \end{cases} \tag{38}$$

Here, $\psi_{i,\chi}$ is calculated in each sub-stage of the time integration. In the time integration, the high-order terms of the DG terms are limited by the limiter $\psi_{i,\chi}^{(I)}$ derived in Section 4.1, and this can further improve numerical robustness. $\psi_{i,\chi}^{(I)}$ is equal to 1 in most of the smooth regions, and approaches zero when a discontinuity is present.

The time step can be estimated by

$$\begin{aligned}
&\Delta t = \min(\Delta t_1, \Delta t_2), \\
&\Delta t_1 = \mathrm{CFL}\frac{\Delta \bar{x}}{\max(c_1, ..., c_C) + \|\boldsymbol{v}\|_{\max}}, \quad \Delta t_2 = \mathrm{CFL}\sqrt{\frac{\Delta \bar{x}}{\|\boldsymbol{a}\|_{\max}}},
\end{aligned} \tag{39}$$

where $\|\boldsymbol{v}\|_{\max}$ and $\|\boldsymbol{a}\|_{\max}$ denote the maximum velocity and acceleration in the computational domain, respectively. In DG, the CFL number should satisfy $\mathrm{CFL} < 1/(1+2p')$, where p' is the highest polynomial order in DG test function space. In this work, the value $\mathrm{CFL} = 0.3$ shows high robustness, and this value is used throughout this work.

5. Grid-to-particle mapping

5.1. Field update of particles

For a particle p, its position should be updated using its own transport velocity $\boldsymbol{w}_p$. The calculation of $\boldsymbol{w}_p$ will be introduced in Section 5.2.2. Using the second-order TVD Runge-Kutta scheme, the first sub-step is updated as

$$\begin{aligned}
V_p^{n+1,*} &= V_p^n(1 + \nabla\cdot\boldsymbol{v}_p^n\Delta t), \\
\boldsymbol{x}_p^{n+1,*} &= \boldsymbol{x}_p^n + \boldsymbol{w}_p^n\Delta t.
\end{aligned} \tag{40}$$

In this equation, we also give the update of particle volume V_p^n, which is needed in Section 5.2, for which the divergence $\nabla \cdot \boldsymbol{v}$ is calculated with a second-order finite difference method, considering that the gradient of DG terms is sharply limited near discontinuities. After the particle positions have been determined, the field values at the particle locations are obtained from the corresponding grid cells according to their positions, where the DG field values have already been updated. The particle field values are then expressed as

$$\begin{aligned} \mathbf{U}_p^{(0),n+1,*} &= \mathbf{U}_{i,\chi}^{(0),n+1,*} + [\mathbf{U}']_{i,\chi}^{n+1,*}[\boldsymbol{\Phi}']_{ip}^{\mathrm{T}}, \quad I = 0, \\ \mathbf{U}_p^{(I),n+1,*} &= \mathbf{U}_{i,\chi}^{(I),n+1,*}, \quad I > 0, \end{aligned} \tag{41}$$

where the relations between i and p, p and χ, satisfy $\kappa(p) = i$ (Equation (9)) and $\omega(p) = \chi$ (Equation (10)), respectively. Similarly, in the second sub-time step,

$$\begin{aligned} V_p^{n+1} &= \frac{1}{2}V_p^n + \frac{1}{2}V_p^{n+1,*} + \frac{1}{2}V_p^{n+1,*}\nabla \cdot \boldsymbol{v}_p^{n+1,*}\Delta t, \\ \boldsymbol{x}_p^{n+1} &= \frac{1}{2}\boldsymbol{x}_p^n + \frac{1}{2}\boldsymbol{x}_p^{n+1,*} + \frac{1}{2}\boldsymbol{w}_p^{n+1,*}\Delta t, \end{aligned} \tag{42}$$

and then

$$\begin{aligned} \tilde{\mathbf{U}}_p^{(0),n+1} &= \mathbf{U}_{i,\chi}^{(0),n+1} + [\mathbf{U}']_{i,\chi}^{n+1}[\boldsymbol{\Phi}']_{ip}^{\mathrm{T}}, \quad I = 0, \\ \tilde{\mathbf{U}}_p^{(I),n+1} &= \mathbf{U}_{i,\chi}^{(I),n+1}, \quad I > 0, \end{aligned} \tag{43}$$

Considering that the transported particles may go out of the corresponding cell after the position update, the field value consistency may be lost. Hence, a post-checking is needed to rectify the final field values of the particle p, i.e.,

$$\left(\mathbf{U}_p^{(0),n+1}, \mathbf{U}_p^{(I),n+1}\right) = \begin{cases} \left(\tilde{\mathbf{U}}_p^{(0),n+1}, \tilde{\mathbf{U}}_p^{(I),n+1}\right), & p_{\mathrm{EOS}}(\tilde{\mathbf{U}}_p^{(0),n+1}) + \varepsilon p_{\infty,\chi} > 0 \text{ and } \tilde{\rho}_p^{(0),n+1} > 0, \\ \left(\mathbf{U}_{i,\chi}^{(0),n+1}, \mathbf{0}\right), & \text{else.} \end{cases} \tag{44}$$

This criterion is the same as that used in the limiter estimation in Equation (28), which avoids the spurious pressure field and therefore renders the simulation robust even in high-Mach flows.

***Remark* 4.** The particle volume we use in this study refers to the space occupied by a particle in the computational domain, and thus it is similar to the "volume" used in the volume-of-fluid approach. We emphasize that although strict mass conservation is not ensured, the particle volume can remarkably restrict numerical dissipation in the advection process, because the particle volume, i.e., the occupied space of a particle, will not be lost given that a particle remains in the computational domain, and this is not affected by the calculation in the grid domain. The mass of the particle m_p is calculated with $m_p = \rho_p V_p$, and the total particle mass is $\sum_p m_p$. The grid mass of phase χ can be expressed as $\sum_i \alpha_{i,\chi}\rho_{i,\chi}V_i$. The mass conservation errors will be discussed in Sections 7.3 and 7.4.

5.2. Other issues related to particles

5.2.1. Particle splitting and merging

In strongly compressible flows, particle distribution cannot remain uniform through convection alone, as the velocity divergence may vary over a wide range. Consequently, some cells may lack neighboring particles, leading to undefined values in the calculation of the phase fraction α. To address this issue, particle splitting and merging (PSM) are employed to handle extremely dilute and dense regions in compressible flows. The PSM technique was originally proposed in [33] and has recently been applied to the simulation of strongly compressible flows using meshless methods [36]. In the present work, the PSM technique is adapted to the PG-DG method, inheriting its main features from [36]. The splitting and merging processes are determined by the particle volume, V_p, which is updated as described in Section 5.1. When the particle volume exceeds the threshold $1.6V_0$, the particle (referred to as the mother particle, denoted by the subscript 'm') is split into 2^{Dim} daughter particles (denoted by the subscript 'd'), where V_0 is the initial particle volume. The field

Journal Pre-proof

values of the 2^{Dim} daughter particles are determined as

$$\begin{cases} \boldsymbol{x}_{\mathrm{d}} = \left(x_{\mathrm{m}} \pm \dfrac{\sqrt[Dim]{V_{\mathrm{m}}}}{4}, y_{\mathrm{m}} \pm \dfrac{\sqrt[Dim]{V_{\mathrm{m}}}}{4}, z_{\mathrm{m}} \pm \dfrac{\sqrt[Dim]{V_{\mathrm{m}}}}{4} \right)^{\mathrm{T}}, \\ \mathbf{U}_{\mathrm{d}} = \mathbf{U}_{\mathrm{m}} + [\mathbf{U}']_{\mathrm{m}} [\boldsymbol{\Phi}']^{\mathrm{T}}_{\mathrm{md}}, \\ [\mathbf{U}']_{\mathrm{d}} = [\mathbf{U}']_{\mathrm{m}}, \\ V_{\mathrm{d}} = \dfrac{V_{\mathrm{m}}}{2^{Dim}}. \end{cases} \tag{45}$$

When the volumes of two particles with the same phase are both smaller than the threshold $V_p < \frac{2}{3} V_0$ and their distance is less than $\sqrt[Dim]{\frac{2}{3} V_0}$, then the field values of the merged mother particle are given by

$$\begin{cases} \boldsymbol{x}_{\mathrm{m}} = \dfrac{V_1 \boldsymbol{x}_1 + V_2 \boldsymbol{x}_2}{V_1 + V_2}, \\ \mathbf{U}_{\mathrm{m}} = \dfrac{V_1 \mathbf{U}_1 + V_2 \mathbf{U}_2}{V_1 + V_2}, \\ [\mathbf{U}']_{\mathrm{m}} = \dfrac{V_1 [\mathbf{U}']_1 + V_2 [\mathbf{U}']_2}{V_1 + V_2}, \\ V_{\mathrm{m}} = V_1 + V_2. \end{cases} \tag{46}$$

In both the merging and splitting steps, the volume of particles is conserved, which is consistent with Remark 4. This ensures that an occupied volume of a particle is not lost in these steps. In implementation, the merging pass is executed four times per splitting pass to ensure a better particle distribution. For details, readers are referred to [33, 36].

5.2.2. Particle distribution

Particle distribution plays an important role in the calculation of the phase fraction and should therefore be kept as uniform as possible. The particle distribution should satisfy three requirements, namely uniformity, prevention of wall penetration, and prevention of non-physical phase penetration.

To maintain a uniform particle distribution, the particle shifting technique (PST) used widely in SPH is adopted. Since the neighboring particles of particles near wall boundaries may not occupy a complete kernel support, the boundary integral approach [73] is employed to address this issue. The shifting velocity is then expressed as

$$\delta \boldsymbol{v}_p^{(1)} = -\frac{\Delta t (5 \left\| \boldsymbol{v} \right\|_{\max})^2}{2} \left[\sum_{s \in \mathfrak{N}_p} \left(1 + 0.2 \mathcal{W}_{ps}^4 \right) \nabla W_{ps} V_s + \sum_{b \in \mathfrak{N}'_p} \left(1 + 0.2 \mathcal{W}_{pb}^4 \right) W_{pb} \boldsymbol{n}_b S_b \right], \tag{47}$$

where $\mathcal{W}_{ps} = W(\|\boldsymbol{x}_{ps}\|, h_p) / W(h_p / 1.6, h_p)$. Here, $\mathfrak{N}_p$ denotes the neighboring particle set of particle p, while $\mathfrak{N}'_p$ denotes the set of boundary particles located within the kernel support of particle p, and $S_b = \Delta \bar{x}^{Dim-1}$. The boundary particles are conveniently placed at the centers of the wall-element cells prior to the simulation. To prevent an excessively large shifting velocity, its magnitude is limited by

$$\delta \boldsymbol{v}_p^{(\mathrm{pst})} = \delta \boldsymbol{v}_p^{(1)} \min \left(\left\| \delta \boldsymbol{v}_p^{(1)} \right\|, \frac{\Delta \bar{x}}{10 \Delta t} \right) / \left\| \delta \boldsymbol{v}_p^{(1)} \right\|. \tag{48}$$

To prevent wall penetration, the wall boundary interface on the grid domain can be leveraged to provide a non-penetration force, and this is based on a mapping between cell i and particle p, i.e., $\kappa(p) = i$. For all boundary interfaces b in cell i, forming a set $\mathfrak{N}'_i$, the shifted non-penetration velocity can be calculated with

$$\delta \boldsymbol{v}_p^{(2)} = \frac{\Delta t}{2} \sum_{b \in \mathfrak{N}'_i, \kappa(p) = i} \frac{0.02 c_p^2}{\boldsymbol{x}_{pb} \cdot \boldsymbol{n}_b} \boldsymbol{n}_b, \quad \text{for } |\boldsymbol{x}_{pb} \cdot \boldsymbol{n}_b| < 0.4 \Delta \bar{x}, \tag{49}$$

Journal Pre-proof

where $\boldsymbol{x}_{pb} = \boldsymbol{x}_p - \boldsymbol{x}_b$ the normal direction $\boldsymbol{n}_b$ points outward. Similarly, its amplitude is limited using

$$\delta\boldsymbol{v}_p^{(\mathrm{np})} = \delta\boldsymbol{v}_p^{(2)} \min\left(\left\|\delta\boldsymbol{v}_p^{(2)}\right\|, \frac{\Delta\bar{x}}{2\Delta t}\right) / \left\|\delta\boldsymbol{v}_p^{(2)}\right\|. \tag{50}$$

Thirdly, the PST usually leads to spurious penetration between phases in immiscible flow. Referring to the repulsive force [74], the shifted velocity induced by particles of other phases is expressed as

$$\delta\boldsymbol{v}_p^{(3)} = -\frac{\Delta t}{20} \sum_{\substack{s'\in\mathfrak{N}_p \\ \omega(s')\neq\omega(p)}} \frac{\rho_p c_p^2 + \rho_{s'} c_{s'}^2}{\rho_p + \rho_{s'}} \nabla W(\|\boldsymbol{x}_{ps'}\|, h_p) V_{s'}. \tag{51}$$

Here, $\omega(s') \neq \omega(p)$ denotes the particles of a different phase. Then the shifting is limited using

$$\delta\boldsymbol{v}_p^{(\mathrm{s})} = \delta\boldsymbol{v}_p^{(3)} \min\left(\left\|\delta\boldsymbol{v}_p^{(3)}\right\|, \frac{\Delta\bar{x}}{10\Delta t}\right) / \left\|\delta\boldsymbol{v}_p^{(3)}\right\|. \tag{52}$$

In summary, taking into account uniform distribution, wall penetration and non-physical phase penetration, the final shifted velocity is given by

$$\delta\boldsymbol{v}_p = \delta\boldsymbol{v}_p^{(\mathrm{np})} + \delta\boldsymbol{v}_p^{(\mathrm{pst})} + \delta v_p^{(\mathrm{s})}, \tag{53}$$

which should be added to the real velocity of the particle p, i.e.,

$$\boldsymbol{w}_p = \boldsymbol{v}_p + \delta\boldsymbol{v}_p, \tag{54}$$

and this transport velocity is used in Equations (40) and (42).

6. Solution Procedure

The solution procedure of the PG-DG method for strongly compressible multiphase flows is summarized in Algorithm 1, where the operations performed within a single time step are illustrated. Lines 1-5 present the particle splitting and merging procedures, while line 6 establishes the particle-to-cell mapping. Lines 7-17 present the Runge-Kutta time integration procedure, which forms the core of the algorithm. This stage involves the calculation of the phase fraction, P2G mapping, slope limiting, flux mixing, field-value update, and G2P mapping. The above procedure is repeated until the final simulation time is reached.

***Remark* 5.** The main advantages of PG-DG over conventional DG and MPM can be summarized as follows. Compared to the pure DG method, the present PG-DG provides a new particle-grid approach with the phase fraction, which eliminates the need for an additional phase-advection equation and facilitates the representation of complex interfaces. Compared with conventional MPM, PG-DG employs geometric Gaussian quadrature points rather than particles as integration points. Moreover, the DG fields are retained and evolved directly on the grid, while P2G reconstruction is performed only when the phase status changes. Together with the phase-fraction and flux-mixing treatments, these features enable the particle-grid framework to handle strongly compressible multiphase flows with complex interfaces and strong discontinuities.

7. Numerical Examples

7.1. Two-phase linear advection

The two-phase linear advection case is considered first. The computational domain is $[0, H] \times [0, H]$ with $H = 2$. An air bubble with a diameter of 0.4 is located at $(0.5, 0.5)$ in the helium phase. Two kinds of initial conditions are tested, where initial condition (I) [75] is

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1,\ 1,\ 1,\ 1,\ 1.4,\ 0), & \text{in air,} \\ (0.138,\ 1,\ 1,\ 1,\ 1.67,\ 0), & \text{in helium,} \end{cases}$$

Journal Pre-proof

Algorithm 1 Solution procedure of PG-DG for strongly compressible multiphase flow: one time step.

Split particles if splitting is needed;
Times = 1;
while Times ¡= 4 **do**
    Merge particles if merging is needed; Times ++;
end while
Map particles to grid cells [Eq. (9)], and establish the particle neighbor set $\mathfrak{N}_p$ of particle p, respectively.
for Time integration steps ¡= 2 **do**
    if Time integration steps = 1 **then**
        Determine the phase fraction $\alpha_{i,\chi}$ for different phases [Eq. (13)] and then *phase status* $PS_{i,\chi}$ [Eq. (18)];
        P2G: Map the DG field values carried by particles to the grid [Eq. (19)] according to the $PS_{i,\chi}$;
    end if
    Compute the limiter $\psi_{i,\chi}$ [Eq. (26)].
    Compute the mixed flux $\hat{\mathbf{F}}_{ijt,\chi}$ on the grid [Eq. (29)] and then $\partial \mathbf{U}_{i,\chi}^{(I)}/\partial t$ [Eq. (20)].
    Update the grid field values using Runge-Kutta time integration [Eq. (37)].
    G2P: Update the particle field values [Eqs. (40, 41, 42, 43)] and calculate transport velocity $\boldsymbol{w}_p$ [Eq. (54)].
    Time integration steps ++;
end for

and initial condition (II) is

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1 + 0.02[1 + \cos(\pi r'/0.2)],\ 1,\ 1,\ 1,\ 1.4,\ 0), & \text{in air,} \\ (0.138,\ 1,\ 1,\ 1,\ 1.67,\ 0), & \text{in helium,} \end{cases}$$

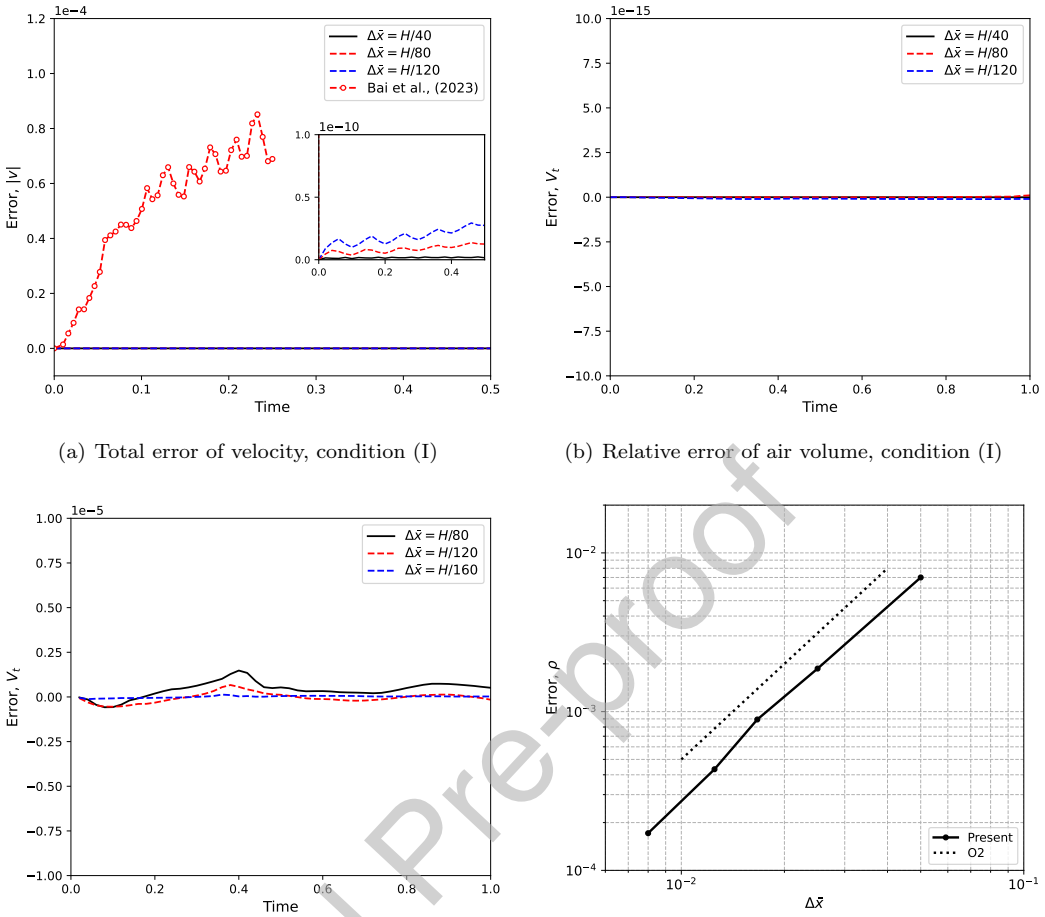
where r' is the distance to the bubble center. Since the initial condition (I) assumes constant field values and error is too small for convergence checking, the initial condition (II) is used for convergence tests. All boundaries are set as the interpolation boundary (the ghost boundary values are interpolated from interior fields).

For initial condition (I), the errors simulated with different resolutions are shown in Figure 4, where the total error of φ is calculated using the particle value φ_p and exact value $\varphi_{p,\text{exact}}$ as $\sum_p |\varphi_p - \varphi_{p,\text{exact}}|$. Figure 4(a) shows that the total velocity error remains at an extremely low level compared with the results reported in [75], where the error is far larger than that of the present method. It is worth noting that the particle volume is used in the calculation of the phase fraction in Equation (11), and therefore attention should be paid to the volume error, as we have discussed in Remark 4. Figure 4(b) shows the relative error of the air volume, which is calculated using $\sum_{p\in Air} |V_p - V_{p,\text{exact}}| / \sum_{p\in Air} V_{p,\text{exact}}$. It shows that the volume error is almost within a negligible 10^{-14}, which can be interpreted as floating-point residuals rather than as a measure of numerical accuracy. The initial condition (II) with variable density is also considered. The relative error of the air volume is still extremely low, as shown in Figure 4(c). The convergence performance under different resolutions is plotted in Figure 4(d), and the results demonstrate second-order convergence, which is consistent with the second-order accuracy of the present DG test function space.

7.2. Shock-bubble interaction

7.2.1. Air/Helium

The interaction between air and helium was first investigated experimentally in [76], and has since been widely used as a benchmark problem for validating numerical algorithms for multiphase flows [8, 77, 18, 54]. The setup of this case is shown in Figure 5, with the corresponding dimensions indicated therein. A shock wave with a Mach number of approximately 1.22 is initially located on the left side and propagates toward



(a) Total error of velocity, condition (I)

(b) Relative error of air volume, condition (I)

(c) Relative error of air volume, condition (II)

(d) Convergence, condition (II)

Figure 4. Two-phase linear advection: (a) total error of velocity and (b) relative error of air volume under different resolutions for initial condition (I); (c) relative error of air volume and (d) convergence of density error under different resolutions for initial condition (II), where the error is calculated using $L_1 = \sum_{p=1}^{N} |u_p - u_{p,\text{exact}}|/N$.

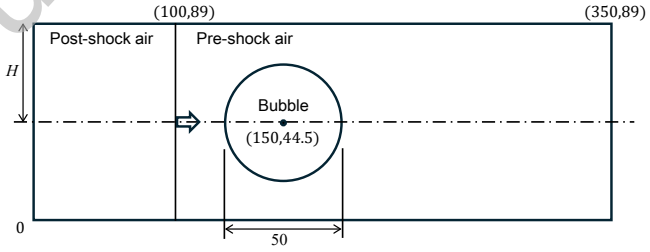


Figure 5. The schematic of shock-bubble interaction and dimensions (unit: mm).

the helium bubble on the right. The initial condition of this case is taken from [54], expressed as

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1.29,\ 0,\ 0,\ 100000,\ 1.4,\ 0), & \text{in pre-shock,} \\ (1.7756,\ 110.62,\ 0,\ 156980,\ 1.4,\ 0), & \text{in post-shock,} \\ (0.2347,\ 0,\ 0,\ 100000,\ 1.648,\ 0), & \text{in bubble,} \end{cases}$$

Journal Pre-proof

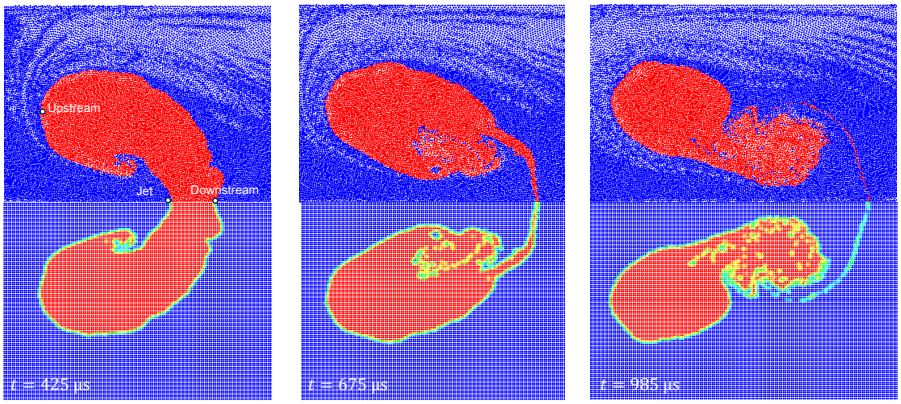


Figure 6. Shock-bubble interaction (Air/He): the phase (particle domain) and phase fraction (grid domain) fields simulated at different times using $\Delta\bar{x} = H/100$.



(a) $t = 427\mu$s



(b) $t = 674\mu$s



(c) $t = 983\mu$s

Figure 7. Shock-bubble interaction (Air/He): Schlieren images of the density field at t = (a) 427, (b) 674 and (c) 983 μs, simulated using $\Delta\bar{x} = H/100$.

Journal Pre-proof

where the units for density, velocity, and pressure are kg/m^3, m/s and Pa, respectively. Considering that the computational domain is symmetric along the centerline, only the upper half is simulated. The reflective boundary condition is imposed on the upper wall and the centerline. The Dirichlet conditions and the interpolation boundary condition (the ghost boundary values are interpolated from interior fields) are imposed on the inlet and outlet, respectively.

The shock reaches the bubble approximately 60 μs after initialization, and hereafter, the time is measured from the instant when the shock hits the bubble. Figure 6 shows the phase fields at different times simulated using $\Delta\bar{x} = H/100$, where the upper and lower halves are from the particle and grid domain, respectively. The interface evolution is clearly captured in both domains, and small-scale vortical structures are even observed, showing a good interface capturing ability and low-dissipation property of the PG-DG method. The Schlieren images of the density field are shown in Figure 7, where the shading function is computed as $\phi = \exp\left(-k' \frac{|\nabla\rho|}{|\nabla\rho|_{\max}}\right)$ following [78]. Here, k' is a user-defined parameter that can be adjusted according to the visualization requirements. The bubble deformation, together with the complex reflected waves generated by the bubble and the walls, can be clearly observed. To quantitatively assess the simulations,

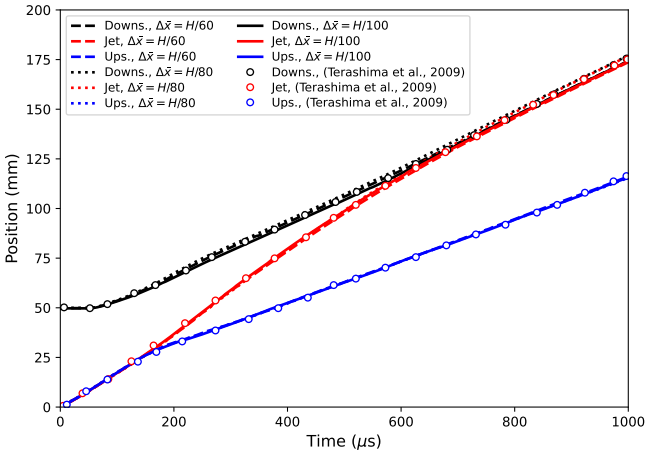


Figure 8. Shock-bubble interaction (Air/He): the displacements of three probes versus time, where the probes are located at upstream, jet and downstream positions, as shown in Figure 6. Various resolutions are deployed for simulation and compared to the reference from [77].

the time histories of three probes located at the upstream, jet, and downstream positions, as shown in Figure 6, are recorded, and the corresponding displacements are presented in Figure 8. Simulations are performed using $\Delta\bar{x} = H/60$, $H/80$, and $H/100$, showing consistent predictions across different resolutions. In addition, the results are compared with the reference solution obtained by the front-tracking method in [77], and excellent agreement is observed. Even with the relatively coarse resolution of $\Delta\bar{x} = H/60$, the present method is able to produce highly accurate predictions. For conventional VOF methods [18, 19, 20], much finer resolutions together with higher-order schemes are generally required to achieve sharp interface resolution. In [31], a cut-cell method is incorporated into the DG framework. Although it is capable of maintaining a sharp interface, the complex interface structures that develop during the later stages of the simulation cannot be adequately resolved, and the cell-cutting procedure makes the algorithm considerably more complicated. This case demonstrates that the present PG-DG is accurate and robust for strongly compressible multiphase flows with complex interfaces.

7.2.2. Air/Helium: 3D cylindrical and spherical cases

The 3D geometric effects of the cylindrical or spherical bubble are experimentally studied in [76]. In the present study, these effects are further verified using the proposed PG-DG method, which can be easily extended to 3D simulations. For the cylindrical case, the cross-section of the 3D case is similar to Figure 5, while a spanwise width of 10 mm is deployed in the z direction. All other initial conditions remain the same as those described in Section 7.2.1. For the spherical case, the experiment conducted in [76] has slight differences, and the following modifications are adopted accordingly: (1) the bubble diameter is 45 mm; (2) the incident shock Mach number is set as approximately 1.25, corresponding to the following initial condition, i.e.,

$$(\rho, v_1, v_2, v_3, p, \gamma, p_\infty) = \begin{cases} (1.179,\ 0,\ 0,\ 0,\ 101325,\ 1.4,\ 0), & \text{in pre-shock,} \\ (1.697,\ 131.6,\ 0,\ 0,\ 167800,\ 1.4,\ 0), & \text{in post-shock,} \\ (0.2347,\ 0,\ 0,\ 0,\ 101325,\ 1.648,\ 0), & \text{in bubble.} \end{cases}$$

Only one quarter of the bubble in the $x - z$ plane is deployed given the axisymmetric property of the case.

Figure 9 displays the case involving a 3D cylindrical bubble, simulated using resolution $\Delta\bar{x} = H/40$. It shows that the deformation profiles are almost the same as those of 2D cases. For the case involving a 3D spherical bubble, simulated using resolution $\Delta\bar{x} = H/40$, the deformation profiles show significant differences from the cylindrical one. As also observed in [76], the spherical bubble is more prone to the Kelvin-Helmholtz instability, leading to larger deformation. The complex interfaces are captured quite well thanks to the phase fraction approach of PG-DG, without requiring extra complicated interface reconstruction methods. A 3D

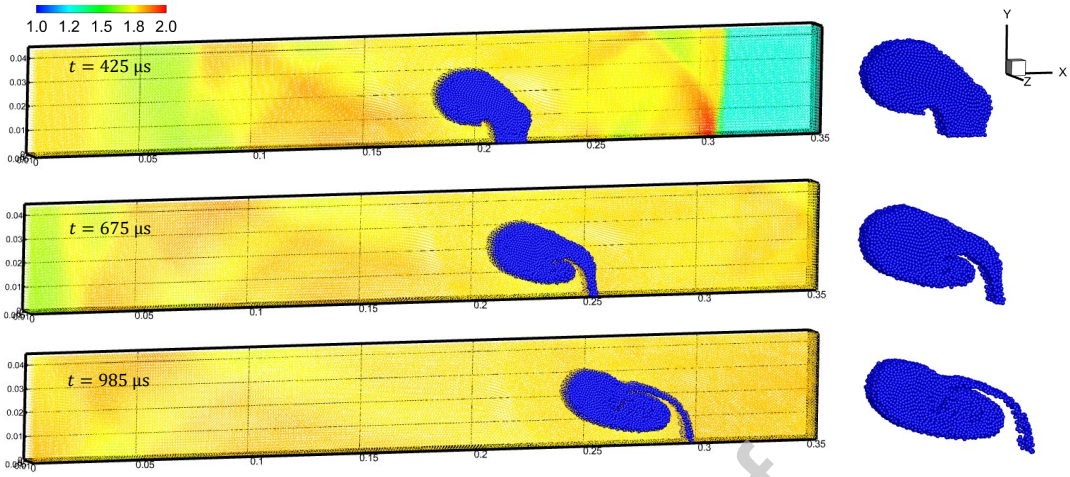


Figure 9. Shock-bubble interaction (Air/He, 3D cylindrical bubble): density fields (left column) and bubble shapes (right column) at different times simulated using resolution $\Delta\bar{x} = H/40$.

simulation of the cylindrical case is performed with a Lagrangian mesh-based CG approach [79], where significant spurious oscillations and numerical dissipation are observed near interfaces, whereas the present PG-DG method avoids these artifacts. A quantitative presentation of the results is shown in Figure 11. This figure plots the displacements of three probes versus time for the cylindrical and spherical bubble cases, where the probes have been illustrated in Figure 6. The results of the cylindrical case align well with the reference from [77], which is simulated using a 2D front-tracking method. For the spherical case, the present results are compared to the experiment conducted in [76], where a reasonable agreement is achieved. Considering that the experiment has measurement errors and the initial condition is not completely the same as the present simulation setup, the slight difference is acceptable. In fact, studies on the shock interaction with 3D spherical bubbles remain relatively scarce, and the present simulation may provide a reference for practitioners. These two 3D cases demonstrate that the present PG-DG method also shows accuracy and robustness in 3D scenarios while maintaining the capability to track complex interfaces.

7.2.3. Air/R22

The setup of the Air/R22 interaction is identical to that of the previous Air/He case, except that the helium bubble is replaced by an R22 bubble. Since R22 is much denser than helium, a significantly different interaction behavior is observed. This case has been extensively investigated in [80, 81, 82, 20]. The initial conditions are taken from [20] and are given as

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1.225,\ 0,\ 0,\ 101325,\ 1.4,\ 0), & \text{in pre-shock,} \\ (1.686,\ 113.5,\ 0,\ 159000,\ 1.4,\ 0), & \text{in post-shock,} \\ (3.863,\ 0,\ 0,\ 101325,\ 1.249,\ 0), & \text{in bubble,} \end{cases}$$

where the units for density, velocity, and pressure are kg/m^3, m/s and Pa, respectively.

The phase (particle domain) and phase fraction (grid domain) fields at different times, simulated with a resolution of $\Delta\bar{x} = H/160$, are presented in Figure 12. Owing to the higher acoustic impedance of the R22 bubble, its deformation differs significantly from that observed in the previous Air/He interaction case. Rich vortical structures develop along the interface, while the particle distribution remains uniform throughout the simulation owing to the PSM and PST treatments. A clearer depiction of the Schlieren images of the density field at different times is displayed in Figure 13, where bubble deformation and complicated shock waves are vividly captured. A quantitative comparison between the present results and the reference results from [80] is presented in Figure 14. The results agree well with the reference solution, demonstrating the high accuracy of the present method.

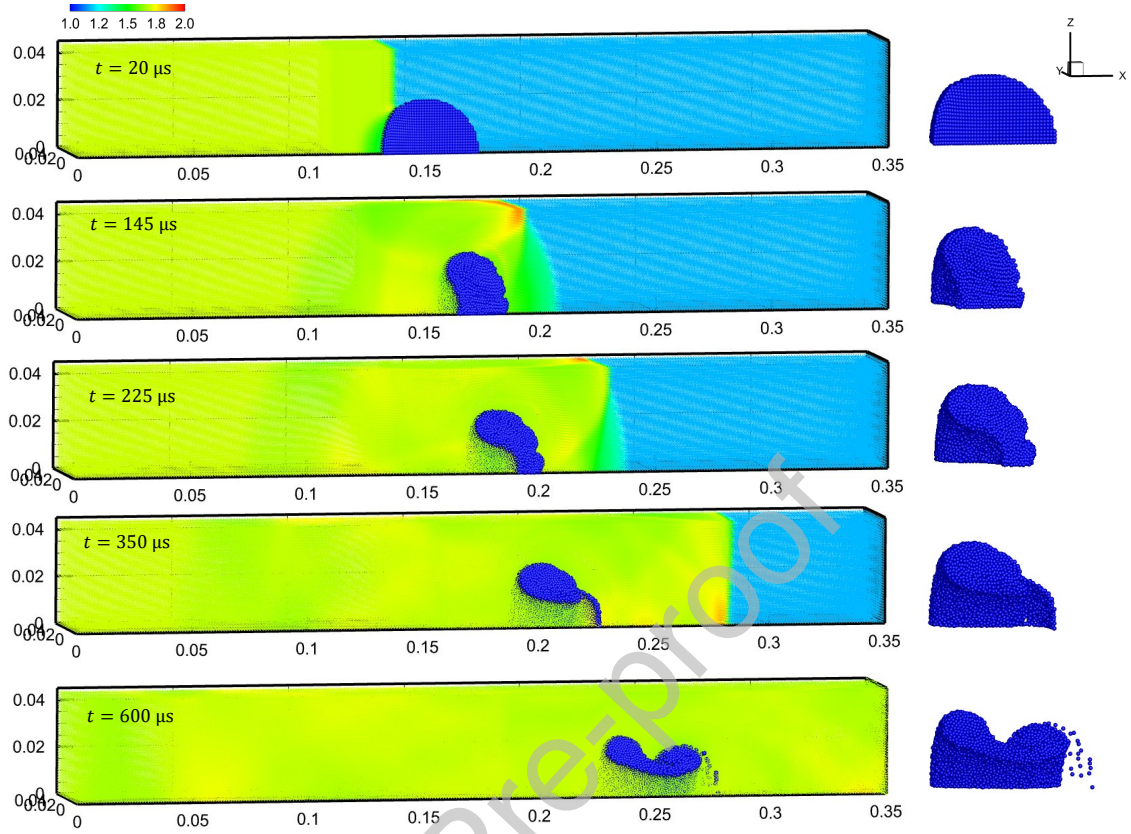
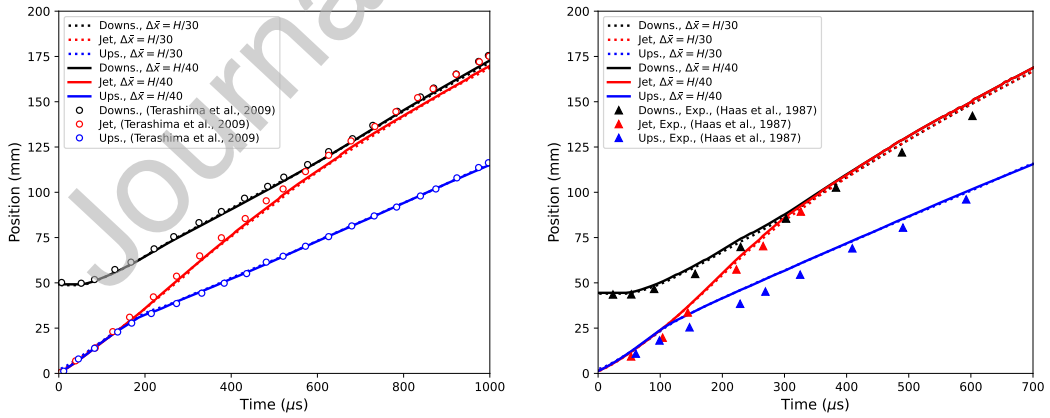


Figure 10. Shock-bubble interaction (Air/He, 3D spherical bubble): density fields (left column) and bubble shapes (right column) at different times simulated using resolution $\Delta\bar{x} = H/40$.



(a) Cylindrical bubble

(b) Spherical bubble

Figure 11. Shock-bubble interaction (Air/He, 3D): the displacements of three probes versus time for the cylindrical and spherical bubble cases, where the probes are located at upstream, jet and downstream positions, as shown in Figure 6. The results are compared to the 2D simulation results from [77] and 3D experimental results from [76].

Journal Pre-proof

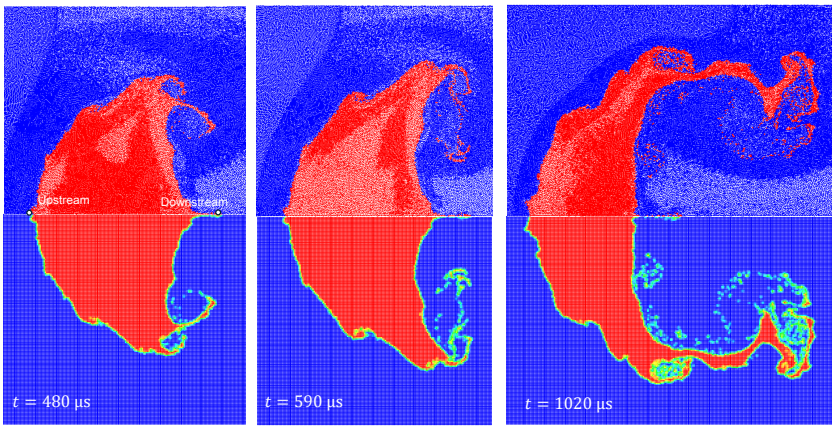
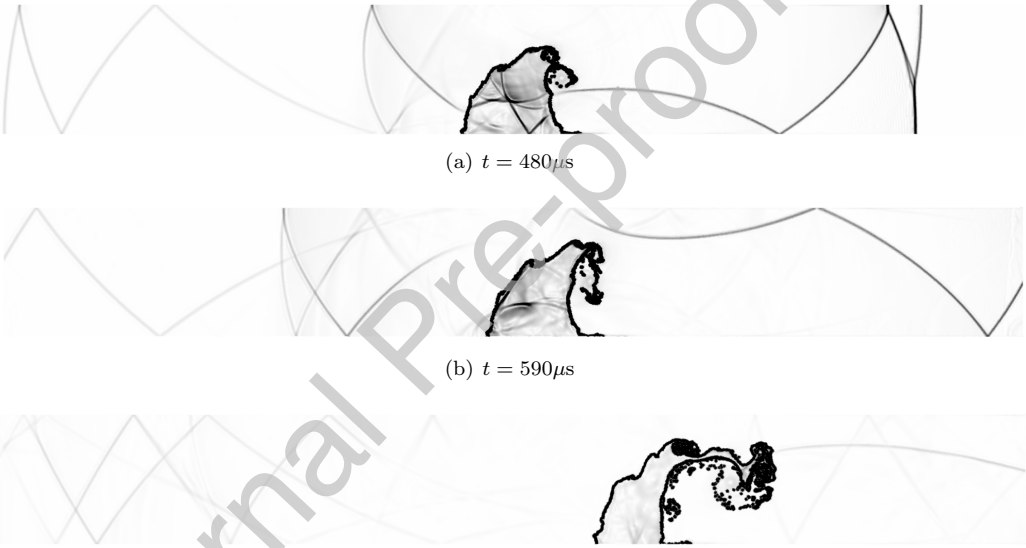


Figure 12. Shock-bubble interaction (Air/R22): the phase (particle domain) and phase fraction (grid domain) fields simulated at different times using $\Delta\bar{x} = H/160$.



(a) $t = 480\mu$s

(b) $t = 590\mu$s

(c) $t = 1020\mu$s

Figure 13. Shock-bubble interaction (Air/R22): Schlieren images of the density field at t = (a) 480, (b) 590 and (c) 1020 μs, simulated using $\Delta\bar{x} = H/160$.

Journal Pre-proof

7.3. Underwater explosion

In this section, we consider an explosion case underwater generated by a highly compressed air bubble. This case involves both highly compressible flows and a large density ratio, which is challenging for most solvers. The computational domain is $[0, 4] \times [0, 3]\text{m}^2$ (Length $L = 4$ m), where the water is located in $[0, 4] \times [0, 1.5]\text{m}^2$, and the ambient air is in the upper part. A highly compressed air bubble of radius 0.12 m is located at $(2, 1.2)$ m. The initial field can be expressed as

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1250,\ 0,\ 0,\ 1 \times 10^9,\ 1.4,\ 0), & \text{in compressed bubble,} \\ (1000,\ 0,\ 0,\ 1.01325 \times 10^5,\ 4.4,\ 6 \times 10^8), & \text{in water,} \\ (1.225,\ 0,\ 0,\ 1.01325 \times 10^5,\ 1.4,\ 0), & \text{in ambient air,} \end{cases}$$

where the units for density, velocity, and pressure are kg/m^3, m/s and Pa, respectively. All walls are assigned reflective boundary conditions. Considering the high density ratio of this case, the HLL Riemann solver is deployed.

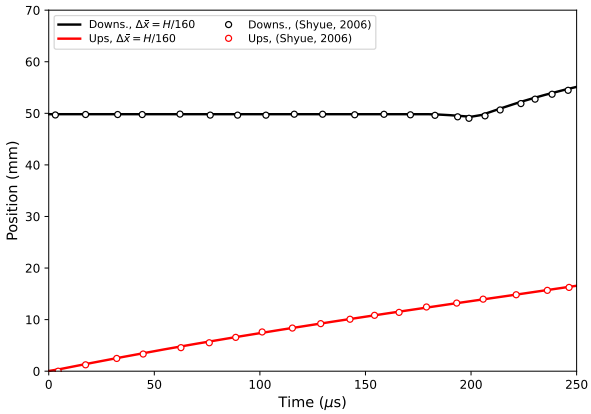
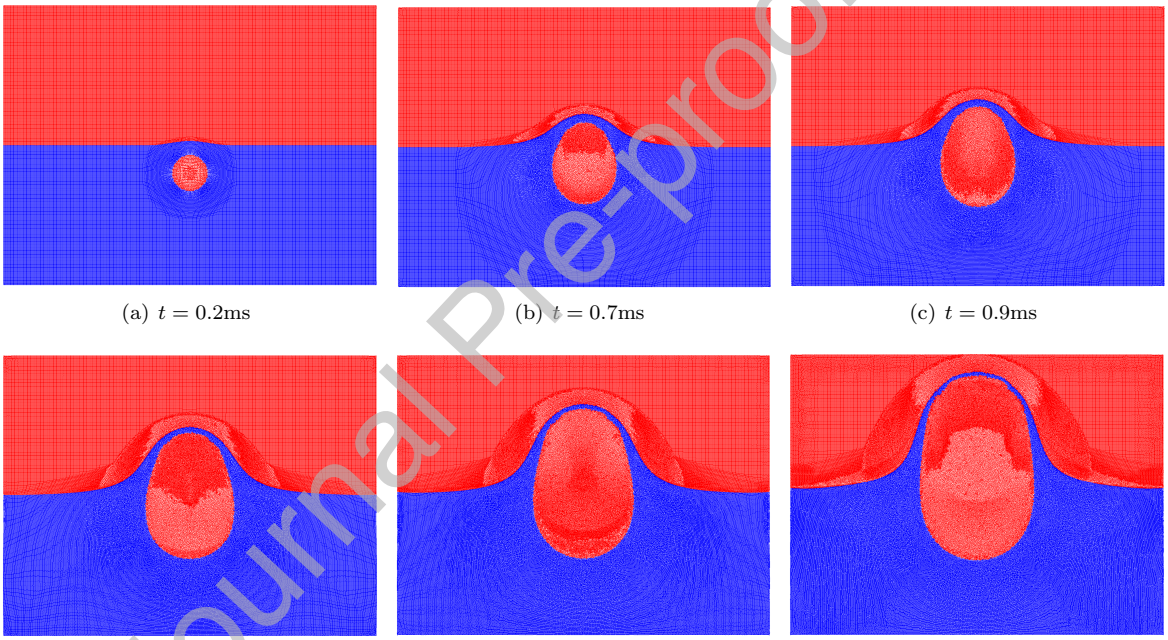


Figure 14. Shock-bubble interaction (Air/R22): the displacements of two probes located at upstream and downstream positions, as shown in Figure 12, where the results are compared to the reference from [80].



(a) $t = 0.2$ms (b) $t = 0.7$ms (c) $t = 0.9$ms

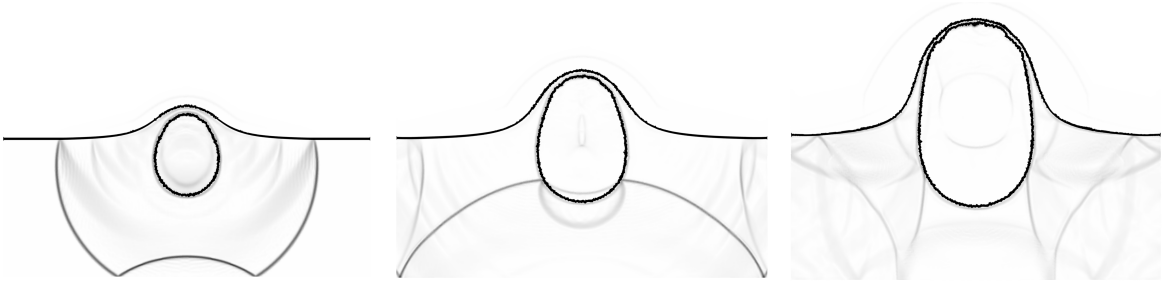
(d) $t = 1.2$ms (e) $t = 1.5$ms (f) $t = 1.9$ms

Figure 15. Underwater explosion: particle phase at different times, simulated using resolution $\Delta\bar{x} = L/400$.

Figure 15 presents the particle phase field of water and air, simulated with a resolution of $\Delta\bar{x} = L/400$, illustrating the deformation process of the highly compressed bubble. Owing to the employed PSM and PST, the particle distribution remains uniform throughout the simulation, thereby improving the numerical robustness. The density Schlieren images at different times are exhibited in Figure 16 at a low resolution $\Delta\bar{x} = L/400$. One can observe that the interface between water and air is sharply captured. Note that, at such a coarse resolution of $\Delta\bar{x} = L/400$, conventional VOF methods generally require specialized interface-reconstruction techniques together with very high-order schemes to maintain a sharp interface. As reported in [19], even with a resolution of $\Delta\bar{x} = L/600$ and a fifth-order WENO scheme, the interface remains noticeably diffusive. In contrast, the present method is able to preserve a sharp interface at a much lower resolution while maintaining a non-oscillatory solution. A quantitative comparison between the present results and those reported in previous studies is shown in Figure 17. At $t = 0.2$ ms, the results obtained with three

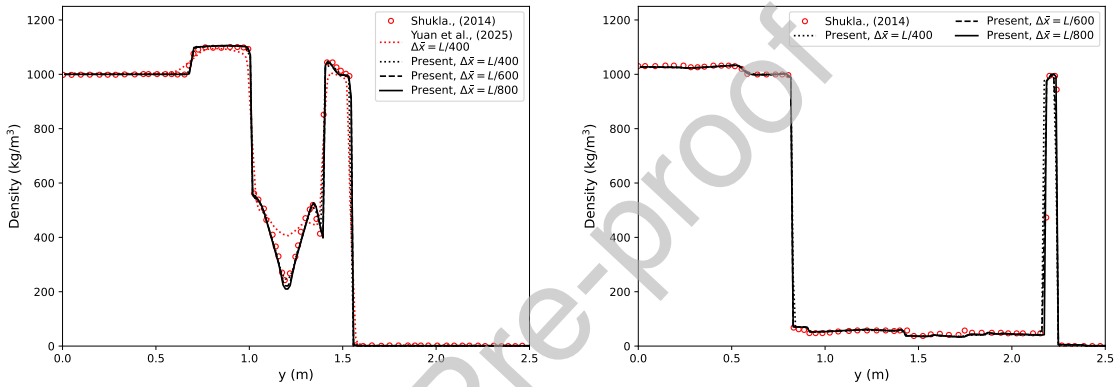
Journal Pre-proof

Journal Pre-proof



(a) $t = 0.7$ms (b) $t = 1.2$ms (c) $t = 1.9$ms

Figure 16. Underwater explosion: Schlieren images of the density field at $t =$ (a) 0.7, (b) 1.2 and (c) 1.9 ms, simulated using resolution $\Delta\bar{x} = L/400$.

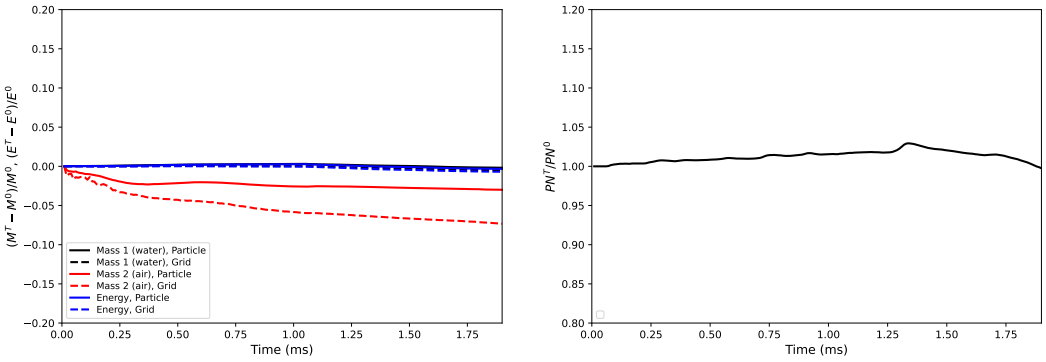


(a) $t = 0.2$ ms (b) $t = 1.2$ ms

Figure 17. Underwater explosion: the density variations along the line $x = 2$ m at times $t = 0.2$ and 1.2 ms, simulated using different resolutions, and comparisons to reference results in [36] and [83].

different resolutions show good consistency and agree well with the reference solution in [83], which was obtained using a fine resolution. Even with the relatively coarse resolution of $\Delta\bar{x} = L/400$, the present result remains close to the reference around $y = 1.2$ m. In contrast, the pure SPH method reported in [36], using the same resolution of $\Delta\bar{x} = L/400$, still deviates significantly from the reference solution, indicating much stronger numerical dissipation. At $t = 1.2$ ms, the present results continue to agree well with the reference solution in [83]. It is worth noting that, around $y = 1.2$ m, conventional VOF methods usually suffer from a loss of the density peak due to excessive numerical dissipation, unless very high-order schemes together with specially designed interface-reconstruction techniques are employed [19]. In comparison, the proposed PG-DG method is able to preserve a sharp interface with low numerical dissipation through the proposed phase fraction and flux-mixing approaches.

As indicated in Remark 3, the conservation properties are not strictly satisfied across interfaces. To further quantitatively investigate the statistics, the histories of the phase-wise mass and total energy with $\Delta\bar{x} = L/400$ are plotted in Figure 18(a). In this figure, we report the errors calculated as $(M^T - M^0)/M^0$ and $(E^T - E^0)/E^0$ in both particle and grid domains, where M^T (initial value, M^0) and E^T (initial value, E^0) denote the mass and total energy, respectively. The errors deviate from zero in both the particle and grid domains, with the maximum relative mass error for the air phase reaching approximately 7–8% in the grid domain, primarily due to the non-conservative interface treatment. The air phase exhibits a larger relative mass error than the water phase. Since the initial air mass is much smaller ($M^0 = 5956.6$ kg for water, $M^0 = 62.3$ kg for air), its normalized conservation error is more sensitive to absolute mass variations. Figure 18(b) exhibits the total particle number PN^T relative to its initial value PN^0, and one can observe that PN^T/PN^0 is close to 1. Since the initial particle number PN^0 is equal to the cell number, the fact



(a) Conservation histories (b) Particle number history

Figure 18. Underwater explosion: (a) the relative errors of total mass of two phases M^T (initial value, M^0) and total energy E^T (initial value, E^0) for both grid and particle domains, with errors calculated using $(M^T - M^0)/M^0$ and $(E^T - E^0)/E^0$. In this case, $M^0 = 5956.6$ kg for water, $M^0 = 62.3$ kg for air; (b) the total particle number PN^T relative to its initial value PN^0.

that PN^T/PN^0 remains close to unity indicates that the method maintains approximately one particle per cell on average throughout the simulation, despite particle splitting and merging.

Table 2: Computational cost of underwater explosion at $\Delta\bar{x} = L/400$, where wall time is that required to reach a simulation time of 1.9 ms. The computation is performed on two AMD EPYC 75F3 32-core processors.

Cell number	Particle number	Wall time (s)
120000	119710 $\sim$ 123500	832

7.4. Triple-point problem

7.4.1. Two-phase case

The triple-point problem is widely used to evaluate the robustness and numerical dissipation of a solver in handling complex flow structures near the triple point and material interfaces. This problem has been extensively simulated using ALE methods [84, 85], pure grid-based methods [86, 87], and particle-based methods [36, 42]. For ALE-based methods, the rich vortical structures cannot be adequately resolved. Pure Eulerian grid-based methods, on the other hand, rely on cell-cutting techniques or high-order diffuse-interface reconstruction schemes to preserve interface sharpness. These difficulties can be effectively avoided with the proposed PG-DG method. The setup of this case is shown in Figure 19, and the initial condition of this case is expressed as

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1,\ 0,\ 0,\ 0.1,\ 1.4,\ 0), & \text{in patch 1,} \\ (0.125,\ 0,\ 0,\ 0.1,\ 1.5,\ 0), & \text{in patch 2,} \\ (1,\ 0,\ 0,\ 1,\ 1.5,\ 0), & \text{in patch 3.} \end{cases}$$

The reflective boundary condition is imposed on all the walls.

Figure 20 presents the particle density fields at $t = 5$, simulated with resolutions of $\Delta\bar{x} = H/150$ and $H/300$. Sharp shock waves are clearly resolved even at the relatively coarse resolution of $\Delta\bar{x} = H/150$. With the finer resolution of $\Delta\bar{x} = H/300$, the shock waves become sharper and more vortical structures are resolved. In [42], this case is simulated using MPM enhanced with reproducing kernel functions, with a particle resolution of $\Delta\bar{x} = H/240$. However, the reported results hardly resolve sharp shock waves, indicating excessive numerical dissipation. In contrast, the present method is able to sharply capture field discontinuities even with the coarser particle resolution of $\Delta\bar{x} = H/150$. This capability is a notable advantage and, to the best of our knowledge, has not been reported for existing particle-grid methods.

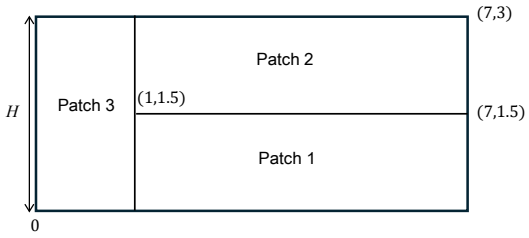


Figure 19. Schematic of the triple-point problem and its dimensions.

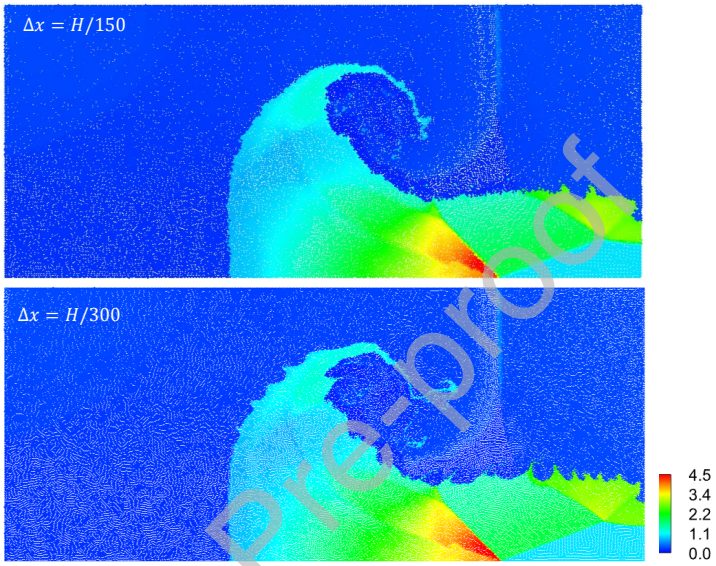


Figure 20. Triple-point problem (two-phase): particle density fields at $t = 5$, simulated using resolutions $\Delta\bar{x} = H/150$ and $H/300$.

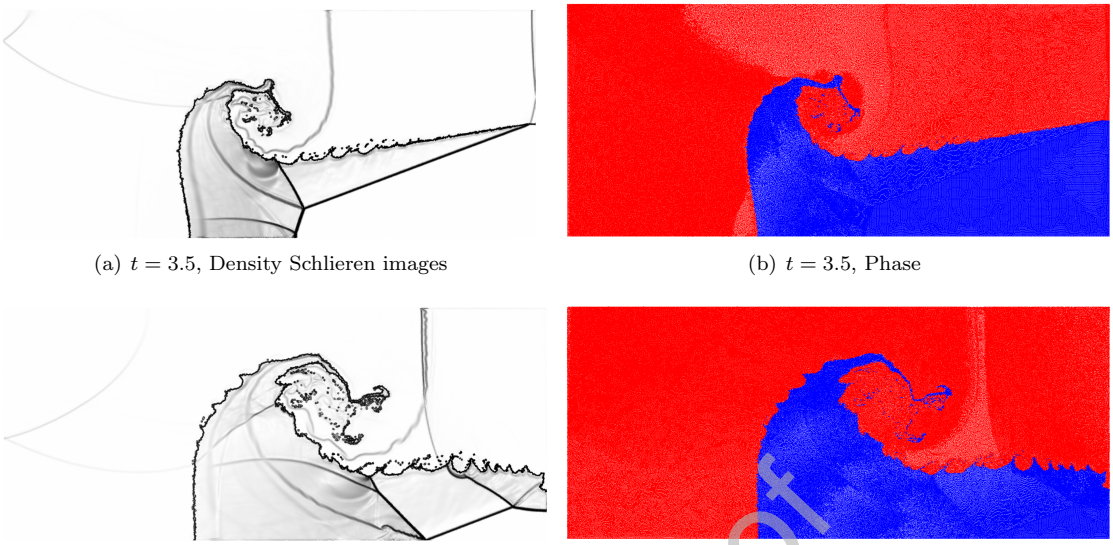
Figure 21 presents the density Schlieren images and particle phase fields at $t = 3.5$ and 5, simulated with a resolution of $\Delta\bar{x} = H/300$. Rich vortical structures can be observed near the intersection point of three patches. In addition, vortical structures develop along the interfaces between different phases. In [36], a pure SPH method is employed to simulate this problem. Even with a finer resolution of $\Delta\bar{x} = H/450$, interfacial vortical structures cannot be resolved, demonstrating significantly lower numerical dissipation of the present method than SPH. An ALE-based DG method proposed in [88] is used to simulate this case, but it shows large numerical dissipation and cannot resolve small structures even when the adaptive mesh is deployed.

This is a closed system case, which is suitable for quantitatively investigating conservation property. The histories of the phase-wise mass and total energy with $\Delta\bar{x} = H/300$ are plotted in Figure 22(a), where the errors are calculated as $(M^T - M^0)/M^0$ and $(E^T - E^0)/E^0$ in both particle and grid domains. The errors deviate from zero, with the maximum relative mass error occurring in the particle domain and remaining within 5% throughout the simulation. The total particle number PN^T relative to its initial value PN^0 is displayed in Figure 22(b), where one can observe that the particle number ratio remains close to 1. Since the initial particle number PN^0 is equal to the cell number, the fact that PN^T/PN^0 remains close to unity indicates that the method maintains approximately one particle per cell on average throughout the simulation, despite particle splitting and merging.

7.4.2. Three-phase case

This section demonstrates that the proposed PG-DG method can readily handle flows involving more than two phases. If the material in Patch 3 is replaced by a fluid with $\gamma = 1.6$ and $p_\infty = 0.1$, the problem

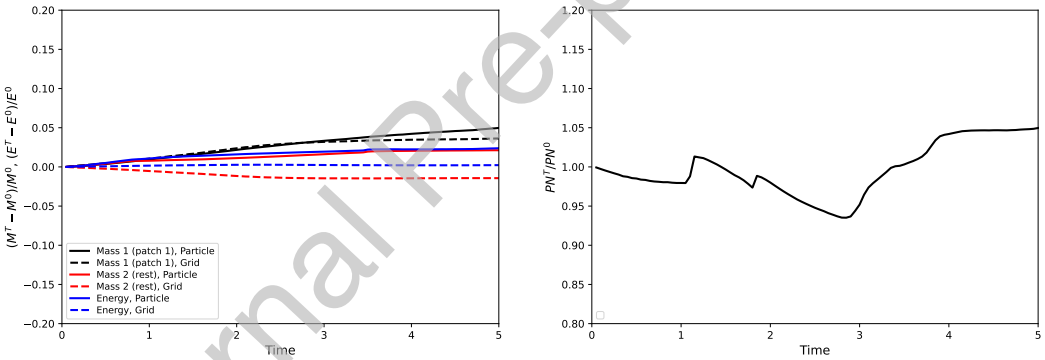
Journal Pre-proof



(a) $t = 3.5$, Density Schlieren images (b) $t = 3.5$, Phase

(c) $t = 5$, Density Schlieren images (d) $t = 5$, Phase

Figure 21. Triple-point problem (two-phase): density Schlieren images and particle phase fields at $t = 3.5$ and 5, simulated using resolution $\Delta\bar{x} = H/300$.



(a) Conservation histories (b) Particle number history

Figure 22. Triple-point problem (two-phase): (a) the relative errors of total mass of two phases M^T (initial value, M^0) and total energy E^T (initial value, E^0) for both grid and particle domains, with errors calculated using $(M^T - M^0)/M^0$ and $(E^T - E^0)/E^0$. In this case, $M^0 = 9.02$ for patch 1, $M^0 = 4.12$ for the rest; (b) the total particle number PN^T relative to its initial value PN^0.

becomes a three-phase triple-point problem. The initial conditions are given as

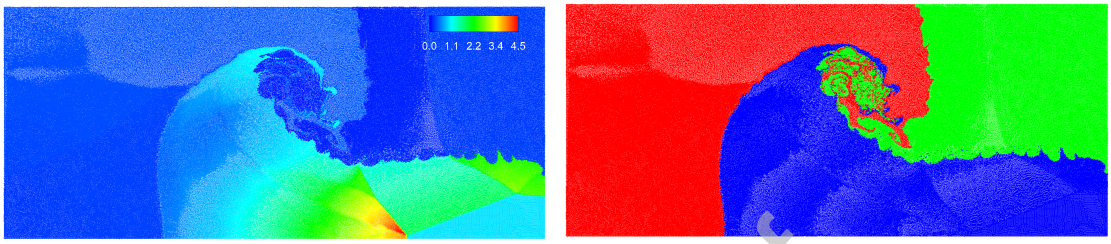
$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1,\ 0,\ 0,\ 0.1,\ 1.4,\ 0), & \text{in patch 1,} \\ (0.125,\ 0,\ 0,\ 0.1,\ 1.5,\ 0), & \text{in patch 2,} \\ (1,\ 0,\ 0,\ 1,\ 1.6,\ 0.1), & \text{in patch 3.} \end{cases}$$

If this problem is simulated using conventional cut-cell methods [32], the treatment of the three-phase interface becomes considerably more complicated. Thanks to the flux-mixing feature of the present method, the flow even with more than two phases can be treated easily in a unified way. Figure 23 displays the particle density and phase fields at $t = 5$, simulated using resolution $\Delta\bar{x} = H/300$. The overall flow pattern is similar to that of the two-phase case, and the vortical structures near the material interfaces are still well resolved, as clearly illustrated by the phase field. A slight difference from the two-phase case is observed because the fluid in Patch 3 has a larger acoustic impedance due to its EOS. The three-phase triple-point

Journal Pre-proof

Table 3: Computational cost of triple-point problem (two-phase) at $\Delta\bar{x} = H/300$, 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.

Cell number	Particle number	Wall time (s)
210000	196720 ∼ 220520	3556



(a) Density (b) Phase

Figure 23. Triple-point problem (three-phase): particle density and phase fields at $t = 5$, simulated using resolution $\Delta\bar{x} = H/300$.

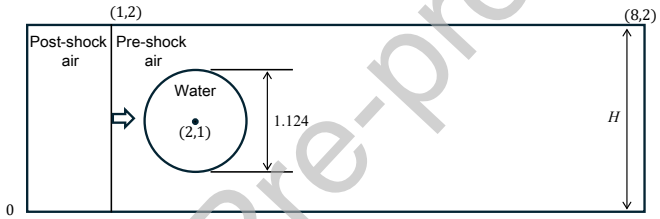


Figure 24. Schematic of Mach 6 air shock hitting water column and its dimensions.

case further demonstrates that the proposed PG-DG method can handle multiple phases in a unified manner while maintaining low numerical dissipation and robustness. This capability makes the PG-DG method well suited for the simulation of complex multi-component flows.

7.5. Mach-6 air shock hitting water column

In this case, a Mach-6 air shock wave impinges on the water column located on the right, as shown in Figure 24. This is a challenging case for testing the robustness of a method for strongly compressible multiphase flows, since this case involves complicated strong shock patterns, large density ratio and water droplets detaching from the column, and each of these could pose a challenge to the numerical stability. The initial condition is taken from [89], written as

$$(\rho, v_1, v_2, p, \gamma, p_\infty) = \begin{cases} (1,\ 0,\ 0,\ 1.013,\ 1.4,\ 0), & \text{in pre-shock,} \\ (5.268,\ 5.790,\ 0,\ 42.39,\ 1.4,\ 0), & \text{in post-shock,} \\ (1000,\ 0,\ 0,\ 1.013,\ 4.4,\ 6000), & \text{in water,} \end{cases}$$

The reflective boundary condition is imposed on the upper and lower walls, and the interpolation boundary condition (the ghost boundary values are interpolated from interior fields) is used on the inlet and outlet. Considering the high density ratio of this case, the HLL Riemann solver is deployed.

Figure 25 presents the particle density fields at different times, simulated with a resolution of $\Delta\bar{x} = H/400$. Complex shock patterns develop when the shock impinges on the water column and reflects from the wall, where the discontinuities are captured sharply without introducing noticeable spurious oscillations. The deformation of the water column is successfully resolved, and the sharp water-air interface is well preserved. In contrast, even a finer resolution of $\Delta\bar{x} = H/500$ produces relatively thick interfaces in [89]. It can be

Journal Pre-proof

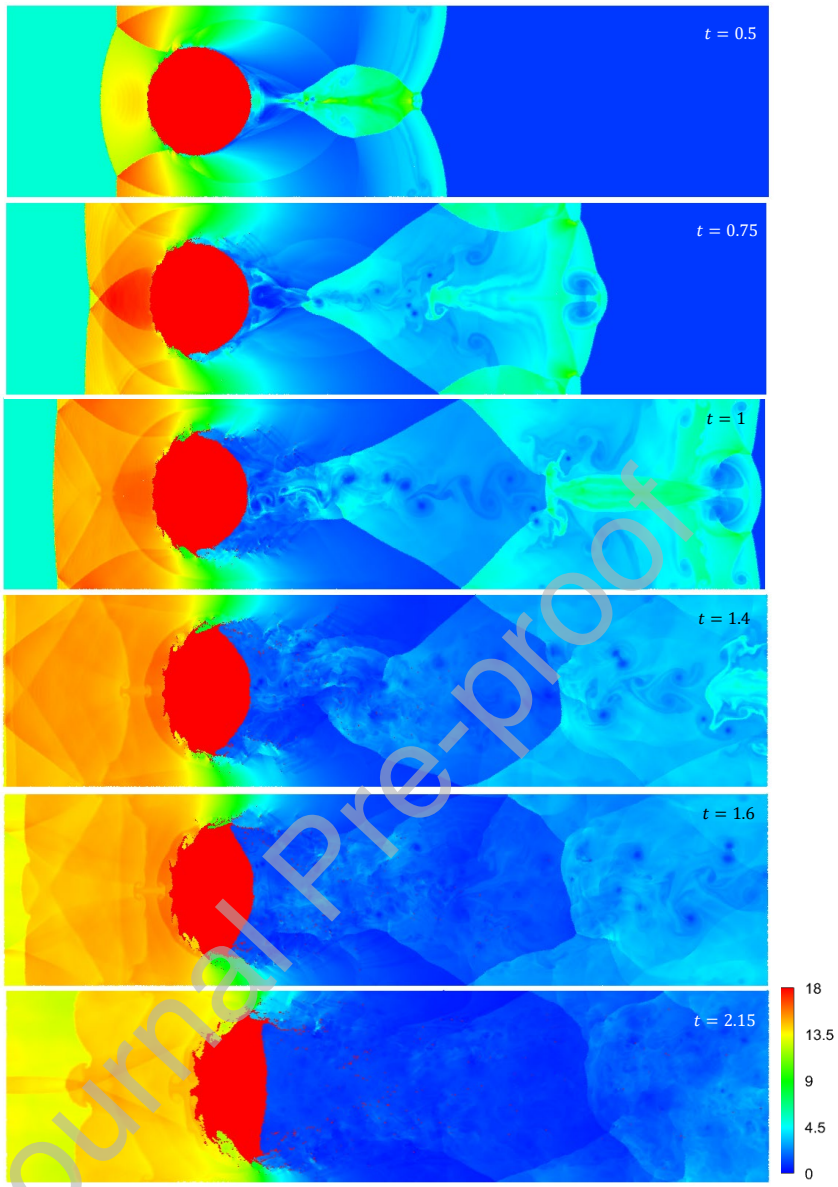


Figure 25. Mach-6 air shock hitting water column: the particle density fields of air at different times, simulated using resolution $\Delta\bar{x} = H/400$.

observed that some water droplets detach from the column when the shock impinges on it, and these detached droplets may affect the downstream flow field, which may explain the slight differences between the present results and those reported in [89]. In fact, the ability to capture detached droplets is a desirable feature, as it qualitatively resembles the actual physical phenomenon. The VOF approach usually predicts continuous water structures behind the water column during the droplet breakup process [89, 19, 20]. This case further demonstrates strong robustness of the PG-DG method for high-Mach-number and high-density-ratio flows with complex interfaces.

8. Concluding Remarks

This study extends the PG-DG method from weakly compressible flows to strongly compressible flows. In the PG-DG method, both particles and the computational grid are employed, where particles constitute

Journal Pre-proof

the primary representation of the flow, while the grid serves to solve the governing equations. Particles of different sets are used to represent different fluid phases, carrying all the field variables, whereas the governing equations are solved on the computational grid to update the particle field values. The method achieves a one-particle-per-cell resolution. The P2G step is performed according to changes in the phase status of each grid cell, which is determined by the phase fraction. The G2P mapping is naturally implemented using the DG test functions. A flux-mixing method is specifically designed for strongly compressible multiphase flows, enabling a unified treatment of flows involving more than two phases while preserving sharp interfaces. A limiter is correspondingly developed for the flux-mixing method. Uniform particle distribution is maintained using the particle splitting and merging (PSM) and particle shifting technique (PST) approaches. The entire algorithm is explicit and is therefore fully compatible with parallel implementation.

The PG-DG method is validated by a set of challenging benchmark cases. High accuracy and robustness are displayed across all the cases from low to high Mach numbers. Even in the cases of large density ratios, e.g., underwater explosion, the PG-DG method can almost approach the converged solution with a relatively low resolution. In addition, the PG-DG method can capture interfaces sharply. To the authors' best knowledge, the accuracy of the present method is much higher than that of most existing particle-based methods, such as SPH or MPM. Regarding limitations, the present PG-DG method focuses on the inviscid Euler equations with a stiffened-gas EOS, and it is established on a Cartesian background grid with second-order accuracy. In addition, strict conservation at material interfaces is not achieved. Future follow-up work should aim to solve these problems and extend it to multiphysics simulation such as fluid-structure interaction and magneto-hydrodynamic flows.

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

To facilitate reproducibility, source codes for a set of representative numerical examples are made publicly available at https://github.com/gaotr3306/PGDG_p2. The data that support the findings of this study are available on request from the corresponding author.

References

[1] E. Lauer, X. Hu, S. Hickel, N. A. Adams, Numerical modelling and investigation of symmetric and asymmetric cavitation bubble dynamics, Computers & Fluids 69 (2012) 1–19.

[2] J. A. McFarland, M. Hagenmaier, Computational study of shock-driven multiphase mixing in scramjet conditions, AIAA Journal 56 (10) (2018) 4004–4015.

[3] F. A. Costanzo, Underwater explosion phenomena and shock physics, in: Structural Dynamics, Volume 3: Proceedings of the 28th IMAC, A Conference on Structural Dynamics, 2010, Springer, 2011, pp. 917–938.

[4] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational physics 39 (1) (1981) 201–225.

[5] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, Journal of Computational Physics 152 (2) (1999) 423–456.

[6] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (2) (1998) 112–152.

[7] J. E. Pilliod Jr, E. G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2) (2004) 465–502.

[8] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of computational physics 152 (2) (1999) 457–492.

[9] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, Journal of computational physics 162 (2) (2000) 301–337.

[10] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Vol. 153 of Applied Mathematical Sciences, Springer, New York, 2003.

[11] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, Journal of Computational physics 169 (2) (2001) 463–502.

[12] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, Journal of Computational Physics 353 (2018) 82–109.

[13] S. Mirjalili, S. S. Jain, M. Dodd, Interface-capturing methods for two-phase flows: An overview and recent developments, Center for Turbulence Research Annual Research Briefs 2017 (2017) 117–135.

[14] M. R. Baer, J. W. Nunziato, A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials, International journal of multiphase flow 12 (6) (1986) 861–889.

Journal Pre-proof

[15] R. Saurel, R. Abgrall, A multiphase Godunov method for compressible multifluid and multiphase flows, Journal of Computational Physics 150 (2) (1999) 425–467.

[16] G. Allaire, S. Clerc, S. Kokh, A five-equation model for the simulation of interfaces between compressible fluids, Journal of Computational Physics 181 (2) (2002) 577–616.

[17] V. Coralic, T. Colonius, Finite-volume WENO scheme for viscous compressible multicomponent flows, Journal of computational physics 274 (2014) 95–121.

[18] E. Johnsen, T. Colonius, Implementation of WENO schemes in compressible multicomponent flow problems, Journal of Computational Physics 219 (2) (2006) 715–732.

[19] X. Deng, S. Inaba, B. Xie, K.-M. Shyue, F. Xiao, High fidelity discontinuity-resolving reconstruction for compressible multiphase flows with moving interfaces, Journal of Computational Physics 371 (2018) 945–966.

[20] Q. Li, Y. Lv, L. Fu, A high-order diffuse-interface method with TENO-THINC scheme for compressible multiphase flows, International Journal of Multiphase Flow 173 (2024) 104732.

[21] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, Journal of computational physics 115 (1) (1994) 200–212.

[22] L. Fu, X. Y. Hu, N. A. Adams, A family of high-order targeted ENO schemes for compressible-fluid simulations, Journal of Computational Physics 305 (2016) 333–359.

[23] F. Xiao, Y. Honma, T. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, International journal for numerical methods in fluids 48 (9) (2005) 1023–1040.

[24] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49.

[25] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, Journal of computational physics 210 (1) (2005) 225–246.

[26] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of computational physics 100 (1) (1992) 25–37.

[27] Y. Bazilevs, I. Akkerman, D. Benson, G. Scovazzi, M. J. Shashkov, Isogeometric analysis of Lagrangian hydrodynamics, Journal of Computational Physics 243 (2013) 224–243.

[28] H. Luo, J. D. Baum, R. Löhner, On the computation of multi-material flows using ALE formulation, Journal of Computational Physics 194 (1) (2004) 304–328.

[29] D. L. Youngs, Time-dependent multi-material flow with large fluid distortion, Numerical methods for fluid dynamics (1982).

[30] F. Gibou, R. P. Fedkiw, L.-T. Cheng, M. Kang, A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains, Journal of Computational Physics 176 (1) (2002) 205–227.

[31] W. E. H. Sollie, O. Bokhove, J. J. van der Vegt, Space–time discontinuous Galerkin finite element method for two-fluid flows, Journal of computational physics 230 (3) (2011) 789–817.

[32] Z.-J. Li, Y. Ren, Y. Shen, H. Ding, 2D conservative sharp interface method for compressible three-phase flows: Ternary fluid flows and interaction of two-phase flows with solid, Journal of Computational Physics 538 (2025) 114187.

[33] P.-N. Sun, D. Le Touzé, G. Oger, A.-M. Zhang, An accurate SPH Volume Adaptive Scheme for modeling strongly-compressible multiphase flows. Part 1: Numerical scheme and validations with basic 1D and 2D benchmarks, Journal of Computational Physics 426 (2021) 109937.

[34] Y.-X. Peng, A.-M. Zhang, F.-R. Ming, Particle regeneration technique for smoothed particle hydrodynamics in simulation of compressible multiphase flows, Computer Methods in Applied Mechanics and Engineering 376 (2021) 113653.

[35] S. Pineda, J.-C. Marongiu, S. Aubert, M. Lance, Simulation of a gas bubble compression in water near a wall using the SPH-ALE method, Computers & Fluids 179 (2019) 459–475.

[36] M. Yuan, T. Gao, L. Fu, A diffuse-interface smoothed particle hydrodynamics method for compressible multiphase flows, Journal of Computational Physics 524 (2025) 113717.

[37] F. H. Harlow, The particle-in-cell computing method for fluid dynamics, Methods Computational Physics 3 (1964) 319–343.

[38] J. U. Brackbill, D. B. Kothe, H. M. Ruppel, FLIP: a low-dissipation, particle-in-cell method for fluid flow, Computer Physics Communications 48 (1) (1988) 25–38.

[39] D. Sulsky, Z. Chen, H. L. Schreyer, A particle method for history-dependent materials, Computer Methods in Applied Mechanics and Engineering 118 (1-2) (1994) 179–196.

[40] L. Tran, J. Kim, M. Berzins, Solving time-dependent PDEs using the material point method, a case study from gas dynamics, International journal for numerical methods in fluids 62 (7) (2010) 709–732.

[41] P. J. Baioni, T. Benacchio, L. Capone, C. De Falco, Portable, massively parallel implementation of a material point method for compressible flows, Computational Particle Mechanics 12 (6) (2025) 4771–4793.

[42] S. Peddavarapu, T.-H. Huang, Reproducing kernel enhanced material point method with improved artificial viscosity formulations for shock wave and vortical compressible flows, Computational Mechanics 76 (3) (2025) 797–828.

[43] F. H. Harlow, J. E. Welch, et al., Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics of fluids 8 (12) (1965) 2182.

[44] R. J. LeVeque, Finite volume methods for hyperbolic problems, Vol. 31, Cambridge university press, 2002.

[45] W. H. Reed, T. R. Hill, Triangular mesh methods for the neutron transport equation, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Lab., N. Mex.(USA) (1973).

[46] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework, Mathematics of computation 52 (186) (1989) 411–435.

[47] 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, Journal of computational Physics 84 (1) (1989) 90–113.

[48] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case, Mathematics of Computation 54 (190) (1990) 545–581.

[49] 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, Journal of Computational Physics 280 (2015) 489–509.

[50] C. Michoski, J. Evans, P. G. Schmitz, A. Vasseur, A discontinuous Galerkin method for viscous compressible multifluids, Journal of Computational Physics 229 (6) (2010) 2249–2266.

[51] S. Fechter, C.-D. Munz, A discontinuous Galerkin-based sharp-interface method to simulate three-dimensional compressible two-phase flow, International Journal for Numerical Methods in Fluids 78 (7) (2015) 413–435.

[52] C. Wang, C.-W. Shu, An interface treating technique for compressible multi-medium flow with Runge–Kutta discontinuous Galerkin method, Journal of Computational Physics 229 (23) (2010) 8823–8843.

[53] F. Pochet, K. Hillewaert, P. Geuzaine, J.-F. Remacle, É. Marchandise, A 3D strongly coupled implicit discontinuous Galerkin level set-based method for modeling two-phase flows, Computers & Fluids 87 (2013) 144–155.

[54] Y. Lv, M. Ihme, Discontinuous Galerkin method for multicomponent chemically reacting flows and combustion, Journal of Computational Physics 270 (2014) 105–137.

[55] H. Lu, J. Zhu, D. Wang, N. Zhao, Runge–kutta discontinuous galerkin method with front tracking method for solving the compressible two-medium flow, Computers & Fluids 126 (2016) 1–11.

[56] X.-L. Deng, M. Li, Simulating compressible two-medium flows with sharp-interface adaptive Runge–Kutta discontinuous Galerkin methods, Journal of Scientific Computing 74 (3) (2018) 1347–1368.

[57] D. Luo, J. Qiu, J. Zhu, Y. Chen, A quasi-conservative discontinuous Galerkin method for multi-component flows using the non-oscillatory kinetic flux, Journal of Scientific Computing 87 (3) (2021) 96.

[58] J. Cheng, F. Zhang, T. Liu, A discontinuous Galerkin method for the simulation of compressible gas-gas and gas-water two-medium flows, Journal of Computational Physics 403 (2020) 109059.

[59] T. Gao, D. Hyde, A particle-grid discontinuous Galerkin method, Part 1: Weakly compressible free-surface and multiphase flows, Journal of Computational Physics 566 (2026) 115290.

[60] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in Computational Mathematics 4 (1) (1995) 389–396.

[61] D. Adalsteinsson, J. A. Sethian, The fast construction of extension velocities in level set methods, Journal of Computational Physics 148 (1) (1999) 2–22.

[62] F. De Vanna, F. Picano, E. Benini, A sharp-interface immersed boundary method for moving objects in compressible viscous flows, Computers & Fluids 201 (2020) 104415.

[63] Y. Jiang, H. Yang, Q. Tang, Y. Zhu, High-order WCNS scheme with a constrained weighted least squares immersed boundary method for inviscid compressible flow simulations, Advances in Aerodynamics 8 (1) (2026) 9.

[64] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: 27th Aerospace sciences meeting, 1989, p. 366.

[65] B. Van Leer, Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme, Journal of computational physics 14 (4) (1974) 361–370.

[66] P. L. Roe, Characteristic-based schemes for the Euler equations, Annual review of fluid mechanics 18 (1) (1986) 337–365.

[67] T. Liu, B. Khoo, K. Yeo, Ghost fluid method for strong shock impacting on material interface, Journal of computational physics 190 (2) (2003) 651–681.

[68] C. Wang, T. Liu, B. C. Khoo, A real ghost fluid method for the simulation of multimedium compressible flow, SIAM Journal on Scientific Computing 28 (1) (2006) 278–302.

[69] A. Harten, P. D. Lax, B. v. Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM review 25 (1) (1983) 35–61.

[70] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock waves 4 (1) (1994) 25–34.

[71] L. Xu, T. Liu, Accuracies and conservation errors of various ghost fluid methods for multi-medium Riemann problem, Journal of Computational Physics 230 (12) (2011) 4975–4990.

[72] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Mathematics of computation 67 (221) (1998) 73–85.

[73] T. Gao, L. Fu, A three-dimensional meshless fluid–shell interaction framework based on smoothed particle hydrodynamics coupled with semi-meshless thin shell, Computer Methods in Applied Mechanics and Engineering 429 (2024) 117179.

[74] N. Grenier, M. Antuono, A. Colagrossi, D. Le Touzé, B. Alessandrini, An Hamiltonian interface SPH formulation for multi-fluid and free surface flows, Journal of Computational Physics 228 (22) (2009) 8380–8393.

[75] X. Bai, M. Li, A conservative sharp-interface numerical method for two-dimensional compressible two-phase flows, Journal of Scientific Computing 97 (2) (2023) 30.

[76] J.-F. Haas, B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, Journal of Fluid Mechanics 181 (1987) 41–76.

[77] H. Terashima, G. Tryggvason, A front-tracking/ghost-fluid method for fluid interfaces in compressible flows, Journal of Computational Physics 228 (11) (2009) 4012–4037.

[78] J. J. Quirk, S. Karni, On the dynamics of a shock–bubble interaction, Journal of Fluid Mechanics 318 (1996) 129–163.

[79] M. Cremonesi, A. Frangi, A Lagrangian finite element method for 3D compressible flow applications, Computer Methods in Applied Mechanics and Engineering 311 (2016) 374–392.

[80] K.-M. Shyue, A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions, Journal of Computational Physics 215 (1) (2006) 219–244.

[81] K. So, X. Hu, N. A. Adams, Anti-diffusion interface sharpening technique for two-phase compressible flow simulations, Journal of Computational Physics 231 (11) (2012) 4304–4323.

[82] L. Han, X. Hu, N. A. Adams, Adaptive multi-resolution method for compressible multi-phase flows with sharp interface model and pyramid data structure, Journal of Computational Physics 262 (2014) 131–152.

[83] R. K. Shukla, Nonlinear preconditioning for efficient and accurate interface capturing in simulation of multicomponent compressible flows, Journal of Computational Physics 276 (2014) 508–540.

[84] X. Zeng, G. Scovazzi, A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations, Journal of Computational Physics 270 (2014) 753–783.

[85] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method, Journal of Computational Physics 229 (12) (2010) 4724–4761.

[86] L. Cheng, X. Deng, B. Xie, Y. Jiang, F. Xiao, Low-dissipation BVD schemes for single and multi-phase compressible flows on unstructured grids, Journal of Computational Physics 428 (2021) 110088.

[87] M. Kucharik, R. V. Garimella, S. P. Schofield, M. J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, Journal of Computational Physics 229 (7) (2010) 2432–2452.

[88] 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, Physics of Fluids 37 (7) (2025) 073348.

[89] R. K. Shukla, C. Pantano, J. B. Freund, An interface capturing method for the simulation of multi-phase compressible flows, Journal of Computational Physics 229 (19) (2010) 7411–7439.

Journal Pre-proof