Weighted ghost fluid discontinuous Galerkin method for two-medium problems

Yun-Long Liu[a], Chi-Wang Shu[b,*], A-Man Zhang[a]

[a] *College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China*
[b] *Division of Applied Mathematics, Brown University, Providence RI 02912, USA*

Abstract

A new interface treating method is proposed to simulate compressible two-medium problems with the Runge-Kutta discontinuous Galerkin (RKDG) method. In the present work, both the Euler equation and the level-set equation are discretized with the RKDG method which is compact and of high-order accuracy. The linearized interface inside an interface cell is recovered by the level-set function. The new solution of this cell is taken as a convex combination of two auxiliary solutions. One is the solution obtained by the RKDG method for a single-medium cell with proper numerical fluxes, and the other one is the intermediate state of the two-medium Riemann problem constructed in the normal direction. The weights of the two auxiliary solutions are carefully chosen according to the location of the interface inside the cell. Thus, it ensures a smooth transition when the interface leaves one cell and enters a neighboring cell. The entropy-fix technique is adopted to minimize the overshoots or undershoots in problems with large entropy ratio across the interface. The scheme is justified in a 1-dimensional situation and extended to 2-dimensional problems. Several 1-dimensional two-medium problems, including both smooth and discontinuous examples, are simulated and compared with exact solutions. Also, three 2-dimensional benchmark problems are simulated to validate the present method in two-medium problems.

Keywords: two-medium flow, Runge-Kutta discontinuous Galerkin method, Compressible flow, Interface treating method

*Corresponding author
Email addresses: yunlong_liu@hrbeu.edu.cn (Yun-Long Liu), Chi-Wang_Shu@brown.edu (Chi-Wang Shu), zhangaman@hrbeu.edu.cn (A-Man Zhang)

1. Introduction

Multi-medium problems have always been a significant aspect of fluid dynamics. The appearance of contact discontinuities at material interface always limits the accuracy of the numerical scheme. Among the enormous amount of methods for treating the multi-medium interface, they can be roughly classified as the diffusive interface methods and the sharp interface methods.

In the diffusive interface methods, the interface is usually represented by a phase function that transits from one fluid to another through a few layers of the grid. Among them, the mixture theory with the interface reconstructed by the volume of fluid (VOF) method [3, 18, 31, 38] is widely used in the commercial hydrodynamics codes. It is robust and able to retain the phase transition within 3-4 cells. However, the accuracy of the interface advancement is usually limited to second-order and interface reconstruction schemes with a higher order of accuracy are non-trivial in multi-dimensional problems [3]. To achieve higher-order of accuracy, So et al. [29] directly solved the convection equation of the volume fraction function with a high-order finite difference method and the construction of the material interface was avoided. To offset the diffusion effect introduced by the numerical scheme, an anti-diffusion term was added in the convection equation of the volume fraction function. The scheme is conservative in the sense of total mass, momentum and energy, but only the gas-gas interfaces with small density ratios were simulated. Shyue [28] proposed the γ-based model which solved two convection equations for two functions of γ and P_w, the two material constants in the Tammann equation of state (EOS), instead of the volume fraction for the compressible Euler equation, which improved the stability significantly. More recently, Cheng et al. [5] combined the γ-based model with the weighted essential non-oscillation (WENO) scheme in the framework of the discontinuous Galerkin method to formulate a uniformly high-order interface resolving method. Their schemes were efficient and the results were promising. However, the γ-based model can only handle the fluids described by the ideal gas or Tammann EOS. A significant feature of the diffusive interface methods is that it is straightforward to apply them in the multi-medium problems containing more than two fluid components. Anderson et al. [1] combined the Lagrangian high-order finite-element approach and a remapping procedure to simulate multi-medium flows. The volume fraction function's remapping between the deformed and the original mesh was implemented with the discontinuous Galerkin method over a fictitious time interval. Aditya et al. [26] developed a reconstructed discontinuous Galerkin method for multi-medium hydrodynamics. The underlying formulations were presented for the mixture cell containing several fluids. Although only 1-dimensional numerical experiments were presented for validation, this method has great potential

in multiple spatial dimensions.

On the other hand, the sharp interface methods have been developed extensively in the past few decades. In these methods, the first task is to accurately advance the interface in the Eulerian grid. One approach is the front-tracking method [13–15, 17] which tracks the interface explicitly. It uses a lower-dimensional grid to represent an interface in the flow such that the computational cost is reduced [15, 17]. However, it requires a complicated mesh reconstruction to deal with the difficulty of any topological change of the fluid domain, which is non-trivial for multi-dimensional problems. The other approach is the level-set method [25] which implicitly defines the interface by the zero iso-surface of a level-set function. The interface is advanced by solving the convection equation similar to that in the VOF methods. Compared with the VOF methods, it is easy to be solved with high-order schemes because the level-set function is smooth near the interface. However, in large deformation problems, the level-set function can be ill-defined after some time of advancing. Then, a re-initialization procedure is introduced to reset the level-set function to the signed distance function while keeping the interface position [10]. More recently, Wen [35] and Nguyen et al. [24] coupled the level-set method and the VOF method in the incompressible interfacial flow simulation to maintain high-order accuracy and conservation. For compressible problems, Hu et al. [2] combined interface description and geometric operation to formulate a conservative interface method with a level-set technique. After the interface position is advanced accurately, a scheme to treat the interface cell is required. As one of the most popular methods in this field, the ghost fluid method (GFM) was originally proposed in Fedkiw et al. [10] (referred to as the original GFM (OGFM) in this paper), which is a method solving two single medium problems separately instead of dealing with their direct interaction. For each medium being solved, the fluid domain is expanded across the interface for several layers of cells which are referred to as the ghost cells. The state variables of these ghost cells are filled artificially by assuming the continuity of the normal velocity, pressure and entropy across the interface. For smooth problems and discontinuous problems without entropy jump for each fluid component, the OGFM presents promising results, although the conservation at the interface cell is not ensured. The conservation error is justified to be at the same order of the cell size if the interface instabilities are not involved [10]. When the problem involves the interaction of a strong shock and an interface, wrong results may be obtained with the OGFM since the assumption of continuous entropy is not reasonable anymore. Thus, Liu et al. [22] proposed the modified GFM (MGFM) which used the intermediate state of the two-fluid Riemann problem to fill the interface cell instead of assuming any continuity. Then Qiu et al. [27] coupled the MGFM and the discontinuous Galerkin method together to simulate

multi-medium flows. In the sense of discontinuity, the fluid-structure interface is the same as the fluid-fluid interface. Thus, the MGFM is also applicable to treat the fluid-structure interfaces [12, 23]. To overcome the inaccurate interfacial condition in shock impedance matching problems, Wang et al. [33] further defined the real fluid nodes just next to the interface which led to the real-GFM. Another variant of the MGFM is the practical GFM proposed in [37] that reduces the degrees of freedom required for defining the ghost cells near the interface. The MGFM and its variants provide simple and practical frameworks for the numerical simulation of multi-medium problems with complex wave interactions, and are employed in a wide range of applications. In these GFMs, the location of the interface is only used to label the single-medium and multi-medium cells. The ghost cell state and final solution are not related to the exact location of the interface inside a cell. As a result, a sudden jump for the system property could arise when the interface leaves one cell and enters its neighbor.

In the present paper, an interface treating method is proposed on the basis of the GFM to resolve the interfacial problem of compressible fluids. The new method is expected to be able to consider the exact position of the interface inside the cell to achieve higher resolution and better robustness. The rest of this paper is organized as follows. Firstly, a brief review is presented as the framework of the current scheme. Secondly, MGFM for 1-dimensional two-medium problems is presented and justified with the wave structure inside an interface cell. Thirdly, the scheme is extended to 2-dimensional problems. Then, some 1-dimensional and 2-dimensional benchmark cases are simulated to test the present scheme. At last, some concluding remarks are given.

2. Brief review of the RKDG method for two-medium Euler equations

2.1. Basic equations

The Euler equation in conservative form describing compressible fluids in 2-dimensional domain reads

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0, \tag{1}$$

where $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{G}$ are the vectors of the conservative quantities and fluxes in the x and y directions, respectively. The subscripts of t, x and y indicate the partial derivatives with respect to them. The three vectors are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix},$$

where p is the pressure, ρ is the fluid density, $\mathbf{u} = [u, v]$ is the velocity vector and $E = \rho e + \frac{1}{2}\rho(u^2+v^2)$ is the total energy with e representing the specific internal energy. The four equations in the system refer to the conservation of mass, momentums in the x and y directions and total energy, respectively. For a 1-dimensional problem, we just set $v = 0$ and remove the third component of each vector. The system is then closed by the EOS of the fluid which relates the density ρ, pressure p and specific internal energy e. In this paper, two kinds of EOS are adopted, i.e., the ideal gas EOS

$$p = \rho e(\gamma - 1), \tag{2}$$

and the Tammann EOS [19]

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

Herein, γ is the specific heat ratio and P_w is the reference pressure. By choosing appropriate constants γ and P_w, they can be used to model a wide range of gases and liquids.

In multi-medium problems, the evolution of the interface is required to be resolved. In the present work, the level-set method is used for this purpose which solves a convection equation [10, 25] for the level-set function ϕ:

$$\phi_t + u\phi_x + v\phi_y = 0. \tag{4}$$

Usually, ϕ is initially defined as the signed distance function such that the two fluids are separated by the iso-line where $\phi = 0$. If ϕ is convected by the velocity of the fluid, the interface can still be implicitly captured by finding the same iso-line. To maintain the stability and resolution of the simulation, the reinitialization process is adopted in this paper following Fedkiw et al. [10] to reset ϕ to the signed distance function by solving the equation

$$\phi_{t'} + s\sqrt{\phi_x^2 + \phi_y^2} = s \tag{5}$$

to steady state, with t' representing the pseudo time and

$$s = \frac{\phi}{\sqrt{\phi^2 + h^2}}$$

representing the smoothed sign function. Herein, h is the cell size.

2.2. Semi-discretization form of the RKDG method

The RKDG method adopted here follows the framework established in [7, 8]. Denote T_h as an uniform rectangular partition of the computational domain Ω and $I_{i,j}$ represents the cell bounded by $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$. It should be noted that the RKDG method allows arbitrary triangulations as shown in [7], and we are using the rectangular meshes in this paper just as an example. The cell size in the x and y directions are both taken as h, which is not necessary but is adopted for simplicity. P^k is the kth degree polynomial space and $\Psi = \{\psi_\ell\}$ is a complete set of orthogonal basis functions for $\ell = 1, 2...N_p$, where $N_p = \frac{1}{2}(k+1)(k+2)$. In this paper, the basis functions are chosen as the normalized 2-dimensional Legendre polynomials such that

$$\int_{\bar{I}} \psi_\ell \psi_m \mathrm{d}\xi \mathrm{d}\zeta = 4\delta(\ell, m), \tag{6}$$

where $\bar{I}$ is the reference cell bounded by $[-1, 1] \times [-1, 1]$ on the $\xi\zeta$ plane with $\xi = \frac{2}{h}[x - \frac{1}{2}(x_i + x_{i+1})]$ and $\zeta = \frac{2}{h}[y - \frac{1}{2}(y_j + y_{j+1})]$, and $\delta(\ell, m)$ is the standard Kronecker delta function. For example, the basis functions for the P^2 space are given by

$$\Psi = \begin{bmatrix} 1 \\ \sqrt{3}\xi \\ \sqrt{3}\zeta \\ 3\xi\zeta \\ \frac{\sqrt{5}}{2}(3\xi^2 - 1) \\ \frac{\sqrt{5}}{2}(3\zeta^2 - 1) \end{bmatrix}. \tag{7}$$

In the present paper, both the Euler equation and the level-set equation are discretized by the RKDG method. For the Euler equation, we follow the procedure of the DG method for conservation law systems in 2-dimensions [7]. The goal is to find the numerical solution $\mathbf{U}^h \in P^k$ satisfying the semi-discretization equation for cell $I_{i,j}$:

$$\begin{aligned} \int_{I_{i,j}} \mathbf{U}_t^h \psi_\ell \mathrm{d}x\mathrm{d}y = & \int_{I_{i,j}} \left[\mathbf{F}(\mathbf{U}^h)(\psi_\ell)_x + \mathbf{G}(\mathbf{U}^h)(\psi_\ell)_y\right] dxdy + \\ & \int_{y_j}^{y_{j+1}} \left[\hat{\mathbf{F}}(\mathbf{U}^h(x_i, y))\psi_\ell(x_i, y) - \hat{\mathbf{F}}(\mathbf{U}^h(x_{i+1}, y))\psi_\ell(x_{i+1}, y)\right] \mathrm{d}y + \\ & \int_{x_i}^{x_{i+1}} \left[\hat{\mathbf{G}}(\mathbf{U}^h(x, y_j))\psi_\ell(x, y_j) - \hat{\mathbf{G}}(\mathbf{U}^h(x, y_{j+1}))\psi_\ell(x, y_{j+1})\right] \mathrm{d}x \end{aligned} \tag{8}$$

where $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ represent the numerical fluxes calculated with $\mathbf{F}$ and $\mathbf{G}$ on the shared edges of the current cell and its immediate adjacent neighbors. In this paper, the

numerical fluxes are calculated with the HLLC Riemann solver [19]. When dealing with the fluxes between fluids with widely different densities, the estimation of signal speed can be critical in the HLLC solver. In the present paper, the pressure-based wave speed estimates [32] is adopted to determine the intermediate state and the signal speed. In the following sections, the superscript h is omitted to simplify the notations.

Because the basis functions are normalized and are orthogonal to each other, by expressing the numerical solution $\mathbf{U} = \sum_{\ell=1}^{N_p} \psi_\ell \mathcal{E}_\ell$ and $\mathbf{U}_t = \sum_{\ell=1}^{N_p} \psi_\ell (\mathcal{E}_\ell)_t$, Eq. (8) can be expressed as the following ODEs about the polynomial coefficients $\mathcal{E}_\ell$,

$$\begin{aligned}\frac{h^2}{4}(\mathcal{E}_\ell)_t = &\int_{I_{i,j}} [\mathbf{F}(\mathbf{U})(\psi_\ell)_x + \mathbf{G}(\mathbf{U})(\psi_\ell)_y]\,\mathrm{d}x\mathrm{d}y - \\ &+ \int_{y_j}^{y_{j+1}} \left[\hat{\mathbf{F}}(\mathbf{U}(x_i, y))\psi_\ell(x_i, y) - \hat{\mathbf{F}}(\mathbf{U}(x_{i+1}, y))\psi_\ell(x_{i+1}, y)\right]\mathrm{d}y + \\ &\int_{x_i}^{x_{i+1}} \left[\hat{\mathbf{G}}(\mathbf{U}(x, y_j))\psi_\ell(x, y_j) - \hat{\mathbf{G}}(\mathbf{U}(x, y_{j+1}))\psi_\ell(x, y_{j+1})\right]\mathrm{d}x \quad (9)\end{aligned}$$

for $\ell = 1, 2...N_p$. Thus, the calculation of the original conservative state vector is converted to the update of $\mathcal{E}_\ell$. As long as they are properly obtained, the conservative state vector at any location of a cell can be recovered by the weighted summation of the basis vector.

As for the level-set function ϕ, the convection equation and the reinitialization equation are also discretized with the RKDG method to keep the uniformity of the solver. These two equations can both be written in the form of Hamilton-Jacobi equations, and the direct DG method proposed in Cheng & Shu [6] is adopted.

2.3. Time discretization

The standard third-order strong stability-preserving Runge-Kutta method [16] is adopted in this paper to discretize the time domain for both the Euler equation and the level-set equation. Denote $L(\mathbf{U}, t)$ as the spatial discretization operator with the DG method, the new solution at t_{n+1} can be expressed as

$$\mathbf{U}(t_{n+1}) = \frac{1}{3}\mathbf{U}(t_n) + \frac{2}{3}\mathbf{U}_{(2)} + \frac{2}{3}\Delta t L(\mathbf{U}_{(2)}, t_n + \frac{1}{2}\Delta t), \quad (10)$$

with

$$\begin{aligned}\mathbf{U}_{(1)} &= \frac{3}{4}\mathbf{U}(t_n) + \frac{1}{4}\Delta t L(\mathbf{U}(t_n), t_n), \text{ and} \\ \mathbf{U}_{(2)} &= \frac{3}{4}\mathbf{U}(t_n) + \frac{1}{4}\mathbf{U}_{(1)} + \frac{1}{4}\Delta t L(\mathbf{U}_{(1)}, t_n + \Delta t),\end{aligned}$$

where Δt is the time increment controlled by the Courant-Friedrichs-Lewy (CFL) condition.

3. New algorithm for two-medium problems

When dealing with interface cells, the GFM solves two single-fluid problems instead of the motion of their mixture. The cells on the other side of and adjacent to the interface are denoted as the ghost cells and filled properly to match the interface condition. In the OGFM [10], the fluid state of the ghost cells is determined by the entropy of the currently concerned fluid, and the velocity and pressure of the other one. The MGFM [22] constructs a Riemann problem and uses the intermediate state to fill the ghost cells such that the interaction of a strong shock and a material interface could be correctly resolved. However, in these GFMs, the ghost cell state is not related to the exact location of the interface inside the cell, and this will result in a stiff jump and loss of accuracy when the interface moves from one cell to its adjacent neighbor. Besides, when the MGFM is combined with the RKDG method, the interface cells are also treated as ghost cells [27, 34]. Thus, their material states are overwritten at every time step, which introduces more conservation errors in smooth problems with the interface moving in a ramping pressure field (see the comparison in section 5.1.1).

In this paper, the GFM is modified to get a continuous transition when the interface moves between cells, and we are trying to establish a new GFM which has both a high conservation accuracy and the accuracy in treating state jumps as the MGFM.

3.1. 1-dimensional problem

For 1-dimensional problems, Eq. (1) is simplified by setting $v = 0$ and removing the third components from $\mathbf{U}$ and $\mathbf{F}$. The multi-medium interface can be represented by a single point location separating the two fluid components, and can be updated with the material velocity explicitly instead of employing the level-set method. The computational domain is partitioned into uniform cells with cell length h and cell i is defined by $[x_i, x_{i+1}]$. Denote $x = a$ as the spatial coordinate of the interface in cell i at t_n, i.e., $x_i < a < x_{i+1}$. The state vector of the left hand side fluid is denoted by $\mathbf{U}_L^i$ and the right hand side fluid is denoted by $\mathbf{U}_R^i$, with the superscript representing the cell number. Typically, they are controlled by different EOSs. Without losing any generality, we only consider the evolution of $\mathbf{U}_L$, as $\mathbf{U}_R$ can be obtained by symmetry. Limited by the CFL condition, the interface can never pass over a whole cell during a single time increment. Thus, over the whole computational domain, cells can be classified as the following 3 types:

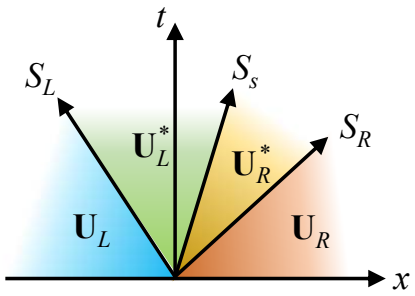


Figure 1: Configuration of a 1-dimensional Riemann problem inside an interface cell

(1) If cell i still remains single-medium from t_n to t_{n+1}, the ordinary single-medium RKDG method is used to discretize the Euler equation and to update the state vector $\mathbf{U}_L^i$. The details will not be repeated here;
(2) If cell i contains the interface at t_n, which is referred as an interface cell, it is solved with the algorithm described in section 3.1.1;
(3) If cell i is single-medium at t_n but changes to an interface cell at t_{n+1}, which is referred to as a prepared interface cell, it is updated with the algorithm described in section 3.1.2.

3.1.1. Algorithm for the update of an interface cell

Firstly, the Riemann problem inside the interface cell should be explained. The interface between the two fluids in the cell i forms a Riemann problem, in which the space-time domain is divided into 4 regions with 3 characteristic lines and the corresponding slopes S_L, S^* and S_R, as shown in Fig. 1. From left to right, the state vectors are $\mathbf{U}_L^i$, $\mathbf{U}_L^*$, $\mathbf{U}_R^*$ and $\mathbf{U}_R^i$, respectively. As long as $\mathbf{U}_L^i$ and $\mathbf{U}_R^i$ are given, all these state vectors and signal speeds can be obtained with an exact or approximate Riemann solver. In this paper, the HLLC solver is used. Following the treatment proposed in Qiu et al. [27], the high-order moments in the polynomials of the state vector of the interface cells are trimmed such that only the average values remain to maintain the stability. Then, if the interface is still inside cell i, we claim that the average state vector of cell i at the time level $n+1$ can be expressed as

$$\bar{\mathbf{U}}_L^i(t_{n+1}) = k_1\tilde{\mathbf{U}}_L^i(t_{n+1}) + k_2\hat{\mathbf{U}}_L^i(t_{n+1}) \tag{11}$$

and the position of the interface can be updated with the contact discontinuity speed S^*. $\tilde{\mathbf{U}}_L^i$ is the result obtained by an explicit ODE solver resolving

$$\begin{cases} \frac{\mathrm{d}\tilde{\mathbf{U}}_L^i}{\mathrm{d}t} = \frac{1}{h}\left[\mathbf{F}_{\mathrm{Rm}}(\bar{\mathbf{U}}_L^{i-1}, \bar{\mathbf{U}}_L^i) - \mathbf{F}_{\mathrm{Rm}}(\bar{\mathbf{U}}_L^i, \bar{\mathbf{U}}_R^{i+1})\right], \\ \tilde{\mathbf{U}}_L^i(t_n) = \bar{\mathbf{U}}_L^i(t_n), \end{cases} \tag{12}$$

with $\mathbf{F}_{\mathrm{Rm}}(\mathbf{U}_1, \mathbf{U}_2)$ representing the numerical flux given by the Riemann problem with inputs of $\mathbf{U}_1$ and $\mathbf{U}_2$. As for $\hat{\mathbf{U}}_L^i(t_{n+1})$, it is taken as the intermediate state of the Riemann problem constructed by $f_1\bar{\mathbf{U}}_L^{i-1}(t_n) + (1-f_1)\bar{\mathbf{U}}_L^i t_n$ and $\bar{\mathbf{U}}_R^{i+1}$. Herein, $f_1 = \frac{1}{h}(a - x_i)$ is the volume fraction of fluid L in cell i at t_n. Similarly, we define $f_2 = \frac{1}{h}(\Delta t S^*)$ and $f_3 = \frac{1}{h}(\Delta t S_L)$ such that the coefficients in Eq. (11) are given by

$$[k_1, k_2] = \begin{cases} [1, 0] & \text{if } f_1 + f_3 \geq 1 - \epsilon \\ [0, 1] & \text{else if } f_1 + f_3 \leq \epsilon \\ \frac{1}{f_1+f_2}[f_1 + f_3, f_2 - f_3] & \text{for else,} \end{cases} \tag{13}$$

where ϵ is a small and positive constant for stability which is taken as 10^{-9} in our computation. Since k_1 and k_2 are both bounded by 0 and 1, the resulting $\bar{\mathbf{U}}_L^i(t_{n+1})$ is a convex combination of the two solutions $\tilde{\mathbf{U}}_L^i$ and $\hat{\mathbf{U}}_L^i$. This ensures good stability such as the maintenance of positivity for density and internal energy.

Justifications:

Consider 3 situations based on $f_1 + f_3$:

(i) $f_1 + f_3$ approaching 0 from the right hand side;
(ii) $f_1 + f_3$ approaching 1 from the left hand side;
(iii) A general situation for $f_1 + f_3$ between 0 and 1.

Case (i) means that cell i is almost occupied by fluid R at t_{n+1}. The leftward signal S_L of the Riemann problem will propagate over the whole area of fluid L. Then the coefficients will make sure that the new solution of $\bar{\mathbf{U}}_L^i$ equals to the intermediate state of the Riemann problem.

Case (ii) means that cell i is almost occupied by fluid L at t_{n+1}. Thus, it is reasonable to treat it as a single medium cell with a right side neighbor of fluid R. Then the coefficients will make sure that the new solution of $\bar{\mathbf{U}}_L^i$ is updated with the original RKDG method in a single medium cell.

As for **Case (iii)**, the coefficients in Eq. (11) make sure that the solution of $\bar{\mathbf{U}}_L^i$ linearly transits between the two cases above by their convex combination. To derive

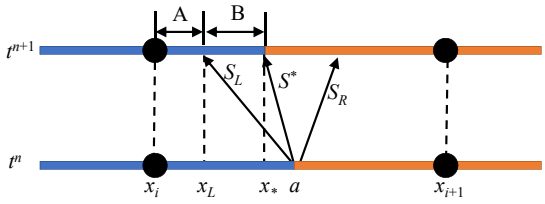


Figure 2: Riemann problems inside the interface cell

the coefficients k_1 and k_2, we shall firstly divide the region occupied by fluid L in cell i at t_{n+1} into partition A and partition B, as shown in Fig. 2, which are separated by the point reached by the slowest wave S_L. The lengths of the two partitions are $x_L - x_i = h(f_1 + f_3)$ and $x_* - x_L = h(f_2 - f_3)$, respectively. Herein, x_L and x_* are the locations of the slowest wave and the material interface at t_{n+1}. At t_{n+1}, $\bar{\mathbf{U}}_L^i$ should be the average value of the two partitions weighted by their lengths, i.e.,

$$\bar{\mathbf{U}}_L^i(t_{n+1}) = \frac{f_1 + f_3}{f_1 + f_2}\tilde{\mathbf{U}}_L^i + \frac{f_2 - f_3}{f_1 + f_2}\hat{\mathbf{U}}_L, \tag{14}$$

with

$$\tilde{\mathbf{U}}_L^i = \mathbf{U}_L^i(t_n) + \frac{\Delta t}{f_1 + f_3}\left(\hat{\mathbf{F}}(x_i) - \hat{\mathbf{F}}(x_L)\right) \tag{15}$$

referring to the state vector of partition A at t^{n+1}. If $\hat{\mathbf{F}}(x_i)$ and $\hat{\mathbf{F}}(x_L)$ are directly given by the real values of $\mathbf{U}_L$ at these two positions, the solution could be unstable since the cell length is scaled by $f_1 + f_3$ which is smaller than 1, and the CFL condition is not satisfied anymore. Thus, based on the fact that, when $f_1 + f_3$ is approaching 0 and 1, $\hat{\mathbf{F}}(a)$ should also be approaching $\hat{\mathbf{F}}(x_i)$ and $\hat{\mathbf{F}}(x_{i+1})$, respectively, a linear interpolation is established to approximate $\hat{\mathbf{F}}(x_L)$ as below

$$\hat{\mathbf{F}}(x_L) = (1 - f_1 - f_3)\hat{\mathbf{F}}(x_i) + (f_1 + f_3)\hat{\mathbf{F}}(x_{i+1}). \tag{16}$$

Plug Eq. (15) and (16) into Eq. (14), we have

$$\bar{\mathbf{U}}_L^i(t_{n+1}) = \frac{f_1 + f_3}{f_1 + f_2}\tilde{\mathbf{U}}_L^i + \frac{f_2 - f_3}{f_1 + f_2}\hat{\mathbf{U}}_L^i. \tag{17}$$

Eq. (17) shows that the new state value of the interface cell can be returned by a convex combination of $\tilde{\mathbf{U}}_L$ and $\hat{\mathbf{U}}_L$.

To calculate $\hat{\mathbf{U}}_L^i$, a multi-medium Riemann problem is formulated. It would be natural to choose the input states of the Riemann problem as $\mathbf{U}_L = \bar{\mathbf{U}}_L$ and

$\mathbf{U}_R = \bar{\mathbf{U}}_R$. However, this choice is not consistent with the calculation of $\tilde{\mathbf{U}}_L^i$. Thus, the input state vector for the Riemann problem should also be changed according to $f_1 + f_3$ instead of taking $\bar{\mathbf{U}}_L$ directly. Thus, $\mathbf{U}_L$ and $\mathbf{U}_R$ are calculated with the following interpolation formulas for the in-cell Riemann problem:

$$\mathbf{U}_L = (1 - f_1 - f_3)\mathbf{U}_L(x_i^-) + (f_1 + f_3)\bar{\mathbf{U}}_L, \tag{18}$$

and

$$\mathbf{U}_R = (1 - f_1 - f_3)\bar{\mathbf{U}}_R + (f_1 + f_3)\bar{\mathbf{U}}_R(x_{i+1}^+). \tag{19}$$

A significant advantage of this interpolation of $\mathbf{U}_L$ and $\mathbf{U}_R$ is that it ensures the continuity of the solution of $\bar{\mathbf{U}}_L^i$ when moving the interface across the edge of cell i. An alternative approach is to recover $\mathbf{U}_L$ from the interpolated numerical flux $\hat{\mathbf{F}}(x_L)$ with

$$\mathbf{U}_L = \mathbf{J}^{-1}\left(\hat{\mathbf{F}}_L(x_L) - \bar{\mathbf{F}}_L\right) + \bar{\mathbf{U}}_L, \tag{20}$$

where $\mathbf{J}$ is the Jacobian matrix of the Euler equation, and $\bar{\mathbf{F}}_L$ is the flux vector corresponding to $\bar{\mathbf{U}}_L$.

3.1.2. Algorithm to update a prepared-interface cell

In this section, a single medium cell of fluid R at t_n which turns into a two-medium one at t_{n+1} is considered. As for the state vector of fluid R, it can be updated as a single-medium cell. As for the state vector of fluid L, we simply fill it with $\hat{\mathbf{U}}_L$ calculated during solving the adjacent two-medium cell. Then, the polynomials for both fluid L and fluid R are trimmed to be constants over the cell.

3.2. 2-dimensional problems

A significant advantage of the GFMs is that they are easy to be extended to multi-dimensional problems. This section presents some special treatments in 2-dimensional problems that should be mentioned for our new approach. For 3-dimensional problems, these treatments could be properly generalized without conceptual difficulties.

For consistency with the 1-dimensional algorithm, we still use the subscripts L and R to denote the two fluids inside a 2-dimensional cell, although they are not referring to the labels of the left and right fluid components. As ϕ represents the level-set function, the unit normal vector of the interface is taken as $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ at the interface, which is assumed to point at fluid R from fluid L. We only discuss the update of fluid L of an interface cell $I_{i,j}$.

3.2.1. Interface construction from the level-set function

For 2-dimensional problems, the first difficulty is to judge whether a cell is a single-medium or interface cell and locate the interface inside it with a given level-set function $\phi(x, y)$. The interface should be a curve defined by $\phi(x, y) = 0$. Then, the fluid fraction f_1 of fluid L can be expressed as

$$\alpha = \int_{I_{i,j}} (1 - H(\phi)) \mathrm{d}x\mathrm{d}y, \tag{21}$$

where H is the Heaviside function

$$H(\phi) = \begin{cases} 0, & \text{when } \phi > 0, \\ 1, & \text{for else.} \end{cases} \tag{22}$$

If ϕ is an arbitrary polynomial with degree 2 or higher, it is non-trivial to calculate the integration above inside a given cell accurately. Thus, the main idea is to truncate the arbitrary polynomial to a linear one such that the interface can be expressed by a line implicitly defined by

$$\mathcal{F}(\mathbf{x}) = \mathbf{x} \cdot \bar{\mathbf{n}} + c = 0. \tag{23}$$

Herein, $\mathbf{x} = (x, y)$ represents the 2-dimensional coordinate vector, $\bar{\mathbf{n}}$ is the normalized normal vector of the linearized interface and can be approximated by the average gradient of ϕ inside the cell:

$$\bar{\mathbf{n}} = \frac{\int_{I_{i,j}} \nabla\phi \mathrm{d}x\mathrm{d}y}{\left| \int_{I_{i,j}} \nabla\phi \mathrm{d}x\mathrm{d}y \right|}, \tag{24}$$

and c is determined with the least square method such that

$$\frac{\partial}{\partial c} \int_{I_{i,j}} (\mathbf{x} \cdot \bar{\mathbf{n}} + c - \phi)^2 \mathrm{d}x\mathrm{d}y = 0. \tag{25}$$

Once $\bar{\mathbf{n}}$ and c are obtained, we can label cell I_{ij} with the following criterion:

$$\begin{cases} I_{i,j} \text{ is a single-medium cell of fluid } L, \text{ if} \max(\mathcal{F}) \leq 0; \\ I_{i,j} \text{ is a single-medium cell of fluid } R, \text{ if} \min(\mathcal{F}) \geq 0; \\ I_{i,j} \text{ is an interface cell for else.} \end{cases}$$

Herein, the maximum and minimum of $\mathcal{F}$ should be evaluated at the entire cell. Because $\mathcal{F}$ is a linear function of $\mathbf{x}$, it is enough to evaluate at the cell's four corners.

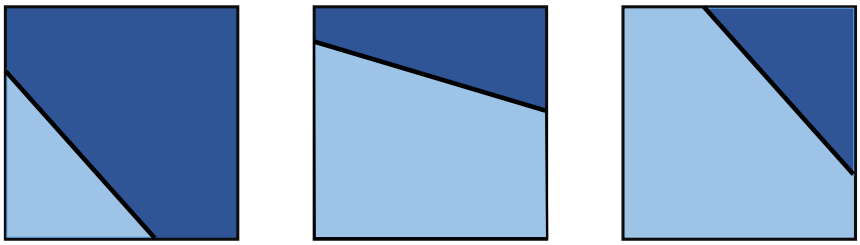


Figure 3: 3 b
interface.

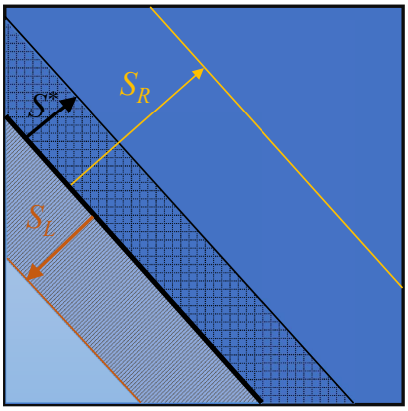


Figure 4: Calculation of f_2 and f_3 in the 2-dimensional problem.

Another significant difference between the 1-dimensional and 2-dimensional problems is the calculation of f_1, f_2 and f_3. In 2-dimensional problems, they represent the ratios between the corresponding areas instead of lengths. Inspired by the VOF method, only the 3 situations shown in Fig. 3 are expected with proper rotating manipulation. Then, the area of cell $I_{i,j}$ cut by the linearized interface is calculated geometrically to obtain f_1. By moving the interface at t_n along the normal direction with the distance of $\Delta t S^*$ and $\Delta t S_L$, f_2 and f_3 are approximated by the ratios of the areas swept by the moving interface over the area of the cell, respectively, as shown by the shadowed area in Fig. 4. It should be noted that these areas can be negative with negative signal speeds S_L and S^*.

It should be noted that Eq. (24) and Eq. (25) are also applicable to an arbitrary triangular cell, for which there will be only one basic geometrical configuration with proper rotating manipulation to calculate the volume fractions f_1, f_2 and f_3.

3.2.2. Algorithm to update an interface cell

Similar to the 1-dimensional problem, Eq. (11) is also used in 2-dimensional problems to compute the new state variables of the interface cells with the convex combination of $\tilde{\mathbf{U}}_L$ and $\hat{\mathbf{U}}_L^{i,j}$. The difficulties are how to calculate $\tilde{\mathbf{U}}_L^{i,j}$ and $\hat{\mathbf{U}}_L^{i,j}$, which are going to be discussed in this section.

As for $\tilde{\mathbf{U}}_L^{i,j}$, we still use the ordinary RKDG solver with proper numerical fluxes. Taking the numerical flux $\hat{\mathbf{F}}(x_{i+\frac{1}{2}}, y)$ on the right side edge of cell $I_{i,j}$ as an example, the input state vector on the left-hand side of the Riemann problem is naturally taken as $\bar{\mathbf{U}}_L^{i,j}$. As for the choice of the right-hand side input vector, we might encounter with the following 3 situations according to the label of its right-hand side neighbor:

(i). Cell $I_{i+1,j}$ is a single-fluid cell of fluid L;
(ii). Cell $I_{i+1,j}$ is a single-fluid cell of fluid R;
(iii). Cell $I_{i+1,j}$ is an interface cell.

For case (i), the flux is calculated with the ordinary procedure of the single-medium RKDG method, with the numerical flux in the x direction computed with the single-medium approximate Riemann solver.

As for case (ii), the flux on the shared edge is taken as

$$\hat{\mathbf{F}}(x_{i+\frac{1}{2},y}) = \begin{cases} \mathbf{F}(\bar{\mathbf{U}}_L^{i,j}) & S_L > 0, \\ \mathbf{F}(\mathbf{U}_L^*(\bar{\mathbf{U}}_L^{i,j}, \bar{\mathbf{U}}_R^{i+1,j}, \bar{\mathbf{n}})) & S_L \leq 0, \end{cases} \tag{26}$$

where $\mathbf{F}(\mathbf{U})$ is the flux defined by the material state $\mathbf{U}$, $\mathbf{U}_L^*(\mathbf{U}_L, \mathbf{U}_R, \bar{\mathbf{n}})$ is the intermediate state of fluid L in the Riemann problem with the input states of $\mathbf{U}_L$ and $\mathbf{U}_R$ in the direction of $\bar{\mathbf{n}}$, and S_L is the slowest signal speed.

As for case (iii), the total flux at the shared edge is calculated with a convex combination of two fluxes calculated in case (i) and case (ii)

$$\hat{\mathbf{F}}(x_{i+1}, y) = k_L \mathbf{F}^{(\mathrm{i})} + k_R \mathbf{F}^{(\mathrm{ii})}, \tag{27}$$

where k_L and k_R are coefficients given by

$$(k_L, k_R) = \begin{cases} (1, 0) & \text{if } l_c \geq l_o \\ \frac{1}{l_o}(l_c, l_o - l_c) & \text{otherwise.} \end{cases} \tag{28}$$

Herein, l_c and l_o are the intercepts of the interface with the current and opposite cell edges, as shown in Fig. 5 for the example that the right side edge is considered. This choice is based on the assumption that the cell is filled with the fluid L when calculating $\tilde{\mathbf{U}}_L^{i,j}$. Thus, the original space occupied by the fluid L (lighter color in the

left-hand side figure in Fig. 5) is firstly scaled proportionally to the shape shown by the right-hand side figure in Fig. 5, and then we fill the whole cell with the fluid L. If the interface is assumed to be continuous across the cell edge, the total flux on the right-hand side edge can be expressed by Eq. (27). This choice of the numerical flux is consistent with the 1-dimensional scheme when the interface becomes horizontal or vertical.

When calculating $\hat{\mathbf{U}}_L^{i,j}$, the procedure is the same as the 1-dimensional algorithm except that the Riemann problem is constructed in the direction of $\bar{\mathbf{n}}$ with the input states of $\mathbf{U}_L$ and $\mathbf{U}_R$. Following the treatment in the 1-dimensional problem, convex combinations of the material states of cell $I_{i,j}$, and its neighbors are used as the Riemann problem's two input states:

$$\mathbf{U}_L = (1 - f_1 - f_3)\mathbf{U}_L^{\text{nbs}} + (f_1 + f_3)\bar{\mathbf{U}}_L^{i,j}, \tag{29}$$

and

$$\mathbf{U}_R = (f_1 + f_3)\mathbf{U}_R^{\text{nbs}} + (1 - f_1 - f_3)\bar{\mathbf{U}}_R^{i,j}, \tag{30}$$

where $\mathbf{U}_L^{\text{nbs}}$ and $\mathbf{U}_R^{\text{nbs}}$ are the material states of the properly chosen neighbors of cell $I_{i,j}$. Taking $\mathbf{U}_L^{\text{nbs}}$ as an example, the two neighbors of cell $I_{i,j}$ are used to calculate $\mathbf{U}_L^{\text{nbs}}$ according to the normal vector $\bar{\mathbf{n}}$:

$$\mathbf{U}_L^{\text{nbs}} = k_x \mathbf{U}_L^{nx} + k_y \mathbf{U}_L^{ny}, \tag{31}$$

where

$$\mathbf{U}_L^{nx} = \begin{cases} \bar{\mathbf{U}}_L^{i-1,j}, & n_x > 0 \\ \bar{\mathbf{U}}_L^{i+1,j}, & n_x \leq 0 \end{cases}, \mathbf{U}_L^{ny} = \begin{cases} \bar{\mathbf{U}}_L^{i,j-1}, & n_y > 0 \\ \bar{\mathbf{U}}_L^{i,j+1}, & n_y \leq 0 \end{cases}, \text{ and } (k_x, k_y) = \frac{(|n_x|, |n_y|)}{|n_x| + |n_y|}$$

with n_x and n_y representing the components of $\bar{\mathbf{n}}$ in x and y directions, respectively. Eq. (29) reduces to Eq. (18) when the interface is aligned with the y axis, and the consistency between the 1 and 2-dimensional algorithms is ensured. We might encounter the situation that a neighbor of cell $I_{i,j}$ that we require above is a single-medium one of fluid R. Then we set the corresponding coefficient in Eq. (31) to zero to avoid using $\bar{\mathbf{U}}_L$ of this cell. In even worse situations where the interface is highly under-resolved, states for fluid L of immediate neighbors in both directions might become unavailable. Then, each neighboring cell's material state is approximated by averaging its surrounding cells containing fluid L with the weight calculated with the distances between cell centers. The detailed procedure follows Wang & Shu [34] and will not be repeated here.

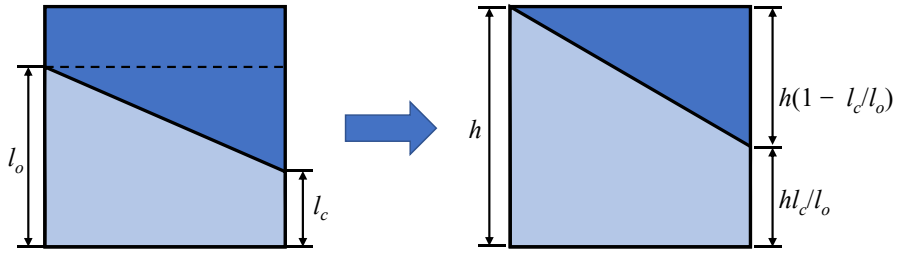


Figure 5: Configuration of the original (left) and the scaled (right) cells for the flux calculation on the right side edge.

3.3. Entropy-fix for interface cells

The convex combination of 2 solutions could lead to entropy problems at the interface cell because the entropy is not linearly dependent on the coefficients. Take a simple example with $\tilde{\mathbf{U}} = [\rho, \rho u, E]$ and $\hat{\mathbf{U}} = [\rho, -\rho u, E]$. Any convex combination of $\tilde{\mathbf{U}}$ and $\hat{\mathbf{U}}$ with nonzero coefficients will result in a pressure and entropy greater than the same combination of those from the two input fluids. Thus, the isobaric fix technique is adopted to alter the resulting conservative quantities to get a reasonable entropy at interface cells. We define the physical entropy [10] as

$$S = \frac{p + P_w}{\rho^\gamma} \tag{32}$$

for the fluid described by the Tammann EOS. In the OGFM [10], the entropy is assumed to be continuous near the interface. Thus, all the conservative fluid quantities of the interface cell are altered without modifying the values of the pressure and velocity according to the entropy of the adjacent single medium cell. However, it has been proved that this treatment may not work well for the case with a strong shock impacting on an interface because the entropy across the shock front can also be discontinuous [22]. Thus, a more reliable prediction of the entropy for the interface cell must be carried out. Given that the MGFM can handle such problems, we take the entropy from the intermediate state of the Riemann problem constructed by the two adjacent single-phase cells. Then, all the following procedures are the same as the entropy-fix in the OGFM.

4. Overall procedures

The overall procedures have the following steps,

(1). Initialize the program;

(2). Linearize the level-set function in each cell to label it as interface cell or single-medium cell, and reconstruct the interface inside each interface cell;
(3). Calculate the numerical fluxes on each cell edge using the HLLC Riemann solver. For interface cells, modify the fluxes based on the criterion given in section 3.2.2;
(4). Calculate $\tilde{\mathbf{U}}$ for interface cells and $\mathbf{U}$ for single-medium cells using the ordinary RKDG method;
(5). Construct the Riemann problem and solve it with the HLLC Riemann solver to get $\hat{\mathbf{U}}$ for interface cells with the fluxes given in step (4);
(6). Calculate the convex combination of $\tilde{\mathbf{U}}$ and $\hat{\mathbf{U}}$ with the weights given by Eq. (13) for interface cells;
(7). Apply the entropy-fix to the old interface cells, and fill the new interface cells following section 3.1.2;
(8). Go back to step.(2) to start the next time increment if the stop criterion is not met.

5. Results and discussion

5.1. 1-dimensional examples

5.1.1. Test for smooth problem

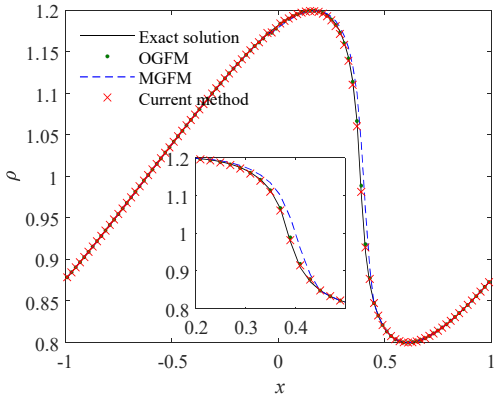
Firstly, a 1-dimensional smooth problem is taken from Fu & Shu [11] and analyzed with the two-medium methods. The initial conditions are given by

$$\begin{cases} \rho(x,0) = 1 + 0.2\sin(\pi x) \\ u(x,0) = \sqrt{\gamma}\rho(x,0) \\ p(x,0) = \rho(x,0)^{\gamma} \end{cases} \tag{33}$$

where $\gamma = 3$ such that the Euler equation of density ρ reduces to a Burgers equation

$$\rho_t + (\sqrt{\gamma}\rho^2)_x = 0. \tag{34}$$

Although this is a single-medium problem, we can solve it as a two-medium one by putting an imaginary interface inside the fluid that splits the flow domain into two parts. These two parts are occupied by two identical fluid components. Theoretically, it represents the same problem as the original one. Since the results are smooth before a certain time, the error of the numerical scheme can be evaluated easily. This example is used to test the stability and accuracy of the procedure when computing smooth solutions without medium jumps. Admittedly this example cannot be used to test difficulties associated with medium jumps, however it is still a good starting



(b) $t = 0.4$

Figure 6: Comparison of density results from the exact solution (solid line), the current scheme (red cross), OGFM (green dots) and MGFM (blue dashed line) with $\gamma = 3$ and initial conditions given by Eq. (33).

point to test the performance of an interface treating procedure. The computational domain is taken as $I = [-1, 1]$ with the periodic boundary condition in the simulation. The interface is initially located at $x = -0.8$, and both of the two fluids' initial conditions are given by Eq. (33). P^2 space is used, and the computational domain is discretized into 100 uniform cells. No limiter is used in this case since the problem is still smooth during the simulation. It should be noted that the entropy-fix procedure may lead to extra artificial density jump at the interface. Thus, the entropy-fix procedure is disabled in this case.

The results from different schemes are compared together at Fig. 6 for $t = 0.2$ and 0.4, respectively. The implementation of the MGFM follows [27] because the same RKDG method was used to discretize the Euler equation. Also, the HLLC Riemann solver is used for consistency in comparison with the current method. The results show that both of the current method and the OGFM can simulate the problem well except that there are small numerical fluctuations near the interface. These fluctuations comes from the loss of accuracy in the interface cell, and they have little influence on the overall performance. On the contrary, the results from the MGFM present more obvious conservative error and the error propagates with the maximum speed of the signal.

The errors $||\rho - \rho^h||_{L2}$ and the orders of accuracy of the total mass at $t = 0.2$ obtained with different methods are compared in Table 1. Because of the truncation of the high-order moments of the numerical solution in the interface cell and its

Table 1: Comparison of errors and accuracy orders of mass for the smooth problem discussed in section 5.1.1 with different cell numbers at $t = 0.2$.

Cell number	OGFM		MGFM		current scheme	
	error/10^{-3}	order	error/10^{-3}	order	error/10^{-3}	order
25	18.37	-	29.15	-	18.99	-
50	6.90	1.41	14.82	0.98	7.09	1.42
100	2.02	1.77	7.13	1.06	2.14	1.73
200	0.70	1.53	3.45	1.05	0.83	1.36

immediate neighbors, all the three methods should theoretically only have the order accuracy between 1 and 2 for smooth problems. However, we can observe significantly smaller errors on the same mesh for the current method and the OGFM than for the MGFM, as well as slightly higher order of accuracy.

5.1.2. Tests for discontinuous problems

(a) Sod problem

Now we consider a discontinuous problem proposed in [30] with the initial condition:

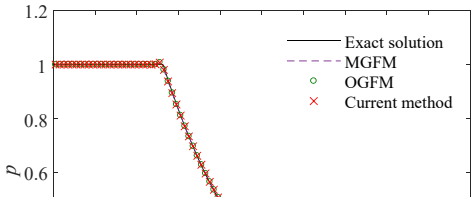
$$[\rho, u, p, \gamma] = \begin{cases} [1, 0, 1, 1.4] & \text{for } x \leq 0, \\ [0.125, 0, 0.1, 1.4] & \text{for } x > 0. \end{cases} \tag{35}$$

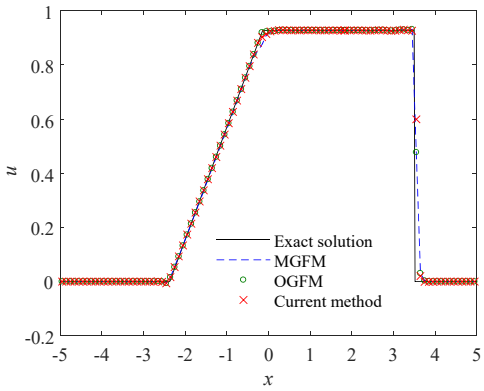
This is a simple 1-dimensional Riemann problem conducted by two polytropic gas with the same entropy. Due to the presence of the discontinuity, the entropy-fix procedure is essential to get desirable results. Besides, we have used the positivity-preserving limiter [39] to keep the simulation stable. The computational domain is I =[-5,5] and the results obtained with the current method, the OGFM and the MGFM at $t = 2.0$ are presented in Fig. 7 and compared with the exact solution. In this case, each wave in the Riemann problem is well simulated with all the three methods and agrees with the exact solution excellently. With the entropy-fix technique, even the density distribution near the contact discontinuity is quite good.

(b) Riemann problem between gas and water

The second 1-dimensional example is taken from Liu et al. [22] to test the treatments of the interface between two mediums with larger density and stiffness ratios. It represents a 1-dimensional underwater explosion problem. The computational domain is $I = [0, 1]$ and the initial conditions are non-dimensionalized as follows

$$[\rho, u, p, \gamma, P_w] = \begin{cases} [1.63, 0, 7.81 \times 10^4, 1.4, 0] & \text{for } x \leq 0.5, \\ [1, 0, 1, 7.15, 3309] & \text{for } x > 0.5. \end{cases} \tag{36}$$





(c) Velocity

Figure 7: Comparison of density(a), pressure(b) and velocity (c) results from the exact solution (solid line), the current scheme (cross symbols), OGFM (circles) and MGFM (dash line) for the Sod problem at $t = 2.0$.

The results of simulated density, pressure and velocity at $t = 1 \times 10^{-3}$ obtained with the current method, the OGFM and MGFM are shown in Fig. 8 with the solid lines indicating the exact solutions. The present method successfully captures the evolution of the shock wave in water, the rarefaction wave in gas, and the material interface. Since we have not used any slope limiter such as the TVB limiter, there are small oscillations on the curves in water. Besides, the current method performs better in the density distribution near the contact discontinuity than the MGFM.

(c) Shock-interface interaction

The final 1-dimensional example is also taken from Liu et al. [22] which is intended to show the superiority of the MGFM over OGFM. A strong shock with $p_L/p_R = 100$ impacts on a gas-gas interface. As shown in Liu et al. [22], the OGFM failed in simulating the movement of the interface because of its inappropriate assumptions. Thus, we only compare the results from the current method and the MGFM. The computational domain is $I = [0, 1]$ and the initial conditions are as follows

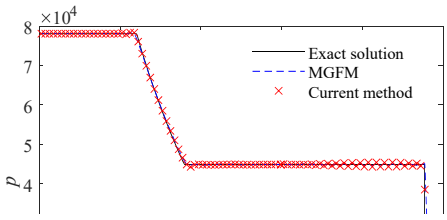
$$[\rho, u, p, \gamma] = \begin{cases} [0.3884, 27.1123, 100, 5/3] & \text{for } 0 < x < 0.3, \\ [0.1, 0, 1.0, 5/3] & \text{for } 0.3 \leq x < 0.4, \\ [1.0, 0, 1.0, 1.4] & \text{for } 0.4 \leq x < 1.0. \end{cases} \tag{37}$$

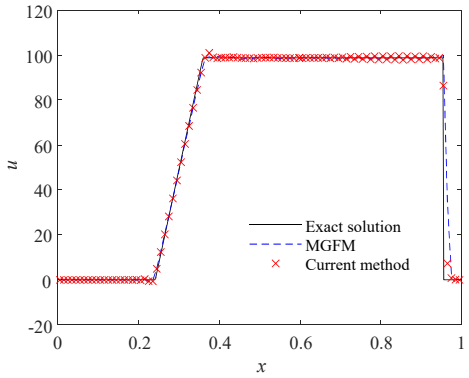
This case formulated the situation of a strong shock impacting on a gas-gas interface from the lighter gas. Similar to the previous case, only the positivity-preserving limiter is used here. The comparisons of simulated results obtained with the current method, the MGFM, and exact solution are shown in Fig. 9. Good agreement can be observed for both of the numerical results. Compared with the MGFM, the current method predicts better the right shock, while the left shock speed is lower than that of the exact solution and the MGFM. These mismatches can also be found in Qiu et al. [27] and might be attributed to the large entropy ratio across the incident shock.

5.2. 2-dimensional examples

5.2.1. Air bubble expanding in water

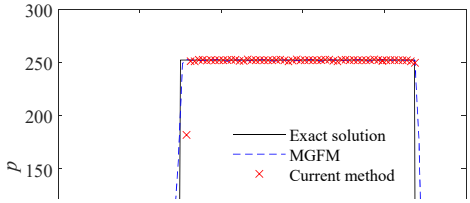
The first 2-dimensional case is to simulate an air bubble expanding in water. The 3-dimensional case of this problem has a lot of applications, such as the underwater explosion and cavitation bubbles. The case is taken from Shyue [28] and the 1-dimensional results with fine enough mesh are extracted as the reference solution. The computational domain is taken as $I = [0, 0.5] \times [0, 0.5]$ and is discretized into 100×100 cells. The results of the other three quadrants are obtained by symmetry.

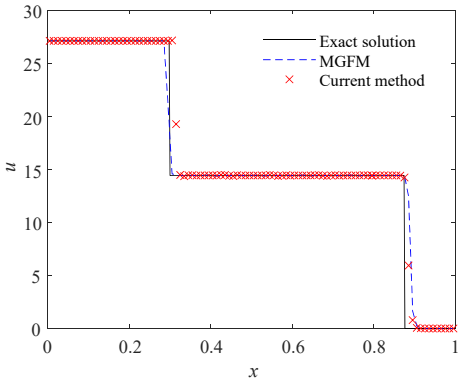




(c) Velocity

Figure 8: Density (a), pressure (b) and velocity (c) obtained with the current method (cross symbols) and comparison to the exact solution (solid line) for the 1-dimensional underwater explosion problem at $t = 0.001$.





(c) Velocity

Figure 9: Comparison of density(a), pressure(b) and velocity (c) results from the exact solution (solid line), the current scheme (cross symbols) and MGFM (dash line) for the shock-interface interaction problem at $t = 0.03$.

Table 2: Comparison of errors and accuracy orders of mass for the internal air for the problem discussed in section 5.2.1 with different cell numbers at $t = 0.058$.

Cell number	MGFM		current scheme	
	error/10^{-4}	order	error/10^{-4}	order
25×25	11.84	-	4.71	-
50×50	5.77	1.04	2.31	1.03
100×100	2.84	1.02	1.10	1.07
200×200	1.45	0.97	0.55	1.01

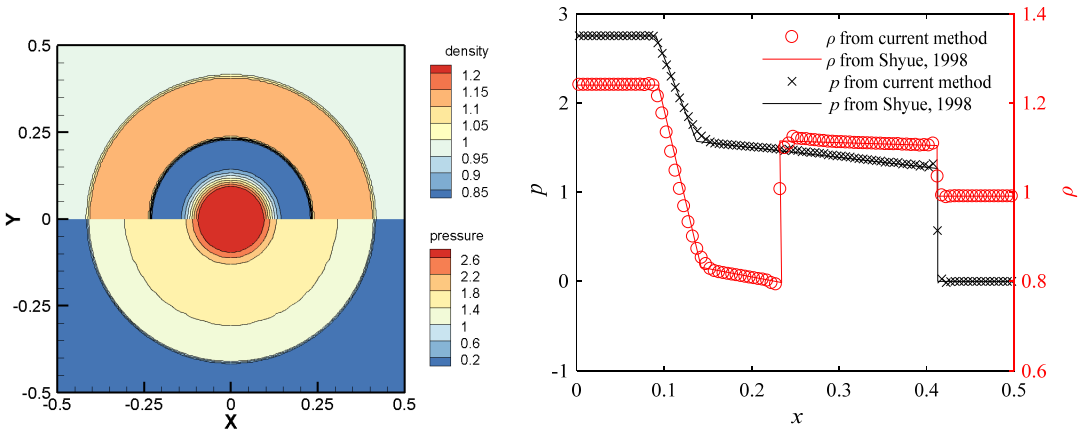
Table 3: Comparison of errors and accuracy orders of mass for the surrounding water for the problem discussed in section 5.2.1 with different cell numbers at $t = 0.058$.

Cell number	MGFM		current scheme	
	error/10^{-4}	order	error/10^{-4}	order
25×25	13.54	-	4.79	-
50×50	7.00	0.98	2.40	0.99
100×100	3.56	0.98	1.21	0.99
200×200	1.81	0.98	0.61	0.99

The initial conditions are given by

$$(\rho, u, v, p, \gamma, P_w) = \begin{cases} (1.241, 0, 0, 2.753, 1.4, 0) & \text{for } x^2 + y^2 \leq 0.2^2, \\ (0.991, 0, 0, 3.059e^{-4}, 5.5, 1.505) & \text{otherwise.} \end{cases} \tag{38}$$

The impenetrable boundary conditions are applied to all four boundaries. The density and pressure results are compared with the convergence results reproduced from Shyue [28] in Fig. 10 at $t = 0.058$. This problem is difficult since there is a continuous pressure gradient pointing to the bubble center near the material interface while the bubble is expanding. Thus, the material interface is always subject to a ramping wave, which makes the simulation more complicated. The two sets of results agree well except for the small numerical oscillations behind the shock wave. To evaluate the conservativeness of the current method, we compare the conservative errors of two fluid components with different mesh refinements with those obtained with the MGFM in Tab. 2 and Tab. 3, respectively. We can see that both the current method and the MGFM have the first order of accuracy, but the current method can obtain smaller conservative errors with the same grid. Compared with the smooth problem discussed in section 5.1.1, the reason for losing accuracy is because of the presence of "real" discontinuity and the employment of the level-set method.



(a) (b)

Figure 10: Results of the 2-dimensional underwater explosion problem taken from Shyue [28] at $t = 0.058$ for the density and pressure. Subfigure (a) shows the color contours of density (upper half) and pressure (lower half) while subfigure (b) shows the comparisons of the present results and those from Shyue [28].

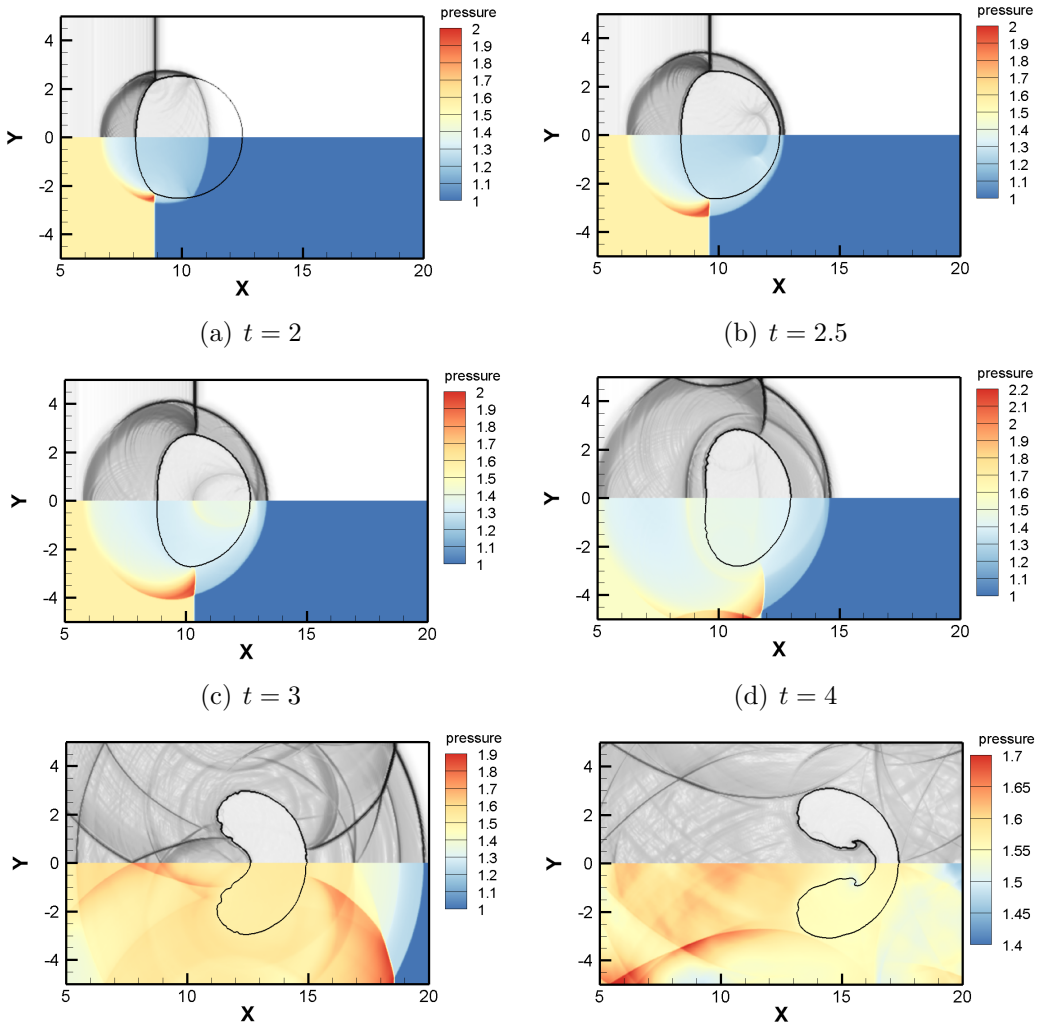
5.2.2. Air-helium bubble impacted by a shock

In this case, a 2-dimensional helium bubble immersed in air is impacted by a air shock with a Mach number of 1.22. The case has been well simulated to evaluate the numerical schemes by several authors, e.g. [22, 29, 34]. The initial conditions for the problem is taken from Wang & Shu [34] and is given by

$$(\rho, u, v, p, \gamma) = \begin{cases} (1, 0, 0, 1, 5/3) & \text{for } (x-10)^2 + y^2 \le 2.5^2, \\ (1.3764, 0.394, 0, 1.5698, 1.4) & \text{for } x \le 6, \\ (1, 0, 0, 1, 1.4) & \text{for else.} \end{cases} \tag{39}$$

In this paper, the computational domain is taken as $I = [0, 20] \times [0, 5]$ with the inlet boundary condition applied on the left boundary, the impenetrable boundary condition applied on the upper and lower boundaries, and the non-reflecting boundary condition is applied on the right boundary. In this and the next 2-dimensional examples, only the positivity-preserving limiter [39] is used to stabilize the simulation. The computational domain is discretized into 600×150 mesh cells. The lower half fluid domain is recovered by symmetry. The results are given in Fig. 11 for $t = 2, 2.5, 3, 4, 8$, and 13 in which the upside half figures represent the numerical Schlieren images and the lower half figures show the pressure distributions.

The current scheme captures all the features reported in the papers mentioned



(a) $t = 2$ (b) $t = 2.5$

(c) $t = 3$ (d) $t = 4$

(e) $t = 8$ (f) $t = 13$

Figure 11: Numerical Schlieren image (upper parts) and pressure distributions (lower parts) of an air-helium bubble impacted by a shock at $t = 2, 2.5, 3, 4, 8$, and 13, which are discussed in section 5.2.2.

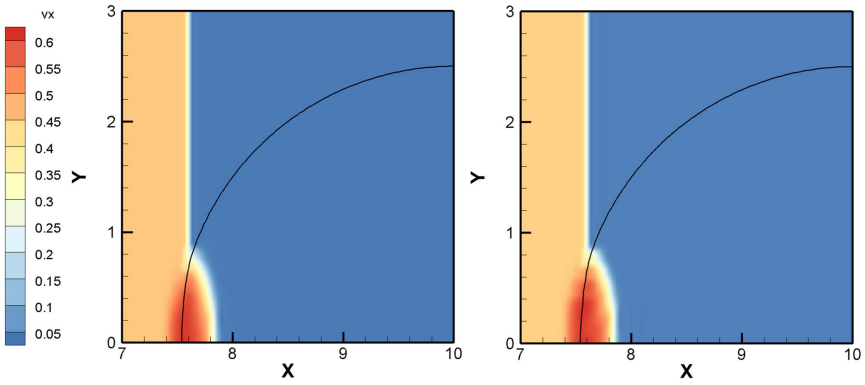
above, e.g. the reflection and transmission of incident waves on the bubble interface, the jet development of the bubble and so on. It should be noted that the bubble surface is not smooth anymore and the oscillation grows after $t = 8.0$. It should be attributed to amplification of the numerical error by the Kelvin-Helmholtz instability, which has also been shown by the results of [5] and [29].

Another comparison is made for the early interactions between the shock and the bubble obtained with the MGFM and our weighted GFM. The contour plots of the x-component velocity u at $t = 1.11$ and 1.58 are shown in Fig. 12. Firstly, it is observed that there are velocity fluctuations along the bubble interface in the results obtained with the MGFM at $t = 1.11$ when the shock wave just impacts on the bubble. Subsequently it leads to the oscillation of the interface, as shown in the results at $t = 1.58$. By contrast, the horizontal velocity distribution is quite smooth when calculated with the current method, which can be attributed to the ability of the current method in considering the exact location of the interface inside the interface cell and to transit the system properties smoothly. The other difference between the two sets of results is that the horizontal velocity near the interface is smeared to generate a band of 4-5 cells in the MGFM results, while our method gives sharper results.

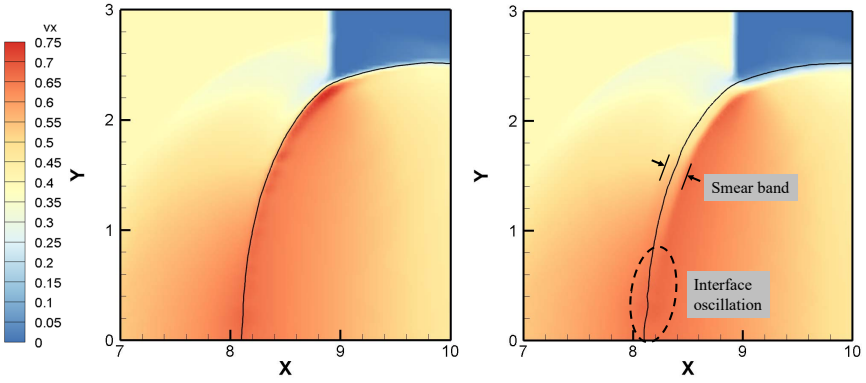
We also examine the conservativeness of the current method by plotting the error curve of the total helium mass inside the bubble in Fig. 13, and the results from the MGFM is added for comparison. There are two regions that the history curves increase suddenly for both methods and are denoted as region A and region B, respectively, as shown in the figure. Region A corresponds to the period when the shock impacts the left side of the bubble, and strong interaction between them leads to the sharp increase of the conservative error. Similarly, region B corresponds to when the reflected shock from the upper boundary reaches the bubble. During these two regions, the current method performs similarly to the MGFM. However, for the residual regions, the error curve's increase rate from the current method is observably lower than that of the MGFM, because most bubble interface moves in a smooth pressure field where our new method has a better performance.

5.2.3. Underwater air bubble impacted by a strong shock

The final 2-dimensional case is similar to the previous one with a much higher density ratio at the fluid interface, which poses significant challenges to the method's robustness. A circular gas bubble is initially immersed in water and then impacted by a strong shock. This problem has been investigated in [4, 20, 22, 27]. Similar to the air-helium bubble case, only half of the flow domain is simulated here. The computational domain is taken as $I = [0, 15] \times [0, 5]$, and is discretized into 600×240



(a) $t = 1.11$



(b) $t = 1.58$

Figure 12: Comparison between the early interaction of a shock and an air-helium bubble obtained with the current method (left) and the MGFM (right) for the case discussed in section 5.2.2 at $t = 1.11$ and 1.58. The color contour represents the velocity component in the x direction.

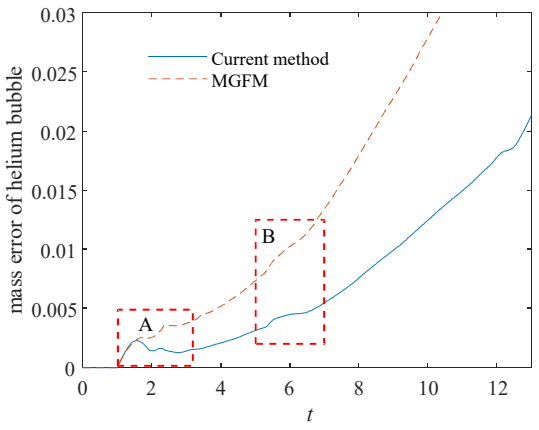


Figure 13: Comparison of the conservativeness errors of the mass of helium bubble obtained with the current method and the MGFM.

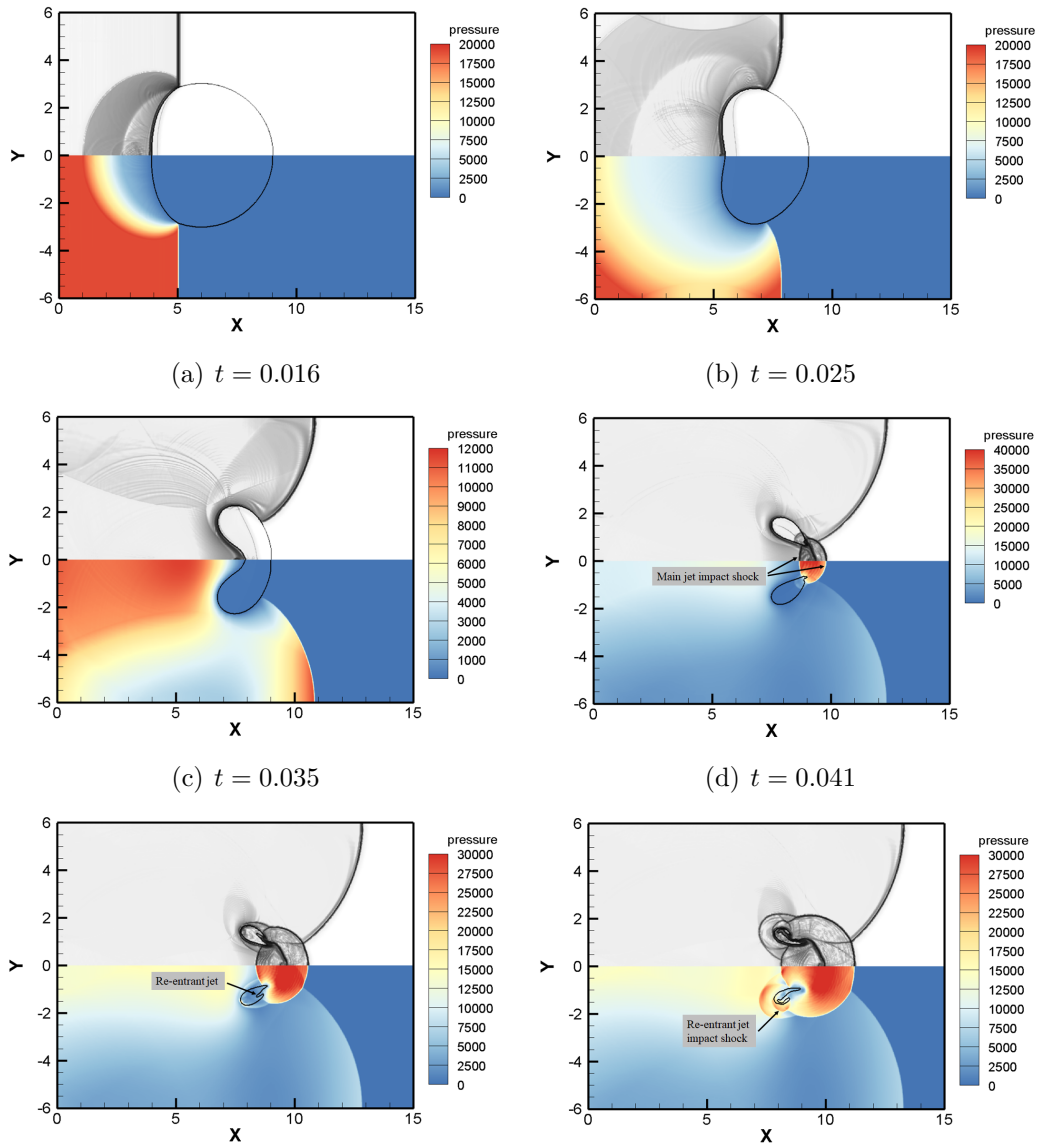
square cells. The left boundary is taken as non-reflecting while the other three boundaries are set to be impenetrable. The non-dimensionalized initial state for the water and the internal air is given by

$$(\rho, u, v, p, \gamma, P_w) = \begin{cases} (0.001, 0, 0, 1, 1.4, 0) & \text{for } (x-6)^2 + y^2 \leq 3^2, \\ (1, 60.3267, 0, 19000, 7, 3319) & \text{for } x \leq 2, \\ (1, 0, 0, 1, 7, 3319) & \text{for else.} \end{cases} \tag{40}$$

The simulated results are given in Fig. 14, with the lower half of the flow domain recovered by symmetry. We can observe the nonlinear interaction of the strong shock and the air bubble, including the reflected rarefaction wave (Fig. 14(a)), the development of high-speed jet (Fig. 14(c)), and jet impact and shock formation (Fig. 14(d)). Because of the absence of the TVB limiter, some numerical oscillations can be found in the results, but more evolution details are reserved. For example, a re-entrant jet forms due to the main jet impact pressure, and then it impacts on the bubble surface again to form another shock in water, as shown in Fig. 14(d) – (f).

6. Concluding remarks

In this paper, a new sharp interface treating method is implemented to resolve the compressible two-medium problem with the RKDG method. By analyzing the wave structure in a multi-medium Riemann problem inside an interface cell, the different portions are solved with different methods, and the new cell state vector is taken as



(a) $t = 0.016$ (b) $t = 0.025$

(c) $t = 0.035$ (d) $t = 0.041$

(e) $t = 0.043$ (f) $t = 0.045$

Figure 14: Numerical Schlieren image (upper parts) and pressure distributions (lower parts) of a water-air bubble impacted by a strong shock at $t = 0.016, 0.025, 0.035, 0.041, 0.043$, and 0.045, which are discussed in section 5.2.3.

their convex combination with carefully chosen weights. Then, the proposed method is extended to a 2-dimensional problem with the linear interface recovered from the level-set function that determines the Riemann problem. The steps to calculate the cell edge fluxes are presented in detail with explanations. Even though we considered only rectangular meshes in this paper, it is straightforward to extend the proposed method to arbitrary triangulation because of the flexibility of the RKDG method. With a well-developed RKDG solver for the Euler equations and the level-set equation in triangular meshes, the only point that one should pay further attention is to make an analogy to volume/area and flux calculation of interface cells described in section 3.2.2.

The proposed method is an extension of the GFM and inherits its simplicity in implementation in multi-dimensional problems. It could be regarded as a combination of the OGFM and the MGFM. With the simulations of several benchmark problems, the proposed method exhibits better accuracy for smooth problems and better capability in resolving the interaction of a strong shock and the material interface. Besides, a smooth transition of the system property is ensured in the proposed method by considering the exact location of the interface inside an interface cell, which reduces the interface oscillations where the shock front is almost parallel to the interface in the shock-interface interaction problem. When the problem involves three or more fluid components, the present method requires additional efforts. For example, there exist multiple level-set functions that need to be resolved, and the interface cells may encounter a multi-cut situation in the construction of in-cell Riemann problems and the area calculation. At last, similar to the GFMs [9, 12, 21, 36], the present method is potentially applicable to the fluid-solid interface problem with proper modifications.

Acknowledgments

This work of the first and third authors are supported by the National Natural Science Foundation of China (grant numbers 51879050, 51925904), the China Postdoctoral Science Foundation (grant numbers 2016M600244 and 2018T110271), and the Heilongjiang Postdoctoral Fund (grant numbers LBH-Z16043, LBH-TZ07). The work of the second author is supported by NSF grants DMS-1719410 and DMS-2010107, and AFOSR grant FA9550-20-1-0055.

References

[1] R. W. Anderson, V. A. Dobrev, T. V. Kolev, R. N. Rieben, and V. Z.Tomov. High-Order Multi-Material ALE Hydrodynamics. *SIAM Journal on Scientific*

Computing, 40(1):B32–B58, 2018.

[2] P. T. Barton, B. Obadia and D. Drikakis A conservative level-set based method for compressible solid/fluid problems on fixed grids. *Journal of Computational Physics*, 39:7867–7890, 2011.

[3] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(2):235–394, 1992.

[4] C.-H. Chang, and M.-S. Liou. A robust and accurate approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme. *Journal of Computational Physics*, 225:840-873, 2007.

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

[6] Y. D. Cheng and C.-W. Shu. A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *Journal of Computational Physics*, 223:398–415, 2007.

[7] B. Cockburn and C.-W. Shu. The rungekutta discontinuous Galerkin method for conservation laws v: Multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[8] B. Cockburn, S.-Y. Lin, and 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):90–113, 1989.

[9] R. P. Fedkiw. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *Journal of Computational Physics*, 175:200–224, 2002.

[10] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152:457–492, 1999.

[11] G. Fu and C.-W. Shu. A new trouble-cell indicator for discontinuous Galerkin methods for hyperbolic conservation laws. *Journal of Computational Physics*, 347:305–327, 2017.

[12] L. Ge, A.-M. Zhang, and S.-P. Wang. Investigation of underwater explosion near composite structures using a combined RKDG-FEM approach. *Journal of Computational Physics*, 404, 2020.

[13] J. Glimm, M. J. Graham, J. Grove, X. L. Li, T. M. Smith, D. Tan, F. Tangerman, and Q. Zhang. Front tracking in two and three dimensions. *Computers & Mathematics with Applications*, 35(7):1–11, 1998.

[14] J. Glimm, J. W. Grove, X. Lin Li, K.-M. Shyue, Y. N. Zeng, and Q. Zhang. Three-dimensional front tracking. *SIAM Journal on Scientific Computing*, 19 (3):703–727, 1998.

[15] J. Glimm and O. A. McBryan. A computational model for interfaces. *Advances in Applied Mathematics*, 6(4):422–435, 1985.

[16] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001.

[17] N. A. Hawker and Y. Ventikos. Interaction of a strong shockwave with a gas bubble in a liquid medium: a numerical study. *Journal of Fluid Mechanics*, 701: 59–97, 2012.

[18] C. W. Hirt and B. D. Nichols. Volume of fluid method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[19] M. J. Ivings, D. M. Causon, and E. F. Toro. On Riemann solvers for compressible liquids. *International Journal for Numerical Methods in Fluids*, 28:395–418, 1998.

[20] K. Kitamura and T. Nonomura Simple and robust HLLC extensions of two-fluid AUSM for multiphase flow computations. *Computers & Fluids*, 100:321–335, 2014.

[21] T. G. Liu, C. L. Feng, and L. Xu. Modified ghost fluid method with acceleration correction (MGFM/AC). *Journal of Scientific Computing*, 81:1906–1944, 2019.

[22] T. G. Liu, B. C. Khoo, and K. S. Yeo. Ghost fluid method for strong shock impacting on material interface. *Journal of Computational Physics*, 190:651–681, 2003.

[23] T. G. Liu, W. F. Xie, and B. C. Khoo. The modified ghost fluid method for coupling of fluid and structure constituted with hydro-elasto-plastic equation of state. *SIAM Journal on Scientific Computing*, 30(3):1105–1130, 2008.

[24] T. Nguyen Duy and T. Hino. An improvement of interface computation of incompressible two-phase flows based on coupling volume of fluid with level-set methods. *International Journal of Computational Fluid Dynamics*, 34(1):75–89, 2020.

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

[26] A. K. Pandare, J. Waltz and J. Bakosi A reconstructed discontinuous Galerkin method for multi-material hydrodynamics with sharp interfaces. *International Journal for Numerical Methods in Fluids*, 92(8):874–898, 2020.

[27] J. Qiu, T. Liu, and B. C. Khoo. Simulations of compressible two-medium flow by Runge-Kutta discontinuous Galerkin methods with the ghost fluid method. *Communications in Computational Physics*, 3:479–504, 2008.

[28] K.-M. Shyue. An efficient shock-capturing algorithm for compressible multicomponent problems. *Journal of Computational Physics*, 142(1):208–242, 1998.

[29] K. K. So, X. Y. Hu, and N. A. Adams. Anti-diffusion interface sharpening technique for two-phase compressible flow simulations. *Journal of Computational Physics*, 231:4304–4323, 2012.

[30] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1987.

[31] Z.-L. Tian, Y.-L. Liu, A.-M. Zhang, and L. Tao. Energy dissipation of pulsating bubbles in compressible fluids using the Eulerian finite-element method. *Ocean Engineering*, 196:106714, 2020.

[32] E. F. Toro. Riemann solvers and numerical methods for fluid dynamics. *Springer*, Berlin, 2009.

[33] C. Wang, T.G. Liu, and B.C. Khoo. A real ghost fluid method for the simulation of multimedium compressible flow. *SIAM Journal on Scientific Computing*, 28: 278–302, 2006.

[34] C. Wang and C.-W. Shu. An interface treating technique for compressible multimedium flow with Runge-Kutta discontinuous Galerkin method. *Journal of Computational Physics*, 229:8823–8843, 2010.

[35] H. L. Wen, C. H. Yu, and T. W. H. Sheu. On the development of LS-assisted VOF method for incompressible interfacial flows. *Journal of Computational Physics*, 406:109188, 2020.

[36] W. F. Xie, Y. L. Young, T. G. Liu, and B. C. Khoo. Dynamic response of deformable structures subjected to shock load and cavitation reload. *Computational Mechanics*, 40(4):667–681, 2007.

[37] L. Xu, C. L. Feng, and T. G. Liu. Practical techniques in ghost fluid method for compressible multi-medium flows. *Communications in Computational Physics*, 20:619–659, 2016.

[38] D. L. Youngs. *Numerical Methods in Fluid Dynamics*. Academic Press, New York, 1982.

[39] X. Zhang and C.-W. Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918 – 8934, 2010.