Journal of Computational Physics 445 (2021) 110597

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A high order conservative finite difference scheme for compressible two-medium flows

Feng Zheng [a], Chi-Wang Shu [b,1], Jianxian Qiu [c,*,2]

[a] College of Mathematics and Informatics, Fujian Normal University, Fuzhou, Fujian 350117, PR China
[b] Division of Applied Mathematics, Brown University, Providence, RI 02912, USA
[c] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling & High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, PR China

ARTICLE INFO

Article history:
Available online 30 July 2021

Keywords:
Finite difference
Conservative method
Two-medium flows

ABSTRACT

In this paper, a high order finite difference conservative scheme is proposed to solve two-medium flows. Our scheme has four advantages: First, our scheme is conservative, which is important to ensure the numerical solution captures the main features properly. Second, our scheme directly applies the WENO interpolation method to the primitive variables so that it can maintain the equilibrium of velocity and pressure across the interface, which is very helpful to obtain a non-oscillatory solution. Third, the usage of nodal values enables us to manipulate algebraic functions easily. Fourth, the scheme can maintain high order accuracy when the solution is smooth. Extensive numerical experiments are performed to verify the high resolution and non-oscillatory performance of this new scheme.

1. Introduction

The computation of two-medium flows is one of the most popular issues in computational fluid dynamics (CFD). It is relevant to many applications including hydrodynamics, aeronautics, material science, and so on. The main difficulty for the computation of two-medium flow is to maintain the equilibrium of velocity and pressure across the material interface, which is the property of the physical interface discontinuities. If the interface is not properly treated numerically, non-physical oscillations will occur, and such oscillations cannot be eliminated by using high resolution methods, such as the total variation diminishing (TVD) method and the weighted essentially non-oscillatory (WENO) method [24]. They even appear when we construct first order schemes.

Numerical methods for the two-medium flow can be divided into two categories: one is the shock-capturing method, and the other is the interface tracking method. An advantage for the shock-capturing method is its simplicity and easiness in its extension to multi-dimensions. The interface is allowed to diffuse numerically and is not explicitly tracked. Abgrall proposed a quasi-conservative scheme based on the γ-law model [1]. Then, Shyue extended the method to more general equations of state [27–29]. Allaire et al. introduced the usage of volume fraction and constructed a five-equation model for the simulation [4]. Abgrall and Saurel further applied the method to different numerical fluxes for multi-phase flows [3,23]. Although the shock-capturing method has achieved great success, it still has some drawbacks: First, the method diffuses the interface,

* Corresponding author.
E-mail addresses: fzbz200808-31@163.com (F. Zheng), chi-wang_shu@brown.edu (C.-W. Shu), jxqiu@xmu.edu.cn (J. Qiu).

[1] Research is partially supported by AFOSR grant FA9550-20-1-0055 and NSF grant DMS-2010107.
[2] Research is partially supported by NSFC grant 12071392 and Science Challenge Project (China), No. TZ 2016002.

https://doi.org/10.1016/j.jcp.2021.110597
0021-9991/

so it is not very clear where the interface is. Second, in order to avoid the jump near the interface, the intermediate state of either the physical parameters or the volume fraction would be introduced, which is inconsistent with what is really happening in physics. Comparing with the shock-capturing method, a sharp interface can be obtained by using the interface tracking method, where the level set method is used to track the interface [18]. Among all methods of this type, the ghost fluid method (GFM) with the isobaric fix is undoubtedly the most successful [8,9]. The method is essentially only solving single medium fluids through a Riemann solver at the interface (which is determined by the level set function) and defining the value of the ghost fluids across the interface. The interface is not explicitly tracked, its location is automatically determined by a level set function, hence the extension of this method to multi-dimensions becomes fairly straightforward. Liu et al. proposed a modified GFM (MGFM) which improves upon the original GFM [16]. The GFM and MGFM work well for strong shock impedance matching problems. Later, Wang et al. proposed the real GFM method focusing on simultaneous influence of different fluids [31]. There are also other similar methods, such as the interface treatment method [7], the simple single fluid algorithm [2], the path-conservative schemes [33] and so on. The drawback of these methods is that the scheme is not conservative, so theoretically we are not assured of the convergence to weak solutions from the numerical solutions. We refer to [2] for a good review of these methods for multi-medium and multi-phase flows.

Our goal is to design a finite difference scheme that is conservative and has a sharp and non-oscillatory interface. Conservative schemes have many advantages in solving hyperbolic problems. The most important property is that the numerical solution will converge to a weak solution as long as it converges (the Lax-Wendroff theorem). However, for the two-medium problem, conservative schemes tend to give oscillatory results if no special care is taken [32]. In [20], the authors designed a discontinuous Galerkin (DG) method [5] to solve the two-medium problem by using the classical DG scheme away from the interface and developing a DG scheme specially for treating the moving interface in one space dimension. High resolution and sharp interface results were achieved. Similar idea is used in [11]. In this paper we use a finite difference framework. Traditional conservative finite difference schemes perform reconstructions on the fluxes [25,26,13]. As is well known, across material interfaces, density will have a discontinuity, but velocity and pressure remain continuous. If the reconstruction is performed on the fluxes or on the conserved variables, then all components will have discontinuities, hence the approximation to velocity and pressure, which are nonlinear functions of the conserved variables or fluxes, will be poor. Moreover, traditional local characteristic decompositions to reduce oscillations has a reduced effect at the interface, whose characteristic structure is not clear since two different fluids are at the two sides of this interface. One possible way to get better results is to perform approximation directly on the primitive variables, namely density, velocity and pressure. However, for finite volume methods of order of accuracy higher than two, this is not possible, since we only have the information of cell averages of the conserved variables. For traditional finite difference schemes, this is not possible either, since the computation of numerical fluxes is through the reconstruction on the fluxes [25,26]. If we use the alternative formulation of high order finite difference schemes in [14], which is based on the high order flux expansion in [25], we will be performing interpolation, rather than reconstruction, on the conserved variables, not on the fluxes. In this paper, we modify the approach in [14] to use point values of the primitive variables, which are readily available from the point values of the conserved variables, to perform high order interpolation for obtaining the numerical fluxes. This would ensure non-oscillatory results for velocity and pressure from high order WENO interpolation. Near the interface, we will also use the specific information of the interface location as determined by the level set method to modify the approximation, based on the ideas in [20,11,15]. Both the distance function and the computational variables are updated at each time step by the third order TVD Runge-Kutta time discretization [25]. Conservative property and sharp and non-oscillatory interface can be obtained for our scheme. Extensive numerical experiments are performed for benchmark problems to verify the capability of the algorithm in obtaining non-oscillatory and high resolution solutions.

The organization of this paper is as follows. In section 2, we introduce our scheme in the one dimensional case. In section 3, we extend our scheme to the two dimensional case. In section 4, numerical benchmarks are shown to demonstrate the performance of our schemes. In section 5, we make some concluding remarks.

2. One-dimensional numerical schemes

The one-dimensional system for the compressible fluid can be written as follows:

$$U_t + F(U)_x = 0$$

where $U = (\rho, \rho u, E)^T$, $F(U) = (\rho u, \rho u^2 + p, u(E + p))^T$. Here ρ is the density, u is the velocity, E is the total energy, p is the pressure. To make the system closed, the equation of state (EOS) is required. In this paper, we mainly use the following EOS:

- γ-law:

$$E = \frac{1}{2}\rho u^2 + \frac{p}{\gamma - 1},$$

where γ is adiabatic index. The γ-law is used for gases in this paper.

- Tait EOS:

$$E = \frac{1}{2}\rho u^2 + \frac{p + \gamma \overline{p}}{\gamma - 1},$$

where $\gamma = 7.15$, $\overline{p} = 3.309 \times 10^8$ Pa. The Tait EOS is used for water in this paper.

2.1. Review of one-dimensional high order finite difference schemes for single medium

Considering the following one-dimensional scalar equation:

$$u_t + f(u)_x = 0$$

The computational domain is divided into N grid points: $a = x_0 < x_1 < \cdots < x_N = b$. For the nodal value u_i, we have the following semi-discrete scheme:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(\widehat{f}_{i+\frac{1}{2}} - \widehat{f}_{i-\frac{1}{2}})$$

where the numerical flux $\widehat{f}$ should satisfy the following condition:

$$\frac{\widehat{f}_{i+\frac{1}{2}} - \widehat{f}_{i-\frac{1}{2}}}{\Delta x} = f(u)_x\Big|_{x_i} + \mathcal{O}(\Delta x^5)$$

for a fifth order scheme. It has been shown in [25] that the following formula can guarantee the fifth order of accuracy:

$$\widehat{f}_{i+\frac{1}{2}} = \widehat{f}(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}) - \frac{1}{24}\Delta x^2 f_{xx}\Big|_{i+\frac{1}{2}} + \frac{7}{5760}\Delta x^4 f_{xxxx}\Big|_{i+\frac{1}{2}} \tag{1}$$

The first term in (1) is a monotone numerical flux in the scalar case or an appropriate Riemann-solver-based flux for systems. For example, we can use the Lax-Friedrichs flux:

$$\widehat{f}(u^-, u^+) = \frac{1}{2}(f(u^-) + f(u^+)) - \frac{1}{2}\alpha(u^+ - u^-)$$

where $\alpha = \max_u |f'(u)|$. $u^{\pm}_{i+\frac{1}{2}}$ are the right and left limits of u at $x_{i+1/2}$, which can be obtained from a WENO interpolation. The remaining terms can be approximated by simple central differences, as their effect on spurious oscillations is minimal due to the small coefficients involving at least Δx^2. We use

$$f_{xx}\Big|_{x_{i+\frac{1}{2}}} = \frac{-5f(u_{i-2}) + 39f(u_{i-1}) - 34f(u_i) - 34f(u_{i+1}) + 39f(u_{i+2}) - 5f(u_{i+3})}{48\Delta x^2} \tag{2}$$

and

$$f_{xxxx}\Big|_{x_{i+\frac{1}{2}}} = \frac{f(u_{i-2}) - 3f(u_{i-1}) + 2f(u_i) + 2f(u_{i+1}) - 3f(u_{i+2}) + f(u_{i+3})}{2\Delta x^4} \tag{3}$$

2.2. Description of the finite difference scheme for two-medium flows in 1D

Now, we describe the high order finite difference scheme for two-medium flows in the one dimensional case. In our scheme, we take the CFL number as 0.5.

The same as before, the computational domain is divided into N grid points: $a = x_0 < x_1 < \cdots < x_N = b$. We denote cell $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ as the corresponding cell of the node x_i. Here, we set $x_{i+\frac{1}{2}} = (x_i + x_{i+1})/2$. We denote the nodal value for fluid I as $U_i^{n,I}$ and the nodal value for fluid II as $U_i^{n,II}$. We introduce the distance function ϕ which is defined at the half nodes. By using this distance function ϕ, we can determine the location of the interface $x(t^n)$ at time level t^n. We define i_n as the sequence of the index of the cell containing the interface at time level t^n, then we have $x(t^n) \in [x_{i_n-\frac{1}{2}}, x_{i_n+\frac{1}{2}}]$. We can define the computational variable $\{U_i^n\}$, which will be the variables updated in time in a conservative fashion, in the following way:

$$\begin{aligned} &U_i^n = U_i^{n,I}, && \text{if} \quad i < i_n \\ &U_i^n = U_i^{n,II}, && \text{if} \quad i > i_n \\ &U_i^n = \alpha_i U_i^{n,I} + (1-\alpha_i)U_i^{n,II}, && \text{if} \quad i = i_n \end{aligned} \tag{4}$$

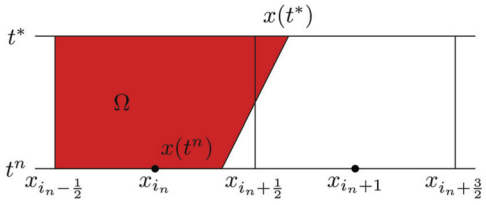

Fig. 1. Sketch for the interface stencil. Red: fluid I. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where α_i is the volume fraction of fluid I in the cell I_i. We set $\alpha_i = \frac{x(t^n)-x_{i-1/2}}{\Delta x}$.

Then, we can update the computational variable $\{U_i^n\}$ to the next time level t^*

$$U_i^* = U_i^n - \frac{\Delta t}{\Delta x}(\widehat{F}^n_{i+\frac{1}{2}} - \widehat{F}^n_{i-\frac{1}{2}}) \tag{5}$$

In order to guarantee high resolution when discontinuities appear and high precision in the smooth region, we need to design suitable numerical fluxes. For the fluxes far from the interface, they will be approximated by single fluid numerical fluxes whose explicit expressions are similar to (1). We denote these numerical fluxes as $\widehat{F}^H$. For the fluxes near the interface, in order to maintain the equilibrium of the velocity and pressure across the interface, we need to calculate the special low order fluxes, denoted as $\widehat{F}^L$, and we will use a combination of $\widehat{F}^H$ and $\widehat{F}^L$ to obtain the numerical fluxes. This combination will be performed carefully in order to ensure high order accuracy.

For the flux $\widehat{F}^H$, we have the following expression:

$$\widehat{F}^H_{i+\frac{1}{2}} = \widehat{F}(U^-_{i+\frac{1}{2}}, U^+_{i+\frac{1}{2}}) - \varphi_{i+\frac{1}{2}}\left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i+\frac{1}{2}}} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i+\frac{1}{2}}}\right) \tag{6}$$

where $\varphi_{i+\frac{1}{2}}$ is a discontinuity indicator at $x_{i+1/2}$ which can improve the resolution when discontinuities appear and maintain high order accuracy in smooth areas. The detailed steps to compute the discontinuity indicator is given in subsection 2.2.4. $\widehat{F}(U^-_{i+\frac{1}{2}}, U^+_{i+\frac{1}{2}})$ is the Lax-Friedrichs flux. Instead of computing the interpolated values $U^{\pm}_{i+1/2}$ from the point values of the conserved variables, we will interpolate the primitive variables using the WENO method. As we know, the variables near the two-medium interface satisfy the contact discontinuity condition. Density and physical parameters will jump, while velocity and pressure will keep continuous. As mentioned in the introduction, it is better to interpolate density, velocity and pressure directly. Therefore, based on the framework in [14], we perform the interpolation on the primitive variables, which are readily available from the nodal values of the conserved variables. This gives us better chance to obtain non-oscillatory results for velocity and pressure from high order WENO interpolation near the interface. As to the higher order derivative terms, we can apply the simple central difference method in a component by component fashion. It should be noted that the nodal values used in the interpolation are chosen with respect to the position of the interface. When calculating the fluxes at a location in fluid I, we will use nodal values of fluid I in the interpolation if they are defined, otherwise we will use the nodal values of fluid II; when calculating the fluxes at a location in fluid II, we will use nodal values of fluid II in the interpolation if they are defined, otherwise we use the nodal values of fluid I.

Next, we will describe the detailed steps to calculate the low order special flux $\widehat{F}^L$ near the interface.

First, by using the level set method, we can evolve the distance function ϕ^n to time level t^* and obtain ϕ^*. Then we can use ϕ^* to determine the interface position $x(t^*)$ at the new time level t^*. We can use the WENO method in a component by component fashion to determine the left and right states of the Riemann problem at the interface. The same as before, we interpolate the primitive variables instead of the conserved variables. The detailed steps of the level set method and WENO method are described in subsections 2.2.1 and 2.2.3.

We assume the interface $x(t^n)$ at time level t^n satisfies $x(t^n) \in [x_{i_n-1/2}, x_{i_n+1/2}]$, and the interface $x(t^*)$ at the next time level t^* satisfies $x(t^*) \in [x_{i_*-1/2}, x_{i_*+1/2}]$. We denote α_{i_*} as the volume fraction of fluid I in the cell I_{i_*}. The mixed cell I_{i_n} should not be calculated for a full time step. It is suggested that the cell should be merged with a neighboring cell, in order to avoid the small "cut-cell" problem. We will merge the cells in the following way:

- If $\alpha_{i_*} < 0.5$, then we choose the cells I_{i_*-1} and I_{i_*} as the interface stencil, and merge them to form two new interface cells: the cell $[x_{i_*-\frac{3}{2}}, x(t^n)]$ occupied by fluid I, and the cell $[x(t^n), x_{i_*+\frac{1}{2}}]$ occupied by fluid II;
- If $\alpha_{i_*} \geq 0.5$, then we choose the cells I_{i_*} and I_{i_*+1} as the interface stencil, and merge them to form two new interface cells: the cell $[x_{i_*-\frac{1}{2}}, x(t^n)]$ occupied by fluid I, and the cell $[x(t^n), x_{i_*+\frac{3}{2}}]$ occupied by fluid II.

Due to the CFL number and the way to generate the interface stencil, the interface stencil chosen above will contain the mixed cell both at the time level t^n and at the time level t^*. For simplicity, we assume cells I_{i_n} and I_{i_n+1} have been merged together, see Fig. 1. Then, we need to calculate the fluxes $\widehat{F}^L$ at the positions $x_{i_n\pm1/2}$ and $x_{i_n+3/2}$.

Regarding the flux at $x_{i_n-1/2}$, it is in the fluid I. Therefore, if $U^{n,I}_{i_n+1}$ is not defined, we will set $U^{n,I}_{i_n+1} = U^{n,I}_{i_n}$. Then we will use the nodal values $\{U^{n,I}_{i_n-3}, U^{n,I}_{i_n-2}, U^{n,I}_{i_n-1}, U^{n,I}_{i_n}, U^{n,I}_{i_n+1}, U^{n,II}_{i_n+2}\}$ to compute the numerical flux $\widehat{F}^L_{i_n-\frac{1}{2}}$ based on equation (6). Regarding the flux at $x_{i_n+3/2}$, it is in the fluid II. Similarly, if $U^{n,II}_{i_n}$ is not defined, we will set $U^{n,II}_{i_n} = U^{n,II}_{i_n+1}$. Then we will use the nodal values $\{U^{n,I}_{i_n-1}, U^{n,II}_{i_n}, U^{n,II}_{i_n+1}, U^{n,II}_{i_n+2}, U^{n,II}_{i_n+3}, U^{n,II}_{i_n+4}\}$ to compute the numerical flux $\widehat{F}^L_{i_n+\frac{3}{2}}$ based on equation (6). As to the flux $\widehat{F}^L_{i_n+\frac{1}{2}}$, it can be obtained by the conservation law, as described below in details.

For fluid I, similar to the ALE method, we have the following integrated conservation law:

$$\int_{t^n}^{t^*} \mathrm{dt} \int_{x_{i_n-\frac{1}{2}}}^{x(t)} \Big(U_t + F(U)_x\Big)\mathrm{dx} = 0 \tag{7}$$

According to the Green's formula, we have

$$\begin{aligned} &\int_{\partial\Omega} -U\mathrm{dx} + F(U)\mathrm{dt} \\ =&-\int_{x_{i_n-\frac{1}{2}}}^{x(t^n)} U\mathrm{dx} + \int_{(x(t^n),t^n)}^{(x(t^*),t^*)} -U\mathrm{dx} + F(U)\mathrm{dt} + \int_{x_{i_n-\frac{1}{2}}}^{x(t^*)} U\mathrm{dx} - \int_{t^n}^{t^*} F(U)\Big|_{x_{i_n-\frac{1}{2}}}\mathrm{dt} \\ =&0 \end{aligned} \tag{8}$$

where Ω is the control volume for fluid I. We denote the velocity of the fluid at the interface as u. Then, we can rewrite the flux $F(U)$ as

$$F(U) = u \cdot U + F_s$$

where $F_s = (0, p, u \cdot p)^T$. We also assume the location of the interface satisfies the following formula:

$$x(t) = x(t^n) + u(t - t^n)$$

Then, we can simplify the integration in (8):

$$\begin{aligned} &\int_{(x(t^n),t^n)}^{(x(t^*),t^*)} -U\mathrm{dx} + F(U)\mathrm{dt} \\ =&\int_{t^n}^{t^*} -u \cdot U + F(U)\mathrm{dt} \\ =&\Delta t F_s \end{aligned} \tag{9}$$

We define the numerical flux $\widehat{F_s}$:

$$\widehat{F_s} = (0, p, u \cdot p)^T.$$

The values of p and u in $\widehat{F_s}$ can be obtained through solving the Riemann problem $RP(U_L, U_R)$ at the interface. The left and right states U_L and U_R can be obtained by WENO interpolation using the primitive variables. Then, we have the average value for fluid I:

$$U^*_I = \frac{1}{x(t^*) - x_{i_n-\frac{1}{2}}}\Big(\int_{x_{i_n-\frac{1}{2}}}^{x(t^n)} U^n\mathrm{dx} - \Delta t(\widehat{F_s} - \widehat{F}^L_{i_n-\frac{1}{2}})\Big) \tag{10}$$

We notice that the mass of each component of the fluid is actually conserved at this stage, since the first component of the numerical flux $\widehat{F_s}$ is zero. We also take

$$\int_{x_{i_n-\frac{1}{2}}}^{x(t^n)} U^n\mathrm{dx} = \Delta x \alpha^n_{i_n} U^{n,I}_{i_n}.$$

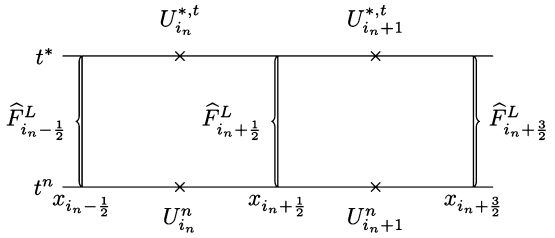

Fig. 2. Sketch for computing $\widehat{F}^L$.

Similarly, we can obtain the average values for fluid II:

$$U^*_{II} = \frac{1}{x_{i_n+\frac{3}{2}} - x(t^*)}\Big(\int_{x(t^n)}^{x_{i_n+\frac{3}{2}}} U^n \mathrm{d}x - \Delta t(\widehat{F}^L_{i_n+\frac{3}{2}} - \widehat{F}_s)\Big) \tag{11}$$

Now, we can define the temporary nodal values $U^{*,t}_{i_n}$ and $U^{*,t}_{i_n+1}$ according to the interface $x(t^*)$:

1. If $x(t^*) \in [x_{i_n-\frac{1}{2}}, x_{i_n+\frac{1}{2}}]$,

   then we define $U^{*,t}_{i_n} = \alpha^*_{i_n} U^*_I + (1-\alpha^*_{i_n})U^*_{II}$ and $U^{*,t}_{i_n+1} = U^*_{II}$, where $\alpha^*_{i_n} = \frac{x(t^*)-x_{i_n-\frac{1}{2}}}{\Delta x}$ is the volume fraction of fluid I in the cell I_{i_n}.
2. If $x(t^*) \in [x_{i_n+\frac{1}{2}}, x_{i_n+\frac{3}{2}}]$,

   then we define $U^{*,t}_{i_n} = U^*_I$ and $U^{*,t}_{i_n+1} = \alpha^*_{i_n+1}U^*_I + (1-\alpha^*_{i_n+1})U^*_{II}$, where $\alpha^*_{i_n+1} = \frac{x(t^*)-x_{i_n+\frac{1}{2}}}{\Delta x}$ is the volume fraction of fluid I in the cell I_{i_n+1}.

Now, we can compute the flux at the position $x_{i_n+1/2}$, see Fig. 2:
From the left part, we have:

$$U^{*,t}_{i_n} = U^n_{i_n} - \frac{\Delta t}{\Delta x}\left(\widehat{F}^L_{i_n+1/2} - \widehat{F}^L_{i_n-1/2}\right)$$

Then, we get:

$$\widehat{F}^L_{i_n+1/2} = \widehat{F}^L_{i_n-1/2} - \frac{U^{*,t}_{i_n} - U^n_{i_n}}{\frac{\Delta t}{\Delta x}} \tag{12}$$

From the right part, we have:

$$U^{*,t}_{i_n+1} = U^n_{i_n+1} - \frac{\Delta t}{\Delta x}\left(\widehat{F}^L_{i_n+3/2} - \widehat{F}^L_{i_n+1/2}\right)$$

Then, we get:

$$\widehat{F}^L_{i_n+1/2} = \frac{U^{*,t}_{i_n+1} - U^n_{i_n+1}}{\frac{\Delta t}{\Delta x}} + \widehat{F}^L_{i_n+3/2} \tag{13}$$

Finally, we take an average of formula (12) and formula (13) to compute the flux $\widehat{F}^L$:

$$\widehat{F}^L_{i_n+1/2} = \frac{1}{2}\left(\widehat{F}^L_{i_n-1/2} - \frac{U^{*,t}_{i_n} - U^n_{i_n}}{\frac{\Delta t}{\Delta x}}\right) + \frac{1}{2}\left(\frac{U^{*,t}_{i_n+1} - U^n_{i_n+1}}{\frac{\Delta t}{\Delta x}} + \widehat{F}^L_{i_n+3/2}\right) \tag{14}$$

Now, we can define the flux $\{\widehat{F}^n_{i+\frac{1}{2}}\}$ which will be used to evolve the computational variables:

$$\widehat{F}^n_{i+1/2} = \widehat{F}^H_{i+1/2} \qquad \text{for} \quad i \neq i_n - 1 \quad \& \quad i_n \quad \& \quad i_n + 1 \tag{15}$$

and

$$\widehat{F}^n_{i+1/2} = \widehat{F}^L_{i+1/2} + \varphi\left(\widehat{F}^H_{i+1/2} - \widehat{F}^L_{i+1/2}\right) \qquad \text{for} \quad i = i_n - 1, i_n, i_n + 1 \tag{16}$$

where $\varphi = \min(\varphi_{i_n-1/2}, \varphi_{i_n+1/2}, \varphi_{i_n+3/2})$. Here $\varphi_{i_n\pm1/2}, \varphi_{i_n+3/2}$ are the discontinuity indicators at $x_{i_n\pm1/2}$ and $x_{i_n+3/2}$.

Then, we can advance the computational variables based on the formula (5).

Now, we can define the nodal values $\{U_i^{*,I}\}$ and $\{U_i^{*,II}\}$ for fluid I and fluid II at time level t^* respectively. We assume the interface at time level t^* satisfies $x(t^*) \in [x_{i_*-1/2}, x_{i_*+1/2}]$. Then, we have

- When $i < i_*$,

$$U_i^{*,I} = U_i^* \tag{17}$$

- When $i > i_*$,

$$U_i^{*,II} = U_i^* \tag{18}$$

- When $i = i_*$,

(1) If $\alpha_{i_*}^* > 0.5$, then the cell I_{i_*} is a small cell for fluid II. Therefore, we take

$$\begin{aligned} U_{i_*}^{*,II} &= \varphi U_{i_*}^* + (1-\varphi) U_{II}^* \\ U_{i_*}^{*,I} &= \frac{U_{i_*}^* - (1-\alpha_{i_*}^*) U_{i_*}^{*,II}}{\alpha_{i_*}^*} \end{aligned} \tag{19}$$

where $U_{i_*}^*$ comes from formula (5), and U_{II}^* is the average value for fluid II which comes from formula (11).

(2) If $\alpha_{i_*}^* \le 0.5$, then the cell I_{i_*} is a small cell for fluid I. Therefore, we take

$$\begin{aligned} U_{i_*}^{*,I} &= \varphi U_{i_*}^* + (1-\varphi) U_I^* \\ U_{i_*}^{*,II} &= \frac{U_{i_*}^* - \alpha_{i_*}^* U_{i_*}^{*,I}}{1-\alpha_{i_*}^*} \end{aligned} \tag{20}$$

where $U_{i_*}^*$ comes from formula (5), and U_I^* is the average value for fluid I which comes from formula (10).

In the formula (19) and (20), we also take

$$\varphi = \min\{\varphi_{i_n-1/2}, \varphi_{i_n+1/2}, \varphi_{i_n+3/2}\}.$$

Remark 1. In formula (19), when the solution is smooth, the formula (5) which is used to compute the nodal value $U_{i_*}^*$ is high order accurate [25,14]. Therefore, we can maintain high order accuracy if the nodal value $U_{i_*}^{*,I}$ for fluid I and the nodal value $U_{i_*}^{*,II}$ for fluid II are close to the nodal value $U_{i_*}^*$. When discontinuity appears at the interface, U_{II}^* is the average value obtained by the formula specifically designed for the interface, so it is more likely to obtain a non-oscillatory solution if the nodal value $U_{i_*}^{*,II}$ is close to U_{II}^*. Therefore, we introduce the discontinuity indicator in formula (19) which approaches one in the smooth region and tends to zero when discontinuity appears. Similarly, we apply the same idea to the formula (20).

To summarize, we have the following algorithm:

Algorithm 1

Input: $U_i^{n,I}$, $U_i^{n,II}$, ϕ_i^n, Δt, Δx
Output: $U_i^{*,I}$, $U_i^{*,II}$, $\widehat{F}_{i+1/2}^n$, ϕ_i^*

1: Apply $U_i^{n,I}$ and $U_i^{n,II}$ to define the computational variables U_i^n.
2: Compute the fluxes $\widehat{F}^H$.
3: Update the distance function, and obtain ϕ_i^*.
4: Merge cells and compute the fluxes $\widehat{F}^L$.
5: Define the flux $\widehat{F}^n$.
6: Update the computational variables U_i^*
7: Define the nodal values $U_i^{*,I}$ for fluid I, and the nodal values $U_i^{*,II}$ for fluid II.

Now, we elaborate on the above ideas and implementation details.

2.2.1. Level set method

We associate the computational domain with the distance function ϕ, which satisfies the following expressions:

$$\phi_t + V_n|\nabla\phi| = 0 \tag{21}$$

where V_n is the normal velocity which can be obtained through solving the Riemann problem at the interface. We define the distance function at the half nodes. Then, we have the following formula

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = -(v^+_{i+\frac{1}{2}}\sqrt{\max((a^+)^2,(b^-)^2)} + v^-_{i+\frac{1}{2}}\sqrt{\max((a^-)^2,(b^+)^2)}) \tag{22}$$

where $v^+_{i+\frac{1}{2}} = \max(v_{i+\frac{1}{2}},0)$, $v^-_{i+\frac{1}{2}} = \min(v_{i+\frac{1}{2}},0)$, and $v_{i+\frac{1}{2}}$ is the normal velocity at $x_{i+\frac{1}{2}}$. The definition of $a^\pm$ and $b^\pm$ is similar. We take $a = \phi_x^-$ and $b = \phi_x^+$, where $\phi_x^\pm$ can be obtained by the WENO method.

2.2.2. Characteristic projection 1D

The WENO method is performed in the local characteristic fields. In systems of nonlinear equations, oscillations can develop in component-wise interpolation [24,21]. In this paper, we will use the local characteristic field decomposition. Because we perform the interpolation using primitive variables rather than conserved variables, we will use the left and right eigenvectors corresponding to the primitive variables for the characteristic projections. Considering the following quasi-linear form of the Euler equation:

$$W_t + A(W)W_x = 0$$

where

$$W = \begin{pmatrix}\rho\\ u\\ p\end{pmatrix} \qquad A(W) = \begin{pmatrix}u & \rho & 0\\ 0 & u & 1/\rho\\ 0 & \rho c^2 & u\end{pmatrix}$$

Here c refers to the sound speed. We then give the left and right eigenvector matrices of matrix $A(W)$ as:

$$R(W) = \begin{pmatrix}1 & 1 & 1\\ -\frac{c}{\rho} & 0 & \frac{c}{\rho}\\ c^2 & 0 & c^2\end{pmatrix} \qquad L(W) = \begin{pmatrix}0 & -\frac{\rho}{2c} & \frac{1}{2c^2}\\ 1 & 0 & -\frac{1}{c^2}\\ 0 & \frac{\rho}{2c} & \frac{1}{2c^2}\end{pmatrix}$$

Firstly, we use the left eigenvector matrices $L(W)$ to project the variables into the respective characteristic fields. Secondly, we interpolate the values in each characteristic fields. Finally, we use the right eigenvector matrices $R(W)$ to project the values back into the physical space [24]. Although it is more expensive computationally, more satisfactory results can be obtained.

2.2.3. WENO method

The WENO method is one of the important parts in our schemes. We need two different types of WENO method: one is for function values, and the other one is for derivative values. The detailed steps to obtain derivative values can be seen in [12]. Here we describe the procedure to obtain the value w at $x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ in fluid I using the WENO interpolation, where w denotes the interpolation variable.

1. Based on the small stencils $S_r = \{x_{i-2+r}, x_{i-1+r}, x_{i+r}\}, r = 0,1,2$ and a big stencil $S = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, we construct polynomials $p_r(x), r = 0,1,2$ and $q(x)$. We have:

$$\begin{aligned}
p_0(x) &= \frac{(x-x_{i-1})(x-x_i)}{2\Delta x^2}w_{i-2} - \frac{(x-x_{i-2})(x-x_i)}{\Delta x^2}w_{i-1} + \frac{(x-x_{i-2})(x-x_{i-1})}{2\Delta x^2}w_i\\
p_1(x) &= \frac{(x-x_i)(x-x_{i+1})}{2\Delta x^2}w_{i-1} - \frac{(x-x_{i-1})(x-x_{i+1})}{\Delta x^2}w_i + \frac{(x-x_{i-1})(x-x_i)}{2\Delta x^2}w_{i+1}\\
p_2(x) &= \frac{(x-x_{i+1})(x-x_{i+2})}{2\Delta x^2}w_i - \frac{(x-x_i)(x-x_{i+2})}{\Delta x^2}w_{i+1} + \frac{(x-x_i)(x-x_{i+1})}{2\Delta x^2}w_{i+2}\\
q(x) &= \frac{(x-x_{i-1})(x-x_i)(x-x_{i+1})(x-x_{i+2})}{24\Delta x^4}w_{i-2}\\
&\quad - \frac{(x-x_{i-2})(x-x_i)(x-x_{i+1})(x-x_{i+2})}{6\Delta x^4}w_{i-1}\\
&\quad + \frac{(x-x_{i-2})(x-x_{i-1})(x-x_{i+1})(x-x_{i+2})}{4\Delta x^4}w_i\\
&\quad - \frac{(x-x_{i-2})(x-x_{i-1})(x-x_i)(x-x_{i+2})}{6\Delta x^4}w_{i+1}
\end{aligned}$$

$$+\frac{(x-x_{i-2})(x-x_{i-1})(x-x_i)(x-x_{i+1})}{24\Delta x^4}w_{i+2}$$

The same as before, we use the position of the interface to decide the nodal values used in the interpolation. If we are interpolating the value w in fluid I, then we will use the nodal values of fluid I in the interpolation if they are defined, otherwise we use the nodal values of fluid II. Likewise, if we are calculating the value w in fluid II, then we will use nodal values of fluid II in the interpolation if they are defined, otherwise we use the nodal values of fluid I.

2. We find the linear weights, denoted as $\gamma_0, \gamma_1, \gamma_2$, such that

$$q(x)=\sum_{k=0}^{2}\gamma_k p_k(x)$$

for all possible nodal values w_i. Then, we can obtain:

$$\gamma_0=\frac{(x-x_{i+1})(x-x_{i+2})}{12\Delta x^2},\quad \gamma_2=\frac{(x-x_{i-2})(x-x_{i-1})}{12\Delta x^2},\quad \gamma_1=1-\gamma_0-\gamma_2.$$

3. We compute the smoothness indicator, which measures the smoothness of the function. We use the same recipe as in [13]:

$$\beta_r=\sum_{k=1}^{2}\int_{x_{i-1/2}}^{x_{i+1/2}}\Delta x^{2k-1}\Big(\frac{\partial^k}{\partial x^k}(p_r(x))\Big)^2\mathrm{dx},\quad r=0,1,2$$

The expression can be written out explicitly:

$$\begin{aligned}
\beta_0&=\frac{13}{12}(w_{i-2}-2w_{i-1}+w_i)^2+\frac{1}{4}(w_{i-2}-4w_{i-1}+3w_i)^2\\
\beta_1&=\frac{13}{12}(w_{i-1}-2w_i+w_{i+1})^2+\frac{1}{4}(w_{i-1}-w_{i+1})^2\\
\beta_2&=\frac{13}{12}(w_i-2w_{i+1}+w_{i+2})^2+\frac{1}{4}(3w_i-4w_{i+1}+w_{i+2})^2
\end{aligned}\tag{23}$$

4. Based on the smoothness indicator, we can compute the nonlinear weights:

$$\omega_r=\frac{\overline{\omega}_r}{\sum_k\overline{\omega}_k},\quad \overline{\omega}_k=\frac{\gamma_k}{(\beta_k+\varepsilon)^2}\tag{24}$$

where ε is a small number to avoid the denominator to become zero. Here, we set $\varepsilon=10^{-6}$. The final WENO expression is given by:

$$w=\sum_{k=0}^{2}\omega_k p_k(x)$$

2.2.4. Discontinuity indicator

The idea of the discontinuity indicator $\varphi_{i+\frac{1}{2}}$ comes from [34]. It can maintain high order accuracy in the smooth region and can achieve high resolution when discontinuities appear. In practice, we take density and pressure as the indicator variables and choose the smaller one. Before discussing the detailed steps to construct the discontinuity indicator, we would like to analyze the accuracy requirement of the indicator in the smooth case first.

We use the discontinuity indicator in three different places: (1) We use the indicator to compute the fluxes $\widehat{F}^H$, see formula (6); (2) We use the indicator to make a convex combination of the fluxes $\widehat{F}^H$ and the fluxes $\widehat{F}^L$, see formula (16); (3) We use the indicator to update the nodal values for fluid I and fluid II, see formula (19). Then, we will analyze the accuracy requirement respectively. We always assume the values at time level t^n are accurate.

Proposition 2.1. *The numerical flux (6) satisfies*

$$\frac{\widehat{F}_{i+\frac{1}{2}}-\widehat{F}_{i-\frac{1}{2}}}{\Delta x}=F(U)_x\Big|_{x_i}+\mathcal{O}(\Delta x^5)\tag{25}$$

if the discontinuity indicator satisfies:

$$\varphi_{i\pm\frac{1}{2}}=1+\mathcal{O}(\Delta x^3)$$

where the $F(U)$ refers to the exact flux.

Proof. As we know, if we define the flux $\widehat{\widehat{F}}$ as follows:

$$\widehat{\widehat{F}}_{i+\frac{1}{2}} = F(U)\Big|_{x_{i+\frac{1}{2}}} - \frac{1}{24}\Delta x^2 F(U)_{xx}\Big|_{x_{i+\frac{1}{2}}} + \frac{7}{5760}\Delta x^4 F(U)_{xxxx}\Big|_{x_{i+\frac{1}{2}}}$$

where $F(U)$ is the exact flux, then we have

$$\frac{\widehat{\widehat{F}}_{i+\frac{1}{2}} - \widehat{\widehat{F}}_{i-\frac{1}{2}}}{\Delta x} = F(U)_x\Big|_{x_i} + \mathcal{O}(\Delta x^5)$$

Therefore, in order to prove the equation (25), it is sufficient to prove the relation:

$$\widehat{F}_{i+\frac{1}{2}} = \widehat{\widehat{F}}_{i+\frac{1}{2}} + \mathcal{O}(\Delta x^5) \tag{26}$$

if the numerical flux is Lipschitz continuous with respect to its arguments.

According to (6), we have:

$$\widehat{F}_{i+\frac{1}{2}} = \widehat{F}(U^-_{i+\frac{1}{2}}, U^+_{i+\frac{1}{2}}) - \varphi_{i+\frac{1}{2}}\left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i+\frac{1}{2}}} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i+\frac{1}{2}}}\right)$$

The values $U^{\pm}_{i+\frac{1}{2}}$ are obtained from the WENO interpolation and we have

$$U^{\pm}_{i+\frac{1}{2}} = U(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^5).$$

Based on the consistency and Lipschitz continuity of the numerical flux, we have:

$$\widehat{F}(U^-_{i+\frac{1}{2}}, U^+_{i+\frac{1}{2}}) = F(U)\Big|_{x_{i+\frac{1}{2}}} + \mathcal{O}(\Delta x^5)$$

In addition, we use central difference methods in a component by component fashion to approximate the derivatives terms, see equations (2) and (3). Therefore, we have:

$$F_{xx}\Big|_{x_{i+\frac{1}{2}}} = F(U)_{xx}\Big|_{x_{i+\frac{1}{2}}} + \mathcal{O}(\Delta x^4)$$
$$F_{xxxx}\Big|_{x_{i+\frac{1}{2}}} = F(U)_{xxxx}\Big|_{x_{i+\frac{1}{2}}} + \mathcal{O}(\Delta x^2)$$

If we have

$$\varphi_{i+\frac{1}{2}} = 1 + \mathcal{O}(\Delta x^3)$$

then we can obtain

$$\begin{aligned}\widehat{F}_{i+\frac{1}{2}} &= \widehat{F}(U^-, U^+) - \varphi_{i+\frac{1}{2}}\left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i+\frac{1}{2}}} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i+\frac{1}{2}}}\right)\\ &= F(U)\Big|_{x_{i+\frac{1}{2}}} - \frac{1}{24}\Delta x^2 F(U)_{xx}\Big|_{x_{i+\frac{1}{2}}} + \frac{7}{5760}\Delta x^4 F(U)_{xxxx}\Big|_{x_{i+\frac{1}{2}}} + \mathcal{O}(\Delta x^5)\\ &\triangleq \widehat{\widehat{F}}_{i+\frac{1}{2}} + \mathcal{O}(\Delta x^5) \quad \square\end{aligned}$$

Proposition 2.2. *The convex combination of $\widehat{F}^H$ and $\widehat{F}^L$ (16) satisfies formula (26), if the parameter $\varphi = 1 + \mathcal{O}(\Delta x^5)$ and $\Delta t = \mathcal{O}(\Delta x^r), r \le 2$.*

The detailed proof is given in Appendix A.

Proposition 2.3. *In formulas (19) and (20), if $\varphi = 1 + \mathcal{O}(\Delta x^4)$, then we have*

$$U^{*,I}_{i_*} = U^*_{i_*} + \mathcal{O}(\Delta x^5), \quad U^{*,II}_{i_*} = U^*_{i_*} + \mathcal{O}(\Delta x^5)$$

where i_* is the subscript of the cell where the interface is located at time level t^*.

Proof. We take $i_* = i_n + 1$ as example. The proof for $i_* = i_n$ or $i_* = i_n - 1$ is similar.

Due to our CFL number, we have $\alpha^*_{i_n+1} \le 0.5$. So, we can obtain

$$U^*_{i_n+1} = U(x_{i_n+1}, t^n) - \frac{\Delta t}{\Delta x}\left(\widehat{F}_{i_n+3/2} - \widehat{F}_{i_n+1/2}\right) = U(x_{i_n+1}, t^n) + \mathcal{O}(\Delta t)$$

According to (43). We have

$$U^*_I = U(x_{i_n+1}, t^n) + \mathcal{O}(\Delta x) = U^*_{i_n+1} + \mathcal{O}(\Delta x)$$

Then, we obtain

$$\begin{aligned} &U^{*,I}_{i_n+1} - U^*_{i_n+1} = \varphi U^*_{i_n+1} + (1-\varphi)U^*_I - U^*_{i_n+1} \\ =&(1-\varphi)\mathcal{O}(\Delta x) = \mathcal{O}(\Delta x^5) \end{aligned}$$

Therefore, we should have

$$\varphi = 1 + \mathcal{O}(\Delta x^4) \quad \square$$

Based on the previous analysis, in order to maintain the accuracy, the discontinuity indicator should satisfy:

$$\varphi_{i+\frac{1}{2}} = 1 + \mathcal{O}(\Delta x^5) \quad \& \quad \Delta t = \mathcal{O}(\Delta x^r),\ r \le 2$$

Now, we describe the detailed steps to calculate the discontinuity indicator satisfying $\varphi_{i+\frac{1}{2}} = 1 + \mathcal{O}(\Delta x^5)$. Later, in the numerical tests section, we will further require the time stepping to satisfy $\Delta t = \mathcal{O}(\Delta x^r), r \le 2$. For simplicity, we use w as the indicator variable.

Step 1. We define a big stencil $S_0 = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+3}\}$, and four small stencils $S_1 = \{x_{i-2}, x_{i-1}, x_i\}$, $S_2 = \{x_{i-1}, x_i, x_{i+1}\}$, $S_3 = \{x_i, x_{i+1}, x_{i+2}\}$, $S_4 = \{x_{i+1}, x_{i+2}, x_{i+3}\}$. Then we need to construct polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$, $p_3(x)$, $p_4(x)$, such that:

$$\begin{aligned} &p_0(x_{i+l}) = w_{i+l} \quad l = -2, \cdots, 3 \\ &p_1(x_{i+l}) = w_{i+l} \quad l = -2, \cdots, 0 \\ &p_2(x_{i+l}) = w_{i+l} \quad l = -1, \cdots, 1 \\ &p_3(x_{i+l}) = w_{i+l} \quad l = 0, \cdots, 2 \\ &p_4(x_{i+l}) = w_{i+l} \quad l = 1, \cdots, 3 \end{aligned}$$

Step 2. We use the same recipe as in [13] to compute the smoothness indicators for each polynomial on cell $[x_i, x_{i+1}]$:

$$\beta_r = \sum_{k=1}^{5} \int_{x_i}^{x_{i+1}} \Delta x^{2k-1} \left(\frac{\partial^k}{\partial x^k}(p_r(x))\right)^2 \mathrm{dx}, \quad r = 0, 1, 2, 3, 4$$

We have the following expressions:

$$\begin{aligned} \beta_0 =& (-w_i + w_{i+1})^2 + \frac{1421461}{2275}\left(\frac{1}{48}w_{i-2} - \frac{1}{16}w_{i-1} + \frac{1}{24}w_i + \frac{1}{24}w_{i+1} - \frac{1}{16}w_{i+2} + \frac{1}{48}w_{i+3}\right)^2 \\ &+ \frac{13}{3}\left(-\frac{131}{3120}w_{i-2} + \frac{391}{1040}w_{i-1} - \frac{521}{1560}w_i - \frac{521}{1560}w_{i+1} + \frac{391}{1040}w_{i+2} - \frac{131}{3120}w_{i+3}\right)^2 \\ &+ \frac{781}{20}\left(\frac{617}{43736}w_{i-2} - \frac{31123}{131208}w_{i-1} + \frac{14019}{21868}w_i - \frac{14019}{21868}w_{i+1} + \frac{31123}{131208}w_{i+2} - \frac{617}{43736}w_{i+3}\right)^2 \\ &+ \frac{21520059541}{1377684}\left(-\frac{1}{120}w_{i-2} + \frac{1}{24}w_{i-1} - \frac{1}{12}w_i + \frac{1}{12}w_{i+1} - \frac{1}{24}w_{i+2} + \frac{1}{120}w_{i+3}\right)^2 \\ \beta_1 =& (w_{i-2} - 3w_{i-1} + 2w_i)^2 + \frac{13}{3}\left(\frac{1}{2}w_{i-2} - w_{i-1} + \frac{1}{2}w_i\right)^2 \\ \beta_2 =& (w_{i+1} - w_i)^2 + \frac{13}{3}\left(\frac{1}{2}w_{i+1} - w_i + \frac{1}{2}w_{i-1}\right)^2 \end{aligned}$$

$$\beta_3 = (w_{i+1} - w_i)^2 + \frac{13}{3}\left(\frac{1}{2}w_{i+2} - w_{i+1} + \frac{1}{2}w_i\right)^2$$

$$\beta_4 = (-2w_{i+1} + 3w_{i+2} - w_{i+3})^2 + \frac{13}{3}\left(\frac{1}{2}w_{i+1} - w_{i+2} + \frac{1}{2}w_{i+3}\right)^2$$

Step 3. Similar to [35], we define the parameter τ:

$$\tau = \frac{(\beta_0 - \beta_1)^2 + (\beta_0 - \beta_2)^2 + (\beta_0 - \beta_3)^2 + (\beta_0 - \beta_4)^2}{4}$$

and the parameter β:

$$\beta = \frac{\frac{\tau}{\beta_1+\varepsilon} + \frac{\tau}{\beta_2+\varepsilon} + \frac{\tau}{\beta_3+\varepsilon} + \frac{\tau}{\beta_4+\varepsilon}}{4}$$

Then, we can construct the discontinuity indicator φ:

$$\varphi_{i+\frac{1}{2}} = \left(\frac{1}{\beta+1}\right)^2$$

It has the following properties:

- $0 \le \varphi_{i+\frac{1}{2}} \le 1$;
- $\varphi_{i+\frac{1}{2}} = 1 + \mathcal{O}(\Delta x^6)$ in smooth areas;
- $\varphi_{i+\frac{1}{2}}$ is close to 0 near discontinuities.

The detailed proof of these properties is given in Appendix B.

2.2.5. Time discretization

In practice, we will use the third order TVD Runge-Kutta method to advance the nodal values:

$$\begin{cases} \mathrm{U}^{(1)} = \mathrm{U}^n + \Delta t \mathcal{L}(\mathrm{U}^n) \\ \mathrm{U}^{(2)} = \mathrm{U}^n + \frac{1}{4}\Delta t\left(\mathcal{L}\left(\mathrm{U}^n\right) + \mathcal{L}\left(\mathrm{U}^{(1)}\right)\right) \\ \mathrm{U}^{n+1} = \mathrm{U}^n + \frac{1}{6}\Delta t\left(\mathcal{L}\left(\mathrm{U}^n\right) + 4\mathcal{L}\left(\mathrm{U}^{(2)}\right) + \mathcal{L}\left(\mathrm{U}^{(1)}\right)\right) \end{cases} \tag{27}$$

In fact, we have only described the first step of the third order Runge-Kutta method (27). As to the second and third steps, we will use the same idea used in the first step. Therefore, for each step, we need to construct the fluxes $\widehat{F}^H$ and the fluxes $\widehat{F}^L$. Take the second step as an example. For the fluxes $\widehat{F}^H$, they can be constructed by combining fluxes from time levels t^n and $t^{(1)}$ together according to equation (27). For the fluxes $\widehat{F}^L$, we need to calculate the fluxes $\widehat{F}^L$ at those positions where they have been calculated in the time levels t^n and $t^{(1)}$. At this time, we can also obtain the corresponding interface stencil. For those $\widehat{F}^L$ located inside the interface stencil, we can use the conservation law to obtain the fluxes; for those $\widehat{F}^L$ located outside the interface stencil, we can follow the equation (6) to compute them. Then, we can construct the fluxes, advance the computational variables and finally define the new nodal values for fluid I and fluid II. Similar method can be applied to calculate the third step.

To summarize, we have the following general algorithm:

Algorithm 2

Input: $U_i^{n,I}$, $U_i^{n,II}$, ϕ_i^n, ϕ_i^*, $\widehat{F}^H_{i+1/2}$, Δt, Δx

Output: $U_i^{*,I}$, $U_i^{*,II}$

1: Apply $U_i^{n,I}$ and $U_i^{n,II}$ to define the computational variables U_i^n.
2: Based on the equations (6)-(14), compute the fluxes $\widehat{F}^L$.
3: Define the flux $\widehat{F}^n$.
4: Update the computational variables U_i^*.
5: Define the nodal values $U_i^{*,I}$ for fluid I and nodal values $U_i^{*,II}$ for fluid II based on the equations (19)-(20)-(17)-(18).

2.2.6. Flowchart 1D

Step 1. According to Algorithm 1, input $U_i^{n,I}$, $U_i^{n,II}$, ϕ_i^n, Δt, Δx, output $U_i^{(1),I}$, $U_i^{(1),II}$, $\widehat{F}^n_{i+1/2}$, $\phi_i^{(1)}$. Up to now, we have finished the first step of the third order TVD Runge-Kutta method (27).

Step 2. According to Algorithm 1, input $U_i^{(1),I}$, $U_i^{(1),II}$, $\phi_i^{(1)}$, Δt, Δx, output $\widetilde{U}_i^{(2),I}$, $\widetilde{U}_i^{(2),II}$, $\widehat{F}^{(1)}_{i+1/2}$, $\widetilde{\phi}_i^{(2)}$.

We construct $\widehat{F}^H$ for the second step of the third order TVD Runge-Kutta method (27):

$$\widehat{F}^H_{i+\frac{1}{2}} = \frac{1}{4}\left(\widehat{F}^n_{i+\frac{1}{2}} + \widehat{F}^{(1)}_{i+\frac{1}{2}}\right),$$

and the distance function $\phi^{(2)}$ at the time level $t^{(2)} = t^n + \Delta t/2$:

$$\phi_i^{(2)} = \frac{3}{4}\phi_i^n + \frac{1}{4}\widetilde{\phi}_i^{(2)}.$$

Then, according to Algorithm 2, input $U_i^{n,I}$, $U_i^{n,II}$, ϕ_i^n, $\phi_i^{(2)}$, $\widehat{F}^H_{i+1/2}$, $\Delta t/2$, Δx, output $U_i^{(2),I}$, $U_i^{(2),II}$. Up to now, we have finished the second step of the third order TVD Runge-Kutta method (27).

Step 3. According to Algorithm 1, input $U_i^{(2),I}$, $U_i^{(2),II}$, $\phi_i^{(2)}$, Δt, Δx, output $\widetilde{U}_i^{(3),I}$, $\widetilde{U}_i^{(3),II}$, $\widehat{F}^{(2)}_{i+1/2}$, $\widetilde{\phi}_i^{(3)}$.

We construct $\widehat{F}^H$ for the third step of the third order TVD Runge-Kutta method (27):

$$\widehat{F}^H_{i+\frac{1}{2}} = \frac{1}{6}\left(\widehat{F}^n_{i+\frac{1}{2}} + 4\widehat{F}^{(2)}_{i+\frac{1}{2}} + \widehat{F}^{(1)}_{i+\frac{1}{2}}\right),$$

and the distance function ϕ^{n+1} at the time level $t^{n+1} = t^n + \Delta t$:

$$\phi_i^{n+1} = \frac{1}{3}\phi_i^n + \frac{2}{3}\widetilde{\phi}_i^{(3)}.$$

Then, according to Algorithm 2, input $U_i^{n,I}$, $U_i^{n,II}$, ϕ_i^n, ϕ_i^{n+1}, $\widehat{F}^H_{i+1/2}$, Δt, Δx, output $U_i^{n+1,I}$, $U_i^{n+1,II}$. Up to now, we have finished the third step of the third order TVD Runge-Kutta method (27).

2.3. Conservation

In summary, our method can be divided into three steps:

1. We use the nodal values for fluid I and fluid II at time level t^n to construct the nodal values for the computational variables.

2. Then, we update the nodal values for the computational variables:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(\widehat{F}_{i+1/2} - \widehat{F}_{i-1/2}\right)$$

where

$$\widehat{F}_{i+1/2} = \widehat{F}^H_{i+1/2} \qquad \text{for} \quad i \neq i_n - 1 \quad \& \quad i_n \quad \& \quad i_n + 1$$

and

$$\widehat{F}_{i+1/2} = \widehat{F}^L_{i+1/2} + \varphi\left(\widehat{F}^H_{i+1/2} - \widehat{F}^L_{i+1/2}\right) \qquad \text{for} \quad i = i_n - 1, i_n, i_n + 1$$

Here, we take $\widehat{F}^H_{i+1/2} = \frac{1}{6}\left(\widehat{F}^n_{i+\frac{1}{2}} + 4\widehat{F}^{(2)}_{i+\frac{1}{2}} + \widehat{F}^{(1)}_{i+\frac{1}{2}}\right)$ and $\varphi = \min(\varphi_{i_n-1/2}, \varphi_{i_n+1/2}, \varphi_{i_n+3/2})$.

3. Finally, we redistribute and obtain the nodal values for fluid I and fluid II at time level t^{n+1}.

We can see that the construction and redistribution in the first step and the third step do not violate the conservation law. As to the second step, we use the conservative scheme to update the values of the computational variables. Therefore, our method is conservative.

3. Two-dimensional numerical schemes

The two-dimensional system for the compressible fluid can be written as follows:

$$U_t + F(U)_x + G(U)_y = 0, \tag{28}$$

where $U = (\rho, \rho u, \rho v, E)^T$, $F(U) = (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T$ and $G(U) = (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T$. Here (u, v) is the velocity vector, and the definition of ρ, E and p is the same as before. The equation of state for 2D is as follows:

$$\gamma\text{-law}: E = \frac{1}{2}\rho(u^2+v^2) + \frac{p}{\gamma-1}, \quad \text{Tait EOS}: E = \frac{1}{2}\rho(u^2+v^2) + \frac{p+\gamma\overline{p}}{\gamma-1}$$

3.1. Description of the finite difference scheme in 2D

Now, we describe the high order finite difference scheme in the two dimensional case. In our scheme, we again take the CFL number as 0.5.

For simplicity, the computational domain is equally divided: $a = x_0 < x_1 < \cdots < x_{N_x} = b$, and $c = y_0 < y_1 < \cdots < y_{N_y} = d$. We denote the cell $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ as the corresponding cell of the node (x_i, y_j).

We denote the nodal values for fluid I as $\{U^{n,I}_{i,j}\}$, and nodal values for fluid II as $\{U^{n,II}_{i,j}\}$. Now, we can define the computational variable $\{U^n_{i,j}\}$:

- When cell $I_{i,j}$ is fully covered by fluid I at time level t^n, we take $U^n_{i,j} = U^{n,I}_{i,j}$
- When cell $I_{i,j}$ is fully covered by fluid II at time level t^n, we take $U^n_{i,j} = U^{n,II}_{i,j}$
- When cell $I_{i,j}$ is a mixed cell at time level t^n: $U^n_{i,j} = \alpha_{i,j} U^{n,I}_{i,j} + (1-\alpha_{i,j}) U^{n,II}_{i,j}$

where $\alpha_{i,j}$ is the volume fraction of the fluid I in the cell $I_{i,j}$.

Then, we can update the computational variable $\{U^n_{i,j}\}$ to next time level t^*

$$U^*_{i,j} = U^n_{i,j} - \frac{\Delta t}{\Delta x}(\widehat{F}^n_{i+\frac{1}{2},j} - \widehat{F}^n_{i-\frac{1}{2},j}) - \frac{\Delta t}{\Delta y}(\widehat{G}^n_{i,j+\frac{1}{2}} - \widehat{G}^n_{i,j-\frac{1}{2}}) \tag{29}$$

We also need to build the flux $\widehat{F}^H$ and the flux $\widehat{F}^L$ in order to construct our numerical flux. For the flux $\widehat{F}^H$, similar to the one dimensional case, we have

$$\widehat{F}^H_{i+\frac{1}{2},j} = \widehat{F}(U^-_{i+\frac{1}{2},j}, U^+_{i+\frac{1}{2},j}) - \varphi_{i+\frac{1}{2},j}\left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i+\frac{1}{2}},y_j} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i+\frac{1}{2}},y_j}\right) \tag{30}$$

Similarly, we can obtain the fluxes $\widehat{G}^H_{i,j+\frac{1}{2}}$.

Next, we will describe the way to construct the low order flux $\widehat{F}^L$.

Similar to the one dimensional case, we evolve the distance function based on the velocity obtained by solving the Riemann problem at the interface, and determine the interface position at the new time level. We will also use the WENO interpolation at the interface. The value will be obtained by calculating the WENO interpolation in the x and y directions and taking the average of them. For simplicity, we regard the interface in the mixed cell as a straight line, and choose the center of the line as the interpolation position in the x and y directions.

When the cell I_{i_n,j_n} contains the interface, it is not suitable to be computed for a full time step. Therefore it is suggested that the cell I_{i_n,j_n} should be merged with its neighboring cells to generate an interface stencil. The detailed steps will be introduced in subsection 3.1.2. Now, we denote the r-th interface stencil at time level t^n as $A^{(r)}(t^n)$. The interface $\Gamma(t^n)$ divide the interface stencil into two parts: $A^{(r)}_I(t)$ for fluid I and $A^{(r)}_{II}(t)$ for fluid II. Now, we begin to deduce the formula for $A^{(r)}_I(t)$.

Similar to the ALE method, we integrate the equation (28) in space and time, then we have

$$\int_{t^n}^{t^*} \mathrm{d}t \iint_{A^{(r)}_I(t)} \Big(U_t + F(U)_x + G(U)_y\Big)\mathrm{d}x\mathrm{d}y = 0 \tag{31}$$

According to the Gauss theorem, we obtain

$$\int_{t^n}^{t^*} \mathrm{d}t \iint_{A^{(r)}_I(t)} U_t \mathrm{d}x\mathrm{d}y + \int_{t^n}^{t^*} \mathrm{d}t \oint_{\partial A^{(r)}_I(t)} \Big(F(U)\cdot n_x + G(U)\cdot n_y\Big)\mathrm{d}s = 0 \tag{32}$$

Then, we use the Reynold's transport theorem:

$$\frac{d}{dt}\iint_{A^{(r)}_I(t)} U\,\mathrm{d}x\mathrm{d}y = \iint_{A^{(r)}_I(t)} U_t\,\mathrm{d}x\mathrm{d}y + \oint_{\partial A^{(r)}_I(t)} U(\overrightarrow{V}\cdot\overrightarrow{n})\mathrm{d}s \tag{33}$$

where $\overrightarrow{V}$ is the normal velocity along the interface. Combining (32) and (33), we get the following equation:

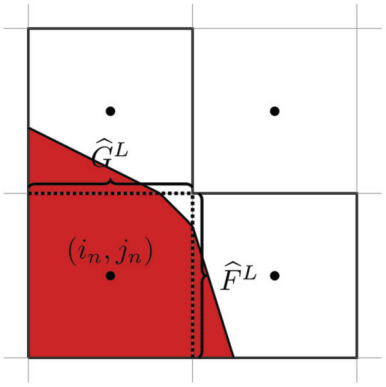

Fig. 3. Example for the interface stencil. Line: the interface stencil; Dotted line: cells within the interface stencil. Red: fluid I.

$$\int_{t^n}^{t^*} \frac{d}{dt} \iint_{A_I^{(r)}(t)} U \mathrm{dxdydt} + \int_{t^n}^{t^*} \mathrm{dt} \oint_{\partial A_I^{(r)}(t)} (F(U)\cdot n_x + G(U)\cdot n_y) - U(\overrightarrow{V}\cdot\overrightarrow{n})\mathrm{ds} = 0 \tag{34}$$

It can be represented by two parts: one consists of the four segments of the interface zone being cut by the interface; the other one consists of the segment of the interface inside the interface zone. Hence, we can rewrite the equation (34):

$$\begin{aligned} U^*_{I,A^{(r)}(t^n)} = \frac{1}{\sum\limits_{I_{i,j}\in A_I^{(r)}(t^n)} \alpha^*_{i,j}} \Big(& \sum_{I_{i,j}\in A_I^{(r)}(t^n)} \alpha^n_{i,j} U^n_I - \sum_{I_{i,j}\in A^{(r)}(t^n)} \frac{\ell^n_{i,j}\Delta t}{\Delta x \Delta y} \widehat{F}_{si,j} \\ & - \frac{\Delta t}{\Delta x} \sum_{j=j_1}^{j_2} [S_{i^2(j)+\frac{1}{2},j} \widehat{F}^L_{i^2(j)+\frac{1}{2},j} - S_{i^1(j)-\frac{1}{2},j} \widehat{F}^L_{i^1(j)-\frac{1}{2},j}] \\ & - \frac{\Delta t}{\Delta y} \sum_{i=i_1}^{i_2} [S_{i,j^2(i)+\frac{1}{2}} \widehat{G}^L_{i,j^2(i)+\frac{1}{2}} - S_{i,j^1(i)-\frac{1}{2}} \widehat{G}^L_{i,j^1(i)-\frac{1}{2}}] \Big) \end{aligned} \tag{35}$$

Here, we choose i_1, i_2, j_1, j_2 such that the set $\{I_{i,j} : i_1 \le i \le i_2, j_1 \le j \le j_2\}$ is the smallest stencil to cover $A^{(r)}(t^n)$. $i^2(j)$ and $i^1(j)$ represent the maximum and minimum indices in the x direction when j is given. Likewise, $j^2(i)$ and $j^1(i)$ represent the maximum and minimum indices in the y direction when i is given. See Fig. 3 for an example. Now, $\{I_{i,j} : i_n \le i \le i_n + 1, j_n \le j \le j_n + 1\}$ covers the interface stencil. When $j = j_n + 1$, there is only one cell I_{i_n, j_n+1} inside the interface stencil. Hence we know that the maximum index in the x direction is $i^2(j) = i_n$, and the minimum index is also $i^1(j) = i_n$. When $j = j_n$, both the cell I_{i_n,j_n} and the cell I_{i_n+1,j_n} are inside the interface stencil. Hence the maximum index in the x direction is $i^2(j) = i_n + 1$, and the minimum index is $i^1(j) = i_n$, and so on. Similar approach can be used to determine the maximum and minimum indices in the stencil in the y direction when i is given. $\alpha^n_{i,j}$, $\alpha^*_{i,j}$, $S_{i\pm\frac{1}{2},j}$ and $S_{i,j\pm\frac{1}{2}}$ are the fraction volumes of fluid I, see section 3.1.1 for detailed steps to determine them. $U^n_{I,A^{(r)}(t^n)}$ is the cell average in the interface stencil for fluid I at time level t^n:

$$U^n_{I,A^{(r)}(t^n)} = \frac{1}{|A_I^{(r)}(t^n)|} \iint_{A_I^{(r)}(t^n)} U \mathrm{dxdy}$$

Here, we assign $U^n_{I,A^{(r)}(t^n)}$ to the nodes in fluid II in the interface stencil $A^{(r)}(t^n)$, and then compute the numerical fluxes $\widehat{F}^L$ and $\widehat{G}^L$. $\ell^n_{i,j}$ is the length of the interface inside the cell $I_{i,j}$. $\widehat{F}_{si,j}$ is the interface flux:

$$\widehat{F}_s = (0, p\cdot n_x, p\cdot n_y, p(u\cdot n_x + v\cdot n_y))^T \tag{36}$$

where (n_x, n_y) is the unit normal at the interface, and p and (u, v) are the pressure and velocity which can be obtained by solving the Riemann problem at the interface.

Similarly, we can also obtain the cell average $U^*_{II,A^{(r)}(t^n)}$ for fluid II in the stencil $A^{(r)}(t^n)$. Then, we can define the temporary nodal value using $U^*_{I,A^{(r)}(t^n)}$ and $U^*_{II,A^{(r)}(t^n)}$:

$$U^{*,t}_{i,j} = \alpha^*_{i,j} U^*_{I,A^{(r)}(t^n)} + (1-\alpha^*_{i,j}) U^*_{II,A^{(r)}(t^n)} \quad (i,j) \in A^{(r)}(t^n)$$

where $\alpha^*_{i,j}$ is the fraction volume of fluid I in the cell $I_{i,j}$ at time level t^*. Then, similar to the one dimensional case, we can figure out the fluxes inside the interface stencil, for example, $\widehat{F}^L_{i_n+1/2,j_n}$ and $\widehat{G}^L_{i_n,j_n+1/2}$ in Fig. 3.

Now, we can determine the flux. Regarding the flux $\widehat{F}^n$, if $(x_{i+1/2}, y_j)$ is not in any interface stencil, we take

$$\widehat{F}^n_{i+1/2,j} = \widehat{F}^H_{i+1/2,j}$$

Otherwise, we have

$$\widehat{F}^n_{i+1/2,j} = \widehat{F}^L_{i+1/2,j} + \varphi\left(\widehat{F}^H_{i+1/2,j} - \widehat{F}^L_{i+1/2,j}\right)$$

where $\varphi = \min\left\{\varphi_{i\pm1/2,j}, \varphi_{i,j\pm1/2} \middle| I_{i,j} \in A^{(r)}(t^n), r = 1, 2, 3, \cdots\right\}$. Similar approach can be used to calculate $\widehat{G}^n$.

Next, we can evolve the computational variables based on equation (29). Now, we can determine the updated nodal values $\{U^{*,I}_{i,j}\}$ and $\{U^{*,II}_{i,j}\}$:

- When cell $I_{i,j}$ is fully covered by fluid I at time level t^*, we take

$$U^{*,I}_{i,j} = U^*_{i,j} \tag{37}$$

- When cell $I_{i,j}$ is fully covered by fluid II at time level t^*, we take

$$U^{*,II}_{i,j} = U^*_{i,j} \tag{38}$$

- When cell $I_{i,j}$ is a mixed cell at time level t^*:

  (1) If $\alpha^*_{i,j} > 0.5$, then it is a small cell for fluid II. We choose:

$$\begin{aligned} U^{*,II}_{i,j} &= \varphi U^*_{i,j} + (1-\varphi) U^*_{II,A^{(r)}(t^n)} \\ U^{*,I}_{i,j} &= \frac{U^*_{i,j} - (1-\alpha^*_{i,j}) U^{*,II}_{i,j}}{\alpha^*_{i,j}} \end{aligned} \tag{39}$$

  where $U^*_{i,j}$ is obtained by equation (29). $U^*_{II,A^{(r)}(t^n)}$ is the cell average value for fluid II in the interface stencil $A^{(r)}(t^n)$ at time level t^*.

  (2) If $\alpha^*_{i,j} \le 0.5$, then it is a small cell for fluid I. We choose:

$$\begin{aligned} U^{*,I}_{i,j} &= \varphi U^*_{i,j} + (1-\varphi) U^*_{I,A^{(r)}(t^n)} \\ U^{*,II}_{i,j} &= \frac{U^*_{i,j} - \alpha^*_{i,j} U^{*,I}_{i,j}}{1-\alpha^*_{i,j}} \end{aligned} \tag{40}$$

  where $U^*_{i,j}$ is obtained by equation (29), $U^*_{I,A^{(r)}(t^n)}$ is the cell average value for fluid I in the interface stencil $A^{(r)}(t^n)$ at time level t^*.

Similarly, we take $\varphi = \min\left\{\varphi_{i\pm1/2,j}, \varphi_{i,j\pm1/2} \middle| I_{i,j} \in A^{(r)}(t^n), r = 1, 2, 3, \cdots\right\}$.

Then, we can perform our two-dimensional scheme following the flowchart similar to the one dimensional one. To save space, we omit the detailed steps here. It should be noted that in two dimensional case we will perform the reinitialization routine every 100 steps to avoid the distance function ϕ from becoming too flat or too steep.

In the remainder of this section, we elaborate on describing the implementations in detail, including the way to generate the interface stencils, the way to calculate the volume fraction, the way to implement the level set method, and so on.

3.1.1. Volume fraction

The volume fraction is computed based on the distance function ϕ. It measures the ratio of $\phi < 0$ to the whole cell. We treat the interface as a straight line in the cell for simplicity. In general, we have many cases which need to be considered. By rotation, they can be divided into five generic cases:

1. $\phi_{i-\frac{1}{2},j+\frac{1}{2}} > 0$, $\phi_{i-\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j+\frac{1}{2}} > 0$

In this case, we take $\alpha_{i,j} = 0$.

2. $\phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0$, $\phi_{i-\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j+\frac{1}{2}} > 0$

See Fig. 4(a). According to the proportion, we have

$$a = \left|\frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}}\right|, \quad b = \left|\frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}}\right|$$

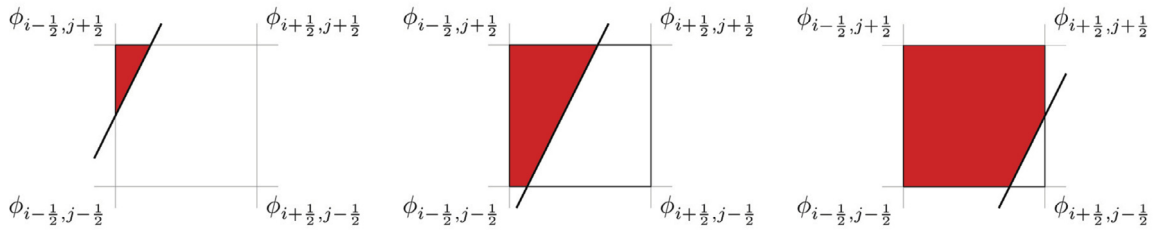

Fig. 4. Red: The part for $\phi < 0$; Line: interface.

Then, based on the triangle area formula, we have:

$$\alpha_{i,j} = \frac{1}{2}ab = \frac{1}{2}\left|\frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}}\right|\left|\frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}}\right|$$

3. $\phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0$, $\phi_{i-\frac{1}{2},j-\frac{1}{2}} < 0$, $\phi_{i+\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j+\frac{1}{2}} > 0$
See Fig. 4(b). Based on the trapezoidal area formula, we have

$$\alpha_{i,j} = \frac{1}{2}\left(\left|\frac{\phi_{i-\frac{1}{2},j-\frac{1}{2}}}{\phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}}}\right| + \left|\frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}}\right|\right)$$

4. $\phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0$, $\phi_{i-\frac{1}{2},j-\frac{1}{2}} < 0$, $\phi_{i+\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j+\frac{1}{2}} < 0$
See Fig. 4(c). Based on the triangle area formula, we have:

$$\alpha_{i,j} = 1 - \frac{1}{2}\left|\frac{\phi_{i+\frac{1}{2},j-\frac{1}{2}}}{\phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}}}\right|\left|\frac{\phi_{i+\frac{1}{2},j-\frac{1}{2}}}{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}}}\right|$$

5. $\phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0$, $\phi_{i-\frac{1}{2},j-\frac{1}{2}} < 0$, $\phi_{i+\frac{1}{2},j-\frac{1}{2}} < 0$, $\phi_{i+\frac{1}{2},j+\frac{1}{2}} < 0$
In this case, we take $\alpha_{i,j} = 1$.

3.1.2. Mixing procedure

In this subsection, we will introduce a way to generate interface stencil. It is based on the normal vector of the interface which can change dynamically with the interface evolution. Now, we will introduce the procedure in detail.

1. Obtain the normal vector

We need to obtain the normal vector at the corners of the cell. We can use the following equation to compute the normal vector at the node $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$:

$$\overrightarrow{n}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{\nabla\phi^*_{i+\frac{1}{2},j+\frac{1}{2}}}{|\nabla\phi^*_{i+\frac{1}{2},j+\frac{1}{2}}|}$$

where $\nabla\phi^*_{i+\frac{1}{2},j+\frac{1}{2}}$ can be obtained using the distance function at time level t^* by the WENO method.

Then, for any cell $I_{i,j}$ which is mixed at time level t^n or t^*, we can compute the sum of the x and y components of the normal vector at the corners of the cell:

$$Sn^x_{i,j} = \sum |\phi^*_{x\,i\pm\frac{1}{2},j\pm\frac{1}{2}}|, \quad Sn^y_{i,j} = \sum |\phi^*_{y\,i\pm\frac{1}{2},j\pm\frac{1}{2}}|$$

We also need to compute the fraction volume of the fluid I in the cell, denoted as $\alpha^*_{i,j}$.

2. Generate the interface stencils [15]

- If $\alpha^*_{i,j} > 0.5$
  - ♡ If $\left||Sn^x_{i,j}| - |Sn^y_{i,j}|\right| < 10^{-6}$,
    - ◇ If $\alpha^*_{i-1,j} < \alpha^*_{i+1,j}$, cell $I_{i,j}$ will merge with the cell $I_{i-1,j}$. Otherwise, cell $I_{i,j}$ will merge with the cell $I_{i+1,j}$.
  - ♡ Otherwise,
    - ◇ If $|Sn^x_{i,j}| > |Sn^y_{i,j}|$,
      - * If $\alpha^*_{i-1,j} < \alpha^*_{i+1,j}$, cell $I_{i,j}$ will merge with the cell $I_{i-1,j}$. Otherwise, cell $I_{i,j}$ will merge with the cell $I_{i+1,j}$. See cell C in Fig. 5.

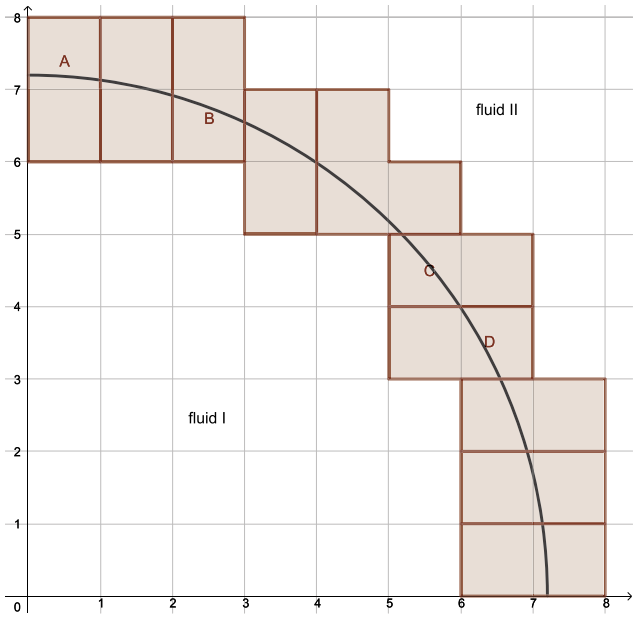

Fig. 5. Formation of the interface stencils. Pink: the interface stencil. Curve: the interface. Fluid I: $\phi < 0$. Fluid II: $\phi > 0$.

◇ Otherwise,

* If $\alpha^*_{i,j+1} < \alpha^*_{i,j-1}$, cell $I_{i,j}$ will merge with the cell $I_{i,j+1}$. Otherwise, cell $I_{i,j}$ will merge with the cell $I_{i,j-1}$. See cell B in Fig. 5.

- Otherwise,

♡ If $\left||Sn^x_{i,j}| - |Sn^y_{i,j}|\right| < 10^{-6}$,

◇ If $\alpha^*_{i-1,j} > \alpha^*_{i+1,j}$, cell $I_{i,j}$ will merge with the cell $I_{i-1,j}$. Otherwise, cell $I_{i,j}$ will merge with the cell $I_{i+1,j}$.

♡ Otherwise,

◇ If $|Sn^x_{i,j}| > |Sn^y_{i,j}|$,

* If $\alpha^*_{i-1,j} > \alpha^*_{i+1,j}$, cell $I_{i,j}$ will merge with the cell $I_{i-1,j}$. Otherwise, cell $I_{i,j}$ will merge with the cell $I_{i+1,j}$. See cell D in Fig. 5.

◇ Otherwise,

* If $\alpha^*_{i,j+1} > \alpha^*_{i,j-1}$, cell $I_{i,j}$ will merge with the cell $I_{i,j+1}$. Otherwise, cell $I_{i,j}$ will merge with the cell $I_{i,j-1}$. See cell A in Fig. 5.

Then, we can obtain the interface stencils.

3.1.3. Characteristic projection in 2D

Considering the following quasi-linear form of the Euler equation:

$$W_t + A(W)W_x + B(W)W_y = 0$$

where

$$W = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} \qquad A(W) = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & 1/\rho \\ 0 & 0 & u & 0 \\ 0 & \rho c^2 & 0 & u \end{pmatrix} \qquad B(W) = \begin{pmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & 1/\rho \\ 0 & 0 & \rho c^2 & v \end{pmatrix}$$

We then give the left and right eigenvector matrices of matrix $A(W)$ as:

$$R^x(W) = \begin{pmatrix} 1 & 1 & 0 & 1 \\ -\frac{c}{\rho} & 0 & 0 & \frac{c}{\rho} \\ 0 & 0 & 1 & 0 \\ c^2 & 0 & 0 & c^2 \end{pmatrix} \qquad L^x(W) = \begin{pmatrix} 0 & -\frac{\rho}{2c} & 0 & \frac{1}{2c^2} \\ 1 & 0 & 0 & -\frac{1}{c^2} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{\rho}{2c} & 0 & \frac{1}{2c^2} \end{pmatrix}$$

and we also give the left and right eigenvector matrices of matrix $B(W)$ as:

$$R^y(W)=\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ -\frac{c}{\rho} & 0 & 0 & \frac{c}{\rho} \\ c^2 & 0 & 0 & c^2 \end{pmatrix} \qquad L^y(W)=\begin{pmatrix} 0 & 0 & -\frac{\rho}{2c} & \frac{1}{2c^2} \\ 1 & 0 & 0 & -\frac{1}{c^2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{\rho}{2c} & \frac{1}{2c^2} \end{pmatrix}$$

Here, c refers to sound speed. It is advised that, when the fluxes are computed along a cell boundary, a one dimensional local characteristic decomposition normal to the boundary is performed. The detailed steps can be seen in section 2.2.2.

3.1.4. Level set method

In the two dimensional case, the distance function satisfies the following expression:

$$\phi_t + V_n|\nabla\phi| = 0 \tag{41}$$

where ϕ is the distance function defined at the half nodes. V_n is the normal velocity which can be obtained by solving the Riemann problem in the mixed cells, assigning the values to the corners of the cells, and extrapolating the values by extending functions [19]:

$$q_\tau + \text{sign}(\phi)\frac{\nabla\phi}{|\nabla\phi|}\nabla q = 0$$

Here, q is the extended variable. Then, we can obtain the semi-discrete scheme of equation (41):

$$\begin{aligned}\frac{\mathrm{d}\phi}{\mathrm{d}t} = &-(v^+_{i+\frac{1}{2},j+\frac{1}{2}}\sqrt{\max\left((a^+)^2,(b^-)^2\right)+\max\left((c^+)^2,(d^-)^2\right)} \\ &+v^-_{i+\frac{1}{2},j+\frac{1}{2}}\sqrt{\max\left((a^-)^2,(b^+)^2\right)+\max\left((c^-)^2,(d^+)^2\right)})\end{aligned} \tag{42}$$

where $v_{i+\frac{1}{2},j+\frac{1}{2}}$ is the normal velocity at the half node $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$, $x^+ = \max(x,0)$, $x^- = \min(x,0)$. Here a, b, c, d refer to $\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+$ respectively. The values of $\phi_x^\pm$ and $\phi_y^\pm$ can be obtained by the WENO method. The third order TVD Runge-Kutta method will be used to improve the temporal accuracy for the scheme (42).

3.1.5. Reinitialization

In order to ensure the ϕ does not become too flat or too steep along the interface, we need to use the reinitialization procedure [30]:

$$\phi_\tau + \text{sign}(\phi_0)(|\nabla\phi| - 1) = 0$$

where ϕ_0 is obtained from the level set method. Then, we have:

$$\begin{aligned}\frac{\mathrm{d}\phi}{\mathrm{d}\tau} = &-\Big(s^+_{i+\frac{1}{2},j+\frac{1}{2}}(\sqrt{\max\left((a^+)^2,(b^-)^2\right)+\max\left((c^+)^2,(d^-)^2\right)}-1) \\ &+s^-_{i+\frac{1}{2},j+\frac{1}{2}}(\sqrt{\max\left((a^-)^2,(b^+)^2\right)+\max\left((c^-)^2,(d^+)^2\right)}-1)\Big)\end{aligned}$$

Here, $s_{i+\frac{1}{2},j+\frac{1}{2}}$ is a sign function valued at $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$. $x^+ = \max(x,0)$, $x^- = \min(x,0)$. The definition of $a^\pm$, $b^\pm$, $c^\pm$, $d^\pm$ is the same as before. In practice, third order TVD Runge-Kutta method will be used to discretize the pseudo-time derivative. The stopping criterion for this iteration is $e_1 < \Delta\tau\Delta x\Delta y$ or $k \le 20$, where the e_1 is the L_1 difference between two consecutive iteration steps and k is the total iteration number. We take $\Delta\tau = 0.1\min(\Delta x, \Delta y)$ in the experiment. The re-initialization procedure is performed every 100 time steps.

3.2. Conservation

In summary, our method in the two dimensional case can also be divided into three steps:

1. We use the nodal values for fluid I and fluid II at time level t^n to construct the nodal values for the computational variables.

2. Then, we update the computational variables

$$U^{n+1}_{i,j} = U^n_{i,j} - \frac{\Delta t}{\Delta x}\left(\widehat{F}_{i+1/2,j} - \widehat{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\widehat{G}_{i,j+1/2} - \widehat{G}_{i,j-1/2}\right)$$

3. Finally, we redistribute and obtain the nodal values for fluid I and fluid II.

The first and third steps are performed in a conservative manner. As to the second step, we update the nodal values based on a conservative scheme. Therefore, our method is conservative.

Table 1
CPU time.

Example	1	2	3	4	5	6	7
Time(s)	64.30	2.71	1.40	2.37	2.27	0.55	0.43

Table 2
Accuracy test for density in 1D.

	L_∞ error	order	L_2 error	order	L_1 error	order
80	1.48E-04		3.32E-05		1.05E-05	
120	2.81E-05	4.09	5.21E-06	4.57	1.55E-06	4.72
160	6.97E-06	4.84	1.28E-06	4.87	3.67E-07	5.01
200	2.48E-06	4.63	4.22E-07	4.98	1.19E-07	5.07
240	9.81E-07	5.09	1.69E-07	5.03	4.68E-08	5.10
280	4.63E-07	4.87	7.72E-08	5.07	2.12E-08	5.15
320	2.33E-07	5.15	3.91E-08	5.09	1.07E-08	5.14

4. Numerical tests

In this section, we present the results of our numerical experiments. The CFL numbers are taken as 0.5 for both the one-dimensional and the two-dimensional cases, except that we will choose $\Delta t = \mathcal{O}(\Delta x^{5/3})$ to guarantee that the spatial error dominates for the accuracy tests. In order to plot the final result, we will restore the computational value using nodal values for fluid I and fluid II, and impose the following conditions to compute density velocity and pressure in the mixed cell:

$$\frac{1}{\gamma-1}=\frac{\alpha}{\gamma^I-1}+\frac{1-\alpha}{\gamma^{II}-1}\quad \frac{\gamma\overline{p}}{\gamma-1}=\frac{\alpha\gamma^I\overline{p}^I}{\gamma^I-1}+\frac{(1-\alpha)\gamma^{II}\overline{p}^{II}}{\gamma^{II}-1}$$

where γ and $\overline{p}$ are the physical parameters defined in the mixed cell, α is the volume fraction of fluid I. γ^I and $\overline{p}^I$ are the physical parameters defined in the fluid I, γ^{II} and $\overline{p}^{II}$ are the physical parameters defined in the fluid II. We emphasize that these definitions of mixed equation of state in the mixed cells are introduced only for the purpose of plotting the final results, they do not participate in the computation of time evolution of the numerical solution at all.

We also list the CPU cost for Example 1 to Example 7, see Table 1.

Example 1. Artificial accuracy test in 1D.

We consider the artificial accuracy test [10]. We take $\gamma = 3$. The initial conditions are:

$$\rho(x,0)=\frac{1+0.2\sin(x)}{2\sqrt{\gamma}},\quad u(x,0)=\sqrt{\gamma}\rho(x,0),\quad p(x,0)=\rho(x,0)^\gamma$$

The computational domain is $[0, 2\pi]$. Periodic boundary conditions are used in this test. By the special choice of the parameter γ, initial conditions and boundary conditions, we can verify that $2\sqrt{\gamma}\rho(x,t)$ is the exact solution of the following Burgers equation:

$$\mu_t+\frac{1}{2}(\mu^2)_x=0,\quad \mu(x,0)=1+0.2\sin(x)$$

The velocity and pressure satisfy the following relation:

$$u(x,t)=\sqrt{\gamma}\rho(x,t),\qquad p(x,t)=\rho(x,t)^\gamma.$$

It is easy to verify that the solution of the Burgers equation above is smooth up to time $T = 5$. We set the final time $T = 3$. At this time, the solution is still smooth. We also put an artificial interface for which the fluids I and II are the same to both sides of the interface, however the full interface treating algorithm is applied. The initial artificial interface is located at $x = \pi$. We list the error and numerical accuracy order in Table 2. We can see that our method can achieve the designed fifth order of accuracy. We also test the accuracy of the discontinuity indicator by using the density from the final time, see Table 3. We can see that the discontinuity indicator can achieve the designed precision.

Example 2. A pure interface problem in 1D.

We solve a Riemann problem consisting of a single contact discontinuity in gas dynamics:

Table 3
Accuracy test for discontinuity indicator in 1D.

	L_∞ error	order	L_2 error	order	L_1 error	order
80	2.38E-06		4.33E-07		1.30E-07	
120	3.07E-07	5.05	4.81E-08	5.42	1.24E-08	5.80
160	6.30E-08	5.51	9.36E-09	5.69	2.28E-09	5.89
200	1.76E-08	5.71	2.56E-09	5.81	6.07E-10	5.94
240	6.09E-09	5.82	8.76E-10	5.88	2.05E-10	5.96
280	2.46E-09	5.88	3.52E-10	5.91	8.15E-11	5.97
320	1.12E-09	5.92	1.59E-10	5.93	3.67E-11	5.98

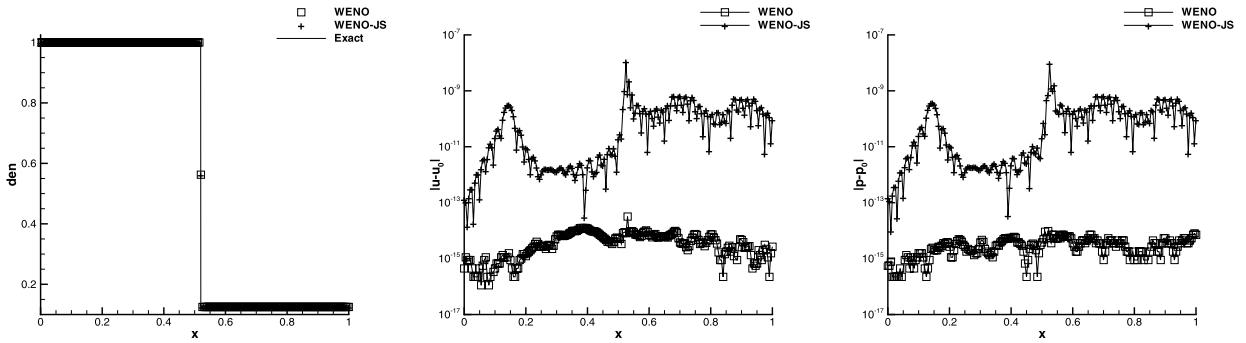

Fig. 6. Pure interface problem. From left to right: density, velocity, pressure. Line: exact solution; Square: numerical solution obtained by using the current WENO method (in velocity and pressure, the base has been subtracted); Plus: numerical solution obtained by using WENO-JS method (in velocity and pressure, the base has been subtracted).

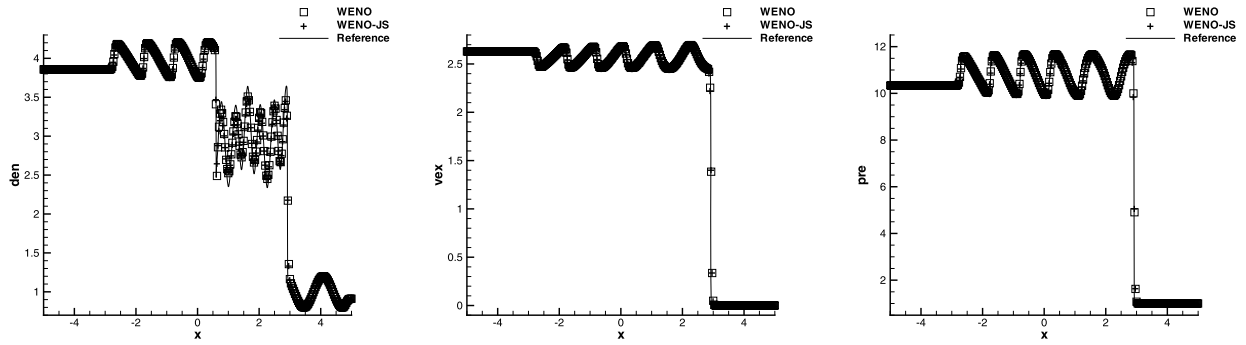

Fig. 7. Shock interacting with sine waves. From left to right: density, velocity, pressure. Square: numerical solution obtained by present WENO method, $N = 300$; Plus: numerical solution obtained by using WENO-JS method, $N = 300$; Line: Reference, $N = 2000$.

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (\quad 1, 1, 1, 1.4, 0), & x < 0.2 \\ (0.125, 1, 1, \quad 4, 1), & x \geq 0.2 \end{cases}$$

The computational domain is $[0, 1]$. We set the final time $T = 0.32$ and $N = 200$. Fig. 6 shows the result. The base velocity and pressure have been subtracted. From the figures, we can see that the interface propagates at the correct speed, and the oscillations solved by using the current WENO method in velocity and pressure are much smaller than the one solved by using the classical WENO method in [13], denoted as the WENO-JS method in the figure.

Example 3. Shock interacting with sine waves.

This example is tested in [32]. We solve the Euler equation with the following initial conditions

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (\quad 3.857143, 2.629369, 10.333333, \quad 1.4, 0), & x < -4.0 \\ (1 + 0.2\sin(5x), \quad 0, \quad 1, 1.666666, 0), & x \geq -4.0 \end{cases}$$

The computational domain is $[-5, 5]$. We compute the solution of this problem to $T = 1.8$ with $N = 300$, and show the final result in Fig. 7. The reference solution is obtained with $N = 2000$. We compare the computed result using the current WENO method with the one using the WENO-JS method, we can see that they are nearly the same. We also compare our result with the one in [6], we can see that the results are comparable although our scheme uses fewer degrees of freedom.

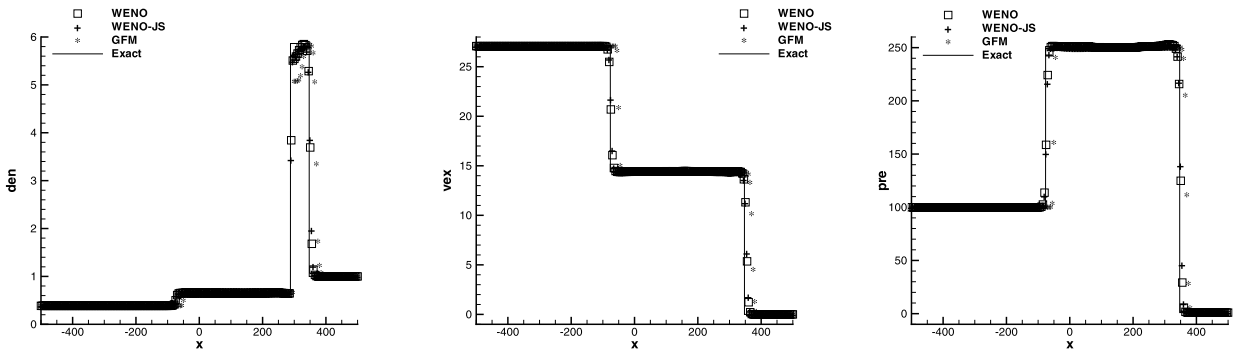

Fig. 8. Strong shock impacting on a gas-gas interface. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution obtained by present WENO method; Star: numerical solution obtained by finite difference WENO based GFM; Line: exact solution.

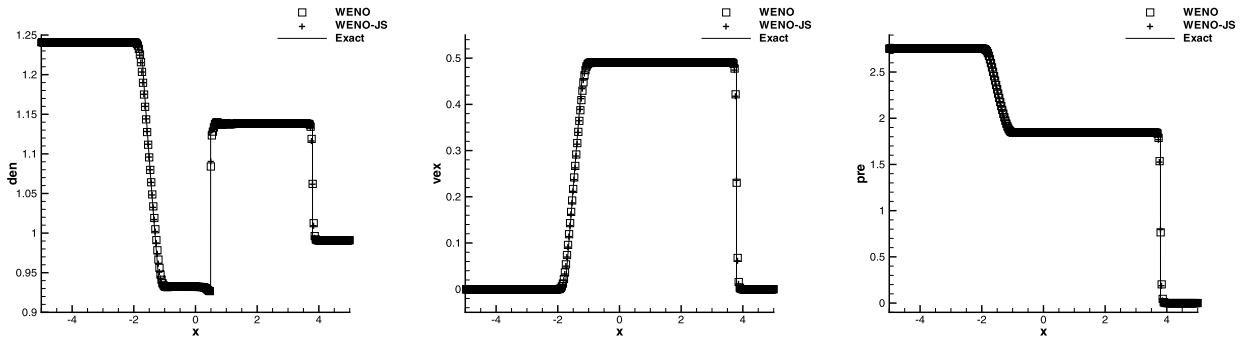

Fig. 9. Gas-water Riemann problem I. $N = 300$. From left to right: density, velocity, pressure. Square: numerical solution obtained by present WENO method; Plus: numerical solution obtained by WENO-JS method; Line: exact solution.

Example 4. Strong shock impacting on a gas-gas interface

We consider the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (0.3856, 27.0784, 100.0, 5/3, 0), & x < 0 \\ (\quad 1, \quad 0, \quad 1, 1.4, 0), & x \geq 0 \end{cases}$$

The example is taken from [16]. The domain is $[-500, 500]$ and the grid number $N = 200$. We list the results using the current WENO method, WENO-JS method, and the finite difference WENO based original GFM at time $t = 20$ in Fig. 8. It is clear that there are discrepancies in locations of the shock front and interface for the result obtained by GFM in comparison to the analytical solution. These incorrect features do not occur for our scheme.

Example 5. Gas-water Riemann problem I

We consider the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (1.241, 0, \quad 2.753, 1.4, \quad 0), & x < 0 \\ (0.991, 0, 3.059 \times 10^{-4}, 5.5, 1.505), & x \geq 0 \end{cases}$$

The example is taken from [6]. The domain is $[-5, 5]$ and the grid number $N = 300$. We list the computed results using both the current WENO method and the WENO-JS method at time $t = 1$ in Fig. 9. From the figures, we can see that both methods obtain correct interface location and high resolution. Comparing with the one in [6], we can see that our scheme produces a small trough along the interface in the density plot, but is otherwise a good approximation to the exact solution.

Example 6. Gas-water Riemann problem II

We consider the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (\ 0.01, -100.0, 100.0, \ 1.4, \quad 0), & x < 0 \\ (1.002, \quad 0, \quad 50, 7.15, 3309), & x \geq 0 \end{cases}$$

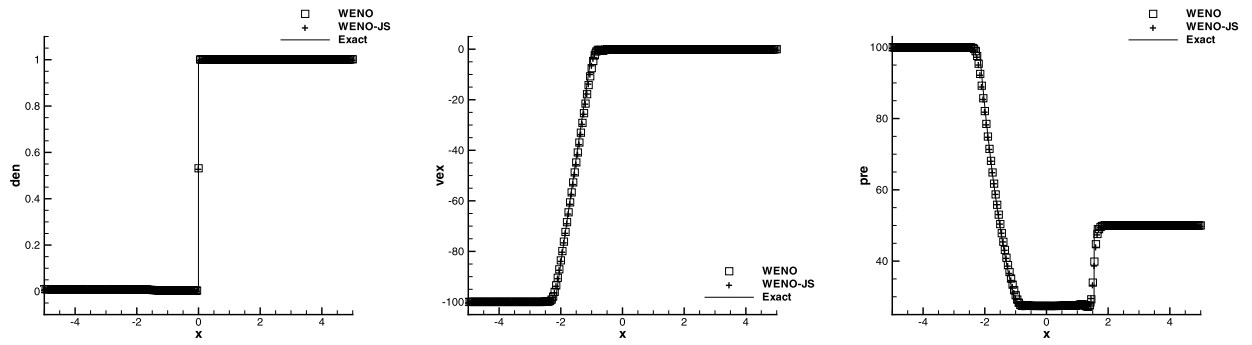

Fig. 10. Gas-water Riemann problem II. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution obtained by present WENO method; Plus: numerical solution obtained by WENO-JS method; Line: exact solution.

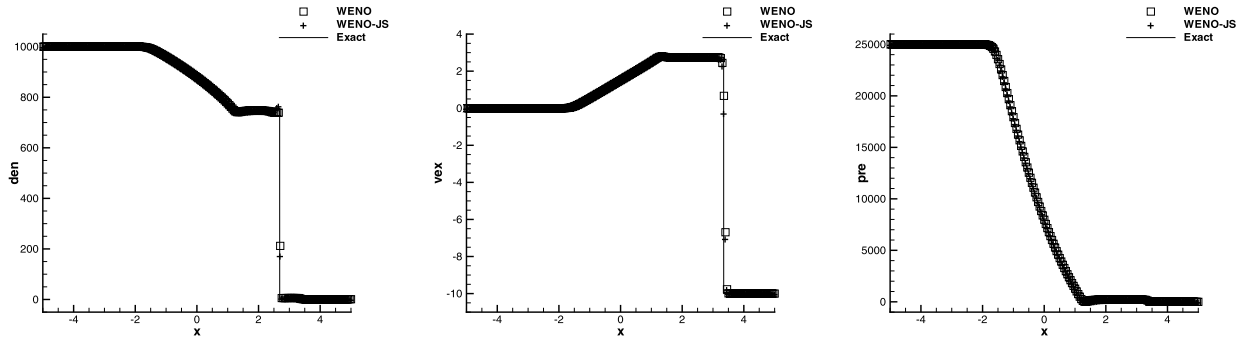

Fig. 11. Gas-water Riemann problem III. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution obtained by present WENO method; Plus: numerical solution obtained by WENO-JS method; Line: exact solution.

The example is taken from [17]. The domain is $[-5, 5]$ and the grid number $N = 200$. In this case, double rarefaction waves are generated in gas and water media respectively. We show the computed results using both the current WENO method and the WENO-JS method at time $t = 0.01$ in Fig. 10. From the figures, we can see that both methods obtain correct interface location and high resolution.

Example 7. Gas-water Riemann problem III.

We consider the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (1000, \quad 0, 25000, 7.15, 3309), & x < 2 \\ (\quad 1, -10, \quad 1, \ 1.4, \quad 0), & x \geq 2 \end{cases}$$

The domain is $[-5, 5]$ and the grid number $N = 200$. In this case, a very strong rarefaction wave is reflected back into the water. We plot the numerical result at time $t = 0.25$. From Fig. 11, we can see that the result obtained by using WENO-JS method produces a jump at the interface, while the correct interface location and high resolution are obtained by our method.

Example 8. Artificial accuracy test in 2D.

We consider the 2D artificial accuracy test. We take $\gamma = 3$. The initial conditions are:

$$\rho(x, y, 0) = \frac{1 + 0.2\sin(\frac{x+y}{2})}{\sqrt{2\gamma}} \quad u(x, y, 0) = v(x, y, 0) = \sqrt{\frac{\gamma}{2}}\rho(x, y, 0) \quad p(x, y, 0) = \rho(x, y, 0)^{\gamma}$$

The computational domain is $[0, 4\pi] \times [0, 4\pi]$. Periodic boundary conditions are used in this test. By the special choice of parameter γ, initial conditions and boundary conditions, we can verify that $\sqrt{2\gamma}\rho(x, y, t)$ is the exact solution of the following Burgers equation:

$$\mu_t + \frac{1}{2}(\mu^2)_x + \frac{1}{2}(\mu^2)_y = 0 \quad \mu(x, y, 0) = 1 + 0.2\sin(\frac{x+y}{2})$$

Table 4
Accuracy test for density in 2D.

	L_∞ error	order	L_2 error	order	L_1 error	order
80×80	1.52E-04		3.19E-05		9.57E-06	
120×120	3.13E-05	3.90	5.51E-06	4.33	1.57E-06	4.46
160×160	8.02E-06	4.73	1.43E-06	4.70	3.96E-07	4.79
200×200	2.89E-06	4.57	4.82E-07	4.87	1.32E-07	4.91
240×240	1.15E-06	5.04	1.96E-07	4.95	5.32E-08	5.00
280×280	5.49E-07	4.81	9.06E-08	4.99	2.44E-08	5.04
320×320	2.78E-07	5.10	4.63E-08	5.03	1.24E-08	5.06

and the velocity and pressure satisfy the relation: $u(x,y,t) = v(x,y,t) = \sqrt{\frac{\gamma}{2}}\rho(x,y,t)$, $p(x,y,t) = \rho(x,y,t)^\gamma$. It is easy to verify that the solution of the Burgers equation above is smooth up to time $T = 5$. We set the final time $T = 3$. At this time, the solution is still smooth. The initial artificial interface is located at $\sqrt{(x-2\pi)^2 + (y-2\pi)^2} = \pi$. We list the error and numerical accuracy order in Table 4. We can see that our method can achieve the designed fifth order accuracy.

Example 9. Pure interface problem in 2D.

We solve a pure interface problem in 2D where the interface is a straight line:

$$(\rho, u, v, p, \gamma, \overline{p}) = \begin{cases} (\quad 1{,}1{,}1{,}1{,}1.4{,}0), & x + 5y - 1.5 > 0 \\ (0.125{,}1{,}1{,}1, \quad 4{,}1), & x + 5y - 1.5 \le 0 \end{cases}$$

The computational domain is $[0,1] \times [0,1]$. We set the final time $T = 0.32$. Fig. 12 shows the result. The solutions along the cut line $y = x$ are also shown in the figures. The base velocity and pressure have been subtracted. From the figures, we can see that the interface propagates at the correct speed, and no oscillations other than those at the round-off error are observed in velocity and pressure.

Example 10. Shock impacting on a gas-gas interface in 2D

We next consider an air shock impacting on a helium bubble. The schematic for this problem is given in Fig. 13 where the upper and lower boundary conditions are non-reflective open boundaries. The left and right boundary conditions are the inflow and outflow, respectively. The initial conditions are:

$$(\rho, u, v, p, \gamma, \overline{p}) = \begin{cases} (\quad 1, \quad 0{,}0, \quad 1{,}1.4{,}0), & \text{Pre-shocked air} \\ (1.3764{,}0.394{,}0{,}1.5698{,}1.4{,}0), & \text{Post-shocked air} \\ (\ 0.138, \quad 0{,}0, \quad 1{,}5/3{,}0), & \text{Helium} \end{cases}$$

and the level set function $\phi = \sqrt{x^2 + y^2} - 1$, where $\phi < 0$ represents helium and $\phi > 0$ represents the air. The post-shock air state is given for $x < -1.2$.

In order to eliminate the "start-up" error mentioned in [22], we will use the numerical shock, namely we run our code for the pure shock condition until it settles down, then we add the bubble and start the computation. We plot density contours at time $t = 0.5$, $t = 1.0$, $t = 2.0$ and $t = 4.0$. From Fig. 14, we can see that the main features of the solution are correctly captured.

Example 11. Shock impacting on a water-gas interface in 2D

In the final problem, we consider an underwater shock interacting with a gas bubble in an open domain. We examine an underwater shock wave making impact on a gas bubble. The schematic for this problem is given in Fig. 13. The non-dimensionalized initial conditions are:

$$(\rho, u, v, p, \gamma, \overline{p}) = \begin{cases} (\quad 1000, \quad 0{,}0, \quad 1{,}7.15{,}3309), & \text{Pre-shocked water} \\ (1176.3576{,}1.1692{,}0{,}9120{,}7.15{,}3309), & \text{Post-shocked water} \\ (\quad 1, \quad 0{,}0, \quad 1, \ 1.4, \quad 0), & \text{Gas} \end{cases}$$

and the level set function $\phi = \sqrt{x^2 + y^2} - 1$, where $\phi < 0$ represents the gas and $\phi > 0$ represents the water. The post-shock water state is given for $x < -1.2$.

In this problem, very complex physics will occur at later time, and we stop our computation before the bubble collapse. We plot the contours of density for the numerical shock in Fig. 15 at $t = 0.06$, $t = 0.19$, $t = 0.357$ and $t = 0.471$ respectively. From the figures, we can see that high resolution is obtained by our method.

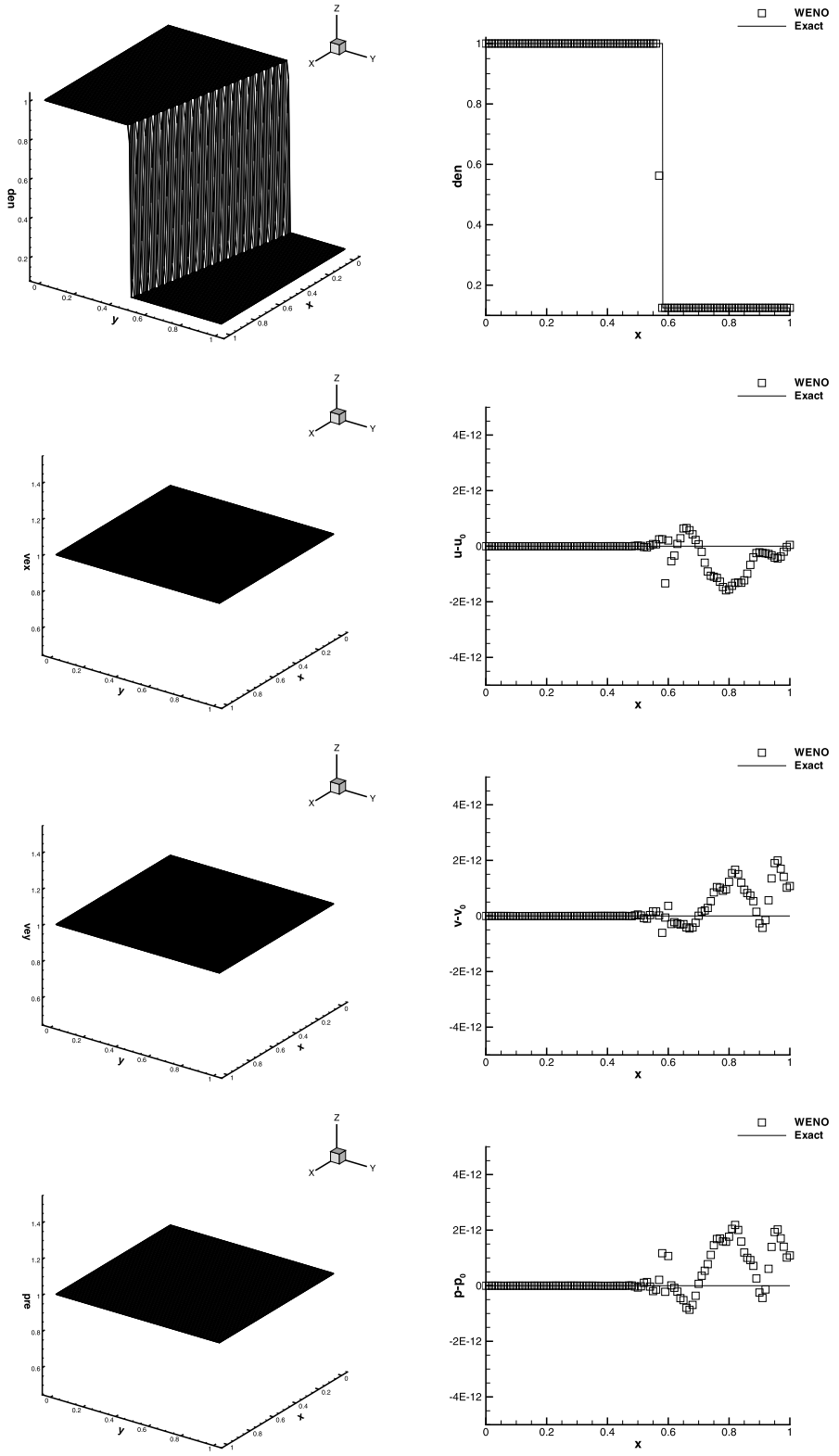

Fig. 12. Pure interface problem. 100×100 cells. From top to bottom: density, velocity in the x direction, velocity in the y direction, pressure. From left to right: solution in the whole domain, solution along the cut line $y = x$. Square: numerical solution (in velocity and pressure, the base has been subtracted); Line: exact solution (in velocity and pressure, the base has been subtracted).

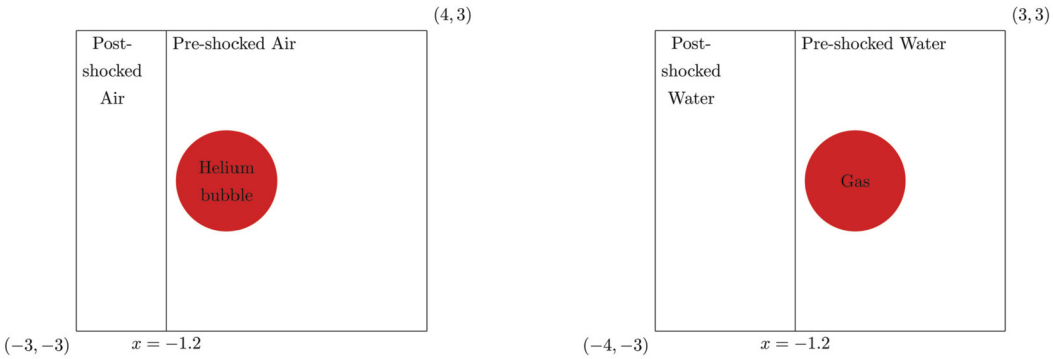

Fig. 13. Schematic for Example 10 and Example 11. Left: Example 10; Right: Example 11.

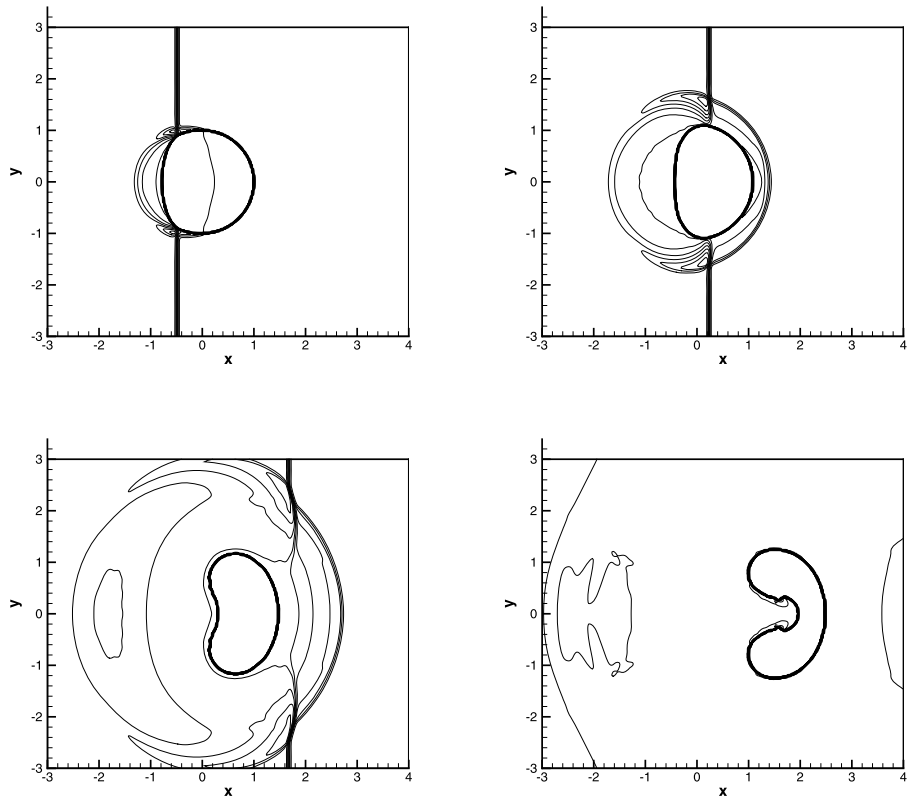

Fig. 14. Shock impacting on a gas-gas interface in 2D, with 280×240 cells, 30 equally spaced density contours from 0.1 to 1.6. Top left: $t = 0.5$; Top right: $t = 1.0$; Bottom left: $t = 2.0$; Bottom right: $t = 4.0$.

5. Concluding remarks

In this paper, we propose a conservative finite difference method to solve the two-medium flows. An alternative high order finite difference formulation is adopted to allow WENO interpolation on the physical variables of velocity and pressure, instead of WENO reconstruction or interpolation on the conserved variables. Numerical benchmarks show that the proposed scheme has high order accuracy and high resolution, ability to locate the correct interface position and non-oscillatory velocity and pressure transition across interfaces. Further research to improve efficiency and robustness of the conservative finite difference scheme for two-medium flows is ongoing. Extension to three-medium flows will also be considered in the future. The methodology extends naturally, however careful algebraic manipulations are needed in the mixed cells where all three media co-exist.

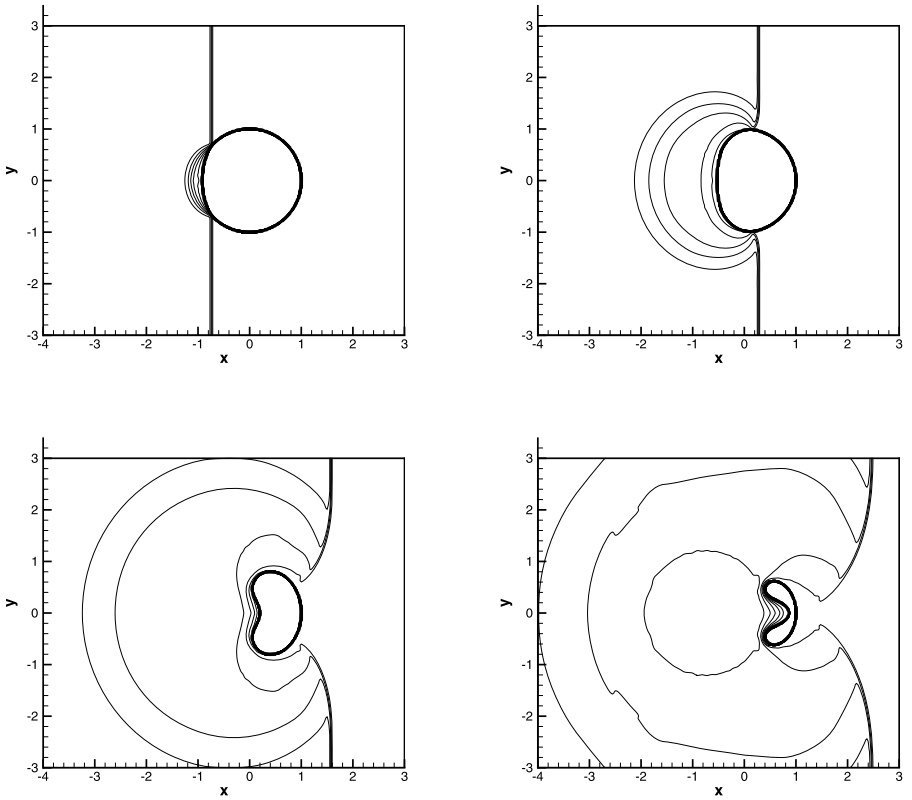

Fig. 15. Shock impacting on a water-gas interface in 2D, with 280×240 cells, 30 equally spaced density contours from 0 to 1200. Top left: $t = 0.06$; Top right: $t = 0.19$; Bottom left: $t = 0.357$; Bottom right: $t = 0.471$.

CRediT authorship contribution statement

Feng Zheng: Design numerical methods, Code, Writing – original draft preparation.
Chi-Wang Shu: Conceptualization, Design numerical methods, Writing – review & editing, Supervision.
Jianxian Qiu: Conceptualization, Design numerical methods, Writing – review & editing, Supervision.

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.

Acknowledgements

The first author would like to thank the China Scholarship Council, Division of Applied Mathematics of Brown University, members in the research group of Prof. Shu during his visit to Brown University, and Prof. J. Zhu and Ph.D. candidate Z. Zhao for the valuable help to him. He also thanks his alma mater Xiamen University and wishes her a happy 100th birthday.

Appendix A. The proof of Proposition 2.2

Proof. The same as before, we assume the interface $x(t^n)$ at time level t^n satisfies $x(t^n) \in [x_{i_n-1/2}, x_{i_n+1/2}]$. Due to the CFL condition, the interface $x(t^*)$ at time level t^* could only be in the cell I_{i_n-1}, I_{i_n} or I_{i_n+1}. For simplicity, we assume $x(t^*) \in [x_{i_n+1/2}, x_{i_n+3/2}]$, see Fig. 1. The proof for the other cases is similar.

According to Fig. 1, we need to calculate the convex combination of $\widehat{F}^H$ and $\widehat{F}^L$ at $x_{i_n-1/2}, x_{i_n+1/2}, x_{i_n+3/2}$. We analyze $\widehat{F}^n_{i_n-1/2}$ and $\widehat{F}^n_{i_n+3/2}$ first. Because the flux $\widehat{F}^L_{i+1/2}, i = i_n \pm 1$ is only first order accurate, we have

$$\widehat{F}^L_{i+1/2} = F(U)_{i+1/2} + \mathcal{O}(\Delta x), \quad i = i_n \pm 1$$

where the $F(U)_{i+1/2}$ refers to the exact flux at $x_{i+1/2}$. Therefore, if we require $\varphi = 1 + \mathcal{O}(\Delta x^4)$, then we have

$$\begin{aligned}\widehat{F}^n_{i+1/2} - F(U)_{i+1/2} &= \widehat{F}^L_{i+1/2} + \varphi(\widehat{F}^H_{i+1/2} - \widehat{F}^L_{i+1/2}) - F(U)_{i+1/2}\\ &= \mathcal{O}(\Delta x^5), \quad i = i_n \pm 1\end{aligned}$$

Next, we will analyze the flux $\widehat{F}^n_{i_n+1/2}$.

We assume $U(x_i, t^n)$ is the exact solution at x_i at time level t^n. We denote $\alpha^*_I = x(t^*) - x_{i_n-1/2}$, $\alpha^n_I = x(t^n) - x_{i_n-1/2}$, $U^n_I = U^{n,I}_{i_n}$, $U(x_{i_n}, t^n) = \alpha^n_I U^{n,I}_{i_n} + (1-\alpha^n_I)U^{n,II}_{i_n}$.

In the smooth region, $\{\phi^*_{i+1/2}\}$ and $\{\phi^n_{i+1/2}\}$ satisfy:

$$\phi^*_{i+1/2} = \phi^n_{i+1/2} - \Delta t(u^+_{i+\frac{1}{2}}\sqrt{\max(((\phi^-_x)^+)^2, ((\phi^+_x)^-)^2)} + u^-_{i+\frac{1}{2}}\sqrt{\max(((\phi^-_x)^-)^2, ((\phi^+_x)^+)^2)})$$

where $\phi^{\pm}_x$ can be obtained by the WENO method. Due to the property of the distance function, we have $|\phi^{\pm}_x| = 1 + \mathcal{O}(\Delta x^5)$. Therefore, we obtain the following equation:

$$\phi^*_{i_n+1/2} = \phi^n_{i_n+1/2} - \Delta t u + \mathcal{O}(\Delta x^5)$$

As to $x(t^n)$, $x(t^*)$ and ϕ^n, ϕ^*, we have

$$\begin{aligned}\frac{\alpha^*_I}{2\Delta x} =& \frac{x(t^*) - x_{i_n-1/2}}{2\Delta x} = \frac{x(t^*) - x_{i_n-1/2}}{x_{i_n+3/2} - x_{i_n-1/2}} = \frac{0 - \phi^*_{i_n-1/2}}{\phi^*_{i_n+3/2} - \phi^*_{i_n-1/2}}\\ =& \frac{0 - \phi^n_{i_n-1/2} + u\Delta t}{\phi^n_{i_n+3/2} - \phi^n_{i_n-1/2}} + \mathcal{O}(\Delta x^4) = \frac{x(t^n) - x_{i_n-1/2}}{2\Delta x} + \frac{u\Delta t}{2\Delta x} + \mathcal{O}(\Delta x^4)\\ =& \frac{\alpha^n_I + u\Delta t}{2\Delta x} + \mathcal{O}(\Delta x^4)\end{aligned}$$

So, we have

$$\alpha^*_I = \alpha^n_I + u\Delta t + \mathcal{O}(\Delta x^5)$$

As to the flux, we have

$$\begin{aligned}\widehat{F}_{i_n-\frac{1}{2}} &= \widehat{F}(U^-, U^+) - \varphi_{i_n-\frac{1}{2}}\left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i_n-\frac{1}{2}}} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i_n-\frac{1}{2}}}\right)\\ &= F(U)\Big|_{x_{i_n-\frac{1}{2}}} + \mathcal{O}(\Delta x^2) = \begin{pmatrix}\rho u\\ \rho u^2 + p\\ u(E+p)\end{pmatrix}\Big|_{x_{i_n-\frac{1}{2}}} + \mathcal{O}(\Delta x^2)\\ &= u_{i_n-\frac{1}{2}}\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix}\Big|_{x_{i_n-\frac{1}{2}}} + \begin{pmatrix}0\\ p\\ up\end{pmatrix}\Big|_{x_{i_n-\frac{1}{2}}} + \mathcal{O}(\Delta x^2)\\ &\triangleq u_{i_n-1/2}U^{n,I}_{i_n-1/2} + F_{si_n-1/2} + \mathcal{O}(\Delta x^2)\\ &= \quad uU^n_I + F_{si_n-1/2} + \mathcal{O}(\Delta x)\end{aligned}$$

and

$$U^n_I = U^{n,I}_{i_n} = \alpha^n_I U^{n,I}_{i_n} + (1-\alpha^n_I)U^{n,II}_{i_n} + \mathcal{O}(\Delta x) = U(x_{i_n}, t^n) + \mathcal{O}(\Delta x)$$

Therefore, according to Fig. 1, we have

$$\begin{aligned}U^{*,t}_{i_n} = U^*_I =& \frac{1}{\alpha^*_I}\left(\alpha^n_I U^n_I - \Delta t(\widehat{F}_s - \widehat{F}^L_{i_n-\frac{1}{2}})\right)\\ =& \frac{\left(\alpha^n_I U^n_I + \Delta t u U^n_I - \Delta t(\widehat{F}_s - F_{si_n-\frac{1}{2}})\right) + \mathcal{O}(\Delta x\Delta t)}{\alpha^*_I}\\ =& \frac{\left(\alpha^n_I U^n_I + \Delta t u U^n_I\right) + \mathcal{O}(\Delta x\Delta t)}{\alpha^*_I}\\ =& U^n_I + \mathcal{O}(\Delta t) = U(x_{i_n}, t^n) + \mathcal{O}(\Delta x)\end{aligned} \tag{43}$$

Similarly, we can also obtain

$$U^{*,t}_{i_n+1} = U(x_{i_n+1}, t^n) + \mathcal{O}(\Delta x)$$

If $\Delta t = \mathcal{O}(\Delta x^r), r \le 2$, according to formula (14) we obtain

$$\begin{aligned}
\widehat{F}^L_{i_n+1/2} &= \frac{1}{2}\left(\widehat{F}^L_{i_n-1/2} - \frac{U^{*,t}_{i_n} - U^n_{i_n}}{\frac{\Delta t}{\Delta x}}\right) + \frac{1}{2}\left(\frac{U^{*,t}_{i_n+1} - U^n_{i_n+1}}{\frac{\Delta t}{\Delta x}} + \widehat{F}^L_{i_n+3/2}\right) \\
&= \frac{1}{2}\left(\widehat{F}^L_{i_n-1/2} - \frac{U^{*,t}_{i_n} - U(x_{i_n}, t^n)}{\frac{\Delta t}{\Delta x}}\right) + \frac{1}{2}\left(\frac{U^{*,t}_{i_n+1} - U(x_{i_n+1}, t^n)}{\frac{\Delta t}{\Delta x}} + \widehat{F}^L_{i_n+3/2}\right) \\
&= \frac{1}{2}\left(\widehat{F}^L_{i_n-1/2} + \widehat{F}^L_{i_n+3/2}\right) + \mathcal{O}(\Delta x^{2-r}) \\
&= \left(F(U) - \frac{1}{24}\Delta x^2 F(U)_{xx} + \frac{7}{5760}\Delta x^4 F(U)_{xxxx}\right)\Big|_{x_{i_n+1/2}} + \mathcal{O}(\Delta x^{2-r})
\end{aligned}$$

That means the flux $\widehat{F}^L_{i_n+1/2}$ would be at least zero order accuracy to approximate the flux $\widehat{\widehat{F}}_{i_n+1/2}$. Due to $\varphi = 1 + \mathcal{O}(\Delta x^5)$, as to the formula (16), we have

$$\begin{aligned}
&\widehat{F}^n_{i_n+1/2} - \widehat{\widehat{F}}_{i_n+1/2} \\
=&(1-\varphi)\widehat{F}^L_{i_n+1/2} + \varphi\widehat{F}^H_{i_n+1/2} - \left(F(U) - \frac{1}{24}\Delta x^2 F(U)_{xx} + \frac{7}{5760}\Delta x^4 F(U)_{xxxx}\right)\Big|_{x_{i_n+1/2}} \\
=&(1-\varphi)\mathcal{O}(\Delta x^{2-r}) + \mathcal{O}(\Delta x^5) \\
=&\mathcal{O}(\Delta x^5)
\end{aligned}$$

To conclude, if $\varphi = 1 + \mathcal{O}(\Delta x^5)$ and $\Delta t = \mathcal{O}(\Delta x^r), r \le 2$, then the convex combination of $\widehat{F}^H$ and $\widehat{F}^L$ (16) satisfies formula (26). □

Appendix B. The proof for properties of discontinuity indicator

Firstly, due to $\beta \ge 0$, we have

$$0 \le \varphi_{i+\frac{1}{2}} \le 1$$

Secondly, we verify the accuracy order in the smooth region. Based on Taylor expansion at $x_{i+\frac{1}{2}}$, we have:

$$\begin{aligned}
\beta_0 &= w^{(1)}(x_{i+1/2})^2\Delta x^2 + \left(\frac{13}{12}w^{(2)}(x_{i+1/2})^2 + \frac{1}{12}w^{(1)}(x_{i+1/2})w^{(3)}(x_{i+1/2})\right)\Delta x^4 \\
&\quad + \left(\frac{7}{80}w^{(2)}(x_{i+1/2})w^{(4)}(x_{i+1/2}) + \frac{1}{960}w^{(1)}(x_{i+1/2})w^{(5)}(x_{i+1/2}) + \frac{1043}{960}w^{(3)}(x_{i+1/2})^2\right)\Delta x^6 + \mathcal{O}(\Delta x^8) \\
&= w^{(1)}(x_{i+1/2})^2\Delta x^2\left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_1 &= w^{(1)}(x_{i+1/2})^2\Delta x^2 + \left(\frac{13}{12}w^{(2)}(x_{i+1/2})^2 - \frac{23}{12}w^{(1)}(x_{i+1/2})w^{(3)}(x_{i+1/2})\right)\Delta x^4 \\
&\quad + \left(-\frac{13}{4}w^{(2)}(x_{i+1/2})w^{(3)}(x_{i+1/2}) + 2w^{(1)}(x_{i+1/2})w^{(4)}(x_{i+1/2})\right)\Delta x^5 + \mathcal{O}(\Delta x^6) \\
&= w^{(1)}(x_{i+1/2})^2\Delta x^2\left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_2 &= w^{(1)}(x_{i+1/2})^2\Delta x^2 + \left(\frac{13}{12}w^{(2)}(x_{i+1/2})^2 + \frac{1}{12}w^{(1)}(x_{i+1/2})w^{(3)}(x_{i+1/2})\right)\Delta x^4 \\
&\quad + \left(-\frac{13}{12}w^{(2)}(x_{i+1/2})w^{(3)}(x_{i+1/2})\right)\Delta x^5 + \mathcal{O}(\Delta x^6) \\
&= w^{(1)}(x_{i+1/2})^2\Delta x^2\left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_3 &= w^{(1)}(x_{i+1/2})^2\Delta x^2 + \left(\frac{13}{12}w^{(2)}(x_{i+1/2})^2 + \frac{1}{12}w^{(1)}(x_{i+1/2})w^{(3)}(x_{i+1/2})\right)\Delta x^4 \\
&\quad + \left(\frac{13}{12}w^{(2)}(x_{i+1/2})w^{(3)}(x_{i+1/2})\right)\Delta x^5 + \mathcal{O}(\Delta x^6)
\end{aligned}$$

$$= w^{(1)}(x_{i+1/2})^2 \Delta x^2 \left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2)$$

$$\beta_4 = w^{(1)}(x_{i+1/2})^2 \Delta x^2 + \left(\frac{13}{12} w^{(2)}(x_{i+1/2})^2 - \frac{23}{12} w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^4$$
$$+ \left(\frac{13}{4} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) - 2 w^{(1)}(x_{i+1/2}) w^{(4)}(x_{i+1/2})\right) \Delta x^5 + \mathcal{O}(\Delta x^6)$$
$$= w^{(1)}(x_{i+1/2})^2 \Delta x^2 \left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2)$$

Therefore, we obtain the following relation:

$$\beta_1 - \beta_0 = -2 w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) \Delta x^4 + \mathcal{O}(\Delta x^5)$$
$$\beta_2 - \beta_0 = -\frac{13}{12} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) \Delta x^5 + \mathcal{O}(\Delta x^6)$$
$$\beta_3 - \beta_0 = \frac{13}{12} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) \Delta x^5 + \mathcal{O}(\Delta x^6)$$
$$\beta_4 - \beta_0 = -2 w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) \Delta x^4 + \mathcal{O}(\Delta x^5)$$

It is easy to verify

$$\tau = \frac{(\beta_0 - \beta_1)^2 + (\beta_0 - \beta_2)^2 + (\beta_0 - \beta_3)^2 + (\beta_0 - \beta_4)^2}{4} = \quad \mathcal{O}(\Delta x^8)$$

and

$$\beta = \frac{1}{4}\left(\frac{\tau}{\beta_1 + \varepsilon} + \frac{\tau}{\beta_2 + \varepsilon} + \frac{\tau}{\beta_3 + \varepsilon} + \frac{\tau}{\beta_4 + \varepsilon}\right) = \mathcal{O}(\Delta x^6)$$

Therefore, in the smooth region, the discontinuity indicator $\varphi_{i+\frac{1}{2}}$ satisfies:

$$\varphi_{i+\frac{1}{2}} = \left(\frac{1}{\beta + 1}\right)^2 = 1 + \mathcal{O}(\Delta x^6)$$

Thirdly, when the big stencil contains a discontinuity, we have

$$\beta_0 = \mathcal{O}(1)$$

As to the small stencils S_1, S_2, S_3, S_4, one of them can avoid the discontinuity. For example, we assume S_1 can avoid discontinuity, then we have:

$$\beta_1 = \mathcal{O}(\Delta x^2)$$

As to τ and β, we obtain

$$\tau = \frac{(\beta_0 - \beta_1)^2 + (\beta_0 - \beta_2)^2 + (\beta_0 - \beta_3)^2 + (\beta_0 - \beta_4)^2}{4} = \mathcal{O}(1)$$

and

$$\beta = \frac{1}{4}\left(\frac{\tau}{\beta_1 + \varepsilon} + \frac{\tau}{\beta_2 + \varepsilon} + \frac{\tau}{\beta_3 + \varepsilon} + \frac{\tau}{\beta_4 + \varepsilon}\right) = \mathcal{O}(\Delta x^{-2})$$

Therefore, the discontinuity indicator satisfies:

$$\varphi_{i+\frac{1}{2}} = \left(\frac{1}{\beta + 1}\right)^2 = \mathcal{O}(\Delta x^4) \longrightarrow 0$$

References

[1] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach, J. Comput. Phys. 125 (1996) 150–160.
[2] R. Abgrall, S. Karni, Computations of compressible multifluids, J. Comput. Phys. 169 (2001) 594–623.
[3] R. Abgrall, R. Saurel, Discrete equations for physical and numerical compressible multiphase mixtures, J. Comput. Phys. 186 (2003) 361–396.
[4] G. Allaire, S. Clerc, S. Kokh, A five-equation model for the simulation of interfaces between compressible fluids, J. Comput. Phys. 181 (2002) 577–616.
[5] S. Bertoluzza, S. Falletta, G. Russo, C.-W. Shu, Numerical Solutions of Partial Differential Equations, Birkhäuser Verlag, 2009.
[6] J. Cheng, F. Zhang, T. Liu, A discontinuous Galerkin method for the simulation of compressible gas-gas and gas-water two-medium flows, J. Comput. Phys. 403 (2020) 109059.

[7] A. Chertock, S. Karni, A. Kurganov, Interface tracking method for compressible multifluids, ESAIM: Math. Model. Numer. Anal. 42 (2008) 991–1019.
[8] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[9] R.P. Fedkiw, A. Marquina, B. Merriman, An isobaric fix for the overheating problem in multimaterial compressible flows, J. Comput. Phys. 148 (1999) 545–578.
[10] G. Fu, C.-W. Shu, A new troubled-cell indicator for discontinuous Galerkin methods for hyperbolic conservation laws, J. Comput. Phys. 347 (2017) 305–327.
[11] X.Y. Hu, B.C. Khoo, N.A. Adams, F.L. Huang, A conservative interface method for compressible flows, J. Comput. Phys. 219 (2006) 553–578.
[12] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, SIAM J. Sci. Comput. 21 (2000) 2126–2143.
[13] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[14] Y. Jiang, C.-W. Shu, M. Zhang, An alternative formulation of finite difference WENO schemes with Lax-Wendroff time discretization for conservation laws, SIAM J. Sci. Comput. 35 (2013) A1137–A1160.
[15] J.-Y. Lin, Y. Shen, H. Ding, N.-S. Liu, X.-Y. Lu, Simulation of compressible two-phase flows with topology change of fluid-fluid interface by a robust cut-cell method, J. Comput. Phys. 328 (2017) 140–159.
[16] T.G. Liu, B.C. Khoo, C.W. Wang, Ghost fluid method for strong shock impacting on material interface, J. Comput. Phys. 190 (2003) 651–681.
[17] T.G. Liu, B.C. Khoo, K.S. Yeo, The ghost fluid method for compressible gas-water simulation, J. Comput. Phys. 204 (2005) 193–221.
[18] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.
[19] D.P. Peng, B. Merriman, S. Osher, H.K. Zhao, M. Kang, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 155 (1999) 410–438.
[20] J.X. Qiu, T.G. Liu, B.C. Khoo, Runge-Kutta discontinuous Galerkin methods for compressible two-medium flow simulations: one-dimensional case, J. Comput. Phys. 222 (2007) 353–373.
[21] J. Qiu, C.-W. Shu, On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, J. Comput. Phys. 183 (2002) 187–209.
[22] J.J. Quirk, S. Karni, On the dynamics of a shock-bubble interaction, J. Fluid Mech. 318 (1996) 129–163.
[23] R. Saurel, R. Abgrall, A simple method for compressible multifluid flows, SIAM J. Sci. Comput. 21 (1999) 1115–1145.
[24] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, A. Quarteroni (Eds.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, in: Lecture Notes in Mathematics, vol. 1697, Springer, Berlin, 1998, pp. 325–432.
[25] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[26] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys. 83 (1989) 32–78.
[27] K.-M. Shyue, An efficient shock-capturing algorithm for compressible multicomponent problems, J. Comput. Phys. 142 (1998) 208–242.
[28] K.-M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with van der Waals equation of state, J. Comput. Phys. 156 (1999) 43–88.
[29] K.-M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with Mie-Grüneisen equation of state, J. Comput. Phys. 171 (2001) 678–707.
[30] M. Sussman, P. Smereka, S. Osher, A PDE-based fast local level set method, J. Comput. Phys. 134 (1994) 146–159.
[31] C. Wang, T.G. Liu, B.C. Khoo, A real ghost fluid method for the simulation of multimedium compressible flow, SIAM J. Sci. Comput. 28 (2006) 278–302.
[32] C. Wang, C.-W. Shu, An interface treating technique for compressible multi-medium flow with Runge-Kutta discontinuous Galerkin method, J. Comput. Phys. 229 (2010) 8823–8843.
[33] T. Xiong, C.-W. Shu, M. Zhang, WENO scheme with subcell resolution for computing nonconservative Euler equations with applications to one-dimensional compressible two-medium flows, J. Sci. Comput. 53 (2012) 222–247.
[34] Z. Xu, C.-W. Shu, Anti-diffusive flux corrections for high order finite difference WENO schemes, J. Comput. Phys. 205 (2005) 458–485.
[35] J. Zhu, J.X. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, J. Comput. Phys. 318 (2016) 110–121.