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

Feng Zheng[1], Chi-Wang Shu[2] and Jianxian Qiu[3]

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 capture 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.

Keywords: finite difference; conservative method; two-medium flows

[1]College of Mathematics and Informatics, Fujian Normal University, Fuzhou, Fujian 350117, P.R. China. E-mail: fzbz200808-31@163.com

[2]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: chi-wang_shu@brown.edu. Research is partially supported by AFOSR grant FA9550-20-1-0055 and NSF grant DMS-2010107.

[3]School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling & High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, P.R. China. E-mail: jxqiu@xmu.edu.cn. Research is partially supported by NASF grant 12071392 and Science Challenge Project (China), No. TZ 2016002.

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 [21]. 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 [24, 25, 26]. 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, 20]. Although the shock-capturing method has achieved great success, it still has some drawbacks: First, the method diffuses the interface, 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 [15]. Among all methods of this type, the ghost fluid method (GFM) with the isobaric fix is undoubtedly the most successful [7, 8]. 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 the original GFM [14]. 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 [28]. There are also other similar methods, such as the interface treatment method [6], the simple single fluid algorithm [2], the path-conservative schemes [30] and so on. The drawback of these methods is that the scheme is not conservative, so theoretically we are not assured for 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 [29]. In [17], 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 [10]. In this paper we use a finite difference framework. Traditional conservative finite difference schemes perform reconstructions on the fluxes [22, 23, 11]. 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 [22, 23]. If we use the alternative formulation of high order finite difference schemes in [12], which is based on the high order flux expansion in [22], we will be performing interpolation, rather than reconstruction, on the conserved variables, not on the fluxes. In this paper, we modify the approach in [12] 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 [17, 10, 13]. Both the distance function and the computational variables are updated at each time step by the third order TVD Runge-Kutta time discretization [22]. Conservative property and sharp and non-oscillatory interface can 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 solution.

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 the 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 [22] 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 $U_i^{n,II}$. Given the interface $x(t^n) \in [x_{i_{l_n}-\frac{1}{2}}, x_{i_{l_n}+\frac{1}{2}}]$, 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:

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

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^*

$$U_i^* = U_i^n - \frac{\Delta t}{\Delta x}(\widehat{F}_{i+\frac{1}{2}}^n - \widehat{F}_{i-\frac{1}{2}}^n) \tag{4}$$

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 expression is similar to (1). We denote those 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:

$$\widehat{F}_{i+\frac{1}{2}}^H = \widehat{F}(U_{i+\frac{1}{2}}^-, U_{i+\frac{1}{2}}^+) - \varphi_{i+\frac{1}{2}} \left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i+\frac{1}{2}}} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i+\frac{1}{2}}} \right) \tag{5}$$

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 $U_{i+1/2}^{\pm}$ 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. If we try to interpolate the conserved variables or the flux, all of the components will jump across the interface, so it is difficult to maintain

the equilibrium of velocity and pressure, which are nonlinear functions of the conserved variables or fluxes. One possible way to get better results is to perform approximation directly on the primitive variables. However, this is impossible for finite volume methods of order of accuracy higher than two, since we only have the information of cell averages of the conserved variables. Neither is it possible for the traditional finite difference methods, since we reconstruct the fluxes directly. Therefore, based on the framework in [12], 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 derivatives terms, we can apply simple central difference method in a component by component fashion. It should be noted that we base on the position of the interface to decide the nodal values used in the interpolation. 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, we evolve the distance function which is based on the velocity obtained by solving the Riemann problem at the interface, and determine the interface position at the new time level. We can use the WENO method component by component 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 WENO method are described in subsection 2.2.3.

We assume the interface $x(t^n)$ at time level t^n satisfies $x(t^n) \in [x_{i_{l_n}-1/2}, x_{i_{l_n}+1/2}]$, and the interface $x(t^*)$ at the next time level t^* satisfies $x(t^*) \in [x_{i_{l_*}-1/2}, x_{i_{l_*}+1/2}]$. We denote $\alpha_{i_{l_*}}$ as the volume fraction of fluid I in the cell $I_{i_{l_*}}$. The mixed cell $I_{i_{l_n}}$ 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 $\alpha_{i_{l_*}} < 0.5$, then we choose the cells $I_{i_{l_*}-1}$ and $I_{i_{l_*}}$ as the interface stencil, and merge them to form two new interface cells: the cell $[x_{i_{l_*}-\frac{3}{2}}, x(t^n)]$ occupied by fluid I, and the cell $[x(t^n), x_{i_{l_*}+\frac{1}{2}}]$ occupied by fluid II;

- If $\alpha_{i_{l_*}} \geq 0.5$, then we choose the cells $I_{i_{l_*}}$ and $I_{i_{l_*}+1}$ as the interface stencil, and merge them to form two new interface cells: the cell $[x_{i_{l_*}-\frac{1}{2}}, x(t^n)]$ occupied by fluid I, and the cell $[x(t^n), x_{i_{l_*}+\frac{3}{2}}]$ 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 we have merged the cells $I_{i_{l_n}}$ and $I_{i_{l_n}+1}$ together, see Figure 1. Then, we need to calculate the fluxes $\widehat{F}^L$ at the positions $x_{l_n\pm 1/2}$ and $x_{l_n+3/2}$.

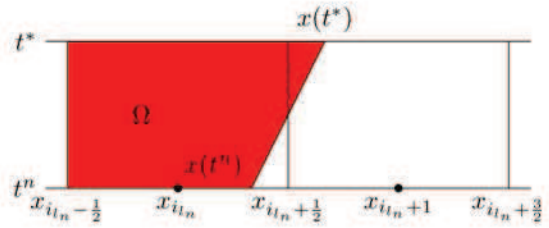


Figure 1: Sketch for the interface stencil. Red: fluid I

Regarding the flux at $x_{l_n-1/2}$, it is in the fluid I. Therefore, if $U^{n,I}_{i_{l_n}+1}$ is not defined, we will set $U^{n,I}_{i_{l_n}+1} = U^{n,I}_{i_{l_n}}$. Then we will use the nodal values $\{U^{n,I}_{i_{l_n}-3}, U^{n,I}_{i_{l_n}-2}, U^{n,I}_{i_{l_n}-1}, U^{n,I}_{i_{l_n}}, U^{n,I}_{i_{l_n}+1}, U^{n,II}_{i_{l_n}+2}\}$ to compute the numerical flux $\widehat{F}^L_{l_n-\frac{1}{2}}$ based on equation (5). Regarding the flux at $x_{l_n+3/2}$, it is in the fluid II. Similarly, if $U^{n,II}_{i_{l_n}}$ is not defined, we will set $U^{n,II}_{i_{l_n}} = U^{n,II}_{i_{l_n}+1}$. Then we will use the nodal values $\{U^{n,I}_{i_{l_n}-1}, U^{n,II}_{i_{l_n}}, U^{n,II}_{i_{l_n}+1}, U^{n,II}_{i_{l_n}+2}, U^{n,II}_{i_{l_n}+3}, U^{n,II}_{i_{l_n}+4}\}$ to compute the numerical flux $\widehat{F}^L_{l_n+\frac{3}{2}}$ based on equation (5). As to the flux $\widehat{F}^L_{l_n+\frac{1}{2}}$, it can be obtained by the conservation law, as described below in details.

For fluid I, we have the following integrated conservation law:

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

According to the Green's formula, we have

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

where Ω is the control volume for fluid I. Then, we have the average value for fluid I

$$U_I^* = \frac{1}{x(t^*) - x_{i_{l_n}-\frac{1}{2}}}\Big(\int_{x_{i_{l_n}-\frac{1}{2}}}^{x(t^n)} U^n\mathrm{dx} - \Delta t(\widehat{F_s} - \widehat{F}^L_{i_{l_n}-\frac{1}{2}})\Big) \tag{6}$$

where

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

The values of p and u in $\widehat{F_s}$ can be obtained through solving a Riemann problem at the interface.

We also take

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

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

$$U_{II}^* = \frac{1}{x_{i_{l_n}+\frac{3}{2}} - x(t^*)}\Big(\int_{x(t^n)}^{x_{i_{l_n}+\frac{3}{2}}} U^n\mathrm{dx} - \Delta t(\widehat{F}^L_{i_{l_n}+\frac{3}{2}} - \widehat{F_s})\Big) \tag{7}$$

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

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

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

2. If $x(t^*) \in [x_{i_{l_n}+\frac{1}{2}}, x_{i_{l_n}+\frac{3}{2}}]$,

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

Now, we can compute the flux at the position $x_{i_{l_n}+1/2}$, see Figure 2:

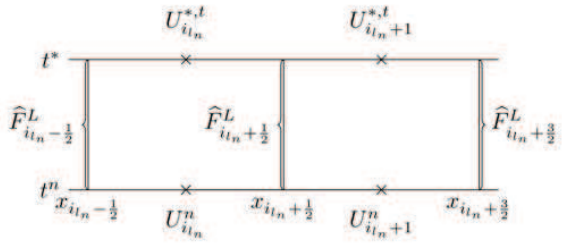


Figure 2: Sketch for computing $\widehat{F}^L$

From the left part, we have:

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

Then, we get:

$$\widehat{F}^L_{i_{l_n}+1/2} = \widehat{F}^L_{i_{l_n}-1/2} - \frac{U^{*,t}_{i_{l_n}} - U^n_{i_{l_n}}}{\frac{\Delta t}{\Delta x}} \tag{8}$$

From the right part, we have:

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

Then, we get:

$$\widehat{F}^L_{i_{l_n}+1/2} = \frac{U^{*,t}_{i_{l_n}+1} - U^n_{i_{l_n}+1}}{\frac{\Delta t}{\Delta x}} + \widehat{F}^L_{i_{l_n}+3/2} \tag{9}$$

Finally, we take an average of formula (8) and formula (9) to compute the flux $\widehat{F}^L$:

$$\widehat{F}^L_{i_{l_n}+1/2} = \frac{1}{2}\left(\widehat{F}^L_{i_{l_n}-1/2} - \frac{U^{*,t}_{i_{l_n}} - U^n_{i_{l_n}}}{\frac{\Delta t}{\Delta x}}\right) + \frac{1}{2}\left(\frac{U^{*,t}_{i_{l_n}+1} - U^n_{i_{l_n}+1}}{\frac{\Delta t}{\Delta x}} + \widehat{F}^L_{i_{l_n}+3/2}\right) \tag{10}$$

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

$$\widehat{F}^n_{i+1/2} = \widehat{F}^H_{i_{l_n}+1/2} \qquad \text{for} \quad i \neq i_{l_n}-1 \quad \& \quad i_{l_n} \quad \& \quad i_{l_n}+1 \tag{11}$$

and

$$\widehat{F}^n_{i+1/2} = \widehat{F}^L_{i+1/2} + \varphi\left(\widehat{F}^H_{i+1/2} - \widehat{F}^L_{i+1/2}\right) \qquad \text{for} \quad i = i_{l_n}-1, i_{l_n}, i_{l_n}+1 \tag{12}$$

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

Then, we can advance the computational variables based on the formula (4).

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 $x(t^*) \in [x_{i_{l_*}-1/2}, x_{i_{l_*}+1/2}]$. Then, we have

- When $i < i_{l_*}$,

$$U_i^{*,I} = U_i^* \tag{13}$$

- When $i > i_{l_*}$,

$$U_i^{*,II} = U_i^* \tag{14}$$

- When $i = i_{l_*}$,

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

$$\begin{aligned} U_{i_{l_*}}^{*,II} &= \varphi U_{i_{l_*}}^* + (1-\varphi) U_{II}^* \\ U_{i_{l_*}}^{*,I} &= \frac{U_{i_{l_*}}^* - (1-\alpha_{i_{l_*}}^*) U_{i_{l_*}}^{*,II}}{\alpha_{i_{l_*}}^*} \end{aligned} \tag{15}$$

  where $U_{i_{l_*}}^*$ comes from formula (4), and U_{II}^* is the average value for fluid II which comes from formula (7).

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

$$\begin{aligned} U_{i_{l_*}}^{*,I} &= \varphi U_{i_{l_*}}^* + (1-\varphi) U_I^* \\ U_{i_{l_*}}^{*,II} &= \frac{U_{i_{l_*}}^* - \alpha_{i_{l_*}}^* U_{i_{l_*}}^{*,I}}{1-\alpha_{i_{l_*}}^*} \end{aligned} \tag{16}$$

  where $U_{i_{l_*}}^*$ comes from formula (4), and U_I^* is the average value for fluid I which comes from formula (6).

In the formula (15) and (16), we also take

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

Remark 1. *In formula (15), when the solution is smooth, the formula (4) which is used to compute the nodal value $U_{i_{l_*}}^*$ is high order accurate [22, 12]. Therefore, we can maintain*

high order accuracy if the nodal value $U_{i_{l_*}}^{*,I}$ for fluid I and the nodal value $U_{i_{l_*}}^{*,II}$ for fluid II are close to the nodal value $U_{i_{l_*}}^{*}$. 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 $U_{i_{l_*}}^{*,II}$ is close to U_{II}^{*}. Therefore, we introduce the discontinuity indicator in formula (15) which approaches one in the smooth region and tends to zero when discontinuity appears. Similarly, we apply the same idea to the formula (16).

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:

$$\phi_t + V_n|\nabla\phi| = 0 \tag{17}$$

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

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = -(v_{i+\frac{1}{2}}^{+}\sqrt{\max\left((a^{+})^2,(b^{-})^2\right)} + v_{i+\frac{1}{2}}^{-}\sqrt{\max\left((a^{-})^2,(b^{+})^2\right)}) \tag{18}$$

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 $\phi^\pm_x$ 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 [21, 18]. 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

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

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 [21]. 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 the function values, and the other one is for

the derivative values. Now, we will describe the detailed steps. We use w to denote the interpolation variable.

First, 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 method. It is used to compute the density, velocity, pressure and so on.

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:

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

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 in [11]:

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

The expression can be written out explicitly:

$$\begin{aligned}
\beta_0 &= \frac{13}{12}(w_{i-2} - 2w_{i-1} + w_i)^2 + \frac{1}{4}(w_{i-2} - 4w_{i-1} + 3w_i)^2 \\
\beta_1 &= \frac{13}{12}(w_{i-1} - 2w_i + w_{i+1})^2 + \frac{1}{4}(w_{i-1} - w_{i+1})^2 \\
\beta_2 &= \frac{13}{12}(w_i - 2w_{i+1} + w_{i+2})^2 + \frac{1}{4}(3w_i - 4w_{i+1} + w_{i+2})^2
\end{aligned} \tag{19}$$

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

$$\omega_r = \frac{\overline{\omega}_r}{\sum_k \overline{\omega_k}}, \quad \overline{\omega}_k = \frac{\gamma_k}{(\beta_k + \varepsilon)^2} \tag{20}$$

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:

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

Second, we describe the procedure to obtain the value $w_{x_i}^-$ using the WENO method. It is mainly used in the level set method, in the reinitialization procedure and in calculating the normal vector.

1. Based on the small stencils $S_r = \{x_{i-3+r}, x_{i-2+r}, x_{i-1+r}, x_{i+r}\}, r = 0, 1, 2$, and a big stencil $S = \{x_{i-3} \cdots x_{i+2}\}$, we obtain the polynomials $p_0(x), p_1(x), p_2(x)$ and $q(x)$. In fact, we only need the values of the derivatives of these polynomials at x_i, which read

$$\begin{aligned}
p_1'(x_i) &= \frac{11w_i - 18w_{i-1} + 9w_{i-2} - 2w_{i-3}}{6\Delta x} \\
p_2'(x_i) &= \frac{2w_{i+1} + 3w_i - 6w_{i-1} + w_{i-2}}{6\Delta x} \\
p_3'(x_i) &= -\frac{w_{i+2} - 6w_{i+1} + 3w_i + 2w_{i-1}}{6\Delta x} \\
q'(x_i) &= -\frac{2w_{i-3} - 15w_{i-2} + 60w_{i-1} - 20w_i - 30w_{i+1} + 3w_{i+2}}{60\Delta x}
\end{aligned} \tag{21}$$

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

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

for all possible value of w_i. Then we have:

$$\gamma_0 = 0.1, \quad \gamma_1 = 0.6, \quad \gamma_2 = 0.3.$$

3. We calculate the smoothness indicator:

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

4. Based on (20), we compute the nonlinear weights, denoted as $\omega_0, \omega_1, \omega_2$. Then the final expression is:

$$w_{x_i}^{-} = \sum_{k=0}^{2} \omega_k p_k'(x_i)$$

The $w_{x_i}^{+}$ is mirror symmetric with respect to x_i of the above procedure.

2.2.4 Discontinuity indicator

The idea of the discontinuity indicator $\varphi_{i+\frac{1}{2}}$ comes from [31]. 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 (5); (2) We use the indicator to make a convex combination of the fluxes $\widehat{F}^H$ and the fluxes $\widehat{F}^L$, see formula (12); (3) We use the indicator to update the nodal values for fluid I and fluid II, see formula (15). 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 (5) satisfies*

$$\frac{\widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}}}{\Delta x} = F(U)_x \Big|_{x_i} + \mathcal{O}(\Delta x^5) \tag{22}$$

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 (22), it is sufficient to prove the relation:

$$\widehat{F}_{i+\frac{1}{2}} = \widehat{\widehat{F}}_{i+\frac{1}{2}} + \mathcal{O}(\Delta x^5) \tag{23}$$

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

According to (5), 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 component by component 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

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

□

Proposition 2.2. *The convex combination of $\widehat{F}^H$ and $\widehat{F}^L$ (12) satisfies formula (23), if the parameter $\varphi = 1 + \mathcal{O}(\Delta x^5)$ and $\Delta t = \mathcal{O}(\Delta x^r), r \leq 2$.*

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

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

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

Therefore, if we require $\varphi = 1 + \mathcal{O}(\Delta x^4)$, then we have

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

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

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

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

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

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:

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

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

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

So, we have

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

As to the flux, we have

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

and

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

Therefore, according to Figure 1, we have

$$\begin{aligned} U^{*,t}_{i_{l_n}} = U^*_I =& \frac{1}{\alpha^*_I}\left(\alpha^n_I U^n_I - \Delta t(\widehat{F_s} - \widehat{F}^L_{i_{l_n}-\frac{1}{2}})\right) \\ =& \frac{\left(\alpha^n_I U^n_I + \Delta t u U^n_I - \Delta t(\widehat{F_s} - F_{s_{i_{l_n}-\frac{1}{2}}})\right) + \mathcal{O}(\Delta x \Delta t)}{\alpha^*_I} \\ =& \frac{(\alpha^n_I U^n_I + \Delta t u U^n_I) + \mathcal{O}(\Delta x \Delta t)}{\alpha^*_I} \\ =& U^n_I + \mathcal{O}(\Delta t) = U(x_{i_{l_n}}, t^n) + \mathcal{O}(\Delta x) \end{aligned} \tag{24}$$

Similarly, we can also obtain

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

If $\Delta t = \mathcal{O}(\Delta x^r), r \leq 2$, according to formula (10) we obtain

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

That means the flux $\widehat{F}^L_{i_{l_n}+1/2}$ would be at least zero order accuracy to approximate the flux $\widehat{\widehat{F}}_{i_{l_n}+1/2}$. Due to $\varphi = 1 + \mathcal{O}(\Delta x^5)$, as to the formula (12), we have

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

To conclude, if $\varphi = 1 + \mathcal{O}(\Delta x^5)$ and $\Delta t = \mathcal{O}(\Delta x^r), r \leq 2$, then the convex combination of $\widehat{F}^H$ and $\widehat{F}^L$ (12) satisfies formula (23). □

Proposition 2.3. *In formulas (15) and (16), if $\varphi = 1 + \mathcal{O}(\Delta x^4)$, then we have*

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

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

Proof. We take $i_{l_*} = i_{l_n} + 1$ as example. The proof for $i_{l_*} = i_{l_n}$ or $i_{l_*} = i_{l_n} - 1$ is similar.

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

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

According to (24), We have

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

Then, we obtain

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

Therefore, we should have

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

□

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

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

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 $\Delta t = \mathcal{O}(\Delta x^r), r \leq 2$. 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:

$$p_0(x_{i+l}) = w_{i+l} \quad l = -2, \cdots, 3$$

$$p_1(x_{i+l}) = w_{i+l} \quad l = -2, \cdots, 0$$

$$p_2(x_{i+l}) = w_{i+l} \quad l = -1, \cdots, 1$$

$$p_3(x_{i+l}) = w_{i+l} \quad l = 0, \cdots, 2$$

$$p_4(x_{i+l}) = w_{i+l} \quad l = 1, \cdots, 3$$

Step 2. We use same the recipe as in [11] to compute the smoothness indicators for each polynomial on cell $[x_i, x_{i+1}]$:

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

We have the following expressions:

$$\begin{aligned}
\beta_0 =& (-w_i + w_{i+1})^2 + \frac{1421461}{2275} \left(\frac{1}{48} w_{i-2} - \frac{1}{16} w_{i-1} + \frac{1}{24} w_i + \frac{1}{24} w_{i+1} - \frac{1}{16} w_{i+2} + \frac{1}{48} w_{i+3} \right)^2 \\
&+ \frac{13}{3} \left(-\frac{131}{3120} w_{i-2} + \frac{391}{1040} w_{i-1} - \frac{521}{1560} w_i - \frac{521}{1560} w_{i+1} + \frac{391}{1040} w_{i+2} - \frac{131}{3120} w_{i+3} \right)^2 \\
&+ \frac{781}{20} \left(\frac{617}{43736} w_{i-2} - \frac{31123}{131208} w_{i-1} + \frac{14019}{21868} w_i - \frac{14019}{21868} w_{i+1} + \frac{31123}{131208} w_{i+2} - \frac{617}{43736} w_{i+3} \right)^2 \\
&+ \frac{21520059541}{1377684} \left(-\frac{1}{120} w_{i-2} + \frac{1}{24} w_{i-1} - \frac{1}{12} w_i + \frac{1}{12} w_{i+1} - \frac{1}{24} w_{i+2} + \frac{1}{120} w_{i+3} \right)^2 \\
\beta_1 =& (w_{i-2} - 3w_{i-1} + 2w_i)^2 + \frac{13}{3} \left(\frac{1}{2} w_{i-2} - w_{i-1} + \frac{1}{2} w_i \right)^2 \\
\beta_2 =& (w_{i+1} - w_i)^2 + \frac{13}{3} \left(\frac{1}{2} w_{i+1} - w_i + \frac{1}{2} w_{i-1} \right)^2 \\
\beta_3 =& (w_{i+1} - w_i)^2 + \frac{13}{3} \left(\frac{1}{2} w_{i+2} - w_{i+1} + \frac{1}{2} w_i \right)^2 \\
\beta_4 =& (-2w_{i+1} + 3w_{i+2} - w_{i+3})^2 + \frac{13}{3} \left(\frac{1}{2} w_{i+1} - w_{i+2} + \frac{1}{2} w_{i+3} \right)^2
\end{aligned}$$

Step 3. Similar to [32], 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:

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

It has the following properties:

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

Next, we will verify these properties:

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

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

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

$$\begin{aligned}
\beta_0 =& w^{(1)}(x_{i+1/2})^2 \Delta x^2 + \left(\frac{13}{12} w^{(2)}(x_{i+1/2})^2 + \frac{1}{12} w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^4 \\
&+ \left(\frac{7}{80} w^{(2)}(x_{i+1/2}) w^{(4)}(x_{i+1/2}) + \frac{1}{960} w^{(1)}(x_{i+1/2}) w^{(5)}(x_{i+1/2}) + \frac{1043}{960} w^{(3)}(x_{i+1/2})^2\right) \Delta x^6 + \mathcal{O}(\Delta x^8) \\
=& w^{(1)}(x_{i+1/2})^2 \Delta x^2 \left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_1 =& w^{(1)}(x_{i+1/2})^2 \Delta x^2 + \left(\frac{13}{12} w^{(2)}(x_{i+1/2})^2 - \frac{23}{12} w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^4 \\
&+ \left(-\frac{13}{4} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) + 2 w^{(1)}(x_{i+1/2}) w^{(4)}(x_{i+1/2})\right) \Delta x^5 + \mathcal{O}(\Delta x^6) \\
=& w^{(1)}(x_{i+1/2})^2 \Delta x^2 \left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_2 =& w^{(1)}(x_{i+1/2})^2 \Delta x^2 + \left(\frac{13}{12} w^{(2)}(x_{i+1/2})^2 + \frac{1}{12} w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^4 \\
&+ \left(-\frac{13}{12} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^5 + \mathcal{O}(\Delta x^6) \\
=& w^{(1)}(x_{i+1/2})^2 \Delta x^2 \left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_3 =& w^{(1)}(x_{i+1/2})^2 \Delta x^2 + \left(\frac{13}{12} w^{(2)}(x_{i+1/2})^2 + \frac{1}{12} w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^4 \\
&+ \left(\frac{13}{12} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^5 + \mathcal{O}(\Delta x^6) \\
=& w^{(1)}(x_{i+1/2})^2 \Delta x^2 \left(1 + \mathcal{O}(\Delta x^2)\right) = \mathcal{O}(\Delta x^2) \\
\beta_4 =& w^{(1)}(x_{i+1/2})^2 \Delta x^2 + \left(\frac{13}{12} w^{(2)}(x_{i+1/2})^2 - \frac{23}{12} w^{(1)}(x_{i+1/2}) w^{(3)}(x_{i+1/2})\right) \Delta x^4 \\
&+ \left(\frac{13}{4} w^{(2)}(x_{i+1/2}) w^{(3)}(x_{i+1/2}) - 2 w^{(1)}(x_{i+1/2}) w^{(4)}(x_{i+1/2})\right) \Delta x^5 + \mathcal{O}(\Delta x^6)
\end{aligned}$$

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

Therefore, we obtain the following relation:

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

It is easy to verify

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

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:

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

2.2.5 Time discretization

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

$$\begin{cases} \mathrm{U}^{(1)} = \mathrm{U}^n + \Delta t \mathcal{L}(\mathrm{U}^n) \\ \mathrm{U}^{(2)} = \mathrm{U}^n + \dfrac{1}{4}\Delta t \left(\mathcal{L}\left(\mathrm{U}^n\right) + \mathcal{L}\left(\mathrm{U}^{(1)}\right) \right) \\ \mathrm{U}^{n+1} = \mathrm{U}^n + \dfrac{1}{6}\Delta t \left(\mathcal{L}\left(\mathrm{U}^n\right) + 4\mathcal{L}\left(\mathrm{U}^{(2)}\right) + \mathcal{L}\left(\mathrm{U}^{(1)}\right) \right) \end{cases} \tag{25}$$

In fact, we have only described the first step of the third order Runge-Kutta method (25). 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 (25). 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 (5) 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 (5)-(10), 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 (15)-(16)-(13)-(14).

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 $U_i^{(1),I}$, $U_i^{(1),II}$, $\widehat{F}_{i+1/2}^n$, $\phi_i^{(1)}$. Up to now, we have finished the first step of the third order TVD Runge-Kutta method (25).

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

We construct $\widehat{F}^H$ for the second step of the third order TVD Runge-Kutta method (25):

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

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, $\phi_i^{(2)}$, $\widehat{F}_{i+1/2}^H$, $\Delta t/2$, 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 (25).

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

We construct $\widehat{F}^H$ for the third step of the third order TVD Runge-Kutta method (25):

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

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}, $\widehat{F}_{i+1/2}^H$, 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 (25).

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

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

and

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

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 $\varphi = \min(\varphi_{i_{l_n}-1/2}, \varphi_{i_{l_n}+1/2}, \varphi_{i_{l_n}+3/2})$.

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:

$$U_t + F(U)_x + G(U)_y = 0, \tag{26}$$

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

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

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 $\{U_{i,j}^{n}\}$ to next time level t^*

$$U_{i,j}^{*} = U_{i,j}^{n} - \frac{\Delta t}{\Delta x}(\widehat{F}_{i+\frac{1}{2},j}^{n} - \widehat{F}_{i-\frac{1}{2},j}^{n}) - \frac{\Delta t}{\Delta y}(\widehat{G}_{i,j+\frac{1}{2}}^{n} - \widehat{G}_{i,j-\frac{1}{2}}^{n}) \tag{27}$$

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

$$\widehat{F}_{i+\frac{1}{2},j}^{H} = \widehat{F}(U_{i+\frac{1}{2},j}^{-}, U_{i+\frac{1}{2},j}^{+}) - \varphi_{i+\frac{1}{2},j} \left(\frac{1}{24}\Delta x^2 F_{xx}\Big|_{x_{i+\frac{1}{2}}, y_j} - \frac{7}{5760}\Delta x^4 F_{xxxx}\Big|_{x_{i+\frac{1}{2}}, y_j} \right) \tag{28}$$

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

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 $I_{i_{l_n},j_{k_n}}$ contains the interface, it is not suitable to be computed for a full time step. Therefore it is suggested that the cell $I_{i_{l_n},j_{k_n}}$ 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: $A_I^{(r)}(t)$ for fluid I and $A_{II}^{(r)}(t)$ for fluid II. Now, we begin to deduce the formula for $A_I^{(r)}(t)$.

We integrate the equation (26) in space and time, then we have

$$\int_{t^n}^{t^*} \mathrm{dt} \iint_{A_I^{(r)}(t)} \Big(U_t + F(U)_x + G(U)_y\Big) \mathrm{dxdy} = 0$$

According to the Gauss theorem, we obtain

$$\int_{t^n}^{t^*} \mathrm{dt} \iint_{A_I^{(r)}(t)} U_t \mathrm{dxdy} + \int_{t^n}^{t^*} \mathrm{dt} \oint_{\partial A_I^{(r)}(t)} \Big(F(U)\cdot n_x + G(U)\cdot n_y\Big) \mathrm{ds} = 0 \tag{29}$$

Then, we use the Reynold's transport theorem:

$$\frac{d}{dt} \iint_{A_I^{(r)}(t)} U \mathrm{dxdy} = \iint_{A_I^{(r)}(t)} U_t \mathrm{dxdy} + \oint_{\partial A_I^{(r)}(t)} U(\overrightarrow{V}\cdot\overrightarrow{n}) \mathrm{ds} \tag{30}$$

where $\overrightarrow{V}$ is the normal velocity along the interface. Combining (29) and (30), we get the following equation:

$$\int_{t^n}^{t^*} \frac{d}{dt} \iint_{A_I^{(r)}(t)} U \mathrm{dxdydt} + \int_{t^n}^{t^*} \mathrm{dt} \oint_{\partial A_I^{(r)}(t)} (F(U)\cdot n_x + G(U)\cdot n_y) - U(\overrightarrow{V}\cdot\overrightarrow{n}) \mathrm{ds} = 0 \tag{31}$$

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 (31):

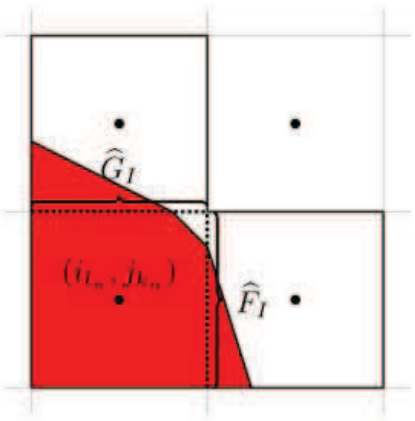


Figure 3: Example for the interface stencil. Line: the interface stencil; Dotted line: cells within the interface stencil. Red: fluid I.

$$
\begin{aligned}
U^*_{I,A^{(r)}(t^n)} = \frac{1}{\sum\limits_{I_{i,j}\in A_I^{(r)}(t^n)} \alpha^*_{i,j}} \Big(& \sum_{I_{i,j}\in A_I^{(r)}(t^n)} \alpha^n_{i,j} U^n_I - \sum_{I_{i,j}\in A^{(r)}(t^n)} \frac{\ell^n_{i,j}\Delta t}{\Delta x \Delta y} \widehat{F}_{si,j} \\
& - \frac{\Delta t}{\Delta x} \sum_{j=j_1}^{j_2} [S_{i^2(j)+\frac{1}{2},j} \widehat{F}^L_{i^2(j)+\frac{1}{2},j} - S_{i^1(j)-\frac{1}{2},j} \widehat{F}^L_{i^1(j)-\frac{1}{2},j}] \\
& - \frac{\Delta t}{\Delta y} \sum_{i=i_1}^{i_2} [S_{i,j^2(i)+\frac{1}{2}} \widehat{G}^L_{i,j^2(i)+\frac{1}{2}} - S_{i,j^1(i)-\frac{1}{2}} \widehat{G}^L_{i,j^1(i)-\frac{1}{2}}] \Big)
\end{aligned} \tag{32}
$$

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 Figure 3 for an example. Now, $\{I_{i,j} : i_{l_n} \le i \le i_{l_n}+1, j_{k_n} \le j \le j_{k_n}+1\}$ covers the interface stencil. When $j = j_{k_n}+1$, there is only one cell $I_{i_{l_n},j_{k_n}+1}$ inside the interface stencil. Hence we know that the maximum index in the x direction is $i^2(j) = i_{l_n}$, and the minimum index is also $i^1(j) = i_{l_n}$. When $j = j_{k_n}$, both the cell $I_{i_{l_n},j_{k_n}}$ and the cell $I_{i_{l_n}+1,j_{k_n}}$ are inside the interface stencil. Hence the maximum index in the x direction is $i^2(j) = i_{l_n}+1$, and the minimum index is $i^1(j) = i_{l_n}$, 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:

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

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 $I_{i,j}$. $\widehat{F_s}_{i,j}$ is the interface flux:

$$\widehat{F_s} = (0, p \cdot n_x, p \cdot n_y, p(u \cdot n_x + v \cdot n_y))^T \tag{33}$$

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, $\widehat{F}^L_{i_{l_n}+1/2,j_{k_n}}$ and $\widehat{G}^L_{i_{l_n},j_{k_n}+1/2}$ in Figure 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 (27). 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

$$U^{*,I}_{i,j} = U^*_{i,j} \tag{34}$$

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

$$U_{i,j}^{*,II} = U_{i,j}^* \tag{35}$$

- 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:

$$\begin{aligned} U_{i,j}^{*,II} &= \varphi U_{i,j}^* + (1-\varphi) U_{II,A^{(r)}(t^n)}^* \\ U_{i,j}^{*,I} &= \frac{U_{i,j}^* - (1-\alpha_{i,j}^*) U_{i,j}^{*,II}}{\alpha_{i,j}^*} \end{aligned} \tag{36}$$

  where $U_{i,j}^*$ is obtained by equation (27). $U_{II,A^{(r)}(t^n)}^*$ is the cell average value for fluid II in the interface stencil $A^{(r)}(t^n)$ at time level t^*.

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

$$\begin{aligned} U_{i,j}^{*,I} &= \varphi U_{i,j}^* + (1-\varphi) U_{I,A^{(r)}(t^n)}^* \\ U_{i,j}^{*,II} &= \frac{U_{i,j}^* - \alpha_{i,j}^* U_{i,j}^{*,I}}{1-\alpha_{i,j}^*} \end{aligned} \tag{37}$$

  where $U_{i,j}^*$ is obtained by equation (27), $U_{I,A^{(r)}(t^n)}^*$ is the cell average value for fluid I in the interface stencil $A^{(r)}(t^n)$ at time level t^*.

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

Now, we have fully described our two-dimensional scheme. Summarizing the previous steps, we have the following algorithm:

Algorithm 3

Input: $U_{i,j}^{n,I}$, $U_{i,j}^{n,II}$, $\phi_{i,j}^n$, Δt, Δx, Δy
Output: $U_{i,j}^{*,I}$, $U_{i,j}^{*,II}$, $\widehat{F}_{i+1/2,j}^n$, $\widehat{G}_{i,j+1/2}^n$, $\phi_{i,j}^*$

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

Similar to the one-dimensional case, we can extend **Algorithm 3** and obtain the following:

Algorithm 4

Input: $U_{i,j}^{n,I}$, $U_{i,j}^{n,II}$, $\phi_{i,j}^{n}$, $\phi_{i,j}^{*}$, $\widehat{F}_{i+1/2,j}^{H}$, $\widehat{G}_{i,j+1/2}^{H}$, Δt, Δx, Δy
Output: $U_{i,j}^{*,I}$, $U_{i,j}^{*,II}$

1: Apply $U_{i,j}^{n,I}$ and $U_{i,j}^{n,II}$ to define the computational variable $U_{i,j}^{n}$.
2: Compute the fluxes $\widehat{F}^{L}$ and $\widehat{G}^{L}$.
3: Define the fluxes $\widehat{F}^{n}$ and $\widehat{G}^{n}$.
4: Update the computational variable $U_{i,j}^{*}$.
5: Define the nodal values $U_{i,j}^{*,I}$ for fluid I, and nodal values $U_{i,j}^{*,II}$ for fluid II based on the equations (34)-(35)-(36)-(37).

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

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

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

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

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 Figure 4(b). Based on the trapezoidal area formula, we have

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

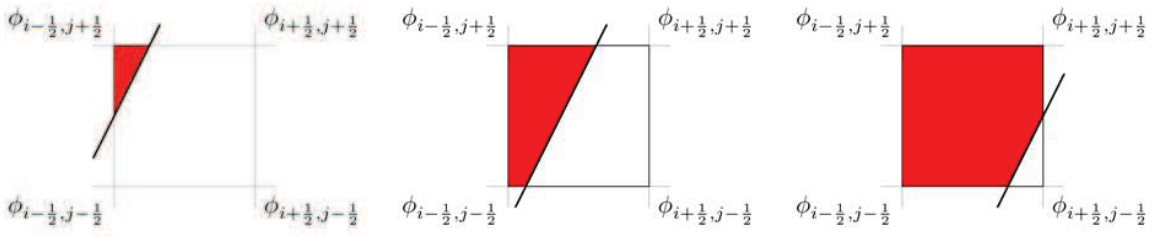
4. $\phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0$, $\phi_{i-\frac{1}{2},j-\frac{1}{2}} < 0$, $\phi_{i+\frac{1}{2},j-\frac{1}{2}} > 0$, $\phi_{i+\frac{1}{2},j+\frac{1}{2}} < 0$

See Figure 4(c). Based on the triangle area formula, we have:

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

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 $\alpha_{i,j} = 1$.



(a) case 2 (b) case 3 (c) case 4

Figure 4: Red: The part for $\phi < 0$.

3.1.2 Mixing procedure

In this subsection, we will introduce the way to generate the 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:

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

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 [13]

- 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 Figure 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 B in Figure 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 Figure 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 Figure 5.

Then, we can obtain the interface stencils.

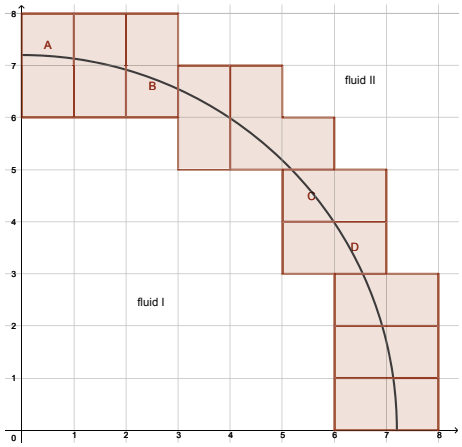


Figure 5: Formation of the interface stencils. Pink: the interface stencil. Curve: the interface. Fluid I: $\phi < 0$. Fluid II: $\phi > 0$

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:

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

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:

$$\phi_t + V_n|\nabla\phi| = 0 \tag{38}$$

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 [16]:

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

Here, q is the extended variable. Then, we can obtain the semi-discrete scheme of equation (38):

$$\begin{aligned} \frac{\mathrm{d}\phi}{\mathrm{d}t} = -(&v^+_{i+\frac{1}{2},j+\frac{1}{2}}\sqrt{\max\left((a^+)^2,(b^-)^2\right)+\max\left((c^+)^2,(d^-)^2\right)} \\ &+v^-_{i+\frac{1}{2},j+\frac{1}{2}}\sqrt{\max\left((a^-)^2,(b^+)^2\right)+\max\left((c^-)^2,(d^+)^2\right)}) \end{aligned} \tag{39}$$

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 (39). The detailed steps can be seen in subsection 3.1.6.

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 [27]:

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

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.

3.1.6 Flowchart in 2D

Step 1. According to **Algorithm 3**, input $U^{n,I}_{i,j}$, $U^{n,II}_{i,j}$, $\phi^n_{i,j}$, Δt, Δx, Δy, output $U^{(1),I}_{i,j}$, $U^{(1),II}_{i,j}$, $\widehat{F}^n_{i+1/2,j}$, $\widehat{G}^n_{i,j+1/2}$, $\phi^{(1)}_{i,j}$. Up to now, we have finished the first step of the third order TVD Runge-Kutta method (25).

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

We construct $\widehat{F}^H$ for the second step of the third order TVD Runge-Kutta method (25):

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

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

$$\phi^{(2)}_{i,j} = \frac{3}{4}\phi^n_{i,j} + \frac{1}{4}\widetilde{\phi}^{(2)}_{i,j}.$$

Then, according to **Algorithm 4**, input $U_{i,j}^{n,I}$, $U_{i,j}^{n,II}$, $\phi_{i,j}^{n}$, $\phi_{i,j}^{(2)}$, $\widehat{F}_{i+1/2,j}^{H}$, $\widehat{G}_{i,j+1/2}^{H}$, $\Delta t/2$, Δx, Δy, output $U_{i,j}^{(2),I}$, $U_{i,j}^{(2),II}$. Up to now, we have finished the second step of the third order TVD Runge-Kutta method (25).

Step 3. According to **Algorithm 3**, input $U_{i,j}^{(2),I}$, $U_{i,j}^{(2),II}$, $\phi_{i,j}^{(2)}$, Δt, Δx, Δy, output $\widetilde{U}_{i,j}^{(3),I}$, $\widetilde{U}_{i,j}^{(3),II}$, $\widehat{F}_{i+1/2,j}^{(2)}$, $\widehat{G}_{i,j+1/2}^{(2)}$, $\widetilde{\phi}_{i,j}^{(3)}$.

We construct $\widehat{F}^{H}$ for the third step of the third order TVD Runge-Kutta method (25):

$$\widehat{F}_{i+\frac{1}{2},j}^{H}=\frac{1}{6}\left(\widehat{F}_{i+\frac{1}{2},j}^{n}+4\widehat{F}_{i+\frac{1}{2},j}^{(2)}+\widehat{F}_{i+\frac{1}{2},j}^{(1)}\right),\quad \widehat{G}_{i,j+\frac{1}{2}}^{H}=\frac{1}{6}\left(\widehat{G}_{i,j+\frac{1}{2}}^{n}+4\widehat{G}_{i,j+\frac{1}{2}}^{(2)}+\widehat{G}_{i,j+\frac{1}{2}}^{(1)}\right)$$

and the distance function $\phi^{(3)}$ at the time level $t^{(3)}=t^{n}+\Delta t$:

$$\phi_{i,j}^{(3)}=\frac{1}{3}\phi_{i,j}^{n}+\frac{2}{3}\widetilde{\phi}_{i,j}^{(3)}.$$

Then, we use the reinitialization procedure, and obtain the distance function ϕ^{n+1}.

Then, according to **Algorithm 2**, input $U_{i,j}^{n,I}$, $U_{i,j}^{n,II}$, $\phi_{i,j}^{n}$, $\phi_{i,j}^{n+1}$, $\widehat{F}_{i+1/2,j}^{H}$, $\widehat{G}_{i,j+1/2}^{H}$, Δt, Δx, Δy, output $U_{i,j}^{n+1,I}$, $U_{i,j}^{n+1,II}$. Up to now, we have finished the third step of the third order TVD Runge-Kutta method (25).

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

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

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.

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.

Example 1. *Artificial accuracy test in 1D*

We consider the artificial accuracy test [9]. We take $\gamma = 3$. The initial conditions are:

$$\rho(x,0) = \frac{1 + 0.2\sin(x)}{2\sqrt{3}} \quad u(x,0) = \sqrt{\gamma}\rho(x,0) \quad p(x,0) = \rho(x,0)^\gamma$$

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 $2\sqrt{3}\rho(x,t)$ 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 1. We can see that our method can achieve the designed fifth order of accuracy.

Example 2. *A pure interface problem in 1D*

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

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

Table 1: Accuracy test for density in 1D.

	L_∞ error	order	L_2 error	order	L_1 error	order
80	1.57E-04		3.51E-05		1.10E-05	
120	2.86E-05	4.19	5.29E-06	4.67	1.56E-06	4.81
160	6.97E-06	4.91	1.27E-06	4.96	3.62E-07	5.09
200	2.42E-06	4.74	4.12E-07	5.05	1.16E-07	5.11
240	9.52E-07	5.12	1.63E-07	5.08	4.52E-08	5.16
280	4.44E-07	4.95	7.42E-08	5.10	2.04E-08	5.17
320	2.24E-07	5.13	3.75E-08	5.11	1.02E-08	5.16

The computational domain is $[0, 1]$. We set the final time $T = 0.32$ and $N = 200$. Figure 6 shows the result. The base velocity and pressure have been subtracted. From the figure, we can see that the interface propagates at the correct speed, and no oscillations other than those at the level of round-off errors are observed in velocity and pressure when using our scheme.

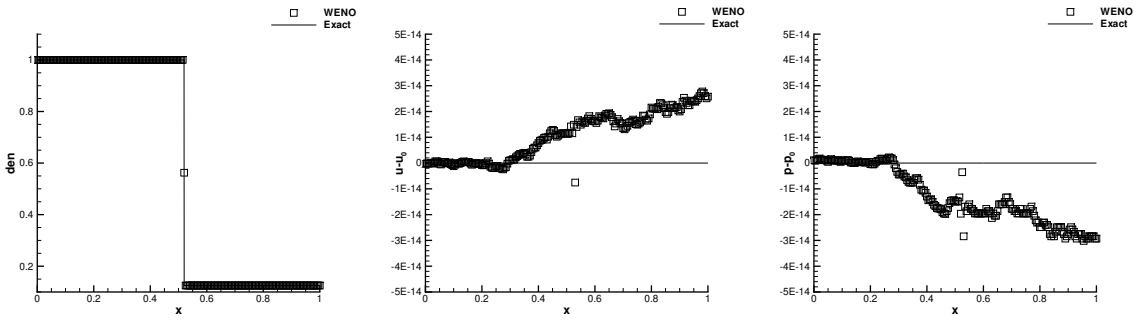


Figure 6: Pure interface problem. From left to right: density, velocity, pressure. Square: numerical solution (in velocity and pressure, the base has been subtracted); Line: exact solution (in velocity and pressure, the base has been subtracted).

Example 3. *Shock interacting with sine waves*

This example is tested in [29]. We solve the Euler equation with the following initial conditions

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

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 Figure 7. The reference solution is obtained with $N = 20000$. We can see that the solution with the coarse mesh matches well with the solution with the fine mesh.

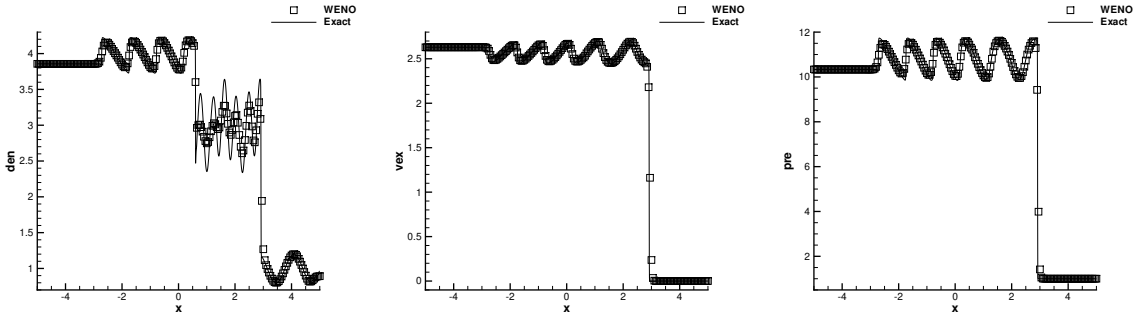


Figure 7: Shock interacting with sine waves. Line: N=20000; Square: N=300.

Example 4. *Shock impacting on a gas-gas interface in 1D, I*

This is an air-helium shock tube problem, see [7]. The initial condition is

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (\quad 1, 0, 10^5, 1.4, 0), & x < 0.5 \\ (0.125, 0, 10^4, 1.2, 0), & x \geq 0.5 \end{cases}$$

The computational domain is $[-1, 1]$. We plot the density, velocity and pressure at $T = 0.0007$ with $N = 200$ against the exact solution in Figure 8. We can see that our results are non-oscillatory near the interface, and reasonably well agreeing with the exact solution.

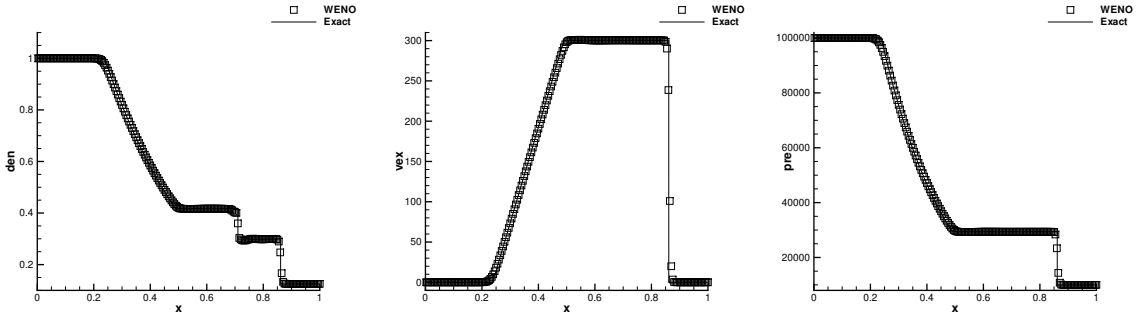


Figure 8: Shock impacting on a gas-gas interface in 1D, I. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

Example 5. *Shock impacting on a gas-gas interface in 1D, II*

This example is also taken from [7], with the initial conditions as:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (1.3333, 0.3535\sqrt{10^5}, 1.5 \times 10^5, \ 1.4, 0), & x < 0.05 \\ (\quad 1, \quad 0, \quad 10^5, \ 1.4, 0), & 0.05 < x \leq 0.5 \\ (0.1379, \quad 0, \quad 10^5, 5/3, 0), & x \geq 0.5 \end{cases}$$

The strength of the shock is $p_L/p_R = 1.5$ and initially located at $x = 0.05$. The initial interface of air and helium is located at $x = 0.5$. The computational domain is the same

as before. We plot the density, velocity and pressure with $N = 200$ against the exact solution at time $T = 0.0012$ in Figure 9. From the figure, we can see the resolution of the computed solution is acceptable.

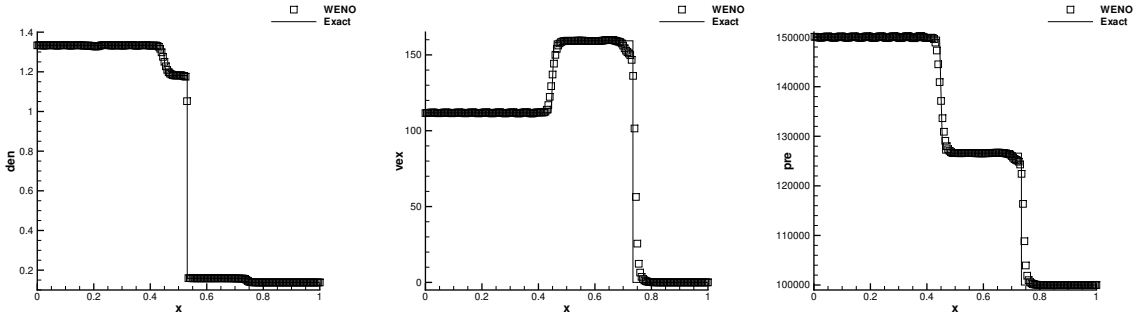


Figure 9: Shock impacting on a gas-gas interface in 1D, II. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

Example 6. *Shock impacting on a gas-gas interface in 1D, III*

We further increase the strength of the shock wave to $p_L/p_R = 15$ with the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (4.3333, 3.2817\sqrt{10^5}, 1.5 \times 10^6,\ 1.4, 0), & x < 0.05 \\ (\quad 1, \quad 0, \quad 10^5,\ 1.4, 0), & 0.05 < x \leq 0.5 \\ (0.1379, \quad 0, \quad 10^5, 5/3, 0), & x \geq 0.5 \end{cases}$$

The computational domain and the locations of the shock and the interface are the same as before. We plot the result at the time $T = 0.0005$. From the Figure 10, we can see that we have obtained satisfactory high resolution results.

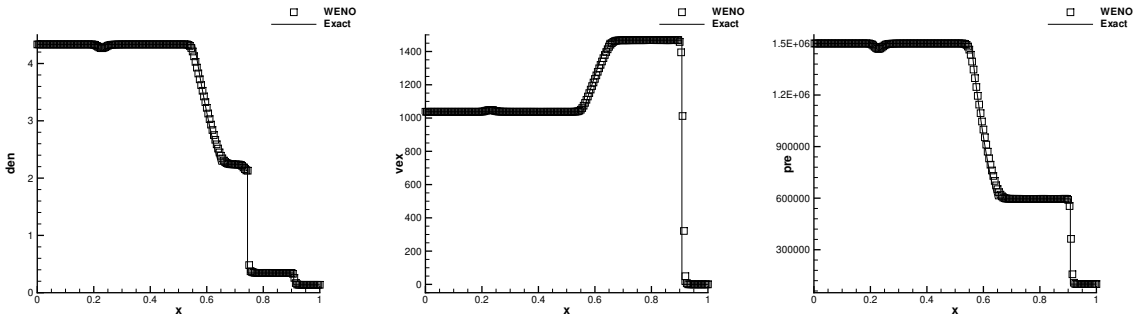


Figure 10: Shock impacting on a gas-gas interface in 1D, III. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

Example 7. *Shock impacting on a gas-gas interface in 1D, IV*

We continue to consider Euler equation with the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (1.3333, 0.3535\sqrt{10^5}, 1.5 \times 10^5, \quad 1.4, 0), & x < 0.05 \\ (\quad 1, \quad 0, \quad 10^5, \quad 1.4, 0), & 0.05 < x \leq 0.5 \\ (3.1538, \quad 0, \quad 10^5, 1.249, 0), & x \geq 0.5 \end{cases}$$

The computational domain and the locations of the shock and the interface are the same as before. We plot the result at the time $T = 0.0017$. From Figure 11, we can see that the interface of the two-medium is located at the correct position, and our method can achieve high resolution for this problem.

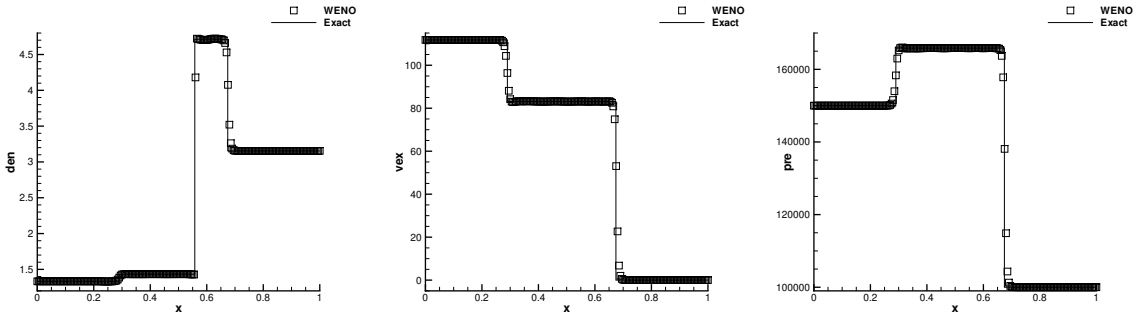


Figure 11: Shock impacting on a gas-gas interface in 1D, IV. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

Example 8. *Shock impacting on a gas-gas interface in 1D, V*

We increase the strength of the shock to $p_L/p_R = 15$ with the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (4.3333, 3.2817\sqrt{10^5}, 1.5 \times 10^6, \quad 1.4, 0), & x < 0.05 \\ (\quad 1, \quad 0, \quad 10^5, \quad 1.4, 0), & 0.05 < x \leq 0.5 \\ (3.1538, \quad 0, \quad 10^5, 1.249, 0), & x \geq 0.5 \end{cases}$$

The computational domain and the locations of the initial shock and the interface are the same as before. We plot the result at time $T = 0.0007$. From Figure 12, we can see that our scheme can still obtain non-oscillatory solution for this tough test case.

Example 9. *Shock impacting on a gas-water interface in 1D, I*

This is a gas-water shock tube problem with the initial condition as follows:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (1270, 0, 8 \times 10^8, \ 1.4, \quad 0), & x < 0.5 \\ (1000, 0, \quad 10^5, 7.15, 3.309 \times 10^8), & x \geq 0.5 \end{cases}$$

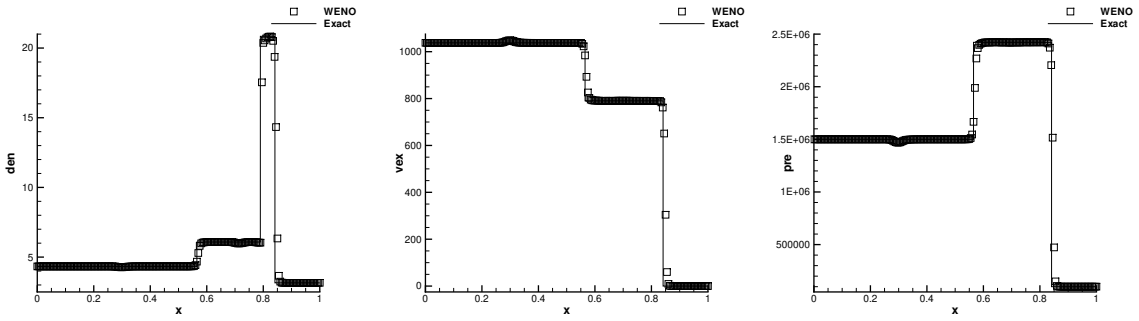


Figure 12: Shock impacting on a gas-gas interface in 1D, V. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

The computational domain is $[-1, 1]$. We list our final result at time $T = 0.00016$ in Figure 13. From the figure, we can see that, although the initial pressure is extremely high, our scheme can still obtain good result for this problem.

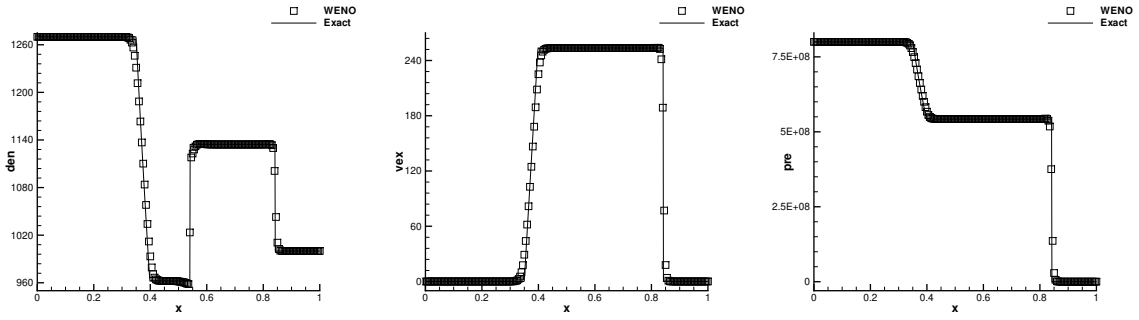


Figure 13: Shock impacting on a gas-water interface in 1D, I. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

Example 10. *Shock impacting on a gas-water interface in 1D, II*

We continue to increase the energy of the gas with the following initial condition:

$$(\rho, u, p, \gamma, \overline{p}) = \begin{cases} (1630, 0, 7.81 \times 10^9, \ 1.4, \qquad 0), & x < 0.5 \\ (1000, 0, \qquad 10^5, 7.15, 3.309 \times 10^8), & x \geq 0.5 \end{cases}$$

The computational domain is the same as before. We plot the numerical result at time $t = 0.0001$. From Figure 14, we can see that correct interface location and high resolution have been obtained by our method.

Example 11. *Artificial accuracy test in 2D*

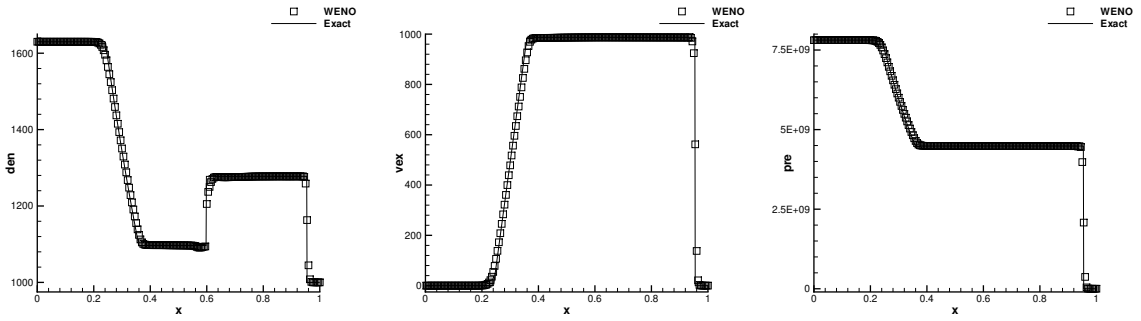


Figure 14: Shock impacting on a gas-water interface in 1D, II. $N = 200$. From left to right: density, velocity, pressure. Square: numerical solution; Line: exact solution.

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

$$\rho(x,y,0) = \frac{1 + 0.2\sin(\frac{x+y}{2})}{\sqrt{6}} \quad u(x,y,0) = v(x,y,0) = \sqrt{\frac{\gamma}{2}}\rho(x,y,0) \quad p(x,y,0) = \rho(x,y,0)^\gamma$$

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 $\sqrt{6}\rho(x,t)$ is the exact solution of the following Burgers equation:

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

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 2. We can see that our method can achieve the designed fifth order accuracy.

Example 12. *Pure interface problem in 2D*

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

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

The computational domain is $[0, 1] \times [0, 1]$. We set the final time $T = 0.32$. Figure 15 shows the result. The solutions along the cut line $y = x$ are also shown in the figures.

Table 2: 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

The base velocity and pressure have been subtracted. From the figure, 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 13. *Shock impacting on a gas-gas interface in 2D, I*

We next consider an air shock impacting on a helium bubble. The schematic for this problem is given in Figure 16 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 [19], 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 Figure 17, we can see that the main features of the solution are correctly captured.

Example 14. *Shock impacting on a water-gas interface in 2D, II*

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

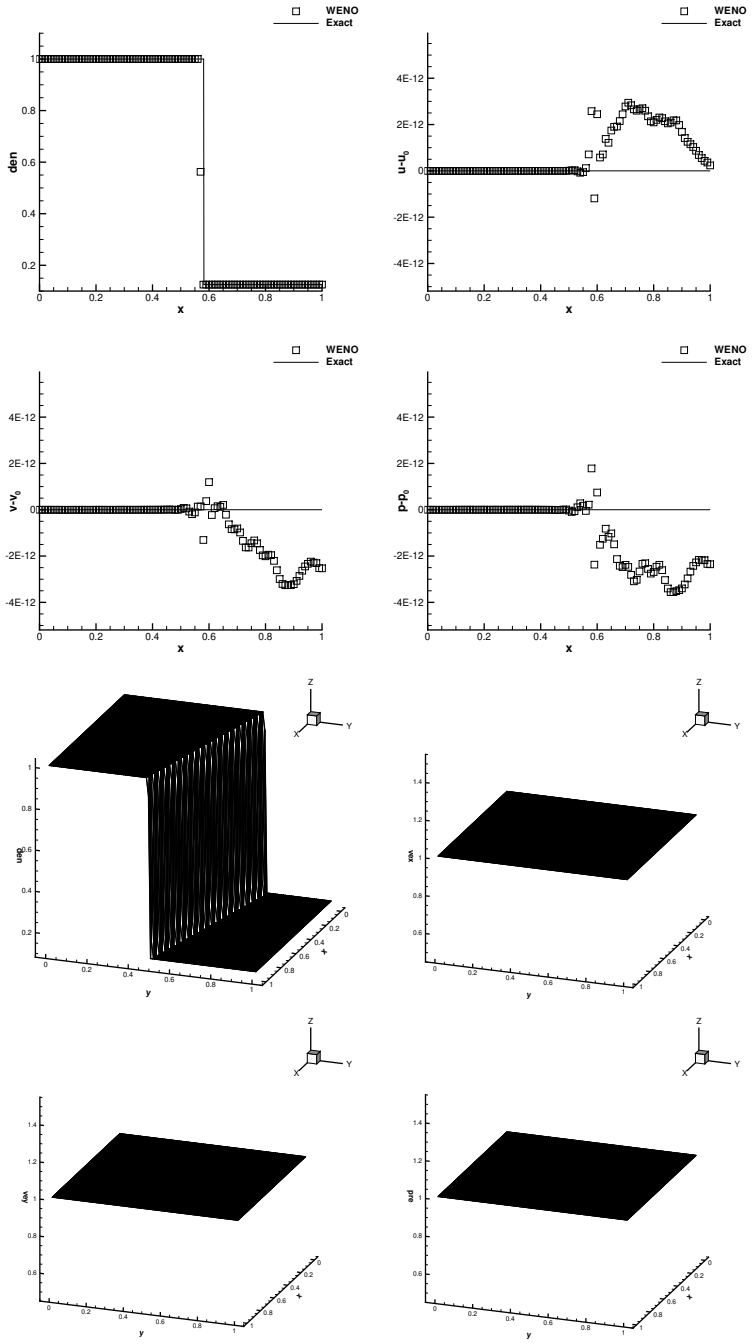


Figure 15: Pure interface problem. 100×100 cells. From top to bottom: density, velocity in the x direction, velocity in the y direction, pressure. From left to right: solution in the whole domain, solution along the cut line $y = x$. Square: numerical solution (in velocity and pressure, the base has been subtracted); Line: exact solution (in velocity and pressure, the base has been subtracted).

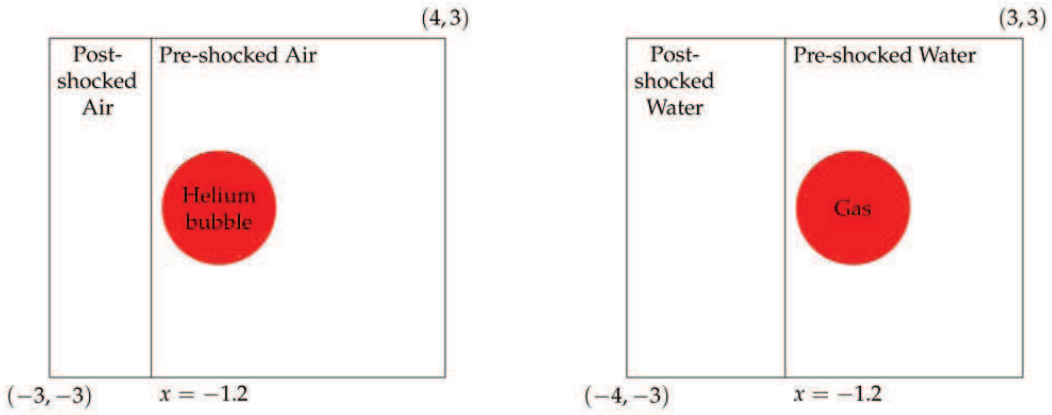


Figure 16: Schematic for Example 13 and Example 14. Left: Example 13; Right: Example 14

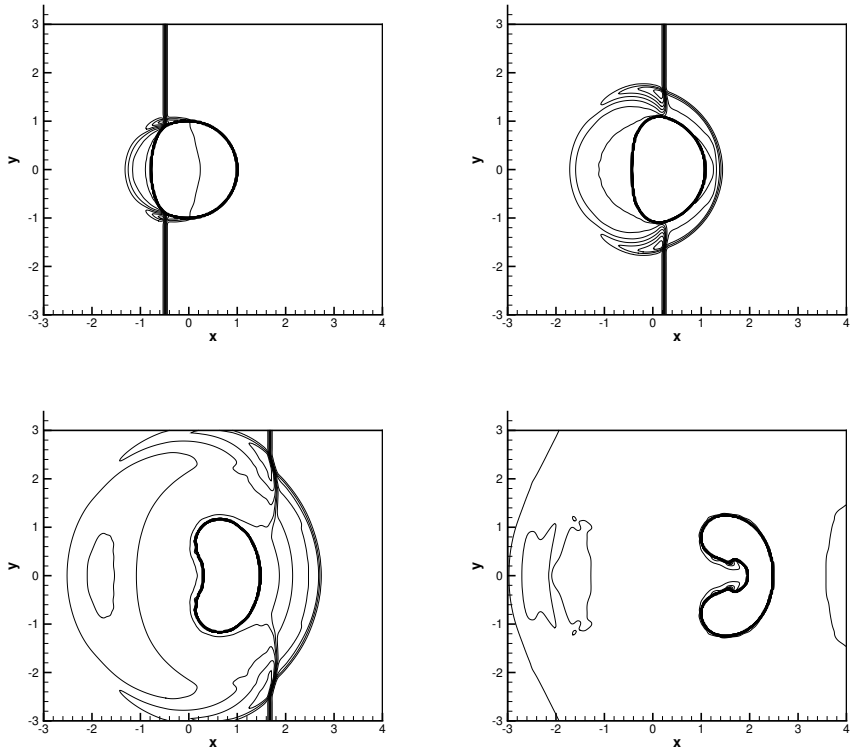


Figure 17: Shock impacting on a water-gas interface in 2D, I, with 280×240 cells, 30 equally spaced density contours from 0.1 to 1.6. Top left: $t = 0.5$; Top right: $t = 1.0$; Bottom left: $t = 2.0$; Bottom right: $t = 4.0$.

bubble. The schematic for this problem is given in Figure 16. 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 helium and $\phi > 0$ represents the air. 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 Figure 18 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.

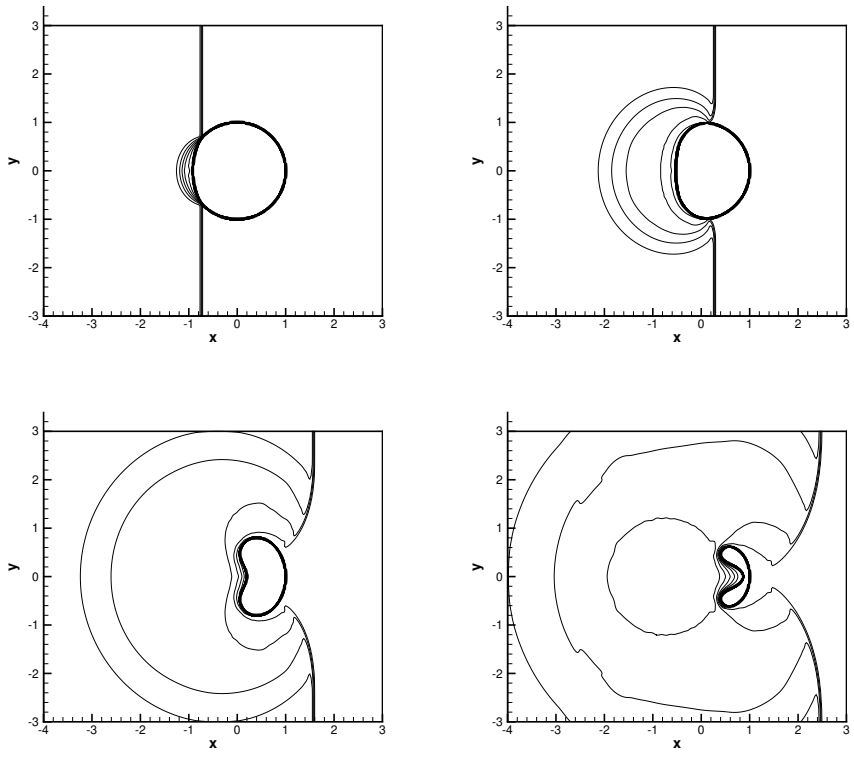


Figure 18: Shock impacting on a water-gas interface in 2D, II, with 280×240 cells, 30 equally spaced density contours from 0 to 1200. Top left: $t = 0.06$; Top right: $t = 0.19$; Bottom left: $t = 0.357$; Bottom right: $t = 0.471$.

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.

6 Acknowledgement

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.

References

[1] R. Abgrall. How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach. *Journal of Computational Physics*, 125:150–160, 1996.

[2] R. Abgrall and S. Karni. Computations of compressible multifluids. *Journal of Computational Physics*, 169:594–623, 2001.

[3] R. Abgrall and R. Saurel. Discrete equations for physical and numerical compressible multiphase mixtures. *Journal of Computational Physics*, 186:361–396, 2003.

[4] G. Allaire, S. Clerc and S. Kokh. A Five-equation model for the simulation of interfaces between compressible fluids. *Journal of Computational Physics*, 181:577–616, 2002.

[5] S. Bertoluzza, S. Falletta, G. Russo, and C.-W. Shu. *Numerical Solutions of Partial Differential Equations*. Birkhäuser Verlag, 2009.

[6] A. Chertock, S. Karni, and A. Kurganov. Interface tracking method for compressible multifluids. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42:991–1019, 2008.

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

[8] R.P. Fedkiw, A. Marquina, and B. Merriman. An isobaric fix for the overheating problem in multimaterial compressible flows. *Journal of Computational Physics*, 148:545–578, 1999.

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

[10] X.Y. Hu, B.C. Khoo, N.A. Adams, and F.L. Huang. A conservative interface method for compressible flows. *Journal of Computational Physics*, 219:553–578, 2006.

[11] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996.

[12] Y. Jiang, C.-W. Shu, and M. Zhang. An alternative formulation of finite difference WENO schemes with Lax-Wendroff time discretization for conservation laws. *SIAM Journal on Scientific Computing*, 35:A1137–A1160, 2013.

[13] J.-Y. Lin, Y. Shen, H. Ding, N.-S. Liu, and X.-Y. Lu. Simulation of compressible two-phase flows with topology change of fluid-fluid interface by a robust cut-cell method. *Journal of Computational Physics*, 328:140–159, 2017.

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

[15] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[16] D.P. Peng, B. Merriman, S. Osher, H.K. Zhao, and M. Kang. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 155:410–438, 1999.

[17] J.X. Qiu, T.G. Liu, and B.C. Khoo. Runge-Kutta discontinuous Galerkin methods for compressible two-medium flow simulations: One-dimensional case. *Journal of Computational Physics*, 222:353–373, 2007.

[18] J. Qiu and C.-W. Shu. On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. *Journal of Computational Physics*, 183:187–209,2002.

[19] J.J. Quirk and S. Karni. On the dynamics of a shock-bubble interaction. *Journal of Fluid Mechanics*, 318:129–163,1996.

[20] R. Saurel and R. Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21:1115–1145, 1999.

[21] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, Berlin, 1998, pp.325-432.

[22] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.

[23] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *Journal of Computational Physics*, 83:32–78, 1989.

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

[25] K.-M. Shyue. A fluid-mixture type algorithm for compressible multicomponent flow with van der Waals equation of state. *Journal of Computational Physics*, 156:43–88, 1999.

[26] K.-M. Shyue. A fluid-mixture type algorithm for compressible multicomponent flow with Mie-Grüneisen equation of state. *Journal of Computational Physics*, 171:678–707, 2001.

[27] M. Sussman, P. Smereka, and S. Osher. A PDE-based fast local level set method. *Journal of Computational Physics*, 134:146–159, 1994.

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

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

[30] T. Xiong, C.-W. Shu, and M. Zhang. WENO scheme with subcell resolution for computing nonconservative Euler equations with applications to one-dimensional compressible two-medium flows. *Journal of Scientific Computing*, 53:222–247, 2012.

[31] Z. Xu and C.-W. Shu. Anti-diffusive flux corrections for high order finite difference WENO schemes. *Journal of Computational Physics*, 205:458–485, 2005.

[32] J. Zhu and J.X. Qiu. A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *Journal of Computational Physics*, 318:110–121, 2016.