Computers and Fluids 284 (2024) 106424

Contents lists available at ScienceDirect

Computers and Fluids

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

High-order finite volume method for solving compressible multicomponent flows with Mie–Grüneisen equation of state ☆

Feng Zheng [a], Jianxian Qiu [b,*]

[a] School of Mathematics and Statistics, Fujian Key Laboratory of Mathematical Analysis and Application, Fujian Normal University, Fuzhou, Fujian 350117, PR China

[b] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling & High-Performance Scientific Computing, Xiamen University, Xiamen Fujian 361005, PR China

ARTICLE INFO

Keywords:
Finite volume
High-order
Multicomponent flows
Mie–Grüneisen equation of state

ABSTRACT

In this paper, we propose a new high-order finite volume method for solving the multicomponent fluids problem with Mie–Grüneisen EOS. Firstly, based on the cell averages of conservative variables, we develop a procedure to reconstruct the cell averages of the primitive variables in a high-order manner. Secondly, the high-order reconstructions employed in computing numerical fluxes are implemented in a characteristic-wise manner to reduce numerical oscillations as much as possible and obtain high-resolution results. Thirdly, advection equation within the governing system is rewritten in a conservative form with a source term to enhance the scheme's performance. We utilize integration by parts and high-order numerical integration techniques to handle the source terms. Finally, all variables are evolved by using Runge–Kutta time discretization. All steps are carefully designed to maintain the equilibrium of pressure and velocity for the interface-only problem, which is crucial in designing a high-resolution scheme and adapting to more complex multicomponent problems. We have performed extensive numerical tests for both one- and two-dimensional problems to verify our scheme's high resolution and accuracy.

1. Introduction

Multicomponent flows play an essential role in aerospace, chemical engineering, biomedical engineering, hydraulic engineering, and other fields. Due to their importance and applicability, we must develop high-accuracy and high-resolution numerical methods to model these flows numerically.

The interface-capturing method is one of the main approaches for solving multicomponent flows. These methods utilize a system of governing equations to characterize the flows, replace sharp interfaces with a diffused zone by introducing numerical dissipation, and dynamically capture the interface. In general, interface-capturing methods can handle large and complex interface deformations, allowing for the dynamic creation or disappearance of interfaces, such as those found in chemical reactions. These methods can be integrated into a unified governing system. Therefore, we can pursue a unified numerical method to model the flows. Furthermore, interface-capturing methods are based on the Eulerian framework, where the grids remain fixed, which facilitates generalization to high-dimensional cases. Due to the advantages mentioned above, interface-capturing methods are becoming increasingly popular in numerically modeling multicomponent flows.

Interface-capturing methods for solving the multicomponent flows are challenging to design. The main difficulty is that they would produce nonphysical oscillations near the interface. These oscillations are already present in first-order schemes and are hard to eliminate even using high-order schemes. Over decades, researchers have proposed many studies based on interface-capturing method frameworks [1–4]. In [5], Abgrall proposed a quasi-conservative scheme to maintain velocity and pressure equilibrium along the interface, which was a breakthrough in developing a high-resolution interface-capturing method. Later, Shyue developed wave propagation method for solving more general equations of state (EOS), such as stiffened gas EOS, van der Waals EOS, and the Mie–Grüneisen equation of states, and extended to second order [6–8]. In [9], Allaire et al. proposed a five-equation model to simulate the multicomponent flows and applied the model to different types of EOS. In [10], Johnsen and

☆ The research is supported partly by National Key R & D Program of China (Grant Number 2022YFA1004500) , National Natural Science Foundation of China (Grant Nos. 12101128, 12071392).

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

https://doi.org/10.1016/j.compfluid.2024.106424
Received 20 May 2024; Received in revised form 1 September 2024; Accepted 4 September 2024
Available online 7 September 2024

Colonius suggested that using the quasi-conservative form coupled with the reconstruction of primitive variables can eliminate spurious oscillations regardless of whether WENO techniques are implemented in a component-wise or characteristic-wise way. Following this idea, researchers have designed numerous schemes to solve the problems of multicomponent flows [11–19].

Many interface-capturing methods are based on the finite volume method, such as [10–14]. All of them claim to be high-order methods, but none can be regarded as genuinely high-order ones. On one hand, the primitive variables used in the reconstructions in these methods are obtained in a low-order way. On the other hand, the source terms obtained by rewriting the advection equations in the governing system are also handled with a low order of accuracy. They may not achieve optimal accuracy in smooth regions. Therefore, it is necessary to develop finite volume methods that can handle multicomponent fluid problems with genuinely high-order accuracy.

In this paper, we propose a new high-order finite volume method for solving the multicomponent fluids problem based on Mie–Grüneisen EOS. Firstly, based on given conservative variables, we develop a procedure to reconstruct the primitive variables in a high-order manner. Secondly, the high-order reconstructions employed in computing numerical fluxes are implemented in a characteristic-wise way to reduce numerical oscillations as much as possible and obtain high-resolution results. Thirdly, the advection equation within the governing system is rewritten in a conservative form with a source term to enhance the scheme's performance. We utilize integration by parts and high-order numerical integration techniques to handle the source terms. Finally, all variables are evolved by using Runge–Kutta time discretization. All steps are carefully designed to maintain the equilibrium of pressure and velocity for the interface-only problem, which is crucial in designing a high-resolution scheme and adapting to more complex multicomponent problems. We have performed extensive numerical tests for both one- and two-dimensional problems to verify our scheme's high resolution and accuracy.

The rest of this paper is organized as follows. In Section 2, we outline the detailed steps of our scheme in a one-dimensional case. In Section 3, we present numerical experiments to verify the numerical accuracy and efficiency of the scheme. In Section 4, we give a conclusion of the paper. In the appendix, we provide detailed proof to demonstrate that our scheme can preserve the equilibrium of the pressure and velocity for the interface-only problem when using the stiffened gas EOS.

2. The framework for the one-dimensional case

We consider the following system of five equations, which can be used to solve multicomponent problems:

$$\begin{cases} \left(\rho_1 z_1\right)_t + \left(\rho_1 z_1 u\right)_x = 0, \\ \left(\rho_2 z_2\right)_t + \left(\rho_2 z_2 u\right)_x = 0, \\ (\rho u)_t + \left(\rho u^2 + p\right)_x = 0, \\ E_t + (u(E+p))_x = 0, \\ \left(z_1\right)_t + u\left(z_1\right)_x = 0. \end{cases} \tag{2.1}$$

Here ρ_1 and ρ_2 are the density of the fluid 1 and 2, ρ is the total density, u is the velocity, ρu is the total momentum, E is the total energy, p is the pressure, $z_1, z_2 \in [0,1]$ are the volume fractions of fluid 1 and fluid 2, and satisfy $z_1 + z_2 = 1$. Furthermore, we have the following equation:

$$\rho = \rho_1 z_1 + \rho_2 z_2, \qquad E = \frac{1}{2}\rho u^2 + z_1\rho_1 e_1 + z_2\rho_2 e_2,$$

where e_1, e_2 are specific internal energy for fluid 1 and fluid 2 respectively. In order to close system (2.1), a mixture equation of state is needed, and each fluid is modeled by Mie–Grüneisen EOS:

$$p_k = p_{ref,k}(\rho_k) + \Gamma_k(\rho_k)\rho_k\left(e_k - e_{ref,k}(\rho_k)\right), \quad k = 1, 2,$$

where p_k is the pressure for fluid k, Γ_k, $p_{ref,k}$, and $e_{ref,k}$ are the Grüneisen coefficient, reference pressure and reference internal energy for fluid k. Physically, Γ_k, $p_{ref,k}$ and $e_{ref,k}$ are used to describe the property of materials and can be determined from experimental data. In this paper, we will focus on following different models in the Mie–Grüneisen form [8]:

1. Stiffened gas EOS,
$$\begin{aligned} \Gamma_k(\rho_k) &= \gamma_k - 1, \\ p_{ref,k}(\rho_k) &= -\gamma_k B_k, \\ e_{ref,k}(\rho_k) &= 0, \end{aligned}$$
where γ_k and B_k are the material-dependent quantities.
2. The Jones–Wilkins–Lee (JWL) EOS (for gaseous explosives),
$$\begin{aligned} \Gamma_k(\rho_k) &= \Gamma_{0,k}, \\ p_{ref,k}(\rho_k) &= A_k exp\left(-\frac{R_{1,k}\rho_{0,k}}{\rho_k}\right) + B_k exp\left(-\frac{R_{2,k}\rho_{0,k}}{\rho_k}\right), \\ e_{ref,k}(\rho_k) &= \frac{A_k}{R_{1,k}\rho_{0,k}} exp\left(-\frac{R_{1,k}\rho_{0,k}}{\rho_k}\right) + \frac{B_k}{R_{2,k}\rho_{0,k}} exp\left(-\frac{R_{2,k}\rho_{0,k}}{\rho_k}\right) - e_{0,k}, \end{aligned}$$
where $\Gamma_{0,k}, A_k, B_k, R_{1,k}, R_{2,k}, \rho_{0,k}$ and $e_{0,k}$ are the material-dependent quantities.
3. The Cochran-Chan (CC) EOS (for solid explosives),
$$\begin{aligned} \Gamma_k(\rho_k) &= \Gamma_{0,k}, \\ p_{ref,k}(\rho_k) &= A_k\left(\frac{\rho_{0,k}}{\rho_k}\right)^{-\varepsilon_{1,k}} - B_k\left(\frac{\rho_{0,k}}{\rho_k}\right)^{-\varepsilon_{2,k}}, \\ e_{ref,k}(\rho_k) &= -\frac{A_k}{(1-\varepsilon_{1,k})\rho_{0,k}}\left(\left(\frac{\rho_{0,k}}{\rho_k}\right)^{1-\varepsilon_{1,k}} - 1\right) + \frac{B_k}{(1-\varepsilon_{2,k})\rho_{0,k}}\left(\left(\frac{\rho_{0,k}}{\rho_k}\right)^{1-\varepsilon_{2,k}} - 1\right) - e_{0,k}, \end{aligned}$$
where $\Gamma_{0,k}, A_k, B_k, \varepsilon_{1,k}, \varepsilon_{2,k}, \rho_{0,k}$ and $e_{0,k}$ are the material-dependent quantities.

Table 1
Typical material-dependent quantities for different models.

JWL EOS	ρ_0 (kg/m^3)	A (GPa)	B (GPa)	R_1	R_2	Γ_0	α
TNT	1840	854.5	20.5	4.6	1.35	0.25	0
Water	1004	1582	−4.67	8.94	1.45	1.17	0
CC EOS	ρ_0 (kg/m^3)	A (GPa)	B (GPa)	ϵ_1	ϵ_2	Γ_0	α
Copper	8900	145.67	147.75	2.99	1.99	2	0
TNT	1840	12.87	13.42	4.1	3.1	0.93	0
Shock EOS	ρ_0 (kg/m^3)	c_0 (m/s)	s	Γ_0	α	p_0	e_0
Aluminum	2785	5328	1.338	2.0	1.0	0	0
Copper	8924	3910	1.51	1.96	1.0	0	0
Molybdenum	9961	4770	1.43	2.56	1.0	0	0
Midocean ridge basalt (MORB)	2660	2100	1.68	1.18	1.0	0	0
Water	1000	1483	2.0	2.0	10^{-4}	0	0

4. The shock wave EOS,

$$\Gamma_k(\rho_k) = \Gamma_{0,k}\left(\frac{\rho_{0,k}}{\rho_k}\right)^{\alpha_k},$$
$$p_{ref,k}(\rho_k) = p_{0,k} + \frac{c_{0,k}^2\left(1/\rho_{0,k} - 1/\rho_k\right)}{\left(1/\rho_{0,k} - s_k(1/\rho_{0,k} - 1/\rho_k)\right)^2},$$
$$e_{ref,k}(\rho_k) = e_{0,k} + \frac{1}{2}\left(p_{ref,k}(\rho_k) + p_{0,k}\right)(1/\rho_{0,k} - 1/\rho_k),$$

where $\Gamma_{0,k}, s_k, c_{0,k}, p_{0,k}, \rho_{0,k}, \alpha_k$ and $e_{0,k}$ are the material-dependent quantities.

These different models can be used to simulate many materials, such as water, copper, TNT, and so on. We list relevant parameters for different materials in Table 1 [8,20].

We assume that for each fluid the sound speed of sound c_k is defined by [8]:

$$c_k^2 = \left(\Gamma_k(\rho_k) + 1 + \rho_k\frac{\Gamma_k'(\rho_k)}{\Gamma_k(\rho_k)}\right)\left(\frac{p_k - p_{ref,k}(\rho_k)}{\rho_k}\right) + \Gamma_k(\rho_k)\frac{p_{ref,k}(\rho_k)}{\rho_k} + p_{ref,k}' - \Gamma_k(\rho_k)\rho_k e_{ref,k}'(\rho_k),$$

where, $\Gamma_k', p_{ref,k}', e_{ref,k}'$ are the derivatives of $\Gamma_k, p_{ref,k}, e_{ref,k}$ with respect to ρ_k. Furthermore, we use isobaric closure assumption for mixed cells:

$$p_1 = p_2 = p,$$

and thus we define sound speed for mixed cells [9]:

$$\xi c^2 = \sum_k y_k\xi_k c_k^2,$$

where c_k is the sound speed for fluid k, $y_k = z_k\rho_k/\rho$ is the mass fraction for fluid k, $\xi_k = \left(\frac{\partial\rho_k e_k}{\partial p_k}\right)_{\rho_k}$ is the partial derivative about the internal energy with respect to pressure for fluid k, and $\xi = z_1\xi_1 + z_2\xi_2$.

For simplicity, the computational domain is divided uniformly. We denote the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, the cell center $x_j = \left(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}\right)/2$ and its cell size $\Delta x = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$. By denoting $U = \left(z_1\rho_1, z_2\rho_2, \rho u, E, z_1\right)^T$, $F(U) = \left(z_1\rho_1 u, z_2\rho_2 u, \rho u^2 + p, u(E+p), uz_1\right)^T$ and $S(U) = \left(0,0,0,0,z_1\right)^T$, we rewrite the system (2.1) into

$$U_t + F(U)_x = S(U)u_x, \tag{2.2}$$

where u_x refers to the derivative of velocity. We integrate (2.2) over the cell I_j, employ the integration by part, and then obtain the following semi-discretization form:

$$\begin{aligned}&\frac{\partial U_j(t)}{\partial t} + \frac{1}{\Delta x}\left(F(U(x_{j+\frac{1}{2}},t)) - F(U(x_{j-\frac{1}{2}},t))\right)\\ =&\frac{1}{\Delta x}\left(S(U(x_{j+\frac{1}{2}},t))u(x_{j+\frac{1}{2}},t) - S(U(x_{j-\frac{1}{2}},t))u(x_{j-\frac{1}{2}},t) - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} S(U(x,t))_x u(x,t)\mathrm{d}x\right),\end{aligned} \tag{2.3}$$

where $U_j(t) = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} U(x,t)\mathrm{d}x$. Next, by introducing the numerical fluxes, we approximate Eq. (2.3) by the following formulation:

$$\frac{\partial U_j}{\partial t} + \frac{1}{\Delta x}\left(\hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}}\right) = \frac{1}{\Delta x}\left(S(U^-_{j+\frac{1}{2}})\hat{u}_{j+\frac{1}{2}} - S(U^+_{j-\frac{1}{2}})\hat{u}_{j-\frac{1}{2}}\right) - \sum_{k=1}^{4} w_k S(U)_{xG_k}u_{G_k}. \tag{2.4}$$

U_j refers to the numerical approximation to the cell average $U_j(t)$. $\hat{F}_{j+\frac{1}{2}}$ and $\hat{u}_{j+\frac{1}{2}}$ represents HLLC numerical flux evaluated at the interface $x_{j+1/2}$ [10,11,21]:

$$\begin{aligned}\hat{F}_{j+\frac{1}{2}} &= \frac{1+\mathrm{sgn}(s^*)}{2}\left(F(U^-_{j+\frac{1}{2}}) + s^-(U_{*L} - U^-_{j+\frac{1}{2}})\right) + \frac{1-\mathrm{sgn}(s^*)}{2}\left(F(U^+_{j+\frac{1}{2}}) + s^+(U_{*R} - U^+_{j+\frac{1}{2}})\right),\\ \hat{u}_{j+\frac{1}{2}} &= \frac{1+\mathrm{sgn}(s^*)}{2}\left(u^-_{j+\frac{1}{2}} + s^-(\chi_{*L} - 1)\right) + \frac{1-\mathrm{sgn}(s^*)}{2}\left(u^+_{j+\frac{1}{2}} + s^+(\chi_{*R} - 1)\right),\end{aligned} \tag{2.5}$$

where we defined

$$U_{*k} = \chi_{*k} \begin{pmatrix} \rho_{1,k} \\ \rho_{2,k} \\ \rho_k s_* \\ E_k + (s_* - u_k)\left(\rho_k s_* + \frac{p_k}{s_k - u_k}\right) \\ z_1 \end{pmatrix}, s_* = \frac{p_R - p_L + \rho_L u_L (s_L - u_L) - \rho_R u_R (s_R - u_R)}{\rho_L (s_L - u_L) - \rho_R (s_R - u_R)},$$

$$s^- = \min(0, s_L), s^+ = \max(0, s_R), s_L = \min\left((u-c)^{ROE}, u_L - c_L\right), s_R = \max\left((u+c)^{ROE}, u_R + c_R\right),$$

and

$$\chi_{*k} = \frac{s_k - u_k}{s_k - s_*}, \quad k = L, R,$$

where $q_L = q^-_{j+\frac{1}{2}}$ and $q_R = q^+_{j+\frac{1}{2}}$, q refers to variables $\rho_1, \rho_2, \rho, u, s, p, c, E, U_*, \chi_*$ respectively, and $(u-c)^{ROE}$ and $(u+c)^{ROE}$ are the velocities obtained from an intermediate state based on the Roe average. In Eq. (2.4), w_k, G_k are Gauss–Lobatto quadrature points and coefficients:

$$\begin{aligned} &G_1 = x_{j-\frac{1}{2}}, \quad G_2 = x_{j-\frac{\sqrt{5}}{10}}, \quad G_3 = x_{j+\frac{\sqrt{5}}{10}}, \quad G_4 = x_{j+\frac{1}{2}}, \\ &\omega_1 = \frac{1}{12}, \quad \omega_2 = \frac{5}{12}, \quad \omega_3 = \frac{5}{12}, \quad \omega_4 = \frac{1}{12}. \end{aligned} \tag{2.6}$$

$U^{\pm}_{j+\frac{1}{2}}$, U_{G_k} and U_{xG_k} are numerical approximation to the solutions and derivatives at the $x_{j+\frac{1}{2}}$ or G_k respectively, which are obtained by the WENO reconstruction method described in Section 2.1.

We denote the semi-discrete system (2.4) as $\mathrm{U}_t = \mathcal{L}(\mathrm{U})$, where $\mathcal{L}$ denotes the operator of the spatial discretization. Then, we use the third-order total variation diminishing (TVD) Runge–Kutta time discretization [22] to solve the semi-discrete form (2.4):

$$\begin{cases} \mathrm{U}^{(1)} = \mathrm{U}^n + \Delta t \mathcal{L}(\mathrm{U}^n), \\ \mathrm{U}^{(2)} = \frac{3}{4}\mathrm{U}^n + \frac{1}{4}(\mathrm{U}^{(1)} + \Delta t \mathcal{L}(\mathrm{U}^{(1)})), \\ \mathrm{U}^{n+1} = \frac{1}{3}\mathrm{U}^n + \frac{2}{3}(\mathrm{U}^{(2)} + \Delta t \mathcal{L}(\mathrm{U}^{(2)})). \end{cases} \tag{2.7}$$

2.1. WENO reconstruction

Now, we list detailed flowchart of the WENO reconstruction method, which is similar to [23–25].

Step 1. Given stencils $S_0 = \{I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2}\}$, $S_1 = \{I_i\}$, $S_2 = \{I_{i-1}, I_i, I_{i+1}\}$, $S_3 = \{I_{i-2}, I_{i-1}, I_i\}$, and $S_4 = \{I_i, I_{i+1}, I_{i+2}\}$, we need to construct polynomials $P_0(x)$, $P_1(x)$, $P_2(x)$, $P_3(x)$, $P_4(x)$ such that:

$$\begin{aligned} &\frac{1}{\Delta x}\int_{I_{i+l}} P_0(x)dx = \overline{q}_{i+l}, \quad l = -2, -1, 0, 1, 2, \\ &\frac{1}{\Delta x}\int_{I_{i+l}} P_1(x)dx = \overline{q}_{i+l}, \quad l = 0, \\ &\frac{1}{\Delta x}\int_{I_{i+l}} P_2(x)dx = \overline{q}_{i+l}, \quad l = -1, 0, 1, \\ &\frac{1}{\Delta x}\int_{I_{i+l}} P_3(x)dx = \overline{q}_{i+l}, \quad l = -2, -1, 0, \\ &\frac{1}{\Delta x}\int_{I_{i+l}} P_4(x)dx = \overline{q}_{i+l}, \quad l = 0, 1, 2, \end{aligned} \tag{2.8}$$

where $\overline{q}_i$ are the cell average on cell I_j for $q(x)$.

Step 2. We compute the smoothness indicators, denoted as $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$ respectively. The smoothness indicators are based on the formula in [26]:

$$\beta_m = \frac{1}{\Delta x}\sum_{k=1}^{r}\int_{I_i}\left(\Delta x^k \frac{\partial^k}{\partial x^k} P_m(x)\right)^2 dx, \qquad m = 0, 2, 3, 4,$$

where $r = 4$ for $P_0(x)$ and $r = 2$ for $P_2(x)$, $P_3(x)$ and $P_4(x)$. As to $P_1(x)$, we magnify smoothness indicators β_1 from zero to a value defined below:

$$\beta_1 = \min\left((\overline{q}_{i+1} - \overline{q}_i)^2, (\overline{q}_{i-1} - \overline{q}_i)^2\right).$$

Step 3. We take the linear weights as

$$r_0 = \frac{1100}{1111}, \quad r_1 = \frac{1}{1111}, \quad r_2 = \frac{10}{1111},$$

and

$$s_0 = 0.97, \quad s_2 = 0.01, \quad s_3 = 0.01, \quad s_4 = 0.01.$$

Then, we can rewrite $P_0(x)$ as

$$P_0(x) = \theta P_0(x) + (1-\theta)\left(r_0 P_0(x) + r_1 P_1(x) + r_2 P_2(x)\right). \tag{2.9}$$

where $\theta = 1 - \left(1 - \min\left(1, s_0/s_0\right)^4\right)^4$.

Step 4. We compute the nonlinear weights

$$\omega_k = \frac{\overline{\omega}_k}{\overline{\omega}_0 + \overline{\omega}_1 + \overline{\omega}_2}, \quad \overline{\omega}_k = \frac{r_k}{\left(\beta_k + \varepsilon\right)^2}, \quad k = 0, 1, 2,$$

and

$$\mu_0 = \frac{\overline{\mu}_0}{\overline{\mu}_0 + \overline{\mu}_2 + \overline{\mu}_3 + \overline{\mu}_4}, \quad \overline{\mu}_k = \frac{s_k}{\left(\beta_k + \varepsilon\right)^4}, \quad k = 0, 2, 3, 4,$$

where $\varepsilon = 10^{-12}$ to avoid dividing by zero. Then, we replace θ, r_0, r_1, r_2 in (2.9) with nonlinear weights:

$$P(x) = \theta P_0(x) + (1-\theta)\left(\omega_0 P_0(x) + \omega_1 P_1(x) + \omega_2 P_2(x)\right), \tag{2.10}$$

and

$$P'(x) = \theta P_0'(x) + (1-\theta)\left(\omega_0 P_0'(x) + \omega_1 P_1'(x) + \omega_2 P_2'(x)\right), \tag{2.11}$$

where $\theta = 1 - \left(1 - \min\left(1, \mu_0/s_0\right)^4\right)^4$ and satisfies $0 < \theta \leq 1$, θ is close to 0 in discontinuous region and $\theta = 1 + \mathcal{O}(\Delta x^4)$ in smooth region.

Now, we give a brief analysis about Eq. (2.10). The Eq. (2.11) is similar.

When the solution is smooth in the stencil S_0, through the Taylor expansion analysis, we have:

$$\mu_0 = s_0 + \mathcal{O}(\Delta x^2).$$

It implies the method can realize the fifth order accuracy:

$$\begin{aligned} P(x) - q(x) &= (1-\theta)\left(\omega_0\left(P_0(x) - q(x)\right) + \omega_1\left(P_1(x) - q(x)\right) + \omega_2\left(P_2(x) - q(x)\right)\right) + \theta(P_0(x) - q(x)) \\ &= \mathcal{O}(\Delta x^8)\left(\omega_1\mathcal{O}(\Delta x^5) + \omega_1\mathcal{O}(\Delta x^3) + \omega_2\mathcal{O}(\Delta x)\right) + \left(1 - \mathcal{O}(\Delta x^8)\right)\mathcal{O}(\Delta x^5) \\ &= \mathcal{O}(\Delta x^5). \end{aligned}$$

When the solution is discontinuous in the stencil S_0, we have $\beta_0 = \mathcal{O}(1)$. It means $\mu_0 \to 0$, and $\theta \to 0$. Therefore, $\omega_0 P_0(x) + \omega_1 P_1(x) + \omega_2 P_2(x)$ plays the major role. If the solution is smooth in the stencil S_m, then $\beta_m = \mathcal{O}(\Delta x^2)$. However, if the solution is discontinuous in the stencil S_m, then $\beta_m = \mathcal{O}(1)$. As to the nonlinear weights ω_m, we have $\omega_m = \mathcal{O}(\Delta x^4)$ when the solution is discontinuous in the stencil S_m, and $\omega_m = \mathcal{O}(1)$ when the solution is smooth in the stencil S_m. Therefore, the method maintain the ENO property.

2.2. Primitive variables reconstruction

To maintain the equilibrium of velocity and pressure during reconstruction, it is necessary to reconstruct the primitive variables from the given conservative variables. These reconstructed primitive variables will then be used to implement the reconstruction process subsequently. Next, we will describe the reconstruction steps for the primitive variables.

Step 1. Follow the steps described in Section 2.1, we can obtain the polynomials and fraction for each cell I_j:

$$P_{0,m}(x), P_{1,m}(x), P_{2,m}(x), \theta_m, \quad m = z_1\rho_1, z_2\rho_2, \rho u, E, z_1,$$

where $P_{0,m}(x), P_{1,m}(x), P_{2,m}(x)$ are the polynomials reconstructed in (2.8) for each variables, and θ_m is the parameter used in (2.10) and (2.11) for each variables.

Step 2. Compute the minimization of the fraction:

$$\theta = \min_m\left(\theta_m\right), m = z_1\rho_1, z_2\rho_2, \rho u, E, z_1.$$

Step 3. Compute the polynomials for the conservative variables:

$$P_m(x) = \theta P_{0,m}(x) + (1-\theta)P_{2,m}(x), \quad m = z_1\rho_1, z_2\rho_2, \rho u, E, z_1. \tag{2.12}$$

Step 4. Compute the values for the primitive variables:

$$\begin{aligned} P_{\rho_1}(x) &= \frac{P_{z_1\rho_1}(x)}{P_{z_1}(x)}, \\ P_{\rho_2}(x) &= \frac{P_{z_2\rho_2}(x)}{1.0 - P_{z_1}(x)}, \\ P_u(x) &= \frac{P_{\rho u}(x)}{P_\rho(x)}, \\ P_p(x) &= \left(P_E(x) - \frac{p_{\rho u}^2(x)}{2P_\rho(x)} - \mathcal{B}_0\right)/\mathcal{A}_0. \end{aligned}$$

where $P_\rho(x) = P_{z_1\rho_1}(x) + P_{z_2\rho_2}(x), \mathcal{A}(\rho) = 1/\Gamma(\rho), \mathcal{B}(\rho) = -p_{ref}(\rho)/\Gamma(\rho) + \Gamma(\rho)e_{ref}(\rho), \mathcal{A}_0 = P_{z_1}(x)\mathcal{A}(P_{\rho_1}(x)) + \left(1 - P_{z_1}(x)\right)\mathcal{A}(P_{\rho_2}(x))$ and $\mathcal{B}_0 = P_{z_1}(x)\mathcal{B}(P_{\rho_1}(x)) + \left(1 - P_{z_1}(x)\right)\mathcal{B}(P_{\rho_2}(x))$.

Step 5. Reconstruct the cell averages of primitive variables for each cell by using numerical integrals:

$$m_j = \sum_G \omega_G P_m(x_G), \quad m = \rho_1, \rho_2, u, p, z_1.$$

Table 2
The CPU time (measured in seconds).

	HO1	HO2	LO
Example 3	7.34	6.95	9.31
Example 4	0.55	0.53	0.78
Example 5	0.55	0.54	0.82
Example 6	1.38	1.34	2.42
Example 7	0.86	0.84	1.51
Example 8	0.94	0.90	1.42

2.3. The maximum principle preserving limiter

The volume fraction z_1 may obtain an invalid value, such as $z_1 < 0$ or $z_1 > 1$. We need a procedure to correct the invalid value. Following the suggestion in [27], we describe the detailed correction steps:

Case 1: two-components flows:

Assuming $z_{1j}(x)$ and $\overline{z}_{1j}$ are the polynomial and cell average on cell I_j respectively, we define the following new polynomial:

$$\widetilde{z}_{1j}(x) = \widetilde{\theta}\left(z_{1j}(x) - \overline{z}_{1j}\right) + \overline{z}_{1j}, \quad \widetilde{\theta} = \min\left\{\left|\frac{1-\epsilon-\overline{z}_{1j}}{z_{1j,max}-\overline{z}_{1j}}\right|, \left|\frac{\epsilon-\overline{z}_{1j}}{z_{1j,min}-\overline{z}_{1j}}\right|, 1\right\},$$

where $z_{1j,min} = \min_{x\in S} z_{1j}(x)$, $z_{1j,max} = \max_{x\in S} z_{1j}(x)$ and S is the set of Legendre Gauss–Lobatto quadrature points for I_j.

It is clear that if cell average $\overline{z}_{1j} \in [\epsilon, 1-\epsilon]$, then we have $\widetilde{z}_{1j}(G_k) \in [\epsilon, 1-\epsilon]$ and $\widetilde{z}_{2j}(G_k) = 1 - \widetilde{z}_{1j}(G_k) \in [\epsilon, 1-\epsilon]$, where $G_k \in S$. We set parameter $\epsilon = 10^{-6}$.

Case 2: three-components flows:

The procedure for more than two components is a little bit different. We make three-component flows as an example. Assuming $z_{1j}(x)$, $z_{2j}(x)$ and $z_{3j}(x)$ are the volume fraction polynomials of the fluid 1,2 and 3 satisfying:

$$z_{1j}(x) + z_{2j}(x) + z_{3j}(x) = 1.$$

For each fluid k, we define the parameter θ_k:

$$\widetilde{z}_{kj}(x) = \widetilde{\theta}_k\left(z_{kj}(x) - \overline{z}_{kj}\right) + \overline{z}_{kj}, \quad \widetilde{\theta}_k = \min\left\{\left|\frac{1-\epsilon-\overline{z}_{kj}}{z_{kj,max}-\overline{z}_{kj}}\right|, \left|\frac{\epsilon-\overline{z}_{kj}}{z_{kj,min}-\overline{z}_{kj}}\right|, 1\right\}, \quad k = 1,2,3.$$

We also set parameter $\epsilon = 10^{-6}$. Then, we define the modified polynomials as the following:

$$\widetilde{z}_{kj}(x) = \widetilde{\theta}\left(z_{kj}(x) - \overline{z}_{kj}\right) + \overline{z}_{kj}, \quad \widetilde{\theta} = \min\left\{\widetilde{\theta}_1, \widetilde{\theta}_2, \widetilde{\theta}_3\right\}, \quad k = 1,2,3.$$

2.4. Algorithm

Now, we list the algorithm of the high-order finite volume method for the multicomponent fluid problem, see Algorithm 1:

Algorithm 1 Algorithm of the scheme

1: Reconstruct the cell average of primitive variables.
2: Obtain the polynomials of primitive variables in characteristic-wise way.
3: Correct the volume fraction.
4: Compute $\hat{u}_{j\pm\frac{1}{2}}$, $\hat{F}_{j\pm\frac{1}{2}}$, $S(U^{\mp}_{j\pm\frac{1}{2}})$ and $S(U)_{xG_k}$ in formula (2.4) and form the scheme.
5: Evolve the scheme by TVD Runge–Kutta method (2.7).

2.5. Property of the schemes

Proposition 1. *The high-order finite volume scheme (2.4) with the numerical flux (2.5) preserves the equilibrium of the pressure and velocity for the interface-only problem with stiffened gas EOS.*

We will provide the detailed proof in Appendix.

3. Numerical test

In this section, we will present the numerical results of our high-order finite volume scheme (denoted as 'HO1'), with the CFL number set to 0.5. We set the time step $\Delta t = CFL\Delta x^{5/3}/\alpha$ for accuracy test and $\Delta t = CFL\Delta x/\alpha$ for other numerical experiments. Although the CFL number does not satisfy the maximum-principle-preserving requirement, it works well for our numerical experiments. For comparison, we also list the computational results based on [28] (denoted as 'HO2'), and the computational results based on [11] (denoted as 'LO') which utilize the same characteristic projection as in our paper but omit interface-sharpening technique. The exact solution in the one-dimensional numerical test can be found in [29,30]. We also list the CPU time of the 'HO1', 'HO2', and 'LO' in Table 2. From the table, we can see that the CPU time of the 'HO1' method and 'HO2' method are comparable, and are less than that of the 'LO' method.

Table 3
Accuracy test for density in 1D.

Mesh size	HO1				LO			
	L_∞ error	order	L_1 error	order	L_∞ error	order	L_1 error	order
10	3.88E−02		1.46E−02		2.99E−02		9.35E−03	
20	1.86E−02	1.06	2.42E−03	2.59	1.03E−02	1.54	1.55E−03	2.59
40	1.13E−03	4.04	1.25E−04	4.28	2.37E−03	2.11	3.25E−04	2.25
80	9.91E−05	3.52	6.10E−06	4.35	4.51E−04	2.40	7.24E−05	2.17
160	4.42E−06	4.49	2.19E−07	4.80	1.02E−04	2.14	1.75E−05	2.05
320	1.49E−07	4.89	7.07E−09	4.96	2.57E−05	2.00	4.35E−06	2.01

Table 4
Accuracy test for density in 2D.

Mesh size	HO1				LO			
	L_∞ error	order	L_1 error	order	L_∞ error	order	L_1 error	order
10 × 10	4.46E−02		1.33E−02		2.44E−02		7.27E−03	
20 × 20	8.11E−03	2.46	2.36E−03	2.50	6.57E−03	1.89	1.39E−03	2.39
40 × 40	1.25E−03	2.70	1.14E−04	4.37	1.99E−03	1.72	2.11E−04	2.72
80 × 80	8.64E−05	3.85	5.93E−06	4.27	3.67E−04	2.44	4.80E−05	2.14
160 × 160	4.14E−06	4.38	2.16E−07	4.78	7.52E−05	2.29	1.15E−05	2.06
320 × 320	1.47E−07	4.81	7.00E−09	4.95	1.83E−05	2.04	2.87E−06	2.01

Example 1. We consider the artificial accuracy test. In this test, we choose stiffened gas EOS by taking $\gamma = 3, B = 0$. Initial conditions are the following:

$$\rho(x,0) = \frac{1+0.2\sin(x)}{2\sqrt{\gamma}}, \quad \rho_1(x,0) = \rho(x,0)z_1(x,0), \quad \rho_2(x,0) = \rho(x,0)(1-z_1(x,0)),$$

$$z_1(x,0) = 0.5 + 0.4\sin(x), \quad u(x,0) = \sqrt{\gamma}\rho(x,0), \quad p(x,0) = \rho(x,0)^\gamma.$$

we set computational domain to be $[0, 2\pi]$ and employ periodic boundary conditions. By the special choice of the parameter γ, B, initial conditions and boundary conditions, we can verify that $2\sqrt{\gamma}\rho(x,t)$ is the exact solution of the following Burgers equation:

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

The velocity, pressure and z_1 satisfy the following relation:

$$u(x,t) = \sqrt{\gamma}\rho(x,t), \quad p(x,t) = \rho(x,t)^\gamma, \quad \gamma(x,0) = 3, \quad p_\infty(x,0) = 0.$$

It is straightforward to demonstrate that the solution of the Burgers equation remains smooth up to time $T = 5$. We set the final time as $T = 3$ and list the results in Table 3. From the table, we can see that our scheme can achieve fifth order accuracy as expected, while the LO scheme can only achieve second-order accuracy.

Example 2. We consider the 2D artificial accuracy test. We also choose stiffened gas EOS and take $\gamma = 3$ and $B = 0$. Then, the system (2.1) becomes a single-component problem. Furthermore, we choose the following special initial conditions:

$$\rho(x,y,0) = \frac{1+0.2\sin(\frac{x+y}{2})}{\sqrt{2\gamma}}, \quad z_1(x,y,0) = 0.5 + 0.4\sin(\frac{x+y}{2}), \quad u(x,y,0) = v(x,y,0) = \sqrt{\frac{\gamma}{2}}\rho(x,y,0),$$

$$\rho_1(x,y,0) = \rho(x,y,0)z_1(x,y,0), \quad \rho_2(x,y,0) = \rho(x,y,0)(1-z_1(x,y,0)), \quad p(x,y,0) = \rho(x,y,0)^\gamma.$$

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

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

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 remains smooth. We list the error and numerical accuracy order in Table 4. We can see that our method can achieve the designed fifth order accuracy, while the LO scheme can only reach second-order accuracy.

Example 3. We solve a Riemann problem consisting of a single contact discontinuity in gas dynamics with stiffened gas EOS. The initial condition and parameters required are as follows:

$$(\rho_1, \rho_2, u, p, z_1, \gamma, B) = \begin{cases} (1, \quad 0.125, \quad 1, \quad 1, \quad 1-10^{-6}, \quad 1.4, \quad 0), & x < 0.5, \\ (1, \quad 0.125, \quad 1, \quad 1, \quad 10^{-6}, \quad 4, \quad 1), & x \geq 0.5. \end{cases}$$

We set the computational domain as $[0, 1]$, periodic condition, the final time $T = 1$ and $N = 200$. We show our results without using the maximum principle preserving limiter in Fig. 1. The base velocity and pressure have been subtracted. From the figures, we can see that all of the three methods match the exact solution well. The errors in velocity and pressure of our method are a little bit smaller than the ones obtained by the LO scheme.

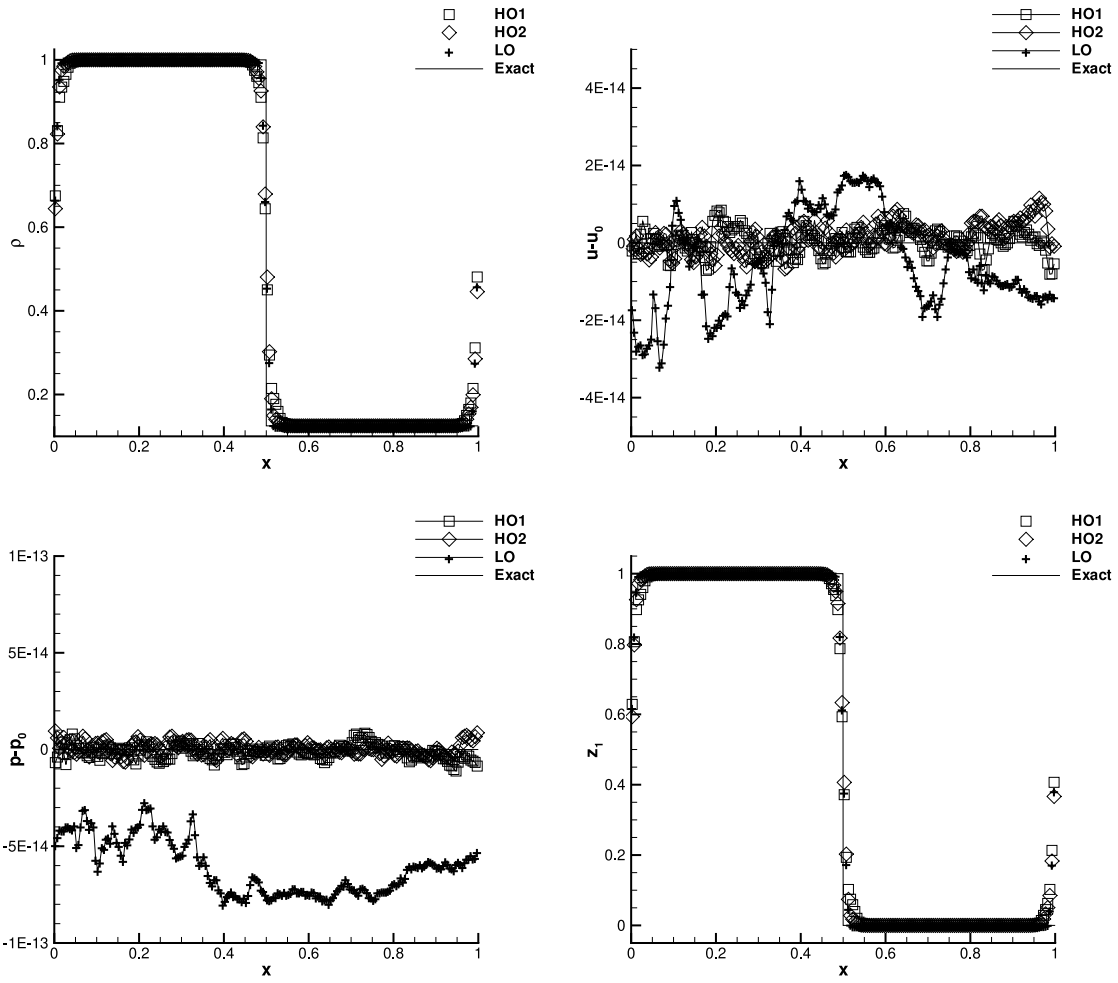

Fig. 1. Numerical results for Example 3. From left top to right bottom: figures of density, velocity, pressure, volume fraction of fluid 1. The base $u_0 = 1$ and $p_0 = 1$..

Example 4. We solve Riemann problem with the following initial condition and parameters required:

$$(\rho_1, \rho_2, u, p, z_1, e_0) = \begin{cases} (1700, & 1000, & 0, & 10^{12}, & 1-10^{-6}, & 0), & x < 0.5, \\ (1700, & 1000, & 0, & 5\times 10^{10}, & 10^{-6}, & 0), & x \geq 0.5. \end{cases}$$

We set the computational domain as $[0, 1]$, the final time $T = 12$ μs and $N = 200$. The problem simulates the interaction of the product gases of the explosive TNT. In this test, we choose the JWL EOS to model the explosive TNT. The relevant material-dependent quantities are given in Table 1. Fig. 2 shows the result. From the figures, we can see that all schemes can give good resolutions to the exact solution. Slight overshoots around $x = 0.4$ are observed in the velocity plot. However, these can be mitigated by refining the mesh size.

Example 5. We solve following Riemann problem:

$$(\rho_1, \rho_2, u, p, z_1, e_0) = \begin{cases} (4000, & 2785, & 0, & 7.93\times 10^{9}, & 1-10^{-6}, & 0), & x < 0.5, \\ (4000, & 2785, & -2000, & 0, & 10^{-6}, & 0), & x \geq 0.5. \end{cases}$$

which is used to model an aluminum slab on the left is hit by a traveling aluminum slab with speed $u = -2000$ m/s on the right. We set the computational domain as $[0, 1]$, the final time $T = 50$ μs and $N = 200$. The aluminum is modeled by the shock wave EOS. The relevant material-dependent quantities are given in Table 1. Fig. 3 shows the result. From the figures, we can observe that the oscillations appear along $x = 0.7$ in the plots of density, velocity, and pressure with the HO2 method. Both the HO1 method and the LO method can obtain better results and match the exact solution well.

Example 6. We solve following Riemann problem:

$$(\rho_1, \rho_2, u, p, z_1, e_0) = \begin{cases} (8900, & 1840, & 1500, & 101325, & 1-10^{-6}, & 117900), & x < 0.5, \\ (8900, & 1840, & 0, & 101325, & 10^{-6}, & 326100), & x \geq 0.5. \end{cases}$$

which is used to model a rightward traveling copper plate with speed $u = 1500$ m/s interacts with a solid explosive. We set the computational domain as $[0, 1]$, with final time $T = 85$ μs and $N = 200$. We use the same CC EOS to model both copper and solid explosives. The relevant material-dependent quantities are given in Table 1. Fig. 4 shows the result. From the figures, we can see that all of the three methods can obtain good performance.

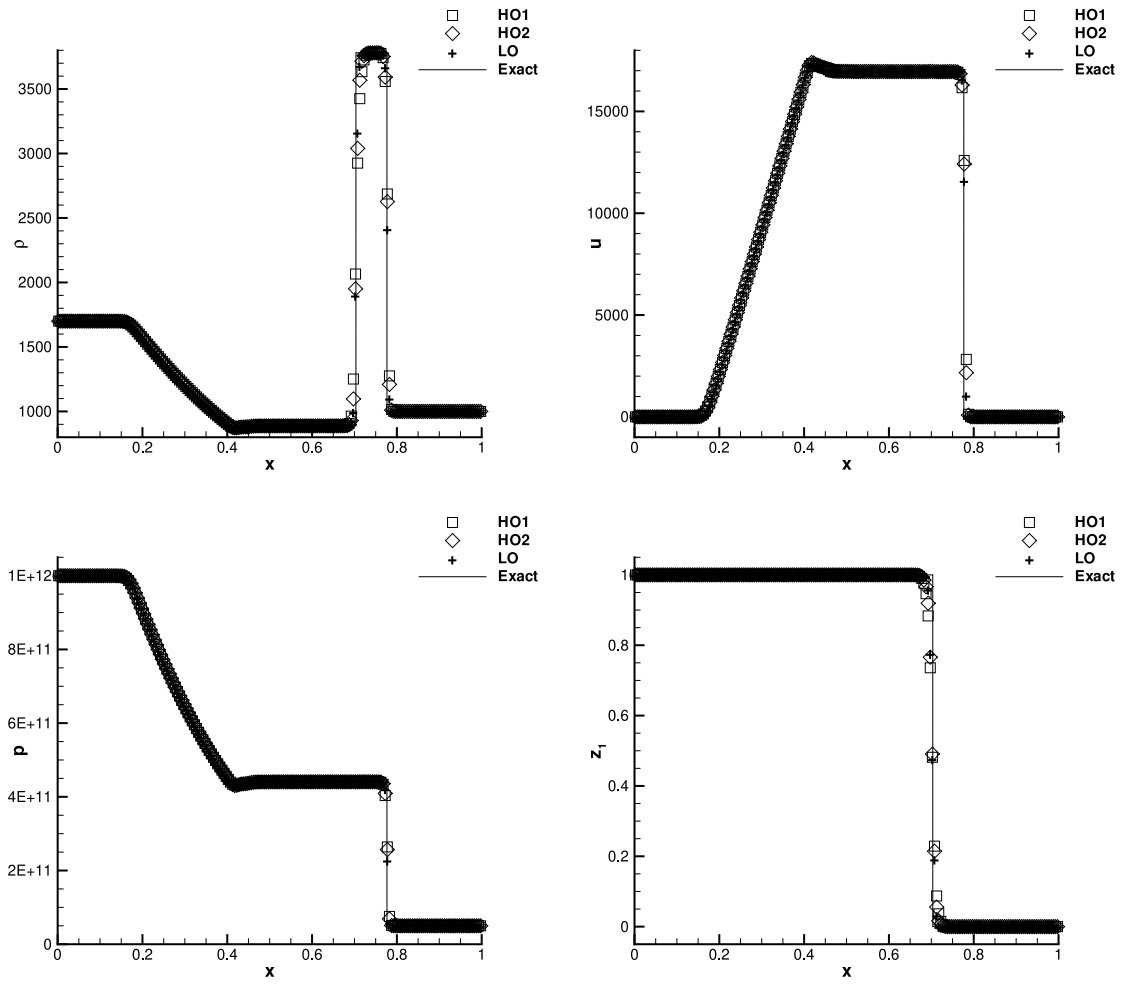

Fig. 2. Numerical results for Example 4. From left top to right bottom: figures of density, velocity, pressure, and volume fraction of fluid 1.

Example 7. We solve following Riemann problem:

$$(\rho_1, \rho_2, u, p, z_1, e_0) = \begin{cases} (2485.37, & 8900, & 0, & 3\times 10^{10}, & 1-10^{-6}, & 8149.158\times 10^3), & x < 0.5, \\ (2485.37, & 8900, & 0, & 10^5, & 10^{-6}, & 117.9\times 10^3), & x \geq 0.5. \end{cases}$$

which is used to model the interaction between gaseous detonation products and a copper plate. We set the computational domain as $[0, 1]$, with final time $T = 73$ μs and $N = 200$. In this test, we use CC EOS to model copper, and JWL EOS to model gaseous explosive. The relevant material-dependent quantities are given in Table 1. Fig. 5 shows the result. From the figures, we can see that the HO1 scheme captures the solution well. As to the LO scheme, there are slight deviations in the velocity and pressure plot, although they do not affect the resolution too much.

Example 8. We solve following Riemann problem:

$$(\rho_1, \rho_2, u, p, z_1, e_0) = \begin{cases} (11042, & 2260, & 543, & 3\times 10^{10}, & 1-10^{-6}, & 0), & x < 0.4, \\ (9961, & 2260, & 0, & 0, & 1-10^{-6}, & 0), & 0.4 \leq x < 0.6, \\ (9961, & 2260, & 0, & 0, & 10^{-6}, & 0), & x \geq 0.6, \end{cases}$$

which is used to model a rightward traveling Mach 1.163 shock wave in molybdenum interacts with MORB. We set the computational domain as $[0, 1]$, the final time $T = 85$ μs and $N = 200$. We use the same shock wave EOS to model molybdenum and MORB. The relevant material-dependent quantities are given in Table 1. Fig. 6 shows the result. From the figures, we can see that all schemes can capture the exact solution well.

Example 9. We consider a test in which a right-moving planar shock wave $Mach = 1.163$ in molybdenum initially located at $x = 0.3$ is interacting with a region $[0.4, 0.7] \times [0, 0.5]$ of MORB liquid in a square domain $[0, 1] \times [0, 1]$. This problem has been studied in [8,11,31]. The schematic for this problem is given in Fig. 7. In this problem, we use the shock wave EOS to model the MORB and molybdenum. The typical set of material

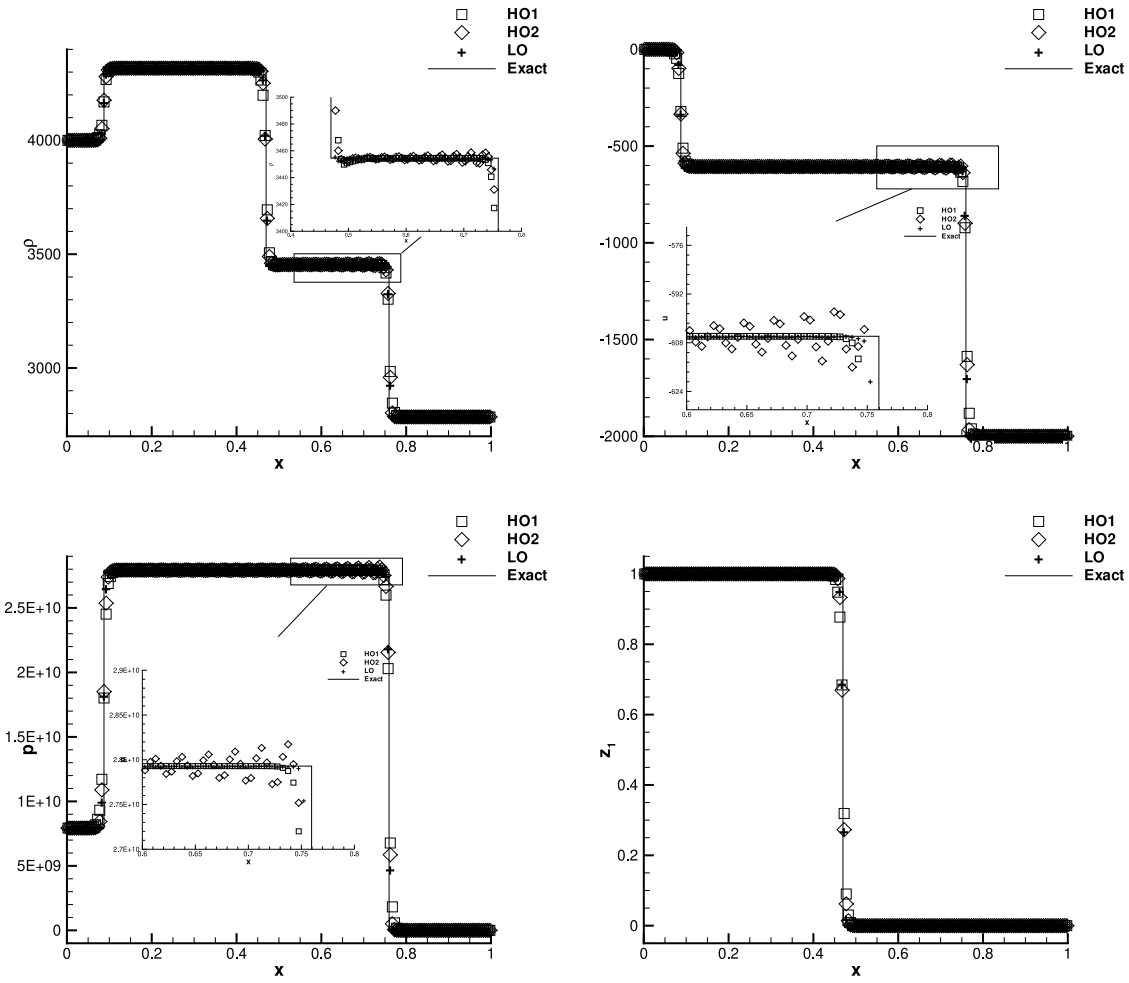

Fig. 3. Numerical results for Example 5. From left top to right bottom: figures of density, velocity, pressure, and volume fraction of fluid 1.

quantities for the MORB and molybdenum is shown in Table 1. The initial conditions and parameters required are:

$$(\rho_1, \rho_2, u, v, p, z_1, e_0) =$$

$$\begin{cases} (11042, & 2260, & 543, & 0, & 3\times 10^{10}, & 1-10^{-6}, & 0), & \text{Post-Shock Molybdenum,} \\ (9961, & 2260, & 0, & 0, & 0, & 1-10^{-6}, & 0), & \text{Pre-Shock Molybdenum,} \\ (9961, & 2260, & 0, & 0, & 0, & 10^{-6}, & 0), & \text{MORB.} \end{cases}$$

The non-reflecting boundary conditions are used. Fig. 8 shows high-resolution results for schlieren-type images of the density and pressure at time 50 μs and 100 μs using 400 × 400 cells. From the figures, we can see that the diffraction of a shock wave by MORB liquid is well captured with our method.

Example 10. We consider a test in which a left-moving copper plate located at $x \geq 0.6$ interacts with a region $[0, 0.6]\times[0, 0.5]$ of water and a region $[0, 0.6] \times [0.5, 1]$ of inert explosive in a square domain $[0, 1] \times [0, 1]$. This problem has been studied in [8]. The schematic for this problem is given in Fig. 9. In this problem, we use the CC EOS to model the copper and explosive, and the JWL EOS to model the water. The typical set of material quantities is shown in Table 1.

The initial conditions and parameters required are:

$$(\rho_1, \rho_2, \rho_3, u, v, p, z_1, z_3, e_0) =$$

$$\begin{cases} (1840, & 1004, & 8900, & 0, & 0, & 101325, & 1-2\times 10^{-6}, & 10^{-6}, & 326100), & \text{TNT,} \\ (1840, & 1004, & 8900, & 0, & 0, & 101325, & 10^{-6}, & 10^{-6}, & 25150), & \text{Water,} \\ (1840, & 1004, & 8900, & -1500, & 0, & 101325, & 10^{-6}, & 1-2\times 10^{-6}, & 117900), & \text{Copper.} \end{cases}$$

The non-reflecting boundary conditions are used. Fig. 10 shows high-resolution results for schlieren-type images of the density and pressure at time 50 μs and 100 μs using 200 × 200 cells. From the figure, we can see the transmitted and reflected shock waves due to the impacting of the copper to the water and explosive. Moreover, a reflected circular wave is observed from the corner of the three materials.

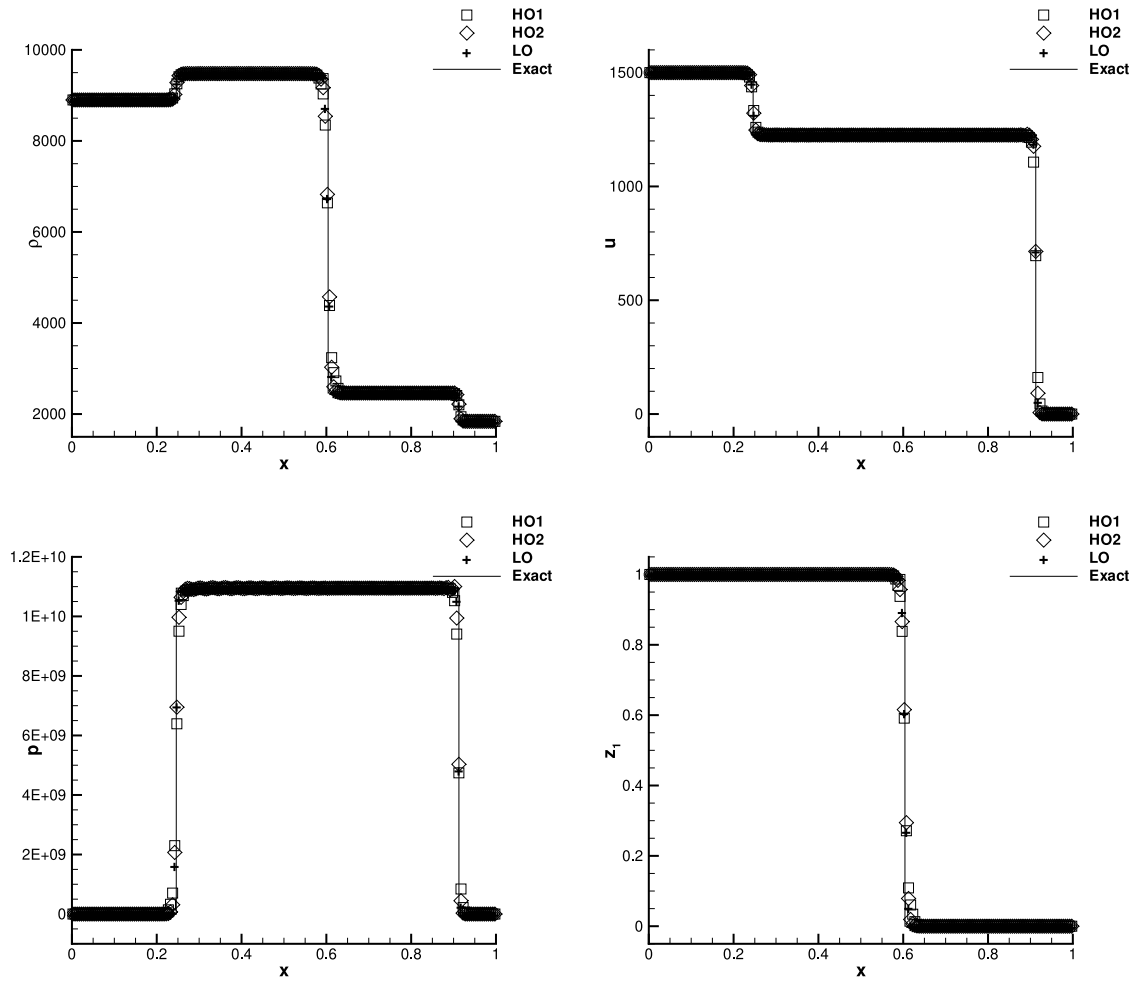

Fig. 4. Numerical results for Example 6. From left top to right bottom: figures of density, velocity, pressure, and volume fraction of fluid 1.

Example 11. Finally, we are interested in an impact problem involving the interaction of underwater, aluminum, and copper within a square domain $[0,1]\times[0,1]$. This problem has been studied in [8]. The schematic for this problem is given in Fig. 11. On the left half of the domain, the material is copper, while on the right half, the water on the top and aluminum on the bottom are separated by a horizontal interface $y = 0.4$. In this problem, all materials are modeled by shock wave EOS. The typical set of material quantities is shown in Table 1. The initial conditions and parameters required are:

$$(\rho_1,\rho_2,\rho_3,u,v,p,z_1,z_3,e_0) =$$
$$\begin{cases} (2785,\ 1000,\ 8924,\ -1500,\ 0,\ 0,\ 1-2\times10^{-6},\ 10^{-6},\ 0), & \text{Aluminum},\\ (2785,\ 1000,\ 8924,\ 0,\ 0,\ 0,\ 10^{-6},\ 10^{-6},\ 0), & \text{Water},\\ (2785,\ 1000,\ 8924,\ 0,\ 0,\ 0,\ 10^{-6},\ 1-2\times10^{-6},\ 0), & \text{Copper}. \end{cases}$$

The non-reflecting boundary conditions are used. Fig. 12 shows high-resolution results for schlieren-type images of the density and pressure at times 50 μs, 100 μs and 150 μs using 400 × 400 cells. Due to the impact of aluminum on copper, the transmitted and reflected shock waves are generated, and a circular shock wave propagates to the water. Moreover, a mushroom-like shape is produced at the corner of the three materials, and becomes larger as the process evolves. From the figures, we can see that our schemes can obtain high-resolution solutions.

4. Conclusion

In this paper, a high-order finite volume method with Mie–Grüneisen EOS is constructed for solving multicomponent fluid problems. Our scheme can maintain the equilibrium of the pressure and velocity for the interface-only problem. Furthermore, our scheme achieves high-order accuracy in the smooth region and high resolution when discontinuities appear. Extensive numerical tests have been performed to verify the scheme's high resolution and high accuracy.

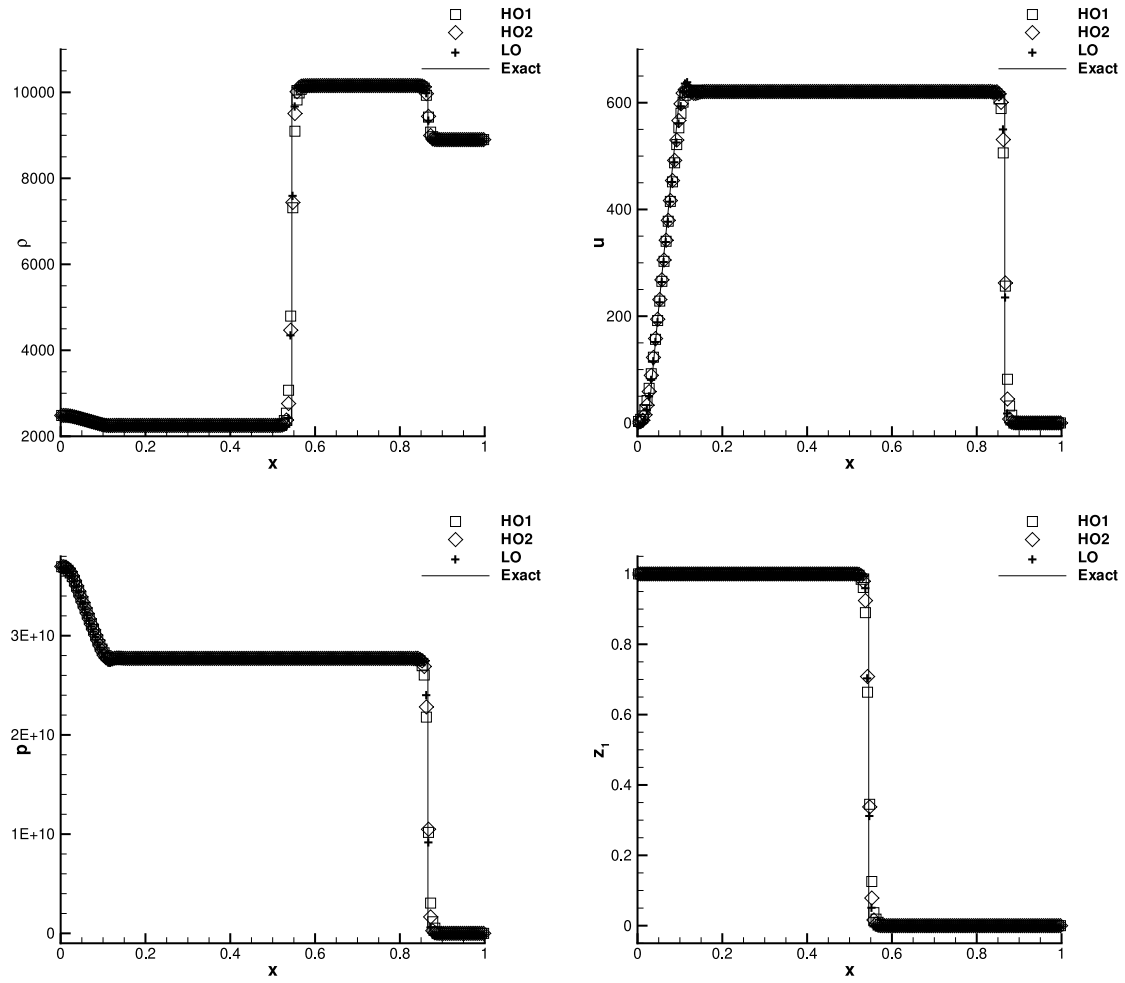

Fig. 5. Numerical results for Example 7. From left top to right bottom: figures of density, velocity, pressure, and volume fraction of fluid 1.

CRediT authorship contribution statement

Feng Zheng: Writing – original draft, Software, Methodology, Conceptualization. **Jianxian Qiu:** Writing – original draft, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Feng Zheng reports financial support was provided by National Natural Sciences Foundation of China; Jianxian Qiu reports financial support was provided by National Key R&D Program of China and National Natural Sciences Foundation of China. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The first author would like to thank Prof. T. Liu, Prof. K.-M. Shyue, Dr. D. Luo and Dr. Z. He for their invaluable assistance.

Appendix. Proof of the proposition

Proof. We assume the conservative variables $U_j = (z_1\rho_{1j}, z_2\rho_{2j}, m_j, E_j, z_{1j})$ maintains the equilibrium of the pressure and velocity. Then, we have the following equality:

$$m_j = \rho_j \bar{u}, \quad E_j = \frac{1}{2}\rho_j \bar{u}^2 + \bar{p}\mathcal{A}_{0j} + \mathcal{B}_{0j},$$

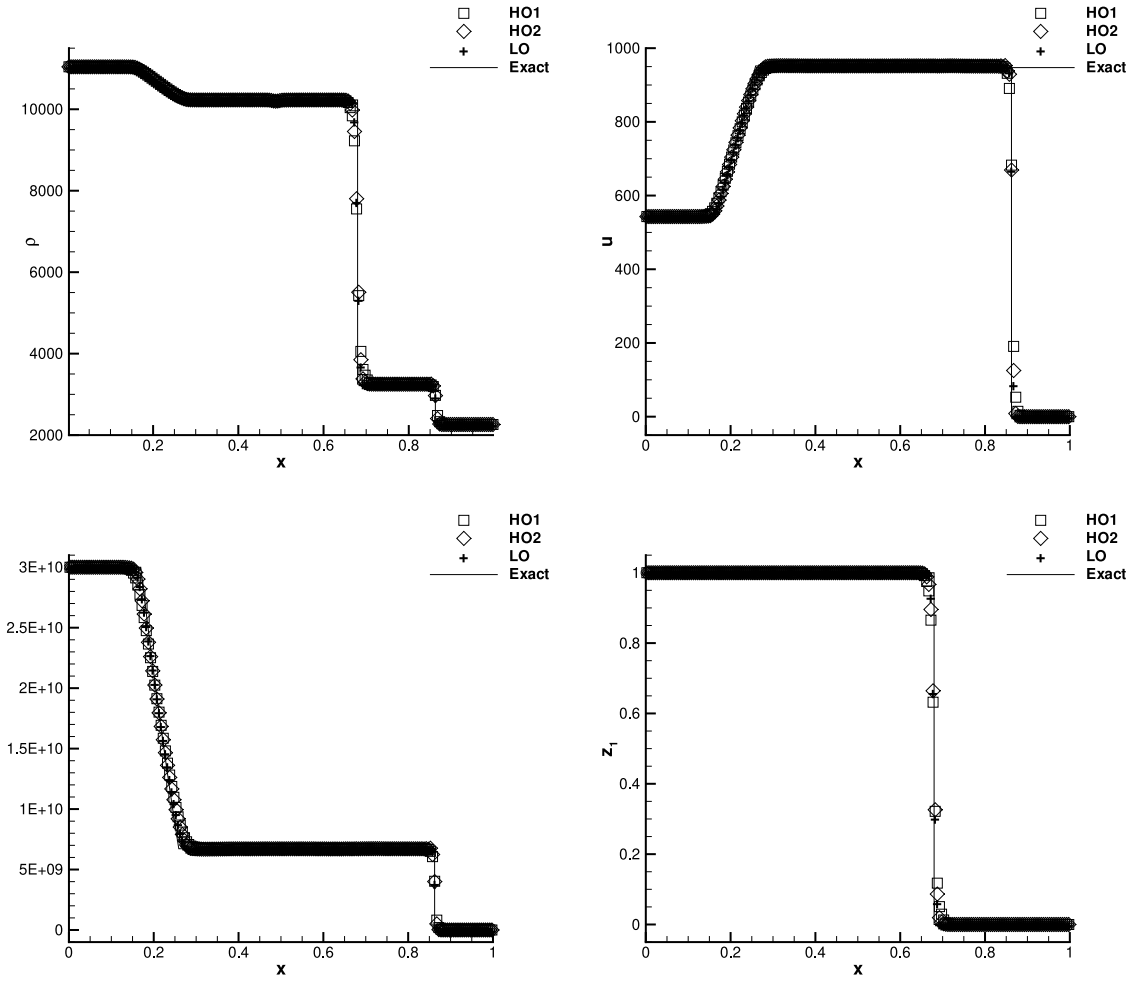

Fig. 6. Numerical results for Example 8. From left top to right bottom: figures of density, velocity, pressure, and volume fraction of fluid 1.

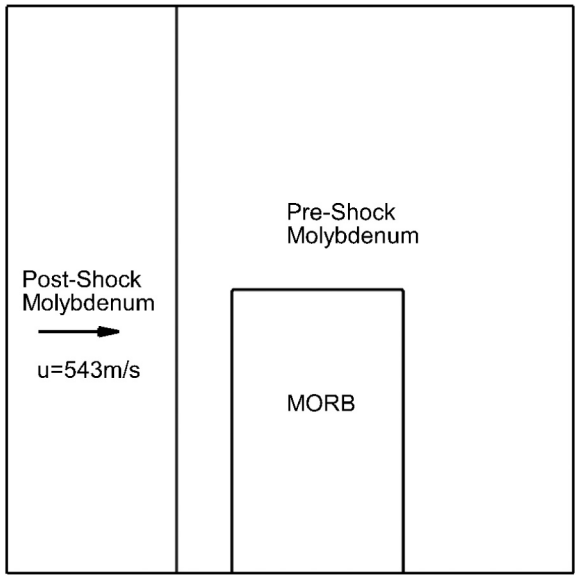

Fig. 7. Schematic for Example 9.

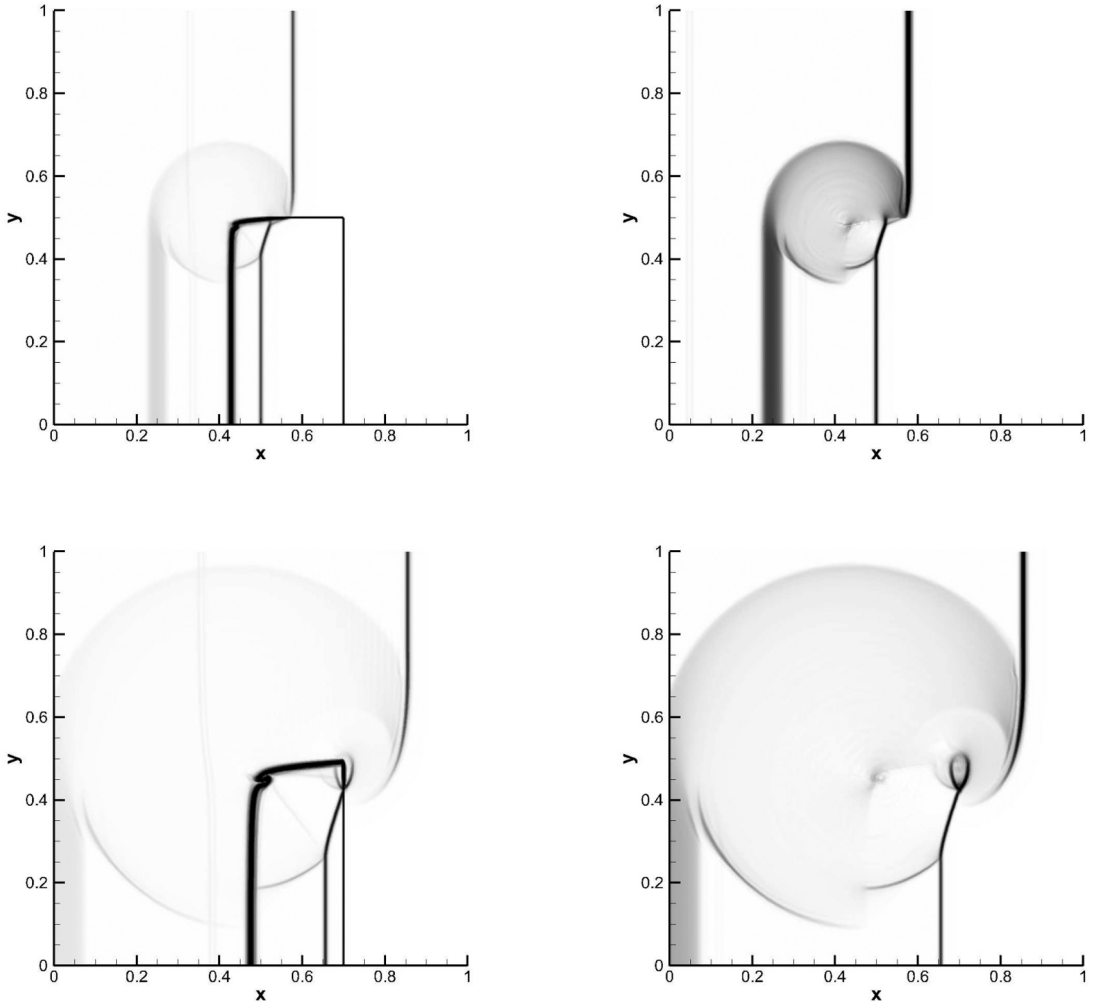

Fig. 8. Numerical schlieren images for Example 9. Left, numerical schlieren images for density; Right, numerical schlieren images for pressure; From top to bottom: t = 50 μs and 100 μs.

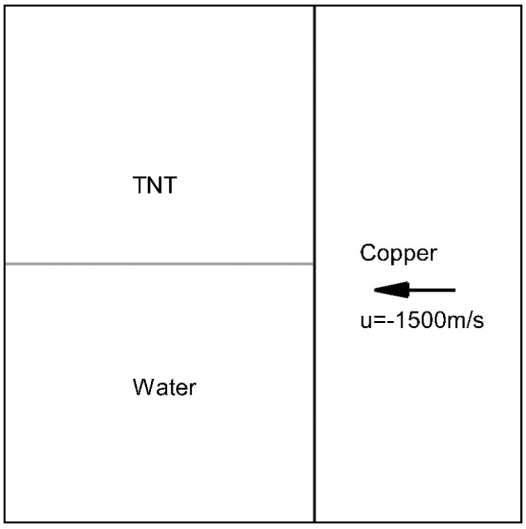

Fig. 9. Schematic for Example 10.

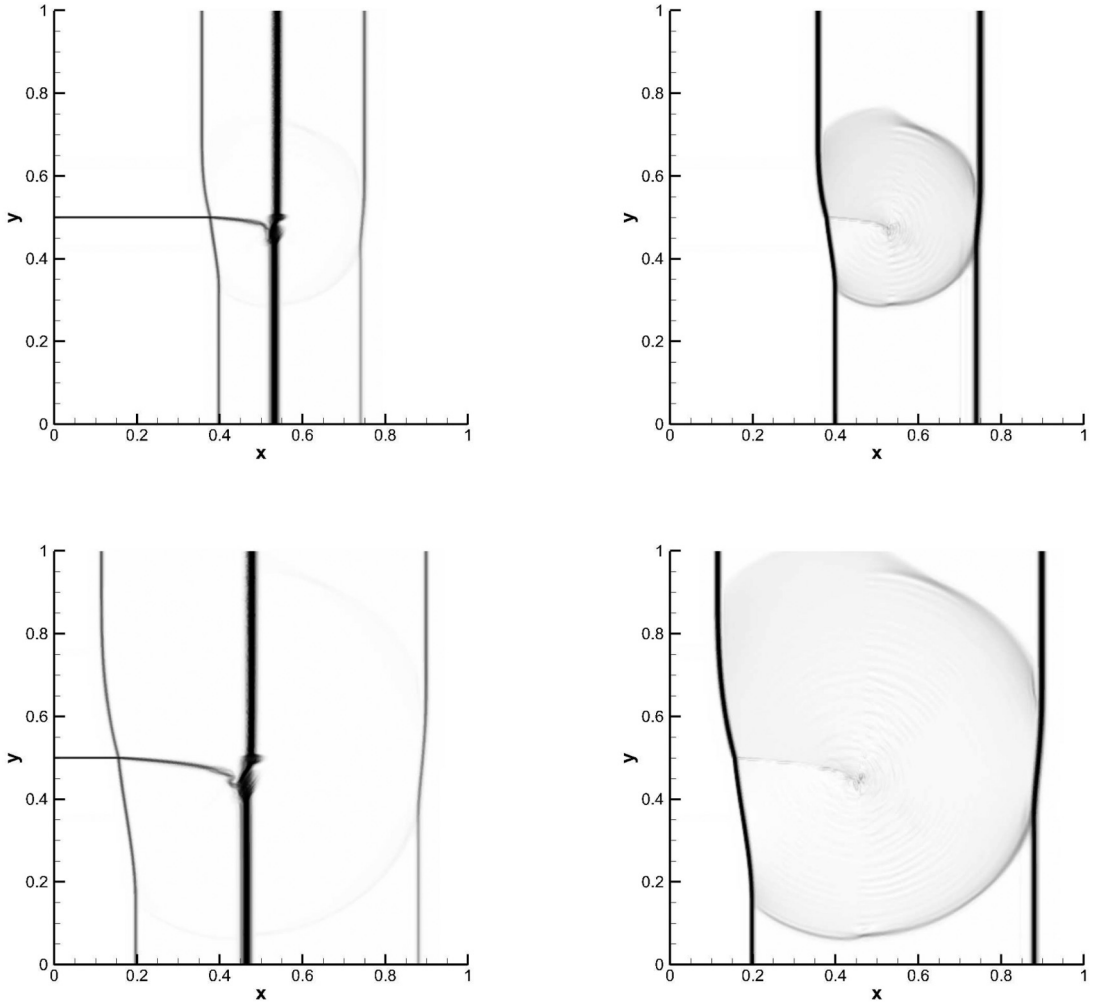

Fig. 10. Numerical schlieren images for Example 10. Left, numerical schlieren images for density; Right, numerical schlieren images for pressure; From top to bottom: t = 50 μs and 100 μs.

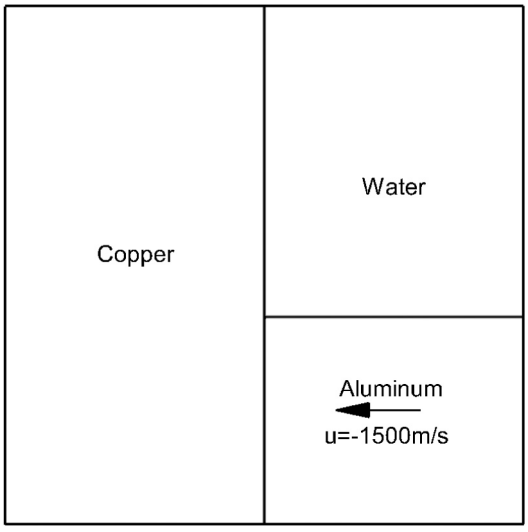

Fig. 11. Schematic for Example 11.

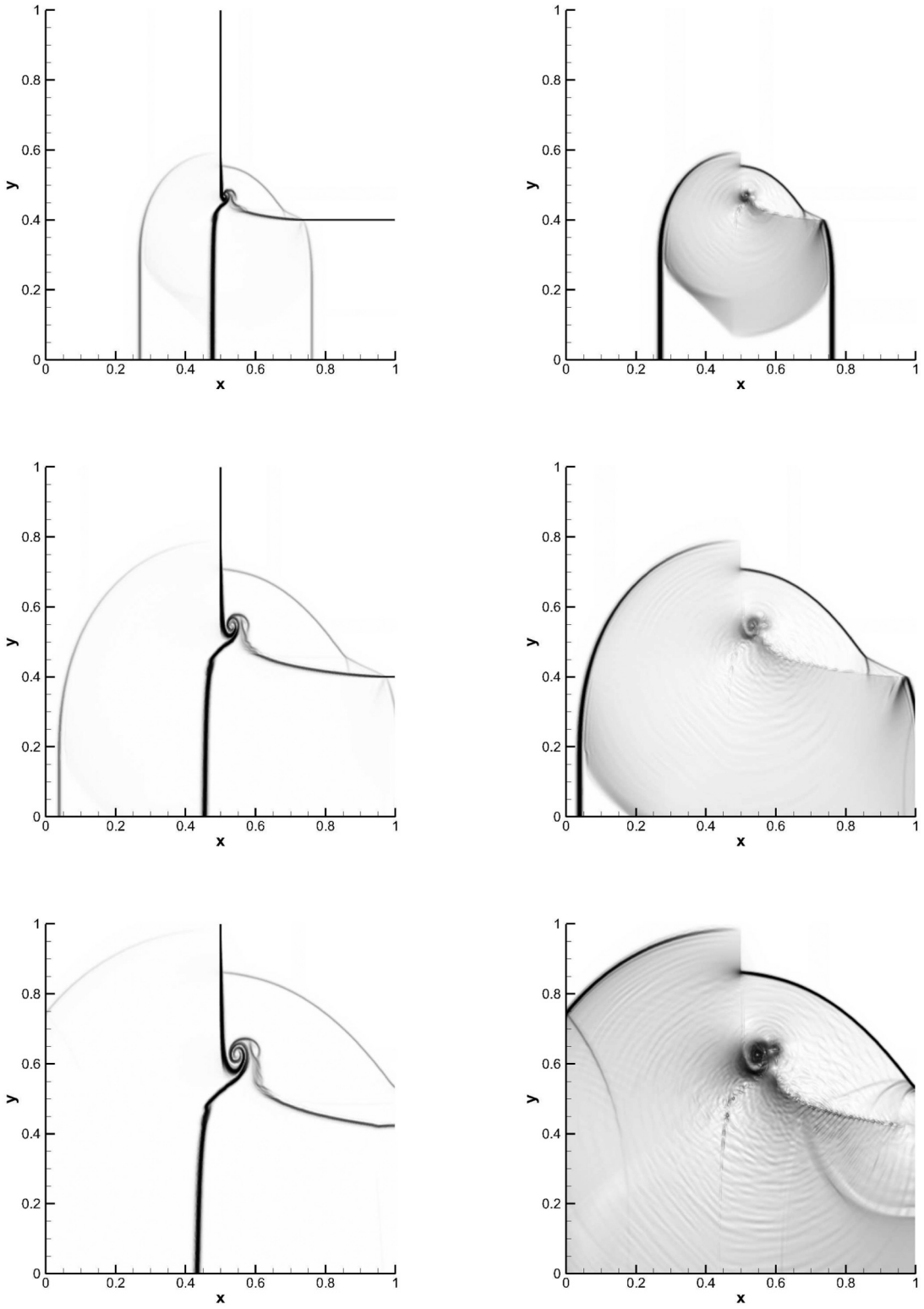

Fig. 12. Numerical schlieren images for Example 11. Left, numerical schlieren images for density; Right, numerical schlieren images for pressure; From top to bottom: t = 50 μs, 100 μs, and 150 μs.

where the $\bar{u}$ and $\bar{p}$ above denote the equilibrium values of the velocity and pressure. $\mathcal{A}_0 = z_1 \frac{1}{\gamma_1 - 1} + (1 - z_1) \frac{1}{\gamma_2 - 1}$ and $\mathcal{B}_0 = z_1 \frac{\gamma_1 B_1}{\gamma_1 - 1} + (1 - z_1) \frac{\gamma_2 B_2}{\gamma_2 - 1}$.

Now, we begin to prove the proposition.

Firstly, we demonstrate that the primitive variables reconstruction procedure from given conservative variables do not destroy the equilibrium property.

From Eq. (2.12), we can see that on cell I_j we have

$$P_m(x) = vP_{0,m}(x) + (1-v)P_{2,m}(x), \quad x \in [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}], \quad m = z_1\rho_1, z_2\rho_2, \rho u, E, z_1.$$

Here, we can write out the explicit expression of $p_{0,m}(x)$:

$$\begin{aligned} P_{0,m}(\xi) = &(3/640 - 5/48\xi - 1/16\xi^2 + 1/12\xi^3 + 1/24\xi^4)m_{j+2} \\ &+ (-29/480 + 17/24\xi + 3/4\xi^2 - 1/6\xi^3 - 1/6\xi^4)m_{j+1} \\ &+ (1067/960 - 11/8\xi^2 + 1/4\xi^4)m_j \\ &+ (-29/480 - 17/24\xi + 3/4\xi^2 + 1/6\xi^3 - 1/6\xi^4)m_{j-1} \\ &+ (3/640 + 5/48\xi - 1/16\xi^2 - 1/12\xi^3 + 1/24\xi^4)m_{j-2}, \end{aligned}$$

where $\xi = \frac{x-x_j}{\Delta x}$. we can see that the $p_{0,m}(x)$ is the linear expression of $m_{j-2}, m_{j-1}, m_j, m_{j+1}, m_{j+2}$ and satisfies linearity:

$$P_{0,C_1m^{(1)}+C_2m^{(2)}}(x) = C_1P_{0,m^{(1)}}(x) + C_2P_{0,m^{(2)}}(x).$$

Therefore, we have

$$\begin{aligned} P_u(x) &= \frac{P_{\rho u}(x)}{P_{z_1\rho_1}(x) + P_{z_2\rho_2}(x)} = \frac{\overline{u}P_{z_1\rho_1+z_2\rho_2}(x)}{P_{z_1\rho_1}(x) + P_{z_2\rho_2}(x)} = \overline{u}, \\ P_p(x) &= \frac{P_E(x) - \frac{P_{\rho u}^2(x)}{2P_\rho(x)} - \mathcal{B}_0}{\mathcal{A}_0} = \frac{\frac{1}{2}\overline{u}^2P_\rho(x) + \mathcal{A}_0\overline{p} + \mathcal{B}_0 - \frac{1}{2}\overline{u}^2P_\rho(x) - \mathcal{B}_0}{\mathcal{A}_0} = \overline{p}, \end{aligned} \tag{A.1}$$

where $\mathcal{A}_0 = P_{z_1}(x)\frac{1}{\gamma_1-1} + \left(1 - P_{z_1}(x)\right)\frac{1}{\gamma_2-1}$ and $\mathcal{B}_0 = P_{z_1}(x)\frac{\gamma_1 B_1}{\gamma_1-1} + \left(1 - P_{z_1}(x)\right)\frac{\gamma_2 B_2}{\gamma_2-1}$.

Then, we demonstrate that the implementation of the WENO method in characteristic-wise way with the primitive variables do not destroy the equilibrium.

We assume the primitive variables $W_j = (\rho_{1j}, \rho_{2j}, \overline{u}, \overline{p}, z_{1j})^T$ maintains the equilibrium of the pressure and velocity, where the $\overline{u}$ and $\overline{p}$ above denote the equilibrium values of the velocity and pressure.

We compute the right and left eigenvector matrices at interface $x = x_{j+\frac{1}{2}}$ of the cells:

$$R = \begin{pmatrix} \frac{\widetilde{\rho_1}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} & 0 & 0 & 1 & \frac{\widetilde{\rho_1}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} \\ \frac{\widetilde{\rho_2}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} & 0 & 1 & 0 & \frac{\widetilde{\rho_2}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} \\ \frac{1}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}} & 0 & 0 & 0 & -\frac{1}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}} \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, L = \begin{pmatrix} 0 & 0 & \frac{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -\frac{\widetilde{\rho_2}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} & 0 \\ 1 & 0 & 0 & -\frac{\widetilde{\rho_1}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} & 0 \\ 0 & 0 & -\frac{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}}{2} & \frac{1}{2} & 0 \end{pmatrix},$$

where we take $\widetilde{c} = \{c\}_{j+1/2}$, $\widetilde{z_1} = \{z_1\}_{j+1/2}$, $\widetilde{\rho_1} = \{z_1\rho_1\}_{j+1/2}$, $\widetilde{\rho_2} = \{z_2\rho_2\}_{j+1/2}$, $\widetilde{z_1\rho_1} = \widetilde{z_1}\widetilde{\rho_1}$, $\widetilde{z_2\rho_2} = \left(1-\widetilde{z_1}\right)\widetilde{\rho_2}$. Here, we choose $\{q\}_{j+1/2} = \left(q_j + q_{j+1}\right)/2$. We project primitive variables $V_j = (\rho_{1j}, \rho_{2j}, \overline{u}, \overline{p}, z_{1j})^T$ into the characteristic space:

$$W_j = \overline{L}U_j = \begin{pmatrix} 0 & 0 & \frac{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -\frac{\widetilde{\rho_2}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} & 0 \\ 1 & 0 & 0 & -\frac{\widetilde{\rho_1}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} & 0 \\ 0 & 0 & -\frac{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}}{2} & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} \rho_{1j} \\ \rho_{2j} \\ \overline{u} \\ \overline{p} \\ z_{1j} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\overline{u}\left(\widetilde{z_1\rho_1} + \widetilde{z_2\rho_2}\right)\widetilde{c} + \frac{\overline{p}}{2} \\ z_{1j} \\ \rho_{2j} - \frac{\widetilde{\rho_2}\overline{p}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} \\ \rho_{1j} - \frac{\widetilde{\rho_1}\overline{p}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\widetilde{c}^2} \\ -\frac{1}{2}\overline{u}\left(\widetilde{z_1\rho_1} + \widetilde{z_2\rho_2}\right)\widetilde{c} + \frac{\overline{p}}{2} \end{pmatrix},$$

which means the first and the last component in W_j are the constants during the characteristic projection. After the WENO reconstruction, we denote the obtained variables as

$$W^{\pm}_{j+\frac{1}{2}} = \begin{pmatrix} \frac{1}{2}\overline{u}\left(\widetilde{z_1\rho_1} + \widetilde{z_2\rho_2}\right)\widetilde{c} + \frac{\overline{p}}{2} \\ w_2 \\ w_3 \\ w_4 \\ -\frac{1}{2}\overline{u}\left(\widetilde{z_1\rho_1} + \widetilde{z_2\rho_2}\right)\widetilde{c} + \frac{\overline{p}}{2} \end{pmatrix}^{\pm}_{j+\frac{1}{2}}.$$

Then, we project the obtained variables back into the physical space. We have

$$V^{\pm}_{j+\frac{1}{2}} = \overline{R} W^{\pm}_{j+\frac{1}{2}} = \begin{pmatrix} \frac{\widetilde{\rho_1}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}^2} & 0 & 0 & 1 & \frac{\widetilde{\rho_1}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}^2} \\ \frac{\widetilde{\rho_2}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}^2} & 0 & 1 & 0 & \frac{\widetilde{\rho_2}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}^2} \\ \frac{1}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}} & 0 & 0 & 0 & -\frac{1}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}} \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2}\overline{u}\left(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2}\right)\tilde{c}+\frac{\overline{p}}{2} \\ w_2 \\ w_3 \\ w_4 \\ -\frac{1}{2}\overline{u}\left(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2}\right)\tilde{c}+\frac{\overline{p}}{2} \end{pmatrix}^{\pm}_{j+\frac{1}{2}}$$

$$= \begin{pmatrix} w_4 + \frac{\widetilde{\rho_1}\overline{p}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}^2} \\ w_3 + \frac{\widetilde{\rho_2}\overline{p}}{(\widetilde{z_1\rho_1}+\widetilde{z_2\rho_2})\tilde{c}^2} \\ \overline{u} \\ \overline{p} \\ w_2 \end{pmatrix}^{\pm}_{j+\frac{1}{2}},$$

which means we can maintain the equilibrium during the reconstruction.

Finally, we demonstrate that the scheme in (2.4) maintains the equilibrium of velocity and pressure.

The variables $V^{\pm}_{j+\frac{1}{2}}$ obtained in characteristic projection are used to compute the HLLC flux $\hat{F}_{j+\frac{1}{2}}$. Due to the equilibrium of the pressure and velocity, we have $u_L = u_R = \overline{u}$, $p_L = p_R = \overline{p}$, $\hat{u}_{j\pm\frac{1}{2}} = \overline{u}$. Therefore, we have

$$s_* = \overline{u}, \quad U_{*k} = U_k,$$

and

$$\hat{F}_{j+\frac{1}{2}} = \frac{1+\mathrm{sgn}(s^*)}{2}F_L + \frac{1-\mathrm{sgn}(s^*)}{2}F_R = \begin{pmatrix} \widetilde{z_1\rho_1}\overline{u} \\ \widetilde{z_2\rho_2}\overline{u} \\ \widetilde{\rho u}^2 + \overline{p} \\ \overline{u}\left(\frac{1}{2}\widetilde{\rho u}^2 + \mathcal{A}_0\overline{p} + \mathcal{B}_0 + \overline{p}\right) \\ \overline{u}\widetilde{z_1} \end{pmatrix},$$

where $\tilde{q} = \frac{1+\mathrm{sgn}(s^*)}{2}q_L + \frac{1-\mathrm{sgn}(s^*)}{2}q_R$, $\quad q = z_1\rho_1, z_2\rho_2, \rho, z_1$, $\mathcal{A}_0 = \widetilde{z_1}\frac{1}{\gamma_1-1} + \left(1-\widetilde{z_1}\right)\frac{1}{\gamma_2-1}$ and $\mathcal{B}_0 = \widetilde{z_1}\frac{\gamma_1 B_1}{\gamma_1-1} + \left(1-\widetilde{z_1}\right)\frac{\gamma_2 B_2}{\gamma_2-1}$.

As to the source term in (2.4), due to the equilibrium of the velocity and the fact that the four point Gauss–Lobatto quadrature rule is exact for the polynomial degree up to five, we have

$$\begin{aligned} &\left(S(U^-_{j+\frac{1}{2}})\hat{u}_{j+\frac{1}{2}} - S(U^+_{j-\frac{1}{2}})\hat{u}_{j-\frac{1}{2}} - \Delta x\sum_k w_k S(U)_{xG_k} u_{G_k}\right) \\ =&\left(S(U^-_{j+\frac{1}{2}})\overline{u} - S(U^+_{j-\frac{1}{2}})\overline{u} - \Delta x\overline{u}\sum_k w_k S(U)_{xG_k}\right) \\ =&\left(S(U^-_{j+\frac{1}{2}})\overline{u} - S(U^+_{j-\frac{1}{2}})\overline{u} - \overline{u}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} S(U)_x \mathrm{d}x\right) \\ =&\left((S(U^-_{j+\frac{1}{2}})\overline{u} - S(U^+_{j-\frac{1}{2}})\overline{u}) - \overline{u}(S(U^-_{j+\frac{1}{2}}) - S(U^+_{j-\frac{1}{2}}))\right) \\ =&0, \end{aligned} \tag{A.2}$$

Then, we obtain the following formula:

$$\frac{\partial}{\partial t}\begin{pmatrix} z_1\rho_1 \\ z_2\rho_2 \\ \rho u \\ E \\ z_1 \end{pmatrix}_j = -\frac{1}{\Delta x}\left(\begin{pmatrix} \widetilde{z_1\rho_1}\overline{u} \\ \widetilde{z_2\rho_2}\overline{u} \\ \widetilde{\rho u}^2 + \overline{p} \\ \overline{u}\left(\frac{1}{2}\widetilde{\rho u}^2 + \mathcal{A}_0\overline{p} + \mathcal{B}_0 + \overline{p}\right) \\ \overline{u}\widetilde{z_1} \end{pmatrix}_{j+\frac{1}{2}} - \begin{pmatrix} \widetilde{z_1\rho_1}\overline{u} \\ \widetilde{z_2\rho_2}\overline{u} \\ \widetilde{\rho u}^2 + \overline{p} \\ \overline{u}\left(\frac{1}{2}\widetilde{\rho u}^2 + \mathcal{A}_0\overline{p} + \mathcal{B}_0 + \overline{p}\right) \\ \overline{u}\widetilde{z_1} \end{pmatrix}_{j-\frac{1}{2}}\right).$$

From the equation for density of fluid 1 and 2 equations, we have

$$\frac{\partial \rho_j}{\partial t} = -\frac{1}{\Delta x}\left(\left(\widetilde{\rho u}\right)_{j+\frac{1}{2}} - \left(\widetilde{\rho u}\right)_{j-\frac{1}{2}}\right),$$

Then, we have

$$\frac{\partial(\rho u)_j}{\partial t} = -\frac{1}{\Delta x}\left(\left(\widetilde{\rho u}^2\right)_{j+\frac{1}{2}} - \left(\widetilde{\rho u}^2\right)_{j-\frac{1}{2}}\right) = -\frac{\overline{u}}{\Delta x}\left(\left(\widetilde{\rho u}\right)_{j+\frac{1}{2}} - \left(\widetilde{\rho u}\right)_{j-\frac{1}{2}}\right) = \overline{u}\frac{\partial \rho_j}{\partial t},$$

which means the velocity will maintain equilibrium when the solution evolves.

Now, we consider the equilibrium of pressure. Due to the equilibrium of velocity, we have

$$\frac{\partial(\rho u^2)_j}{\partial t} = \overline{u}^2 \frac{\partial \rho_j}{\partial t} = -\frac{\overline{u}^2}{\Delta x}\left(\left(\widetilde{\rho u}\right)_{j+\frac{1}{2}} - \left(\widetilde{\rho u}\right)_{j-\frac{1}{2}} \right).$$

By comparing energy equation with above equations, we have

$$\frac{\partial(\mathcal{A}_0 p + \mathcal{B}_0)_j}{\partial t} = -\frac{1}{\Delta x}\left((\overline{p}\mathcal{A}_{0_{j+\frac{1}{2}}}\overline{u} + \mathcal{B}_{0_{j+\frac{1}{2}}}\overline{u}) - (\overline{p}\mathcal{A}_{0_{j+\frac{1}{2}}}\overline{u} + \mathcal{B}_{0_{j+\frac{1}{2}}}\overline{u}) \right).$$

By resorting to the volume fraction equation, we can further simplify the above equation and obtain

$$\frac{\partial(z_1 p)_j}{\partial t} = \overline{p}\frac{\partial z_{1_j}}{\partial t},$$

which means the pressure will also maintain equilibrium when the solution evolves. □

References

[1] Karni S. Multicomponent flow calculations by a consistent primitive algorithm. J Comput Phys 1994;112:31–43.
[2] Abgrall R, Karni S. Computations of compressible multifluids. J Comput Phys 2001;169:594–623.
[3] Marquina A, Mulet P. A flux-split algorithm applied to conservative models for multicomponent compressible flows. J Comput Phys 2003;185:120–38.
[4] Johnsen E. Spurious oscillations and conservation errors in interface-capturing schemes. Cent Turbulence Res Annu Res Briefs 2008.
[5] Abgrall R. How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. J Comput Phys 1996;125:150–60.
[6] Shyue K-M. An efficient shock-capturing algorithm for compressible multicomponent problems. J Comput Phys 1998;142:208–42.
[7] Shyue K-M. A fluid-mixture type algorithm for compressible multicomponent flow with van der Waals equation of state. J Comput Phys 1999;156:43–88.
[8] Shyue K-M. A fluid-mixture type algorithm for compressible multicomponent flow with Mie-Grüneisen equation of state. J Comput Phys 2001;171:678–707.
[9] Allaire G, Clerc S, Kokh S. A five-equation model for the simulation of interfaces between compressible fluids. J Comput Phys 2002;181:577–616.
[10] Johnsen E, Colonius T. Implementation of WENO schemes in compressible multicomponent flow problems. J Comput Phys 2006;219:715–32.
[11] He Z, Tian B, Zhang Y, Gao F. Characteristic-based and interface-sharpening algorithm for high-order simulations of immiscible compressible multi-material flows. J Comput Phys 2017;333:247–68.
[12] Wang B, Xiang G, Hu X. An incremental-stencil WENO reconstruction for simulation of compressible two-phase flows. Int J Multiph Flow 2018;104:20–31.
[13] Wang Q, Deiterding R, Pan J, Ren Y-X. Consistent high resolution interface-capturing finite volume method for compressible multi-material flows. Comput Fluids 2020;202:104518.
[14] Coralic V, Colonius T. Finite-volume WENO scheme for viscous compressible multicomponent flows. J Comput Phys 2014;274:95–121.
[15] Gu Y, Gao Z, Hu G, Li P, Wang L. High order finite difference alternative WENO scheme for multi-component flows. J Sci Comput 2021;89:52.
[16] Gu Y, Gao Z, Hu G, Li P, Wang L. A robust high order alternative WENO scheme for the five-equation model. J Sci Comput 2021;88:12.
[17] Boe D, Shahbazi K. A positivity preserving high-order finite difference method for compressible two-fluid flows. Numer Methods Partial Differ Equ 2023;39:4087–125.
[18] Shahbazi K. High-order finite difference scheme for compressible multi-component flow computations. Comput Fluids 2019;190:425–39.
[19] Nonomura T, Fujii K. Characteristic finite-difference WENO scheme for multicomponent compressible fluid analysis: Overestimated quasi-conservative formulation maintaining equilibriums of velocity, pressure, and temperature. J Comput Phys 2017;340:358–88.
[20] Qiu J, Liu T, Khoo BC. Runge-kutta discontinuous Galerkin methods for compressible two-medium flow simulations: one-dimensional case. J Comput Phys 2007;222:353–73.
[21] Toro EF. Riemann solvers and numerical methods for fluid dynamics. Springer; 2009.
[22] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77:439–71.
[23] Zhu J, Shu C-W. A new type of multi-resolution WENO schemes with increasingly higher order of accuracy on triangular meshes. J Comput Phys 2019;392:19–33.
[24] Zhu J, Qiu J. A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. J Comput Phys 2016;318:110–21.
[25] Zhu J, Shu C-W. A new type of multi-resolution WENO schemes with increasingly higher order of accuracy. J Comput Phys 2018;375:659–83.
[26] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126:202–28.
[27] Luo D, Qiu J, Zhu J, Chen Y. A quasi-conservative discontinuous Galerkin method for multi-component flows using the non-oscillatory kinetic flux. J Sci Comput 2021;87:96.
[28] Balsara DS, Garain S, Shu C-W. An efficient class of WENO schemes with adaptive order. J Comput Phys 2016;326:780–804.
[29] Kamm JR. An exact, compressible one-dimensional Riemann solver for general, convex equations of state. U.S. Dep Energy Office Sci Tech Inf 2015.
[30] Banks JW. On exact conservation for the Euler equations with complex equations of state. Commun Comput Phys 2010;8(5):995–1015.
[31] Miller GH, Puckett EG. A high-order Godunov method for multiple condensed phases. J Comput Phys 1996;128:134–64.