An improved unstructured WENO method for compressible multi-fluids flow

Hongwei Zheng[*,†] and Wengeng Zhao

LHD, Institute of Mechanics, Chinese Academy of Sciences, No. 15 Beisihuanxi Road, Beijing, 100190, China

SUMMARY

An improved high-order accurate WENO finite volume method based on unstructured grids for compressible multi-fluids flow is proposed in this paper. The third-order accuracy WENO finite volume method based on triangle cell is used to discretize the governing equations. To have higher order of accuracy, the P1 polynomial is reconstructed firstly. After that, the P2 polynomial is reconstructed from the combination of the P1. The reconstructed coefficients are calculated by analytical form of inverse matrix rather than the numerical inversion. This greatly improved the efficiency and the robustness. Four examples are presented to examine this algorithm. Numerical results show that there is no spurious oscillation of velocity and pressure across the interface and high-order accurate result can be achieved. Copyright © 2016 John Wiley & Sons, Ltd.

1. INTRODUCTION

Compressible multi-fluid flows have many applications such as in combustion process and chemical process engineering. Hence, many methods [1–7] have been developed to model and simulate them. Among them, the front capturing method is very promising for its efficiency and simplicity when dealing with different fluids. However, most of these existing methods are of first-order or second-order accuracy. The main deficiency of these low-order numerical methods is excessive numerical dissipation. The tiny structures dissipate prematurely, and the interface is smeared because of the numerical dissipation. One way to improve the resolution is to use high-order numerical methods to obtain accurate result in relatively coarse grids.

In fact, there are many high-order numerical methods [8, 9] developed in the past decades: the essentially non-oscillatory method (ENO) [10], the weighted ENO (WENO) method [11, 12], the discontinuous Galerkin method [13], and the spectral volume method [14]. In this article, the WENO method is employed. The key idea of WENO scheme is the use of a convex combination of all candidate stencils to achieve high-order accuracy in smooth regions and ENO transition for solution discontinuities. Johnsen [15] presented the work by using the finite difference WENO method to simulate compressible multi-fluid flows. Recently, Dumbser [16] adopted the Arbitrary high order schemes using DERivatives (ADER) WENO finite volume schemes to the Baer–Nunziato model of compressible multi-fluid flows. However, this WENO reconstruction for multi-fluid flows in literature is performed on the structured grid, and their extension to unstructured grid is not trivial [17–20]. The main reason is that the expansion of stencil, which is quite straightforward in structured grid, becomes difficult because of the complicated topology in unstructured grid. Besides, the

---

*Correspondence to: Hongwei Zheng, Institute of Mechanics, Chinese Academy of Sciences, No. 15 Beisihuanxi Road, Beijing 100190, China.

†E-mail: holyvey@gmail.com

reconstructed coefficients in the unstructured mesh are not constants as those in the structured mesh-based solver. In fact, they are related to the topology of the mesh. Hence, matrix inversion has to be employed to obtain the reconstruction coefficients. This troublesome procedure takes too much computational cost and is easy to blow up when the grid deforms greatly.

In this article, an improved WENO finite volume method based on unstructured grids for compressible multi-fluids flow is presented. The quasi-conservative form [5, 21, 25] is applied to avoid the appearance of spurious pressure oscillations at material interfaces. To have higher order of accuracy, the P1 polynomial is reconstructed firstly. After that, the P2 polynomial is reconstructed from the combination of the P1. The non-negative reconstructed coefficients are calculated by analytical form of inverse matrix rather than the numerical inversion. This greatly improved the efficiency and the robustness. Four two-dimensional examples are presented to examine this method. The outline of this paper is described as follows. The discretization of space and time for the governing equations for multi-fluid flows under unstructured grids is given in Section 2. Numerical experiments are presented in Section 3. And Section 4 is devoted to the conclusions.

2. WENO RECONSTRUCTION

In this section, the governing equations for multi-fluids flows are described. The WENO reconstruction on unstructured mesh is stated in detail here.

2.1. Governing equations

Neglecting the viscosity, heat transfer, and chemical reaction, the governing equation for compressible flow can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = 0 \tag{1}$$

where $\mathbf{u}, \mathbf{f}$, and $\mathbf{g}$ are the state vector of conservative variables, the flux vector in x direction, and the flux vector in y direction,

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (E + p) u \end{pmatrix}, \mathbf{g} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (E + p) v \end{pmatrix} \tag{2}$$

Here, ρ, u, v, E, and p are the density, velocity in x direction, velocity in y direction, total energy, and pressure. The equation of state for each fluid component is the stiffened gas equation,

$$p = (\gamma - 1) \left[E - \frac{1}{2} \rho \left(u^2 + v^2 \right) \right] - \gamma \pi \tag{3}$$

To model multi-fluid flows, equations for property of material should be included to the original Euler equations,

$$\frac{\partial \Pi}{\partial t} + \vec{u} \cdot \nabla \Pi = 0 \tag{4}$$

with the definition

$$\Pi = \begin{pmatrix} \beta \\ \theta \end{pmatrix}, \beta = \frac{1}{\gamma - 1}, \theta = \frac{\gamma \pi}{\gamma - 1} \tag{5}$$

Here, γ and π are the property of material. Equation (4) can be rewritten as another form

$$\frac{\partial \Pi}{\partial t} + \nabla \cdot \Lambda - \Pi \nabla \cdot \vec{u} = 0 \tag{6}$$

with the numerical flux,

$$\Lambda = \vec{u}\,\Pi \tag{7}$$

2.2. WENO reconstruction on unstructured mesh

In this sub-section, the discretization of Eqs. (1) and (6) by using unstructured WENO scheme is presented. Without losing generality, the governing equations are discretized in a triangle cell 0,

$$\frac{d}{dt}\mathbf{u}_0(t) + \frac{1}{\Omega_0}\oint_\Omega \mathbf{F}\cdot\mathbf{n}\,ds = 0 \tag{8}$$

with the definition

$$\mathbf{u}_0 = \frac{1}{\Omega_0}\int_{\Omega_0} u_0\,dV \tag{9}$$

where $\mathbf{F} = \mathbf{f}\mathbf{i} + \mathbf{g}\mathbf{j}$ is the numerical flux, Ω_0 represents the control volume, $\mathbf{n}$ is the unit normal vector outward to the surface, and $\mathbf{u}_0$ is the mean state value in the control volume.

The second term in the Eq. (8) is discretized using s-point Gaussian formulation with degree $2s - 1$ and order of accuracy $2s$. It can be expressed as

$$\oint_{\Omega_0} \mathbf{F}\cdot\mathbf{n}\,ds \approx |\partial\Omega_0| \sum_{j=1}^{s} \omega_j \mathbf{F}_j \cdot \mathbf{n} \tag{10}$$

where s represents the number of Gauss points in numerical integral and ω_j is the weights. For example, for the third-order WENO reconstruction, s is equal to 2.

For the third-order WENO scheme in triangle cells, a quadratic polynomial reconstruction denoted by p^2 is applied. The stencils for reconstruction are shown in Figure 1. A least square reconstruction is suggested by Hu [22]. For instance, we suppose the reconstruction has the following form:

$$\begin{aligned} p^2(x, y) &= U_0 + a\xi + b\eta + c\xi^2 + d\xi\eta + e\eta^2 \\ &= \lambda_0 U_0 + \lambda_i U_i + \ldots + \lambda_9 U_k b \end{aligned} \tag{11}$$
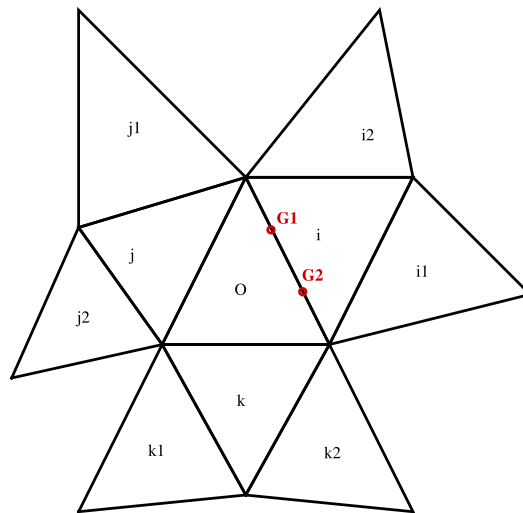

Figure 1. Stencil for weighted essentially non-oscillatory reconstruction.

where $\xi = x - x_0$, $\eta = y - y_0$ and U_0 is the average value on cell 0. For cell i, we can obtain the following formulation:

$$U_i - U_0 = a\left[\int_i \xi d\Omega - \int_0 \xi d\Omega\right] + b\left[\int_i \eta d\Omega - \int_0 \eta d\Omega\right] + c\left[\int_i \xi^2 d\Omega - \int_0 \xi^2 d\Omega\right]$$
$$+ d\left[\int_i \xi\eta d\Omega - \int_0 \xi\eta d\Omega\right] + e\left[\int_i \eta^2 d\Omega - \int_0 \eta^2 \Omega\right]$$

(12)

The same relationship can be obtained on the other eight neighbor cells. A group of Eq. (12) in all these cells can be rewritten as the following matrix form:

$$\begin{pmatrix} \int_i \xi d\Omega & \int_i \eta d\Omega & \int_i \xi^2 d\Omega & \int_i \xi\eta d\Omega & \int_i \eta^2 d\Omega \\ \int_j \xi d\Omega & \int_j \eta d\Omega & \int_j \xi^2 d\Omega & \int_j \xi\eta d\Omega & \int_j \eta^2 d\Omega \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \int_{k2} \xi d\Omega & \int_{k2} \eta d\Omega & \int_{k2} \xi^2 d\Omega & \int_{k2} \xi\eta d\Omega & \int_{k2} \eta^2 d\Omega \end{pmatrix}_{(9\times5)} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} U_i - U_0 \\ U_j - U_0 \\ U_k - U_0 \\ U_{i1} - U_0 \\ U_{i2} - U_0 \\ U_{j1} - U_0 \\ U_{j2} - U_0 \\ U_{k1} - U_0 \\ U_{k2} - U_0 \end{pmatrix}$$

(13)

There are five unknowns and nine equations in Eq. (13). It can be solved using the least square method

$$A^T A \Phi = A^T \Psi$$ (14)

with the definition

$$A = \begin{pmatrix} \int_i \xi d\Omega_i & \cdots & \int_i \eta^2 d\Omega_i \\ \vdots & \ddots & \vdots \\ \int_{k2} \xi d\Omega_{k2} & \cdots & \int_{k2} \eta^2 d\Omega_{k2} \end{pmatrix}, \Phi = \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix}, \Psi = \begin{pmatrix} U_i - U_0 \\ U_j - U_0 \\ U_k - U_0 \\ U_{i1} - U_0 \\ U_{i2} - U_0 \\ U_{j1} - U_0 \\ U_{j2} - U_0 \\ U_{k1} - U_0 \\ U_{k2} - U_0 \end{pmatrix}$$

(15)

To make the reconstruction more robust and reliable, we employ the weighted least square reconstruction in the present algorithm. The coefficient matrix is modified by multiplying a weight on each row.

$$\hat{A} = \begin{pmatrix} \alpha_i \int_i \xi d\Omega_i & \cdots & \alpha_i \int_i \eta^2 d\Omega_i \\ \alpha_j \int_j \xi d\Omega_j & \cdots & \alpha_j \int_j \eta^2 d\Omega_j \\ \vdots & \ddots & \vdots \\ \alpha_{k2} \int_{k2} \xi d\Omega_k2 & \cdots & \alpha_{k2} \int_{k2} \eta^2 d\Omega_k2 \end{pmatrix}_{(9\times5)}$$

(16)

In Eq. (16), the weights are inversely proportional to the distance from centroid of current cell to the centroid of the neighboring cell and can be written in the form

$$\alpha_i = \frac{1}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \tag{17}$$

Therefore, the reconstruction equation will be written in another form

$$\hat{A}^T \hat{A} \Phi = \hat{A}^T \hat{\Psi} \tag{18}$$

It should be noted that Ψ on the right side of Eq. (14) must be changed correspondingly. Clearly, the quadratic polynomial only depends on the geometry of the grid and has nothing to do with the solution. Hence, we can work out the coefficient and store them in advance.

Another important step in WENO reconstruction is to build the linear polynomials. As shown in Figure 1, nine different stencils are chosen here: $S_1 = \{0, i, j\}$, $S_2 = \{0, i, k\}$, $S_3 = \{0, j, k\}$, $S_4 = \{0, i, i1\}$, $S_5 = \{0, i, i2\}$, $S_6 = \{0, j, j1\}$, $S_7 = \{0, j, j2\}$, $S_8 = \{0, k, k1\}$, and $S_9 = \{0, k, k2\}$. We can obtain a reconstructed polynomial on each stencil. Just as the p^2 polynomial in Eq. (11), the p^1 polynomial can be defined as

$$\begin{aligned} p^1 &= a^1 + b^1 \xi + c^1 \eta \\ &= \sigma_0 U_0 + \sigma_1 U_i + \sigma_2 U_j \end{aligned} \tag{19}$$

For each Gauss point, taking the *G1* point on side i for example, we can find the linear weights γ_j to make the combination of all the nine polynomials satisfy

$$W(x, y) = \sum_{j=1}^{9} \gamma_i p_j(x, y) \tag{20}$$

where $W(x, y)$ is the value of p^2 on Gauss point. To make sure the reconstructed value is a convex combination of all candidate stencils, which is of vital importance to suppress oscillation when the discontinuity appears, we need to obtain a set of non-negative γ. Clearly, the coefficients γ should satisfy the following relationship:

$$\sum_{j=1}^{9} \gamma_j = 1 \tag{21}$$

Besides, for cell 0, the coefficient before U_0 in p^2 should be equal to all the coefficients before U_0 in p^1. Applying this formulation to all the 10 cells and combining Eqs. (11), (19), and (20), 10 equations to solve γ can be obtained

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & a_{1,7} & a_{1,8} & a_{1,9} \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & 0 & 0 & 0 & 0 \\ a_{3,1} & 0 & a_{3,3} & 0 & 0 & a_{3,6} & a_{3,7} & 0 & 0 \\ a_{4,1} & a_{4,2} & 0 & 0 & 0 & 0 & 0 & a_{4,8} & a_{4,9} \\ 0 & 0 & 0 & a_{5,4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{6,5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{7,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{8,7} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{9,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{10,9} \end{pmatrix}_{(10 \times 9)} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \\ \gamma_7 \\ \gamma_8 \\ \gamma_9 \end{pmatrix} = \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \\ \lambda_9 \end{pmatrix} \tag{22}$$

It is further simplified into the following form:

$$
\begin{pmatrix}
0 & a_{3,3} & 0 & 0 & a_{3,6} & a_{3,7} & 0 & 0 \\
a_{4,2} & 0 & 0 & 0 & 0 & 0 & a_{4,8} & a_{4,9} \\
0 & 0 & a_{5,4} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{6,5} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a_{7,6} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a_{8,7} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a_{9,8} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{10,9}
\end{pmatrix}_{(8\times8)}
\begin{pmatrix}
\gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \\ \gamma_7 \\ \gamma_8 \\ \gamma_9
\end{pmatrix}
=
\begin{pmatrix}
\lambda_2 - \gamma_1 a_{3,1} \\
\lambda_3 - \gamma_1 a_{4,1} \\
\lambda_4 \\
\lambda_5 \\
\lambda_6 \\
\lambda_7 \\
\lambda_8 \\
\lambda_9
\end{pmatrix}
\tag{23}
$$

Usually, it is solved by numerical inversion of the matrix. To improve the efficiency, an analytical expression of the inverse matrix can be written as

$$
B =
\begin{pmatrix}
0 & \frac{1}{a_{4,2}} & 0 & 0 & 0 & 0 & -\frac{a_{4,8}}{a_{4,2}a_{9,8}} & -\frac{a_{4,9}}{a_{4,2}a_{10,9}} \\
\frac{1}{a_{3,3}} & 0 & 0 & 0 & -\frac{a_{3,6}}{a_{3,3}a_{7,6}} & -\frac{a_{3,7}}{a_{3,3}a_{8,7}} & 0 & 0 \\
0 & 0 & \frac{1}{a_{5,4}} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{a_{6,5}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{a_{7,6}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{a_{8,7}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{a_{9,8}} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{a_{10,9}}
\end{pmatrix}_{(8\times8)}
\tag{24}
$$

Thus, Eq. (23) can be reformed as

$$
\begin{pmatrix}
\gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \\ \gamma_7 \\ \gamma_8 \\ \gamma_9
\end{pmatrix}
= B
\begin{pmatrix}
\lambda_2 - \gamma_1 a_{3,1} \\
\lambda_3 - \gamma_1 a_{4,1} \\
\lambda_4 \\
\lambda_5 \\
\lambda_6 \\
\lambda_7 \\
\lambda_8 \\
\lambda_9
\end{pmatrix}
\tag{25}
$$

In most cases, we can obtain a set of non-negative γ following this method. Unfortunately, the aforementioned procedure cannot guarantee that all γ are non-negative. According to Hu [22], a regrouping process is suggested. For instance, the nine stencils are regrouped into three new stencils for the first Gauss point on side i : $S_1^{new} = \{S_2, S_4, S_5\}, S_2^{new} = \{S_3, S_6, S_7\}$, and $S_3^{new} = \{S_1, S_8, S_9\}$.

Finally, we come to the point of nonlinear part. If discontinuities are emerged, nonlinear weights ω_j will be used to measure the smoothness of stencil

$$
\omega_j = \frac{o_i}{\sum_i o_i}, o_i = \frac{\gamma_i}{(\epsilon + IS_i)^2}
\tag{26}
$$

where IS_i is the oscillation indicator and can be written into the following form:

$$IS = \sum_{1 \leq |\alpha| \leq k} \int_{\Omega} |\Omega|^{|\alpha|-1} (D^{\alpha} p(x, y))^2 \, dxdy \tag{27}$$

In smooth regions, the nonlinear weights are equal to the linear weights. Near the discontinuities, the nonlinear weights of the sub-stencil that contains the discontinuities are far more less than the weights of sub-stencils with smooth solution. In this way, numerical oscillation is avoided near the shock and interfaces. The preservation of the pressure equilibrium has been shown in papers [5, 21]. As shown in these papers, the discretization of the transport equation in non-conservative form Eq. (6) is the key factor to the preservation of the pressure equilibrium. Hence, the present method preserves the pressure equilibrium across the interface.

The time integration of the semi-discrete system is accomplished by means of an explicit multi-stage third-order total variation diminishing (TVD) Runge–Kutta scheme,

$$\begin{cases} \mathbf{U}^{(1)} = \mathbf{U}^n + \Delta t R(\mathbf{U}^n) \\ \mathbf{U}^{(2)} = \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\left[\mathbf{U}^{(1)} + \Delta t R(\mathbf{U}^{(1)})\right] \\ \mathbf{U}^{n+1} = \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\left[\mathbf{U}^{(2)} + \Delta t R(\mathbf{U}^{(2)})\right] \end{cases} \tag{28}$$

2.3. The approximate Riemann solver

Here, the Harten-Lax-Van Leer with contact discontinuities HLLC flux is adopted. It approximates the solution with two intermediate states enclosed by the estimated right-going and left-gonging waves and separated by a contact discontinuity. Hence, it is very suitable for problems with shock wave and contact wave. The HLLC flux [5] can be written in the following form:

$$F^{HLLC} = F^{HLLC}\left(\vec{U}^L, \vec{U}^R, \vec{n}\right) = \begin{cases} F_L^* & 0 \leq s_m \\ F_R^* & s_m \leq 0 \end{cases} \tag{29}$$

where the flux with superscript is defined as

$$\begin{cases} F_L^* = F\left(U^L, \vec{n}\right) + s_L\left(U_L^* - U^L\right) \\ F_R^* = F\left(U^R, \vec{n}\right) + s_R\left(U_R^* - U^R\right) \end{cases} \tag{30}$$

with

$$\begin{cases} s_L = min\left(u_n^L - a^L, \tilde{q} - \tilde{c}\right) \\ s_r = max\left(u_n^R - a^R, \tilde{q} + \tilde{c}\right) \end{cases} \tag{31}$$

Here, a^L, a^R, and $\tilde{a}$ are the left sound speed, the right sound speed, and the averaged sound speed respectively. For the fluid with the stiffened gas equation of state (EOS), the sound speed can be calculated by

$$a = \sqrt{\gamma \frac{p + \pi}{\rho}} \tag{32}$$

3. NUMERICAL RESULTS

In order to verify the present method for compressible multi-fluid flows on unstructured mesh, four different numerical cases are tested.

3.1. Two-dimensional vortex evolution problem

To investigate the order of accuracy of the present method, the two-dimensional vortex evolution problem is employed here. A similar case can be found in Frahan [23]. In this test, a perturbation is added to the mean flow $(\rho, u, v, p) = (1, 1, 1, 1)$. The perturbation is

a) $h = 1$ b) $h = 1/2$ c) $h = 1/4$

d) $h = 1/8$ e) $h = 1/16$ f) $h = 1/32$
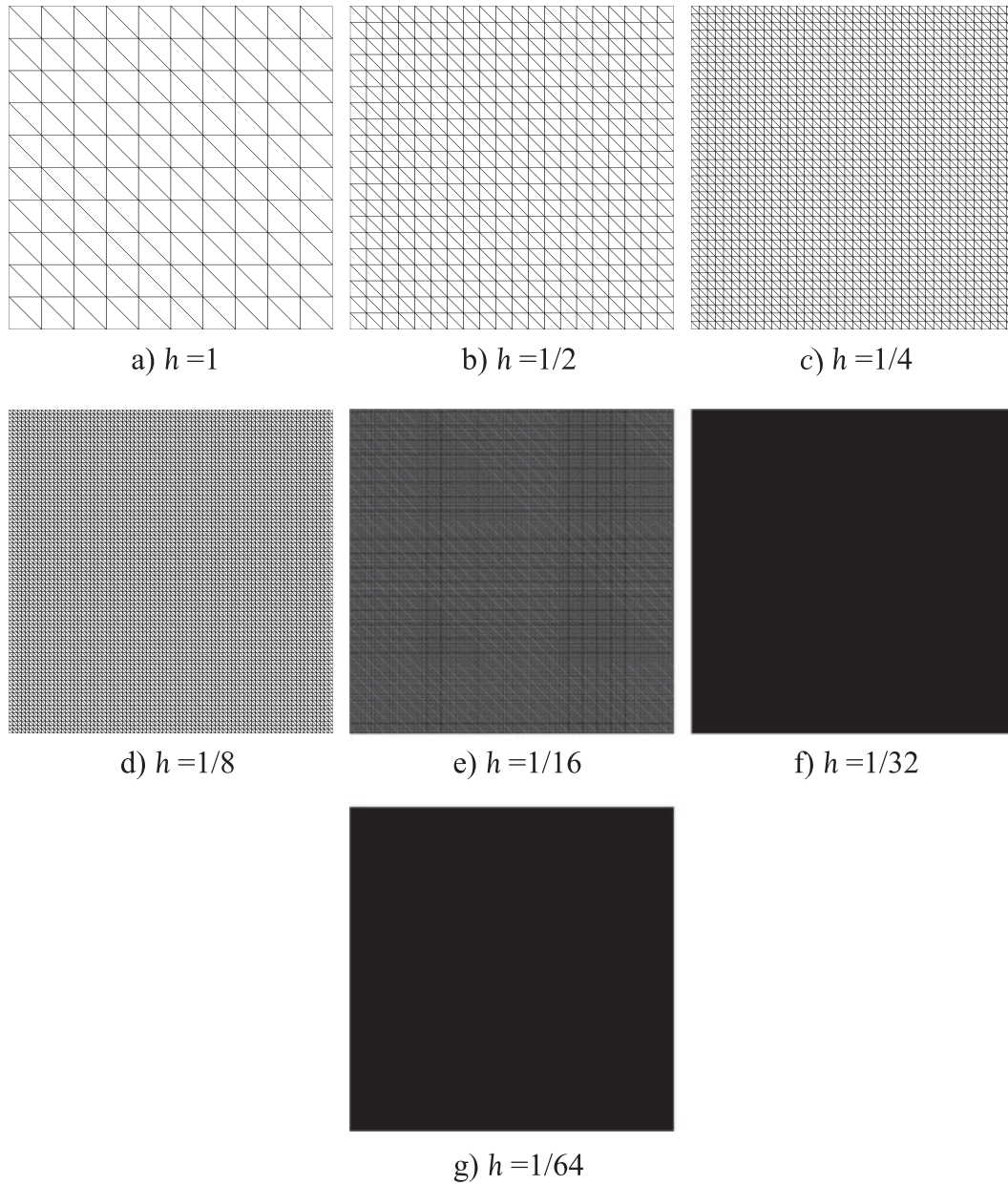
g) $h = 1/64$

Figure 2. Uniform grids for vortex evolution problem.

$$\begin{cases} \delta\rho = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}e^{(1-r^2)} \\ \delta\gamma = \frac{\varepsilon}{2\pi}e^{0.5(1-r^2)} \\ \rho = 1 + \delta\rho \\ \gamma = 1.4 + \delta\gamma \end{cases} \tag{33}$$

with

$$r^2 = (x-5)^2 + (y-5)^2 \tag{34}$$

Here, the vortex strength $\varepsilon = 5$. The computational domain is taken to be $(0, 10) \times (0, 10)$. It can be easily observed that the ratio of specific heat changes smoothly. The exact solution of this problem is the perturbation moves with the speed $(1, 1)$ in the diagonal direction. The simulation was carried
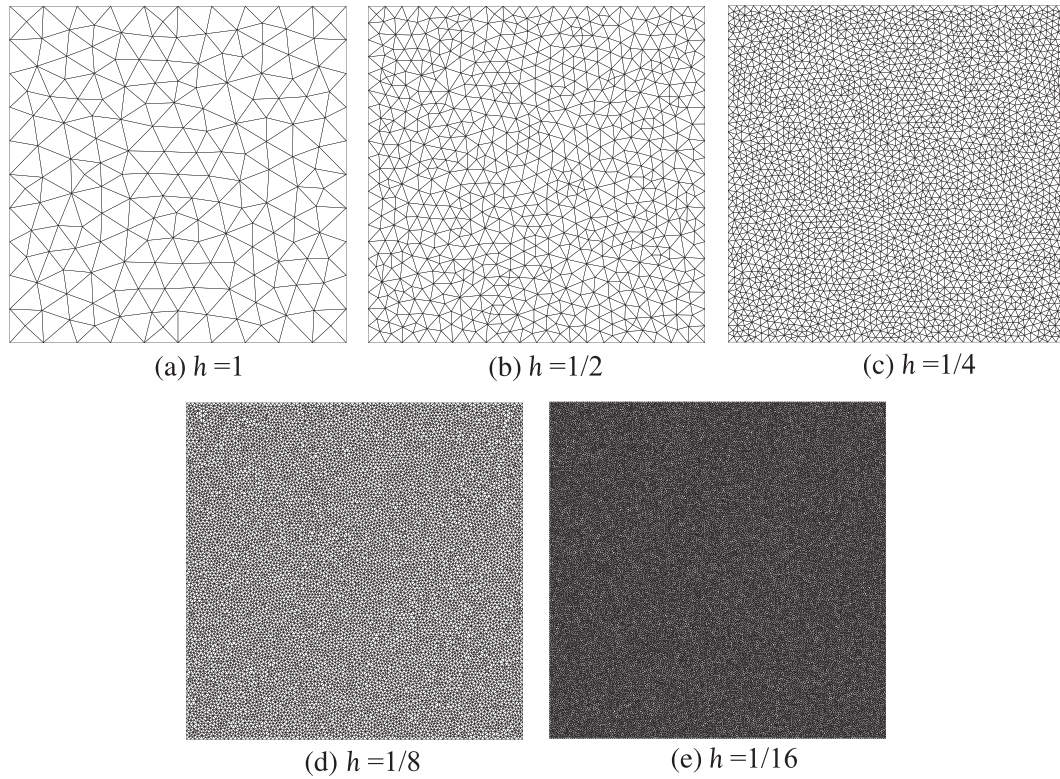
(a) $h = 1$ (b) $h = 1/2$ (c) $h = 1/4$

(d) $h = 1/8$ (e) $h = 1/16$

Figure 3. Unstructured irregular grids for vortex evolution problem.

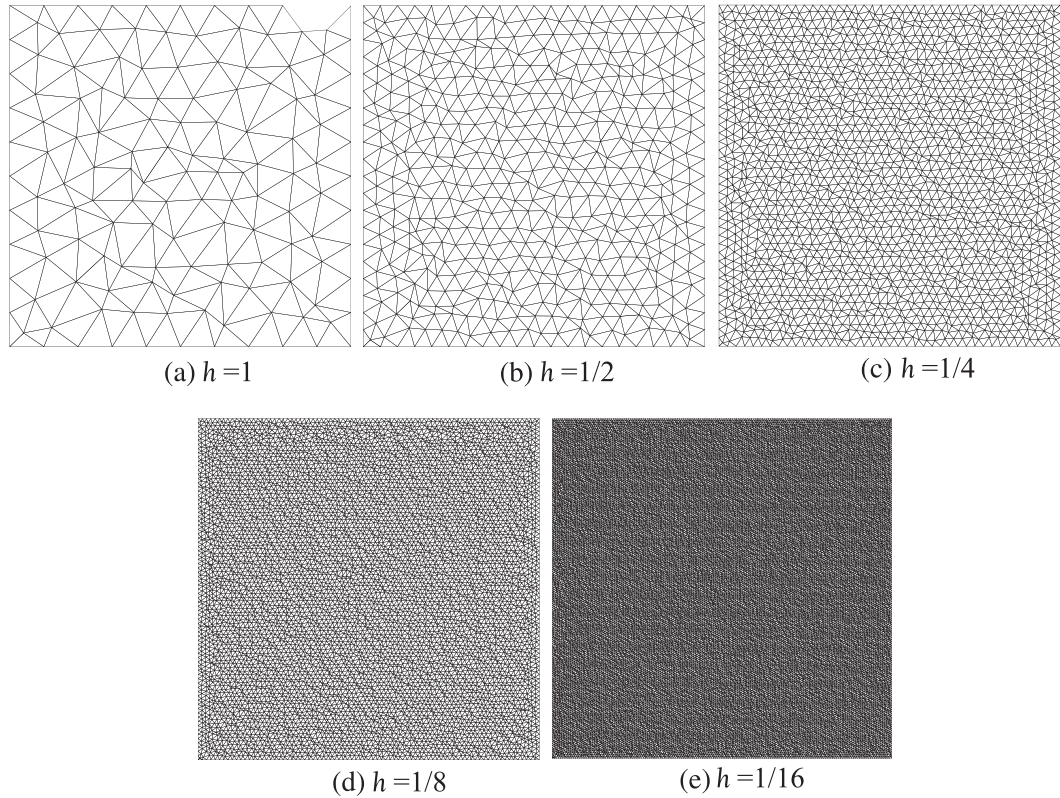(a) $h = 1$ (b) $h = 1/2$ (c) $h = 1/4$

(d) $h = 1/8$ (e) $h = 1/16$

Figure 4. Unstructured stretched grids for vortex evolution problem.

out until $t = 0.5$. Three different kinds of mesh, uniform meshes, irregular grids, and stretched grids, are employed and shown in Figures 2–4.

The order of accuracy can be calculated by

$$O = \frac{lg\left(\frac{L_{p,c}}{L_{p,f}}\right)}{lg\,2} \tag{35}$$

where c denotes the coarse mesh and f denotes the fine mesh. L is the norm based on the error between the numerical and exact solutions. It can be defined as

Table I. Errors and order of accuracy in density (uniform meshes).

h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	4.22E−3	–	1.53E−2	–	0.14E0	–
1/2	1.24E−3	1.76	5.47E−3	1.48	8.61E−2	0.76
1/4	2.75E−4	2.18	1.57E−3	1.80	3.69E−2	1.22
1/8	2.85E−5	3.27	1.80E−4	3.13	5.50E−3	2.75
1/16	2.69E−6	3.40	1.34E−5	3.74	3.30E−4	4.06
1/32	3.42E−7	2.98	1.45E−6	3.21	2.33E−5	3.82
1/64	6.19E−8	2.47	1.84E−7	2.98	2.43E−6	3.27

Table II. Errors and order of accuracy in $1/(\gamma - 1)$ (uniform meshes).

h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	1.51E−2	–	2.81E−2	–	0.17E0	–
1/2	4.48E−3	1.75	8.33E−3	1.75	5.19E−2	1.75
1/4	7.77E−4	2.53	1.52E−3	2.45	1.19E−2	2.12
1/8	8.80E−5	3.14	1.69E−4	3.17	8.41E−4	3.82
1/16	8.93E−6	3.30	1.70E−5	3.31	8.42E−5	3.32
1/32	9.00E−7	3.23	1.80E−6	3.21	8.26E−6	3.35
1/64	1.44E−7	2.64	2.42E−7	2.89	1.03E−6	3.00

Table III. Errors and order of accuracy in density (irregular meshes).

h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	2.64E−3	-s-	9.91E−3	–	0.11E0	–
1/2	7.91E−4	1.74	3.72E−3	1.41	6.13E−2	0.86
1/4	1.26E−4	2.65	7.44E−4	2.32	1.95E−2	1.65
1/8	1.19E−5	3.40	3.73E−5	2.39	2.10E−3	3.22
1/16	1.25E−6	3.26	6.03E−6	3.56	1.35E−4	3.96

Table IV. Errors and order of accuracy in $1/(\gamma - 1)$ (irregular meshes).

h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	1.02E−2	–	1.75E−2	–	7.36E−2	–
1/2	2.26E−3	2.18	4.11E−3	2.09	2.47E−2	1.57
1/4	3.31E−4	2.77	6.18E−4	2.74	2.66E−3	3.21
1/8	3.73E−5	3.15	7.05E−5	3.13	3.19E−4	3.06
1/16	4.21E−6	3.15	7.77E−6	3.18	6.22E−5	2.35

Table V. Errors and order of accuracy in density (stretched meshes).

h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	2.87E−3	–	1.08E−2	–	0.13E0	–
1/2	8.32E−4	1.79	3.75E−3	1.52	6.60E−2	0.96
1/4	1.45E−4	2.52	7.10E−4	2.40	1.80E−2	1.87
1/8	1.51E−5	3.27	8.69E−5	3.03	2.63E−3	2.78
1/16	1.69E−6	3.16	9.03E−6	3.27	2.88E−4	3.19

Table VI. Errors and order of accuracy in $1/(\gamma - 1)$ (stretched meshes).

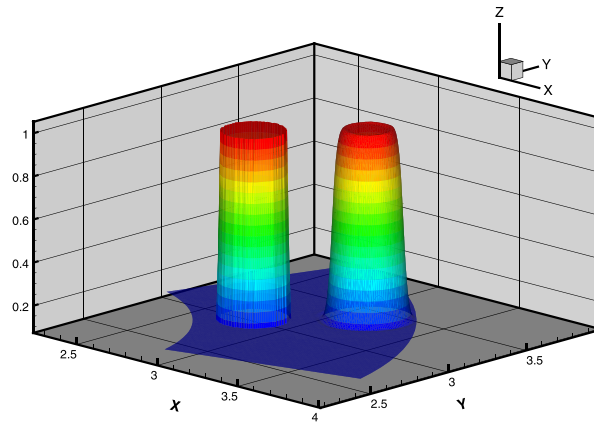| h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	1.21E-2	-	1.08E-2	-	9.35E-2	-
1/2	2.72E-3	2.15	5.24E-3	1.98	3.57E-2	1.39
1/4	4.53E-4	2.58	8.73E-4	2.59	6.96E-3	2.39
1/8	5.60E-5	3.02	1.12E-4	2.96	1.15E-3	2.59
1/16	6.70E-6	3.06	1.39E-5	3.01	2.43E-4	2.25

Figure 5. Comparisons of the density plot of interface only problem at $t = 0$ and $t = 0.36$.

Table VII. Errors and order of accuracy in the derivative of density for x (irregular meshes).

h	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
1	7.93E−3	–	2.94E−2	–	2.37E−1	–
1/2	3.01E−3	1.40	1.22E−2	1.27	1.23E−1	0.94
1/4	7.68E−4	1.97	3.36E−3	1.86	5.39E−2	1.19
1/8	1.69E−4	2.19	7.15E−4	2.23	9.67E−3	2.48
1/16	3.93E−5	2.10	1.66E−4	2.11	1.65E−3	2.55

$$L_m = \left(\frac{\sum_{i=1}^{N_{cv}} A_i \left| \bar{E}_i^m \right|}{\sum_{i=1}^{N_{cv}} A_i} \right)^{\frac{1}{m}} \tag{36}$$

$$\bar{E}_i = \frac{1}{A_i} \iint_{CV_i} (U^{exa} - U^{num}(x, y)) \, dA_i \tag{37}$$

In Eq. (37), U^{exa} denotes the exact solution and U^{num} the numerical solution. The integration is calculated by Gaussian quadrature. The infinity norm is a measure of local accuracy

$$L_\infty = max \left| \bar{E}_i \right| \tag{38}$$

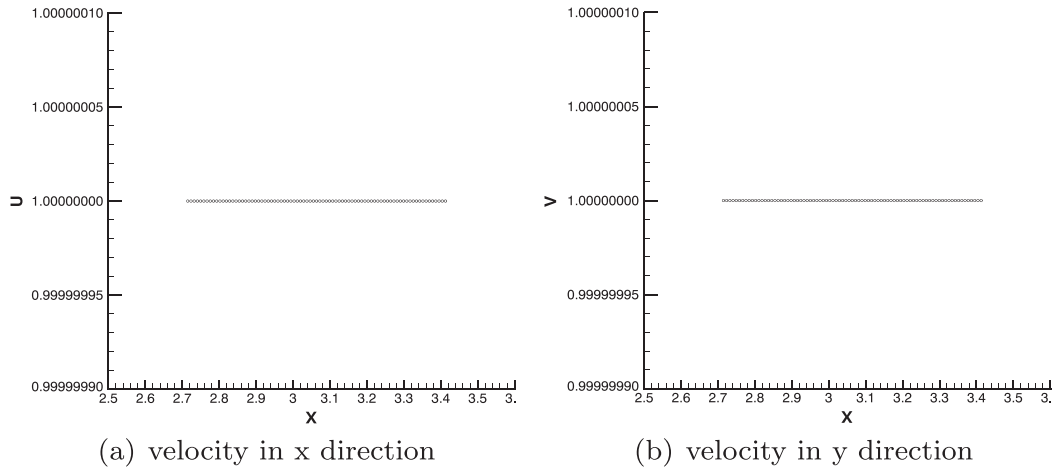(a) velocity in x direction   (b) velocity in y direction

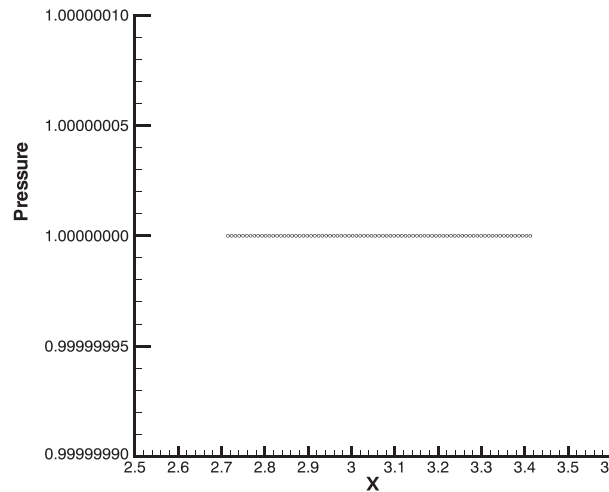Figure 6. Velocity profile for interface problem at time $t = 0.36$.

Figure 7. Pressure profile for interface problem at time $t = 0.36$.

The error analysis and order of convergence using different girds have been given in Tables I–VI. These results show that the present method can achieve third-order accuracy in uniform grids, irregular grids, and stretched grids. However, the convergence rates in coarse meshes ($h = 1, 1/2$;) are lower than those in fine meshes. Similar findings are shown in other papers using the finite difference (FD) method [12] or the finite volume (FV) method [22]. Besides, the convergence of the derivative of density in x direction is also tested in the irregular meshes. The results in Table VII show that derivative of density can reach second order, which is consistent with the convergence of the density.

3.2. Advection of a gas–gas interface problem

To verify that the present method can prevent the oscillation of density and pressure across the interface, the interface-only problem is investigated. Initially, one fluid with a circular shape is surrounded by another fluid. The radius of the circular interface r_0 is 0.16. The computational domain is an annulus where the inner and outer radiuses are 1 and 2, respectively. The circle is located at $(2.9, 2.9)$. There are no shocks and other perturbations in the flow field. The initial values are as follows:

$$(\rho, \gamma, p_\infty) = \begin{cases} (1, 1.4, 0) & r < r_0 \\ (0.125, 4, 0) & r \leqslant r_0 \end{cases} \tag{39}$$

Both fluids move at the same speed $(1, 1)$. The inner fluid with a circular shape should move with the constant velocity along the diagonal direction. It can be easily observed that the inner fluid with a circular shape moves along the diagonal direction and its position agrees well with the exact solution. The surface plots of density at $t = 0$ and $t = 0.36$ are also presented in Figure 5. From it, it is found that the interface at $t = 0.36$ is relatively sharp as compared with the initial one. Besides, the velocity profiles in x direction and in y direction are shown in Figure 6. The pressure profile along the center line is shown in Figure 7. As shown in Figures 6 and 7, the velocity and pressure are plotted in extremely fine scale (10^{-7}). The velocity still keeps exactly the same as the initialized one, and there is no oscillation at all. It shows that the present method can prevent the oscillation of density and pressure across the interface.
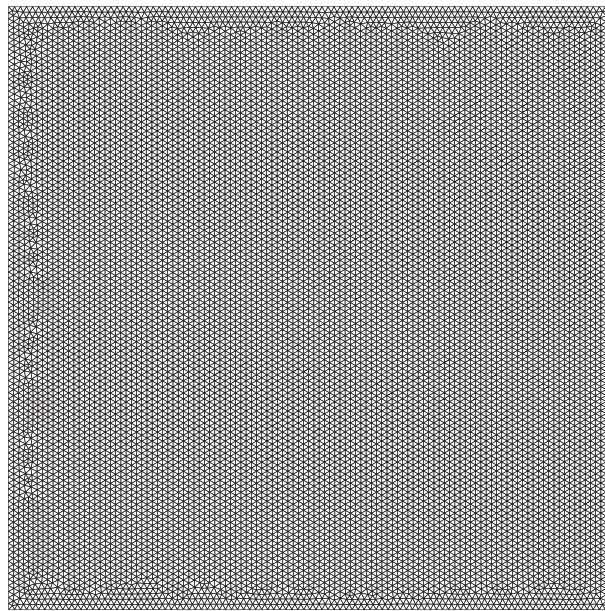

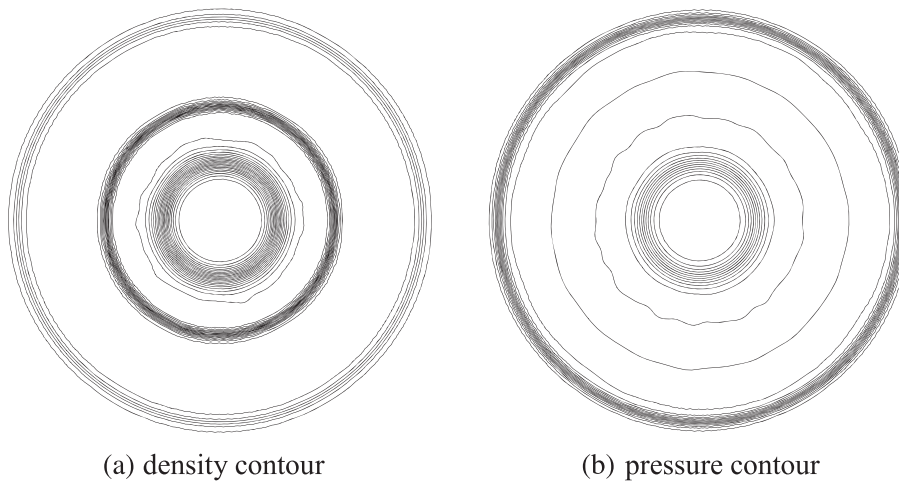Figure 8. Computational domain and the grids for bubble explosion under water.

(a) density contour   (b) pressure contour

Figure 9. Contours of density and pressure at $t = 0.058$.

3.3. Bubble explosion under water

Initially, one circular fluid is surrounded by another fluid and is located at the center of the domain with radius r_0 as 0.2. The computational domain is $(0, 1) \times (0, 1)$, and the unstructured grid is shown in Figure 8. The primitive variables of two fluids have been given,

$$(\rho, u, v, p, \gamma, \pi) = \begin{cases} (1.241, 0, 0, 2.753, 1.4, 0) \\ (0.991, 0, 0, 3.059 \times 10^{-4}, 5.5, 1.505) \end{cases} \tag{40}$$

The two fluids are at rest initially, and there is a jump of the density and pressure across the interface. The results at $t = 0.058$ are shown in Figure 9. When the bubble explodes under the water, there are an outgoing shock wave in the water and an incoming rarefaction wave in the water. It can be easily observed that the contours of density and pressure by our method are captured
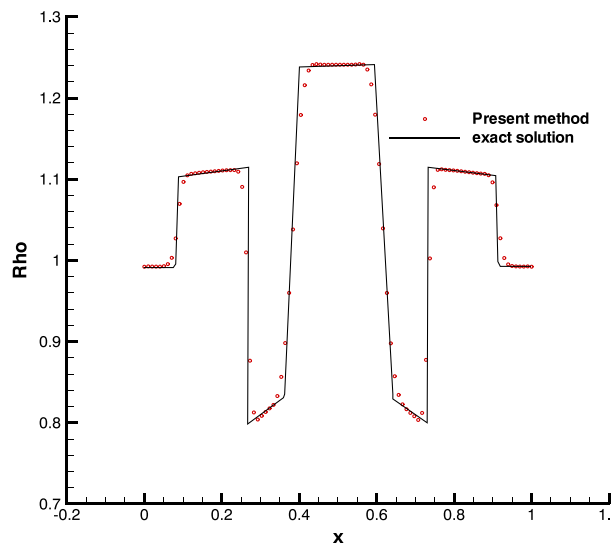

Figure 10. Density profile along vertical centerline for bubble explosion under water.
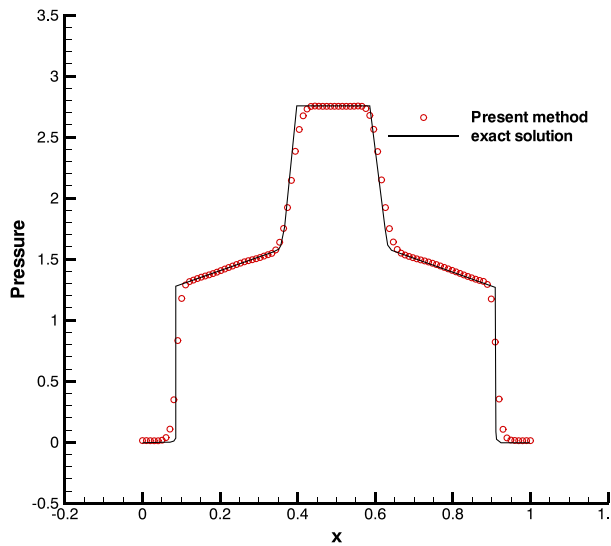

Figure 11. Pressure profile along vertical centerline for bubble explosion under water.

symmetrically and better than those in the work of Shyue [4]. Besides, the profiles along $y = 0.5$ are also plotted in Figures 10 and 11. They show that the present results agree well with the reference result and there is no pressure oscillation on the interface.

3.4. Shock–interface interaction inside the cylindrical vessel

The purpose of this case is to demonstrate the ability of the current method in dealing with relatively complex geometry (e.g., a rigid cylindrical vessel). The vessel is filled with air and helium. Initially, two gases are separated by a planer interface [24]. Air is on the left side and helium on the other side. The material interface is located at $x = 0$. The vessel is then impulsively driven in the x direction causing a curved shock along the left portion of the boundary and a rarefaction to form along the right portion of the boundary. For the convenience of computation, the vessel is fixed, and two gases move with an initial uniform velocity. The reflective boundary condition is employed at the boundary. The unstructured grid is employed. The reflective boundary condition is employed at the boundary. The initial conditions are as follows:

$$(\rho, u, v, p, \gamma, \pi) = \begin{cases} (1, -1, 0, 1, 1.4, 0) \\ (0.138, -1, 0, 1, 1.67, 0) \end{cases} \qquad (41)$$

Firstly, a semi-circular shock wave is formed at the left side and begins to move to the right. A rarefaction is formed at the right boundary and moves to the left. Afterwards, the shock wave continues moving to the right and interacts with the interface. At $t = 0.5$, a diverging shock is formed as shown in Figure 12(b). The right-going shock wave also makes the interface move to the right at the same time. The right-going shock wave will reflect back when it arrives at the boundary of the vessel. As shown in Figure 12(c), the reflected shock wave interacts with the interface again

(a) t=0.25   (b) t=0.50
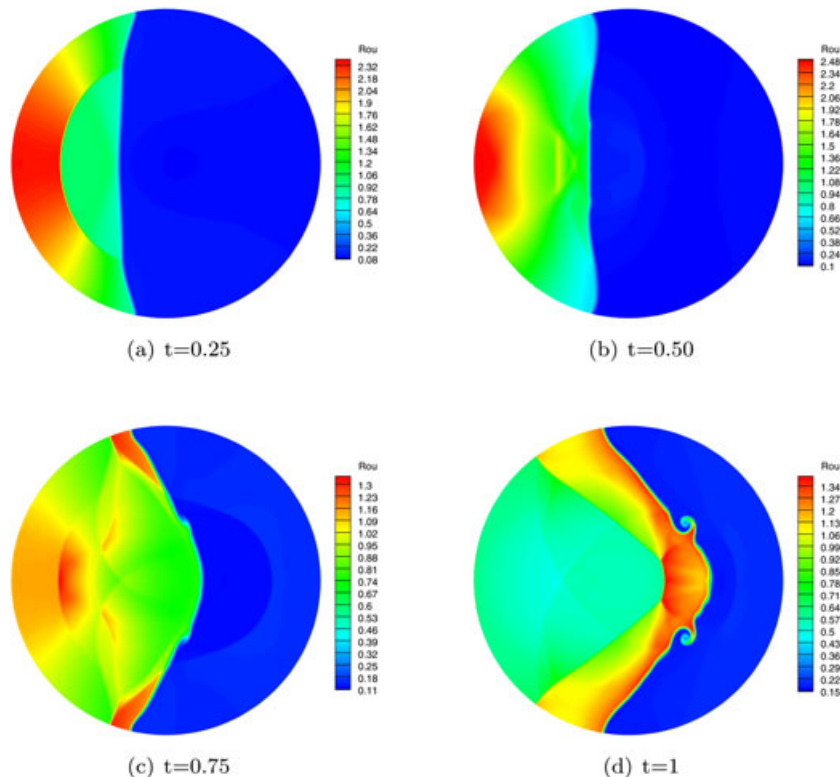
(c) t=0.75   (d) t=1

Figure 12. Density contour for shock-interface interaction at different time.

and travels into the air. The Richtmyer–Meshkov instability is finally formed around the interface. As shown in Figure 12(d), these tiny structures are clearly captured by the present higher-order WENO method.

4. CONCLUSION

In this article, an improved WENO method based on unstructured grid for the compressible multi-fluid flows is presented. The third-order accuracy WENO finite volume method based on unstructured grid is used to discretize compressible Euler equation. Usually, matrix inversion has to be numerically solved to obtain the linear weights for WENO. This procedure takes too much computational cost and is easy to blow up when the grid deforms greatly. In our improved method, the inverse matrix of coefficient is analytically calculated instead of the numerical calculation. It makes the present method more robust than the original one. But the extension to higher order (more than three) is not straightforward because the inversion of the matrix cannot be analytically expressed. Four different cases are used to examine the performance of our method. They are vortex evolution problem, advection of a gas–gas interface problem, bubble explosion under water, and shock–interface interaction inside the cylindrical vessel. Numerical results show that there is no spurious oscillation of velocity and pressure across the interface and three-order accurate result can be achieved. The extension to 3D on tetrahedral grids is still in progress and will be reported in another paper.

REFERENCES

1. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics* 1981; **39**:201–225.
2. Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics* 1999; **152**:457–492.
3. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
4. Shyue KM. An efficient shock-capturing algorithm for compressible multicomponent problems. *Journal of Computational Physics* 1998; **142**:208–242.
5. Zheng HW, Shu C, Chew Y. An object-oriented and quadrilateral-mesh based solution adaptive algorithm for compressible multi-fluid flows. *Journal of Computational Physics* 2008; **227**:6895–6921.
6. Li W, Ren YX. High-order k-exact WENO finite volume schemes for solving gas dynamic Euler equations on unstructured grids. *International Journal for Numerical Methods in Fluids* 2012; **70**:742–763.
7. Xu K. BGK-based scheme for multicomponent flow calculations. *Journal of Computational Physics* 1997; **134**: 122–133.
8. Wang ZJ. High-order methods for the Euler and Navier–Stokes equations on unstructured grids. *Progress in Aerospace Sciences* 2007; **43**:1–41.
9. Ekaterinaris JA. High-order accurate, low numerical diffusion methods for aerodynamics. *Progress in Aerospace Sciences* 2005; **41**:192–300.
10. Harten A, Engquist B, Osher S, Chakravarthy SR. Uniformly high order accurate essentially non-oscillatory schemes-III. *Journal of Computational Physics* 1987; **71**:231–303.
11. Liu XD, Osher S, Chan T. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics* 1994; **115**:200–212.
12. Jiang GS, Shu CW. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics* 1996; **126**:202–228.
13. Cockburn B, Karniadakis GE, Shu CW. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin Methods*. Springer: Berlin Heidelberg, 2000.
14. Wang ZJ. Spectral (finite) volume method for conservation laws on unstructured grids basic formulation. *Journal of Computational Physics* 2002; **178**:210–251.
15. Johnsen E, Colonius T. Implementation of WENO schemes in compressible multicomponent flow problems. *Journal of Computational Physics* 2006; **219**:715–732.

16. Dumbser M, Boscheri W. High-order unstructured Lagrangian one-step WENO finite volume schemes for non-conservative hyperbolic systems: applications to compressible multi-phase flows. *Computers & Fluids* 2013; **86**: 405–432.
17. Movahed P, Johnsen E. A solution-adaptive method for efficient compressible multifluid simulations, with application to the Richtmyer–Meshkov instability. *Journal of Computational Physics* 2013; **239**:166–186.
18. Abgrall R. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics* 1994; **114**:45–58.
19. Dumbser M, Kaser M, Titarev VA, Toro EF. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics* 2007; **226**:204–243.
20. Coralic V, Colonius T. Finite-volume WENO scheme for viscous compressible multicomponent flows. *Journal of Computational Physics* 2014; **274**:95–121.
21. Abgrall R. How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach. *Journal of Computational Physics* 1996; **125**:150–160.
22. Hu C, Shu CW. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics* 1999; **150**:97–127.
23. de Frahan MTH, Varadan S, Johnsen E. A new limiting procedure for discontinuous Galerkin methods applied to compressible multiphase flows with shocks and interfaces. *Journal of Computational Physics* 2015; **280**:489–509.
24. Banks JW, Schwendeman DW, Kapila AK, Henshaw WD. A high-resolution Godunov method for compressible multi-material flow on overlapping grids. *Journal of Computational Physics* 2007; **223**:262–297.
25. Zheng HW, Shu C, Chew YT, Qin N. A solution adaptive simulation of compressible multi-fluid flows with general equation of state. *International Journal for Numerical Methods in Fluids* 2011; **67**:616–637.