

A Vertex-Centered Arbitrary Lagrangian-Eulerian Finite Volume Method with Sub-Cells for Two-Dimensional Compressible Flow

Xiaolong Zhao¹, Xijun Yu², Zupeng Jia^{5,*}, Shijun Zou^{3,*}
and Meilan Qiu⁴

¹ School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, Henan 450001, China

² Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100088, China

³ School of Mathematical Sciences, Capital Normal University, Beijing 100048, China

⁴ School of Mathematics and Statistics, Huizhou University, Huizhou, Guangdong 516007, China

⁵ Institute of Applied Physics and Computational Mathematics, Beijing 100088, China

Received 23 March 2022; Accepted (in revised version) 14 October 2022

Abstract. In this paper, we present a new vertex-centered arbitrary Lagrangian-Eulerian (ALE) finite volume scheme for two-dimensional compressible flow. In our scheme, the momentum equation is discretized on the vertex control volume, while the mass equation and the energy equation are discretized on the sub-cells which are included in the vertex control volume. We attain the average of the fluid velocity on the vertex control volume directly by solving the conservation equations. Then we can obtain the fluid velocity at vertex with the reconstructed polynomial of the velocity. This fluid velocity is chosen as the mesh velocity, which makes the mesh move in a Lagrangian manner. Two WENO (Weighted Essentially Non-Oscillatory) reconstructions for the density (the total energy) and the velocity are used to make our scheme achieve the anticipated accuracy. Compared with the general vertex-centered schemes, our scheme with the new approach for the space discretization can simulate some multi-material flows which do not involve large deformations. In addition, our scheme has good robustness, and some numerical examples are presented to demonstrate the anticipated accuracy and the good properties of our scheme.

AMS subject classifications: 65M60

Key words: Vertex-centered, arbitrary Lagrangian-Eulerian, sub-cells, multi-material flows.

*Corresponding author.

Emails: xiaolongzhao109@zzu.edu.cn (X. Zhao), yuxj@iapcm.ac.cn (X. Yu), zpjia@iapcm.ac.cn (Z. Jia), js_zousj@foxmail.com (S. Zou), meilanqiu16@163.com (M. Qiu)

1 Introduction

Simulations of the compressible multi-material fluid flows can be classified into two computational frameworks, each with its own advantages and disadvantages. The first one is Lagrangian [1–4] in which the mesh is embedded in fluid and moves with it; the second, known as Eulerian [5,6] treats the mesh as a fixed reference frame through which the fluid moves. In order to combine the advantages of the Lagrangian method and the Eulerian method, Hirt et al. [7] proposed an Arbitrary Lagrangian-Eulerian (ALE) method in which the grid points may move in some arbitrarily specified ways. The ALE method is a powerful tool to solve fluid dynamic problems with large deformation. It was primarily developed from the finite difference method for fluid flows and later extended in the context of the finite element method and the finite volume method for both fluid and structure dynamics. Since the computational mesh of the ALE method can move with arbitrary velocity which is independent of the fluid velocity, the best features from the traditional Lagrangian method and the Eulerian method can be combined and embodied in the ALE method. Basically, there are two categories of the ALE methods which are the so-called indirect method [8–14] and the direct one [15–19]. In the indirect ALE method, the computational mesh firstly moves in a Lagrangian manner. When the mesh becomes too distorted, a remeshing step is carried out to generate a new grid with better quality. Then the numerical solutions on the old mesh are projected to the new one, which is called the remapping step. This method has been widely used for solving the multi-phase flow problems and the multi-material flow problems. For instance, Maire et al. [12] developed a cell-centered multi-material indirect ALE scheme for solving the compressible gas dynamics equations on two-dimensional unstructured grid, and the multi-material simulations utilized either the concentration equations for miscible fluids or the Volume Of Fluid (VOF) capability with interface reconstruction for immiscible fluids; Barlow and Maire et al. [14] reviewed the recent developments in the indirect ALE methods for modeling the high speed compressible multi-material flows on complex geometry with general polygonal meshes. In the direct ALE method, the mesh movement is taken into consideration directly in the computation of the numerical flux. Therefore, this method has no remapping step, and it is easier to be used for constructing high-order accuracy schemes than other methods with the remapping step. For instance, Zhao et al. [19] presented a high-order direct ALE Discontinuous Galerkin (DG) method for compressible single-material flow on the adaptive moving unstructured meshes; Boscheri et al. [17] proposed a high-order accurate ADER schemes in the direct ALE context.

There are two commonly used approaches for discretizing the gas dynamic equations. One is the staggered-grid hydrodynamics (SGH) [20–23] in which the velocity is defined at nodes while the other variables (the density, the pressure and the specific internal energy) are located inside the cells. The other is the cell-centered hydrodynamics (CCH) [1, 2, 4, 17, 24–34] in which all the primary variables are defined in the cells. The CCH method offers some advantages over the SGH one. For example, the SGH schemes use different control volumes for the primary variables, and it is difficult to

construct the consistent high-order SGH schemes for all these variables. On the contrary, the CCH schemes use the same control volume for all the primary variables, thus it is possible to construct the coherent high-order SGH schemes for all these variables with the ideas of Godunov methods.

Besides the SGH schemes and the CCH schemes, the vertex-centered hydrodynamics (VCH) has attracted more and more interests in recent years, see [35–38]. In this kind of schemes, the conserved variables (the mass, the momentum, and the total energy) are all stored at the nodes of mesh. Morgan et al. [35] proposed a vertex-centered hydrodynamic scheme which is suitable for modeling the complex compressible flows on tetrahedron meshes. The scheme reduced to a purely Lagrangian approach when the flow is linear. The evolution equations for momentum and total energy are discretized on the vertex control volume using an edge-based finite element approach with linear basis functions. A multidimensional Riemann-like problem was introduced at the center of the tetrahedron to account for the discontinuities in the flow such as a shock. Morgan et al. [36] presented a vertex-centered Arbitrary Lagrangian-Eulerian hydrodynamic approach for tetrahedral meshes, and this ALE version was derived from the Lagrangian one in [35]. Aguirre et al. [37] developed an upwind vertex-centered finite volume solver for Lagrangian solid dynamics. Liu et al. [38] developed a vertex-centered DG method in the direct ALE form for compressible single-material flow, and the vertex control volumes of this scheme are constructed with curved edges. This scheme is third-order accurate both in space and time, and it is conservative for the mass, the momentum and the total energy.

In the simulations for the multi-material flows, general Lagrangian vertex-centered methods often need to deal with the mixed control volumes involving the material interfaces. It is worthy of studying the problem of how to distinguish the materials and obtain the pressure conveniently in the mixed control volumes. Considering the research values of the vertex-centered methods and the advantages of the direct ALE methods for dealing with the multi-material flows, we propose a new vertex-centered direct ALE method for the compressible multi-material flows in this paper. In our scheme, the fluid velocity at vertex which is derived directly from the numerical solution is selected as the mesh velocity. In other words, the vertexes move in a Lagrangian manner. In our scheme, a new approach for the space discretization is employed, which can help our scheme deal with some multi-material flows conveniently and effectively. The control volume in our scheme includes several sub-cells, the momentum is defined on the vertex control volume, while the density and the total energy are defined on the sub-cells of the control volume. The space discretization for momentum conservation equation is carried out on the control volume, which is similar to other general vertex-centered methods. The space discretizations for mass equation and energy equation are carried out on the sub-cells of the control volume, which is different to other general vertex-centered methods. Hence, there is unified momentum on each control volume, and there are independent energy, density, pressure and specific heat coefficient on each sub-cell of the control volume. This way of implementing the space discretization allows our scheme to simulate the multi-material flows conveniently. The key reason for

choosing this new approach for the space discretization is as follows. Our method belongs to the Godunov-type methods which use the approximate Riemann solvers. In order to solve the Riemann problems on the edges of the vertex control volume, we need not only the kinematic variables (momentum, etc.) on the control volume but also the thermodynamic variables (such as the density and the energy, etc.). If we define the thermodynamic variables on each sub-cell of the vertex control volume, we can obtain the total thermodynamic variables on the vertex control volume by achieving a weighted average of the thermodynamic variables on the sub-cells. If the thermodynamic variables are also defined on the vertex control volume directly, it is difficult to distinguish the multi-material flows directly and conveniently. Therefore, we define the thermodynamic variables on each sub-cell of the control volume, and the material interfaces can be clearly and directly depicted by using the grid edges. Finally, the multi-material flows can be simulated by our scheme conveniently, which is the most critical design target of our scheme in this paper.

For clarity, a summary of the pros and cons of our scheme in contrast to some other existing methods is given as follows. Compared with the general vertex-centered methods, our method has the ability of simulating some multi-material flows which do not involve large deformations conveniently, while the algorithm of our method is more complex because of the specificity of the space discretizations for mass equation and energy equation. The general cell-centered Lagrangian methods usually use the velocity solver to obtain an approximate fluid velocity, and then they take the approximate fluid velocity as the mesh velocity. For example, the cell-centered Lagrangian scheme [2] studied by Maire et al. used a strong sufficient condition to construct a nodal solver which is based on the conservation of momentum, the conservation of energy and the entropy conditions. When this kind of schemes is used for dealing with the cases involving the discontinuities like shocks, the numerical dissipation caused by the velocity solver will have certain impacts on the result of the simulation. Our method obtains the fluid velocity by solving the Euler equations directly on the vertex control volume and then further uses the fluid velocity to determine the Lagrangian nodal velocity. This approach avoids the numerical dissipation and the error caused by the solver of the approximate fluid velocity. The general staggered Lagrangian methods usually introduce the artificial viscosity into the algorithms to deal with the discontinuities, while our method which belongs to the Godunov-type methods uses an approximate Riemann solver to achieve this. In the case with other same conditions, the resolution of discontinuities of the numerical solution obtained by the staggered Lagrangian methods may be not as sharp as the one obtained by our method. Some numerical examples are presented to demonstrate the anticipated accuracy and the good properties of our scheme. For instance, compared with the Lagrangian methods in [29] and [32], our scheme obtains higher resolution in the vicinity of discontinuous solutions with maintaining better quality of the meshes, and this scheme runs for a longer time when it is used for the simulation of the Saltzman problem. A noteworthy point is that our scheme does not apply to the single/multi-material cases involving large deformations because of the mesh distortion. Although the scope of the application of our scheme is

currently restricted, it provides a concise idea for the simulations of the multi-material flows, and we will consider combining other techniques with our scheme to simulate the multi-material cases involving large deformations in the future work.

The layout of the rest of this paper is as follows. The governing equations of two-dimensional compressible gas dynamics are given in Section 2. The process for solving the governing equations with our vertex-centered ALE method is described in Section 3. In Section 4, our method is tested by a suite of numerical tests to assess its accuracy and properties. The whole paper is concluded in Section 5.

2 Governing equations

The Euler equations of two-dimensional compressible gas dynamics are as follows:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0, \\ \frac{\partial (\rho \mathbf{V})}{\partial t} + \nabla \cdot (\rho \mathbf{V} \otimes \mathbf{V}) + \nabla P = 0, \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{V}) + \nabla \cdot (P \mathbf{V}) = 0, \end{cases} \quad (2.1)$$

where $\mathbf{U} = (\rho, \rho \mathbf{V}, \rho E)$ denotes the vector of the conservative variables; ρ , $\mathbf{V} = (u, v)$, P , and E are the density, the velocity vector, the pressure, and the total energy, respectively; $\mathcal{E} = E - \frac{1}{2} \|\mathbf{V}\|^2$ denotes the internal energy. The system is closed by the following equation of state (EOS): $P = (\gamma - 1) \rho \mathcal{E}$, where γ is the ratio of specific heats of fluid. This system of PDEs is valid within a set of admissible states where $\rho > 0$, $\mathcal{E} > 0$, and $\gamma > 1$.

3 The vertex-centered scheme in the framework of ALE

3.1 The structure of the vertex control volume

In our scheme, the governing equations are solved on a set of non-overlapping vertex control volumes. These control volumes are constructed with the following procedure. Firstly, the whole domain is partitioned into an ensemble of non-overlapping structured quadrilateral cells with straight edges. The vertex is denoted by p , and the primal cell is denoted by z . Considering vertex p in the primal computational mesh, we use the midpoints of the related edges which involve p and the center points of the related cells which involve p to build the vertex control volume Ω_p at vertex p . Firstly, let z_j denote the j th related cell ($j = 1, \dots, m$; m denotes the total number of the related cells) of p (z_j also denotes the center point of the j th related cell of p , and it is defined by coordinates that are the simple average of those of the related cell points), and let the hollow circles denote the midpoints of the primal related edges of p . Secondly, we connect z_j and the midpoints (the hollow circles) of the primal related edges of p by dashed lines in counter-clockwise order. One can also refer to [21] for the process. There are several sub-cells named $\Omega_{z_j}^p$ ($j = 1, \dots, m$) in each Ω_p , where m also denotes the total number of

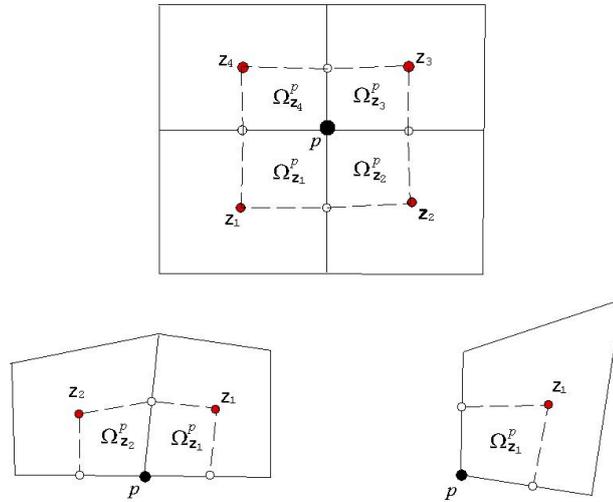


Figure 1: The configurations of control volume Ω_p : solid lines are the edges of the primal grid cells; dashed lines are the boundaries of control volumes; black solid circles are the vertexes of the primal grid denoted by p ; hollow circles are the midpoints of the primal related edges of p ; red solid circles are the center points of the primal related cells p . Top: the control volume for inner vertex p ; bottom left: the control volume for a boundary (not corner) vertex p ; bottom right: the control volume for a corner vertex p .

sub-cells in Ω_p . The configurations of the control volumes for three types are shown in Fig. 1.

3.2 Spatial discretization

In this subsection, we show the process of the space discretization. The space discretization is accomplished by using a vertex-centered finite volume method. For momentum conservation equation, we carry out the discretization on Ω_p , which is similar to other classical vertex-centered methods. For mass equation and energy equation, we discrete them separately on sub-cell $\Omega_{z_j}^p$ ($j=1, \dots, m$). There is unified momentum on Ω_p , and there are independent energy, density, pressure and specific heat coefficient on each $\Omega_{z_j}^p$. An integral ALE scheme of Eqs. (2.1) over Ω_p with arbitrary mesh velocity $\mathbf{V}^g = (u^g, v^g)$ is written in the following form:

$$\begin{cases} \frac{d}{dt} \int_{\Omega_{z_j}^p} \rho d\Omega_{z_j}^p + \int_{s_{z_j}^p} (\rho(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n}) ds = 0, \\ \frac{d}{dt} \int_{\Omega_p} (\rho \mathbf{V}) d\Omega_p + \int_{s_p} ((\rho \mathbf{V})(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n} + P \mathbf{n}) ds = 0, \\ \frac{d}{dt} \int_{\Omega_{z_j}^p} (\rho E) d\Omega_{z_j}^p + \int_{s_{z_j}^p} ((\rho E)(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n} + (P \mathbf{V}) \cdot \mathbf{n}) ds = 0, \end{cases} \quad (3.1)$$

where s_p and $s_{z_j}^p$ denote the boundaries of Ω_p and $\Omega_{z_j}^p$, respectively; $\mathbf{n} = (n_x, n_y)$ is the unit outward normal vector of the corresponding boundary. The mesh velocity $\mathbf{V}^g = (u^g, v^g)$ in Eqs. (3.1) is a function of time and position and it can be arbitrary. Eqs. (3.1) will evolve into the Eulerian form when $\mathbf{V}^g = \mathbf{0}$ and the Lagrangian form when $\mathbf{V}^g = \mathbf{V}$.

Then, we obtain the discrete scheme as follows:

$$\begin{cases} \frac{d}{dt}(\rho_{z_j}^p |\Omega_{z_j}^p|) + \int_{s_{z_j}^p} (\rho(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n}) ds = 0, \\ \frac{d}{dt}((\rho \mathbf{V})_p |\Omega_p|) + \int_{s_p} ((\rho \mathbf{V})(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n} + P \mathbf{n}) ds = 0, \\ \frac{d}{dt}((\rho E)_{z_j}^p |\Omega_{z_j}^p|) + \int_{s_{z_j}^p} ((\rho E)(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n} + (P \mathbf{V}) \cdot \mathbf{n}) ds = 0, \end{cases} \quad (3.2)$$

where, $\mathbf{U}_p = (\rho_{z_j}^p, (\rho \mathbf{V})_p, (\rho E)_{z_j}^p)$ is the vector of the integral average values of the unknowns for $\Omega_{z_j}^p$ or Ω_p ; $|\Omega_p|$ and $|\Omega_{z_j}^p|$ denote the areas of Ω_p and $\Omega_{z_j}^p$, respectively. Eqs. (3.2) can further evolve into:

$$\begin{cases} \frac{d}{dt}(\rho_{z_j}^p |\Omega_{z_j}^p|) = - \sum_{k=1}^4 \int_{(s_{z_j}^p)_k} (F_1(\mathbf{U}), G_1(\mathbf{U})) \cdot \mathbf{n} ds, \\ \frac{d}{dt}((\rho u)_p |\Omega_p|) = - \sum_{k=1}^{\tau} \int_{(s_p)_k} (F_2(\mathbf{U}), G_2(\mathbf{U})) \cdot \mathbf{n} ds, \\ \frac{d}{dt}((\rho v)_p |\Omega_p|) = - \sum_{k=1}^{\tau} \int_{(s_p)_k} (F_3(\mathbf{U}), G_3(\mathbf{U})) \cdot \mathbf{n} ds, \\ \frac{d}{dt}((\rho E)_{z_j}^p |\Omega_{z_j}^p|) = - \sum_{k=1}^4 \int_{(s_{z_j}^p)_k} (F_4(\mathbf{U}), G_4(\mathbf{U})) \cdot \mathbf{n} ds, \end{cases} \quad (3.3)$$

where, $(s_p)_k$ and $(s_{z_j}^p)_k$ denote the k th edge of Ω_p and $\Omega_{z_j}^p$, respectively; τ denotes the total number of the edges of Ω_p ;

$$F(\mathbf{U}) = \begin{pmatrix} F_1(\mathbf{U}) \\ F_2(\mathbf{U}) \\ F_3(\mathbf{U}) \\ F_4(\mathbf{U}) \end{pmatrix} = \begin{pmatrix} (\rho(u - u^g)) \\ (\rho u(u - u^g) + P) \\ (\rho v(u - u^g)) \\ (\rho E(u - u^g) + Pu) \end{pmatrix},$$

$$G(\mathbf{U}) = \begin{pmatrix} G_1(\mathbf{U}) \\ G_2(\mathbf{U}) \\ G_3(\mathbf{U}) \\ G_4(\mathbf{U}) \end{pmatrix} = \begin{pmatrix} (\rho(v - v^g)) \\ (\rho u(v - v^g)) \\ (\rho v(v - v^g) + P) \\ (\rho E(v - v^g) + Pv) \end{pmatrix},$$

$\int_{(s_p)_k} (F_2, G_2) \cdot \mathbf{n} ds$ and $\int_{(s_p)_k} (F_3, G_3) \cdot \mathbf{n} ds$ are the fluxes for the momentum on $(s_p)_k$; $\int_{(s_{z_j}^p)_k} (F_1, G_1) \cdot \mathbf{n} ds$ and $\int_{(s_{z_j}^p)_k} (F_4, G_4) \cdot \mathbf{n} ds$ are the fluxes for density and total energy on

$(s_{z_j}^p)_k$, respectively. In this paper, we choose the local L-F (Lax-Friedrichs) numerical flux to estimate approximately the fluxes, and obtain the following scheme:

$$\left\{ \begin{aligned} \frac{d}{dt}(\rho_{z_j}^p|\Omega_{z_j}^p|) &= - \sum_{k=1}^4 \int_{(s_{z_j}^p)_k} H_1(\mathbf{U}^+, \mathbf{U}^-) ds, \\ \frac{d}{dt}((\rho u)_p|\Omega_p|) &= - \sum_{k=1}^{\tau} \int_{(s_p)_k} H_2(\mathbf{U}^+, \mathbf{U}^-) ds, \\ \frac{d}{dt}((\rho v)_p|\Omega_p|) &= - \sum_{k=1}^{\tau} \int_{(s_p)_k} H_3(\mathbf{U}^+, \mathbf{U}^-) ds, \\ \frac{d}{dt}((\rho E)_{z_j}^p|\Omega_{z_j}^p|) &= - \sum_{k=1}^4 \int_{(s_{z_j}^p)_k} H_4(\mathbf{U}^+, \mathbf{U}^-) ds, \end{aligned} \right. \tag{3.4}$$

where $H_1(\mathbf{U}^+, \mathbf{U}^-)$ and $H_4(\mathbf{U}^+, \mathbf{U}^-)$ denote the the numerical fluxes on $(s_{z_j}^p)_k$, $H_2(\mathbf{U}^+, \mathbf{U}^-)$ and $H_3(\mathbf{U}^+, \mathbf{U}^-)$ denote the the numerical fluxes on $(s_p)_k$, and their expressions are presented as follows:

$$\left\{ \begin{aligned} H_1(\mathbf{U}^+, \mathbf{U}^-) &= \frac{1}{2} [(F_1(\mathbf{U}^+), G_1(\mathbf{U}^+)) \cdot \mathbf{n} + (F_1(\mathbf{U}^-), G_1(\mathbf{U}^-)) \cdot \mathbf{n} \\ &\quad - \beta(\mathbf{U}^+, \mathbf{U}^-)(\rho^+ - \rho^-)], \\ H_2(\mathbf{U}^+, \mathbf{U}^-) &= \frac{1}{2} [(F_2(\mathbf{U}^+), G_2(\mathbf{U}^+)) \cdot \mathbf{n} + (F_2(\mathbf{U}^-), G_2(\mathbf{U}^-)) \cdot \mathbf{n} \\ &\quad - \beta(\mathbf{U}^+, \mathbf{U}^-)((\rho u)^+ - (\rho u)^-)], \\ H_3(\mathbf{U}^+, \mathbf{U}^-) &= \frac{1}{2} [(F_3(\mathbf{U}^+), G_3(\mathbf{U}^+)) \cdot \mathbf{n} + (F_3(\mathbf{U}^-), G_3(\mathbf{U}^-)) \cdot \mathbf{n} \\ &\quad - \beta(\mathbf{U}^+, \mathbf{U}^-)((\rho v)^+ - (\rho v)^-)], \\ H_4(\mathbf{U}^+, \mathbf{U}^-) &= \frac{1}{2} [(F_4(\mathbf{U}^+), G_4(\mathbf{U}^+)) \cdot \mathbf{n} + (F_4(\mathbf{U}^-), G_4(\mathbf{U}^-)) \cdot \mathbf{n} \\ &\quad - \beta(\mathbf{U}^+, \mathbf{U}^-)((\rho E)^+ - (\rho E)^-)]. \end{aligned} \right. \tag{3.5}$$

\mathbf{U}^+ denotes the vector of conservative variables inside Ω_p (or $\Omega_{z_j}^p$), \mathbf{U}^- denotes the vector of conservative variables inside the control volume which shares the edge $(s_p)_k$ with Ω_p (or the sub-cell which shares the edge $(s_{z_j}^p)_k$ with $\Omega_{z_j}^p$), c^+ and c^- denote the values of the corresponding sound speed. $\beta(\mathbf{U}^+, \mathbf{U}^-)$ is an estimate of the largest absolute value of eigenvalues of Jacobian of the flux functions $(F(\mathbf{U})$ and $G(\mathbf{U}))$ with respect to \mathbf{U} , and we set the β to

$$\beta(\mathbf{U}^+, \mathbf{U}^-) = \max(|(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n} + c^+|, |(\mathbf{V} - \mathbf{V}^g) \cdot \mathbf{n} + c^-|)$$

in our scheme. The line integral in Eqs. (3.4) can be discretized by a four-point Gauss-

Lobatto integral formula, and Eqs. (3.4) become

$$\left\{ \begin{aligned} \frac{d}{dt}(\rho_{z_j}^p |\Omega_{z_j}^p|) &= - \sum_{k=1}^4 \sum_{r=1}^4 \omega_r H_1(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_{z_j}^p)_k|, \\ \frac{d}{dt}((\rho u)_p |\Omega_p|) &= - \sum_{k=1}^{\tau} \sum_{r=1}^4 \omega_r H_2(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_p)_k|, \\ \frac{d}{dt}((\rho v)_p |\Omega_p|) &= - \sum_{k=1}^{\tau} \sum_{r=1}^4 \omega_r H_3(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_p)_k|, \\ \frac{d}{dt}((\rho E)_{z_j}^p |\Omega_{z_j}^p|) &= - \sum_{k=1}^4 \sum_{r=1}^4 \omega_r H_4(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_{z_j}^p)_k|, \end{aligned} \right. \quad (3.6)$$

where $|(s_{z_j}^p)_k|$ and $|(s_p)_k|$ are the lengths of $(s_{z_j}^p)_k$ and $(s_p)_k$, respectively; G_r is the r th Gaussian quadrature point on the edge; ω_r is the corresponding Gaussian weight. In the Gauss-Lobatto integral formula, the following condition is used for implementing the calculation:

$$\begin{aligned} G_1 &= p_1, & G_2 &= \frac{1}{2}(p_1 + p_2) - \frac{\sqrt{5}}{10}(p_2 - p_1), \\ G_3 &= \frac{1}{2}(p_1 + p_2) + \frac{\sqrt{5}}{10}(p_2 - p_1), & G_4 &= p_2, \\ \omega_1 &= \omega_4 = \frac{1}{12}, & \omega_2 &= \omega_3 = \frac{5}{12}, \end{aligned}$$

for the edge with endpoints p_1 and p_2 .

Remark 3.1 (The special operations for the multi-material flows). For simplicity, we take the two-material flows as an example to show the special operations for the simulations of the multi-material flows in our scheme. We only consider the cases that the material interfaces are depicted by the edges of the primal grid cells and do not consider the ones that the material interfaces pass through the sub-cells. In the cases we consider, each sub-cell only contains one kind of material with its own ratio of specific heat. The equation of state is obtained as follows.

- Set the equation of state to $P = (\gamma_1 - 1)\rho\mathcal{E}$ in the sub-cells which only involve the material "1" with the ratio of specific heats γ_1 .
- Set the equation of state to $P = (\gamma_2 - 1)\rho\mathcal{E}$ in the sub-cells which only involve the material "2" with the ratio of specific heats γ_2 .

In order to prevent the exchange of different materials on the edge of the primal cell describing the interface and the generation of hybrid sub-cells, we set β to zero in the numerical fluxes which belong to the edge describing the material interfaces.

3.3 The velocity of the mesh

In this subsection, the process of obtaining the mesh velocity $\mathbf{V}^g = (u^g, v^g)$ is given as follows.

Taking control volume Ω_p for inner vertex p as an example, with

$$\mathbf{U}_p = (\rho_{z_j}^p, (\rho u)_p, (\rho v)_p, (\rho E)_{z_j}^p)$$

obtained by solving Eqs. (3.6), we can get $\mathbf{V}_p = (u_p, v_p)$, which is the average fluid velocity on Ω_p :

$$u_p = \frac{(\rho u)_p}{\rho_p}, \quad v_p = \frac{(\rho v)_p}{\rho_p},$$

where ρ_p is the integral average value of density on Ω_p derived from $\rho_{z_j}^p$ and $|\Omega_{z_j}^p|$ with the weighted average form:

$$\rho_p = \frac{\sum_{j=1}^m |\Omega_{z_j}^p| \rho_{z_j}^p}{|\Omega_p|}.$$

Then, \mathbf{V}_p is considered as the fluid velocity at vertex p and is selected as the mesh velocity at p , which makes the meshes move in a Lagrangian manner. For the high-order scheme, a WENO reconstruction for the velocity vector is used for obtaining the related polynomial, and then the mesh velocity at vertex p can be determined by the reconstructed polynomial.

A key point we should consider is how to obtain \mathbf{V}^g in the fluxes. With the fluid velocity at each vertex, we obtain \mathbf{V}^g at each node of Ω_p by using the approach which is similar to the one for obtaining the coordinates of the node. Then, \mathbf{V}^g in the fluxes can be determined, too. Taking a boundary segment l of Ω_p as an example, we assume that the two endpoints of l are p_1^l and p_2^l . An approximate fluid velocity at the target point will be chosen as \mathbf{V}^g in the fluxes. Taking the midpoint p_0^l of l as an example, we show the steps of obtaining the mesh velocity $\mathbf{V}^g(p_0^l)$ as follows.

Firstly, we let $\mathbf{n}_l = ((n_l)_x, (n_l)_y)$ denote the unit outward normal vector of l and let $(-(n_l)_y, (n_l)_x)$ denote the unit tangent vector of l , and then we take the projection in the tangent direction of the arithmetic average of $\mathbf{V}^g(p_1^l)$ and $\mathbf{V}^g(p_2^l)$ as the tangential component (denoted by w_t) of $\mathbf{V}^g(p_0^l)$. Secondly, we determine the normal component (denoted by w_n) of $\mathbf{V}^g(p_0^l)$. In [39], the HLLC approximate Riemann solver in the Eulerian form has been employed. In our scheme, the normal component of $\mathbf{V}^g(p_0^l)$ is set to the middle wave (also called the contact wave) velocity which is a by-product of the HLLC approximate Riemann solver in the Eulerian form, see [39]. The middle wave velocity S_* is

$$S_* = \frac{P^- - P^+ + \rho^+ q_{n_l}^+ (S^+ - q_{n_l}^+) - \rho^- q_{n_l}^- (S^- - q_{n_l}^-)}{\rho^+ (S^+ - q_{n_l}^+) - \rho^- (S^- - q_{n_l}^-)}, \quad (3.7)$$

where

$$\begin{aligned} S^- &= \min[q_{n_l}^- - c^-, \dot{\mathbf{V}} \cdot \mathbf{n}_l - \dot{c}], & S^+ &= \max[q_{n_l}^+ + c^+, \dot{\mathbf{V}} \cdot \mathbf{n}_l + \dot{c}], \\ q_{n_l}^- &= \mathbf{V}^- \cdot \mathbf{n}_l, & q_{n_l}^+ &= \mathbf{V}^+ \cdot \mathbf{n}_l. \end{aligned}$$

The items with superscripts “+” (or “-”) denote the values of the corresponding variables at l , and this definition is similar to the case in the local L-F (Lax-Friedrichs) numerical flux (3.5); \bar{V} and \bar{c} are the Roe’s average values of the fluid velocity and the speed of sound from two sides of l , respectively. Thirdly, we can get the x - and y - components (denoted by w_x and w_y , respectively) of $\mathbf{V}^s(p_0^l)$ with the relationship between w_x, w_y, w_n and w_t as follows:

$$w_x = w_n(n_l)_x - w_t(n_l)_y, \quad w_y = w_n(n_l)_y + w_t(n_l)_x.$$

3.4 The WENO reconstruction

For constructing the second-order scheme, we will use the second-order WENO reconstructions to obtain the corresponding reconstructions of polynomials for the variables. Here, we present two kinds of efficient WENO reconstructions. The reconstructions for different variables are separately considered as follows.

3.4.1 The reconstruction for $\rho, \rho E$

The process of the reconstruction for ρE and the one for ρ are the same, so we only show the one for ρ . Taking sub-cell $\Omega_{z_j}^p$ as an example and defining $e = \Omega_{z_j}^p$ for convenience, we construct a linear polynomial $\tilde{\rho}$ for ρ in e :

$$\tilde{\rho} = \rho_e + \tilde{\rho}_x(x - x_e) + \tilde{\rho}_y(y - y_e), \tag{3.8}$$

where (x_e, y_e) is the centroid of e :

$$x_e = \frac{1}{|e|} \int_e x de, \quad y_e = \frac{1}{|e|} \int_e y de,$$

$(\tilde{\rho}_x, \tilde{\rho}_y)$ denotes the gradient of ρ at the centroid of e ; ρ_e is the integral average value of ρ in e ; $|e|$ denotes the area of e ; $\tilde{\rho}$ maintains the average value of ρ in e . In order to evaluate the coefficients $\tilde{\rho}_x$ and $\tilde{\rho}_y$, we need to specify two conditions.

Firstly, we identify a number of admissible stencils S_1, S_2, \dots, S_8 for sub-cell e consisting of the neighboring sub-cells, such that e itself belongs to each stencil. For clarity, we show e and its eight adjacent sub-cells denoted by a, b, \dots, g in Fig. 2. Each stencil consists of three sub-cells. Taking stencil $S_1 = \{e, a, b\}$ as an example, we will introduce two specified conditions as follows:

$$\left\{ \begin{aligned} \frac{1}{|a|} \int_a \tilde{\rho} d(a) &= \rho_a, & (3.9a) \\ \frac{1}{|b|} \int_b \tilde{\rho} d(b) &= \rho_b, & (3.9b) \end{aligned} \right.$$

where $|a|$ and $|b|$ denote the areas of a and b , respectively. The other seven stencils are $\{e, b, c\}, \{e, c, f\}, \{e, f, i\}, \{e, i, h\}, \{e, h, g\}, \{e, g, d\}, \{e, d, a\}$. Secondly, we use the corresponding specified conditions to reconstruct the polynomial $\tilde{\rho}^k = \rho_e + \tilde{\rho}_x^k(x - x_e) + \tilde{\rho}_y^k(y -$

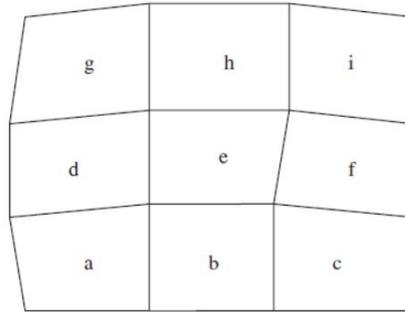


Figure 2: The configuration of sub-cell e and its eight adjacent sub-cells.

y_e) for each stencil S_k ($k=1, \dots, 8$). For sub-cell a , we denote: $x_{ae} = x_a - x_e$ and $y_{ae} = y_a - y_e$, where (x_a, y_a) is the centroid of a . With the following linear system:

$$\begin{cases} x_{ae}\tilde{\rho}_x^k + y_{ae}\tilde{\rho}_y^k = \rho_a - \rho_e, & (3.10a) \\ x_{be}\tilde{\rho}_x^k + y_{be}\tilde{\rho}_y^k = \rho_b - \rho_e, & (3.10b) \end{cases}$$

obtained by (3.9), we can get the two coefficients $\tilde{\rho}_x^k$ and $\tilde{\rho}_y^k$. Thirdly, we calculate the weight μ_k for each $\tilde{\rho}^k$:

$$\mu_k = \frac{\bar{\mu}_k}{\sum_{k=1}^8 \bar{\mu}_k},$$

where $\bar{\mu}_k = \frac{1}{(o_k + \sigma)^2}$, $\sigma = 10^{-6}$, and $o_k = [(\tilde{\rho}_x^k)^2 + (\tilde{\rho}_y^k)^2]|e|^2$ is an oscillation indicator [40] of the polynomial $\tilde{\rho}$. Finally, we can obtain the final reconstruction of polynomial: $\tilde{\rho} = \sum_{k=1}^8 \mu_k \tilde{\rho}^k$.

3.4.2 The reconstruction for $\mathbf{V} = (u, v)$

In this subsection, we reconstruct the polynomial for variable u (or v) on control volume Ω_p . The reconstruction process for variable u (or v) is similar to the one for ρ above, and we refer the reader to the Subsection 3.4.1 for details. The only key point we should notice is that the elements in the stencils of this reconstruction are control volumes, which is different from the process of the reconstruction in Subsection 3.4.1.

3.5 Time discretization

In this work, the Runge-Kutta (RK) method [41] is used for the time discretization of Eqs. (3.6). Since the mesh will change with time advancing, the information of each control volume should be updated at each RK stage. In this paper, the symbols with superscripts " n " and " $(n+1)$ " denote the values of the corresponding variables at the n th and $(n+1)$ th time steps, respectively; the symbols with subscript " p " denote the values of the corresponding variables at vertex p .

For keeping consistent with the accuracy of the spatial discretization, the time discretization in our scheme is carried out by the second-order RK method. With the values of all variables at the n th time step, the time marching is carried out as follows.

Step 1. The time steplength in our scheme is

$$\Delta t^n = \lambda \min_{p=1, \dots, \iota} (\Delta l_{z_j}^p / c_{z_j}^p)^n, \tag{3.11}$$

where ι is the total number of control volumes; $\Delta l_{z_j}^p$ is the shortest edge length of the sub-cell $\Omega_{z_j}^p$; $c_{z_j}^p$ is the average sound speed of $\Omega_{z_j}^p$. We let the Courant number λ be 0.3 unless otherwise stated in the following tests.

Step 2. With the approach in Subsection 3.3, the mesh velocity at vertex p denoted by $((u_p^s)^n, (v_p^s)^n)$ can be obtained. Then, the mesh vertex p is updated as follows:

$$x_p^{(1)} = x_p^n + (u_p^s)^n \Delta t^n, \quad y_p^{(1)} = y_p^n + (v_p^s)^n \Delta t^n. \tag{3.12}$$

The new Ω_p can be obtained, and the variables are updated as follows.

Case 1. For $\varphi_p = (\rho u)_p$ or $(\rho v)_p$:

$$|\Omega_p^{(1)}| \varphi_p^{(1)} = |\Omega_p^n| \varphi_p^n + L_p^n \Delta t^n, \tag{3.13}$$

where

$$L_p^n = \begin{cases} \left\{ -\sum_{k=1}^{\tau} \sum_{r=1}^4 \omega_r H_2(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_p)_k| \right\}^n, & \text{if } \varphi_p = (\rho u)_p, \\ \left\{ -\sum_{k=1}^{\tau} \sum_{r=1}^4 \omega_r H_3(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_p)_k| \right\}^n, & \text{if } \varphi_p = (\rho v)_p. \end{cases} \tag{3.14}$$

Case 2. For $\varphi_{z_j}^p = \rho_{z_j}^p$ or $(\rho E)_{z_j}^p$:

$$|(\Omega_{z_j}^p)^{(1)}| (\varphi_{z_j}^p)^{(1)} = |(\Omega_{z_j}^p)^n| (\varphi_{z_j}^p)^n + (L_{z_j}^p)^n \Delta t^n, \tag{3.15}$$

where

$$(L_{z_j}^p)^n = \begin{cases} \left\{ -\sum_{k=1}^4 \sum_{r=1}^4 \omega_r H_1(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_{z_j}^p)_k| \right\}^n, & \text{if } \varphi_{z_j}^p = \rho_{z_j}^p, \\ \left\{ -\sum_{k=1}^4 \sum_{r=1}^4 \omega_r H_4(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_{z_j}^p)_k| \right\}^n, & \text{if } \varphi_{z_j}^p = (\rho E)_{z_j}^p. \end{cases} \tag{3.16}$$

Step 3. With the variables obtained from **Step 2** and the approach in Subsection 3.3, the new vertex velocity $((u_p^s)^{(1)}, (v_p^s)^{(1)})$ can be determined. Then, the vertex p is updated as follows:

$$x_p^{n+1} = \frac{1}{2} x_p^n + \frac{1}{2} (x_p^{(1)} + (u_p^s)^{(1)} \Delta t^n), \quad y_p^{n+1} = \frac{1}{2} y_p^n + \frac{1}{2} (y_p^{(1)} + (v_p^s)^{(1)} \Delta t^n). \tag{3.17}$$

With the new Ω_p , the variables are updated as follows.

Case 1. For $\varphi_p = (\rho u)_p$ or $(\rho v)_p$:

$$|\Omega_p^{n+1}| \varphi_p^{n+1} = \frac{1}{2} |\Omega_p^n| \varphi_p^n + \frac{1}{2} (|\Omega_p^{(1)}| \varphi_p^{(1)} + L_p^{(1)} \Delta t^n), \quad (3.18)$$

where

$$L_p^{(1)} = \begin{cases} \left\{ -\sum_{k=1}^{\tau} \sum_{r=1}^4 \omega_r H_2(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_p)_k| \right\}^{(1)}, & \text{if } \varphi_p = (\rho u)_p, \\ \left\{ -\sum_{k=1}^{\tau} \sum_{r=1}^4 \omega_r H_3(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_p)_k| \right\}^{(1)}, & \text{if } \varphi_p = (\rho v)_p. \end{cases} \quad (3.19)$$

Case 2. For $\varphi_{z_j}^p = \rho_{z_j}^p$ or $(\rho E)_{z_j}^p$:

$$|(\Omega_{z_j}^p)^{n+1}| (\varphi_{z_j}^p)^{n+1} = \frac{1}{2} |(\Omega_{z_j}^p)^n| (\varphi_{z_j}^p)^n + \frac{1}{2} (|(\Omega_{z_j}^p)^{(1)}| (\varphi_{z_j}^p)^{(1)} + (L_{z_j}^p)^{(1)} \Delta t^n), \quad (3.20)$$

where

$$(L_{z_j}^p)^{(1)} = \begin{cases} \left\{ -\sum_{k=1}^4 \sum_{r=1}^4 \omega_r H_1(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_{z_j}^p)_k| \right\}^{(1)}, & \text{if } \varphi_{z_j}^p = \rho_{z_j}^p, \\ \left\{ -\sum_{k=1}^4 \sum_{r=1}^4 \omega_r H_4(\mathbf{U}^+(G_r), \mathbf{U}^-(G_r)) |(s_{z_j}^p)_k| \right\}^{(1)}, & \text{if } \varphi_{z_j}^p = (\rho E)_{z_j}^p. \end{cases} \quad (3.21)$$

4 Numerical examples

In this work, several tests are presented to demonstrate the accuracy and the performance of our scheme. It should be noted that our scheme can achieve the second-order accuracy in the regions of single-material flow, while it may not achieve the second-order accuracy near the material interfaces in the simulations of multi-material flows. We preliminarily extrapolate that, the approach for dealing with the material interfaces and the selection of the fluxes at the material interfaces may have some impacts on the accuracy of the spatial discretization. The detailed reason for the case mentioned above, the way of improving the accuracy, and the accuracy test for the multi-material flows still need to be studied in the future work. We need more time to consider these questions. In this work, we preliminarily provide a usable and concise algorithm for the simulations of the multi-material flows, and we only carry out the accuracy test for the single-material flow.

We show the results obtained by our scheme as follows. If there is no special explanation, all the boundary conditions are the wall conditions in the following test cases. All the simulations are run on a PC with one Inter Core i7(2.6GHz).

4.1 The accuracy test: Taylor-Green vortex problem

This problem is a vortical flow problem with an analytical solution [30]. The initial condition is

$$\begin{aligned} \rho &= 1, & P &= \frac{1}{4}[\sin(2\pi x) + \cos(2\pi y)] + 1, \\ u &= \sin(\pi x)\cos(\pi y), & v &= -\cos(\pi x)\sin(\pi y), \end{aligned}$$

with $\gamma = 1.4$ in the computational domain $(x, y) \in [0, 1] \times [0, 1]$. In the compressible inviscid case, an energy source term is used to maintain the steady state solution,

$$S_E = \frac{\pi}{4(\gamma - 1)} [\cos(3\pi x)\cos(\pi y) - \cos(3\pi y)\cos(\pi x)].$$

The initial computational mesh is Cartesian grid. The mesh and the pressure map at $t = 0.75$ for 40×40 cells are displayed in Fig. 3. The L^1 , L^2 , L^∞ errors and corresponding convergence orders of our scheme for density ρ , horizontal velocity u , and vertical velocity v at $t = 0.75$ are shown in Table 1, Table 2, and Table 3, respectively. It can be seen that our scheme achieves the anticipated accuracies.

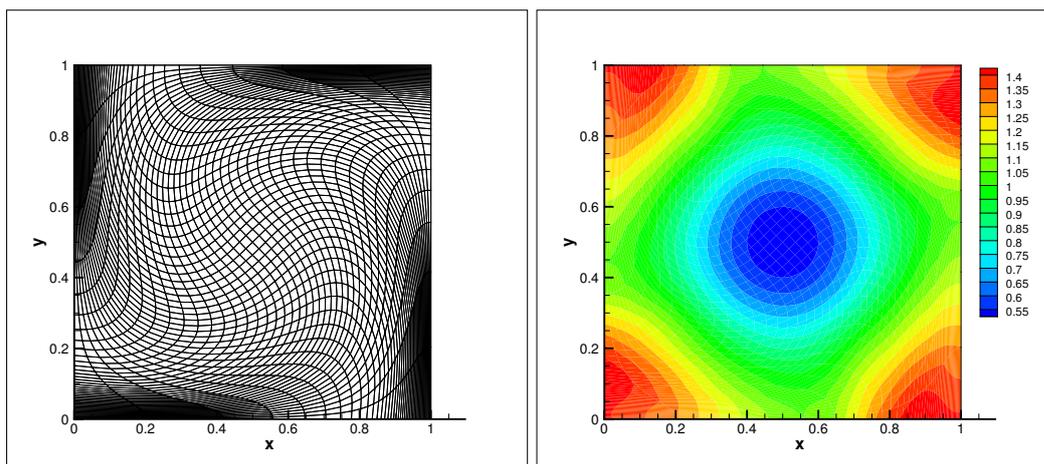


Figure 3: The result of the Taylor-Green vortex problem at $t = 0.75$ for 40×40 cells. Left: the mesh; right: the pressure map.

4.2 The Sod's shock tube problem in 2D

The Sod's shock tube problem is used for evaluating the shock-capturing and the non-oscillatory properties of our scheme. It consists of a shock tube of unity length. The left state of this case is high pressure fluid, and the right state is low pressure fluid. This problem can be regarded as the simulation of the one-dimensional Sod's shock tube problem on two-dimensional mesh with a final time $t = 0.2$, and the numerical results are shown by the scatter plots which involve all the control volumes.

Table 1: The errors and orders of the density ρ for the Taylor-Green vortex problem.

Elements	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	1.1266E-002		1.5267E-002		3.4852E-002	
20×20	4.2959E-003	1.3910	5.3321E-003	1.5177	1.9447E-002	0.8416
40×40	9.2677E-004	2.2126	1.2857E-003	2.0521	4.0315E-003	2.2701
80×80	1.7574E-004	2.3990	2.2293E-004	2.5278	7.5320E-004	2.4202
160×160	3.0235E-005	2.5388	3.7460E-005	2.5731	1.2910E-004	2.5444
320×320	6.4383E-006	2.2314	8.1038E-006	2.2087	2.8602E-005	2.1743

Table 2: The errors and orders of the horizontal velocity u for the Taylor-Green vortex problem.

Elements	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	2.8367E-002		3.5114E-002		9.1151E-002	
20×20	9.4389E-003	1.5875	1.2758E-002	1.4606	4.2251E-002	1.1092
40×40	2.2654E-003	2.0588	2.9621E-003	2.1067	1.1058E-002	1.9339
80×80	4.6581E-004	2.2820	4.8975E-004	2.5965	2.0659E-003	2.4202
160×160	7.6295E-005	2.6101	8.5523E-005	2.5176	3.4527E-004	2.5810
320×320	1.6389E-005	2.2189	2.1031E-005	2.0237	8.0548E-005	2.0998

Table 3: The errors and orders of the vertical velocity v for the Taylor-Green vortex problem.

Elements	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	3.0652E-002		3.8756E-002		8.8872E-002	
20×20	8.9426E-003	1.7772	1.3302E-002	1.5427	4.5861E-002	0.9545
40×40	2.3538E-003	1.9257	3.0865E-003	2.1075	1.0119E-002	2.1803
80×80	4.8794E-004	2.2702	5.1273E-004	2.5897	1.9378E-003	2.3845
160×160	7.5586E-005	2.6905	8.4285E-005	2.6049	3.2629E-004	2.5702
320×320	1.5958E-005	2.2438	1.9535E-005	2.1089	7.7502E-005	2.0738

4.2.1 The single-material Sod's shock tube problem

The initial condition of the single-material Sod's shock tube is

$$\begin{cases} (\rho_L, u_L, v_L, P_L) = (1.000, 0, 0, 1.0), & x \leq 0.5, \\ (\rho_R, u_R, v_R, P_R) = (0.125, 0, 0, 0.1), & x > 0.5, \end{cases}$$

with $\gamma = 1.4$. The partition of computational mesh is 200×5 in the domain $[0, 1] \times [0, 0.1]$. The numerical results of our scheme are given in Fig. 4. From the comparison of our results with the exact solutions, we can see that our scheme satisfies the essentially non-oscillatory property, and our results approximate the exact solutions well.

4.2.2 The multi-material Sod's shock tube problem

The second case is a multi-material variant of the Sod's shock tube problem [12]. At the initial time, the interface is located at $x = 0.5$. The left state is a high pressure fluid characterized by $(\rho_l, P_l, u_l, v_l) = (1.0, 2.0, 0, 0)$, the right state is a low pressure fluid defined by $(\rho_r, P_r, u_r, v_r) = (0.125, 0.1, 0, 0)$. The specific heat ratio on the left is defined as $\gamma_l = 2.0$,

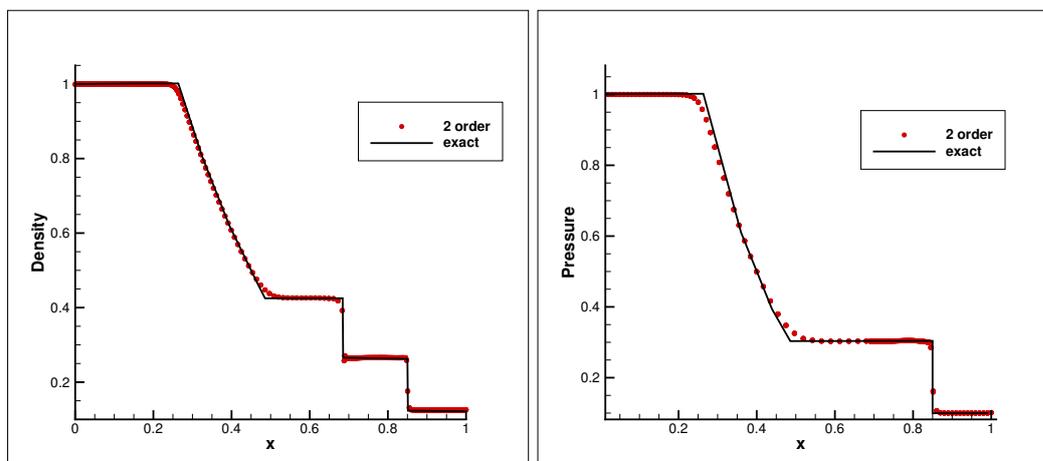


Figure 4: The single-material Sod's shock tube at $t=0.2$. Left: the density; right: the pressure.

and the one on the right is defined as $\gamma_r = 1.4$. The partition of computational mesh is 100×5 in the domain $[0,1] \times [0,0.1]$. Other conditions of this case are similar to the ones of the single-material Sod's shock tube above.

The numerical results of our scheme are given in Fig. 5. We can see that the elementary waves of the numerical results are resolved well. Under the precondition of simulating this case on the meshes with the similar sizes, it can be seen that our results are comparable to the ones obtained by the indirect cell-centred ALE scheme using the VOF interface reconstruction (or using the concentration equations) in [12] except for some slight oscillations and overshoots in the vicinity of discontinuities.

4.3 The Sedov problem

The Sedov problem [42] consists in the propagation of a high intensity cylindrical shock wave generated by a strong explosion. It is a classical test case for the Lagrangian schemes. The initial computational domain is set to $[0,1.21] \times [0,1.21]$ with a Cartesian grid whose initial uniform grid consists of 45×45 rectangular mesh elements. The initial density is unity and the initial velocity is zero. In the sub-cells included by the mesh element which contains the origin $O = (0,0)$, the values of the specific internal energy are 400, and the ones in other sub-cells are zero. The polytropic index γ is 1.4. When the time reaches $t = 1$, the analytical solution gives a shock at radius unity with a peak density of 6. Fig. 6 shows the mesh and the scatter plots of density as a function of the radius at $t = 1$, respectively. The density peak obtained by our scheme is 5.89.

We can notice that our numerical solution approximates the exact one well, and the mesh has good quality. The CPU time consumed in this simulation is 290 seconds. The computational efficiency of our scheme is lower than the one of the general vertex-centered schemes in the simulations of single-material flow, because of the additional computation in the sub-cells. Even so, the major advantage of our scheme is that it

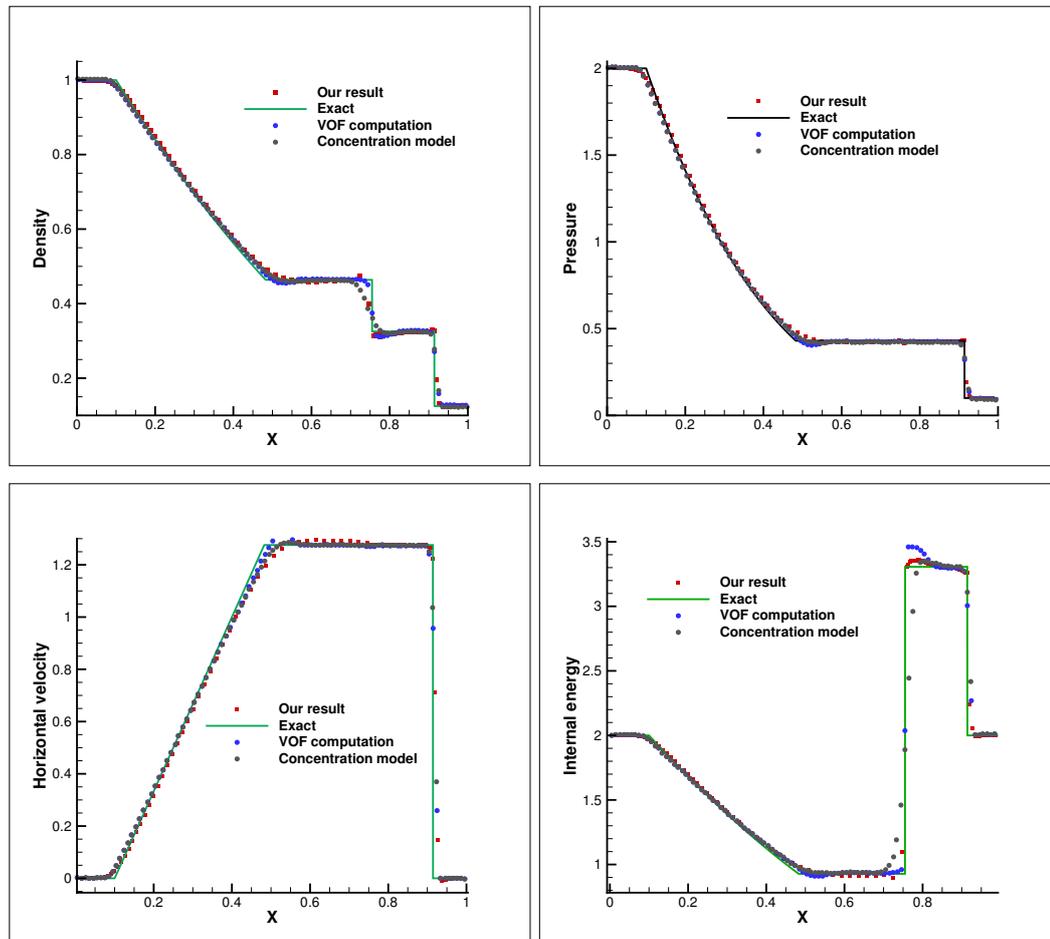


Figure 5: The multi-material Sod's shock problem at $t=0.2$. Top left: the density; top right: the pressure; bottom left: the horizontal velocity; bottom right: the internal energy.

can simulate some multi-material flows conveniently, and so we consider it is worth making some acceptable sacrifices in the computational efficiency of our scheme.

4.4 The Dukowicz problem

The Dukowicz problem is a two-dimensional shock refraction problem on an uneven mesh. The domain consists of two adjacent regions with different densities but equal pressure. Here, we carry out this simulation with regarding it as a two-material case. The left region has a 37×30 mesh partition with a normal left boundary and a right boundary aligned at 30° to the horizontal direction. The right region has a 53×30 mesh partition uniformly slanted at 30° to the horizontal direction, see Fig. 7. The initial conditions of two regions are $\rho_L = 1$, $u_L = 0$, $v_L = 0$, $P_L = 1$ and $\rho_R = 1.5$, $u_R = 0$, $v_R = 0$, $P_R = 1$, respectively. The upper and the lower boundaries are the reflective boundaries,

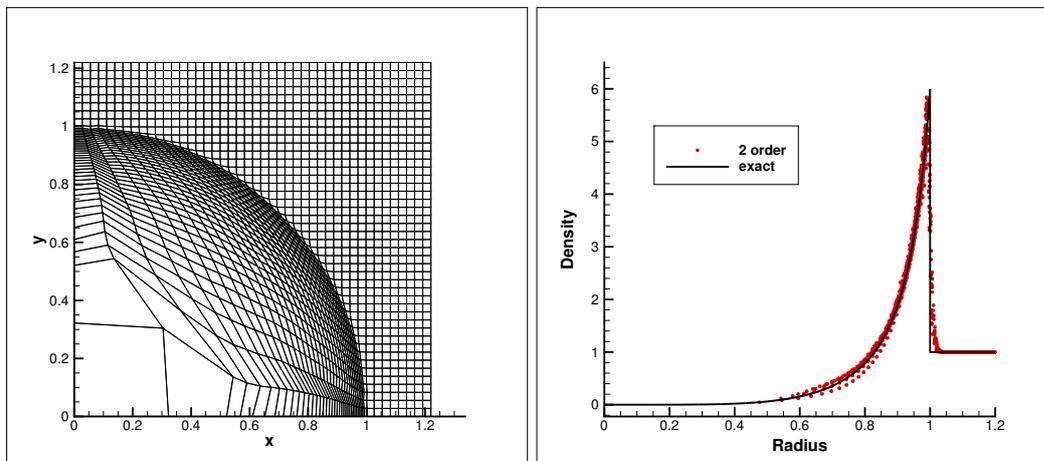


Figure 6: The Sedov problem at $t=1$. Left: the mesh; right: the scatter plots of density.

and the left boundary is a piston which moves from left with horizontal velocity 1.48. This problem runs to a time $t = 1.3$, just before the shock would leave the right region. Fig. 8 shows the result obtained by our scheme. The interface and the transmitted shocks can be observed clearly, and the reflective shock does not show up clearly due to the small difference in density across it.

By comparison, we can see that the density contours of our result are similar to the ones for reference in [43]. In addition, we consider the position of the interface on the lower boundary. The interface position of our result is located at 2.03. Table 4 gives the interface positions of the numerical results shown in [43]. Comparing the interface location of our result and the ones in Table 4, we can see that our interface location is within the scope of the ones in [43]. All the comparisons above show that our result is reasonable.

Table 4: The interface positions in [43].

Algorithm	Interface position
Edge viscosity	1.95
Symmetric tensor tangential viscosity	2.12
Nonsymmetric tensor tangential viscosity	2.20
Tensor normal viscosity	2.18

4.5 The Saltzman problem

The Saltzman test case describes the motion of a piston which is impinging on a fluid at rest contained in the initial computational domain $[0,1] \times [0,0.1]$. This is a well known challenging test problem [44] to validate the robustness of any Lagrangian scheme when the mesh is not aligned with fluid flow. The problem contains a rectangular

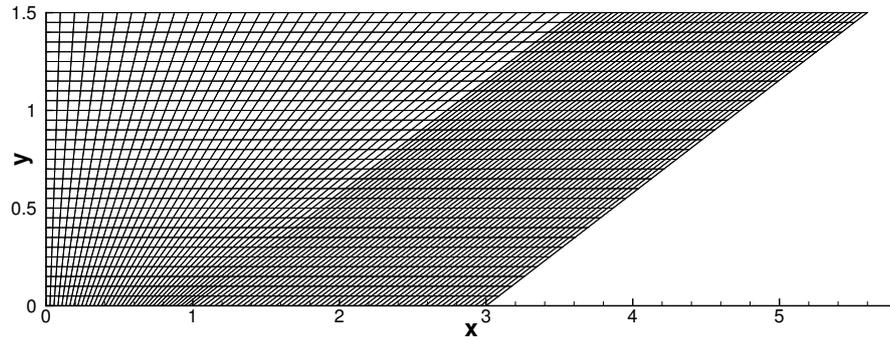


Figure 7: The initial mesh of the Dukowicz problem.

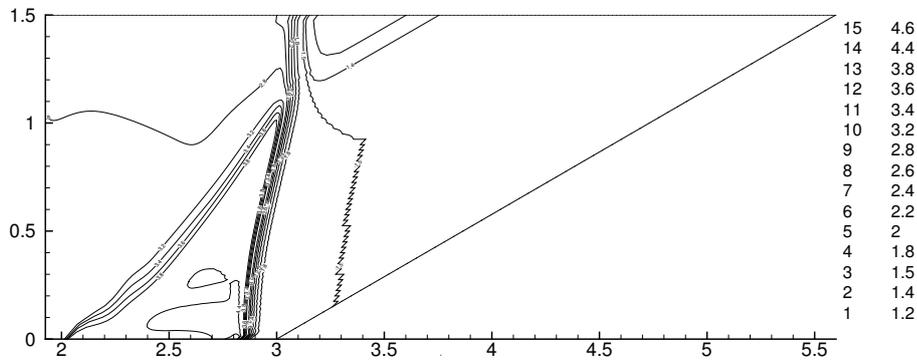
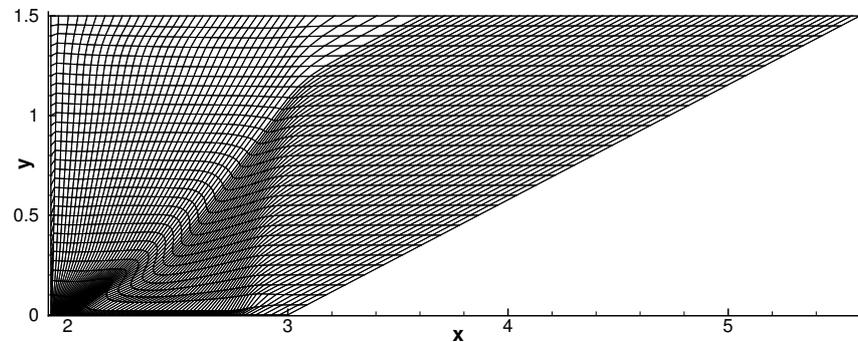


Figure 8: The Dukowicz problem at $t=1.3$. Top: the mesh; bottom: the density contours.

box whose left end is a piston. The piston moves with velocity $\mathbf{V}=(1,0)$ and a strong shock wave is generated which travels along the main direction of the computational domain. The elements near the piston are highly compressed when the piston moves. The computational mesh composed of 100×10 elements can be obtained with the steps as follows.

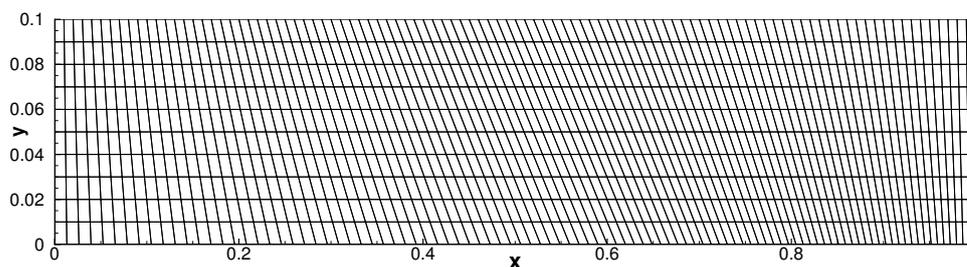


Figure 9: The initial mesh configuration of the Saltzman problem.

- Firstly, we build a Cartesian mesh with 100×10 square elements.
- Secondly, the uniform grid defined by coordinates (x, y) is skewed with the mapping:

$$x' = x + (0.1 - y)\sin(\pi x), \quad y' = y,$$

where x' and y' are the coordinates of the mesh points. The initial mesh is displayed in Fig. 9.

A moving slip wall boundary condition is chosen as the boundary condition of the the piston on left side of domain and the wall boundary conditions are imposed on the remaining boundaries. The working fluid is described by an ideal gas with $\gamma = 5/3$, and the initial condition \mathbf{Q}_0 is given by the state

$$\mathbf{Q}_0 = (\rho_0, u_0, v_0, (\rho E)_0) = (1, 0, 0, 10^{-4}).$$

The piston starts moving fast towards the fluid which is initially at rest. For maintaining stability, we set an initial Courant number to CFL=0.01 and increase it to 0.1 after $t = 0.01$, one can refer to [32] for details. Under these conditions, a strong shock wave travelling from left to right is generated. Then, more and more reflections of the shock occur on the right-hand wall and the piston until the time $t = 1$ is reached. The meshes and the density contours at $t = 0.925$ and $t = 0.98$ are shown in Fig. 10. At $t = 0.925$, the shock wave is at the location $x = 0.95$. Compared with the second-order results of the Lagrangian DG scheme in [29] at $t = 0.925$, our scheme captures the sharper shock wave and obtains the mesh with better quality. In addition, the scatter plots for density and horizontal velocity are shown in Fig. 11. It can be noticed that the scatter plots approximate the exact solution closely, which also shows that our scheme can preserve the symmetry of the solution.

Some cell-centered Lagrangian schemes fail at $t = 0.75$, because of nodes crossing. Other schemes based on finite volume method run a little longer, but they always fail at about $t = 0.93$, see [27]. The Lagrangian DG method in [32] runs longer which can reach $t = 0.9719$. In our scheme, we manage to make the simulation time reach a relatively large value. As can be seen in Fig. 10, we give the results at $t = 0.98$, which corresponds to several successive rebounds of the shock wave on the vertical boundaries of the domain. It is observed that our scheme preserves one-dimensional solution well except

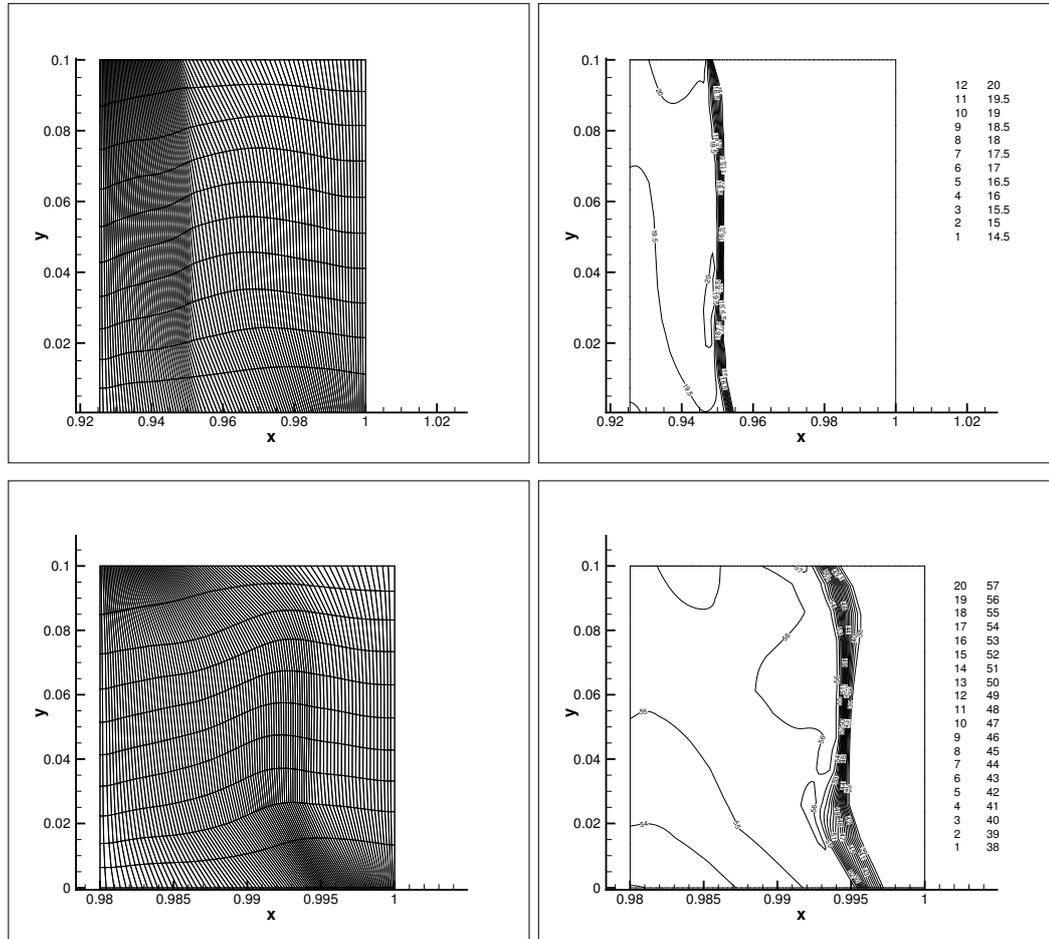


Figure 10: The Saltzman problem. Left: the mesh; right: the density contours; top: $t = 0.925$, bottom: $t = 0.98$.

for the regions near up and bottom wall boundaries. Finally, the robustness of our scheme is clearly demonstrated by this test case: this simulation can make the final time reach 0.99909, the meshes at $t = 0.99$ and $t = 0.99909$ given in Fig. 12 still have acceptable quality.

4.6 The triple point problem

The triple point problem which has significant vorticity, large shear, and complex interacting shocks is used to assess the property of simulating the multi-material flows of our scheme. This is a two-material problem which corresponds to a three states two-dimensional Riemann problem in a rectangular domain. The computational domain $[0,7] \times [0,3]$ is split into the following three sub-domains $\mathcal{D}_1 = [0,1] \times [0,3]$, $\mathcal{D}_2 = [1,7] \times [0,1.5]$ and $\mathcal{D}_3 = [1,7] \times [1.5,3]$. The sub-domain \mathcal{D}_1 contains a high-pressure and

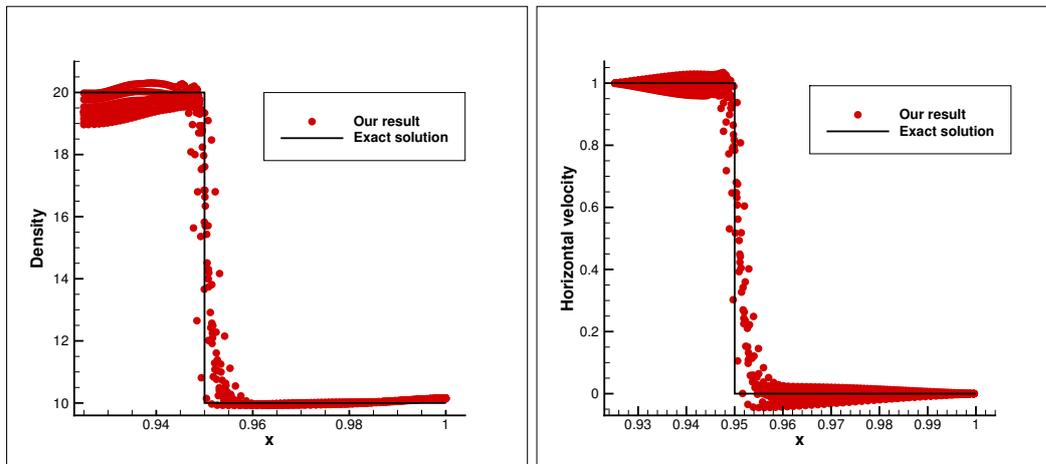


Figure 11: The scatter plots of the Saltzman problem at $t = 0.925$. Left: the density; right: the horizontal velocity.

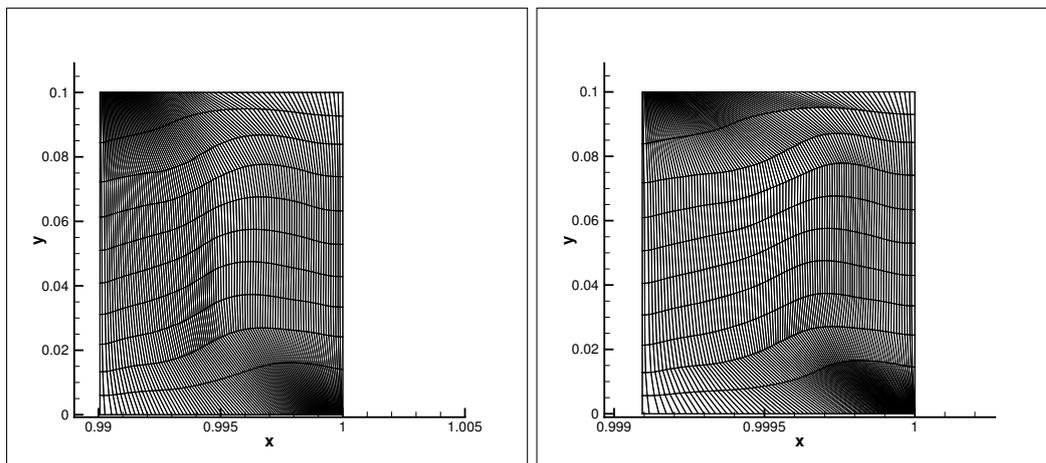


Figure 12: The meshes of the Saltzman problem. Left: $t = 0.99$; right: $t = 0.99909$.

high-density gas whose initial state is $(\rho_1, P_1, \mathbf{V}_1) = (1, 1, \mathbf{0})$. The sub-domain \mathcal{D}_2 contains a low-pressure and high-density gas whose initial state is $(\rho_2, P_2, \mathbf{V}_2) = (1, 0.1, \mathbf{0})$. The sub-domain \mathcal{D}_3 contains a low-pressure, low-density gas whose initial state is $(\rho_3, P_3, \mathbf{V}_3) = (0.125, 0.1, \mathbf{0})$. The sub-domains \mathcal{D}_1 and \mathcal{D}_3 are filled with the same material characterized by the polytropic index $\gamma_1 = \gamma_3 = 1.5$, whereas the sub-domain \mathcal{D}_2 is filled with a different material with $\gamma_2 = 1.4$. This simulation is carried out with mesh partition 70×30 . Due to the difference of the acoustic impedance, two shocks in \mathcal{D}_2 and \mathcal{D}_3 propagate with different speeds, which creates a strong shear along the initial contact discontinuity located at the interface between \mathcal{D}_2 and \mathcal{D}_3 . The shear produces a Kelvin-Helmholtz instability and a vortex formation occurs.

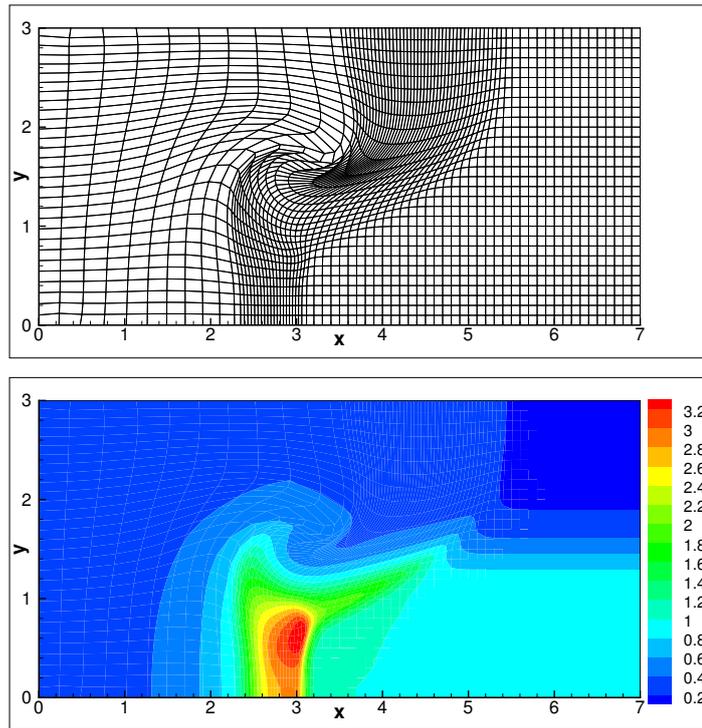


Figure 13: The triple point problem at $t=3.03$. Top: the mesh; bottom: the density.

This simulation fails at time $t = 3.03$, because of distorted mesh. The mesh and the density at $t=3.03$ are shown in Fig. 13. As illustrated, the result shows that our scheme has the primary ability to carry out the simulations of the multi-material flows which do not involve large deformations.

5 Conclusions

In this paper, we have proposed an ALE type vertex-centered finite volume method for two-dimensional compressible flow. In our method, the space discretization for momentum equation is carried out on the vertex control volume, and the ones for mass equation and energy equation are implemented on the sub-cells of the vertex control volume. The mesh velocity is given by the fluid velocity at vertex derived directly from the numerical solution. These operations make our scheme can simulate the multi-material flows which do not involve large deformations conveniently. Some numerical examples are presented to demonstrate the accuracy and the good performance of our scheme. Although the scope of the application of our scheme is currently restricted, it provides a concise and usable idea for the simulations of the multi-material flows. In the future work, we will consider combining other techniques (such as the interface capturing methods) with our scheme for simulating the single/multi-material cases

involving large deformations.

Acknowledgements

This work is supported by Natural Science Foundation of Guangdong province of China (Grant No. 2018A030310038), National Natural Science Foundation of China (Grant Nos. 11571002, 11772067, 11702028 and 12071046).

References

- [1] B. DESPRES AND C. MAZERAN, *Lagrangian gas dynamics in two dimensions and Lagrangian systems*, Arch. Rational Mech. Anal., 178(3) (2005), pp. 327–372.
- [2] P.-H. MAIRE, R. ABGRALL, J. BREIL, AND J. OVADIA, *A cell-centered Lagrangian scheme for two-dimensional compressible flow problems*, SIAM J. Sci. Comput., 29(4) (2007), pp. 1781–1824.
- [3] S. SAMBASIVAN, M. SHASHKOV, AND D. BURTON, *A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids*, Int. J. Numer. Methods Fluids, 72(7) (2013), pp. 770–810.
- [4] T. WU, M. SHASHKOV, N. MORGAN, D. KUZMIN, AND H. LUO, *An updated Lagrangian discontinuous Galerkin hydrodynamic method for gas dynamics*, Comput. Math. Appl., 78(2) (2019), pp. 258–273.
- [5] P. WOODWARD AND P. COLELLA, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54(1) (1984), pp. 115–173.
- [6] J. GLIMM, X. LI, Y. LIU, Z. XU, AND N. ZHAO, *Conservative front tracking with improved accuracy*, Siam J. Numer. Anal., 41(5) (2004), pp. 1926–1947.
- [7] C. HIRT, A. AMSDEN, AND J. COOK, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comput. Phys., 14(3) (1974), pp. 227–253.
- [8] D. BENSON, *An efficient, accurate, simple ALE method for nonlinear finite element programs*, Comput. Methods Appl. Mech. Eng., 72(3) (1989), pp. 305–350.
- [9] D. BENSON, *Computational methods in Lagrangian and Eulerian hydrocodes*, Comput. Methods Appl. Mech. Eng., 99 (1992), pp. 235–394.
- [10] L. MARGOLIN, *Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comput. Phys., 135 (1997), pp. 198–202.
- [11] P.-H. MAIRE, J. BREIL, AND S. GALERA, *A cell-centered arbitrary Lagrangian-Eulerian (ALE) method*, Int. J. Numer. Methods Fluids, 56(8) (2008), pp. 1161–1166.
- [12] S. GALERA, P.-H. MAIRE, AND J. BREIL, *A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction*, J. Comput. Phys., 229(16) (2010), pp. 5755–5787.
- [13] R. LOUBÈRE, P.-H. MAIRE, M. SHASHKOV, J. BREIL, AND S. GALERA, *ReALE: A reconnection-based arbitrary-Lagrangian-Eulerian method*, J. Comput. Phys., 229(12) (2010), pp. 4724–4761.
- [14] A. BARLOW, P.-H. MAIRE, W. RIDER, R. RIEBEN, AND M. SHASHKOV, *Arbitrary Lagrangian-Eulerian methods for modeling high-speed compressible multimaterial flows*, J. Comput. Phys., 322(C) (2016), pp. 603–665.
- [15] V. NGUYEN, *An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for simulations of flows over variable geometries*, J. Fluids Struct., 26(2) (2010), pp. 312–329.

- [16] X. XU, G. NI, AND J. SONG, *A high-order moving mesh kinetic scheme based on WENO reconstruction for compressible flows on unstructured meshes*, J. Sci. Comput., 57(2) (2013), pp. 278–299.
- [17] W. BOSCHERI AND R. LOUBÈRE, *High order accurate direct arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for non-conservative hyperbolic systems with stiff source terms*, Commun. Comput. Phys., 21(1) (2017), pp. 271–312.
- [18] E. GABURRO, M. DUMBSER, AND M. CASTRO, *Direct arbitrary-Lagrangian-Eulerian finite volume schemes on moving nonconforming unstructured meshes*, Comput. Fluids, 159(15) (2017), pp. 254–275.
- [19] X. ZHAO, X. YU, M. DUAN, F. QING, AND S. ZOU, *An arbitrary Lagrangian-Eulerian RKDG method for compressible Euler equations on unstructured meshes: Single-material flow*, J. Comput. Phys., 396 (2019), pp. 451–469.
- [20] M. WILKINS, *Use of artificial viscosity in multidimensional shock wave problems*, J. Comput. Phys., 36(3) (1980), pp. 281–303.
- [21] E. CARAMANA, D. BURTON, M. SHASHKOV, AND P. WHALEN, *The construction of compatible hydrodynamics algorithms utilizing conservation of total energy*, J. Comput. Phys., 146(1) (1998), pp. 227–262.
- [22] P.-H. MAIRE, R. LOUBÈRE, AND P. VACHAL, *Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme*, Commun. Comput. Phys., 10(04) (2011), pp. 940–978.
- [23] N. MORGAN, K. LIPNIKOV, D. BURTON, AND M. KENAMOND, *A Lagrangian staggered grid Godunov-like approach for hydrodynamics*, J. Comput. Phys., 259(4) (2014), pp. 568–597.
- [24] S. GODUNOV, A. ZABRODINE, M. IVANOV, A. KRAIKO, AND G. PROKOPOV, *Resolution numreque des problemes multidimensionnels de la dynamique des gaz*, Mir, 1979.
- [25] S. GODUNOV, *Reminiscences about difference schemes*, J. Comput. Phys., 153(1) (1989), pp. 6–25.
- [26] J. CHENG AND C.-W. SHU, *A high order ENO conservative Lagrangian type scheme for the compressible Euler equations*, J. Comput. Phys., 227(2) (2007), pp. 1567–1596.
- [27] P.-H. MAIRE, *A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes*, J. Comput. Phys., 228(7) (2009), pp. 2391–2425.
- [28] R. LOUBÈRE, J. OVADIA, AND R. ABGRALL, *A Lagrangian discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems*, Int. J. Numer. Methods Fluids, 44(6) (2004), pp. 645–663.
- [29] Z. JIA AND S. ZHANG, *A new high-order discontinuous Galerkin spectral finite element method for Lagrangian gas dynamics in two-dimensions*, J. Comput. Phys., 230(7) (2011), pp. 2496–2522.
- [30] F. VILAR, *Cell-centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics*, Comput. Fluids, 64 (2012), pp. 64–73.
- [31] F. VILAR, P.-H. MAIRE, AND R. ABGRALL, *A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids*, J. Comput. Phys., 276(276) (2014), pp. 188–234.
- [32] Z. LI, X. YU, AND Z. JIA, *The cell-centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two-dimensions*, Comput. Fluids, 96(11) (2014), pp. 152–164.
- [33] X. LIU, N. MORGAN, AND D. BURTON, *A Lagrangian discontinuous Galerkin hydrodynamic method*, Comput. Fluids, 163 (2018), pp. 68–85.
- [34] X. LIU, N. MORGAN, AND D. BURTON, *Lagrangian discontinuous Galerkin hydrodynamic methods in axisymmetric coordinates*, J. Comput. Phys., 373 (2018), pp. 253–283.
- [35] N. MORGAN, J. WALTZ, D. BURTON, M. CHAREST, T. CANFIELD, AND J. WOHLBIER, *A Godunov-like point-centered essentially Lagrangian hydrodynamic approach*, J. Comput. Phys.,

- 281 (2015), pp. 614–652.
- [36] N. MORGAN, J. WALTZ, D. BURTON, M. CHAREST, T. CANFIELD, AND J. WOHLBIER, *A point-centered arbitrary Lagrangian eulerian hydrodynamic approach for tetrahedral meshes*, J. Comput. Phys., 290 (2015), pp. 239–273.
- [37] M. AGUIRRE, A. GIL, J. BONET, AND C. LEE, *An upwind vertex centred finite volume solver for Lagrangian solid dynamics*, J. Comput. Phys., 300(C) (2015), pp. 387–422.
- [38] L. LIU, Z. SHEN, Q. ZENG, AND Z. JIA, *A high-order vertex-centered quasi-Lagrangian discontinuous Galerkin method for compressible Euler equations in two-dimensions*, Comput. Fluids, pending publication, 2020.
- [39] E. F. TORO, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, Springer Science & Business Media, 2013.
- [40] G. JIANG AND C.-W. SHU, *Efficient implementation of weighted eno schemes*, J. Comput. Phys., 126(1) (1996), pp. 202–228.
- [41] B. COCKBURN AND C.-W. SHU, *The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, J. Comput. Phys., 141(2) (1998), pp. 199–224.
- [42] L. SEDOV, *Similarity and Dimensional Methods in Mechanics*, New York: Academic Press, 1959.
- [43] J. CAMPBELL AND M. SHASHKOV, *A tensor artificial viscosity using a mimetic finite difference algorithm*, J. Comput. Phys., 172(2) (2001), pp. 739–765.
- [44] J. DUKOWICZ AND B. MELTZ, *Vorticity errors in multidimensional Lagrangian codes*, J. Comput. Phys., 99(1) (1992), pp. 115–134.