A high order positivity-preserving conservative WENO remapping method based on a moving mesh solver

Xiaolu Gu[1], Juan Cheng[2], Yue Li[3], and Chi-Wang Shu[4]

Abstract

In this paper, a high order accurate positivity-preserving conservative remapping algorithm is developed. Quadrilateral meshes in two dimensions are used as examples. This remapping method is based on the numerical solution of the trivial equation $\frac{\partial u}{\partial t} = 0$ on a moving mesh, which is the old mesh before remapping at $t = 0$ and is the new mesh after remapping at $t = T$. A high order finite volume scheme on the moving mesh is used to solve this problem. Specifically, we adopt the multi-resolution weighted essentially non-oscillatory (WENO) method for the spatial discretization and a strong stability preserving (SSP) Runge-Kutta method for the temporal discretization. The remapping algorithm is high order accurate under very mild smoothness requirement (Lipschitz continuity) on the mesh movement velocity, which can always be satisfied with a suitable choice of the final pseudo-time T. Furthermore, we design our remapping algorithm to have positivity-preserving property by using the linear scaling positivity-preserving limiter so that the algorithm could ensure the positivity-preserving property of relevant physical variables and maintain conservation and original order of accuracy. A series of numerical experiments are given to demonstrate the properties of our remapping algorithm such as high order accuracy, essentially non-oscillatory performance, positivity-preserving and high computational efficiency.

Keywords: Remapping, High order accuracy, Non-oscillatory, Positivity-preserving, Multi-resolution WENO.

[1]Graduate School, China Academy of Engineering Physics, Beijing 100088, China. E-mail: guxiaolu20@gscaep.ac.cn.

[2]Corresponding author. Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100088, China and HEDPS, Center for Applied Physics and Technology, and College of Engineering, Peking University, Beijing 100871, China. E-mail: cheng_juan@iapcm.ac.cn. Research is supported in part by NSFC grants 12031001 and 11871111, and CAEP Foundation No. CX20200026.

[3]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: yueli43@ustc.edu.cn.

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

1 Introduction

The Eulerian description and the Lagrangian description are two classical approaches in computational fluid dynamics (CFD). In a pioneering paper [10], Hirt et al. developed the arbitrary Lagrangian-Eulerian (ALE) framework, combining the best properties of the Lagrangian and Eulerian methods. We consider the indirect ALE approach which consists of three distinct stages: a Lagrangian stage, in which the solution and the computational mesh are updated; a rezoning stage, in which the nodes of the computational mesh are moved to more optimal positions to improve the quality of the mesh; and a remapping stage, in which the Lagrangian solution is transferred from the old distorted Lagrangian mesh to the new rezoned mesh.

The focus of this paper is on the last step of the indirect ALE method, i.e., the remapping step. The remapping algorithm is an essential and important component of the indirect ALE method since it must preserve important mathematical and physical properties of the Lagrangian solution. A good remapping method is required to be conservative, high order accurate, essentially non-oscillatory, positivity-preserving for certain physical variables and efficient. There are two classical classes of remapping methods, namely the intersection based remapping methods and the transport equation based (or flux based) remapping methods.

The most straightforward approach is remapping based on intersections [3,5,8,9,15,20,24], which involves computing the intersections of the old and new cells exactly. The geometrical intersection of a new cell and the old mesh can be computed by an exact polygon clipping algorithm [5,15] or the construction of a supermesh [8,24], for instance. The cell averages of the remapping variables in the new mesh can be obtained by the sum of integrated variables over the intersection regions. The intersection-based method can be used for remapping between two arbitrary meshes, but suffers from relatively high computational cost resulting from exactly calculating the overlaps between the old and new meshes.

The transport equation based (or flux-based) remapping method is another typical remapping strategy. This method has underlying assumptions that the old mesh and the new mesh

must have the same number of cells and the same mesh connectivity topology. In this approach, the remapping procedure can be expressed as a dynamic process governed by a linear transport equation. If the new mesh is constrained to be close enough to the old mesh, more precisely, if the movement of each node in the rezoning step does not exceed the size of its neighboring cells, the remapping values can be written in simple flux form [7,14,19,21–23,26]. That is, the values of the variables on each new cell can be set to those on the corresponding old cell plus terms which define the exchange of variables with nearest neighbors. For the more general case, that is, the size of the mesh displacement has no limitation, the remapping is typically carried out by solving a transport (or advection) equation [1, 2, 17, 18, 25]. The remapping results are obtained by marching from the old mesh to the new mesh over several pseudo time steps. This method avoids the precise calculation of intersecting regions between the cells of the old and new meshes. It is easy in coding and reduces complexity and computational cost.

In recent years, the field of high order remapping algorithms is under very active research since it is the basis of the high order indirect ALE method. Most remapping methods start with function reconstruction on the Lagrangian mesh while the high order reconstruction polynomials may produce numerical oscillations near the discontinuities. The classical technique to prevent numerical oscillation is to use some limiting strategies such as essentially non-oscillatory (ENO) method [5], weighted essentially non-oscillatory (WENO) method [15], slope limiter [27] or a posteriori limiting [3] in the reconstruction stage. The authors in [2, 17, 18] design the high order remapping algorithm by solving the advection equation with high accuracy in space and time. It should be noted that this method requires the mapping from the old mesh to the new mesh to be sufficiently smooth so as to achieve high order accuracy.

Another important issue for the remapping algorithm is the positivity-preserving property. The positivity of certain physical quantities such as density or internal energy must be preserved during the remapping process to avoid the failure of the numerical solution.

Margolin and Shashkov [22] proposed a second order, sign-preserving conservative interpolation for remapping inspired from the multidimensional positive definite advection transport algorithm (MPDATA). The authors in [14, 21, 23] utilized a conservative repair procedure re-distributing the remapping results in the neighboring cells to maintain local bounds. The remapping algorithm can also preserve local bounds by the flux corrected method, the core of which is the convex combination of the bound-preserving low order numerical fluxes and the high order fluxes [2, 19, 25, 26]. A posteriori multi-dimensional optimal order detection (MOOD) limiting added in the high order accurate remapping algorithm can lead to robustness and maintain intrinsic physical properties such as positivity [3]. Recently, Lei et al. [15] proposed a high order positivity-preserving conservative WENO remapping method based on intersections. The positivity-preserving limiter proposed by Zhang and Shu in [28] is utilized to maintain the positivity property of physical quantities without losing the conservation and the original high order accuracy.

In this paper, we will focus on designing a new remapping method, based on the numerical solution of the trivial equation $\frac{\partial u}{\partial t} = 0$ on a moving mesh, which is the old mesh before remapping at $t = 0$ and is the new mesh after remapping at $t = T$. Quadrilateral meshes in two dimensions are used as examples. A high order finite volume scheme on the moving mesh is used to solve this problem. In order to design a high order accurate and positivity-preserving remapping algorithm, first we will construct a high order accurate finite volume scheme to solve the trivial equation $\frac{\partial u}{\partial t} = 0$ on a moving mesh, based on the recently developed moving mesh finite volume technique [12, 13, 16]. An important feature of the moving mesh finite volume technique in [12, 13, 16] is its weak reliance on the smoothness of mesh movements: only Lipschitz continuity of the mesh movement is required to obtain high order accuracy, which can always be satisfied by choosing the ending pseudo-time T suitably. The spatial approximation of our scheme is based on the new multi-resolution WENO reconstruction procedure of Zhu and Shu [30], which constructs a series of unequal-sized hierarchical central spatial stencils and produces a polynomial on each cell. The linear weights of this

type of WENO scheme can be any fixed positive numbers on the condition that they sum to one. Compared with the traditional WENO reconstruction, this new WENO scheme uses fewer stencils and does not need to recalculate the linear weights at each time step, hence it is particularly suitable and efficient for the remapping problem where the shape of the mesh cells always changes. The temporal discretization is based on the high order strong stability preserving (SSP) Runge-Kutta method. We apply the Zhang-Shu positivity-preserving framework to maintain the positivity of certain variables such as density and internal energy by using a positivity-preserving limiter which is valid under suitable time step restriction. This method is widely adopted for its easy implementation and maintenance of high order accuracy.

Compared with the existing remapping algorithms, the novelty of our new algorithm is as follows. It does not require strong smoothness of the mesh movement for maintaining high order accuracy, and it can achieve high order accuracy under very mild conditions on the mesh movement. Specifically, only boundedness and Lipschitz continuity of the mesh movement velocity is required. This method does not limit the range of mesh movement and it can always be applied and yield high order accuracy as long as the final pseudo-time T is chosen suitably. When the new mesh is only a small perturbation of the old mesh, T can be chosen to be very small, requiring very few pseudo time steps for the moving mesh solver, hence saving computational cost. When the new mesh is very far from the old mesh, T has to be chosen suitably large to ensure Lipschitz continuity of the mesh movement, hence increasing the computational cost. Thus our remapping method is particularly advantageous for problems which involve small mesh changes, such as for ALE algorithms with remapping performed every time step or every few time steps. The positivity-preserving limiter is added to make the remapping algorithm have the positivity-preserving property without losing accuracy and conservation.

The organization of this paper is as follows. The main procedures of the remapping algorithm are given in Section 2, including four subsections: designing a finite volume scheme

to solve the trivial equation $\frac{\partial u}{\partial t} = 0$, where the old mesh is set at the initial time, the new mesh is set at the final time; determining the final pseudo-time T according to the boundedness and Lipschitz continuity requirements on the mesh movement velocity; using the multi-resolution WENO method for spatial discretization and high order SSP Runge-Kutta time discretization to obtain high order of accuracy in space and time and designing a positive-preserving remapping algorithm based on a positivity-preserving limiter. Section 3 will give several numerical tests to assess the performance of the algorithm on several types of moving meshes. Furthermore, we apply our remapping algorithm in an indirect ALE method and show its performance on certain benchmark flow problems, such as the Sedov, the Saltzman and the Noh problems. Finally, concluding remarks are given in Section 4.

2 Remapping algorithm

In this section, as an example, we design a third order positivity-preserving conservative WENO remapping method based on a moving mesh solver of the trivial equation $\frac{\partial u}{\partial t} = 0$. The input to the remapping problem includes the old distorted mesh, its corresponding physical quantities (their cell averages) and the new rezoned mesh. The goal of the remapping problem is to provide data (cell averages) on the new mesh which is transferred from the old mesh. We introduce a pseudo-time T and connect the corresponding nodes of the old and new meshes by straight lines to obtain a moving mesh. The remapping results can be obtained by solving the trivial equation $\frac{\partial u}{\partial t} = 0$ on this moving mesh. We only require boundedness and Lipschitz continuity of mesh movement velocity to obtain high order accuracy. The multi-resolution WENO method and the SSP Runge-Kutta method are used in the spatial discretization and time discretization respectively. Finally, the positive-preserving limiter is adopted so that we can obtain a high order accurate positivity-preserving conservative remapping algorithm.

2.1 The moving mesh algorithm in pseudo time

Now, we consider a connected bounded computational domain Ω in two-dimension. We regard the old mesh as the initial mesh configuration at the time $t = 0$ and denote it as $\mathcal{M}^0$. The new mesh is taken as the final mesh configuration at the final time $t = T$ and denoted as $\mathcal{M}^T$. The final time T is the pseudo time we introduce. For consistency with the discussion that follows, we use the following notation. The old mesh $\mathcal{M}^0$ consists of groups of quadrilateral cells $\{I^0_{i,j}\}, i = 1, \cdots, N_x + 1, j = 1, \cdots, N_y + 1$, where the superscript 0 means at $t = 0$, N_x and N_y are the number of cells in the x and y directions respectively. Each quadrilateral cell $\{I^0_{i,j}\}$ has four nodes $\left\{P^0_{i-\frac{1}{2},j-\frac{1}{2}}, P^0_{i+\frac{1}{2},j-\frac{1}{2}}, P^0_{i+\frac{1}{2},j+\frac{1}{2}}, P^0_{i-\frac{1}{2},j+\frac{1}{2}}\right\}$ and the coordinate of the node $P^0_{i-\frac{1}{2},j-\frac{1}{2}}$ is $\left(x^0_{i-\frac{1}{2},j-\frac{1}{2}}, y^0_{i-\frac{1}{2},j-\frac{1}{2}}\right)$. Similarly, replace the symbol superscript with T to get the elements, nodes and coordinates of the new mesh $\mathcal{M}^T$. Here we require the meshes $\mathcal{M}^0$ and $\mathcal{M}^T$ have the same number of nodes and cells. We use quadrilateral cells in the meshes as these are commonly used in Lagrangian methods.

We assume that there is a function u, such as density, momentum or total energy, defined on the computational domain, whose cell averages need to be remapped from the old mesh to the new mesh. The function u is a time independent quantity, namely the equation we are solving is the trivial equation

$$\frac{\partial u}{\partial t} = 0, \tag{2.1}$$

to be solved on a moving mesh connecting the old mesh at $t = 0$ and the new mesh at $t = T$.

We connect the corresponding nodes of the old and new grids by straight lines, to get a time-related control volume $I_{i,j}(t)$. Converting the Eq.(2.1) into the integral form which holds for any control volume

$$\iint\limits_{I_{i,j}(t)} \frac{\partial u}{\partial t} \mathrm{d}x\mathrm{d}y = 0,$$

by the Reynolds transport theorem and the divergence theorem, we can get the following

integral equation

$$\frac{\mathrm{d}}{\mathrm{d}t}S_{i,j}(t)\bar{u}_{i,j}(t)+\int\limits_{\partial I_{i,j}(t)}\boldsymbol{F}(\boldsymbol{\omega},u)\cdot\boldsymbol{n}\mathrm{d}l=0,\quad \boldsymbol{F}(\boldsymbol{\omega},u)=(-\omega_x u,-\omega_y u)^T. \tag{2.2}$$

Here, $S_{i,j}(t)$ is the area of $I_{i,j}(t)$, $\bar{u}_{i,j}(t)$ is the cell averages of the function u in the cell $I_{i,j}(t)$. $\boldsymbol{\omega}=(\omega_x,\omega_y)^T$ is the mesh movement velocity. ω_x and ω_y represent the mesh movement velocity in the x direction and in the y direction, respectively. $\boldsymbol{n}=(n_x,n_y)^T$ is the unit outward normal to the cell boundary.

2.2 Mesh movement velocity

In order to solve the Eq.(2.2), first we have to determine the mesh movement velocity $\boldsymbol{\omega}$. We introduce the pseudo-time T and connect the old node $P^0_{i-\frac{1}{2},j-\frac{1}{2}}$ and the new node $P^T_{i-\frac{1}{2},j-\frac{1}{2}}$ by a straight line, therefore $\boldsymbol{\omega}$ is a constant (independent of t) at this node and is given as

$$\omega_{x_{i-\frac{1}{2},j-\frac{1}{2}}}=\frac{x^T_{i-\frac{1}{2},j-\frac{1}{2}}-x^0_{i-\frac{1}{2},j-\frac{1}{2}}}{T},\quad \omega_{y_{i-\frac{1}{2},j-\frac{1}{2}}}=\frac{y^T_{i-\frac{1}{2},j-\frac{1}{2}}-y^0_{i-\frac{1}{2},j-\frac{1}{2}}}{T}. \tag{2.3}$$

We denote the points at the n-th time level as $\left\{\left(x^n_{i-\frac{1}{2},j-\frac{1}{2}},y^n_{i-\frac{1}{2},j-\frac{1}{2}}\right)\right\}$. According to the mesh movement velocity, we can have the points $P_{i-\frac{1}{2},j-\frac{1}{2}}(t)$ between the time level n (denoted as t_n) and the time level $n+1$ (denoted as t_{n+1})

$$\begin{aligned}
x_{i-\frac{1}{2},j-\frac{1}{2}}(t)&=x^n_{i-\frac{1}{2},j-\frac{1}{2}}+\omega_{x_{i-\frac{1}{2},j-\frac{1}{2}}}(t-t_n),\quad t\in[t_n,t_{n+1}],\\
y_{i-\frac{1}{2},j-\frac{1}{2}}(t)&=y^n_{i-\frac{1}{2},j-\frac{1}{2}}+\omega_{y_{i-\frac{1}{2},j-\frac{1}{2}}}(t-t_n),\quad t\in[t_n,t_{n+1}].
\end{aligned}$$

The authors pointed out in [12, 13] that the mesh movement velocity is a linear function on the edge connecting two nodes. For example, the mesh movement velocity $\boldsymbol{\omega}=(\omega_x,\omega_y)^T$ at any point $P=(x,y)$ on the edge connecting nodes $P_{i-\frac{1}{2},j-\frac{1}{2}}(t)$ and $P_{i+\frac{1}{2},j-\frac{1}{2}}(t)$ is as follows,

$$\begin{aligned}
\omega_x(x,y)&=\omega_{x_{i+\frac{1}{2},j-\frac{1}{2}}}\theta(x,y)+\omega_{x_{i-\frac{1}{2},j-\frac{1}{2}}}\left(1-\theta(x,y)\right),\\
\omega_y(x,y)&=\omega_{y_{i+\frac{1}{2},j-\frac{1}{2}}}\theta(x,y)+\omega_{y_{i-\frac{1}{2},j-\frac{1}{2}}}\left(1-\theta(x,y)\right),
\end{aligned} \tag{2.4}$$

where

$$\theta(x,y)=\frac{\sqrt{(x-x_{i-\frac{1}{2},j-\frac{1}{2}}(t))^2+(y-y_{i-\frac{1}{2},j-\frac{1}{2}}(t))^2}}{\sqrt{(x_{i+\frac{1}{2},j-\frac{1}{2}}(t)-x_{i-\frac{1}{2},j-\frac{1}{2}}(t))^2+(y_{i+\frac{1}{2},j-\frac{1}{2}}(t)-y_{i-\frac{1}{2},j-\frac{1}{2}}(t))^2}}.$$

In order to ensure the accuracy of the scheme on a moving mesh, the mesh movement velocity should satisfy the following properties [12, 13].

- The time-dependent cells $I_{i,j}(t)$ are well defined for $i=1,\cdots,N_x$, $j=1,\cdots,N_y$ and $t\in[t_n,t_{n+1}]$.

$$x_{i+\frac{1}{2},j-\frac{1}{2}}(t)-x_{i-\frac{1}{2},j-\frac{1}{2}}(t)=(\omega_{x_{i+\frac{1}{2},j-\frac{1}{2}}}-\omega_{x_{i-\frac{1}{2},j-\frac{1}{2}}})(t-t_n)+x^n_{i+\frac{1}{2},j-\frac{1}{2}}-x^n_{i-\frac{1}{2},j-\frac{1}{2}}>0,$$
$$y_{i-\frac{1}{2},j+\frac{1}{2}}(t)-y_{i-\frac{1}{2},j-\frac{1}{2}}(t)=(\omega_{y_{i-\frac{1}{2},j+\frac{1}{2}}}-\omega_{y_{i-\frac{1}{2},j-\frac{1}{2}}})(t-t_n)+y^n_{i-\frac{1}{2},j+\frac{1}{2}}-y^n_{i-\frac{1}{2},j-\frac{1}{2}}>0.$$

- Boundedness of mesh movement velocity, here C_0 is a constant independent of mesh size,

$$|\omega_{x_{i-\frac{1}{2},j-\frac{1}{2}}}|\leqslant C_0,\ |\omega_{y_{i-\frac{1}{2},j-\frac{1}{2}}}|\leqslant C_0.$$

- Lipschitz continuity of mesh movement velocity, here $C_{0,1}$ is a constant independent of mesh size,

$$\left|\frac{\partial\omega_x}{\partial x}\right|\leqslant C_{0,1},\quad \left|\frac{\partial\omega_y}{\partial x}\right|\leqslant C_{0,1},\quad \left|\frac{\partial\omega_x}{\partial y}\right|\leqslant C_{0,1},\quad \left|\frac{\partial\omega_y}{\partial y}\right|\leqslant C_{0,1}.$$

We take $C_0=C_{0,1}=10$ in numerical experiments. The requirements for the boundedness and Lipschitz continuity of mesh movement velocity can be transformed into the constraints of the final pseudo-time T. We can choose a suitable final pseudo-time T, then we have the mesh movement velocity $\boldsymbol{\omega}$ according to (2.3).

2.3 The finite volume scheme on the moving mesh

If the line integral in (2.2) is discretized by a Gauss-Lobatto integration formula on each edge, then we have

$$\int\limits_{\partial I_{i,j}(t)}\boldsymbol{F}(\boldsymbol{\omega},u)\cdot\boldsymbol{n}\mathrm{d}l\approx\sum_{k=1}^{K}l^k_{i,j}\sum_{q=1}^{q_N}\tilde{w}_q\boldsymbol{F}(\boldsymbol{\omega}(G^{(k)}_q(t)),u(G^{(k)}_q(t)))\cdot\boldsymbol{n}_k.$$

K is the number of the cell boundary edges, for the quadrilateral mesh $K = 4$. $l_{i,j}^k$ is the length of the k-th edge of $\partial I_{i,j}(t)$. q_N is the number of Gauss-Lobatto quadrature points which is determined by the order of accuracy of the scheme. For our third order scheme we take $q_N = 3$. $G_q^{(k)}(t)$ is the q-th Gauss-Lobatto quadrature point on the k-th edge of quadrilateral $I_{i,j}(t)$ and $\tilde{w}_q$ is the corresponding quadrature weight. $\boldsymbol{n}_k$ is the unit outward normal of $I_{i,j}(t)$ along the k-th cell edge. Notice that we have used the Gauss-Lobatto quadrature rule rather than the Gaussian quadrature rule, because this would then be easier to implement the positivity-preserving technique as described in Section 2.6.

$\boldsymbol{F}(\boldsymbol{\omega}(G_q^{(k)}(t)), u(G_q^{(k)}(t))) \cdot \boldsymbol{n}_k$ is approximated by a numerical flux. Here we choose the local Lax-Friedrichs flux,

$$\begin{aligned}
&\hat{\boldsymbol{F}}(\boldsymbol{\omega}(G_q^{(k)}(t)), u^-(G_q^{(k)}(t)), u^+(G_q^{(k)}(t))) \cdot \boldsymbol{n}_k \\
=&\frac{1}{2}\left[\left(\boldsymbol{F}(\boldsymbol{\omega}(G_q^{(k)}(t)), u^-(G_q^{(k)}(t))) + \boldsymbol{F}(\boldsymbol{\omega}(G_q^{(k)}(t)), u^+(G_q^{(k)}(t)))\right) \cdot \boldsymbol{n}_k\right. \\
&\left.-\alpha_{i,j}\left(u^+(G_q^{(k)}(t)) - u^-(G_q^{(k)}(t))\right)\right],
\end{aligned}$$

where

$$\alpha_{i,j} = \max_{G_q^{(k)}(t) \in I_{i,j}(t)} \left|\boldsymbol{\omega}(G_q^{(k)}(t)) \cdot \boldsymbol{n}_k\right|. \tag{2.5}$$

Here $\boldsymbol{\omega}(G_q^{(k)}(t))$ can be calculated in the same way as (2.4), $u^-(G_q^{(k)}(t))$ and $u^+(G_q^{(k)}(t))$ are the values of the reconstruction polynomial at the Gauss-Lobatto quadrature points inside and outside the cell $I_{i,j}(t)$ respectively.

Thus the semi-discrete finite volume scheme of (2.2) is given by

$$\frac{\mathrm{d}}{\mathrm{d}t} S_{i,j}(t)\bar{u}_{i,j}(t) = -\sum_{k=1}^{4} l_{i,j}^k \sum_{q=1}^{q_N} \tilde{w}_q \hat{\boldsymbol{F}}(\boldsymbol{\omega}(G_q^{(k)}(t)), u^-(G_q^{(k)}(t)), u^+(G_q^{(k)}(t))) \cdot \boldsymbol{n}_k. \tag{2.6}$$

We will use the multi-resolution WENO reconstruction procedure, described in detail in Section 2.4, to determine $u^-(G_q^{(k)}(t))$ and $u^+(G_q^{(k)}(t))$ in the spatial discretization. After that, we will further use the SSP Runge-Kutta time discretization described in Section 2.5 to obtain a high order scheme both in space and time.

2.4 Multi-resolution WENO reconstruction in the spatial discretization

In order to obtain a high order approximation to $u^-(G_q^{(k)}(t))$ and $u^+(G_q^{(k)}(t))$ in (2.6), we use the multi-resolution WENO reconstruction proposed by Zhu and Shu [30]. It selects a series of unequal-sized hierarchical central spatial stencils to construct high-order and low-order polynomials on these stencils respectively. The final reconstruction polynomial is a linear combination of high-order and low-order polynomials obtained with appropriate linear and nonlinear weights. This new type of multi-resolution WENO method can achieve optimal accuracy on the largest stencil in the smooth regions and is non-oscillatory near discontinuities. The linear weights of such WENO scheme can be any fixed positive numbers so this scheme is particularly suitable and easy to implement for irregular and moving meshes. This method can be applied to arbitrarily high order accuracy. We will design the third order scheme as an example in this paper and the procedure of the third order multi-resolution WENO reconstruction in two-dimension includes the following steps.

Step 1. We choose two central spatial stencils of different sizes. The small stencil is $T_1 = \{I_{i,j}\}$ and the large stencil is

$$T_2 = \{I_{i-1,j-1}, I_{i,j-1}, I_{i+1,j-1}, I_{i-1,j}, I_{i,j}, I_{i+1,j}, I_{i-1,j+1}, I_{i,j+1}, I_{i+1,j+1}\}.$$

We can reconstruct a zeroth degree polynomial $q_1(x, y)$ and a quadratic polynomial $q_2(x, y)$ on T_1 and T_2 respectively which satisfy

$$\iint_{I_{i,j}} q_s(x, y)\mathrm{d}x\mathrm{d}y = \bar{u}_{i,j}S_{i,j}, \quad s = 1, 2.$$

It is easy to get $q_1(x, y) = \bar{u}_{i,j}$. We set the second degree polynomial $q_2(x, y) = a_0 + a_1(x - x_{i,j}^c) + a_2(y - y_{i,j}^c) + a_3(x - x_{i,j}^c)^2 + a_4(x - x_{i,j}^c)(y - y_{i,j}^c) + a_5(y - y_{i,j}^c)^2$, where $(x_{i,j}^c, y_{i,j}^c)$ is the center of $I_{i,j}$. As a quadratic polynomial has six degrees of freedom, while the large stencil T_2 has nine cells, we adopt a constraint least-square procedure. The approximation

$q_2(x,y)$ is obtained as

$$q_2(x,y) = \arg\min_{\tilde{q}_2 \in P^2} \sum_{m=-1}^{1} \sum_{n=-1}^{1} \Big| \iint_{I_{i+m,j+n}} \tilde{q}_2(x,y)\mathrm{d}x\mathrm{d}y - \bar{u}_{i+m,j+n} S_{i+m,j+n} \Big|^2,$$
$$\text{s.t.} \iint_{I_{i,j}} \tilde{q}_2(x,y)\mathrm{d}x\mathrm{d}y = \bar{u}_{i,j} S_{i,j},$$

where P^2 is the set of polynomials of degree at most 2.

Step 2. We take $p_1(x,y) = q_1(x,y)$ and define a second degree polynomial $p_2(x,y)$ by

$$p_2(x,y) = \frac{1}{\gamma_2} q_2(x,y) - \frac{\gamma_1}{\gamma_2} p_1(x,y).$$

Here γ_1 and γ_2 are the linear weights and $\gamma_1 + \gamma_2 = 1$. If γ_2 is chosen larger, the eventual WENO polynomial will have a better accuracy in smooth regions yet may produce oscillations near strong discontinuities. Following [30], we choose $\gamma_1 = \frac{1}{11}, \gamma_2 = \frac{10}{11}$ as a good balance.

Step 3. Compute the smoothness indicators β_1 and β_2, which measure how smooth the function $p_1(x,y)$ and $p_2(x,y)$ are in the cell $I_{i,j}$ respectively.

$$\beta_2 = \sum_{\substack{l_1+l_2=1,2 \\ l_1,l_2=0,1,2}} \iint_{I_{i,j}} S_{i,j}^{|l_1+l_2|-1} \left(\frac{\partial^{l_1+l_2}}{\partial x^{l_1} \partial y^{l_2}} p_2(x,y) \right)^2 \mathrm{d}x\mathrm{d}y. \tag{2.7}$$

Since $p_1(x,y)$ is a zeroth degree polynomial, if we use (2.7) to determine β_1, we would get $\beta_1 = 0$. Zhu and Shu pointed out in [30] that this does not seem to cause any problems in the accuracy test, however it does lead to more smearing for problems containing strong shocks or contact discontinuities. We take an alternative definition of β_1 which is proposed in [30],

$$\beta_1 = \min\{\zeta_1 + \zeta_2,\, \zeta_2 + \zeta_3,\, \zeta_3 + \zeta_4,\, \zeta_4 + \zeta_1\},$$

where

$$\zeta_1 = (\bar{u}_{i,j} - \bar{u}_{i,j-1})^2, \quad \zeta_2 = (\bar{u}_{i,j} - \bar{u}_{i-1,j})^2,$$
$$\zeta_3 = (\bar{u}_{i,j} - \bar{u}_{i,j+1})^2, \quad \zeta_4 = (\bar{u}_{i,j} - \bar{u}_{i+1,j})^2.$$

Step 4. We compute the nonlinear weights based on the linear weights and the smoothness indicators, which follows the WENO-Z strategy as shown in [4]. The nonlinear weights

are given as

$$w_l = \frac{\bar{w}_l}{\bar{w}_1 + \bar{w}_2}, \quad l = 1, 2,$$

where

$$\bar{w}_l = \gamma_l \left(1 + \left(\frac{\tau}{\beta_l + \varepsilon_1} \right)^2 \right), \ \tau = |\beta_2 - \beta_1|, \quad l = 1, 2,$$

and ε_1 is taken as 10^{-3} in our code.

Step 5. The final reconstruction polynomial for the cell $I_{i,j}$ is given by

$$u_{i,j}(x, y) = w_1 p_1(x, y) + w_2 p_2(x, y).$$

Therefore, we can get a third order approximation to $u^-(G_q^{(k)}(t))$ at the Gauss-Lobatto integration points on the boundary of $I_{i,j}(t)$. Similarly, we can reconstruct the polynomials on the neighboring cells to obtain a third order approximation of $u^+(G_q^{(k)}(t))$.

2.5 High order SSP Runge-Kutta time discretization

To obtain high accuracy in time discretization for the scheme (2.6), the time marching is implemented by a high order SSP Runge-Kutta method. In this paper, we use the third order SSP Runge-Kutta method as follows

$$\begin{aligned}
&S_{i,j}^{n+1} \bar{u}_{i,j}^{(1)} = S_{i,j}^n \bar{u}_{i,j}^n + \triangle t L(u_{i,j}^n, t^n), \\
&\frac{1}{2}(S_{i,j}^n + S_{i,j}^{n+1}) \bar{u}_{i,j}^{(2)} = \frac{3}{4} S_{i,j}^n \bar{u}_{i,j}^n + \frac{1}{4}(S_{i,j}^{n+1} \bar{u}_{i,j}^{(1)} + \triangle t L(u_{i,j}^{(1)}, t^{n+1})), \\
&S_{i,j}^{n+1} \bar{u}_{i,j}^{n+1} = \frac{1}{3} S_{i,j}^n \bar{u}_{i,j}^n + \frac{2}{3}(\frac{1}{2}(S_{i,j}^n + S_{i,j}^{n+1}) \bar{u}_{i,j}^{(2)} + \triangle t L(u_{i,j}^{(2)}, \frac{1}{2}(t^n + t^{n+1}))),
\end{aligned} \tag{2.8}$$

where the operator L represents the terms at the right hand side of (2.6). Thus, up to now we have a fully discrete numerical scheme with third order accuracy both in space and time.

We calculate the time step $\triangle t$ by the CFL condition,

$$\triangle t = \frac{C_{cfl} h}{\alpha}, \quad \alpha = \max_{i,j} \alpha_{i,j}, \tag{2.9}$$

where C_{cfl} is the Courant number. In the computation we choose it as 0.8. h is the minimum diameter of the inscribed circles for all the quadrilateral cells in the computational mesh. The coefficient $\alpha_{i,j}$ is computed by (2.5).

In some cases α in (2.9) could be very small, leading to a very large $\triangle t$ determined by the stability constraint (2.9). In such cases, it is necessary to reduce $\triangle t$ in order to ensure temporal accuracy. Referring to [16], if $\triangle t$ determined by (2.9) is larger than $5h$ in our numerical tests, we will set it to be $5h$.

2.6 The positivity-preserving property

The indirect ALE method in computational fluid dynamics requires frequent remapping of conserved quantities such as density ρ, momentum $\boldsymbol{m}$ and total energy E from the old distorted Lagrangian mesh to a new rezoned mesh. Positivity is a very valuable property in the simulation of fluid flow since non-physical negative density ρ or internal energy $e = E - \frac{|\boldsymbol{m}|^2}{2\rho}$ can lead to an ill-posed system. During the remapping process, density and internal energy may become negative, which violates the physical properties. In this section we design our remapping algorithm to have the positivity-preserving property based on the positivity-preserving limiter valid under suitable time step restriction.

2.6.1 First order positivity-preserving remapping algorithm

Considering the variables in the fluid dynamics, we denote the vector of conservative variables $\boldsymbol{U} = (\rho, \boldsymbol{m}, E)^T$. $\bar{\boldsymbol{U}}_{i,j} = (\bar{\rho}_{i,j}, \bar{\boldsymbol{m}}_{i,j}, \bar{E}_{i,j})^T$ is the cell average of $\boldsymbol{U}$ in the cell $I_{i,j}$. We define the set of admissible states by

$$G = \left\{ \boldsymbol{U} = (\rho,\, \boldsymbol{m},\, E)^T,\ \rho > 0,\ e > 0 \right\}.$$

G can be proven to be a convex set. The scheme (2.8) is called positivity-preserving if $\{\bar{\boldsymbol{U}}^n_{i,j} \in G, i = 1, \cdots, N_x, j = 1, \cdots, N_y\}$ implies $\{\bar{\boldsymbol{U}}^{n+1}_{i,j} \in G, i = 1, \cdots, N_x, j = 1, \cdots, N_y\}$.

In order to make our high order finite volume scheme (2.8) positivity-preserving, we start with the first order finite volume scheme,

$$S^{n+1}_{i,j}\bar{\boldsymbol{U}}^{n+1}_{i,j} = S^n_{i,j}\bar{\boldsymbol{U}}^n_{i,j} - \triangle t \sum_{k=1}^{4} l^k_{i,j}\hat{\boldsymbol{F}}(\boldsymbol{\omega}(G^{(k)}_q(t_n)), \bar{\boldsymbol{U}}^n_{i,j}, \bar{\boldsymbol{U}}^{ext(I_{i,j})}_k) \cdot \boldsymbol{n}_k, \tag{2.10}$$

where $\bar{\boldsymbol{U}}_{i,j}^n$ is the cell average for the conserved variables in the cell $I_{i,j}$ and $\bar{\boldsymbol{U}}_k^{ext(I_{i,j})}$ is the cell average of $\boldsymbol{U}$ in the neighboring cell of $I_{i,j}$ along the k-th edge of cell $I_{i,j}$.

According to the proof in [6, 29], the following theorem can ensure the first order finite volume scheme is positivity-preserving.

Theorem 2.1. *The first order scheme* (2.10) *is positivity-preserving under the time step restriction*

$$\triangle t \leqslant \frac{\lambda}{\alpha} \min_{i,j} \left\{ \frac{S_{i,j}^n}{\sum_{k=1}^{4} l_{i,j}^k} \right\}, \ \lambda = \frac{1}{2}.$$

2.6.2 High order positivity-preserving remapping algorithm

Next, we derive the sufficient conditions to enable our high order remapping algorithm have the positive-preserving property. First we use the Euler forward method for time discretization and we get the fully discrete scheme as

$$S_{i,j}^{n+1}\bar{\boldsymbol{U}}_{i,j}^{n+1} = S_{i,j}^n\bar{\boldsymbol{U}}_{i,j}^n - \triangle t \sum_{k=1}^{4} l_{i,j}^k \sum_{q=1}^{q_N} \tilde{w}_q \hat{\boldsymbol{F}}(\boldsymbol{\omega}(G_q^{(k)}(t_n)), \boldsymbol{U}^-(G_q^{(k)}(t_n)), \boldsymbol{U}^+(G_q^{(k)}(t_n))) \cdot \boldsymbol{n}_k. \tag{2.11}$$

Motivated by the derivation in the previous papers [6, 28, 29], to find a sufficient condition for the scheme (2.11) to be positivity-preserving, we need to decompose the cell average $\bar{\boldsymbol{U}}_{i,j}^n$ by a Gauss quadrature rule that takes care of both area and line integrals, hence we would use Gauss-Lobatto quadrature rules. Such a quadrature for a general quadrilateral can be constructed by using a coordinate transformation. We transform the cells $I_{i,j}$ with the general quadrilateral shape in the $x-y$ coordinates to the square grid $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$ in $\xi - \eta$ coordinates. In order to make the Gauss-Lobatto quadrature to be exact for our reconstructed quadratic polynomial, we choose the 3×3-point tensor product Simpson quadrature rule where the quadrature points consist of the cell vertices, the middle points of the cell edges and the cell center. Fig. 1 shows the specific information of the quadrature points.

We define the set of Gauss-Lobatto quadrature points for the cell $I_{i,j}$ to be $G_{i,j} = \{(x_{m_1,m_2}, y_{m_1,m_2}), m_1, m_2 = 1, 2, 3\}$ and the quadrature weights $\tilde{w}_1 = \tilde{w}_3 = \frac{1}{6}, \tilde{w}_2 = \frac{2}{3}$. We

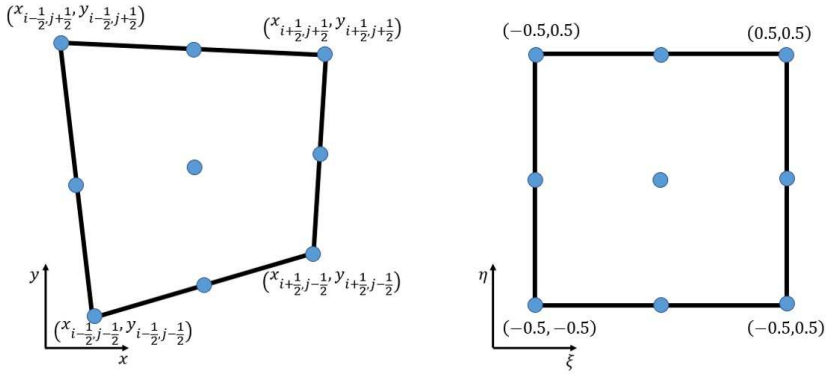


Figure 1: The transformation from $x-y$ coordinates to $\xi-\eta$ coordinates for the calculation of integration.

denote $\boldsymbol{U}_{i,j}^{m_1,m_2}$ to be the values of reconstruction polynomial $\boldsymbol{U}_{i,j}(x,y)$ for the cell $I_{i,j}$ at the corresponding Gauss-Lobatto quadrature points, namely $\boldsymbol{U}_{i,j}^{m_1,m_2} = \boldsymbol{U}_{i,j}(x_{m_1,m_2}, y_{m_1,m_2})$. For simplicity, we have another way to denote the values of $\boldsymbol{U}_{i,j}(x,y)$ at the Gauss-Lobatto quadrature points along the boundaries of the cell $I_{i,j}$, namely $\boldsymbol{U}_{1,m_1} = \boldsymbol{U}_{i,j}^{1,m_1}$, $\boldsymbol{U}_{2,m_1} = \boldsymbol{U}_{i,j}^{m_1,1}$, $\boldsymbol{U}_{3,m_1} = \boldsymbol{U}_{i,j}^{3,m_1}$, $\boldsymbol{U}_{4,m_1} = \boldsymbol{U}_{i,j}^{m_1,3}$. $\boldsymbol{\omega}$ with the same subscript means the value of the mesh movement velocity at the Gauss-Lobatto quadrature point on the corresponding edge. In the same way, the value of the reconstruction polynomial of adjacent cells can be defined. $J_{i,j}$ is the Jacobian for the coordinate transformation and $|J|_{i,j}^{m_1,m_2}$ are the values of $J_{i,j}$ at the corresponding Gauss-Lobatto quadrature points.

By some algebraic manipulations, $\bar{\boldsymbol{U}}_{i,j}^{n+1}$ can be rewritten as

$$\begin{aligned} S_{i,j}^{n+1}\bar{\boldsymbol{U}}_{i,j}^{n+1} =& \frac{1}{2}\left[\sum_{m_1=1}^{3}\tilde{w}_2\tilde{w}_{m_1}|J|_{i,j}^{2,m_1}\boldsymbol{U}_{i,j}^{2,m_1} + \sum_{m_1=1}^{3}\tilde{w}_{m_1}\tilde{w}_2|J|_{i,j}^{m_1,2}\boldsymbol{U}_{i,j}^{m_1,2}\right] \\ &+ \frac{\tilde{w}_1}{2}\sum_{m_1=1}^{3}\tilde{w}_{m_1}\left[\hat{\mathcal{F}}_{m_1}^{1} + \hat{\mathcal{F}}_{m_1}^{2} + \hat{\mathcal{F}}_{m_1}^{3} + \hat{\mathcal{F}}_{m_1}^{4}\right], \end{aligned} \tag{2.12}$$

where

$$\hat{\mathcal{F}}^1_{m_1} = |J|^{1,m_1}_{i,j}\boldsymbol{U}_{1,m_1} - \frac{2\triangle t}{\tilde{w}_1}\Big[l^1_{i,j}\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},(\boldsymbol{U}_{1,m_1})^+\right)\cdot\boldsymbol{n}_1 + l^2_{i,j}\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},\boldsymbol{U}_{2,m_1}\right)\cdot\boldsymbol{n}_2$$
$$+l^3_{i,j}\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},\boldsymbol{U}_{3,m_1}\right)\cdot\boldsymbol{n}_3 + l^4_{i,j}\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},\boldsymbol{U}_{4,m_1}\right)\cdot\boldsymbol{n}_4\Big],$$
$$\hat{\mathcal{F}}^2_{m_1} = |J|^{m_1,1}_{i,j}\boldsymbol{U}_{2,m_1} - \frac{2l^2_{i,j}\triangle t}{\tilde{w}_1}\Big[\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{2,m_1},\boldsymbol{U}_{2,m_1},(\boldsymbol{U}_{2,m_1})^+\right)\cdot\boldsymbol{n}_2 - \hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},\boldsymbol{U}_{2,m_1}\right)\cdot\boldsymbol{n}_2\Big],$$
$$\hat{\mathcal{F}}^3_{m_1} = |J|^{3,m_1}_{i,j}\boldsymbol{U}_{3,m_1} - \frac{2l^3_{i,j}\triangle t}{\tilde{w}_1}\Big[\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{3,m_1},\boldsymbol{U}_{3,m_1},(\boldsymbol{U}_{3,m_1})^+\right)\cdot\boldsymbol{n}_3 - \hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},\boldsymbol{U}_{3,m_1}\right)\cdot\boldsymbol{n}_3\Big],$$
$$\hat{\mathcal{F}}^4_{m_1} = |J|^{m_1,3}_{i,j}\boldsymbol{U}_{4,m_1} - \frac{2l^4_{i,j}\triangle t}{\tilde{w}_1}\Big[\hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{4,m_1},\boldsymbol{U}_{4,m_1},(\boldsymbol{U}_{4,m_1})^+\right)\cdot\boldsymbol{n}_4 - \hat{\boldsymbol{F}}\left(\boldsymbol{\omega}_{1,m_1},\boldsymbol{U}_{1,m_1},\boldsymbol{U}_{4,m_1}\right)\cdot\boldsymbol{n}_4\Big].$$

$\hat{\mathcal{F}}^1_{m_1}$ is a two-dimensional first order positivity-preserving finite volume scheme which has the same type as (2.10). $\hat{\mathcal{F}}^2_{m_1}$, $\hat{\mathcal{F}}^3_{m_1}$ and $\hat{\mathcal{F}}^4_{m_1}$ are one-dimensional first order positivity-preserving finite volume schemes. So $\bar{\boldsymbol{U}}^{n+1}_{i,j}$ is the convex combination of $\boldsymbol{U}^{m_1,m_2}_{i,j}$ and first order finite volume schemes. If all the terms at the right hand side of (2.12) are in the set G, then it is obvious that $\bar{\boldsymbol{U}}^{n+1}_{i,j} \in G$ since G is convex. We can enforce a linear scaling positivity-preserving limiter so that $\boldsymbol{U}^{m_1,m_2}_{i,j} \in G$. The first order finite volume scheme can be positivity-preserving under a suitable time step condition. More detailed proof can be referred to [6, 29].

In summary, for solving a linear partial differential equation by a finite volume scheme with the local Lax-Friedrichs flux on a general quadrilateral moving mesh, in order to obtain a high order positive-preserving scheme, we have the following theorem.

Theorem 2.2. *Assume* $\bar{\boldsymbol{U}}^n_{i,j} \in G$ *and* $\boldsymbol{U}^{m_1,m_2}_{i,j} \in G$ *for all* $i = 1,\cdots,N_x$, $j = 1,\cdots N_y$, $m_1, m_2 = 1,2,3$, *then the scheme* (2.11) *is positivity-preserving under the time step restriction*

$$\triangle t \leqslant \frac{\tilde{w}_1}{2\alpha}\lambda \min_{i,j}\left\{\frac{|J|_{i,j}}{\sum_{k=1}^4 l^k_{i,j}}\right\}. \tag{2.13}$$

where $\tilde{w}_1 = \frac{1}{6}$, $\lambda = \frac{1}{2}$, $|J|_{i,j} = \min\limits_{m_1=1,2,3}\{|J|^{m_1,1}_{i,j}, |J|^{m_1,3}_{i,j}, |J|^{1,m_1}_{i,j}, |J|^{3,m_1}_{i,j}\}$ *and the coefficient* α *is the same as* (2.9).

The SSP high order Runge-Kutta scheme (2.8) will keep the positivity since they are the convex combinations of Euler forward time discretization and G is convex.

Remark. The above theorem theoretically proves that the high order finite volume scheme has the positivity-preserving property under the time step restriction (2.13). As we know the smaller time step will lead to higher computational cost. In our numerical experiments, we take another approach to ensure the positivity-preserving property of the our high order remapping algorithm. This approach obeys the standard CFL condition first, namely (2.9), which is more relaxed than the time step (2.13). In this approach we check at each time step whether all new cell averages belong to the set G. If yes, we continue the computation, otherwise we need to return to the previous time step and march time with $\triangle t/2$. Compared with the method with the time step restriction (2.13), this implementation approach is more efficient. Theorem 2.2 makes sure that we just need to return only a finite number of times to have the cell averages belonging to the set G.

2.6.3 The positivity-preserving limiter

The scheme (2.8) cannot automatically satisfy the sufficient condition $\boldsymbol{U}_{i,j}^{m_1,m_2} \in G$, it can be enforced by the linear scaling positivity-preserving limiter proposed in [28]. The idea of the linear scaling positivity-preserving limiter is to perform a linear compression on the original reconstruction polynomial $\boldsymbol{U}_{i,j}(x,y)$ to modify it into a new polynomial $\tilde{\boldsymbol{U}}_{i,j}(x,y)$. The modified polynomial $\tilde{\boldsymbol{U}}_{i,j}(x,y)$ satisfies that its values at all Gauss-Lobatto quadrature points are positivity-preserving. The specific implementation can be taken as follows.

The first step is to enforce the positivity of density. We modify the reconstructed quadratic polynomial $\rho_{i,j}(x,y)$ by

$$\hat{\rho}_{i,j}(x,y) = \theta_1(\rho_{i,j}(x,y) - \bar{\rho}_{i,j}) + \bar{\rho}_{i,j},$$
$$\theta_1 = \min\{1, |\frac{\bar{\rho}_{i,j} - \varepsilon_2}{\bar{\rho}_{i,j} - b}|\},\ b = \min_{(x,y)\in G_{i,j}} \rho_{i,j}(x,y),$$

where $G_{i,j}$ is a set of Gauss-Lobatto quadrature points in $I_{i,j}$, ε_2 is a very small positive constant which satisfies $\bar{\rho}_{i,j} \geqslant \varepsilon_2$ for all i,j. For example, we take $\varepsilon_2 = 10^{-13}$ in our code.

The second step is to enforce the positivity of the internal energy e for the cells. Define $\hat{\boldsymbol{U}}_{i,j}(x,y) = (\hat{\rho}_{i,j}(x,y), \boldsymbol{m}_{i,j}(x,y), E_{i,j}(x,y))^T$ after the first step. Then we get the limited

polynomial

$$\tilde{\boldsymbol{U}}_{ij}(x,y)=\theta_2(\hat{\boldsymbol{U}}_{i,j}(x,y)-\bar{\boldsymbol{U}}_{i,j})+\bar{\boldsymbol{U}}_{i,j},\quad \theta_2=\min_{(x,y)\in G_{i,j}}\frac{e(\bar{\boldsymbol{U}}_{i,j})}{e(\bar{\boldsymbol{U}}_{i,j})-e(\hat{\boldsymbol{U}}_{i,j}(x,y))}.$$

It is easy to show that the cell average of the limited polynomial $\tilde{\boldsymbol{U}}_{i,j}(x,y)$ over $I_{i,j}$ is still $\bar{\boldsymbol{U}}_{i,j}$ and $\tilde{\boldsymbol{U}}_{i,j}^{m_1,m_2}\in G_{i,j}$. More importantly, this limiter will not destroy conservation and accuracy, its detailed proof can be found in [28]. Thus this positivity-preserving limiter can keep conservation, accuracy and positivity.

So far, the main procedures of the high order positivity-preserving conservative remapping algorithm are accomplished. In the next section, we will verify the accuracy, non-oscillatory and positivity-preserving properties of our remapping algorithm through a series of numerical examples. Finally, we apply our remapping algorithm in an indirect ALE method and show its performance on certain benchmark flow problems, such as the Sedov, Saltzman and Noh problems.

3 Numerical examples

In this section, we provide a series of numerical examples to demonstrate the conservative, high order accurate, non-oscillatory, and positivity-preserving properties of the remapping algorithm. For the sake of simplicity, we set the computational area Ω as $[0,1]\times[0,1]$. Suppose $\mathcal{M}^0$ is a uniform mesh with the step size $h_x=\frac{1}{N_x}$ along the x axes and the step size $h_y=\frac{1}{N_y}$ along the y axes, we can define the vertex $(x_i,y_j)=((i-1)h_x,(j-1)h_y)$. We have the following two ways to get a distorted mesh. Note that here the old and new meshes have the same number of nodes and cells. Besides, we require that the boundary points of the old and new meshes remain the same for simplicity.

1. The smoothly moving mesh $\mathcal{M}_S^n$

$$x^n_{i-\frac{1}{2},j-\frac{1}{2}}=x_i+C_s\frac{n}{N}\sin(2\pi x_i)\sin(2\pi y_j),$$
$$y^n_{i-\frac{1}{2},j-\frac{1}{2}}=y_j+C_s\frac{n}{N}\sin(2\pi x_i)\sin(2\pi y_j).$$

where $C_s = 0.1$. The superscript n represents the times of remapping and N is the total number of remapping. Fig. 2 shows the schematic diagrams of uniform mesh $\mathcal{M}^0$ to smoothly moving mesh $\mathcal{M}_S^1$ and smoothly moving mesh $\mathcal{M}_S^7$ to smoothly moving mesh $\mathcal{M}_S^8$.

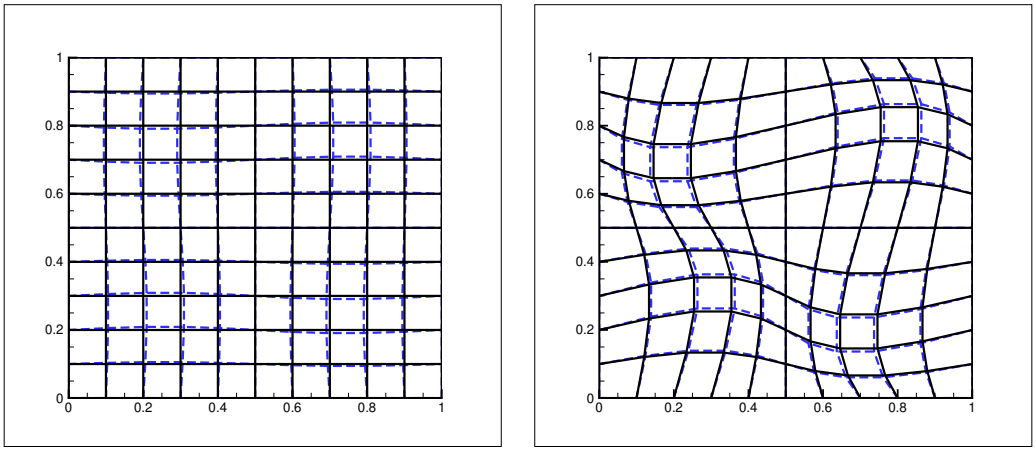


Figure 2: The uniform mesh and the smoothly moving mesh, $N_x = N_y = 10, N = 10$. Left: the black solid mesh is the uniform mesh $\mathcal{M}^0$ and the blue dashed mesh is the smoothly moving mesh $\mathcal{M}_S^1$; Right: the black solid mesh is the smoothly moving mesh $\mathcal{M}_R^7$ and the blue dashed mesh is the smoothly moving mesh $\mathcal{M}_S^8$.

2. The randomly moving mesh $\mathcal{M}_R^n$

$$x^n_{i-\frac{1}{2},j-\frac{1}{2}} = x_i + C_r r^n_{i,j} h_x,$$
$$y^n_{i-\frac{1}{2},j-\frac{1}{2}} = y_j + C_r s^n_{i,j} h_y.$$

where $r^n_{i,j}, s^n_{i,j} \in [-0.5, 0.5]$ are two sequences of independent random numbers. The points on the boundary of the randomly moving mesh $\mathcal{M}_R^n$ are the same as the uniform mesh $\mathcal{M}^0$. We take $C_r = 0.5$ in the following examples and the schematic diagrams are given by Fig. 3.

In the following numerical experiments, the remapping process first transfers the physical variables from the uniform mesh $\mathcal{M}^0$ to the smoothly moving mesh $\mathcal{M}_S^1$ or the randomly moving mesh $\mathcal{M}_R^1$ by our remapping algorithm, and then to the next mesh.

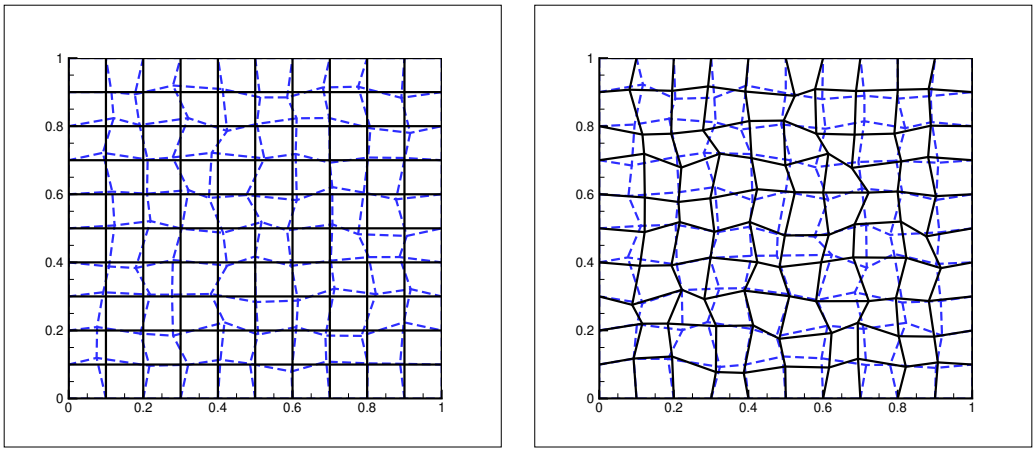


Figure 3: The uniform mesh and the randomly moving mesh, $N_x = N_y = 10, N = 10$. Left: the black solid mesh is the uniform mesh $\mathcal{M}^0$ and the blue dashed mesh is the randomly moving mesh $\mathcal{M}_R^1$; Right: the black solid mesh is the randomly moving mesh $\mathcal{M}_R^7$ and the blue dashed mesh is the randomly moving mesh $\mathcal{M}_R^8$.

This process ends after remapping N times. In order to measure errors and convergence rates simply, we enforce the final mesh to coincide with the original one $\mathcal{M}^0$ after several times of remapping. Suppose the initial cell average of physical quantity in the cell $I_{i,j}$ is $\bar{u}_{i,j}^0$ and after remapping N times the cell average is $\bar{u}_{i,j}^N$ in the same cell, the norms of the error ϵ are given by

$$\|\epsilon\|_{L^1} = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}|\bar{u}_{i,j}^N - \bar{u}_{i,j}^0|S_{i,j}}{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}S_{i,j}},$$

$$\|\epsilon\|_{L^2} = \sqrt{\frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(\bar{u}_{i,j}^N - \bar{u}_{i,j}^0)^2 S_{i,j}}{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}S_{i,j}}},$$

$$\|\epsilon\|_{L^\infty} = \max_{1\leqslant i\leqslant N_x, 1\leqslant j\leqslant N_y}|\bar{u}_{i,j}^N - \bar{u}_{i,j}^0|.$$

3.1 Accuracy tests

To verify the convergence property of our remapping algorithm, we perform accuracy tests on two remapping algorithms: the multi-resolution WENO remapping algorithm without the positivity-preserving limiter (WENO) and the multi-resolution WENO remapping algorithm with the positivity-preserving limiter (P-WENO). We choose the following smooth function with periodic boundary condition for the accuracy test.

$$u(x, y) = \sin^2(2\pi x) \sin^2(2\pi y). \tag{3.1}$$

Table 3.1: Error and order of the WENO and P-WENO remapping algorithms on the smoothly moving meshes, $N = 10$. N_c is the percentage of cells with negative cell averages.

	Mesh	L^1	order	L^2	order	L^∞	order	$N_c(\%)$
WENO	20×20	3.34E-02		6.33E-02		2.43E-01		0.67
	40×40	5.35E-03	2.64	1.29E-02	2.30	7.89E-02	1.62	0.47
	80×80	4.38E-04	3.61	8.99E-04	3.84	5.62E-03	3.81	0.01
	160×160	5.53E-05	2.98	9.60E-05	3.23	4.42E-04	3.67	0
	320×320	5.12E-06	3.43	8.57E-06	3.49	3.05E-05	3.85	0
P-WENO	20×20	3.33E-02		6.33E-02		2.43E-01		0
	40×40	5.35E-03	2.64	1.29E-02	2.30	7.89E-02	1.62	0
	80×80	4.39E-04	3.61	8.99E-04	3.84	5.62E-03	3.81	0
	160×160	5.55E-05	2.98	9.60E-05	3.23	4.42E-04	3.67	0
	320×320	5.14E-06	3.43	8.57E-06	3.49	3.05E-05	3.85	0

We choose five levels of meshes: $N_x = N_y = 20, 40, 80, 160, 320$ with 10 remapping times to investigate convergence. Table 3.1 and 3.2 summarize the errors and numerical rates of convergence. The last column shows the percentage of the cells with negative cell averages during the remapping, denoted as N_c,

$$N_c = \frac{N_{pp}}{N_l \times N_x \times N_y},$$

where N_{pp} is the total amount of cells with negative cell averages during the remapping and N_l is the total times of remapping. From these tables, we can clearly see that all the remapping algorithms have the expected third order of accuracy on both types of moving meshes.

Table 3.2: Error and order of the WENO and P-WENO remapping algorithms on the randomly moving meshes, $N = 10$. N_c is the percentage of cells with negative cell averages.

	Mesh	L^1	order	L^2	order	L^∞	order	$N_c(\%)$
WENO	20×20	4.84E-02		7.15E-02		2.46E-01		1.87
	40×40	3.86E-03	3.65	7.30E-03	3.29	4.69E-02	2.39	2.91
	80×80	1.73E-04	4.48	2.56E-04	4.83	1.34E-03	5.12	0.89
	160×160	1.29E-05	3.74	1.72E-05	3.90	1.10E-04	3.61	0.24
	320×320	6.48E-07	4.32	8.63E-07	4.31	8.64E-06	3.68	0.06
P-WENO	20×20	4.84E-02		7.14E-02		2.46E-01		0
	40×40	3.83E-03	3.66	7.29E-03	3.29	4.69E-02	2.39	0
	80×80	1.74E-04	4.46	2.57E-04	4.83	1.34E-03	5.12	0
	160×160	1.31E-05	3.73	1.73E-05	3.90	1.10E-04	3.61	0
	320×320	6.61E-07	4.31	8.74E-07	4.30	8.64E-06	3.68	0

In order to verify that our remapping method is not limited by mesh movement, that is, it can remap between the old and new meshes where the mesh nodes move more than one cell, we give the following initial mesh configuration

$$h_x^1 < h_x^2 < \cdots < h_x^{N_x},\ h_x^{N_x} = 2h_x^1,$$

where the mesh size $h_x^i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and the y direction is divided equally.

Then we design the flipping mesh with the mesh size

$$\tilde{h}_x^1 = h_x^{N_x}, \cdots, \tilde{h}_x^{N_x} = h_x^1.$$

Fig. 4 shows the schematic diagrams of these two meshes. We remap from the initial mesh to the flipping mesh and return to the initial mesh for $N = 10$ times. The remapping results of the WENO remapping algorithm and the P-WENO remapping algorithm have been shown in Table 3.3 and both of them have the expected third order accuracy, which indicates our remapping method can also have good accuracy when the old and new meshes are not close.

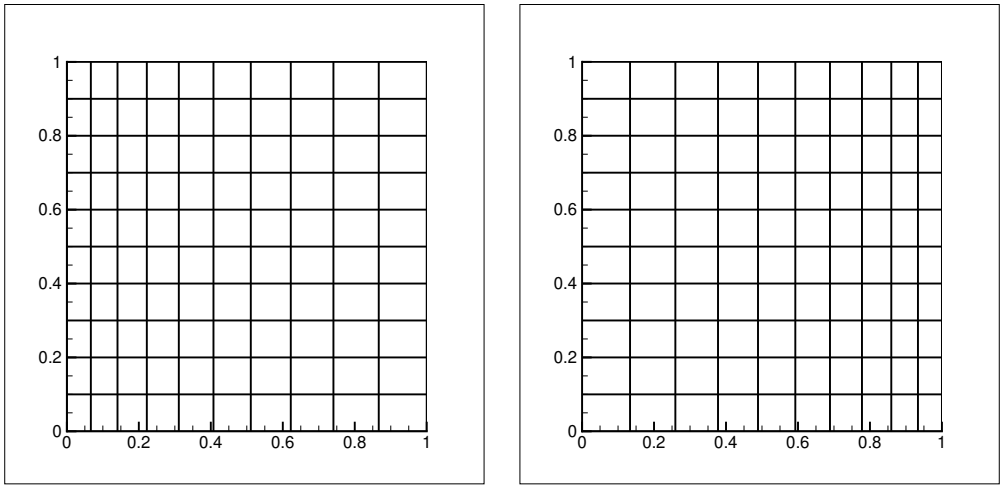


Figure 4: The schematic diagrams of the flipping meshes.

Table 3.3: Error and order of the WENO and P-WENO remapping algorithms on the flipping meshes, $N = 10$. N_c is the percentage of cells with negative cell averages.

	Mesh	L^1	order	L^2	order	L^∞	order	$N_c(\%)$
WENO	20×20	1.62E-01		2.07E-01		5.75E-01		0.33
	40×40	3.53E-02	2.20	5.53E-02	1.90	2.03E-01	1.50	5.73
	80×80	3.47E-03	3.35	5.57E-03	3.31	2.43E-02	3.06	3.73
	160×160	4.33E-04	3.00	6.04E-04	3.21	1.60E-03	3.92	1.68
	320×320	4.87E-05	3.15	6.66E-05	3.18	1.68E-04	3.25	0.70
P-WENO	20×20	1.62E-01		2.07E-01		5.76E-01		0
	40×40	3.38E-02	2.26	5.47E-02	1.92	2.03E-01	1.51	0
	80×80	3.41E-03	3.31	5.54E-03	3.30	2.43E-02	3.06	0
	160×160	4.33E-04	2.98	6.07E-04	3.19	1.60E-03	3.92	0
	320×320	4.87E-05	3.15	6.71E-05	3.18	1.82E-04	3.14	0

3.2 Non-oscillatory test

In this subsection, we test the following discontinuous function inspired by [11] to verify the essentially non-oscillatory property of the WENO remapping algorithm and the positivity-preserving remapping algorithm.

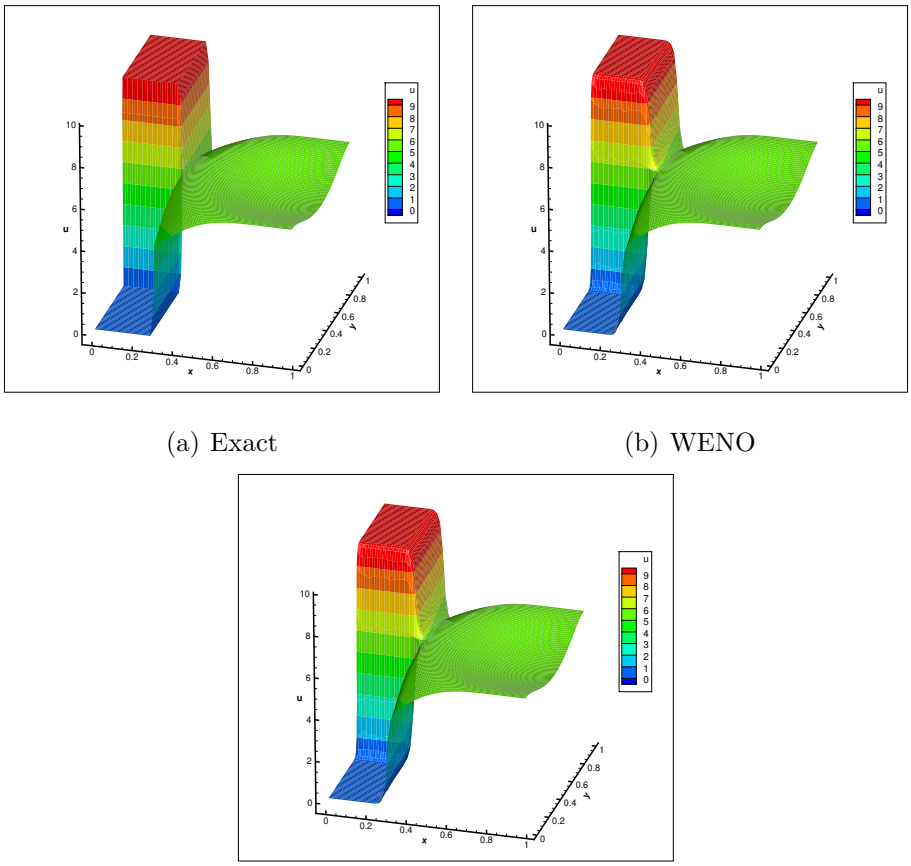
$$u(x,y) = \begin{cases} 10, & \text{if } x \leqslant 0.3 \text{ and } y \geqslant 0.5, \\ 0.1, & \text{if } x \leqslant 0.3 \text{ and } y \leqslant 0.5, \\ \sin(2\pi((x-0.9)^2 + (y-0.5)^2)), & \text{otherwise.} \end{cases}$$

We perform the remapping algorithms 10 times on a 80×80 mesh. We test both on smoothly and randomly moving meshes and the results are similar, so only those with randomly moving meshes are listed to save space.

We plot here the results of the WENO and P-WENO remapping algorithms against the exact results in Fig. 5. It can be observed that our numerical results have no oscillation near the discontinuity, which validates that the WENO and P-WENO remapping algorithms are essentially non-oscillatory and robust.

3.3 Positivity-preserving tests

Compared with the WENO remapping algorithm, the P-WENO remapping algorithm ensures that the remapping results are always positive by using a positivity-preserving limiter. In this subsection we perform several numerical experiments to demonstrate the positivity-preserving property of our P-WENO remapping algorithm. The remapping process still starts from a uniform grid, and the mesh returns to a uniform grid after 10 times randomly moving. In addition, we perform the tests on the flipping mesh to demonstrate that our remapping algorithms can still handle the problem with large mesh deformation.



(a) Exact

(b) WENO

(c) P-WENO

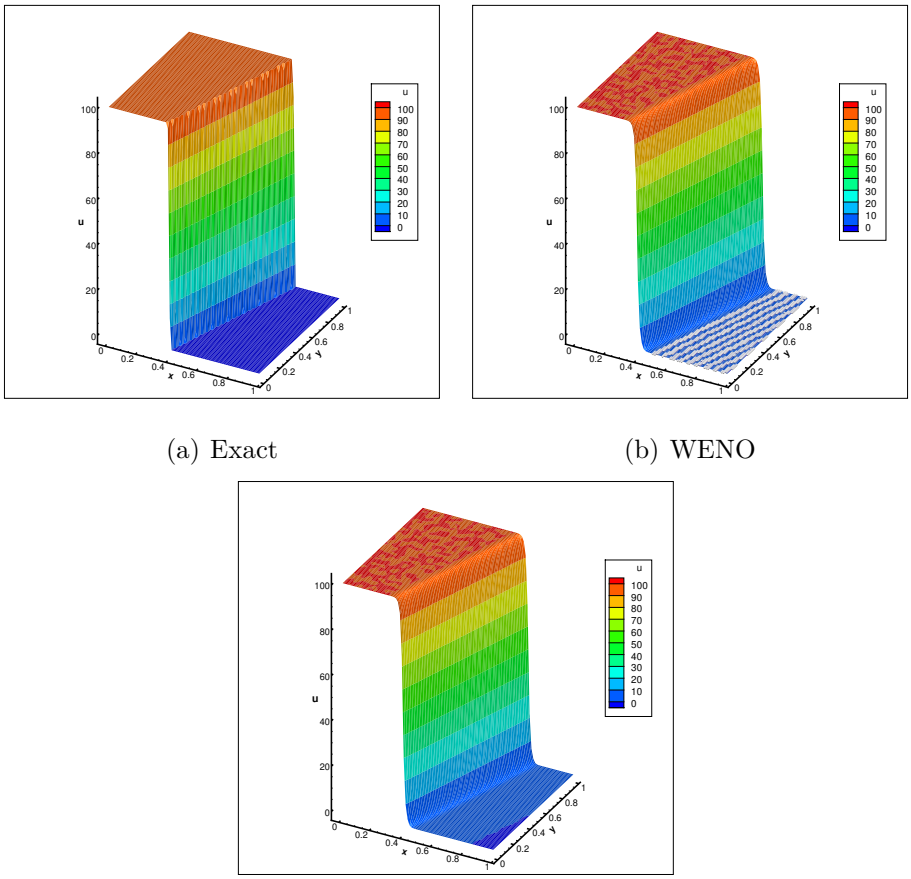
Figure 5: The non-oscillatory test: remapping the discontinuous function by the WENO and P-WENO remapping algorithms, randomly moving meshes, $N = 10, N_x = N_y = 80$.

3.3.1 The step function

We start with the step function (3.2) to test the positivity-preserving property of our remapping algorithm.

$$u(x,y) = \begin{cases} 100, & y > \dfrac{10}{3}(x-0.4), \\ 0, & y \leqslant \dfrac{10}{3}(x-0.4). \end{cases} \tag{3.2}$$

In this test, when $y \leqslant \frac{10}{3}(x-0.4)$, the value of the function is zero, possibly producing negative cell averages during remapping.



(a) Exact (b) WENO

(c) P-WENO

Figure 6: The positivity-preserving test on the step function by the WENO and P-WENO remapping algorithms, $N = 10, N_x = N_y = 80$. Top left: exact; Top right: WENO; Bottom: P-WENO. The white symbols in the top right subfigure represent the cells with negative cell averages.

Fig. 6 shows the results of the WENO and P-WENO remapping algorithms with the mesh

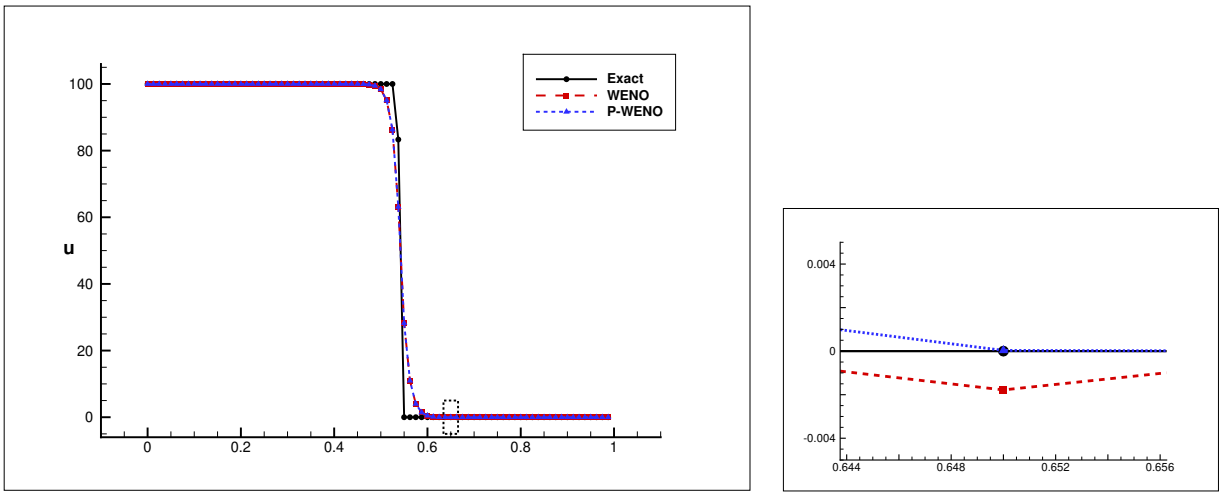


Figure 7: The positivity-preserving test on the step function by the WENO and P-WENO remapping algorithms, $N = 10, N_x = N_y = 80$, remapping results at $j = 40$, there are 14 cells with negative cell averages and the 52nd cell with the smallest value of the cell average, zoomed-in figure at $i = 52$ on the right.

size 80×80. We can observe that the WENO remapping algorithm produces negative cell averages, which are marked with white symbols in Fig. 6. For a more intuitive observation, we cut Fig. 6 along $j = 40$ and show the results in Fig. 7. According to numerical results along $j = 40$, we count that the WENO remapping algorithm produces 14 cells with negative cell averages and the 52nd cell with the smallest value of the cell average of -1.78×10^{-3}. The results from the P-WENO remapping algorithm, by contrast, can always preserve positivity.

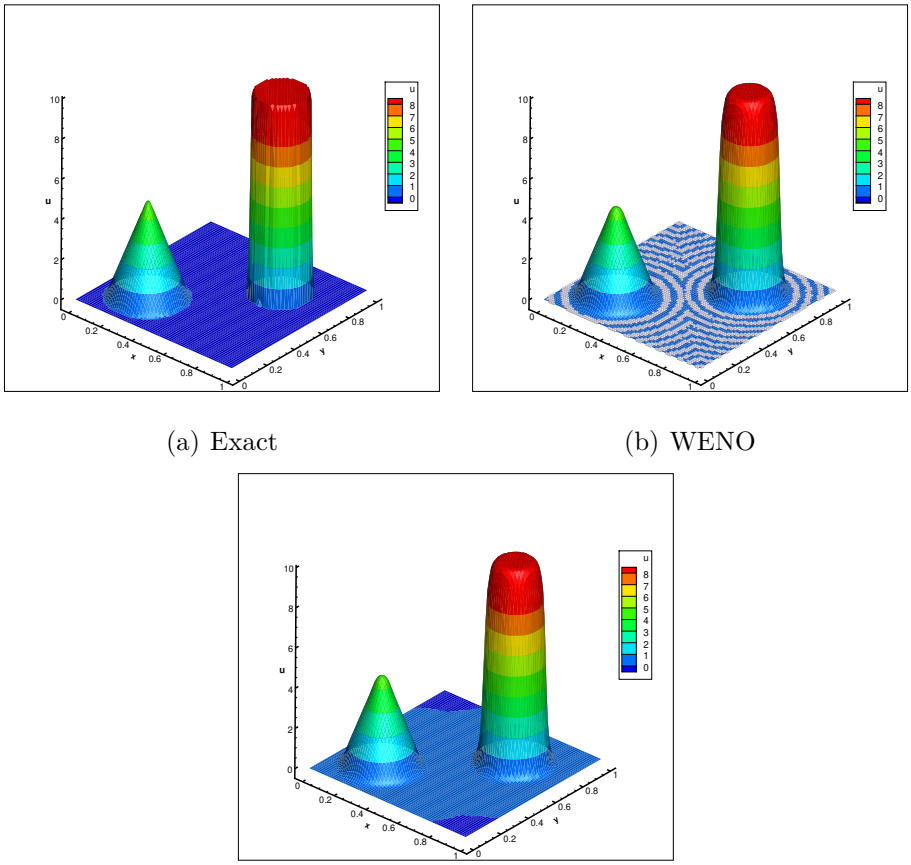
3.3.2 Discontinuous profiles

Next we perform a set of discontinuous profiles (3.3) which is composed of the cylinder function and the cone function.

$$u(x,y) = \begin{cases} 10, & d_1(x,y) < 0.15, \\ 5\max(1 - 5d_2(x,y), 0), & \text{otherwise}, \end{cases} \tag{3.3}$$

where $d_1(x,y) = \sqrt{(x-0.7)^2 + (y-0.7)^2}$ and $d_2(x,y) = \sqrt{(x-0.25)^2 + (y-0.25)^2}$.

Fig. 8 shows the results of the WENO and P-WENO remapping algorithms with the



(a) Exact (b) WENO

(c) P-WENO

Figure 8: The positivity-preserving test on the discontinuous profiles by the WENO and P-WENO remapping algorithms, $N = 10, N_x = N_y = 80$. Top left: exact; Top right: WENO; Bottom: P-WENO. The white symbols in the top right subfigure represent the cells with negative cell averages.

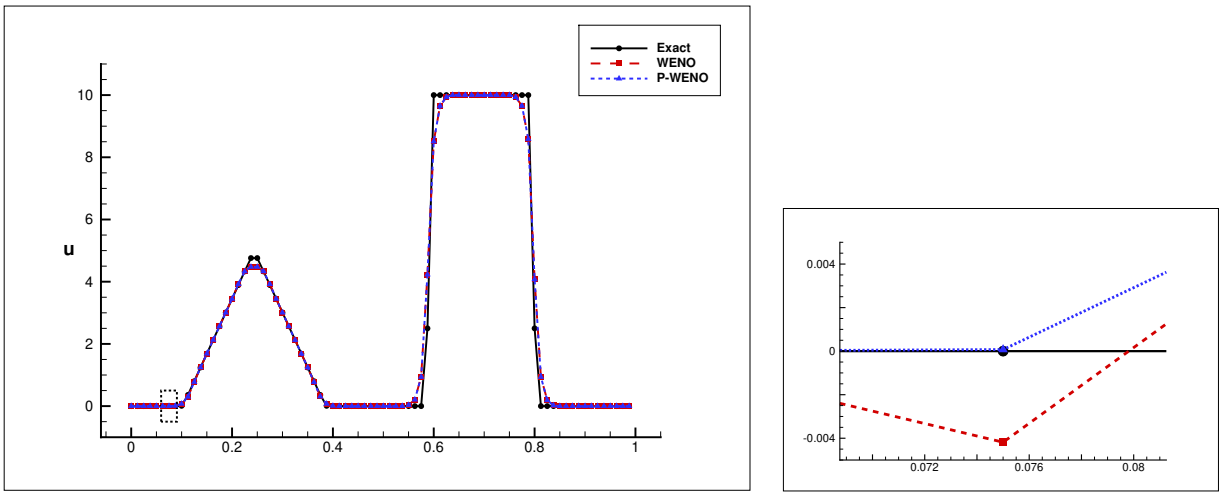
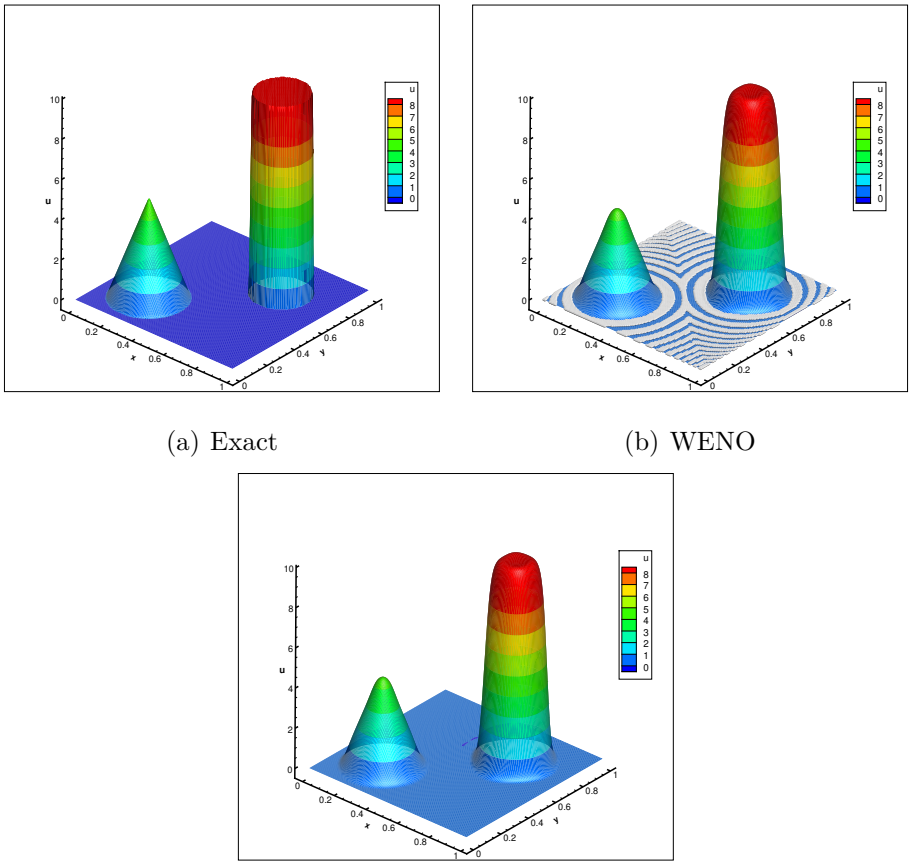


Figure 9: The positivity-preserving test on the discontinuous profiles by the WENO and P-WENO remapping algorithms, $N = 10, N_x = N_y = 80$, remapping results along the diagonal, there are 17 cells with negative cell averages and the 6th cell with the smallest value of the cell average, zoomed-in figure at $i = 6$ on the right.

mesh size 80×80. We cut Fig. 8 along the diagonal to see more clearly and show the results in Fig. 9. According to the numerical results along the diagonal, there are 17 cells with negative cell averages which are produced by the WENO remapping algorithm and the smallest value of the cell average is -4.18×10^{-3} at $i = 6$ near the discontinuity. However the P-WENO remapping algorithm can ensure that the solution remains positive.

In addition, we carry out the positivity-preserving test on the flipping mesh. In Fig. 10, we present the results of the WENO and P-WENO remapping algorithms with the mesh size 160×160 on the flipping mesh. We cut Fig. 10 along the diagonal and show the results in Fig. 11. We can see that neither the WENO nor the P-WENO remapping results contain numerical oscillations around the discontinuities. The P-WENO remapping algorithm can still guarantee the remapping solution to be positive on the flipping mesh.

From the above numerical examples, we illustrate that the WENO remapping algorithm may generate negative cell averages in the regions where the values are close to 0 while no negative cell averages appear if the positive-preserving remapping algorithm is used.



(a) Exact (b) WENO

(c) P-WENO

Figure 10: The positivity-preserving test for the discontinuous profiles by the WENO and P-WENO remapping algorithms on the flipping mesh, $N = 10, N_x = N_y = 160$. Top left: exact; Top right: WENO; Bottom: P-WENO. The white symbols in the top right subfigure represent the cells with negative cell averages.

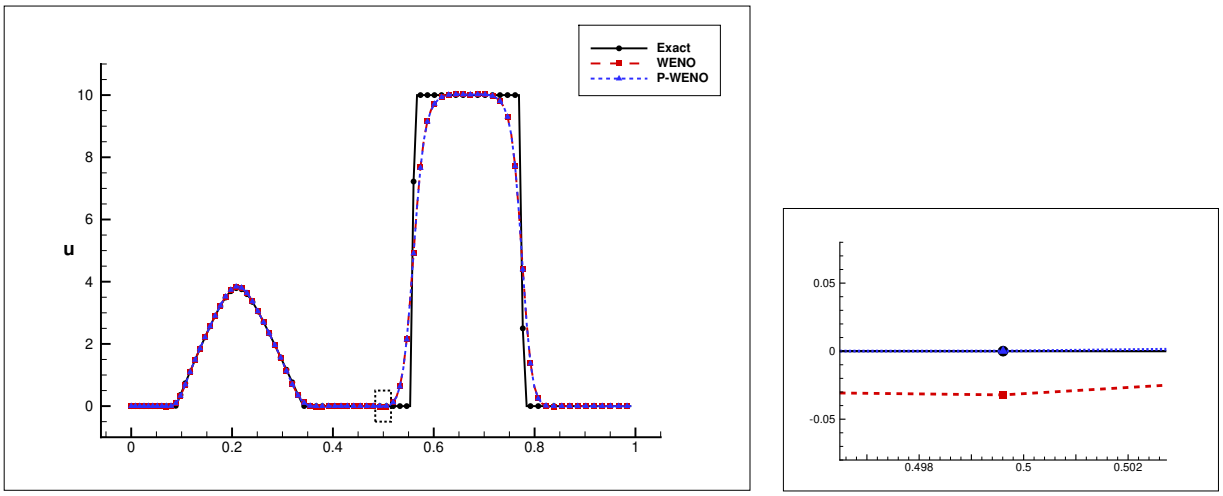


Figure 11: The positivity-preserving test for the discontinuous profiles by the WENO and P-WENO remapping algorithms on the flipping mesh, $N = 10, N_x = N_y = 160$, remapping results along the diagonal, there are 37 cells with negative cell averages and the 93rd cell with the smallest value of the cell average, zoomed-in figure at $i = 93$ on the right.

3.4 The cost of the high order remapping algorithm

In this subsection we demonstrate that our remapping algorithm has low computational cost when the mesh is mildly changed. We consider the following mesh where the node movement does not exceed the length of one cell,

$$\begin{aligned} \tilde{x}_{i-\frac{1}{2},j-\frac{1}{2}} &= ih_x + h_x \sin(2\pi i h_x)\sin(2\pi j h_y), \\ \tilde{y}_{i-\frac{1}{2},j-\frac{1}{2}} &= jh_y + h_y \sin(2\pi i h_x)\sin(2\pi j h_y). \end{aligned} \tag{3.4}$$

For simplicity, we only run the remapping algorithm two times. We set the initial mesh to be the uniform mesh, remap the variables to the smoothly moving mesh described by (3.4). Then the final mesh moves back to the original mesh. The WENO and P-WENO remapping algorithms are tested on the function (3.1). As shown in Table 3.4, we record the number of pseudo time steps and the total computational cost. It can be seen that only three pseudo time steps are required for each remapping, which means that the number of pseudo time steps required for each remapping does not increase with the mesh refinement. This is because the final pseudo-time T in our remapping algorithm depends on the magnitude

of the mesh movement. As the mesh is refined, the time step constraint (2.9) and the final pseudo-time T are halved simultaneously. Consequently, the number of pseudo time steps remains the same on different mesh sizes. This is different from most existing transport equation based remapping methods in which the final pseudo-time is fixed as a constant such as 1. The computational cost is only proportioned to $N_x \times N_y$ and the extra cost brought by the use of positivity-preserving limiter is not too much. It illustrates that our new remapping algorithm is efficient when dealing with the remapping problems with mild grid movement.

Table 3.4: The number of pseudo time steps N_l and the total computational cost.

	Mesh	N_l	Cost(s)
WENO	20×20	6	0.672
	40×40	6	1.563
	80×80	6	5.172
	160×160	6	19.766
	320×320	6	78.063
P-WENO	20×20	6	0.688
	40×40	6	1.563
	80×80	6	5.234
	160×160	6	21.156
	320×320	6	81.375

3.5 The tests in the ALE simulation

In this subsection we will test the performance of our P-WENO conservative remapping algorithm applied in an indirect ALE method. We use the 2D third order positive-preserving multi-resolution WENO Lagrangian scheme proposed in [15] and adopt this scheme in the Lagrangian step in ALE method. We will test three certain benchmark flow problems i.e., the Sedov problem, the Saltzman problem and the Noh problem, in two ways, the purely positivity-preserving Lagrangian scheme and the positivity-preserving ALE method, and compare their results.

3.5.1 The Sedov problem

First we test the Sedov problem which describes the evolution of a blast wave. The initial computational domain is the initially uniform grid consisting of 30×30 rectangular cells in $[0, 1.1] \times [0, 1.1]$. The initial condition is,

$$\rho = 1,\ u = 0,\ v = 0.$$

The internal energy of the system is 10^{-14} almost everywhere except the cell contained the origin where it has a value of 182.09. The computational domain is filled with a prefect gas with $\gamma = 1.4$ and the pressure is given by $p = (\gamma - 1)\rho e$. Reflective boundary conditions are applied on the four boundaries.

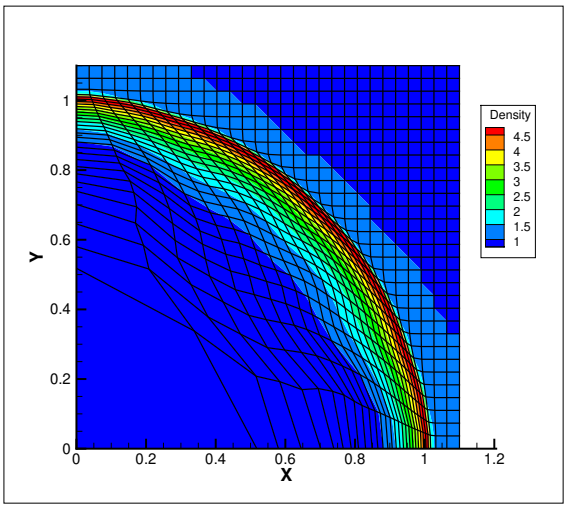
The Sedov problem is solved by the pure Lagrangian scheme and the ALE method with the P-WENO conservative remapping algorithm, respectively. The the rezone and remapping algorithm are implemented every 20 time steps in the ALE computation. Fig. 12 shows the solution for density and pressure at the final time $t = 1$ of the simulation as well as the mesh configuration. We can clearly see that both methods handle this situation well, and the mesh quality in the ALE method is superior to that in the Lagrange scheme.

3.5.2 The Saltzman problem

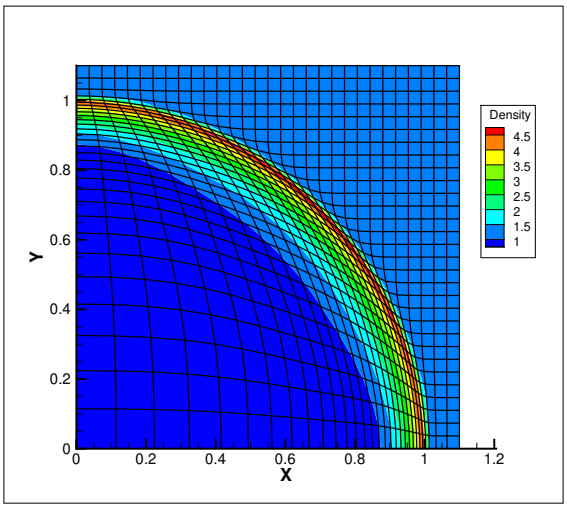
Next, we consider the Saltzman problem which describes the prescribed motion of a piston impacting on the fluid in an enclosed space. The initial computational domain is $[0, 1]\times[0, 0.1]$ which is discretized with 100 cells in the x-direction and 10 cells in the y-direction. To valid the robustness of the Lagrangian method, the initial mesh (Fig. 13) is set to be not aligned with the fluid flow. In this test, the initial condition is

$$\rho = 1,\ u = 0,\ v = 0,\ p = 10^{-10},$$

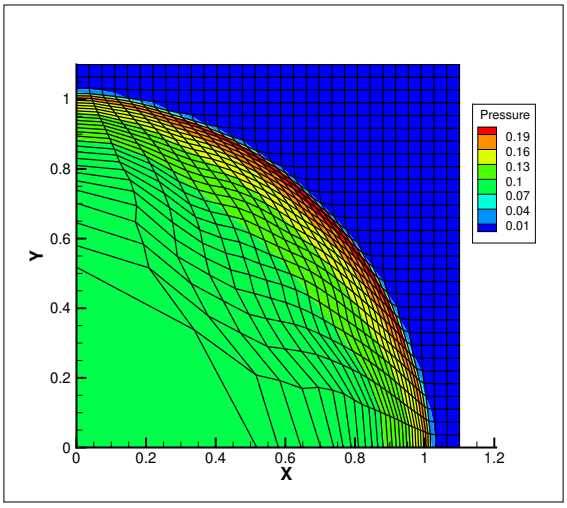
the adiabatic gas constant $\gamma = 5/3$. The left boundary of the computation domain is a moving piston with a constant velocity of 1. All the other boundaries satisfy the reflective boundary conditions.



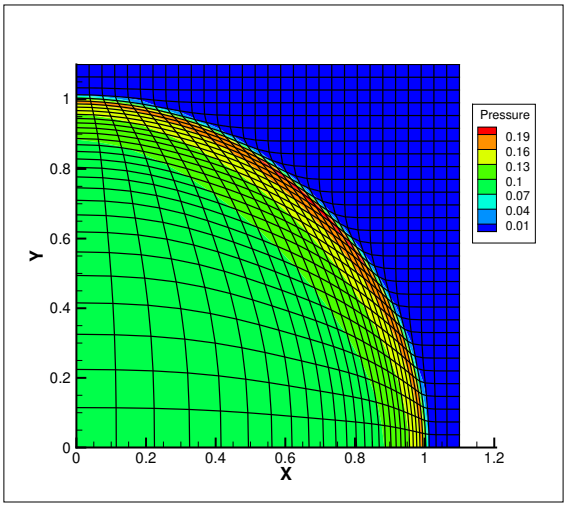
(a) Lagrangian density



(b) ALE density



(c) Lagrangian pressure



(d) ALE pressure

Figure 12: The results for the Sedov problem with $N_x = N_y = 30, t = 1$. Top left: grid and density by the Lagrangian scheme; Top right: grid and density by the ALE method. Bottom left: grid and pressure by the Lagrangian scheme; Bottom right: grid and pressure by the ALE method.

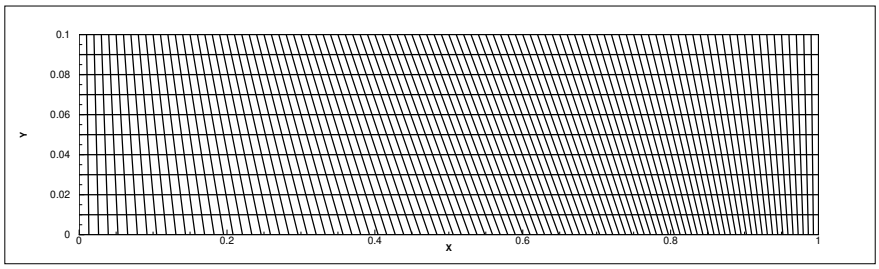
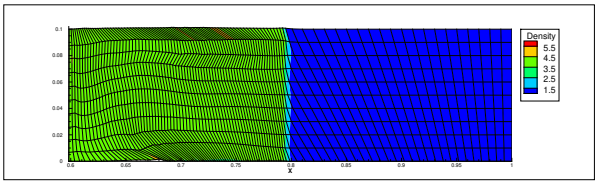


Figure 13: Initial mesh configuration for the Saltzman problem, $N_x = 100, N_y = 10$.

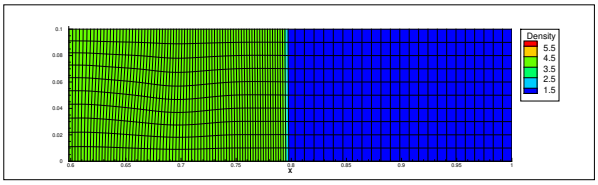
The numerical results for the purely positivity-preserving Lagrangian scheme and the ALE method with our P-WENO remapping algorithm are shown in Fig. 14 at $t = 0.6$. The Lagrangian scheme will blow up at the later time since the meshes are highly compressed and distorted, so we only run the Saltzman problem with the ALE method at $t = 0.8$. The mesh optimization technique at the rezone step is to straighten the mesh in the y direction and the rezone and remapping steps are implemented every 20 time steps. In Fig. 14, we can observe that the shock has arrived at the right boundary and returned when $t = 0.8$. The ALE results can ensure the high quality of the grids and capture the strong shock wave accurately. It demonstrates the more robustness of the ALE method with our remapping algorithm in handling large deformation problems by comparing with the pure Lagrangian scheme.

3.5.3 The Noh problem

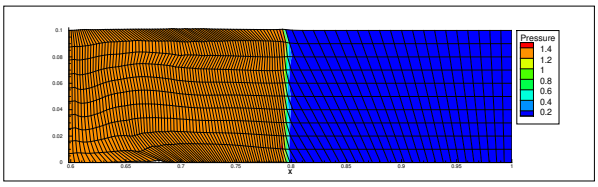
Finally, we test the Noh problem by the purely positivity-preserving Lagrangian scheme and the ALE method with the P-WENO remapping algorithm. The simulation is performed on the initially uniform grid consisting of 35×35 rectangular cells in $[0, 1] \times [0, 1]$. The initial density is 1, the initial internal energy is 10^{-14}, and the initial velocity is directed toward the origin with the magnitude 1. The adiabatic gas constant $\gamma = 5/3$. The right and upper boundaries are set as the free boundary conditions and the lower and left boundaries are taken as the reflective boundary conditions. It is a challenge for the Lagrangian scheme since the mesh will be severely distorted near the origin with the time marching, causing the



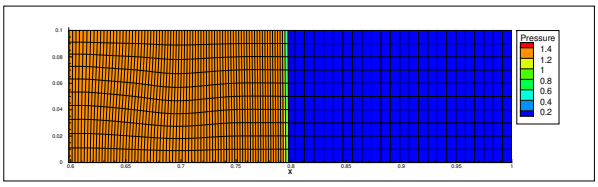
(a) Lagrangian density $t = 0.6$



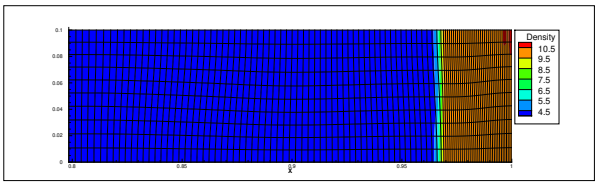
(b) ALE density $t = 0.6$



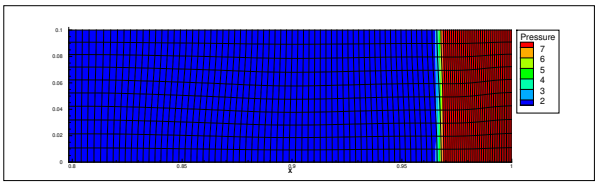
(c) Lagrangian pressure $t = 0.6$



(d) ALE pressure $t = 0.6$



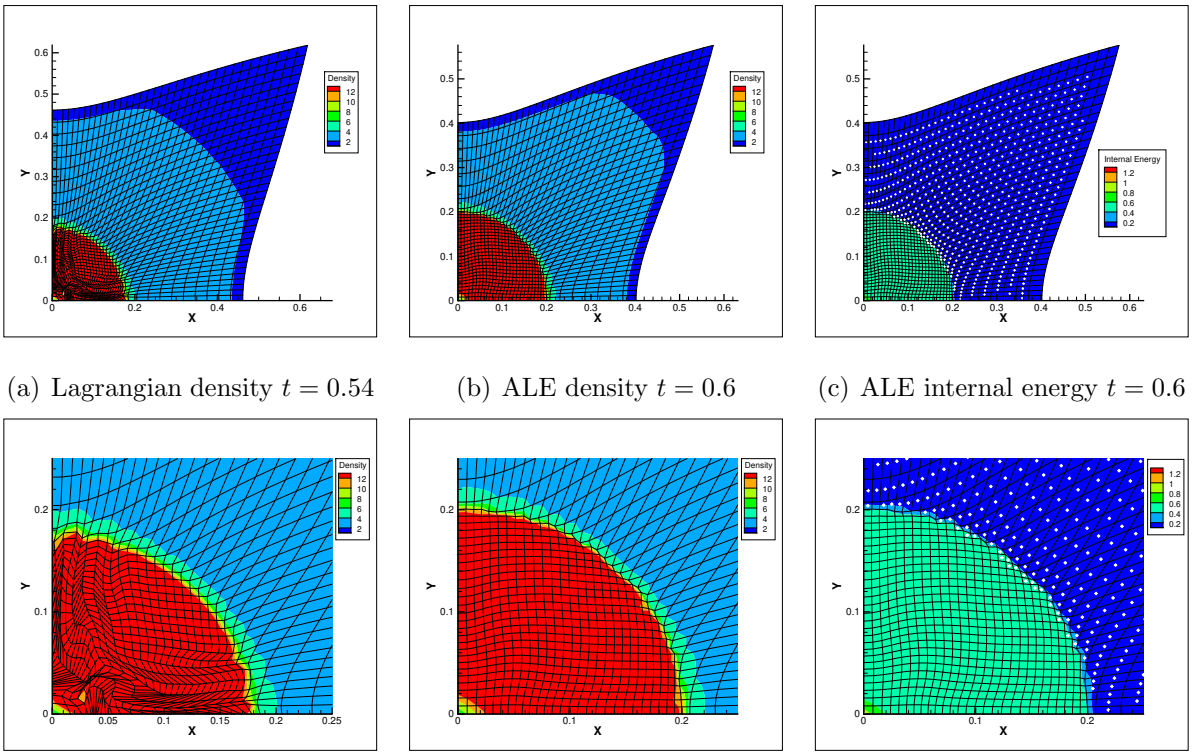
(e) ALE density $t = 0.8$



(f) ALE pressure $t = 0.8$

Figure 14: The results for the Saltzman problem with $N_x = 100$, $N_y = 10$. Top left: grid and density by the Lagrangian scheme at $t = 0.6$; Top right: grid and density by the ALE method at $t = 0.6$. Middle left: grid and pressure by the Lagrangian scheme at $t = 0.6$; Middle right: grid and pressure by the ALE method at time $t = 0.6$; Bottom: grid, density and pressure by the ALE method at $t = 0.8$.

simulation to fail.



(a) Lagrangian density $t = 0.54$ (b) ALE density $t = 0.6$ (c) ALE internal energy $t = 0.6$

(d) Zoomed-in Lagrangian density $t = 0.54$ (e) Zoomed-in ALE density $t = 0.6$ (f) Zoomed-in ALE internal $t = 0.6$

Figure 15: The results for the Noh problem with $N_x = N_y = 35$. Top left: grid and density by the Lagrangian scheme at $t = 0.54$; Top middle: grid and density by the ALE method at $t = 0.6$; Top right: grid and internal energy by the ALE method at $t = 0.6$. Bottom: the zoomed grid, density and internal energy near the origin.

Initially, we adopt the purely positivity-preserving Lagrangian scheme to simulate the Noh problem. After the time $t = 0.54$, the simulation will fail as the time step tends to zero due to the heavily distorted mesh. The results calculated by the pure Lagrangian scheme at $t = 0.54$ is shown in Fig.15. In this case, we replace the Lagrangian scheme with the positivity-preserving ALE method after the time $t = 0.4$ and perform the rezone and remapping algorithms every 10 time steps. We choose to perform the rezone and remapping algorithms after $t = 0.4$, in order to prevent the computational mesh from being so distorted that the shape of the cells cannot keep convex. We have the results of the ALE method at

the time $t = 0.6$ with better mesh quality shown in Fig. 15. Furthermore, the positivity-preserving limiter in the remapping algorithm is needed since the code cannot continue when the negative internal energy and pressure appear. We recorded the cells which have been modified by the positivity-preserving limiter during the remapping. The cell centers of these cells have been labeled in white color in Fig. 15.

In conclusion, the indirect ALE method with our P-WENO remapping algorithm in the above three examples is positivity-preserving and more robust than the pure Lagrangian scheme.

4 Conclusion

In this paper we have presented a conservative, high order accurate, non-oscillatory and positivity-preserving remapping algorithm based on solving the trivial equation $\frac{\partial u}{\partial t} = 0$ with a moving mesh, with the old mesh before remapping as the initial mesh at $t = 0$ and the new mesh after remapping as the final mesh at $t = T$. This remapping algorithm can be used in the ALE methods simulating the fluid flows. We construct a finite volume scheme to solve this moving mesh problem, with the multi-resolution WENO method adopted for the spatial discretization and the SSP Runge-Kutta method used for the temporal discretization. Our remapping algorithm does not restrict the mesh movement and does not need to accurately calculate the intersection regions. It can obtain high order accuracy under very mild requirement on the mesh movement velocity, as long as the boundedness and Lipschitz continuity requirement is satisfied, which can always be satisfied by appropriately choosing the pseudo-time T. Finally, we use a positivity-preserving limiter valid under suitable time step restriction, to ensure the remapping results maintaining the positivity of certain variables such as density and internal energy. This approach does not destroy the original high order of accuracy. We demonstrate numerically that our new remapping algorithm is high order accurate, essentially non-oscillatory and positivity-preserving for a series of test problems. In addition, our remapping algorithm is efficient in dealing with the mesh changing

mildly since the final pseudo-time is related to the magnitude of the mesh movement. We also test the performance of our positivity-preserving remapping algorithm in an indirect ALE simulation. Compared with the pure Lagrangian scheme, the results computed by the ALE method with our remapping algorithm can exhibit better performance in terms of the robustness.

Our future work will involve the extension of high order conservative positivity-preserving remapping algorithm to three dimensions and the design of 3D high order conservative positivity-preserving ALE method based on our remapping algorithm.

References

[1] R. W. Anderson, V. A. Dobrev, T. V. Kolev and R. N. Rieben, *Monotonicity in high-order curvilinear finite element arbitrary LagrangianEulerian remap*, International Journal for Numerical Methods in Fluids, 77, 2015, 249273.

[2] R. W. Anderson, V. A. Dobrev, T. V. Kolev, R. N. Rieben and V. Z. Tomov, *High-order multi-material ALE hydrodynamics*, SIAM Journal on Scientific Computing, 40, 2018, B32-B58.

[3] G. Blanchard and R. Loubere, *High order accurate conservative remapping scheme on polygonal meshes using a posteriori MOOD limiting*, Computers & Fluids, 136, 2016, 83-103.

[4] M. Castro, B. Costa and W. S. Don, *High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws*, Journal of Computational Physics, 230, 2011, 1766-1792.

[5] J. Cheng and C.-W. Shu, *A high order accurate conservative remapping method on staggered meshes*, Applied Numerical Mathematics, 58, 2008, 1042-1060.

[6] J. Cheng and C.-W. Shu, *Positivity-preserving Lagrangian scheme for multi-material compressible flow*, Journal of Computational Physics, 257, 2014, 143-168.

[7] J. K. Dukowicz and J. R. Baumgardner, *Incremental remapping as a transport/advection algorithm*, Journal of Computational Physics, 160, 2000, 318-335.

[8] P. E. Farrell, M. D. Piggott, C. C. Pain, G. J. Gorman and C. R. Wilson, *Conservative interpolation between unstructured meshes via supermesh construction*, Computer Methods in Applied Mechanics and Engineering, 198, 2009, 2632-2642.

[9] J. Grandy, *Conservative remapping and region overlays by intersecting arbitrary polyhedra*, Journal of Computational Physics, 148, 1999, 433-466.

[10] C. W. Hirt, A. A. Amsden and J. L. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, Journal of Computational Physics, 14, 1997, 227-253.

[11] M. Kenamond and M. Shashkov, *The distribution-based remapping of the nodal mass and momentum between arbitrary meshes for staggered arbitrary Lagrangian-Eulerian hydrodynamics*, Computers & Fluids, 201, 2020, 104469.

[12] C. Klingenberg, G. Schnucke and Y. Xia, *Arbitrary Lagrangian-Eulerian discontinuous Galerkin method for conservation laws: Analysis and application in one dimension*, Mathematics of Computation, 86, 2017, 1203-1232.

[13] C. Klingenberg, G. Schnucke and Y. Xia, *An arbitrary Lagrangian-Eulerian local discontinuous Galerkin method for Hamilton-Jacobi equations*, Journal of Scientific Computing, 73 (2017) 906-942.

[14] M. Kucharik, M. Shashkov and B. Wendroff, *An efficient linearity-and-bound-preserving remapping method*, Journal of Computational Physics, 188, 2003, 462-471.

[15] N. Lei, J. Cheng and C.-W. Shu, *A high order positivity-preserving conservative WENO remapping method on 2D quadrilateral meshes*, Computer Methods in Applied Mechanics and Engineering, 373, 2021, 113497.

[16] Y. Li, J. Cheng, Y. Xia and C.-W. Shu, *High order arbitrary Lagrangian-Eulerian finite difference WENO scheme for Hamilton-Jacobi equations*, Communications in Computational Physics, 26, 2019, 1530-1574.

[17] K. Lipnikov and N. Morgan, *A high-order conservative remap for discontinuous Galerkin schemes on curvilinear polygonal meshes*, Journal of Computational Physics, 399, 2019, 108931.

[18] K. Lipnikov and N. Morgan, *Conservative high-order discontinuous Galerkin remap scheme on curvilinear polyhedral meshes*, Journal of Computational Physics, 420, 2020, 109712.

[19] R. Liska, M. Shashkov, P. Vachal and B. Wendroff, *Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian-Eulerian methods*, Journal of Computational Physics, 229, 2010, 1467-1497.

[20] R. Loubere, P. H. Maire, M. Shashkov, J. Breil and S. Galera, *ReALE: A reconnection-based arbitrary-LagrangianEulerian method*, Journal of Computational Physics, 229, 2010, 47244761.

[21] R. Loubere and M. Shashkov, *A subcell remapping method on staggered polygonal grids for Arbitrary-Lagrangian-Eulerian methods*, Journal of Computational Physics, 209, 2005, 105-138.

[22] L. G. Margolin and M. Shashkov, *Second-order sign-preserving conservative interpolation (remapping) on general grids*, Journal of Computational Physics, 184, 2003, 266-298.

[23] L. G. Margolin and M. Shashkov, *Remapping, recovery and repair on a staggered grid*, Computer Methods in Applied Mechanics and Engineering, 193, 2004, 4139-4155.

[24] S. Menon and D. P. Schmidt, *Conservative interpolation on unstructured polyhedral meshes: An extension of the supermesh approach to cell-centered finite-volume variables*, Computer Methods in Applied Mechanics and Engineering, 200, 2011, 2797-2804.

[25] A. L. Ortega and G. Scovazzi, *A geometrically-conservative, synchronized, flux-corrected remap for arbitrary LagrangianEulerian computations with nodal finite elements*, Journal of Computational Physics, 230, 2011, 6709-6741.

[26] J. Velechovsky, J. Breil and R. Liska, *Flux corrected remapping using piecewise parabolic reconstruction for 2D cell-centered ALE methods*, International Journal for Numerical Method in Fluids, 76, 2014, 575-586.

[27] J. Velechovsky, R. Liska and M. Shashkov, *High-order remapping with piece-wise parabolic reconstruction*, Computers & Fluids, 83, 2013, 164-169.

[28] X. Zhang and C.-W. Shu, *On positivity preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, Journal of Computational Physics, 229, 2010, 8918-8934.

[29] X. Zhang and C.-W. Shu, *Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments*, Proceedings of the Royal Society A, 467, 2011, 2752-2776.

[30] J. Zhu and C.-W. Shu, *A new type of multi-resolution WENO schemes with increasingly higher order of accuracy*, Journal of Computational Physics, 375, 2018, 659-683.