A high order positivity-preserving conservative WENO remapping method on 3D tetrahedral meshes

Nuo Lei[1], Juan Cheng[2] and Chi-Wang Shu[3]

Abstract

We propose a high order positivity-preserving conservative remapping method on three-dimensional (3D) tetrahedral meshes, based on the weighted essentially non-oscillatory (WENO) reconstruction method. By precisely computing the overlaps between the meshes before and after the rezoning step in the arbitrary Lagrangian-Eulerian (ALE) framework, our method does not limit the range of mesh movements and has wider applications. This also makes our remapping process simpler to attain high-order accuracy. We use the third order multi-resolution WENO reconstruction procedure in this paper as an example, in which we reconstruct three polynomials of different orders via nested central spatial stencils and distribute nonlinear weights based on the smoothness of the polynomials, ensuring optimal accuracy in the smooth region while avoiding numerical oscillations in the non-smooth region. The multi-resolution WENO procedure involves fewer high-order reconstruction polynomials and can use arbitrary positive linear weights, making it more effective for our 3D remapping problem. We incorporate an efficient local limiting to preserve positivity for the positive physical variables involved in the ALE framework without sacrificing the original high-order accuracy and conservation. A set of numerical tests are provided to verify properties of our remapping algorithm, such as high-order accuracy, conservation, essentially non-oscillatory performance, positivity-preserving and efficiency.

Keywords: 3D remapping; High-order accuracy; Non-oscillatory; Positivity-preserving; Conservative; Multi-resolution WENO

[1]Graduate School, China Academy of Engineering Physics, Beijing 100088, China. E-mail: leinuo19@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]Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: chi-wang_shu@brown.edu. Research is supported in part by AFOSR grant FA9550-20-1-0055 and NSF grant DMS-2010107.

1 Introduction

The arbitrary Lagrangian-Eulerian (ALE) framework, which combines the benefits of two fundamental strategies in computational fluid dynamics, namely the Lagrangian framework and the Eulerian framework, has a wide range of applications in numerical simulation of fluid flow. The evolution of the indirect ALE method consists of three steps: the Lagrangian step for solving the fluid equations, which involves mesh movements with the fluid flow, the rezoning step, which may be carried out after one or several time steps, for improving the mesh quality, and the remapping step for transferring physical variables to the rezoned mesh.

In this paper, we focus on the three-dimensional remapping algorithm in the indirect ALE method. After the Lagrangian process, we have the physical variables on the old deformed mesh. The rezoning step provides a new high-quality mesh, and the remapping process has to transfer the physical variables from the old mesh to the new mesh. The mesh rezoning and the subsequent remapping steps are complementary to the Lagrangian step, without which the mesh may be seriously distorted, making the computation difficult to proceed. A suitable remapping algorithm should be consistent in accuracy with the Lagrangian step. In particular, the remapping method for the ALE framework should introduce as little error as possible and adhere to physical properties of the fluid flow variables, such as conservation and positivity. In practice, this procedure should be efficient, to economically cope with any mesh movements.

The flux-based method and the intersection-based method are two widely employed types of remapping algorithms. The flux-based method attempts to determine the exchanges between the old and new mesh cells, which can also be described as a transport equation [6, 16], which transfers the numerical variables from the old mesh to the new mesh conservatively. The advantage of the flux-based method is its simplicity: only a transport equation needs to be solved, for which one can use standard numerical methods. The disadvantage of the flux-based method is that it would require mesh connectivity from the old mesh to the new mesh, which must have the same number of mesh cells with the same connection topology. Also,

if the new mesh changes significantly from the old mesh, or if the change is not smooth, the flux-based method is difficult to implement, may run slowly, and may be difficult to achieve high order accuracy.

Another commonly used remapping algorithm, the intersection-based method, attempts to determine the exact overlaps between the new and old meshes, and integrate the reconstruction polynomials on these overlaps to acquire information on the new mesh. This remapping technique has no requirement on the range of changes from the old mesh to the new mesh. The new mesh could have different number of cells and different connection topology from the old mesh, making the remapping procedure more flexible and easier to attain high-order accuracy. However, finding overlaps between the old and new mesh cells is a complicated procedure, making the code much more laborious than the flux-based technique, especially for the complicated mesh cells in three dimension (3D). Besides that, due to the cost of the clipping algorithm, the intersection-based method could be more expensive than the flux-based method, especially if the new mesh is only a small and smooth movement from the old one. The authors in [14, 11, 12] have examined the benefits and drawbacks of these two strategies, while dealing with the two-dimensional multi-material remapping problems with small mesh movements. The advocated procedure is to use the fast and easy flux-based technique for the cells with only one material, while for the cells with complicated multi-material situation, the intuitive but expensive intersection-based method is adopted.

In the Lagrangian framework, the mesh nodes move with the fluid flow, which may not be smooth. Especially, if one decides to perform the rezoning and remapping after many time steps, or only when the mesh quality becomes too poor, the change from the old mesh to the new mesh can be very large and non-smooth. It may even happen that the number of cells and the cell connectivity are different from the old mesh to the new mesh, such as during local refinement or coarsening of meshes in adaptivity. In such situations, the intersection-based remapping method shows a significant advantage. By clipping the exact intersections between the old and new mesh cells, the intersection-based remapping method

has no limit on the mesh movements and other changes.

To obtain a high-order accurate ALE method, the accuracy of the remapping algorithm must be at least as high as that of the Lagrangian step. A high-order intersection-based remapping approach depends on a high-order reconstruction method. However, high-order reconstruction polynomials may produce numerical oscillations around discontinuities or large gradient regions, which should be addressed. In the meantime, widely used in the computational fluid dynamics, the ALE framework involves some physical variables with special physical properties, such as the conservation of mass, momentum, total energy as well as the positivity of density and internal energy. As a result, the remapping algorithm should obey these physical properties.

After years of development, there are currently lots of work about two-dimensional and three-dimensional remapping algorithms, and most of the three-dimensional methods are developed from the lower dimensional cases. Grandy developed a first-order accurate intersection-based remapping algorithm for polyhedral meshes [8]. After that, a few high-order accurate remapping methods have been proposed [4, 16, 15], based on the high-order essentially non-oscillatory (ENO) reconstruction or on solving the transport equation over curvilinear polyhedral meshes. Besides the high-order accuracy, there are some three-dimensional remapping algorithms focused on the bound preserving property for certain positive physical variables in the simulation of fluid flow. With the flux corrected transport approach [1] or the Barth-Jespersen method [7, 3, 5, 2], all of them are firstly designed in the two-dimensional case [2, 13, 18]. Based on the assumption that low-order methods guarantee bound preservation but high-order methods produce out-of-bounds solutions, these bound preservation methods typically substitute the high-order method near the extrema which are near the bounds by a low order method, and could lead to the loss of accuracy. However, there has been seldom discussion on designing a bound preservation remapping method with more than second-order accuracy.

In this paper, we would like to design a high-order accurate, essentially non-oscillatory

intersection-based remapping algorithm on three-dimensional tetrahedral meshes, following our previous work on the two-dimensional quadrilateral meshes [15]. We also would like to preserve physical properties such as conservation and positivity without losing the high-order accuracy. We adopt an exact clipping approach to calculate the intersections after first reconstructing high-order polynomials on the existing meshes with the known information such as the cell averages. After that, to retain positivity for the physical positive variables, we will apply a local positivity-preserving limiter and compute the new information by integrating the modified polynomials over the intersections.

As mentioned above, a high-order reconstruction method is necessary for a high-order remapping algorithm. The weighted essentially non-oscillatory (WENO) [17, 9] idea used in solving hyperbolic equations can achieve high-order accuracy in smooth regions and can avoid numerical oscillations in non-smooth regions. It defines smoothness indicators to measure the smoothness of the reconstruction polynomials on different stencils and assign smaller weights for the reconstruction polynomials that may cause numerical oscillations. More recently, Zhu and Shu [22] have designed the multi-resolution WENO method, which is more effective especially on unstructured meshes or moving meshes, due to the allowance of arbitrary positive linear weights and simpler central stencil combinations. Based on these high-order non-oscillatory reconstruction methods, the two-dimensional remapping algorithm can achieve high-order accuracy [4, 15], and we extend it to the three-dimensional remapping problem in this paper.

Generally, it is not difficult to preserve positivity for first-order remapping methods, but it is more difficult to keep high-order accuracy while preserving positivity. The positivity-preserving limiter of Zhang and Shu [20, 21] is a good attempt to tackle this problem, and it has been validated in the 2D remapping procedure [15]. By compressing out-of-bounds point values toward the cell average at specific quadrature points, this limiter, originally designed to solve compressible Euler equations, can preserve positivity of the physical variables such as density and internal energy without sacrificing the high-order accuracy and conservation.

Meanwhile, this positivity-preserving limiter is simple to implement, requiring only a few lines in the code.

In this paper, we will focus on the three-dimensional remapping method, which is organized as follows. Section 2 presents our three-dimensional intersection-based remapping algorithm in detail. In the beginning, we describe the architecture of our remapping method, which is divided into four parts. With the multi-resolution WENO idea, we reconstruct high-order, non-oscillatory polynomials, and then compute the intersections of the new and old tetrahedral cells. After that, we apply a positivity-preserving limiter on modifying the high-order polynomials for the positive physical variables, and finally we integrate the reconstruction polynomials across the overlaps to obtain the new cell averages. Several numerical experiments in Section 3 are given to verify the properties of our algorithm, including tests for accuracy, non-oscillatory performance, dissipation effects, positivity-preserving, with a cost analysis. Concluding remarks are given in Section 4.

2 The remapping algorithm

2.1 Basic concepts

In this section, we will describe our three-dimensional remapping algorithm step by step in detail. After the Lagrangian process and the rezoning process, in the ALE framework, a remapping algorithm is needed to convert variables from the old mesh to the new mesh.

Suppose the three-dimensional computational domain Ω is a connected domain decomposed into tetrahedral cells $\{I_i\}_{i=1}^{N}$. We use the notation $|I_i|$ to represent the volume of cell I_i and $\{\tilde{I}_i\}_{i=1}^{\tilde{N}}$ to represent the new rezoned mesh cells. Obviously, we require that there are neither overlaps nor gaps between any two neighboring cells

$$\bigcup_{i=1}^{N} I_i = \bigcup_{i=1}^{\tilde{N}} \tilde{I}_i = \Omega, \quad \left|I_i \bigcap I_j\right| = 0, \quad \left|\tilde{I}_i \bigcap \tilde{I}_j\right| = 0, \quad i \neq j,$$

and this rule will not be broken by the new mesh in the rezoning step of a typical indirect ALE method.

The physical variables in a typical indirect ALE technique for computational fluid dynamics are normally conservative variables, and the remapping algorithm makes no distinction between these conservative variables. We denote the cell average of a conservative variable (such as mass (density), momentum, or total energy) as $\bar{u}_i$. The ALE framework's conservative remapping procedure requires us to transfer the cell average $\bar{u}_i$ from the old mesh $\{I_i\}_{i=1}^{N}$ created after the Lagrangian step to the new mesh $\{\tilde{I}_i\}_{i=1}^{\tilde{N}}$ updated by the rezoning step in a conservative, accurate and non-oscillatory manner. In order to achieve high-order accuracy in the ALE approach, the remapping phase requires a high-order reconstruction method. When high-order reconstruction polynomials are included, more emphasis should be placed on keeping the solution non-oscillatory, particularly in numerical simulations with nonlinear conservation laws. Besides conservation, high-order accuracy and non-oscillatory performance, the remapping method should maintain positivity (non-negativity) for non-negative physical quantities such as density and internal energy.

We extend the two-dimensional remapping approach proposed in [15] to three-dimensional tetrahedral meshes in this paper. Reconstructing high-order polynomials $\{u^i(x,y,z)\}_{i=1}^{N}$ on the old meshes $\{I_i\}_{i=1}^{N}$, and calculating the new cell averages $\{\tilde{\bar{u}}_i\}_{i=1}^{\tilde{N}}$ on the new meshes $\{\tilde{I}_i\}_{i=1}^{\tilde{N}}$ by integrating over the intersections between the old and new meshes, are the main components of the remapping procedure. Meanwhile, to avoid numerical oscillations and negative values for physically positive quantities, the WENO procedure and a positivity-preserving limiter are used. Hence, we break the description of the remapping algorithm into several parts:

- High-order reconstruction: reconstruct high-order polynomial $u^i(x,y,z)$ in each cell I_i with the multi-resolution WENO method;
- Clipping: utilize the clipping algorithm in [19] to compute the intersection $I_i \bigcap \tilde{I}_j$;
- Positivity-preserving modification: modify the reconstruction polynomials by the positivity-preserving limiter in [20, 21] to maintain positivity for the physically positive variables

such as density and internal energy;

- Numerical integration: calculate exactly (module round-off error) the integration of the modified polynomials over the intersections.

Next, we will describe our remapping algorithm step by step.

2.2 High-order reconstruction

It is not difficult to reconstruct high-order polynomials based on the cell averages, but how to avoid numerical oscillation effectively deserves more attention. The WENO idea widely used in finite volume and finite difference methods is a good attempt to overcome this difficulty. In recent years, Zhu and Shu have proposed a finite volume multi-resolution WENO scheme for solving hyperbolic conservation laws [22]. By reconstructing unequal degree polynomials on several unequal-sized central stencils with the WENO idea, this new type of WENO reconstruction is simpler and more effective, especially on unstructured meshes and moving meshes, and they inherit the advantages of the classical WENO reconstructions.

Here we try to perform the multi-resolution WENO reconstruction procedure on the cell I_i. Following the reconstruction step in [22], we reconstruct a zeroth degree polynomial $q_1^i(x,y,z)$ on the spatial stencil $T_1^i = \{I_i\}$ and a first degree polynomial $q_2^i(x,y,z)$ on the spatial stencil $T_2^i = \{I_i, I_{i1}, I_{i2}, I_{i3}, I_{i4}\}$, where $I_{ij}, j = 1,2,3,4$ are the four neighbors of the cell I_i. For the second degree polynomial $q_3^i(x,y,z)$, we select a central spatial stencil T_3^i which includes the cell I_i, its four neighbors $I_{i1}, I_{i2}, I_{i3}, I_{i4}$, and the neighboring cells of these four neighbors

$$T_3^i = \{I_i, I_{i1}, I_{i2}, I_{i3}, I_{i4}, I_{i11}, I_{i12}, I_{i13}, I_{i21}, I_{i22}, I_{i23}, I_{i31}, I_{i32}, I_{i33}, I_{i41}, I_{i42}, I_{i43}\},$$

where $I_{ijk}, k = 1,2,3$ are the three neighbors of the cell I_{ij} other than the central cell I_i for $j = 1,2,3,4$. Notice that some of these neighbors' neighboring cells might be the same. In Figure 2.1, we give an example of these nested central spatial stencils.

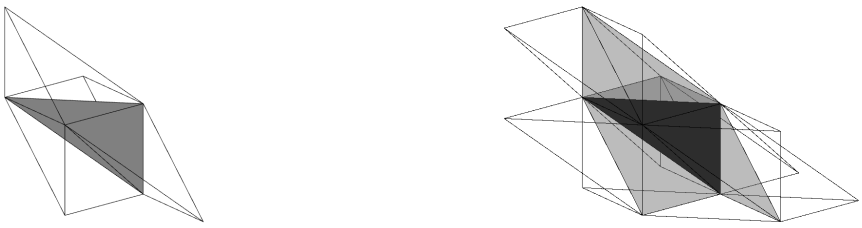

Figure 2.1: Nested central spatial stencils. Left: spatial stencil T_1^i with dark gray color and transparent spatial stencil T_2^i; Right: spatial stencil T_1^i with black color, spatial stencil T_2^i with gray color and transparent spatial stencil T_3^i.

To keep conservation, the above three polynomials must satisfy

$$\int_{I_i} q_l^i(x,y,z)dxdydz = \bar{u}_i|I_i|, \quad l = 1,2,3,$$

where we recall that $|I_i|$ is the volume of the cell I_i. It is easy to reconstruct the zeroth degree polynomial $q_1^i(x,y,z)$, but we need a constraint least-square procedure to determine $q_2^i(x,y,z)$ and $q_3^i(x,y,z)$,

$$\begin{array}{rcl} q_2^i(x,y,z) & = & \arg \min\limits_{p_2(x,y,z)\in\mathcal{P}^1} \sum\limits_{I_l\in T_2^i\setminus I_i} \left(\int_{I_l} p_2(x,y,z)dxdydz - |I_l|\bar{u}_l\right)^2, \\ q_3^i(x,y,z) & = & \arg \min\limits_{p_3(x,y,z)\in\mathcal{P}^2} \sum\limits_{I_l\in T_3^i\setminus I_i} \left(\int_{I_l} p_3(x,y,z)dxdydz - |I_l|\bar{u}_l\right)^2, \\ \text{s.t.} & & \int_{I_i} q_2^i(x,y,z)dxdydz = \int_{I_i} q_3^i(x,y,z)dxdydz = \bar{u}_i|I_i|, \end{array} \tag{2.1}$$

where $p_2(x,y,z)$ and $p_3(x,y,z)$ are taken from $\mathcal{P}^1$ and $\mathcal{P}^2$, respectively. Here, $\mathcal{P}^m$ is the space of polynomials of degree less than or equal to m for $m = 1,2$. Furthermore, we can extend our reconstruction method to higher-order by adding the next layer of neighboring cells around the central spatial stencil T_3^i, etc., and reconstructing higher-order polynomials.

Next, we introduce positive linear weights and equivalent expressions for $q_1^i(x,y,z)$, $q_2^i(x,y,z)$, $q_3^i(x,y,z)$,

$$\begin{array}{rcl} p_1^i(x,y,z) & = & q_1^i(x,y,z) \\ p_2^i(x,y,z) & = & \frac{1}{\gamma_{2,2}} q_2^i(x,y,z) - \frac{\gamma_{1,2}}{\gamma_{2,2}} p_1^i(x,y,z) \\ p_3^i(x,y,z) & = & \frac{1}{\gamma_{3,3}} q_3^i(x,y,z) - \frac{\gamma_{1,3}}{\gamma_{3,3}} p_1^i(x,y,z) - \frac{\gamma_{2,3}}{\gamma_{3,3}} p_2^i(x,y,z) \end{array} \tag{2.2}$$

where $\sum_{l=1}^{k}\gamma_{l,k}=1,\gamma_{k,k}\neq 0,k=2,3$ are the linear weights. Following [22], we assume $\gamma_{l,k}=\frac{\bar{\gamma}_{l,k}}{\sum_{l=1}^{k}\bar{\gamma}_{l,k}}$, where $\bar{\gamma}_{l,k}=10^{l},l=1,\cdots,k$. Actually, different values of $\bar{\gamma}_{l,k}$ have modestly different results in the numerical tests of [22]. By combing $p_l^i(x,y,z),l=1,2,3$ with the linear weights $\gamma_{l,3},l=1,2,3$, one can achieve optimal accuracy on the biggest stencil T_3^i,

$$\gamma_{1,3}p_1^i(x,y,z)+\gamma_{2,3}p_2^i(x,y,z)+\gamma_{3,3}p_3^i(x,y,z)=q_3^i(x,y,z).$$

In smooth regions, one can get optimal accuracy by combing $p_l^i(x,y,z),l=1,2,3$ with the linear weights, but in non-smooth regions, the WENO idea should be adopted to avoid numerical oscillations.

In non-smooth regions, the high-order reconstruction polynomials may generate numerical oscillations, so we use the smoothness indicators β_2 and β_3 to measure the smoothness of $p_2^i(x,y,z)$ and $p_3^i(x,y,z)$, respectively. Following the practice in [9, 22], we define the smoothness indicators as

$$\beta_2=\sum_{|l|=1}^{1}|I_i|^{\frac{2|l|}{3}-1}\int_{I_i}\left(\frac{\partial^{|l|}}{\partial x^{l_1}\partial y^{l_2}\partial z^{l_3}}p_2^i(x,y,z)\right)^2dxdydz, \tag{2.3}$$

$$\beta_3=\sum_{|l|=1}^{2}|I_i|^{\frac{2|l|}{3}-1}\int_{I_i}\left(\frac{\partial^{|l|}}{\partial x^{l_1}\partial y^{l_2}\partial z^{l_3}}p_3^i(x,y,z)\right)^2dxdydz, \tag{2.4}$$

where $l=(l_1,l_2,l_3),|l|=l_1+l_2+l_3,l_1,l_2,l_3\geq 0$. If the high-order polynomials are not smooth enough, the smoothness indicators will be large and we will distribute smaller nonlinear weights to them to avoid numerical oscillations.

The determination of β_1 is more special. It would be equal to zero if β_1 is calculated in the same way as those for β_2 and β_3. In smooth regions, the adoption of $\beta_1=0$ would have no effect on the optimal high order accuracy. However, in non-smooth regions, $\beta_1=0$ would cause the zero-order polynomial $p_1^i(x,y,z)$ to take too large weight, leading to excessive numerical smearing. Therefore, we would like to slightly magnify β_1. We select four new stencils $T_{1,l}^i=\{I_{il1},I_{il2},I_{il3}\},l=1,2,3,4$, and construct four linear polynomials $p_{1,l}^i(x,y,z)\in span\{\frac{x-x_{il}}{|I_{il}|^{\frac{1}{3}}},\frac{y-y_{il}}{|I_{il}|^{\frac{1}{3}}},\frac{z-z_{il}}{|I_{il}|^{\frac{1}{3}}}\}$ satisfying $p_{1,l}^i(x_{ilk},y_{ilk},z_{ilk})=\bar{u}_{ilk}-\bar{u}_{il},l=1,2,3,4,k=1,2,3$, where $\bar{u}_{ilk}$ and $(x_{ilk},y_{ilk},z_{ilk})$ are the cell average and the barycenter of $I_{ilk}\in T_3^i$. Using the WENO

idea to combine these four polynomials, we set $\lambda_{1,l} = \frac{1}{4}$ and

$$\sigma_\ell = \lambda_{1,\ell}\left(1 + \frac{\left(\frac{|\beta_{1,1}-\beta_{1,2}|+|\beta_{1,1}-\beta_{1,3}|+|\beta_{1,1}-\beta_{1,4}|+|\beta_{1,2}-\beta_{1,3}|+|\beta_{1,2}-\beta_{1,4}|+|\beta_{1,3}-\beta_{1,4}|}{6}\right)^2}{\beta_{1,\ell}+\varepsilon}\right),\ \varepsilon = 10^{-6},$$

for $l = 1, 2, 3, 4$, where $\beta_{1,l}$ are the associated smoothness indicators of $p^i_{1,l}(x, y, z)$ calculated by (2.3). Then β_1 is defined as

$$\beta_1 = \sum_{|l|=1}^{1} |I_i|^{\frac{2}{3}|l|}\left(\frac{\partial^{|l|}}{\partial x^{l_1}\partial y^{l_2}\partial z^{l_3}}\left(\sum_{\ell=1}^{4}\frac{\sigma_\ell}{\sigma}p^i_{1,\ell}(x, y, z)\right)\right)^2 \tag{2.5}$$

where $\sigma = \sum_{l=1}^{4}\sigma_l$.

Based on the linear weights $\gamma_{l,3}$ and the smoothness indicators β_l, we introduce the nonlinear weights ω_l as

$$\omega_l = \frac{\tilde{\omega}_l}{\sum_{l=1}^{3}\tilde{\omega}_l},\quad \tilde{\omega}_l = \gamma_{l,3}\left(1 + \frac{\tau_3}{\varepsilon+\beta_l}\right),\quad \tau_3 = \left(\frac{|\beta_3-\beta_1|+|\beta_3-\beta_2|}{2}\right)^2. \tag{2.6}$$

Here ε is chosen as 10^{-6} to avoid zero in the denominator.

Finally, we get the reconstruction polynomial over the cell I_i

$$u^i(x, y, z) = \sum_{l=1}^{3}\omega_l p^i_l(x, y, z). \tag{2.7}$$

2.3 Clipping

The primary principle of the intersection-based remapping algorithm is to determine precisely the intersections between the new and old meshes. The cost of computing intersections is rather high when compared to the flux-based remapping algorithm, which is the main drawback of the intersection-based remapping method. However, it has almost no restrictions on the relationship between the old and new meshes, and it is easier to achieve high-order accuracy since the integration error on the intersections obtained by an exact clipping method is close to the machine zero. Because of the aforementioned benefits, the intersection-based remapping method is more flexible in the ALE framework.

In this subsection, we will show how to use the clipping approach in our remapping process. Powell and Abel [19] proposed an exact clipping algorithm for clipping any two convex polyhedrons and we will adopt this exact clipping algorithm in this paper. Assume we have a new tetrahedral cell $\tilde{I}_j$ named 'target cell' and an old tetrahedral cell I_i named 'window cell'. Calculating the intersection of the target cell and the window cell is what we need to perform. The fundamental component of the clipping technique for the 3D tetrahedral meshes is identical to that of the clipping technique adopted in the 2D quadrilateral meshes [4, 15]. Both specify the window cell's visible and invisible sides, then clip the target cell using each face of the window cell in turn. Now we will go over the flow chart for clipping the target tetrahedral cell $\tilde{I}_j$ with one of the window cell's plane face P. Notice that the barycenter of the window cell W_c must be in the visible side of the plane face P, so the points which have the same signs of the directional distance with the barycenter W_c are in the visible side, otherwise, they are in the invisible side.

1. We calculate the directional distance d_l between the plane face P and each vertex $\tilde{v}_l, l = 1, 2, 3, 4$ of the target tetrahedral cell $\tilde{I}_j$. After that, we calculate the directional distance d_c between the plane face P and the barycenter of the window cell W_c.

2. If the signs of the directional distance at both ends of an edge are different, which indicates that they are on the different sides of the plane face P, we put them in the list of the visible points and the list of the invisible points, respectively. Then we calculate the crosspoint $\tilde{v}_0$ between that edge and the plane face P, and put the crosspoint $\tilde{v}_0$ to the list of the visible points.

3. Once a new point $\tilde{v}_0$ is added, we will connect $\tilde{v}_0$ with the other points in the list of the visible points.

4. We remove all the points in the list of the invisible points.

5. Finally, we get a new polyhedron within the visible side of the plane face P.

By performing the above processes in turn for each plane face of the window cell I_i, we will find the intersection polyhedron of the target cell $\tilde{I}_j$ and the window cell I_i.

In the schematic Figure 2.2, we mark the target cell in blue and the window cell in green. For each iteration, we utilize one face in the window cell to cut against the target cell and gain a new visible polyhedron marked in red with the common face marked in black, which will act as the new target cell in the next iteration.

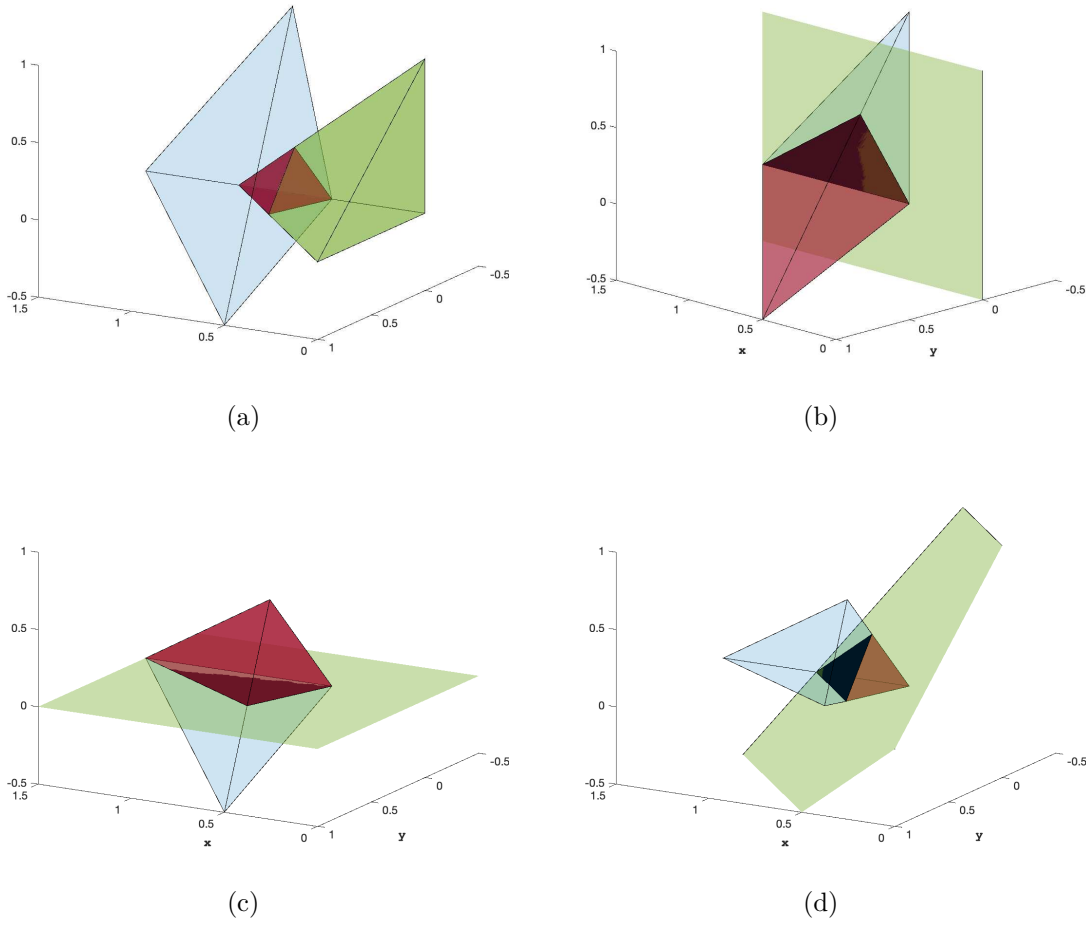

Figure 2.2: The clipping procedure. The blue polyhedron is the target cell, the green polyhedron is the window cell, the red polyhedron is the clipping result and the black face is the common face.

The error of the clipping method is close to machine zero and is unaffected by the mesh size or other mesh qualities. Therefore our intersection-based remapping process is simpler to achieve high-order precision, particularly when the mesh is extensively deformed.

2.4 Numerical integration

After the high-order reconstruction step and the clipping step, we only need to calculate the new cell averages $\bar{\tilde{u}}_j$ on the new cell $\tilde{I}_j$ by integrating the reconstruction polynomials $u^i(x,y,z)$ over the intersections $\tilde{I}_j \bigcap I_i$ for $i = 1, \cdots, N$,

$$\bar{\tilde{u}}_j = \frac{1}{|\tilde{I}_j|} \sum_{i=1}^{N} \int_{\tilde{I}_j \bigcap I_i} u^i(x,y,z) dxdydz. \tag{2.8}$$

The intersection $\tilde{I}_j \bigcap I_i$ of any two tetrahedral cells may have a complicated shape, which makes the numerical integration difficult to compute. Actually, by linking the barycenter with the vertices of the intersection, we divide the intersection into a few tetrahedrons and integrate $u^i(x,y,z)$ on each of these tetrahedrons. The above solution only works for convex polyhedrons, for non-convex polyhedrons, care must be taken to divide them into several tetrahedrons by connecting the vertices. In practice, we have designed a loop, and each time we cut off one piece of the tetrahedral by connecting the vertices from the non-convex polyhedron until we end up with a tetrahedral.

We will then perform a numerical integration on the tetrahedral T, which is required to be exact for polynomials of degree 2 in our third order remapping method,

$$\int_T u^i(x,y,z) dxdydz = \frac{|T|}{6} \sum_{l=1}^{L} \omega_l u^i(x_l, y_l, z_l), \quad \omega_l \geq 0, \tag{2.9}$$

where we take the $L = 4$ quadrature points proposed in [10]. Suppose P_1, P_2, P_3, P_4 are the vertices of the tetrahedron T, then the four quadrature points have the following coordinates

$$\begin{aligned}
(x_1, y_1, z_1) &= 0.5854101966249685 \times P_1 + 0.1381966011250105 \times (P_2 + P_3 + P_4), \\
(x_2, y_2, z_2) &= 0.5854101966249685 \times P_2 + 0.1381966011250105 \times (P_1 + P_3 + P_4), \\
(x_3, y_3, z_3) &= 0.5854101966249685 \times P_3 + 0.1381966011250105 \times (P_1 + P_2 + P_4), \\
(x_4, y_4, z_4) &= 0.5854101966249685 \times P_4 + 0.1381966011250105 \times (P_1 + P_2 + P_3),
\end{aligned}$$

and the weights are $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 0.25$. In Figure 2.3, we show the quadrature points in the tetrahedral with the red color. When using this high-order quadrature rule, the integration error can be ignored because the degree of the reconstruction polynomials is at most 2. The remapping method without the positivity-preserving modification phase

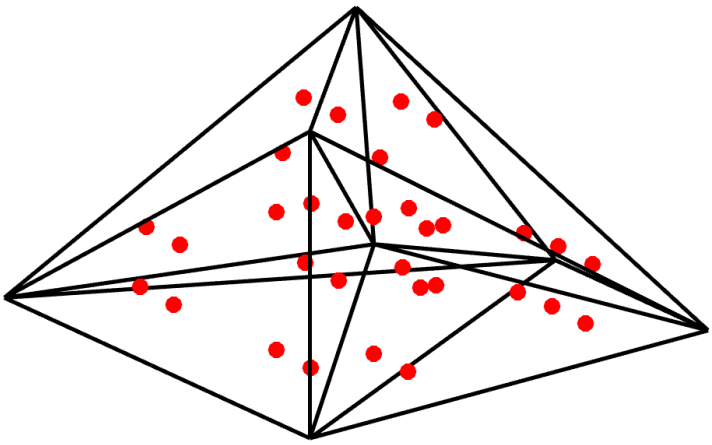

Figure 2.3: Schematic figure of the quadrature rule. The red points are the quadrature points.

has now been completed, and we will refer to it as the 'WENO-Remap' algorithm in the following.

We can easily prove that the WENO-Remap algorithm is conservative, module the round-off error in the clipping step and the numerical integration which is mathematically exact for our reconstruction polynomials,

$$\begin{aligned} \sum_{i=1}^{N} \bar{u}_i |I_i| &= \sum_{i=1}^{N} \int_{I_i} u^i(x,y,z)dxdydz \\ &= \sum_{i=1}^{N} \sum_{j=1}^{\tilde{N}} \int_{I_i \bigcap \tilde{I}_j} u^i(x,y,z)dxdydz \\ &= \sum_{j=1}^{\tilde{N}} \sum_{i=1}^{N} \int_{I_i \bigcap \tilde{I}_j} u^i(x,y,z)dxdydz \\ &= \sum_{j=1}^{\tilde{N}} \bar{\tilde{u}}_j |\tilde{I}_j|. \end{aligned} \tag{2.10}$$

2.5 Positivity-preserving modification

After the reconstruction step, we obtain the high-order polynomials $u^i(x,y,z)$ satisfying $\int_{I_i} u^i(x,y,z)dxdydz = \bar{u}_i|I_i|$. However, when the cells are close to the local extrema and near zero, these high-order polynomials may generate negative cell averages on the new mesh, which might violate physical principles. As a result, the above WENO-Remap technique

requires a positivity-preserving modification step for the physically positive variables, which should maintain both positivity and the original high order accuracy in smooth regions.

To modify the high-order polynomials $u^i(x,y,z)$ without violating conservation and high-order accuracy, a positivity-preserving limiter is required [15, 20, 21]. The core concept of this limiter is to compress the reconstruction polynomial $u^i(x,y,z)$ toward its non-negative cell average $\bar{u}_i \geq 0$ until the polynomial's minimum across the cell I_i is greater than ε,

$$\begin{aligned} &\tilde{u}^i(x,y,z) = \theta u^i(x,y,z) + (1-\theta)\bar{u}_i, \\ &\theta = \min\left\{1, \frac{|\bar{u}_i - \varepsilon|}{|\bar{u}_i - m|}\right\}, \quad m = \min_{(x,y,z)\in I_i} u^i(x,y,z), \end{aligned} \tag{2.11}$$

where ε is a very small positive constant satisfying $\bar{u}_i \geq \varepsilon$. In our numerical tests we take $\varepsilon = 10^{-14}$. It is proved in [20, 21] that this limiter maintains the original high order accuracy. One thing to keep in mind is that we can only make this adjustment when the cell averages are not negative $\bar{u}_i \geq 0$. For example, in the ALE framework, the cell averages calculated by the Lagrangian step should be non-negative because they are the input of the next remapping step. If we need to remap continually for several times, the positivity-preserving modification step should be applied in every remapping step.

As shown in Remark 2.1 of [15], the above modification methods can preserve $\tilde{u}^i(z,y,z) \geq \varepsilon$ for all $(x,y,z) \in I_i$, and these new polynomials are still conservative since

$$\begin{aligned} \int_{I_i} \tilde{u}^i(x,y,z)dxdydz &= \theta \int_{I_i} u^i(x,y,z)dxdydz + (1-\theta)|I_i|\bar{u}_i \\ &= \theta|I_i|\bar{u}_i + (1-\theta)|I_i|\bar{u}_i \\ &= |I_i|\bar{u}_i \\ &= \int_{I_i} u^i(x,y,z)dxdydz. \end{aligned}$$

In the following numerical experiments, we will verify that this adjustment can keep the original high-order accuracy.

Finding the minimum of $u^i(x,y,z)$ over cell I_i is not practical in reality and the cost may be prohibitive. The numerical integration is made up of the values of the polynomials at the quadrature points, according to the quadrature rule (2.9). So, as indicated in [20, 21], we will

replace the minimum over the whole cell $m = \min_{(x,y,z)\in I_i} u^i(x,y,z)$ by the minimum over such quadrature points $m = \min_{(x_l,y_l,z_l)\in Ga(I_i)} u^i(x_l,y_l,z_l)$, where $Ga(I_i)$ is the set of the quadrature points over the cell I_i. Conservation and high order accuracy are still maintained by this change, as indicated in [15, 20, 21].

By following the flow chart in Subsection 2.1, we have created a third-order accurate positivity-preserving conservative WENO remapping method. We have presented the numerical integration step after the positivity-preserving modification step in Subsection 2.1, which is the correct order of the algorithm. However, to save computational cost, we could also first perform a preliminary numerical integration step, and then perform the positivity-preserving modification only in those cells of the old mesh which intersect with any new cell with a negative cell average, followed by a recomputation of the numerical integration over all affected intersections. In most cases there are only a very small number of cells in the new mesh which have negative cell averages, thus this alternative procedure may lead to a significant saving of computational cost.

In the next section we will present several positivity-preserving tests to see if our remapping approach is third-order accurate with or without this modification step. In addition, discontinuity test, dissipation test and cost analysis will also be shown in the next section.

3 Numerical results

In this section, we will show numerical tests to verify the performance of our remapping algorithm on tetrahedral meshes. Suppose (x_p, y_p, z_p) is the coordinate of an interior node of the computational domain Ω, and $h = \min_{I_i\in\Omega} h_i$ is the minimum of the diameter h_i, where h_i is the circumscribed sphere's diameter of the tetrahedral I_i. In order to represent the remapping procedure, we design two different mesh movements to emulate the rezoned mesh from the ALE method.

1. The randomly moving mesh $(x_p, y_p, z_p)_R^t$,

$$(x_p, y_p, z_p)_R^{t+1} = (x_p, y_p, z_p)^0 + c_R h (r_x, r_y, r_z)^t.$$

2. The smoothly moving mesh $(x_p, y_p, z_p)_S^t$,

$$\begin{aligned} x_{p,S}^{t+1} &= x_p^0 + c_S h \tfrac{\max\{t, T-t\}}{T} \sin(\tfrac{\pi}{2} x_p^0) \\ y_{p,S}^{t+1} &= y_p^0 + c_S h \tfrac{\max\{t, T-t\}}{T} \sin(\tfrac{\pi}{2} y_p^0) \\ z_{p,S}^{t+1} &= z_p^0 + c_S h \tfrac{\max\{t, T-t\}}{T} \sin(\tfrac{\pi}{2} z_p^0). \end{aligned}$$

The superscript t represents the remapping times, for example, we use $(x_p, y_p, z_p)^0$ to represent the origin coordinate. $r_x, r_y, r_z \in [-1, 1]$ are random numbers with uniform distribution, t, T are the current remapping times and the total remapping times. In our numerical test, the constants c_R, c_S are taken as $c_R = 0.1, c_S = 0.5$.

3.1 Accuracy test

First, we verify the high-order accuracy of our remapping algorithm. Suppose the original cell averages are calculated by the function

$$u_0(x, y, z) = \cos^2(\frac{\pi}{2} x) \cos^2(\frac{\pi}{2} y) \cos^2(\frac{\pi}{2} z), \quad -2 \le x, y, z \le 2. \tag{3.1}$$

Initially, the computational domain $\Omega = [-2, 2] \times [-2, 2] \times [-2, 2]$ is equally divided into small cubes with mesh size $d = \frac{4}{N_x}$, where $N_x = N_y = N_z$ are the number of cells in the x, y, z directions, then each small cube is divided into six tetrahedrons with the same volume. After remapping T times on the randomly moving mesh or the smoothly moving mesh, we require the final mesh to move back to the original mesh in the accuracy tests, to compare the remapping results.

Figure 3.1 shows the profiles of the randomly moving mesh and the smoothly moving mesh with 6,000 tetrahedral cells at $z = 0$, respectively. Figure 3.2 is the three dimensional view of these meshes without the first layer in the z direction.

The original cell average over I_i is obviously positive $\bar{u}_i^0 = \frac{1}{|I_i|} \int_{I_i} u_0(x, y, z) dx dy dz > 0$, that means the final remapping results $\bar{u}_i^T$ should also be positive after being remapped T times.

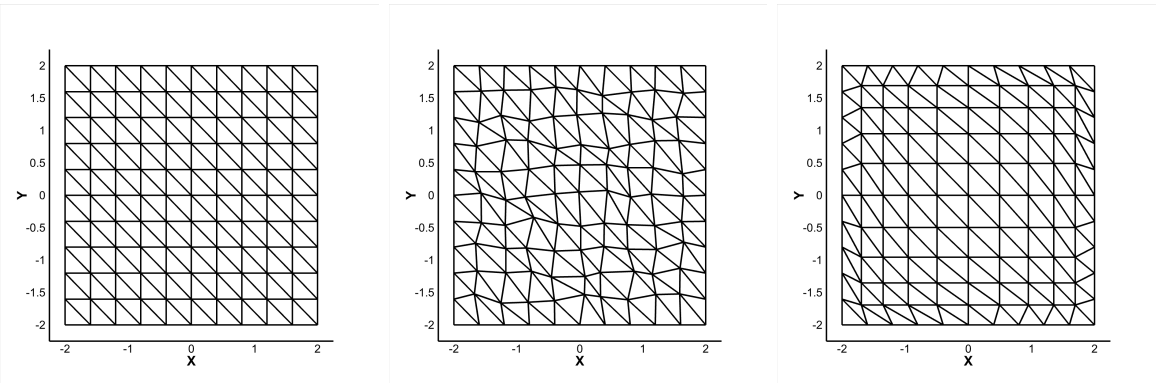

Figure 3.1: Profiles of the mesh at $z = 0$. Left: the initial mesh; Middle: the randomly moving mesh; Right: the smoothly moving mesh.

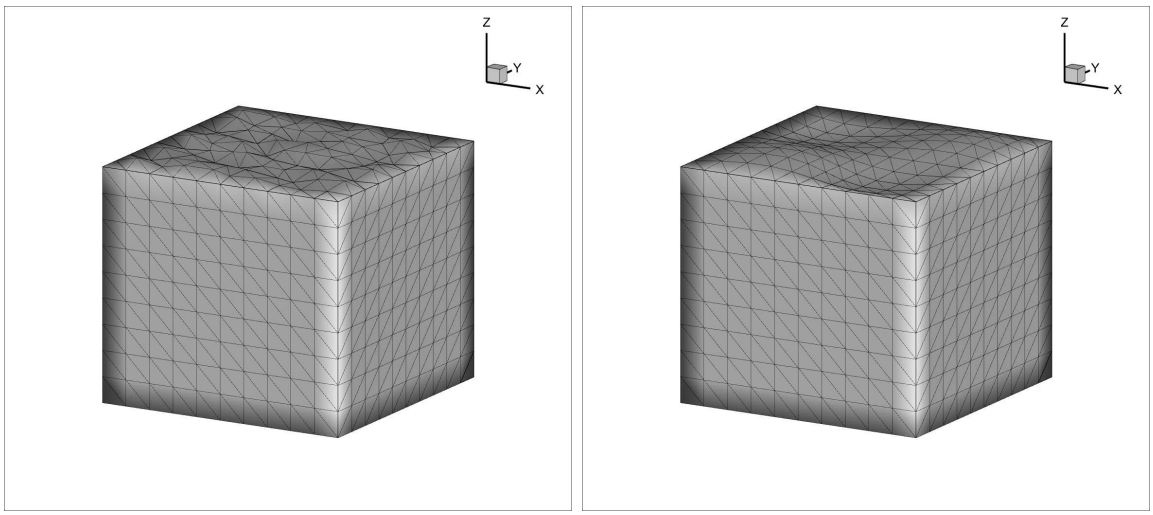

Figure 3.2: Three dimensional view of the tetrahedral meshes without the first layer in the z direction. Left: the randomly moving mesh; Right: the smoothly moving mesh.

We compare the performance of our multi-resolution WENO remapping algorithm with or without the positivity-preserving modification and these two algorithms are denoted as 'WENO-Remap' and 'P-WENO-Remap' in the following. Table 3.1 shows the L^1 and L^∞ norms of the remapping error on the randomly moving mesh with $T = 10$ remapping times, and Table 3.2 shows the corresponding results on the smoothly moving mesh. We use Error_c to represent the conservation error

$$\text{Error}_c = \sum_{i=1}^{N} |I_i| \cdot \left| \bar{u}_i^T - \bar{u}_i^0 \right|,$$

and use 'PP' to represent the percentage of the cells being modified in the positivity-preserving step.

Table 3.1: Accuracy test: error and order of the WENO-Remap and P-WENO-Remap algorithms on the randomly moving tetrahedral meshes with $T = 10$.

	WENO-Remap					
N	L^1 error	order	L^∞ error	order	Error_c	PP(%)
750	1.4365E-2		1.3841E-1		2.6645E-14	-
6000	3.6379E-3	1.98	5.2788E-2	1.39	2.8422E-14	-
20250	9.6994E-4	3.26	2.3215E-2	2.03	1.9540E-14	-
48000	3.5056E-4	3.54	9.5046E-3	3.10	1.4566E-13	-
	P-WENO-Remap					
N	L^1 error	order	L^∞ error	order	Error_c	PP(%)
750	1.4431E-2		1.3874E-1		1.8652E-14	0.22
6000	3.6377E-3	1.99	5.2787E-2	1.39	1.2434E-14	1.91
20250	1.0084E-3	3.16	2.3215E-2	2.03	2.3981E-14	2.65
48000	4.1878E-4	3.05	9.5046E-3	3.10	2.7001E-13	1.74

From Table 3.1 and Table 3.2, we observe that our multi-resolution WENO remapping algorithm achieves the designed third-order accuracy on both randomly moving and smoothly moving meshes, regardless of whether the positivity-preserving modification step is involved. According to the numerical results, we can see that about 2% of the cells may emerge negative cell averages during the remapping and there is no significant difference of the remapping error between the WENO-Remap and the P-WENO-Remap algorithms, that

Table 3.2: Accuracy test: error and order of the WENO-Remap and P-WENO-Remap algorithms on the smoothly moving tetrahedral meshes with $T = 10$.

	WENO-Remap					
N	L^1 error	order	L^∞ error	order	Error$_c$	PP(%)
750	1.0261E-2		5.8618E-2		2.3093E-14	-
6000	2.0097E-3	2.35	1.0261E-2	2.51	4.0856E-14	-
20250	4.5928E-4	3.64	2.8426E-3	3.17	4.2633E-14	-
48000	1.6958E-4	3.46	1.2820E-3	2.77	2.1316E-13	-
	P-WENO-Remap					
N	L^1 error	order	L^∞ error	order	Error$_c$	PP(%)
750	1.0347E-2		5.8640E-1		3.3751E-14	0.22
6000	1.9876E-3	2.38	1.0277E-2	2.51	8,8818E-15	2.07
20250	4.8623E-4	3.47	2.8426E-3	3.17	7.9937E-14	3.05
48000	2.1533E-4	2.83	1.3354E-3	2.63	2.7445E-13	2.08

means our positivity-preserving modification step will not destroy the original high-order accuracy. The conservation error Error$_c$ in these tables are small enough to be neglected.

To verify that our remapping method has no limits on the mesh movements, where the flux-based method requires the node movement should not exceed the size of its neighboring cells, we still divide Ω into small cubes, but we divide unequally in the x direction with mesh size

$$d_1^x < d_2^x < \cdots < d_{N_x}^x,\ d_{N_x}^x = 2d_1^x,$$

and $N_x = N_y = N_z$ are the number of cells in each direction. Then, we design a flipping mesh with mesh size

$$\tilde{d}_1^x = d_{N_x}^x, \cdots, \tilde{d}_{N_x}^x = d_1^x.$$

The y and z directions are still divided equally $d^y = \tilde{d}^y = \frac{4}{N_y}$, $d^z = \tilde{d}^z = \frac{4}{N_z}$. Figure 3.3 shows the profiles of these two meshes with 6,000 tetrahedral cells at $z = 0$. This time, we remap from the initial mesh to the flipping mesh and return to the initial mesh for T times.

Table 3.3 shows the remapping results for the WENO-Remap algorithm and the P-WENO-Remap algorithm. Both of them are conservative and third-order accurate.

Above all, the accuracy tests verify that our positivity-preserving multi-resolution WENO

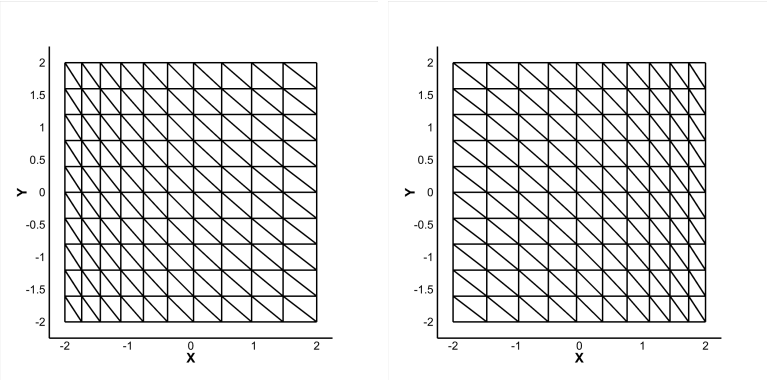

Figure 3.3: Profiles of the mesh at $z = 0$. Left: the initial mesh; Right: the flipping mesh in the x direction.

Table 3.3: Accuracy test: error and order of the WENO-Remap and P-WENO-Remap algorithms on the flipping tetrahedral meshes with $T = 10$.

	WENO-Remap					
N	L^1 error	order	L^∞ error	order	Error_c	PP(%)
750	3.9356E-02		2.5904E-01		5.3291E-15	-
6000	1.2791E-02	1.62	1.2652E-01	1.03	2.0428E-14	-
20250	3.5085E-03	3.19	4.9805E-02	2.30	2.7534E-14	-
48000	1.2536E-03	3.58	2.0021E-02	3.17	2.3981E-14	-
	P-WENO-Remap					
N	L^1 error	order	L^∞ error	order	Error_c	PP(%)
750	3.9433E-02		2.5902E-01		8.8818E-15	1.60
6000	1.3133E-02	1.59	1.2642E-01	1.03	2.0428E-14	4.33
20250	4.1491E-03	2.84	4.9695E-02	2.30	5.6843E-14	4.40
48000	1.5617E-03	3.40	2.1620E-02	2.89	2.1316E-14	2.59

remapping algorithm P-WENO-Remap is third-order accurate, positivity-preserving and conservative.

3.2 Discontinuity test

In this subsection, we test a function with discontinuity which is typical for the numerical simulation of hyperbolic equations. Inspired by [16], we put a ball and a cube in the computational domain $\Omega = [-2, 2] \times [-2, 2] \times [-2, 2]$. The center of the ball is $(1, 1, 1)$ with radius $R = 0.75$ and the center of the cube is $(-1, -1, -1)$ with the edge size $L = 1.5$,

$$u_0(x, y, z) = \begin{cases} 1 & r \leq 0.75 \\ 1 & -1.75 \leq x, y, z \leq -0.25 \\ 10^{-12} & \text{else} \end{cases}, \quad -2 \leq x, y, z \leq 2, \tag{3.2}$$

where $r = \sqrt{(x-1)^2 + (y-1)^2 + (z-1)^2}$. In this test, we use the same equally divided tetrahedral meshes as the accuracy test with 48,000 cells and remap on the randomly moving meshes for 10 times.

In Figure 3.4 and Figure 3.5, the blue cells represent the cell averages with $\bar{u}_i > 0.01$ and the red cells represent the negative cell averages with $\bar{u}_i < -10^{-4}$. The right subfigure of Figure 3.4 is based on the third-order reconstruction polynomial $q_3^i(x, y, z)$, and these remapping results do not involve the WENO method and the positivity-preserving modification. Figure 3.5 shows the remapping results of the third-order WENO-Remap and the P-WENO-Remap methods.

As one can see, there are fewer red cells in the remapping results of the third-order WENO-Remap method (left of Figure 3.5), compared with the third-order remapping results (right of Figure 3.4), because the numerical oscillations produced from the high-order reconstruction have been largely eliminated by the WENO method. After adding the positivity-preserving modification phase, there are no negative cell averages in the remapping results of the P-WENO-Remap method (right of Figure 3.5).

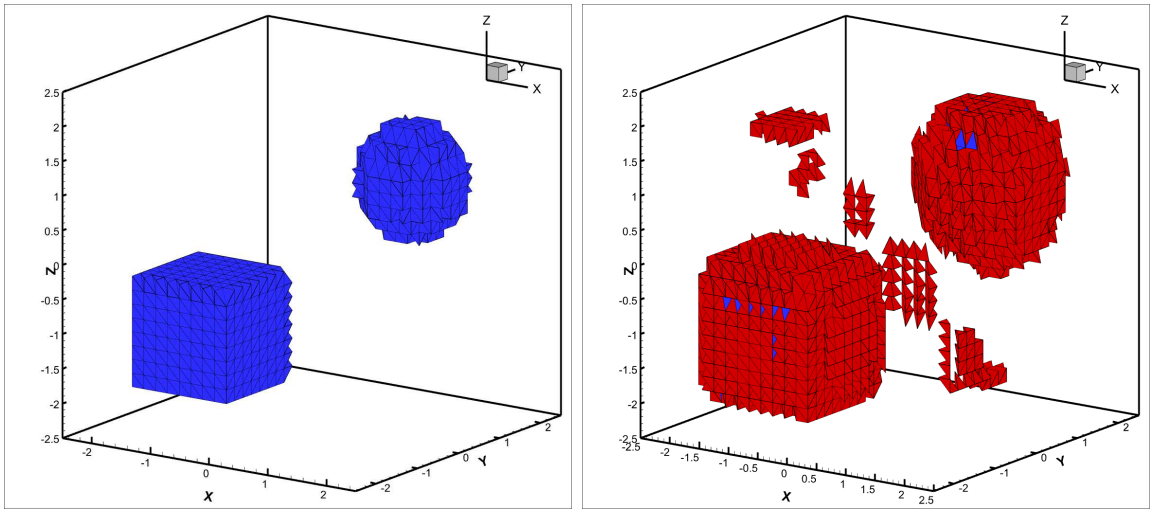

Figure 3.4: Discontinuity test: we mark the tetrahedral cells in blue with cell averages $\bar{u}_i > 0.01$ and mark the tetrahedral cells in red with cell averages $\bar{u}_i < -10^{-4}$. Left: the initial cell averages; Right: the remapping results of the third-order reconstruction polynomial $q_3^i(x, y, z)$.

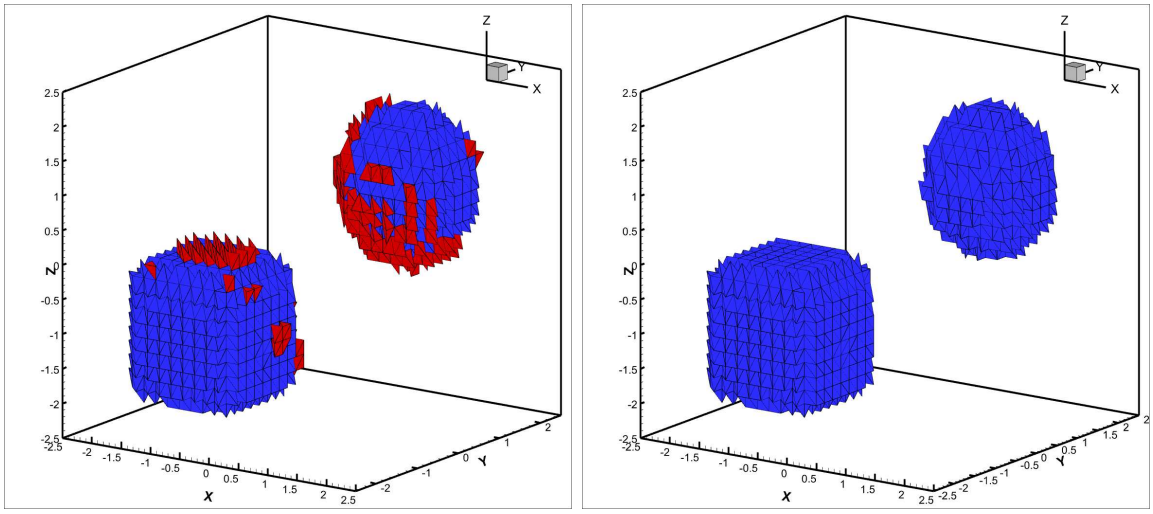

Figure 3.5: Discontinuity test: we mark the tetrahedral cells in blue with cell averages $\bar{u}_i > 0.01$ and mark the tetrahedral cells in red with cell averages $\bar{u}_i < -10^{-4}$. Left: the third-order WENO-Remap remapping results; Right: the third-order P-WENO-Remap remapping results.

3.3 Dissipation test

In this subsection, we compare the first-order reconstruction polynomials $q_1^i(x, y, z)$, denoted as 'P0' and our third-order P-WENO reconstruction remapping method with the positivity-preserving modification. To do this, we give an initial function:

$$u_0(x, y, z) = e^{-\alpha r^2}, \quad r = \sqrt{x^2 + y^2 + z^2}, \tag{3.3}$$

where the computational domain and the mesh are the same as those in Subsection 3.2 and we remap on the randomly moving meshes for 50 times with 6,000 cells. We take only 10 cells on the cut line $y = z = 0$ and plot the reconstruction polynomials with constant $\alpha = 0.5$. As one can see from Figure 3.6, there are different degrees of dissipation near the origin after remapping, but the dissipation of the higher-order algorithm noted as 'P-WENO' in the figure is obviously smaller than that of the lower-order algorithm noted as 'P0'. In the meantime, our remapping algorithm can maintain sharper edge transition, indicating a smaller numerical dissipation by the high order remapping method.

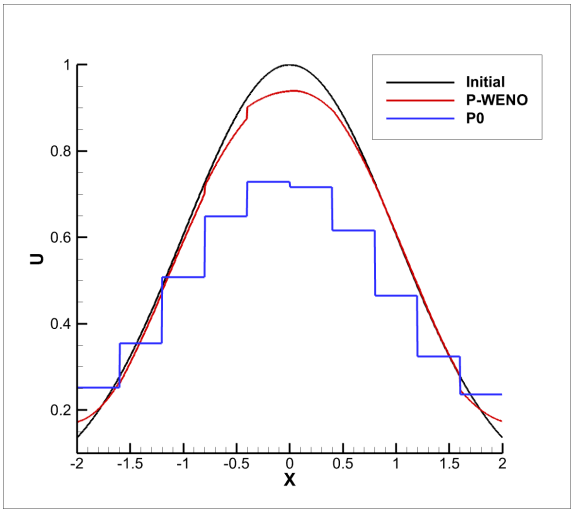

Figure 3.6: The dissipation test on the function (3.3) by the P0 and P-WENO-Remap remapping methods, reconstruction polynomials on the cut line $y = z = 0$.

3.4 Positivity-preserving tests

In this subsection, we verify our remapping algorithm P-WENO-Remap can preserve positivity for the physically positive variables such as density or internal energy on different types of tetrahedral meshes.

First, we design a 'ball' function

$$u_0(x,y,z) = \begin{cases} 1 & r \le 0.8 \\ 10^{-12} & r > 0.8 \end{cases}, \ r = \sqrt{(x-1)^2+(y-1)^2+(z-1)^2}, \ 0 \le x,y,z \le 2. \quad (3.4)$$

This time, we use a more general unstructured tetrahedral mesh with 5,800 cells and remap on the randomly moving meshes for 10 times. Since such a discontinuous ball function is easy to become negative near the discontinuity during the remapping process, it is necessary to add the positivity-preserving modification step so that the result after remapping preserves positivity which is essential in the simulation of computational fluid dynamics.

Figure 3.7 shows the initial cell averages in the three-dimensional view and the two-dimensional cut planes at $x = 0$ and $y = 0$. Figure 3.8 and Figure 3.9 show the remapping results of the WENO-Remap and the P-WENO-Remap remapping algorithms, respectively. The white symbols near the discontinuity in Figure 3.8 represent the cells in which the averages are negative without the positivity-preserving modification. It can be seen that our P-WENO-Remap algorithm achieves good positivity-preserving property. In the meantime, our remapping does a good job on maintaining a sharp discontinuity of the ball function (3.4).

Next, we dig a ball of radius 1.4 centered at $(0,0,0)$ in the cube computational domain $[-2,2]\times[-2,2]\times[-2,2]$ and design a positive initial function

$$u_0(x,y,z) = \begin{cases} 10^{-12} & r \le 1.8 \\ r & r > 1.8 \end{cases}, \quad r = \sqrt{x^2+y^2+z^2}, \ -2 \le x,y,z \le 2. \quad (3.5)$$

Just as before, we move the interior nodes randomly for 10 times and return to the initial tetrahedral mesh. We only show the part of the computational domain that is in $[0,2]\times[0,2]\times[0,2]$ and the two-dimensional cut planes at $x = 0$ and $y = 0$, in Figure 3.10, 3.11

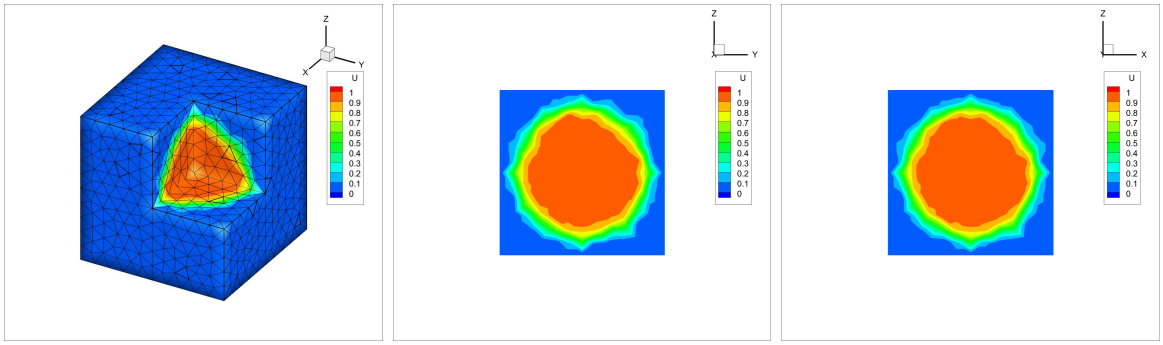

Figure 3.7: Positivity-preserving test on the ball function (3.4). Left: the 3D view of the initial cell averages $\{\bar{u}_i^0\}_{i=1}^N$; Middle: the 2D cut plane at $x = 0$; Right: the 2D cut plane at $y = 0$.

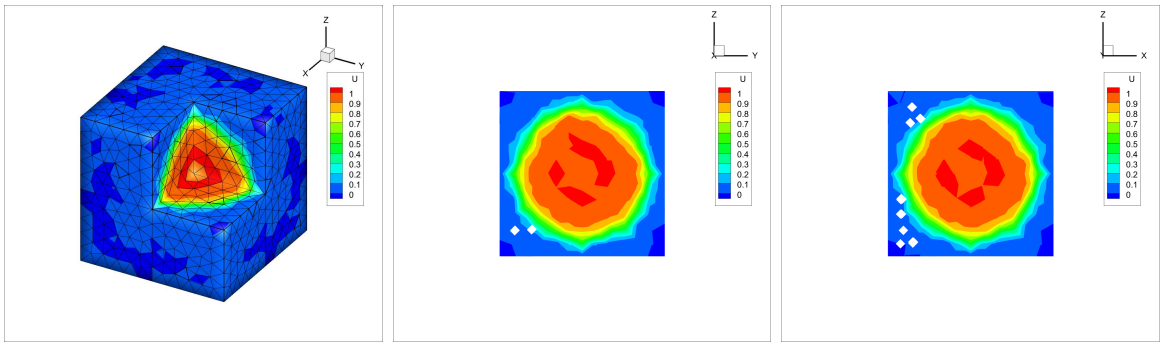

Figure 3.8: Positivity-preserving test on the ball function (3.4). Left: the 3D view of the cell averages $\{\bar{u}_i^{10}\}_{i=1}^N$ after remapping 10 times by the WENO-Remap procedure; Middle: the 2D cut plane at $x = 0$; Right: the 2D cut plane at $y = 0$. White symbols represent the cells where the cell-averages are negative without the positivity-preserving modification.

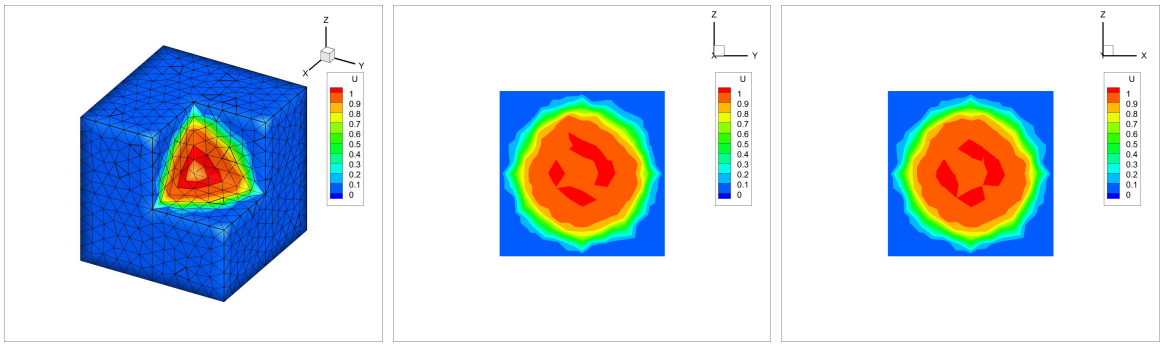

Figure 3.9: Positivity-preserving test on the ball function (3.4). Left: the 3D view of the cell averages $\{\bar{u}_i^{10}\}_{i=1}^N$ after remapping 10 times by the P-WENO-Remap procedure; Middle: the 2D cut plane at $x = 0$; Right: the 2D cut plane at $y = 0$.

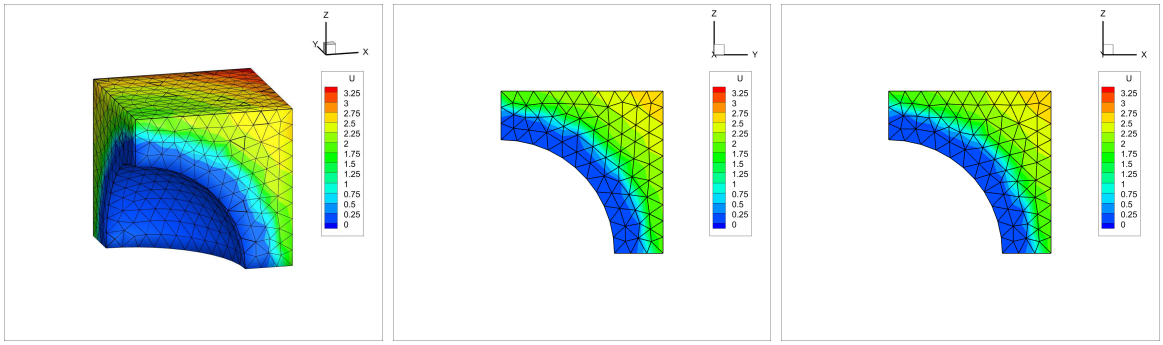

Figure 3.10: Positivity-preserving test on the function (3.5) in $[0,2]\times[0,2]\times[0,2]$. Left: the 3D view of the initial cell averages $\{\bar{u}_i^0\}_{i=1}^N$; Middle: the 2D cut plane at $x=0$; Right: the 2D cut plane at $y=0$.

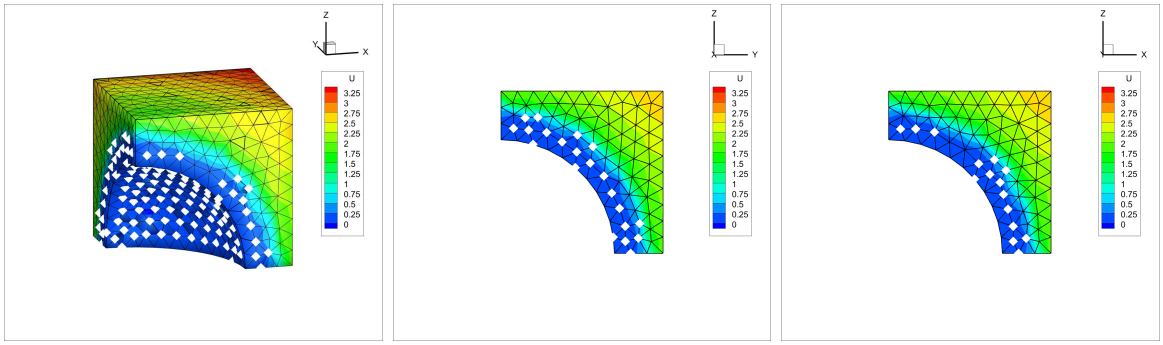

Figure 3.11: Positivity-preserving test on the function (3.5) in $[0,2]\times[0,2]\times[0,2]$. Left: the 3D view of the initial cell averages $\{\bar{u}_i^{10}\}_{i=1}^N$ after remapping 10 times by the WENO-Remap procedure; Middle: the 2D cut plane at $x=0$; Right: the 2D cut plane at $y=0$. White symbols represent the cells where the cell-averages are negative without the positivity-preserving modification.

and 3.12. Near the discontinuity, there are many negative cell averages marked in white in Figure 3.11 without the positivity-preserving modification. On the other hand, our P-WENO-Remap algorithm preserves positivity property well on this special computational domain.

3.5 Cost analysis

In this subsection, we will compare the costs of the four parts in our remapping algorithm. We denote the cost of the high-order multi-resolution WENO reconstruction step, the clipping step, the numerical integration step and the positivity-preserving modification step in the P-WENO-Remap and the WENO-Remap algorithms as 'WENO', 'Clip', 'Int' and 'Pos',

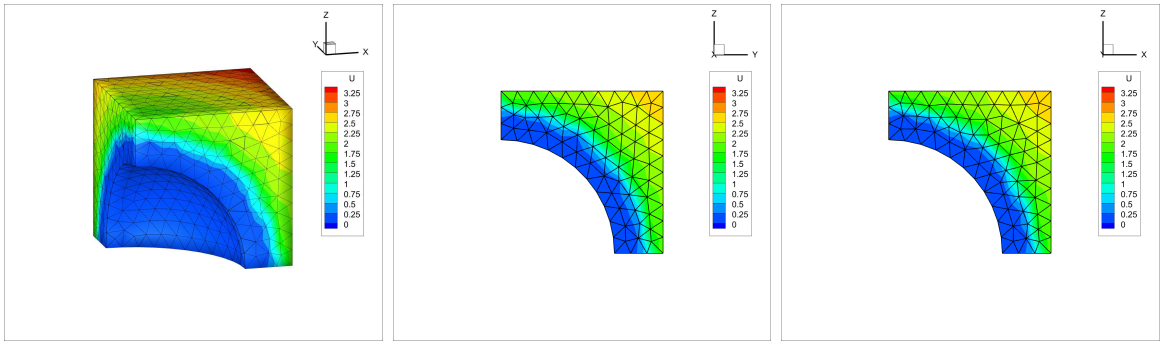

Figure 3.12: Positivity-preserving test on the function (3.5) in $[0,2]\times[0,2]\times[0,2]$. Left: the 3D view of the initial cell averages $\{\bar{u}_i^{10}\}_{i=1}^N$ after remapping 10 times by the P-WENO-Remap procedure; Middle: the 2D cut plane at $x=0$; Right: the 2D cut plane at $y=0$.

respectively. Compared with the P-WENO-Remap algorithm, the WENO-Remap algorithm consists of the first three steps without the 'Pos' step.

Table 3.4 shows the cost of the WENO-Remap, P-WENO-Remap algorithms and their separate parts with different mesh scales. The second to last column 'Ratio' means the ratio of the total cost of the P-WENO-Remap algorithm and that of the WENO-Remap algorithm. The meaning of 'PP' in the last column has been mentioned in Subsection 3.1. The cost of the numerical integration step is much less than the other three parts. For a sufficiently refined mesh, the costs of the 'Clip' part and the 'WENO' part are roughly equivalent, that means the cost of the clipping step is not too high towards the total cost. After the introduction of the positivity-preserving modification step, the cost of the P-WENO-Remap algorithm is about 5% more than the cost of the WENO-Remap algorithm. Overall, we claim that our modification step can achieve the positivity-preserving property efficiently.

Table 3.4: The cost of the WENO-Remap, P-WENO-Remap algorithm and their separate parts.

N	WENO	Clip	Int	Pos	WENO-Remap	P-WENO-Remap	Ratio	PP(%)
750	6.77E-2	1.67E-1	8.95E-3	5.00E-3	2.44E-1	2.49E-1	1.02	0.22
6000	8.70E-1	2.14E+0	4.68E-2	4.03E-1	3.06E+0	3.46E+0	1.13	2.07
20250	5.58E+0	8.88E+0	2.40E-1	1.09E+0	1.47E+1	1.58E+1	1.07	3.05
48000	2.40E+1	2.46E+1	4.75E-1	2.33E+0	4.91E+1	5.14E+1	1.05	2.08

4 Concluding remarks

Following our previous work [15], we extend our high order positivity-preserving conservative WENO remapping technique from 2D quadrilateral meshes to 3D tetrahedral meshes. In comparison with the flux-based remapping algorithm, our intersection-based remapping algorithm has a broader application, allowing it to cope with more complicated situations and is more flexible in the ALE method framework.

We reconstruct three polynomials of distinct orders from three nested central stencils, and mix them with nonlinear weights impacted by the smoothness of the polynomials using the multi-resolution WENO reconstruction procedure. All of the values of smoothness indicators are small in smooth regions, causing the nonlinear weights to approach the linear weights and the final reconstruction polynomial to attain the designed third-order accuracy. Higher-order polynomials have considerably larger values of smoothness indicators than the lower-order ones in non-smooth regions, hence the latter plays a role in the final reconstruction polynomial to avoid numerical oscillations. We obtain the intersections of the new and old cells by clipping every new mesh cell against all of the old mesh cells, then integrate the reconstruction polynomials over these intersections exactly by suitable numerical quadratures to produce the required cell averages on the new meshes. This remapping method is conservative, and can yield third-order results, since the errors of the clipping algorithm and the quadrature rule can be ignored. Then, to retain the positivity property for the relevant physical variables such as density and internal energy, which are common when using the ALE method to solve fluid flow problems, we add a positivity-preserving modification step. This adjustment is conservative and preserves the origin high order accuracy.

Numerical experiments on three-dimensional tetrahedral meshes show that our remapping approach is conservative, third-order accurate, non-oscillatory, and positivity-preserving. To test the accuracy of our remapping approach, we have designed two different mesh movement strategies, and the designed third-order accuracy can be observed with or without the positivity-preserving modification. Several special functions are designed for the discontinu-

ity test, dissipation test and positivity-preserving test, to show that our algorithm performs well in maintaining the sharpness of the discontinuity and positivity. Meanwhile, we compare the costs of different components of our algorithm, demonstrating that the clipping step is not overly expensive towards the total cost of the algorithm, and the positivity-preserving step accounts for only a small percentage of the total cost.

One of the major applications of the remapping algorithm developed in this paper is to serve for the remapping step of high order conservative positivity-preserving ALE methods. After solving the hyperbolic PDEs in the Lagrangian method, the mesh nodes move with the local fluid velocity which may make the mesh tangling or distorted at the shock front or in large gradient regions, especially for the three-dimensional PDEs. Therefore, a rezoning step is necessary to improve the mesh quality to complete the calculation and to maintain the original material interfaces. The new rezoned mesh may change significantly or may have different connectivity from the old mesh, making the remapping algorithm developed in this paper particularly suitable, which could faithfully maintain high order accuracy, non-oscillatory, positive and conservative properties of the original Lagrangian or ALE methods.

To be more specific, in the calculation of a hyperbolic PDEs such as the Euler equation, assume we have a new rezoned mesh $\{\tilde{I}_j\}$ and the cell averages of the physical variables such as density $\bar{\rho}_i$, momentum $\overline{\rho\vec{u}}_i$ and total energy $\bar{E}_i$, which are defined on the old distorted mesh $\{I_i\}$ after the Lagrangian step. We can apply the high order positivity-preserving WENO remapping method developed in this paper to remap these conservative physical variables one by one to the new mesh and obtain the new cell averages $\bar{\tilde{\rho}}_j$, $\overline{\widetilde{\rho\vec{u}}}_j$, $\bar{\tilde{E}}_j$ as described in Section 2. Positivity can be maintained both for the density and for the internal energy (or pressure) after remapping. Then we can use these new physical variables on the new mesh to solve the PDEs again in the next Lagrangian step. During the Lagrangian calculation, we can apply the rezoning step and the remapping step after several time steps, or whenever necessary to maintain a high mesh quality. The implementation of our remapping procedure in a three-dimensional ALE solver constitutes our future work.

References

[1] A. J. Barlow, P.-H. Maire, W. J. Rider, R. N. Rieben, M. J. Shashkov, *Arbitrary Lagrangian-Eulerian methods for modeling high-speed compressible multimaterial flows*, Journal of Computational Physics, 322 (2016), 603-665.

[2] D. E. Burton, N. R. Morgan, M. R. J. Charest, M. A. Kenamond, J. Fung, *Compatible, energy conserving, bounds preserving remap of hydrodynamic fields for an extended ALE scheme*, Journal of Computational Physics, 355 (2018), 492-533.

[3] X. Chen, X. Zhang, Z. Jia, *A robust and efficient polyhedron subdivision and intersection algorithm for three-dimensional MMALE remapping*, Journal of Computational Physics, 338 (2017), 1-17.

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

[5] V. P. Chiravalle, N. R. Morgan, *A 3D finite element ALE method using an approximate Riemann solution*, International Journal for Numerical Methods in Fluids, 83 (2017), 642-663.

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

[7] R. Garimella, M. Kucharik, M. Shashkov, *An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes*, Computers and Fluids, 36 (2007), 224-237.

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

[9] G. Jiang, C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), 202-228.

[10] P. Keast, *Moderate-degree tetrahedral quadrature formulas*, Computer Methods in Applied Mechanics and Engineering, 55 (1986), 339-348.

[11] M. Klima, M. Kucharik, M. Shashkov, *Local error analysis and comparison of the swept- and intersection- based remapping methods*, Communications in Computational Physics, 21 (2017), 526-558.

[12] M. Klima, M. Kucharik, M. Shashkov, *Combined swept region and intersection-based single-material remapping method*, International Journal for Numerical Methods in Fluids, 85 (2017), 363-382.

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

[14] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, M. Shashkov, *Hybrid remap for multi-material ALE*, Computers and Fluids, 46 (2011), 293-297.

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

[17] X.-D. Liu, S. Osher, T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), 200-212.

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

[19] D. Powell, T. Abel, *An exact general remeshing scheme applied to physically conservative voxelization*, Journal of Computational Physics, 297 (2015), 340-356.

[20] X. Zhang, 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.

[21] X. Zhang, 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.

[22] J. Zhu, C.-W. Shu, *A new type of third-order finite volume multi-resolution WENO schemes on tetrahedral meshes*, Journal of Computational Physics, 406 (2020), 109212.