FISEVIER

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



A robust and efficient polyhedron subdivision and intersection algorithm for three-dimensional MMALE remapping

CrossMark

Xiang Chen^a, Xiong Zhang^{a,*}, Zupeng Jia^b

^a School of Aerospace Engineering, Tsinghua University, Beijing 100084, China
 ^b Institute of Applied Physics and Computational Mathematics, Beijing 100094, China

ARTICLE INFO

Article history: Received 11 November 2016 Received in revised form 11 January 2017 Accepted 9 February 2017 Available online 28 February 2017

Keywords:

Multi-material ALE method Intersection-based remapping Polyhedron intersection Polyhedron subdivision

ABSTRACT

The Multi-Material Arbitrary Lagrangian Eulerian (MMALE) method is an effective way to simulate the multi-material flow with severe surface deformation. Comparing with the traditional Arbitrary Lagrangian Eulerian (ALE) method, the MMALE method allows for multiple materials in a single cell which overcomes the difficulties in grid refinement process. In recent decades, many researches have been conducted for the Lagrangian, rezoning and surface reconstruction phases, but less attention has been paid to the multi-material remapping phase especially for the three-dimensional problems due to two complex geometric problems: the polyhedron subdivision and the polyhedron intersection. In this paper, we propose a "Clipping and Projecting" algorithm for polyhedron intersection whose basic idea comes from the commonly used method by Grandy (1999) [29] and Jia et al. (2013) [34]. Our new algorithm solves the geometric problem by an incremental modification of the topology based on segment-plane intersections. A comparison with Jia et al. (2013) [34] shows our new method improves the efficiency by 55% to 65% when calculating polyhedron intersections. Moreover, the instability caused by the geometric degeneracy can be thoroughly avoided because the geometry integrity is preserved in the new algorithm. We also focus on the polyhedron subdivision process and describe an algorithm which could automatically and precisely tackle the various situations including convex, non-convex and multiple subdivisions. Numerical studies indicate that by using our polyhedron subdivision and intersection algorithm, the volume conversation of the remapping phase can be exactly preserved in the MMALE simulation.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Simulation of fluid with multiple-materials is a great challenge in computational mechanics and the choice of the computational grid is significant. Generally, there are two basic descriptions, the Lagrangian description and the Eulerian description. In Lagrangian description, the computational grid is embedded with material so that the material interface is always the grid boundary which can be tracked innately but the grid will be distorted when materials experience a large deformation. On the contrary, in the Eulerian description, the computational grid is fixed which overcomes the grid distortion but loses the surface information because it will advect across the cells during the simulation. In order to combine the merits of Lagrangian and Eulerian description, Hirt et al. developed an arbitrary Lagrangian–Eulerian (ALE) method [1]

http://dx.doi.org/10.1016/j.jcp.2017.02.029 0021-9991/© 2017 Elsevier Inc. All rights reserved.

^{*} Corresponding author. *E-mail address:* xzhang@tsinghua.edu.cn (X. Zhang).

which allows the motion of grid being defined as an independent degree of freedom. In ALE frame, the best properties of Lagrangian and Eulerian descriptions are preserved and many authors have demonstrated that ALE schemes are important due to its accuracy, robustness and efficiency [2–6].

The traditional ALE scheme requires the coincidence of the material surface and cell boundary which can be easily achieved when the material surface deforms slightly. However, when it deforms severely, it is very difficult to obtain a new grid with high quality and in some extreme cases where the topology of the surface changes, it is almost impossible to perform rezoning phase successfully. To solve the problem, Peery et al. developed a multi-material ALE (MMALE) method to simulate the strong shearing flow with severe surface deformation [7]. The main improvement of MMALE method is that it allows for multiple materials in a single cell so that the material surface need not to be coincided with the cell boundary. Therefore, the MMALE method can simulate problems involving severe grid and surface deformation which are very difficult for the traditional ALE method.

The major process of the MMALE is similar to the traditional ALE method. Firstly, the Lagrangian phase is performed to update the node position and material state [8–15] but in MMALE method, a closure model [16–19] is required to determine the pressure of the mixed cells. After several Lagrangian steps, the grid will be distorted and the rezoning phase [20–25] is performed to generate a new grid with high quality. Afterwards, the remapping phase will interpolate the variables, such as the density, internal energy, velocity, from the old distorted grid to the new rezoned grid and in the MMALE method, the information for surface reconstruction, such as the volume fraction and the material centroid, should also be interpolated. Finally, we discard the distorted grid and the Lagrangian phase will be restarted again on the rezoned grid.

Generally, the remapping phase can be constructed by two approaches: the "face-based" scheme [26–28] and the "intersection-based" scheme [29–31]. These two schemes are based on the geometrical relationship between the old and the rezoned grid. The "face-based" scheme is in a flux form and the flux is determined by the relative position between the new cell's boundary and its associated old cell's boundary. This scheme only considers the flux across the cell face and ignores the flux from the cell corner. Therefore, the "face-based" scheme is highly efficient and can be implemented easily. However, there are two major limitations of the "face-based" scheme. 1. The rezoned grid must be in the same topology of the old grid and close to it in order to find the correct flux region. 2. In the multi-material remapping process [32], the "face-based" scheme may lead to problematic situations, such as negative mass and inaccurate material fragmentation.

On the other hand, the "intersection-based" scheme interpolates the variables by calculating the intersection portions of the old and the rezoned grid. Each intersection piece will carry the information from the old grid and then be reassembled into the rezoned grid to obtain the variables on it [28]. There are many advantages in the "intersection-based" schemes: 1. The rezoned grid need not to be close to the old grid and their topology can even be different. 2. The materials' volume fraction and centroid for surface reconstruction can be interpolated accurately by the "intersection-based" schemes. However, the toughest difficulty in this scheme is calculating the intersection of different grids which is a very complicated geometric problem. A sampling technique was used by Horak in a two-dimensional grid intersection [31] to avoid the difficult geometric analysis but it is inaccurate and suffers from a low speed of convergence. A three-dimensional grid intersection method was proposed by Dukowicz and Padial [30], which defines the cell boundaries as a bilinear quadric surfaces and then approximate the profile of intersection by the surface-edge intersection. This algorithm runs into difficulty in some situations such as highly distorted meshes. Powell and Abel calculated the intersection of two convex polyhedrons by sequentially clipping one against the faces of the other [33], but this method is only suitable for convex polyhedrons. Grandy proposed a way to calculate the intersection between a polyhedron and a tetrahedron [29] which can precisely calculate the intersection portion through an extremely complex geometric analysis. Jia et al. simplified part of this algorithm [34]. The complexity of this algorithm will hinder the efficient assessment and it may fail in real implementation due to the geometric degeneracies [34].

Recently, some hybrid remapping methods for multi-material remapping were developed which combine the advantages of the above two basic schemes [32,35,36] but all of them are only applied in two-dimensional problem. Because the calculation of polyhedron–polyhedron intersection is significantly more complicated than polygon–polygon intersection, research on MMALE method mostly focuses on 2D problem [7,37–39]. To the best of our knowledge, there is only one public paper which involves 3D-MMALE method [34] but it does not solve the geometric problem successfully because many nonphysical material fragments occur in its 3D numerical examples. In summary, despite many accomplishments that have been made on MMALE method, efforts are still needed for a more accurate, robust, efficient and programming friendly algorithm in calculating the grid intersection especially in 3D problem.

Besides the polyhedron intersection technique, the polyhedron subdivision is another crucial issue in the multi-material remapping phase. The polyhedron subdivision will be performed after the surface reconstruction in the mixed cells to divide them into sub-polyhedrons which only contain one material. However, because of the non-convexity of the Lagrangian cells, one planar material surface may divide a mixed cell into more than two sub-polyhedrons. It is a common circumstance in 3D problems and must be precisely considered for a correct remapping phase. Moreover, if a mixed cell contains more than two materials, it will be subdivided multiple times. In summary, the non-convex subdivision and multiple subdivision cases impose difficulty in the existing subdivision algorithms, such as the clipping and capping method [40], and extra effort is required for an accurate, robust and general polyhedron subdivision algorithm.

In this paper, we firstly present an improved polyhedron subdivision algorithm which could automatically and precisely handle the convex subdivision, non-convex subdivision and multiple subdivision. Afterwards for the polyhedron intersection problem, we follow the basic idea from Grandy (1999) [29] and Jia et al. (2013) [34] but propose a different computational



Fig. 1. The illustration of staggered grid and compatible discretization in Lagrangian phase.

scheme. The new scheme solves the geometric problem in a global viewpoint which is analogous to a product manufactured on a pipelining. A comparison with Jia et al. (2013) shows our new method improves the efficiency by 55% to 65% when calculating polyhedron intersections because it remarkably reduces the complexity of the original algorithm. Moreover, the instability caused by the geometric degeneracy can be thoroughly avoided because the geometry integrity is preserved in the new algorithm. With the application of our subdivision and intersection algorithm, the multi-material remapping phase will always preserve the volume conservation without the occurrence of nonphysical material fragments.

Finally, it should be mentioned that other methods involving the grid intersection process, such as the discrete element method and immersed boundary method, can also benefit from our new intersection algorithm.

The outline of the paper is as follows. In Section 2 we briefly review the multi-material ALE method and emphasize the difficulties in the polyhedron subdivision and intersection in the remapping phase. In Section 3 we will present the improved polyhedron subdivision algorithm and in Section 4, the new polyhedron intersection algorithm is introduced in details. Several numerical results are presented in Section 5 to validate our algorithm and finally the conclusions are drawn in Section 6.

2. Review of multi-material ALE method

In this section, we will briefly review the major process of multi-material ALE method including the Lagrangian phase, the surface reconstruction phase, the rezone phase and the multi-material remapping phase. More attention will be paid on the remapping phase with emphasis on the difficulties in polyhedron subdivision and polyhedron intersection. Afterwards, the volume conservation index will be introduced which is an important quantity to evaluate the robustness of the remapping phase.

2.1. Lagrangian phase

During the Lagrangian phase, the gas dynamics equations are solved in Lagrangian form using the staggered compatible discretization. In the staggered grid, the position x and velocity v are located at the nodes while the other variables such as density ρ , specific internal energy e and pressure p are located at the cell center as shown in Fig. 1. The compatible discretization [8,9] is used to update the internal energy as in Eq. (1) which can rigorously preserve the total energy conservation.

$$M\frac{\mathrm{d}e}{\mathrm{d}t} = -\sum_{c} \boldsymbol{f}_{c} \cdot \boldsymbol{v}_{c} \tag{1}$$

where *M* is the mass of the cell obtained by the sum of the corner mass m_c (c = 1, 2, 3, 4), *e* is the internal energy of the cell, f_c is the corner force contributed to each node from the cell and v_c is the velocity of the cell node. The artificial viscosity and hourglass viscosity [10–13] are also applied to mitigate the nonphysical grid distortion. The time integration is conducted using the predictor–corrector scheme [16] where the variables are firstly updated to the half time step level to estimate the pressure, and then all physical quantities will be updated to the full time step. For the mixed cells, we use the Tipton pressure relaxation model [16], which was proposed by Tipton and summarized by Shashkov, to determine the pressure in mixed cells for solving the momentum equation. The material centroid should also be updated in Lagrangian phase for surface reconstruction and we follow the idea of the constant parametric coordinate method [41].

2.2. Surface reconstruction phase

If the grid is distorted after several Lagrangian steps, the materials' surface will be reconstructed to distinguish different materials before the remapping phase. In this paper, the MoF method [42–44] is used to reconstruct the material surface from the materials' volume fraction and centroid in the mixed cells. The MoF method is an advanced interface reconstruction method with higher accuracy. The approximate linear surface is chosen to precisely match the materials' volume and to



Fig. 2. The major process of multi-material remapping phase: (a) The distorted grid (black solid lines) and the material surface (dashed lines) reconstructed by the MoF method. (b) The rezoned grid (black lines) and the material polyhedrons (white lines) from the distorted grid are intersected to obtain the intersection pieces. (c) Each new cell accumulates the intersection pieces which belong to it. (d) The material surface after the remapping phase. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

minimize the discrepancy between the given reference materials' centroid and the approximate materials' centroid, namely calculating a particular plane $\mathbf{n}^* \cdot \mathbf{x} + d^* = 0$ to minimize the objective function

$$f(\mathbf{n},d) = \left\| \mathbf{x}(\mathbf{n},d) - \mathbf{x}^{\text{ref}} \right\|$$
(2)

subject to the volume condition

$$v\left(\boldsymbol{n},d\right) = v^{\text{ref}} \tag{3}$$

where $\mathbf{x}(\mathbf{n}, d)$ and $v(\mathbf{n}, d)$ are the centroid and volume below the approximate surface, \mathbf{x}^{ref} and v^{ref} are the given material centroid and volume in the mixed cell. The MoF method does not need any information from the neighbor cells which allows it to be implemented as a cell-by-cell black-box routine and be parallelized innately.

2.3. Rezone phase

A new grid with high quality should be created when the old grid is distorted after several Lagrangian steps. Many algorithms have been proposed for this phase [20–25,31] and we use the pseudo-structural approach [20–23,25] to smooth the grid if the boundary of the computational region is deformed. In this method, the computational grid is regarded as a pseudo continuous structure and the new position of grid nodes can be obtained by the statics FEM approach:

$$Kd = \begin{bmatrix} K^{II} & K^{IB} \\ K^{BI} & K^{BB} \end{bmatrix} \begin{bmatrix} d^{I} \\ d^{B} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
(4)

where K is the stiffness matrix of this pseudo structure, d^{I} is the displacement of interior nodes to be solved and d^{B} is the given displacement of the boundary nodes. However, in the numerical examples of this paper, the boundary of the computational region is fixed so we use the initial grid as the new grid directly.

2.4. Multi-material remapping phase

The physical variables should be interpolated from the old distorted grid to the new high-quality grid in the remapping phase. Fig. 2(a) shows an enlarged local view of a distorted grid: the solid lines represent the cell boundaries and the dashed lines are the reconstructed surface by the MoF method. The reconstructed surface will subdivide the cells into the red and blue regions named as material polyhedrons. For a second order remapping, the piecewise linear functions for ρ and ρe are reconstructed on each material polyhedrons, namely

$$g_{k}^{i}(x, y, z) = \bar{g}_{k}^{i} + \left(\frac{\partial g}{\partial x}\right)_{k}^{i}\left(x - x_{k}^{i}\right) + \left(\frac{\partial g}{\partial y}\right)_{k}^{i}\left(y - y_{k}^{i}\right) + \left(\frac{\partial g}{\partial z}\right)_{k}^{i}\left(z - z_{k}^{i}\right)$$
(5)

where $g_k^i(x, y, z)$ is the approximate piecewise linear function for ρ or ρe on the *k*th material polyhedron in cell *i*, \bar{g}_k^i is the value of ρ or ρe at the centroid of this material polyhedron, x_k^i, y_k^i, z_k^i are the coordinates of the polyhedron's centroid and $\left(\frac{\partial g}{\partial x}\right)_k^i, \left(\frac{\partial g}{\partial y}\right)_k^i, \left(\frac{\partial g}{\partial z}\right)_k^i$ are the reconstructed gradients from neighbors by a least squares approximation [26] with the Barth–Jespersen limiter [45] to ensure the monotonicity. Fig. 2(b) shows the rezoned new grid in black solid lines and the material polyhedrons from the old distorted grid in white solid lines. In order to interpolate the variables from the old grid to the new grid, the rezoned grid will intersect the material polyhedrons into separated intersection pieces. For the sake of clarity, only one intersection piece *abPcd* is illustrated which comes form the intersection of the cell *ABCD* and the material polyhedron *PQRS*. After calculating the intersection, the old grid will be decomposed into a large number of

non-overlapping intersection pieces. Each intersection piece only contains one material and the mass/internal energy carried by this intersection piece can be calculated from the volume integral of Eq. (5). For instance, the mass of an intersection piece is:

$$m\left(C_{j}\cap P_{k}^{i}\right) = \int_{C_{j}\cap P_{k}^{i}} \rho_{k}^{i}\left(x, y, z\right) dxdydz$$

$$\tag{6}$$

where C_j is the *j*th cell of the new grid, P_k^i is the *k*th material polyhedron in cell *i* of the old grid and $\rho_k^i(x, y, z)$ is the reconstructed density distribution on P_k^i by Eq. (5). Then, each new cell will accumulate the intersection pieces belonged to it as shown in Fig. 2(c) and the cell will also sum the physical quantities carried on these pieces. Finally, the variables on the new cell can be obtained by calculating the "average" of the accumulated physical quantities. For instance, the mass of the *k*th material in C_j is calculated from the mass accumulation of each intersection piece as

$$m_k(C_j) = \sum_i m\left(C_j \cap P_k^i\right) \ i \in \left\{i \mid C_j \cap P_k^i \neq \emptyset\right\}$$

$$\tag{7}$$

and the density of the *k*th material is:

$$\rho_k\left(C_j\right) = \frac{m_k\left(C_j\right)}{v_k\left(C_j\right)} \tag{8}$$

where $v_k(C_j)$ is the volume of the *k*th material in C_j . Fig. 2(d) shows the material surface after the remapping phase which is very close to the original one. For the details of the multi-material remapping phase, the reader can refer to the literature [32,34–36].

2.5. Challenges in the multi-material remapping phase

From the description above, we can see the two geometric challenges in the remapping phase. The reconstructed material surface will subdivide the mixed cell into material polyhedrons, so a polyhedron subdivision process must be performed. It is pretty easy in 2D problems but in 3D case, the four nodes on a distorted cell face will seldom lie on the same plane, so it should be represented as a non-convex polyhedron [26,34]. If the volume fraction in a mixed cell is not too small, only one cutting face will be generated as shown in Fig. 5(a) and we call it convex subdivision. However, if the volume fraction is very small, multiple cutting faces may be occurred and more than two material polyhedrons are generated as shown in Fig. 5(b) so we call it non-convex subdivision. An accurate polyhedron subdivision process is the prerequisite of a volume conservation multi-material remapping phase, so an algorithm which can precisely and automatically tackle the convex and non-convex subdivision is required in the MMALE method.

The second challenge is the polyhedron intersection. It is very complex even in 2D case and the complexity will rise remarkably when solving 3D problems. Moreover, the difficulty will be strengthened by the arbitrariness of shape and location of the material polyhedrons in the old distorted grid. The volume conservation of a remapping phase can be preserved only when all the intersection pieces are precisely calculated which pose a great challenge on the robustness of the intersection algorithm. On the other hand, the grid intersection process is time consuming, so the efficiency of the algorithm will significantly influence the whole efficiency of the MMALE method.

2.6. Volume conservation index

In the multi-material remapping phase, each new cell C_e will accumulate the intersection pieces from the old grid and reassemble them to interpolate the variables. Assume there are *n* intersection pieces belong to C_e and denote them as p_j , we have

$$p_1 \cup p_2 \cup \ldots \cup p_n = C_e \tag{9}$$

Therefore, the sum of the volumes from all the intersection pieces must equal to the volume of C_e , namely

$$\sum_{j=1}^{n} V_{p_j} = V_{C_e}$$
(10)

However, Eq. (10) can not be exactly fulfilled in practice because there are two kinds of error source. The first is the truncation error of a float number which is inevitable and the second is the system error due to the instability of the intersection algorithm. The system error always occurs in the geometry degeneracy cases where one or more vertices of one polyhedron are exactly lying on or very close to a face of the other polyhedron. The system error is significantly larger than the truncation error and if it appears, the truncation error is invisible.



Fig. 3. The decomposition of a generalized hexahedron. The black lines are the original cell and the red numbers/lines are the auxiliary nodes/edges which decompose the cell into 24 triangle pieces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Denote the calculated volume of p_j and C_e as $V_{p_j}^*$ and $V_{C_e}^*$ respectively and the volume conservation index of this cell can be defined as

$$\Delta_{C_e} = |V_{C_e}^* - \sum_{j=1}^n V_{p_j}^*| = |(V_{C_e}^* - V_{C_e}) - \sum_{j=1}^n (V_{p_j}^* - V_{p_j})|$$
(11)

where $V_{C_e}^*$ can be directly calculated from the coordinates of the cell nodes, so $V_{C_e}^* - V_{C_e}$ only contains the truncation error. On the other hand, all the intersection pieces p_j are mutually independent, so the *n* terms in the summation on the RHS of Eq. (11) are also independent to each other. Therefore, if Δ_{C_e} is around the amount of the truncation error, for instance less than 10^{-10} , all the terms $V_{p_j}^* - V_{p_j}$, (j = 1, 2, ...n) must only contain the truncation error without any system error. Namely, all the intersection pieces belong to C_e are precisely calculated. Conversely, if Δ_{C_e} is far more than the truncation error, the system error must exist which indicates that some intersection pieces are not correctly calculated.

The volume conservation index for the *k*th remapping phase Δ_{\max}^k can be defined as the maximal Δ_{C_e} in the rezoned grid, namely

$$\Delta_{\max}^{k} = \max_{e} \left(\Delta_{C_{e}} \right) \tag{12}$$

and the volume conservation index for the simulation Δ_{max} can be defined as the maximal of Δ_{max}^k in each remapping phase, namely

$$\Delta_{\max} = \max_{k} \left(\Delta_{\max}^{k} \right) \tag{13}$$

Finally, if Δ_{max} is around the amount of the truncation error, all the intersection pieces during the simulation are all precisely calculated.

3. Polyhedron representation and subdivision

In the Lagrangian phase, the grid is deformed with the material, so the cells will become generalized hexahedrons where the four nodes on a cell face may not lie on the same plane. To represent these cells, a virtual node, whose coordinate equals to the average of the 4 nodes' coordinate on this face, is added at the center of each cell face and then we decompose the face into four auxiliary triangle pieces [26,34,42]. As shown in Fig. 3, the nodes in red numbers are the virtual nodes and they will decompose the 6 cell faces into 24 triangle pieces by the red auxiliary lines. Therefore, after the decomposition, the cell becomes a polyhedron with 24 faces named as "cell polyhedron" and if the 4 nodes on the cell face, for instance 1, 2, 3, 4, do not lie on a same plane, the "cell polyhedron" may be a non-convex polyhedron.

As described in Section 2, in the first step of the multi-material remapping phase, each mixed cell will be subdivided by the planar approximate material surface to generate the material polyhedrons. Therefore, if a "cell polyhedron" is non-convex and the approximate material surface lies across the non-convex region, the non-convex subdivision occurs where the "cell polyhedron" will be subdivided into more than 2 sub-polyhedrons as shown in Fig. 5(b). It is a common situation in 3D MMALE simulation but little attention has been paid to it so far. However, in order to preserve the volume conservation of the remapping phase, the non-convex subdivision must be precisely considered.

In this section, we firstly construct a simple polyhedron data structure which contains a list of nodes' coordinate and a list of triangle pieces and then apply it with a modified "clipping and capping" subdivision algorithm. We will show that the new polyhedron subdivision algorithm can precisely and automatically tackle various situations including convex subdivision, non-convex subdivision and multiple subdivision.



Fig. 4. The major process of clipping and capping algorithm. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

3.1. Polyhedron representation

The "cell polyhedron" consists of 14 nodes and 24 triangle pieces, so it can be represented by node coordinate list and a triangle piece list. Analogous to the data structure on unstructured grid, each triangle piece can be identified by the node ID of its vertices, for instance 941, 912, 923, 934, \cdots , in Fig. 3, which are named as EN (Element Node) vectors. Generally, after a triangulation, any polyhedron can be represented by a combination of a node coordinate list and a EN list. These two lists are the basic data of a polyhedron and they can determine the derived topology information such as the adjacent pieces of each triangle piece and the surrounding pieces of each node.

It should be mentioned that one data structure may contain multiple polyhedrons. As shown in Fig. 5(b), the original polyhedron is subdivided into three parts but the two green polyhedrons can be combined in one data structure. This feature guarantees that different types of subdivision can be performed in an uniform way because no matter it is convex subdivision or non-convex subdivision and how many sub-polyhedrons are generated, the subdivision process is consistent: one original data structure will always be subdivided into two data structures.

If a mixed cell contains more than two materials, the subdivision process will be performed more than once. In order to do the multiple subdivision automatically, the subdivided polyhedrons must also be represented in the same data structure, namely all the faces of the subdivided polyhedrons must be decomposed into triangles. In this way, the subdivision process can be performed sequentially.

3.2. Polyhedron subdivision

The "clipping and capping" algorithm [40,46] is an effective way for polyhedron subdivision. This algorithm first calculates the intersection lines of the cutting plane and the polyhedron, as the segments ab, bc, cd and da in the left side of Fig. 4, and meanwhile it triangulates the truncated faces by the red lines. Sequently, each triangle piece will be allocated into the corresponding "upper" or "lower" data structure as shown in the middle of Fig. 4. Afterwards, the cutting face abcd is triangulated and these triangle pieces will be added into both "upper" and "lower" data structures and finally renumerate the node number to obtain the two final subdivided data structures as shown in the right side of Fig. 4. For the details of this algorithm, readers can refer to the literature [40,46].

It should be mentioned that the traditional clipping and capping algorithm can handle the non-convex subdivision automatically by adopting our polyhedron data structure. The only difference between non-convex and convex subdivision is that there is more than one cutting face and more than two sub-polyhedrons are generated. However, no matter how many cutting faces exist, we only need to allocate the triangle pieces into the appropriate "upper" or "lower" data structure and after the renumeration, the multiple sub-polyhedrons can be automatically merged in one data structure.

Fig. 5 shows an example of the convex subdivision, non-convex subdivision and the multiple subdivision on a generalized hexahedron. The dashed lines on the original hexahedron are the cutting plane and the dashed lines on the subdivided polyhedron are the auxiliary lines for the triangulation. The two green subdivided polyhedrons in Fig. 5(b) are merged in one data structure in the non-convex subdivision process.

4. Polyhedron intersection algorithm

In the remapping phase, the intersection of the rezoned cell and the material polyhedrons will be calculated. However, in practice, we only need the volume and centroid of the intersection portion for a second order accuracy interpolation [26,34] and do not need its specific profile. Therefore, to simplify the problem, the rezoned cell will usually be decomposed into several tetrahedrons and the final volume/centroid can be obtained by the sum of the volume/centroid of each sub-tetrahedron and the material polyhedrons intersection.

In this section, we will first introduce the basic idea of Grandy (1999) [29] and Jia (2013) [34] which aims for solving the tetrahedron–polyhedron intersection. Afterwards, we will describe our new intersection algorithm in details.



Fig. 5. The convex subdivision (a), non-convex subdivision (b) and the multiple subdivision (c) on a generalized hexahedron. The left side is the original hexahedron and the right side are the subdivided polyhedrons. The dashed lines in the left represent the cutting plane and the dashed lines in the right represent the auxiliary lines for the triangulation. The two green subdivided polyhedrons in (b) are merged in one data structure in the non-convex subdivision process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 6. The relative position of the unit tetrahedron and the triangle piece.

4.1. Basic idea of tetrahedron-polyhedron intersection

In Grandy (1999) [29] and Jia et al. (2013) [34], both the tetrahedron T and the polyhedron P are cast into a reference coordinate system by an affine transformation, X = Ax + b, which transform T to an unit tetrahedron U with vertices O(0, 0, 0), E(1, 0, 0), F(0, 1, 0) and G(0, 0, 1) as shown in Fig. 6. X is the coordinate in the reference system and x is the coordinate in the physical system. The details of the affine transformation can be referred to in Jia et al. (2013) [34]. Meanwhile, the polyhedron P in physical system is transformed to \mathbb{P} whose triangle pieces are denoted as S_i . For the sake of clarity, only one triangle piece of \mathbb{P} , namely PQR, is shown in Fig. 6. In Fig. 6, the dashed lines are used to represent the invisible portions to make the relative position of the unit tetrahedron and the triangle piece clearer: the segment QR intersects with face EFG at I_1 , the segment PR intersects with face OGF at I_2 and the segment GF intersects with face PQR at I_3 . The segment PQ is outside the tetrahedron and the vertex R is in the interior of the tetrahedron.

Then, after a geometric calculation, two key polygons with vertices 1237 and 34567 are obtained as shown in Fig. 7(a) and denoted them as *A* and *B* respectively. *A* is the portion of *PQR* which lies in the interior of *U* and *B* is the projection of *PQR* from the +z direction onto the face *EFG* of the unit tetrahedron. Polygon *A* and *B* correspond to two columns C_A



Fig. 7. The basic idea of the intersection algorithm in Grandy (1999) [29].

and C_B whose inclined planes are A or B and their bottoms lie on the OXY plane as shown in Fig. 7(b). Denote the union of C_A and C_B as Ω and the volume and centroid of $U \cap \mathbb{P}$ can be obtained by

$$V (\boldsymbol{U} \cap \mathbb{P}) = \sum_{i} \operatorname{sign}(n_{iz}) V (\Omega_{i}) = \sum_{i} \int_{\Omega_{i}} \operatorname{sign}(n_{iz}) dX dY dZ$$
(14)

$$\boldsymbol{X}(\boldsymbol{U} \cap \mathbb{P}) = \frac{\sum_{i} \int_{\Omega_{i}} \operatorname{sign}(n_{iz}) \boldsymbol{X} dX dY dZ}{V(\boldsymbol{U} \cap \mathbb{P})}$$
(15)

where V (*) is the volume of *, X (*) is the centroid of *, Ω_i is the union of C_A and C_B which are corresponded to triangle piece S_i and n_{iz} is the z-component of the outward normal of S_i . The volume integral in Eq. (14) and Eq. (15) can be converted to a surface integral on the slope by Gauss theorem and they can be easily calculated using the Gauss integration scheme [44].

It should be mentioned that Eq. (14) and Eq. (15) are not easy to be understood for 3D problem and the previous papers [29,34] did not give a specific explanation, so we will illustrate a 2D example about the intersection of an unit triangle and a polygon in Appendix A to facilitate the comprehension of Eq. (14) and Eq. (15).

The results above are the intersection volume and centroid in the reference system and they can be transformed into the physical system by

$$V(\mathbf{T} \cap \mathbf{P}) = \det(\mathbf{A}) \times V(\mathbf{U} \cap \mathbb{P})$$
(16)

$$\boldsymbol{x}(\boldsymbol{T} \cap \boldsymbol{P}) = \boldsymbol{A}^{-1}\left(\boldsymbol{X}\left(\boldsymbol{U} \cap \mathbb{P}\right) - \boldsymbol{b}\right) \tag{17}$$

In summary, the key issue of this intersection algorithm is calculating the polygon *A* and *B*. In Grandy (1999) [29], both polygons *A* and *B* are calculated and in Jia et al. (2013) [34], they only calculate polygon *A* and then find an alternative polygon for *B* according to the union of polygon *A* from each triangle piece of \mathbb{P} . Both of these two algorithms involve extremely complicated computations and geometric judgments and need extra approaches to reduce the instability caused by geometric degeneracy.

4.2. Clipping and projecting algorithm

In order to calculate the polygons *A* and *B* efficiently and robustly, we propose a "Clipping and Projecting" method. Taking the case in Fig. 7 as an example, the flowchart of this method is shown in Fig. 8. This algorithm will involve node insertion and deletion, so a circular linked list data structure is recommended. In Fig. 8(a), the initial triangle is PQR which can be represented in a circular linked list as $P \rightarrow Q \rightarrow R \rightarrow P$. In the first step, the triangle piece will be cut by OYZ plane and nodes K_1 , K_2 are generated. Then, discard the outside portion which is illustrated as the red dashed lines in Fig. 8(b) and the circular linked list changes to $R \rightarrow K_1 \rightarrow K_2 \rightarrow Q \rightarrow R$. Similarly, cutting the circular linked list by X + Y = 1 plane and discarding the outside portion, the circular linked list comes to $R \rightarrow K_1 \rightarrow K_2 \rightarrow K_3 \rightarrow K_4 \rightarrow R$ as shown in Fig. 8(c). Afterwards, the circular linked list should also be clipped by OZX and OXY plane but they are not intersected, so the circular linked list is unchanged after the clipping. Sequentially, the circular linked list will be cut by the XYZ plane, i.e. X + Y + Z = 1 plane, and the circular linked list changes to $R \rightarrow K_1 \rightarrow K_6 \rightarrow K_2 \rightarrow K_3 \rightarrow K_4 \rightarrow K_5 \rightarrow R$ as shown in Fig. 8(d). Finally, we do not discard the outside portion but project it onto the X + Y + Z = 1 plane and the final circular linked list is obtained as $R \rightarrow K_1 \rightarrow K_6 \rightarrow K'_2 \rightarrow K'_3 \rightarrow K'_4 \rightarrow K_5 \rightarrow R$ as shown in Fig. 8(e). The final circular linked list is the slope of column C_A and C_B and the volume integrations on them can be calculated as the description in the previous section.

The "Clipping and Projecting" process can be summarized in five steps:

- 1. Clip the initial triangle piece by the OYZ plane and discard the outside portion to obtain a new polygon Ω_1 ;
- 2. Clip the Ω_1 by the X + Y = 1 plane and discard the outside portion to obtain a new polygon Ω_2 ;



Fig. 8. The flowchart of the "Clipping and Projecting" method. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 1

The comparison of "Clipping and Projecting" method and the tradition methods.

	New method	Original method
Intersection type Vertices reorganization	segment-plane automatically decomposition reorganization	segment-triangle need extra calculation
liller logic	decomposition-reorganization	giobal viewpoliit

- 3. Clip the Ω_2 by the *OZX* plane and discard the outside portion to obtain a new polygon Ω_3 ;
- 4. Clip the Ω_3 by the OXY plane and discard the outside portion to obtain a new polygon Ω_4 ;
- 5. Clip the Ω_4 by the X + Y + Z = 1 and project the outside portion onto the X + Y + Z = 1 plane to obtain the final polygon *A* and *B*.

It should be mentioned that the order of the first four steps above is arbitrary.

We will analyze the efficiency and the robustness of our new algorithm in the next subsection and show its advantages over the methods in Grandy (1999) [29] and Jia et al. (2013) [34].

4.3. Comparison with the original intersection algorithms

The inner logic of the original methods from Grandy (1999) [29] and Jia et al. (2013) [34] is "decompositionreorganization". In the "decomposition" process, the whole task is decomposed into sub-tasks for calculating the intersection of each pair of triangle and segment to generate a series of intersecting points. Afterwards in the "reorganization" process, all the vertices, including the original vertices and the intersecting vertices, are reorganized to obtain the final polygons.

However, our new method solves the problem in a global viewpoint which looks like a pipelining. The original triangle piece PQR will be "manufactured" to the target polygons A and B in the five procedures above. Each procedure only involves at most 2 "segment-plane" intersections and the corresponding vertex insertions and deletions on the linked list. The whole linked list is analogous to a production manufactured on a pipelining and its integrity is preserved after each procedure so our new method is in a global viewpoint.

The comparison of our "Clipping and Projecting" method with the original methods is listed in Table 1 and we will show that the first two rows are resulting in an efficiency improvement and the third row is resulting in a robustness improvement.

4.3.1. The efficiency improvement

The efficiency improvement of our new algorithm comes from the following two aspects.



Fig. 9. The stability analysis of the "Clipping and Projecting" method.

Firstly, the calculation cost of the segment-triangle intersection is much more than the segment-plane intersection. Denote the plane where the triangle located as $\mathbf{n} \cdot \mathbf{x} + d = 0$ and the endpoints of the segment are \mathbf{x}_1 and \mathbf{x}_2 , the segment and plane is intersected only when

$$(\mathbf{n} \cdot \mathbf{x}_1 + d) (\mathbf{n} \cdot \mathbf{x}_2 + d) \le \mathbf{0} \tag{18}$$

and the intersection point is obtained directly from a linear interpolation

$$\mathbf{x}^* = \mathbf{x}_1 + \frac{\mathbf{n} \cdot \mathbf{x}_1 + d}{\mathbf{n} \cdot (\mathbf{x}_1 - \mathbf{x}_2)} (\mathbf{x}_2 - \mathbf{x}_1)$$
(19)

However, if the segment and the triangle is intersected, x^* must be inside the triangle and the corresponding calculation cost to check this condition is much more than that in Eq. (18) and Eq. (19). Moreover, in our new method, the components of the normal of clipping plane is either 0 or 1 which will simplify the calculation in Eq. (18) and Eq. (19) further.

Secondly, in the traditional methods, the vertices of the final polygons should be selected from all vertices, including the origin vertices and the intersecting vertices, and then reorganize them in a circular order. This process involves complicated geometric judgments and extra calculation. However, in our new algorithm, after the five procedures, the final polygons are automatically obtained with the vertices in a circular order, so no extra calculation is required.

4.3.2. The robustness improvement

The traditional methods will decompose the whole task into sub-tasks about segment-triangle intersections and then reorganize the vertices to obtain the key polygons. However, in the degenerated cases, where one vertex, edge or face of a polyhedron is located on or very close to a face of the other polyhedron, the results of the segment-triangle intersections are instable due to the truncation error of a float number. This instability may affect the reorganization of vertices and result in an incorrect key polygon.

On the contrary, our new method is based on a global viewpoint, namely the whole linked list is been operated and its integrity is preserved after each procedure. The integrity preservation can protect the new method from the instability caused by the truncation error in geometry degeneracy cases. For instance, as shown in Fig. 9, the circular linked list \mathbf{R}_0 is $P_2 \rightarrow P_1 \rightarrow P_4 \rightarrow P_3 \rightarrow P_2$ in Fig. 9(a) and it will be clipped by the plane represented in the dashed line and the segment P_1P_4 is exactly lying on the cutting plane. Because of the truncation error, P_1 and P_4 may actually lie on the both sides of the cutting plane and it will derive many possible results. We will only analyze two possible situations as shown in Fig. 9(b) and Fig. 9(c) and other situations can be analyzed in the same way.

In case Fig. 9(b), P_1 is outside the cutting plane while P_4 is inside so two intersection points S_1 , S_2 are generated from Eq. (19). Therefore, the modified circular linked list \mathbf{R}_b is $P_2 \rightarrow S_1 \rightarrow S_2 \rightarrow P_4 \rightarrow P_3 \rightarrow P_2$. In case Fig. 9(c), P_1 is inside the cutting plane while P_4 is outside so two different intersection points S_3 and S_4 are generated and the modified circular linked list R_c is $P_2 \rightarrow P_1 \rightarrow S_3 \rightarrow S_4 \rightarrow P_3 \rightarrow P_2$. The difference between \mathbf{R}_0 and \mathbf{R}_b is the tiny triangle $S_1S_2P_1$ and the difference between R_0 and R_c is also the tiny triangle $S_3S_4P_4$. The sizes of these two tiny triangle are within the range of the truncation error, so that it only produces minor error in the volume integration.

It should be mentioned that the coordinates of the intersection points S_2 in (b) and S_3 in (c) are instable because in these two cases, the denominator in Eq. (19) is close to zero. However, no matter where they are located, the sizes of triangles $S_1S_2P_1$ in (b) and $S_3S_4P_4$ in (c) are always within the range of the truncation error.

Although this is a very simple example, it demonstrates how the new method overcomes the instability caused by the numerical perturbation. In summary, because the integrity of circular linked list, namely the polygon, is preserved after each clipping or projecting process, the truncation error sized perturbation will only lead to a truncation error sized difference in the final configuration. Therefore, the volume of the intersection portion only contains the truncation error and the numerical examples in Section 5 will validate the high robustness of this new intersection algorithm.

4.4. Algorithm procedure for polyhedron intersection

The overall process for calculating the intersection volume and centroid of a polyhedron intersected with a tetrahedron can be summarized as follows:



Fig. 10. The old grids and new grids for efficiency comparison.

- 1. Calculate the reference coordinate system of the tetrahedron and transform the polyhedron into this system;
- 2. Perform the "Clipping and Projecting" process to obtain the key polygons A and B and the corresponding columns C_A and C_B ;
- 3. Calculate the volume integration on C_A and C_B by Eq. (14) and Eq. (15);
- 4. Sum the volume integrations of every triangle piece and convert the results back to the physical system by Eq. (16) and Eq. (17).

It should be mentioned that the initial circular linked list is a triangle in this paper because our polyhedron is represented by a series of triangle pieces. But in general, the initial circular list can be an arbitrary polygon as long as all the vertices lie on a same plane.

5. Numerical examples

In this section, four numerical tests are presented to show the efficiency and robustness of our new algorithms. The simulations in Section 5.2 to Section 5.4 are run on a PC with one Intel Core i7-3770 processor using 4 threads (OpenMP parallel). It should be mentioned that the boundary of the computational region is fixed in the hydro tests so we use the initial grid as the new grid directly, namely, all the hydro tests are performed as Eulerian.

5.1. Efficiency comparison

The first example will show the efficiency improvement of this new method. A linear function f(x, y, z) = x + 2y + 3z + 4 will interpolate from one grid to another grid in the "intersection-based" scheme by both previous algorithms in Jia et al. (2013) [34] and the new "Clipping and Projecting" method. Two tests are performed as shown in Fig. 10. The left side is the old grid and the right side is the new grid. In case A, the old and new grids have 317575 and 64204 cells respectively and in case B, the old grid is the same as the new grid in case A while the new grid is extremely distorted. A linear function can be precisely interpolated by the "intersection-based" scheme so the average interpolation error,

$$\operatorname{error} = \frac{\sum_{i=1}^{N} \left| f_i - f_i^{\operatorname{ref}} \right|}{N}$$
(20)

where *N* is the number of cells in the new grid, f_i is the interpolated value on cell *i*, f_i^{ref} is the exact value on cell *i*, and the error should be close to the range of the truncation error.

The results¹ are shown in Table 2. The CPU times only include the time for calculating the intersection and do not include the time in other parts such as the bucket searching. The results indicate that the efficiency is improved by 64% in case A and 55% in case B. The lower efficiency improvement in case B may be caused by the severe distortion of the new grid. But

¹ We run two different codes using our algorithm and Jia's algorithm in the same computer to do the comparison for fairness. The only difference between the two codes is the function for calculating the intersection of tetrahedrons. Jia provides his code for tetrahedron intersection to do this comparison.

Intersection algorithm	Error		CPU time/s	
Case	A	В	A	В
Jia et al. (2013) [34] Our method	$\begin{array}{c} 8.59 \times 10^{-9} \\ 2.90 \times 10^{-15} \end{array}$	$\begin{array}{c} 7.36 \times 10^{-9} \\ 1.32 \times 10^{-15} \end{array}$	26.06 9.5	11.91 5.34

Table 2The efficiency comparison of intersection algorithms.



Fig. 11. Initial state of the triple-point problem.



Fig. 12. The material surface of triple-point problem at (a) t = 3.5 and (b) t = 5.0.

generally, the efficiency of "Clipping and Projecting" method is significantly higher than the previous algorithm in Jia et al. (2013) [34]. The average interpolation errors of the new method are around 10^{-15} which is in the range of truncation error. However, the errors from previous algorithm are only around 10^{-9} which is much larger than the truncation error. The extra error comes from the pseudo perturbation of the nodes.² Grandy [29] described a method to reduce the instability form the truncation error, but in order to achieve the further improvement of the robustness, Jia et al. applied a slight and random perturbation on each node to avoid geometric degeneracy. This approach can improve the robustness indeed but the accuracy of the interpolation will be decreased as the results listed in Table 2.

5.2. 2D triple-point problem

Triple-point problem is a standard test for large surface deformation problem [6] which involves three states 2D Riemann problem in a rectangular region as shown in Fig. 11. The initial density, pressure and polytropic index of each part are listed in Fig. 11. Although it is a 2D problem, we simulate it on our 3D code with $140 \times 60 \times 1$ cells. Ω_1 contains the ideal gas with high pressure while the pressures in Ω_2 and Ω_3 are much lower. Because of the different acoustic impedance, two shocks in Ω_2 and Ω_3 will propagate in different speeds. Therefore, a strong shear along the interface of Ω_2 and Ω_3 is generated and it will produce vortex formation so a large surface deformation occurs.

Fig. 12 shows the material surface at t = 3.0 and t = 5.0 which are close to the results in Jia et al. (2013) [34] and Kucharik et al. (2014) [32]. The multi-material remapping phase is performed 38 times and 1.7×10^6 pairs of polyhedron intersection are calculated during the simulation and the volume conservation index is $\Delta_{max} = 3.4 \times 10^{-16}$ which indicates that all the intersection portions are precisely calculated.

5.3. 3D Sedov problem

The Sedov problem [47] simulates the wave from an intense explosion in an ideal gas with $\gamma = 1.4$. The initial density is unity and the initial velocity is zero. At the beginning, all the internal energy E_0 is concentrated at the origin. We simulate it on a $[0, 1.2] \times [0, 1.2] \times [0, 1.2]$ domain with $48 \times 48 \times 48$ cells and put a pseudo material surface at $x^2 + y^2 + z^2 = 1/9$ as shown in Fig. 13. During the simulation, 4 times of remapping phase will be performed at t = 0.2, t = 0.4, t = 0.6 and t = 0.8.

 $^{^{2}}$ The pseudo perturbation is not mentioned in the original paper but we find it from the direct discussion with the author.



Fig. 14. The density comparison with the analytical solution at t = 1 (a) and the evolution of the pseudo material surface in the Sedov problem (b).

The density distribution comparison with the analytical solution at t = 1 is shown in Fig. 14(a) which indicates that our numerical results are reliable. The pseudo material surface at t = 0, t = 1/3, t = 2/3 and t = 1 are shown in Fig. 14(b) which demonstrates that the surface preserves its shape and smoothness during the simulation. There are 9.6×10^6 times of polyhedron intersection during the simulation and the volume conservation index is $\Delta_{max} = 3.4 \times 10^{-16}$ which indicates that all the intersection portions are precisely calculated.

5.4. 3D Taylor-Rayleigh problem

In this case, we will simulate the same 3D Taylor–Rayleigh instability problem described in Jia et al. (2013) [34] and compare with its results. The computational domain is $[0, 0.25] \times [0, 0.25] \times [0, 1]$ discretized by $25 \times 25 \times 100$ cells as shown in Fig. 15. Initially, the two ideal gases are separated by a perturbed interface which is described by $\tilde{z}(x, y) = 0.5 + 0.0125 \times (\cos 8\pi x + \cos 8\pi y)$. The heavy gas with $\rho_h = 2$ is located above the light gas with $\rho_h = 1$ and the polytropic index of both gases are $\gamma = 1.4$. A vertically downward gravity field is applied as $\mathbf{g} = (0, 0, -0.1)$. The initial velocities of both gases are zero and the initial pressure distribution is deduced by setting hydrostatic equilibrium as

$$\begin{cases} p(x, y, z) = 1 + \rho_h g_z(z - 1) & z > \tilde{z}(x, y) \\ p(x, y, z) = 1 + \rho_h g_z(\tilde{z}(x, y) - 1) + \rho_l(z - \tilde{z}(x, y)) & z \le \tilde{z}(x, y) \end{cases}$$
(21)

The material surface comparisons³ with Jia et al. (2013) [34] at t = 3, 4, 5, 6, 7, 8, 9, 10 are shown in Fig. 16. The profiles of both results are similar, but the differences in the detail are obvious. The configurations in the left side are the results from Jia et al. (2013) [34], which shows many nonphysical material pieces may be caused by the error in calculating the intersection. On the other hand, the configurations in the right side are the results from our improved remapping phase. Although the material surface experiences a large deformation, it still preserves its smoothness and almost no nonphysical pieces appears. There are 32 multi-material remapping phases performed during the simulation with 1.9×10^7 times of

³ In order to make a fair comparison, we use the same computational parameters and the same visualization software with Jia et al.



Fig. 15. The initial state of 3D Taylor-Rayleigh instability problem.



Fig. 16. The material surface comparison of the Taylor-Rayleigh instability problem.

polyhedron intersection. The volume conservation index is 2.2×10^{-17} which indicates that all the intersection portions are precisely calculated.

6. Conclusions

In this paper, we focus on the two core geometric problems in MMALE method, the polyhedron subdivision and the polyhedron intersection, and propose an efficient and robust scheme to solve these problems. For polyhedron subdivision, we describe an algorithm which could automatically and precisely tackle the various situations including convex, non-convex and multiple subdivisions. For polyhedron intersection, we propose a "Clipping and Projecting" method which is based on a global viewpoint and act as a production manufactured on a pipelining. The new method converts the previous "segment-triangle" intersection to the "segment-plane" intersection and does not need any extra calculation to determine the vertices' order. This improvement significantly reduce the complexity of the algorithm and improve the efficiency remarkably. Moreover, our method avoids the instability caused by geometric degeneracy in essence because the geometry integrity is preserved during the clipping and projecting process. We compare the efficiency of our method with the existing one by a linear function remapping test and the result shows that the efficiency is improved by 55% to 65%. Subsequently, we simulate the 2D triple-point problem, 3D Sedov problem and 3D Taylor–Rayleigh problem and the results demonstrate that all the polyhedron intersections in the multi-material remapping phases are precisely calculated without the occurrence of the nonphysical material pieces appeared in the existing literature.



Fig. 17. The intersection of an unit triangle and a polygon.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 11390363).

Appendix A. The intersection of an unit triangle and a polygon

To facilitate the understanding of the 3D algorithm in Section 4.1, the algorithm for calculating the area and centroid of the intersection portion of an unit triangle OXY and a polygon ABC shown in Fig. 17(a) is presented.

In this algorithm, each edge of the polygon *ABC* is analyzed individually as shown in Fig. 17(b), Fig. 17(c) and Fig. 17(d) for the edge *AC*, *AB* and *BC* respectively. In Fig. 17(a), two key segments are calculated, namely *AP* and *PQ*. Segment *AP* is the portion of *AC* which lies in the interior of the unit triangle which is analogous to the polygon *A* in Fig. 7(a) and segment *PQ* is the projection of *AC* from the +*y* direction onto the edge *XY* of the unit triangle which is analogous to the polygon *B* in Fig. 7(a). The area below segment *AP* and *PQ* is denoted as *S*₁ which is analogous to the union of column *C*_A and *C*_B in Fig. 7(b). Similarly, the key segments for the other two edges are illustrated in Fig. 17(b) and Fig. 17(c), and the area of the intersection portion *S* is

$$\iint\limits_{S} dxdy = \iint\limits_{s_1} dxdy - \iint\limits_{s_2} dxdy + \iint\limits_{s_3} dxdy$$
(A.1)

and if we use the y component of the outward normal of each edge to represent the sign of integrals on the RHS of Eq. (A.1), it can be rewritten as

$$\iint\limits_{S} dxdy = \operatorname{sign}(n_{1y}) \iint\limits_{s_1} dxdy + \operatorname{sign}(n_{2y}) \iint\limits_{s_2} dxdy + \operatorname{sign}(n_{3y}) \iint\limits_{s_3} dxdy$$
$$= \sum_{i} \int\limits_{S_i} \operatorname{sign}(n_{iy}) dxdy$$
(A.2)

Eq. (A.2) is the 2D version of Eq. (14) in Section 4.1 and the centroid of the intersection portion can be obtained from its definition.

References

- [1] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (1997) 203-216.
- [2] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (1992) 235-394.
- [3] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Methods Appl. Mech. Eng. 72 (1989) 305-350.
- [4] P. Kjellgren, J. Hyvarinen, An arbitrary Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1998) 81–90.
- [5] L. Margolin, Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (1997) 198-202.
- [6] D. Kershaw, M. Prasad, M. Shaw, J. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, Comput. Methods Appl. Mech. Eng. 158 (1998) 81–116.
- [7] J. Peery, D. Carroll, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (2000) 591-619.
- [8] E. Caramana, D. Burton, M. Shashkov, P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998) 227–262.

- [9] E. Caramana, C. Rousculp, D. Burton, A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional cartesian geometry, J. Comput. Phys. 157 (2000) 89–119.
- [10] T. Kolev, R. Rieben, A tensor artificial viscosity using a finite element approach, J. Comput. Phys. 228 (2009) 8336-8366.
- [11] E. Caramana, M. Shashkov, P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1998) 70–97.
- [12] E. Caramana, R. Loubere, Curl-q: A vorticity damping artificial viscosity for essentially irrotational Lagrangian hydrodynamics calculations, J. Comput. Phys. 215 (2006) 385–391.
- [13] E. Caramana, M. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.
- [14] J. Campbell, M. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (2001) 739-765.
- [15] R. Loubere, P.-H. Maire, P. Vachal, 3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity, Int. J. Numer. Methods Fluids 72 (2013) 22–42.
- [16] M. Shashkov, Closure models for multimaterial cells in arbitrary Lagrangian-Eulerian hydrocodes, Int. J. Numer. Methods Fluids 56 (2008) 1479–1504.
- [17] A. Barlow, R. Hill, M. Shashkov, Constrained optimization framework for interface-aware sub-scale dynamics closure model for multimaterial cells in Lagrangian and arbitrary Lagrangian–Eulerian hydrodynamics, J. Comput. Phys. 276 (2014) 92–135.
- [18] D.J. Benson, A mixture theory for contact in multi-material Eulerian formulations, Comput. Methods Appl. Mech. Eng. 140 (1997) 59-86.
- [19] Y.V. Yanilkin, E.A. Goncharov, V.Y. Kolobyanin, V.V. Sadchikov, J.R. Kamm, M.J. Shashkov, W.J. Rider, Multi-material pressure relaxation methods for Lagrangian hydrodynamics, Comput. Fluids 83 (2013) 137–143.
- [20] F.J. Blom, Considerations on the spring analogy, Int. J. Numer. Methods Fluids 32 (2000) 647-688.
- [21] C.L. Bottasso, D. Detomi, R. Serra, The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes, Comput. Methods Appl. Mech. Eng. 194 (2005) 4244–4264.
- [22] C. Degand, C. Farhat, A three-dimensional torsional spring analogy method for unstructured dynamic meshes, Comput. Struct. 80 (2002) 305-316.
- [23] C. Farhat, C. Degand, B. Koobus, M. Lesoinne, Torsional springs for two dimensional dynamic unstructured fluid meshes, Comput. Methods Appl. Mech. Eng. 163 (1998) 231–245.
- [24] P. Knupp, L. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.
- [25] P. Bar-Yoseph, S. Mereu, S. Chippada, V. Kalro, Automatic monitoring of element shape quality in 2-D and 3-D computational mesh dynamics, Comput. Mech. 27 (2001) 378–395.
- [26] R. Garimella, M. Kucharik, M. Shashkov, An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes, Comput. Fluids 36 (2007) 224–237.
- [27] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, J. Comput. Phys. 188 (2003) 462-471.
- [28] L. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (2003) 266–298.
 [29] J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys. 148 (1999) 433–466.
- [30] J. Dukowicz, N. Padial, REMAP3D: A Conservative Three-Dimensional Remapping Code, Tech. Rep. LA-12136-MS, Los Alamos National Laboratory, 1991.
- [31] H.G. Horak, E.M. Jones, J.W. Kodis, An algorithm for the discrete rezoning of lagrangian meshes, J. Comput. Phys. 26 (1978) 277–284.
- [32] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered multi-material arbitrary-Lagrangian-Eulerian methods, J. Comput. Phys. 258 (2014) 268-304.
- [33] D. Powell, T. Abel, An exact general remeshing scheme applied to physically conservative voxelization, J. Comput. Phys. 297 (2015) 340–356.
- [34] Z. Jia, J. Liu, S. Zhang, An effective integration of methods for second-order three-dimensional multi-material ALE method on unstructured hexahedral meshes using MOF interface reconstruction, J. Comput. Phys. 236 (2013) 513–562.
- [35] M. Kucharik, M. Shashkov, One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian method, J. Comput. Phys. 231 (2012) 2851–2864.
- [36] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two-step hybrid conservative remapping (conservative interpolation) for multimaterial arbitrary Lagrangian–Eulerian method, J. Comput. Phys. 230 (2010) 6664–6687.
- [37] S. Galera, J. Breil, P.-H. Maire, A 2D unstructured multi-material cell-centered arbitrary Lagrangian–Eulerian (CCALE) scheme using MOF interface reconstruction, Comput. Fluids 46 (2011) 237–244.
- [38] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (2010) 5755–5787.
- [39] R. Loubere, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method, J. Comput. Phys. 229 (2010) 4724–4761.
- [40] M.B. Stephenson, H.N. Christiansen, A polyhedron clipping and capping algorithm and a display system for three dimensional finite element models, ACM SIGGRAPH Comput. Graph. 9 (1975) 1–16.
- [41] M. Kucharik, R.V. Garimella, S.P. Schofield, M.J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, J. Comput. Phys. 229 (2010) 2432–2452.
- [42] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096–2132.
- [43] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361–5384.
- [44] X. Chen, X. Zhang, An improved 3D MoF method based on analytical partial derivatives, J. Comput. Phys. 326 (2016) 156-170.
- [45] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: D. Kröner, M. Ohlberger, C. Rohde (Eds.), An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, in: Lect. Notes Comput. Sci. Eng., Springer, Berlin, Heidelberg, 1999, pp. 195–285.
- [46] H.T. Ahn, M. Shashkov, Multi-Material Interface Reconstruction on Generalized Polyhedral Meshes, Tech. Rep. LA-UR-07-0656, Los Alamos National Laboratory. 2007.
- [47] L. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.