



A robust MoF method applicable to severely deformed polygonal mesh

Fang Qing^{a,b}, Xijun Yu^a, Zupeng Jia^{a,*}

^a Institute of Applied Physics and Computational Mathematics, Beijing 100088, China

^b Graduate School of China Academy of Engineering Physics, Beijing 100088, China

ARTICLE INFO

Article history:

Received 3 April 2018

Received in revised form 24 September 2018

Accepted 21 October 2018

Available online 24 October 2018

Keywords:

Interface reconstruction

Moment of fluid method

Severely deformed polygonal mesh

Multi-material fluid flow

ABSTRACT

The MoF (Moment of Fluid) method is an accurate approach for interface reconstruction in numerical simulation of multi-material fluid flow. So far, most works focus on improving its accuracy and efficiency, such as developing analytic reconstruction method and deducing the iteration schemes based on high order derivatives of the objective function. In this paper, we mainly concern on improving its robustness, especially for severely deformed polygonal meshes, in which case the objective function has multiple minimum value points. By using an efficient method for solving multiple roots of the nonlinear equation in large scope, a new algorithm is developed to enhance robustness of the MoF method. The main idea of this algorithm is as follows. The first derivative of the objective function is continuous, so the minimum value points of the objective function must be the zero points of the first derivative. Instead of finding the zero points of the first derivative directly, we turn to calculating the minimum value points (also zero points) of the square of the first derivative, which is a convex function on a neighborhood of each zero point. Applying the properties of convex function, the neighbor of each extreme minimum point of it can be obtained efficiently. Then each zero point of the square of the first derivative can be obtained using the iterative formula in its neighbor. Finally, by comparing the values of the objective function at these zero points of the first derivative, the global minimum value point of the objective function can be found and is the desired solution. The new algorithm only uses the first derivative of the objective function. It doesn't need an initial guess for the solution, which has to be carefully chosen in previous works. Numerical results are presented to demonstrate the accuracy and robustness of this new algorithm. The results show that it is applicable to severely deformed polygonal mesh, even with concave cells.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Simulations of compressible multi-material fluid flows can be classified into Lagrangian or Eulerian computational framework. In the Lagrangian framework, the computational mesh is embedded in the fluid and the material interface can be tracked easily and accurately. While in the Eulerian framework, the mesh is fixed with the space through which the fluid flows as time passes. So it can deal with the problem of large deformation. However, it has difficulty in tracking the mate-

* Corresponding author.

E-mail address: zpjia@iapcm.ac.cn (Z. Jia).

rial interface. For this reason, the interface capturing method has been developed for computing the motion of deforming interface problems in recent decades.

The volume of fluid (VoF) method [1–3] is an effective and popular method for tracking the material interface. It reconstructs the interface using the volume fraction of each fluid. A piecewise linear method was developed by Youngs [2], in which the interface line was defined by a slope and an intercept. Significant developments of VoF method have been made in recent years. An improved VoF method has been proposed [4], which combined the modified piecewise linear interface calculation (PLIC) based interfacial reconstruction [3] with the Lagrangian split fluid advection method and redistribution of volume fraction. This improved algorithm are applicable both in 2D and 3D spaces. One approach combining VoF method with the ghost fluid method has been presented [5]. The interface is tracked by VoF method with piecewise linear interface reconstruction of Youngs' method, and the interfacial states are reconstructed by a modified version of real ghost fluid method approach [6].

The level set method [7–10] is another popular method which has become widely used for interface tracking. It is an implicit framework for tracking interface and works with a discrete signed distance function. The level set equation can be solved using a semi-Lagrangian method. Unfortunately, numerical methods for solving the level set equation disobey the geometric conservation law. In order to combine the advantages of the VoF method and the level set method, a coupled level set and VoF method has been presented [11]. The advection of the VoF function is performed using a mixed second-order Eulerian and Lagrangian scheme with a piecewise linear interface calculation (PLIC). After the VoF advection step, an extra mass correction scheme is added to preserve the mass conservation and maintain a divergence-free velocity field.

The moment of fluid (MoF) method [12–14] is a new approach to compute the motion of deforming boundary problems. Differing from the VoF method, it uses not only the volume fraction of each fluid but also the material centroid of the cell fraction to calculate the interface line. The interface line of each mixed cell is chosen to preserve volumes of cell fractions and provide the best approximation to the given reference material centroid. When performing reconstruction in a mixed cell, the MoF method doesn't need information from its neighboring cells. It has been demonstrated that the MoF method achieved a second-order accurate approximation and provided more accurate tracking of interfaces than the VoF method [12,15]. Based on the MoF method, the MMALE method can achieve second-order accuracy for continuous solutions [16–18]. In order to deal with errors in the reference centroid data with greater robustness and accuracy, several schemes have been developed. For instance, Hill et al. [19] proposed the symmetric MoF interface reconstruction algorithm, Jemison et al. [20] presented the MoF method utilizing filament capturing. And Zinjala et al. [21] presented the coupled MoF interface reconstruction with Lagrangian–Eulerian advection scheme. MoF analytic reconstruction algorithm in [22] could achieve accurate results for a lower computational cost.

In order to achieve numerical optimization, the MoF method need to calculate an implicit nonlinear optimization problem. In [12] it is stated that even when the objective function has a unique global minimum, it may still possess some other local minima. So the initial guess and trial step strategy should be carefully chosen to get the unique global minimum of the objective function. A safe initial guess is provided in [12]. The analytical first derivative of the objective function has been deduced [12]. Chen and Zhang [23] deduced and analyzed the high order derivatives on the convex polygon to improve the convergence rate. However, the high order derivatives are discontinuous whenever the trial interface passes through a vertex. These discontinuous points must be calculated to determine the target interval where the optimal solution is located.

In this paper, a new algorithm is proposed to enhance the robustness of the MoF method for severely deformed polygonal meshes, in which case the objective function often has multiple minimum value points. This new algorithm uses neither the initial guess nor the high order derivatives of the objective function. It only depends on the first derivative which is continuous [23]. The minimum value points of the objective function must be zero points of the first derivative. We don't compute the zero points of the first derivative directly. Instead, we calculate the minimum value points (also zero points) of the square of the first derivative, which is a nonlinear convex function on a neighborhood of each zero point. Applying the properties of convex function, the neighbor of each extreme minimum point of the objective function can be obtained. Then each zero point of it is calculated by Brent's iteration method in the corresponding neighbor. Finally, by comparing the values of the objective function at these zero points of the first derivative, the global minimum value point of the objective function can be found and is the desired solution.

The rest of this paper is outlined as follows. We begin in section 2 with a simple review of the MoF method. In section 3, our method is detailed. And several numerical tests are showed in section 4. Finally, the conclusion is drawn in section 5.

2. A simple review of the MoF method

In this section, let us review some basic concepts at first. We use similar definitions as that in [12]. Consider a mixed cell given by an open polygon $\Omega \in \mathbb{R}^2$. Let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^2$ be the vertexes of Ω indexed by counterclockwise order. The area and centroid enclosed by the polygon Ω are given by

$$|\Omega| = \frac{1}{2} \sum_{i=1}^n [\mathbf{x}_i \times \mathbf{x}_{i+1}] \quad (1)$$

$$\mathbf{x}_c(\Omega) = \frac{1}{6|\Omega|} \sum_{i=1}^n [\mathbf{x}_i \times \mathbf{x}_{i+1}] (\mathbf{x}_i + \mathbf{x}_{i+1}) \quad (2)$$

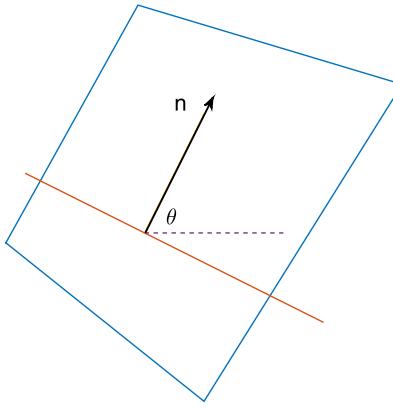


Fig. 1. Sketch map of the MoF method.

where vertex $i = n + 1$ is assumed to coincide with vertex $i = 1$, and $[\cdot \times \cdot]$ is vector product. The reference material is occupied by a non-trivial open subset $\omega \in \Omega$. The area of the reference material ω is defined as

$$|\omega| = \int_{\omega} d\mathbf{x} \quad (3)$$

and the centroid is given by

$$\mathbf{x}_c(\omega) = \frac{1}{|\omega|} \int_{\omega} \mathbf{x} d\mathbf{x}, \quad (4)$$

the volume fraction $f(\omega)$ is defined by

$$f(\omega) = \frac{|\omega|}{|\Omega|}. \quad (5)$$

The MoF interface reconstruction problem is described as follows. Suppose Ω contains only two fluid components. Adopting the same denomination as in [12], a cut-off Ω_h means a cell fraction with linear interface. The reference material centroid \mathbf{x}_c^* and the reference volume fraction f^* are given. As shown in Fig. 1, assume that an interface line forms a cut-off Ω_h in the mixed cell. It is defined by the equation

$$\mathbf{n} \cdot \mathbf{x} + \rho = 0 \quad (6)$$

where \mathbf{n} is the unit outward normal of the interface, and ρ is a constant determined by enforcement of volume conservation ($f(\omega_h) = f^*$). The interface normal vector \mathbf{n} can be identified with the polar angle θ , i.e. $\mathbf{n}(\theta) = (\cos \theta, \sin \theta)$. The interface equation will be different in each mixed cell. Our optimistic objective is to find a cut-off of the reference volume fraction f^* and minimize the difference between the reference material centroid \mathbf{x}_c^* and the approximate material centroid $\mathbf{x}_h(\theta)$. The objective function is defined by

$$f(\theta) = \| \mathbf{x}_h(\theta) - \mathbf{x}_c^* \|_2^2, \quad 0 \leq \theta \leq 2\pi \quad (7)$$

In order to calculate the minimum value of the objective function $f(\theta)$, the first derivative $f'(\theta)$ is required. Although $f(\theta)$ is an implicit function, the expression of the first derivative $f'(\theta)$ has been deduced in [12,23],

$$f'(\theta) = 2((\mathbf{x}_h(\theta) - \mathbf{x}_c^*) \cdot \mathbf{x}'_h(\theta)), \quad 0 \leq \theta \leq 2\pi \quad (8)$$

where (\dots) is a dot product, and the first derivative $\mathbf{x}'_h(\theta)$ is defined as:

- whenever the interface $\Gamma(\theta)$ consists of a single segment,

$$\mathbf{x}'_h(\theta) = -\frac{1}{12} \frac{|\Gamma(\theta)|^3}{f^* |\Omega|} (-\sin \theta, \cos \theta) \quad (9)$$

- whenever the interface $\Gamma(\theta)$ consists of several separate segments of the cut-line,

$$\mathbf{x}'_h(\theta) = -\frac{M(\Gamma(\theta))}{f^* |\Omega|} (-\sin \theta, \cos \theta) \quad (10)$$

where $M(\Gamma) = \int_{\Gamma} \|\mathbf{x} - \mathbf{x}_h(\Gamma)\|_2^2 d\Gamma$, $\mathbf{x}_h(\Gamma) = \frac{1}{|\Gamma|} \int_{\Gamma} \mathbf{x} d\Gamma$.

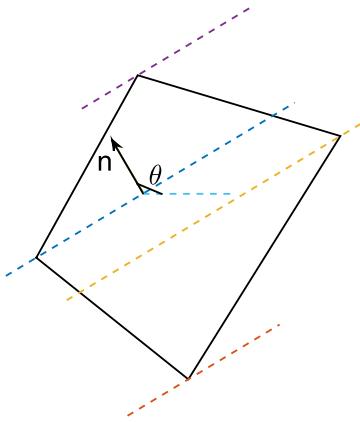


Fig. 2. Lines possessing the interface normal $\mathbf{n}(\theta)$ passing through each vertex of the polygonal cell.

3. Numerical optimization

In this section, a new algorithm is proposed to calculate the minimum value of the objective function $f(\theta)$. It is mentioned in [23] that $f'(\theta)$ is a continuous function with respect to θ . So the minimum value points of the objective function $f(\theta)$ must be its stationary points, where the first derivative $f'(\theta)$ is zero. By comparing the values of the objective function $f(\theta)$ at its stationary points and the end points of the interval $[0, 2\pi]$, the global minimum value point of it can be found. Define

$$g(\theta) = (\mathbf{x}_h(\theta) - \mathbf{x}_c^*) \cdot (-\sin \theta, \cos \theta), \quad 0 \leq \theta \leq 2\pi. \quad (11)$$

Note that $g(\theta)$ is a factor of the first derivative $f'(\theta)$. In formulae (9)–(10) $|\Gamma(\theta)|$, $M(\Gamma(\theta))$, f^* and $|\Omega|$ are all positive numbers, so $g(\theta) = 0$ is the necessary and sufficient condition for $f'(\theta) = 0$. That is to say, to find all stationary points of the objective function $f(\theta)$ is equivalent to finding all zero points of $g(\theta)$.

First, let us consider such problem as calculating the value of the function $g(\theta)$ for any given normal $\mathbf{n}(\theta)$. The interface line is determined by the interface normal \mathbf{n} and the line constant ρ . When the interface normal $\mathbf{n} = \mathbf{n}(\theta)$ is given, the line constant ρ is uniquely determined by enforcement of volume conservation. There are many approaches to solve it [3,12]. We choose the approach in [3]. Since the volume conservation is necessary, the line passes through the cell with a truncation volume $V(\rho)$ which must equal to the material reference volume V^* . The constant ρ can be found by solving the following equation

$$h(\rho) = V^* - V(\rho) = 0. \quad (12)$$

As shown in Fig. 2, lines possessing the interface normal $\mathbf{n}(\theta)$ are passed through each vertex of the polygon Ω , and the resulting truncation volumes are computed. These truncation volumes are sorted in a non-descending order

$$0 = V(\rho_1) \leq V(\rho_2) \leq \cdots \leq V(\rho_n) = |\Omega| \quad (13)$$

If $V^* = V(\rho_k)$, $1 \leq k \leq n$, ρ_k is the root of the equation (12). Otherwise, there are two truncation volumes $V(\rho_k)$, $V(\rho_{k+1})$ that bound the reference volume V^* because of $0 < V^* < |\Omega|$, i.e. $V(\rho_k) < V^* < V(\rho_{k+1})$. On the interval $[\rho_k, \rho_{k+1}]$, we have $h(\rho_k) \cdot h(\rho_{k+1}) < 0$. Then we can use Brent's method [24] to find the root of the equation (12). Brent's method is a numerical algorithm combining inverse quadratic interpolation and bisection method. After determining the value of line constant ρ , the interface line is reconstructed, and the approximate material centroid $\mathbf{x}_h(\theta)$ is uniquely determined. And then $g(\theta)$ can be subsequently calculated.

Next, let us calculate zero points of $g(\theta)$. In numerical applications, when the mesh deforms severely, the objective function often has multiple minimum value points, thus the function $g(\theta)$ has multiple zero points in the interval $[0, 2\pi]$. Here we apply the method in [25] to calculate multiple zero points of $g(\theta)$.

As in [25], let $G(\theta) = g^2(\theta)$. Suppose that there exists $\theta^* \in [0, 2\pi]$, such that $g(\theta^*) = 0$. Then according to [25], there is a neighborhood $D : |\theta - \theta^*| \leq \delta$ of θ^* , where $G(\theta)$ is a convex function on D . In [25], properties of convex function were applied to construct an algorithm for solving multiple roots of a nonlinear equation. The algorithm is specified by the following steps.

1. Given an initial guess θ_0 , which is generally recommended as $\theta_0 = 0$, and step size h .
2. If $\theta_0 + 2h \leq 2\pi$, compute $g(\theta_k)$, $g(\theta_{k+1})$, $g(\theta_{k+2})$, $G(\theta_k)$, $G(\theta_{k+1})$, $G(\theta_{k+2})$, where $\theta_k = \theta_0 + (k-1)h$.

3. Set $S = \frac{(G(\theta_{k+2}) - G(\theta_{k+1}))/(\theta_{k+2} - \theta_{k+1})}{(G(\theta_{k+1}) - G(\theta_k))/(\theta_{k+1} - \theta_k)}$. When $S \geq 0$ and $\theta_0 + (k+2)h \leq 2\pi$, if $S \leq 1$, set $\theta_{k+3} = \theta_0 + (k+3)h + h$, compute $g(\theta_{k+3}), G(\theta_{k+3})$; if $S > 1$, set $\theta_{k+3} = \theta_0 + (k+3)h + 2h$, compute $g(\theta_{k+3}), G(\theta_{k+3})$. Set $k = k + 1$ and loop step 3. If the criterion for judging in step 3 does not hold, turn to the next step.
4. Set $a_i = \theta_0 + (k-1)h, b_i = \theta_0 + (k+1)h, c_i = g(a_i), d_i = g(b_i)$, and set $m_i = d_i, b_{1i} = b_i + h, n_i = g(b_{1i})$. If $c_i \times d_i > 0$, set $\theta_0 = \theta_0 + (k+1)h$; if $m_i \times n_i \leq 0$, output k, b_i, b_{1i}, m_i, n_i and set $\theta_0 = b_{1i}$. If $c_i \times d_i \leq 0$, set $\theta_0 = b_i$ and output k, a_i, b_i, c_i, d_i . Set $i = i + 1$, go back to step 2.

According to this algorithm, k is an iterative step counter, and the output intervals $[b_i, b_{1i}]$ or $[a_i, b_i]$ are the neighbors of roots of $g(\theta) = 0$. If $g(\theta) = 0$ at any end point of the interval $[b_i, b_{1i}]$ or $[a_i, b_i]$, then the end point is the root θ_i . Otherwise, we can use Brent's method to find the root θ_i . Finally, by comparing the value of the objective function $f(\theta)$ for each zero point θ_i of the function $g(\theta)$, the global minimum value point of the objective function can be found.

4. Numerical tests

In order to verify the accuracy and robustness of the algorithm in this paper, several numerical tests are performed in this section.

First in test 4.1, we demonstrate graphically and compare the variation of the objective function and its derivative in the interval $0 < \theta < 2\pi$ for a convex cell and a concave cell. This is aimed to showing the fact that the objective function has multiple minimum value points especially on severely deformed polygonal meshes. Then in tests 4.2–4.3 we reconstruct interfaces on random quadrilateral meshes and random polygonal meshes using the MoF algorithm in [12] and our new MoF algorithm, which are called “method 1” and “method 2” respectively throughout this section. Accuracy and CPU times are compared for these two methods in tests 4.2–4.3. Then in tests 4.4–4.7, the 2D cell-centered MMALE method in [17] is used to perform four multi-material ALE simulations. In this MMALE method, the hydrodynamic equations are discretized by the cell-centered Lagrangian scheme in [26,27]. In the Lagrangian phase, Tipton's pressure relaxation model [28] is adopted as the closure model for mixed cells, the positions of material centroids are updated using the constant parametric coordinate method presented in [15]. When the Lagrangian mesh deforms severely, the MoF method is used to reconstruct interfaces in the mixed cells, and then a new mesh with better quality is generated in the rezoning phase. In the remapping phase, an integral conservative remapping method based on a polygon-intersection algorithm is used to remap such variables as density, momentum, energy, volume fraction and material centroid from the deformed Lagrangian mesh to the smoothed new mesh. The mesh smoothing strategy adopted in the rezoning stage is as follows. In test 4.4, the deformed Lagrangian mesh is just put back to initial orthogonal mesh, while Knupp's algorithm [29] is used in tests 4.5–4.7. In tests 4.4–4.7, we reconstruct the interfaces using method 1 and method 2. In tests 4.2–4.3 and test 4.7, the nested dissections algorithm in [13] is used for mixed cells which have more than two materials.

The calculations are performed on a computer with a Pentium 4 CPU core (2.8 GHz) and 1 gigabytes of main memory. For interpretation of the references to color in the figure legends of this section, the reader is referred to the web version of this article.

4.1. Graphically demonstration of the variation of objective function and its derivative

In previous sections, we emphasize that the objective function has multiple minimum value points especially on severely deformed polygonal meshes. In order to show this property numerically, we demonstrate graphically and compare the variation of the objective function $f(\theta)$ and its derivative $f'(\theta)$ in the interval $\theta \in [0, 2\pi]$ for a convex cell and a concave cell.

The first cell is a convex polygon, as shown in Fig. 3a. Its five vertexes $\{\mathbf{x}_i, i = 1, \dots, 5\}$, the reference material centroid \mathbf{x}_c^* and the reference volume fraction f^* are given as following

$$\begin{cases} \mathbf{x}_1 = (0.20, 0.70); & \mathbf{x}_2 = (0.24, 0.67); & \mathbf{x}_3 = (0.23, 0.73); \\ \mathbf{x}_4 = (0.21, 0.76); & \mathbf{x}_5 = (0.18, 0.75); \\ \mathbf{x}_c^* = (0.21, 0.71); & f^* = 0.88 \end{cases} \quad (14)$$

For this example, the graphs of the objective function $f(\theta)$ and the factor $g(\theta)$ of its derivative $f'(\theta)$ in the interval $\theta \in [0, 2\pi]$ are shown in Fig. 3b and Fig. 3c respectively. It is seen that $f(\theta)$ has only one minimum value point in the interval $[0, 2\pi]$, and $g(\theta)$ has two zero points in the interval $[0, 2\pi]$ (one of which corresponds to the minimum value point of $f(\theta)$).

The second cell is a concave polygon, as shown in Fig. 4a. Its five vertexes $\{\mathbf{x}_i, i = 1, \dots, 5\}$, the reference material centroid \mathbf{x}_c^* and the reference volume fraction f^* are given as following:

$$\begin{cases} \mathbf{x}_1 = (0.20, 0.70); & \mathbf{x}_2 = (0.24, 0.67); & \mathbf{x}_3 = (0.23, 0.73); \\ \mathbf{x}_4 = (0.21, 0.72); & \mathbf{x}_5 = (0.18, 0.75); \\ \mathbf{x}_c^* = (0.21, 0.71); & f^* = 0.58 \end{cases} \quad (15)$$

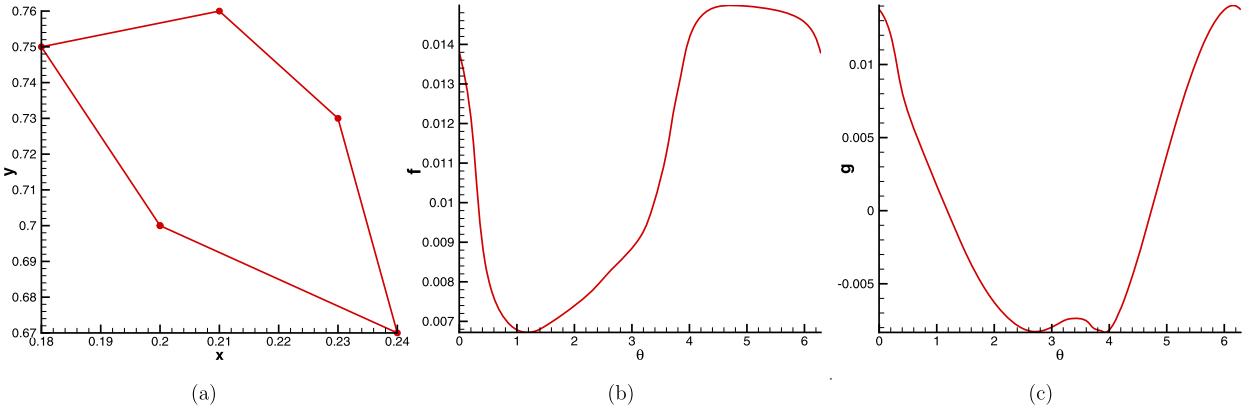


Fig. 3. Test 4.1. Graphs of the convex polygon (a), the function $f(\theta)$ (b) and the function $g(\theta)$ (c).

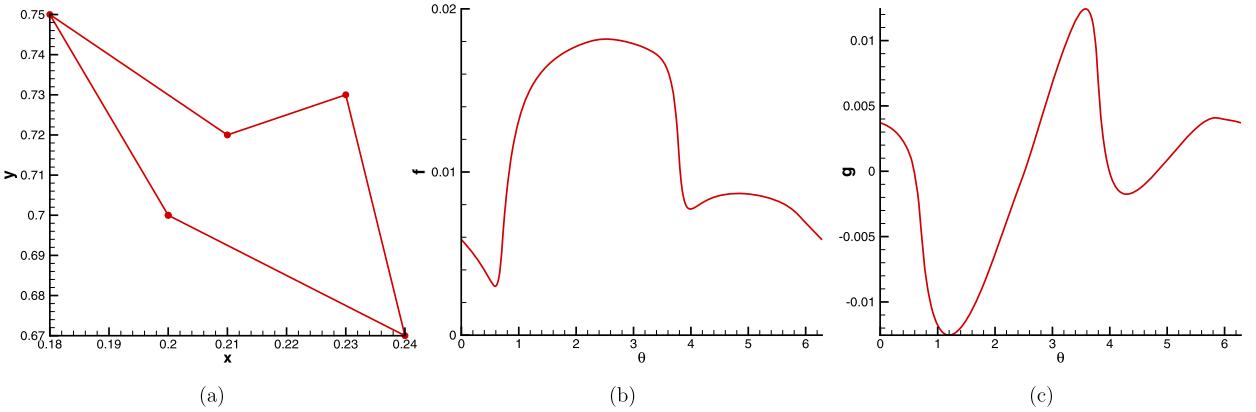


Fig. 4. Test 4.2. Graphs of the concave polygon (a), the function $f(\theta)$ (b) and the function $g(\theta)$ (c).

For this example, the graphs of the objective function $f(\theta)$ and the factor $g(\theta)$ of its derivative $f'(\theta)$ in the interval $\theta \in [0, 2\pi]$ are shown in Fig. 4b and Fig. 4c respectively. It is seen that $f(\theta)$ has two minimum value points in the interval $[0, 2\pi]$, and $g(\theta)$ has four zero points in the interval $[0, 2\pi]$ (two of which correspond to the minimum value points of $f(\theta)$).

4.2. Reconstruct interfaces on random quadrilateral meshes

The computational domain is the square $[0, 1] \times [0, 1]$. Two random meshes are generated on this domain which have $N_x \times N_y$ quadrilateral cells, where $N_x = N_y = 50$. The first mesh $\{(x_{i,j}^1, y_{i,j}^1), i = 1, \dots, N_x + 1, j = 1, \dots, N_y + 1\}$ is generated as following:

$$\begin{cases} x_{i,j}^1 = (i - 1) \cdot \Delta x + r_{i,j}^x \cdot \Delta x \cdot 1.2, \\ y_{i,j}^1 = (j - 1) \cdot \Delta y + r_{i,j}^y \cdot \Delta y \cdot 1.2 \end{cases} \quad (16)$$

where $\Delta x = 1.0/N_x$, $\Delta y = 1.0/N_y$, and $r_{i,j}^x, r_{i,j}^y$ are random numbers which satisfy

$$\begin{cases} -0.5 \leq r_{i,j}^x \leq 0.5, & -0.5 \leq r_{i,j}^y \leq 0.5, \\ r_{i,j}^x = 0, & \text{if } i = 1, \text{ or } i = N_x + 1, \\ r_{i,j}^y = 0, & \text{if } j = 1, \text{ or } j = N_y + 1 \end{cases} \quad (17)$$

The mesh is moderately distorted with several concave cells. It will be referenced as mesh 1 hereinafter.

The second mesh $\{(x_{i,j}^2, y_{i,j}^2), i = 1, \dots, N_x + 1, j = 1, \dots, N_y + 1\}$ is generated as

$$\begin{cases} x_{i,j}^2 = (i - 1) \cdot \Delta x + r_{i,j}^x \cdot \Delta x \cdot 1.3, \\ y_{i,j}^2 = (j - 1) \cdot \Delta y + r_{i,j}^y \cdot \Delta y \cdot 1.3 \end{cases} \quad (18)$$

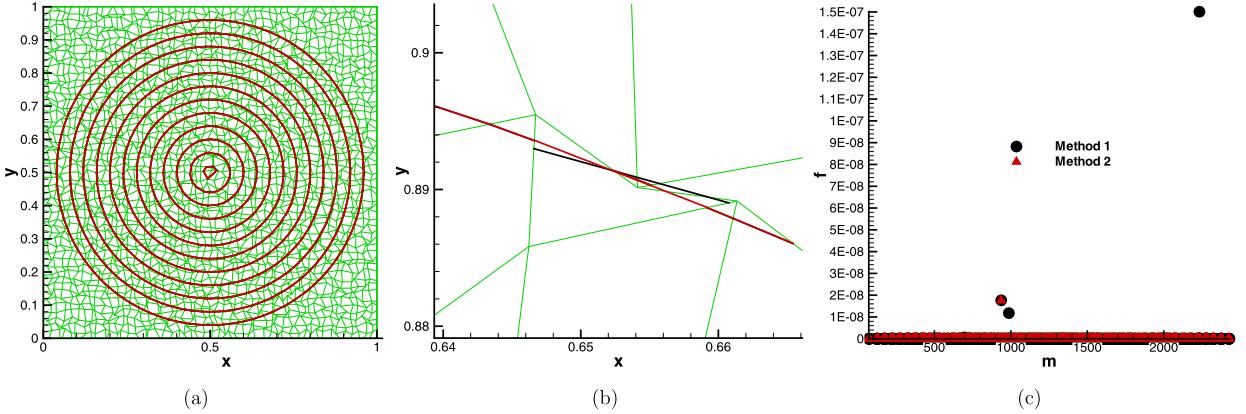


Fig. 5. Test 4.2. Reconstruct twelve circles on the random quadrilateral mesh 1. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where Δx , Δy and $r_{i,j}^x, r_{i,j}^y$ are the same as above. The mesh is severely distorted with many concave cells and some tangled cells. It will be referenced as mesh 2 hereinafter.

First, we reconstruct twelve circles on the mesh 1. These circles $\{C_k, k = 1, \dots, 12\}$ have a common center $(0.5, 0.5)$ and their radii are $\{r_k = 0.02 + (k - 1) \cdot 0.04, k = 1, \dots, 12\}$. These circles intersect with the mesh, thus make many “mixed” cells. We reconstruct these circles using method 1 and method 2 respectively. For method 2, the step size h in section 3 is set to be $h = 0.01$. The reference material centroids and the reference volume fractions in these “mixed” cells are computed as follows. First the intersection points of the circles and the mesh are computed accurately. Then in each “mixed” cell the circular arc is approximately considered as a linear segment and two cut-offs are generated in the “mixed” cell. Lastly the reference material centroids and the reference volume fractions are computed in these “mixed” cells. The results are shown in Fig. 5. In Fig. 5a–Fig. 5b, the meshes are in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. Fig. 5a is the full graph, while Fig. 5b is a zoomed local graphs. Fig. 5a shows that the circles are reconstructed very well by two methods in general. Fig. 5b shows that in concave cells, method 2 works a little better than method 1. The maximum values of the objective function computed by method 1 and method 2 are 1.50×10^{-7} and 1.76×10^{-8} respectively. Fig. 5c plots values of the objective function computed by two methods versus cell serial numbers. In Fig. 5c, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 5c shows that method 2 works a little better than method 1. The CPU times consumed by method 1 and method 2 are $2.734375s$ and $10.515625s$ respectively.

Next, we reconstruct twelve circles on the mesh 2. These circles $\{C_k, k = 1, \dots, 12\}$ are the same as above. We reconstruct these circles using method 1 and method 2 respectively. For method 2, the step size h in section 3 is set to be $h = 0.01$. The results are shown in Fig. 6. In Fig. 6a–Fig. 6d, the meshes are in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. Fig. 6a is the full graph, while Fig. 6b–Fig. 6d are zoomed local graphs. Fig. 6a shows that the circles are reconstructed very well by two methods in general. Fig. 6b shows that in concave cells, method 2 works a little better than method 1. Fig. 6c and Fig. 6d show that method 2 works well even in tangled cells, while method 1 fails. The maximum values of the objective function computed by method 1 and method 2 are 2.37758 and 3.44×10^{-8} respectively. Fig. 6e and Fig. 6f plot values of the objective function computed by two methods versus cell serial numbers. In Fig. 6e and Fig. 6f, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 6e shows all values of the objective function, while Fig. 6f shows values between 0.0 and 4×10^{-6} . Fig. 6e and Fig. 6f show that method 2 works better than method 1. The CPU times consumed by method 1 and method 2 are $2.9375s$ and $10.71875s$ respectively.

Next, we generate randomly the reference material centroids and the reference volume fractions for all cells of the mesh 2 and reconstruct the interface in each cell using method 1 and method 2. For method 2, the step size h is set to be $h = 0.01$. The reference volume fraction is generated directly by a random number $0 < f^* < 1$. The reference material centroid is generated as follows. Suppose that the vertexes of the cell are $\{\mathbf{x}_i, i = 1, \dots, 4\}$. For each vertex \mathbf{x}_i , a random number $0 < \mu_i < 1$ is generated. Let μ_{sum} is the sum of the four random numbers $\{\mu_i, i = 1, \dots, 4\}$, and let

$$\tilde{\mu}_i = \frac{\mu_i}{\mu_{sum}}, \quad i = 1, \dots, 4 \quad (19)$$

Then the reference material centroid is computed as

$$\mathbf{x}_c^* = \sum_{i=1}^4 \tilde{\mu}_i \mathbf{x}_i \quad (20)$$

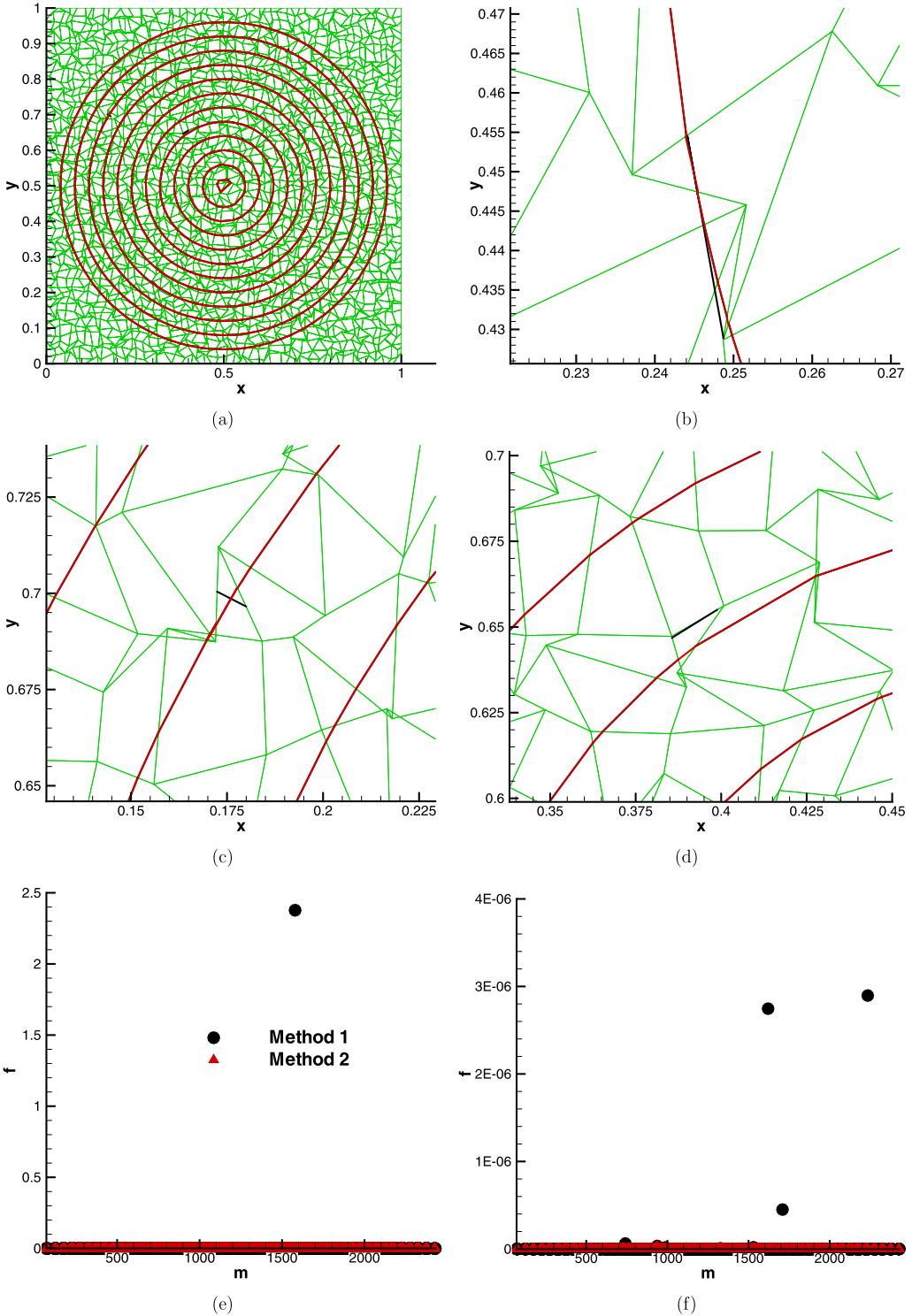


Fig. 6. Test 4.2. Reconstruct twelve circles on the random quadrilateral mesh 2.

The results are shown in Fig. 7. Fig. 7a is a zoomed local graph. In Fig. 7a, the mesh is in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. The maximum values of the objective function computed by method 1 and method 2 are 12.0232 and 5.3373×10^{-4} respectively. Fig. 7b–Fig. 7f plot values of the objective function computed by two methods versus cell serial numbers. In Fig. 7b–Fig. 7f, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 7b shows all values of the

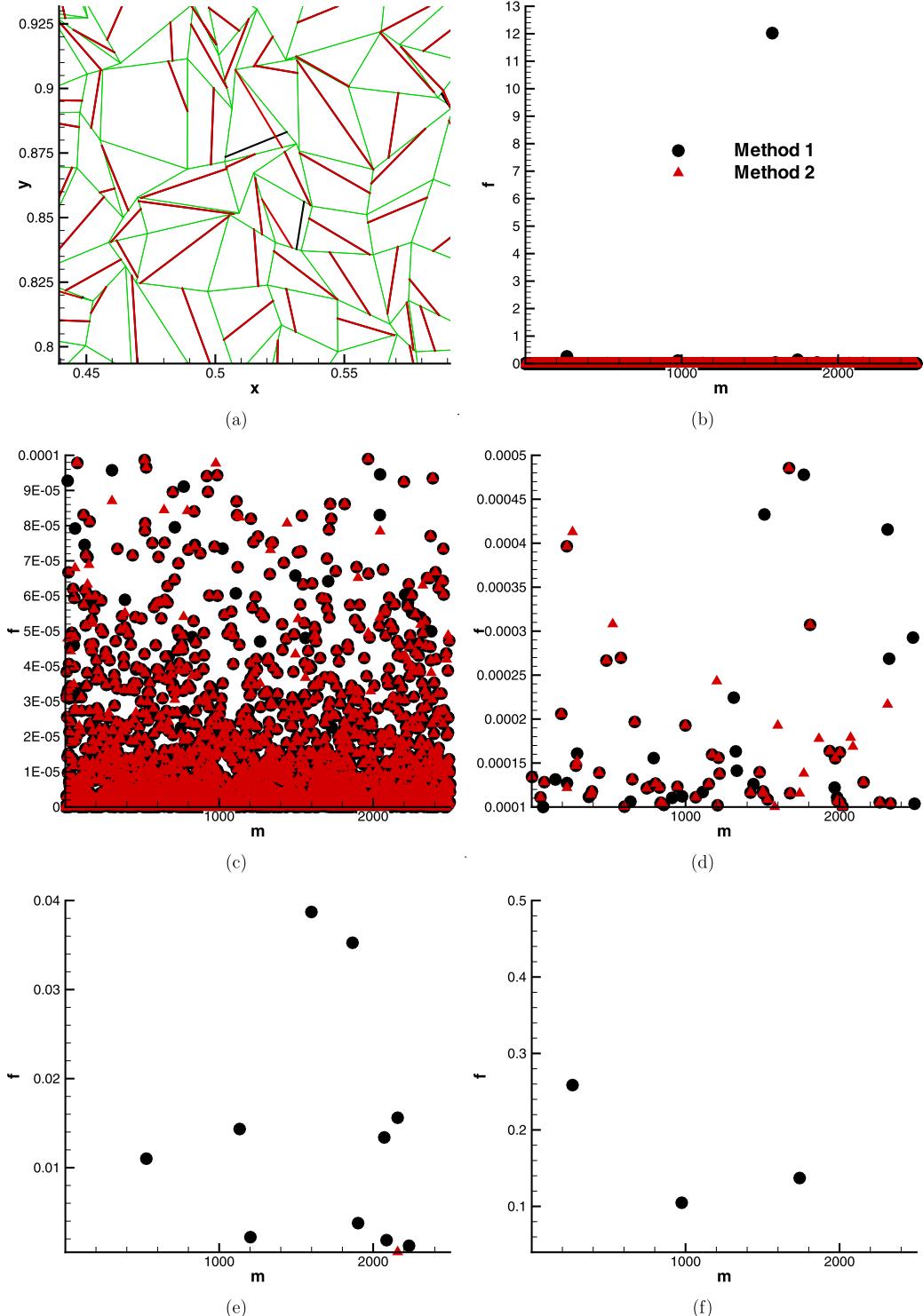


Fig. 7. Test 4.2. Reconstruct interfaces in all cells of the random quadrilateral mesh 2 for random reference material centroids and random reference volume fractions.

objective function. Fig. 7c shows values between 0.0 and 0.0001. Fig. 7d shows values between 0.0001 and 0.0005. Fig. 7e shows values between 0.0005 and 0.04. Fig. 7f shows values between 0.04 and 0.5. Fig. 7b–Fig. 7f show that method 2 works much better than method 1. The CPU times consumed by method 1 and method 2 are 8.328125s and 20.078125s respectively.

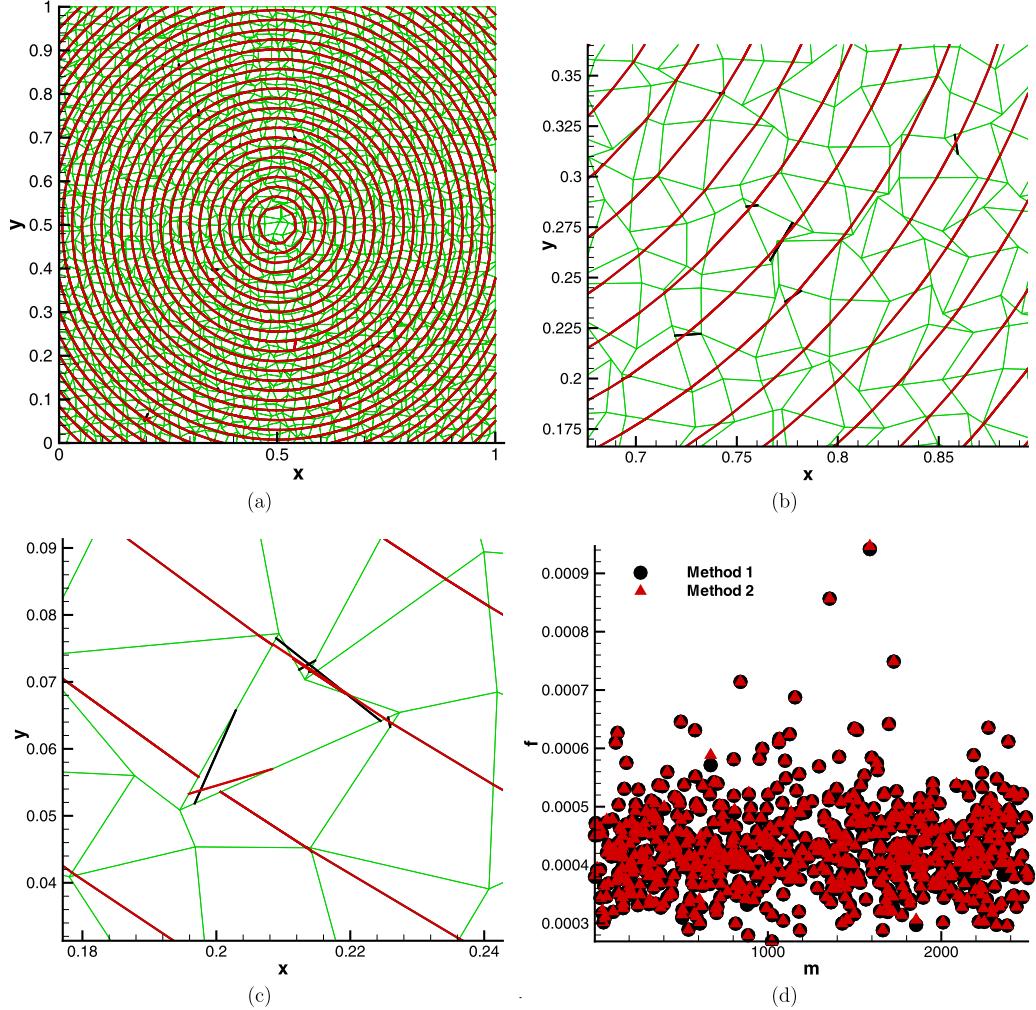


Fig. 8. Test 4.2. Reconstruct 29 circles on the random quadrilateral mesh 1.

Finally, we reconstruct 29 circles on the mesh 1. These circles $\{C_k, k = 1, \dots, 29\}$ have a common center $(0.5, 0.5)$ and their radii are $\{r_k = 0.02 + k \cdot 0.0225, k = 1, \dots, 29\}$. These circles intersect with the mesh. Some cells intersect with one of these circles, and many cells intersect with two of these circles thus make many “mixed” cells with three “materials”. The coordinates of several vertexes of the mesh are adjusted artificially and slightly so that none of the cells of the mesh intersect with more than two circles. We reconstruct these circles using method 1 and method 2 respectively. For method 2, the step size h in section 3 is set to be $h = 0.01$. The results are shown in Fig. 8. In Fig. 8a–Fig. 8c, the meshes are in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. Fig. 8a is the full graph, while Fig. 8b and Fig. 8c are zoomed local graphs. Fig. 8a shows that the circles are reconstructed rather well by two methods in general. Fig. 8b shows that method 2 works a little better than method 1 in many cells (include convex cells and concave cells). Fig. 8c shows that both methods work not very well in several concave cells. The maximum values of the objective function computed by method 1 and method 2 are 9.4157×10^{-4} and 9.4674×10^{-4} respectively. Fig. 8d plots values of the objective function computed by two methods versus cell serial numbers for “mixed” cells with three “materials”. In Fig. 8d, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 8d shows that the results of the two methods are comparable, which is a bit strange. The CPU times consumed by method 1 and method 2 are 89.671875s and 171.125s respectively.

4.3. Reconstruct interfaces on random polygonal meshes

The computational domain is the square $[0, 1] \times [0, 1]$. Two random meshes are generated on this domain which have 830 polygonal cells and 1662 nodes. The first mesh $\{(x_n^1, y_n^1), n = 1, \dots, 1662\}$ is generated as follows. Firstly, an initial

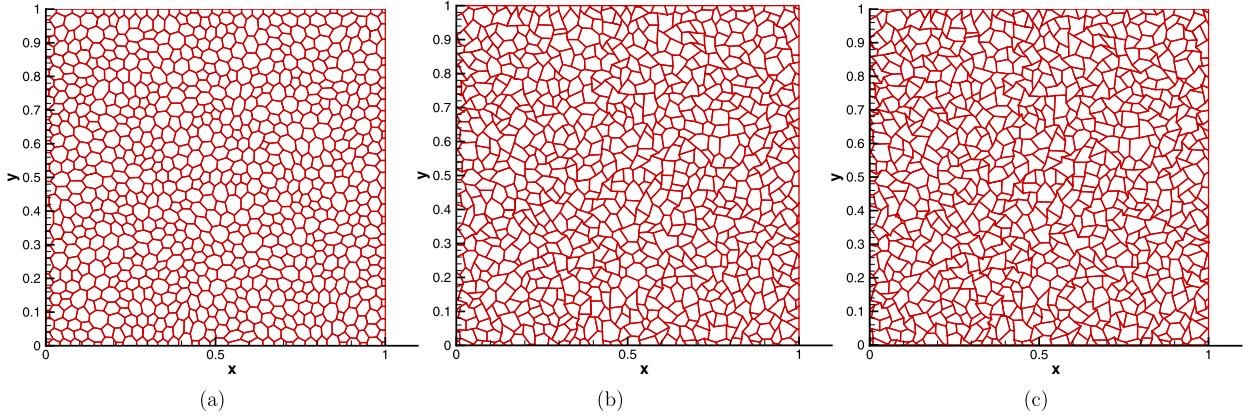


Fig. 9. Test 4.3. An initial polygonal mesh (a) and two random polygonal meshes (b), (c).

polygonal mesh $\{(x_n^0, y_n^0), n = 1, \dots, 1662\}$ is generated, which is displayed in Fig. 9a. There are triangles, quadrilaterals, pentagons, hexagons, heptagons and octagons in the mesh.

Then for each node (x_n^0, y_n^0) , $n = 1, \dots, 1662$, two random numbers (r_n^x, r_n^y) are generated which satisfy

$$\begin{cases} -0.5 \leq r_n^x \leq 0.5, & -0.5 \leq r_n^y \leq 0.5, \\ r_n^x = 0, & \text{if } x_n^0 = 0, \text{ or } x_n^0 = 1, \\ r_n^y = 0, & \text{if } y_n^0 = 0, \text{ or } y_n^0 = 1 \end{cases} \quad (21)$$

Finally, the random polygonal mesh is defined as

$$\begin{cases} x_n^1 = x_n^0 + r_n^x \cdot \Delta x, \\ y_n^1 = y_n^0 + r_n^y \cdot \Delta y \end{cases} \quad (22)$$

where $\Delta x = 0.02$, $\Delta y = 0.02$. The second random mesh $\{(x_n^2, y_n^2), n = 1, \dots, 1662\}$ is generated similarly with $\Delta x = 0.025$, $\Delta y = 0.025$. These two random meshes are displayed respectively in Fig. 9b and Fig. 9c and will be referenced as mesh 1 and mesh 2 respectively hereinafter. The mesh 1 is moderately distorted. It has some concave cells. The mesh 2 is severely distorted. It has many concave cells and some tangled cells.

First, we reconstruct eleven circles on the mesh 1. These circles $\{C_k, k = 1, \dots, 11\}$ have a common center $(0.5, 0.5)$ and their radii are $\{r_k = 0.02 + k \cdot 0.04, k = 1, \dots, 11\}$. These circles intersect with the mesh, thus make many “mixed” cells. We reconstruct these circles using method 1 and method 2 respectively. For method 2, the step size h is set to be $h = 0.01$. The reference material centroids and the reference volume fractions in these “mixed” cells are computed as in Test 4.2. The results are shown in Fig. 10. In Fig. 10a–Fig. 10b, the meshes are in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. Fig. 10a is the full graph, while Fig. 10b is a zoomed local graphs. Fig. 10a shows that the circles are reconstructed very well by two methods in general. Fig. 10b shows that in concave cells, method 2 works a little better than method 1. The maximum values of the objective function computed by method 1 and method 2 are 2.70×10^{-8} and 1.06×10^{-10} respectively. Fig. 10c plots values of the objective function computed by two methods versus cell serial numbers. In Fig. 10c, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 10c shows that method 2 works a little better than method 1. The CPU times consumed by method 1 and method 2 are 1.765625s and 7.34375s respectively.

Next, we reconstruct eleven circles on the mesh 2. These circles $\{C_k, k = 1, \dots, 11\}$ are the same as above. We reconstruct these circles using Method 1 and Method 2 respectively. For method 2, the step size h is set to be $h = 0.01$. The results are shown in Fig. 11. In Fig. 11a–Fig. 11c, the meshes are in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. Fig. 11a is the full graph, while Fig. 11b–Fig. 11c are zoomed local graphs. Fig. 11a shows that the circles are reconstructed very well by two methods in general. Fig. 11b indicates that in concave cells, method 2 works a little better than method 1, and that two methods are both capable of reconstructing multiple interfacial segments in a given cell. Fig. 11c illustrates that method 2 works well even in tangled cells, while method 1 fails. The maximum values of the objective function computed by method 1 and method 2 are 0.3352 and 6.8478×10^{-8} respectively. Fig. 11d plots values of the objective function computed by two methods versus cell serial numbers in all “mixed” cells. In Fig. 11d, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 11d shows that method 2 works better than method 1. The CPU times consumed by method 1 and method 2 are 1.859375s and 7.625s respectively.

Next, we generate randomly the reference material centroids and the reference volume fractions for all cells of the mesh and reconstruct the interface in each cell using method 1 and method 2. For method 2, the step size h is set to be $h = 0.01$.

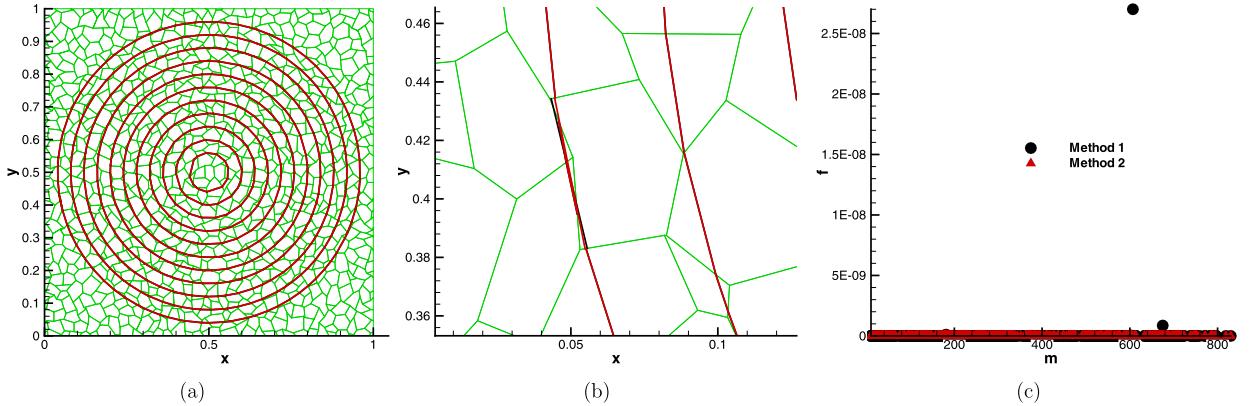


Fig. 10. Test 4.3. Reconstruct eleven circles on the random polygonal mesh 1.

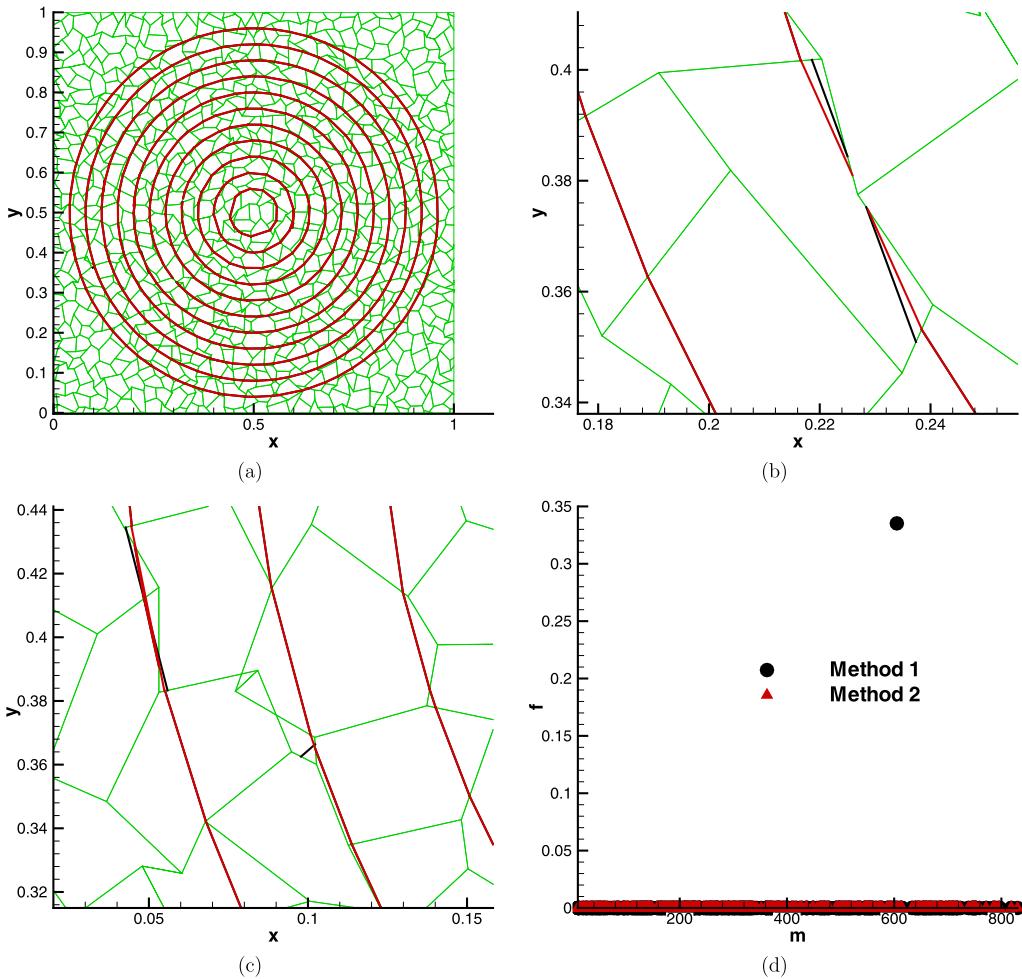


Fig. 11. Test 4.3. Reconstruct eleven circles on the random polygonal mesh 2.

The reference volume fractions and the reference material centroids are generated as in Test 4.2. The results are shown in Fig. 12. Fig. 12a is a zoomed local graph. In Fig. 12a, the mesh is in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. The maximum values of the objective function computed by method 1 and method 2 are 0.2255 and 4.005×10^{-4} respectively. Fig. 12b–Fig. 12d plot values of the objective function computed by two methods versus cell serial numbers. In Fig. 12b–Fig. 12d, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 12b shows all values of the objective function. Fig. 12c

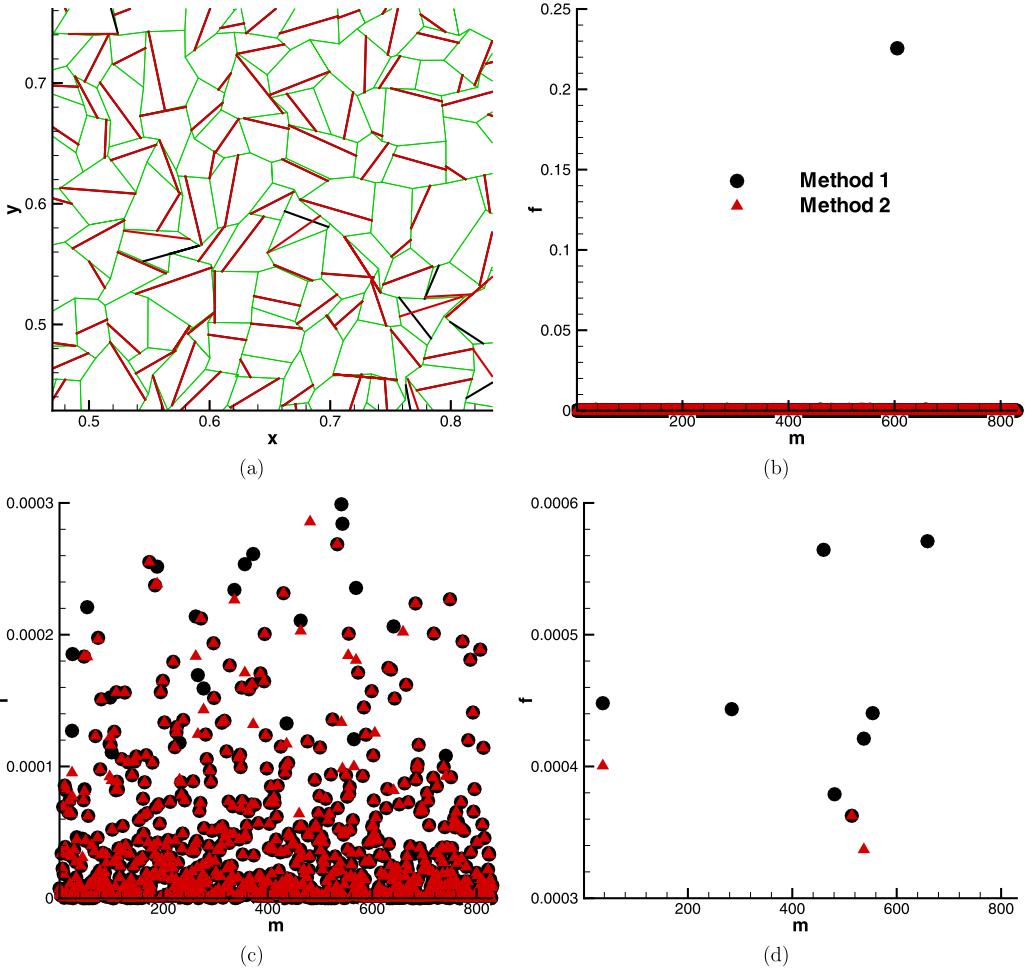


Fig. 12. Test 4.3. Reconstruct interfaces in all cells of the random polygonal mesh 2 for random reference material centroids and random reference volume fractions.

shows values between 0.0 and 0.0003. Fig. 12d shows values between 0.0003 and 0.0006. Fig. 12b–Fig. 12d show that method 2 works better than method 1. The CPU times consumed by method 1 and method 2 are 3.171875s and 9.046875s respectively.

Finally, we reconstruct 27 circles on the initial polygonal mesh. These circles $\{C_k, k = 1, \dots, 27\}$ have a common center $(0.5, 0.5)$ and their radii are $\{r_k = 0.02 + k \cdot 0.025, k = 1, \dots, 27\}$. These circles intersect with the mesh. Some cells intersect with one of these circles, and many cells intersect with two of these circles thus make many “mixed” cells with three “materials”. We reconstruct these circles using method 1 and method 2 respectively. For method 2, the step size h in section 3 is set to be $h = 0.01$. For this example, method 2 successfully reconstructed interfaces in all “mixed” cells, while method 1 failed in three “mixed” cells whose serial numbers are 127, 255, 623. The results are shown in Fig. 13. In Fig. 13a–Fig. 13c, the meshes are in green, while the interfaces reconstructed by method 1 and method 2 are in black and red respectively. Fig. 13a is the full graph, while Fig. 13b and Fig. 13c are zoomed local graphs. Fig. 13a shows that the circles are reconstructed rather well by two methods in general. Fig. 13b and Fig. 13c show that method 2 works a little better than method 1 in many cells. The maximum values of the objective function computed by method 1 and method 2 are 1.3492×10^{-3} and 1.3489×10^{-3} respectively. Fig. 13d plots values of the objective function computed by two methods versus cell serial numbers for “mixed” cells with three “materials”. In Fig. 13d, black solid dots represent values computed by method 1, and red solid triangles stand for values computed by method 2. Fig. 13d shows that the results of the two methods are comparable. The CPU times consumed by method 1 and method 2 are 82.6406s and 146.5468s respectively.

4.4. Rayleigh–Taylor instability problem

A Rayleigh–Taylor problem [30,17] is considered here. The computational domain is $[0, 1/3] \times [0, 1]$ and is initially paved with a Cartesian grid with 35×100 rectangle cells. Initially, the heavier gas with density $\rho_h = 2$ and the lighter gas with density $\rho_l = 1$ are separated by a perturbed interface which is described by formula $y_i(x) = 0.5 + 0.01 \cos(6\pi x)$. The heavier

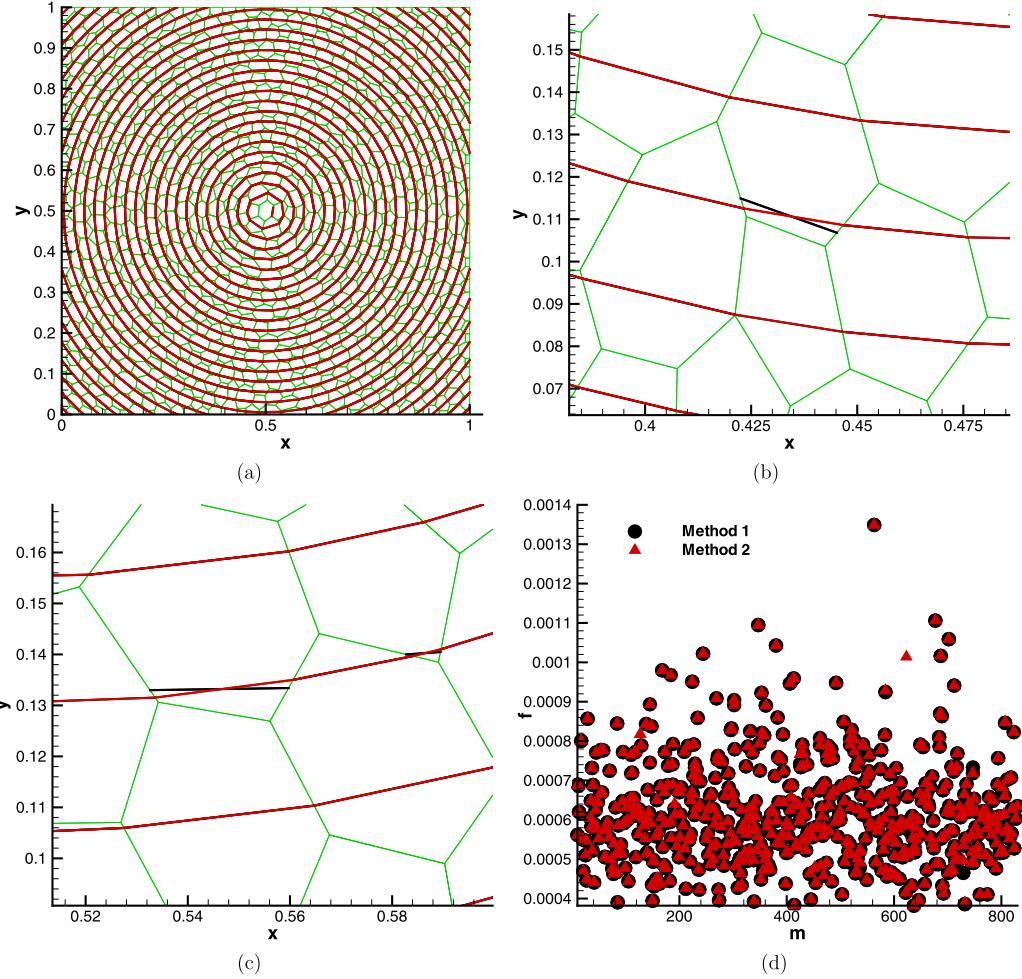


Fig. 13. Test 4.3. Reconstruct 27 circles on the initial polygonal mesh.

gas is above the lighter gas and for both gases the adiabatic constant are $\gamma = 1.4$. A vertically downward gravity field is applied as $\mathbf{g} = (g_x, g_y)^T = (0, -0.1)^T$. Initially both gases are at rest and the pressure distribution is deduced by setting hydrostatic equilibrium as

$$\begin{cases} p_h(x, y) = 1 + \rho_h g_y(y - 1), & y > y_i(x) \\ p_l(x, y) = 1 + \rho_h g_y(y_i(x) - 1) + \rho_l g_y(y - y_i(x)), & y \leq y_i(x) \end{cases} \quad (23)$$

It is well known that this configuration is unstable. Due to the sinusoidal interface, vortices develop in the vicinity of the interface. As time goes on, the heavier gas goes down into the lighter gas to form a spike and the lighter gas is going up into the heavier gas to form a bubble. We compute this problem using the cell-centered MMALE method in [17], and use method 1 and method 2 to reconstruct the interfaces. For method 2, the step size h is set to be $h = 0.001$. The meshes and the interfaces at time instances $t = 7, 8, 9, 10$ are plotted in Fig. 14. In Fig. 15, the interfaces at times $t = 9, 10$ computed by two methods are plotted on one mesh. The pressure contours at $t = 10$ computed by two methods are plotted in Fig. 16. The results in Fig. 14 indicate that the two MoF methods performed rather well in general. Fig. 15 and Fig. 16 show that method 2 conserved symmetry more good than method 1 did.

4.5. Sedov problem

The well-known Sedov problem [31] is considered in this test case. This is a point explosion problem model. The computational domain is the square $\Omega = [0, 1.125] \times [0, 1.125]$ and the initial mesh is a Cartesian grid with 45×45 rectangle cells. The initial density is $\rho = 1$, and the velocity is $(u, v) = (0, 0)$. At the origin, there is a unit internal energy. And the initial pressure is $p = 0$ at the remaining points. The specific internal energy on the first cell closer to the origin is $e_0 = 400$. At the initial time, the interface is located on the circle whose center is at the origin and radius is 0.43. We note that in the

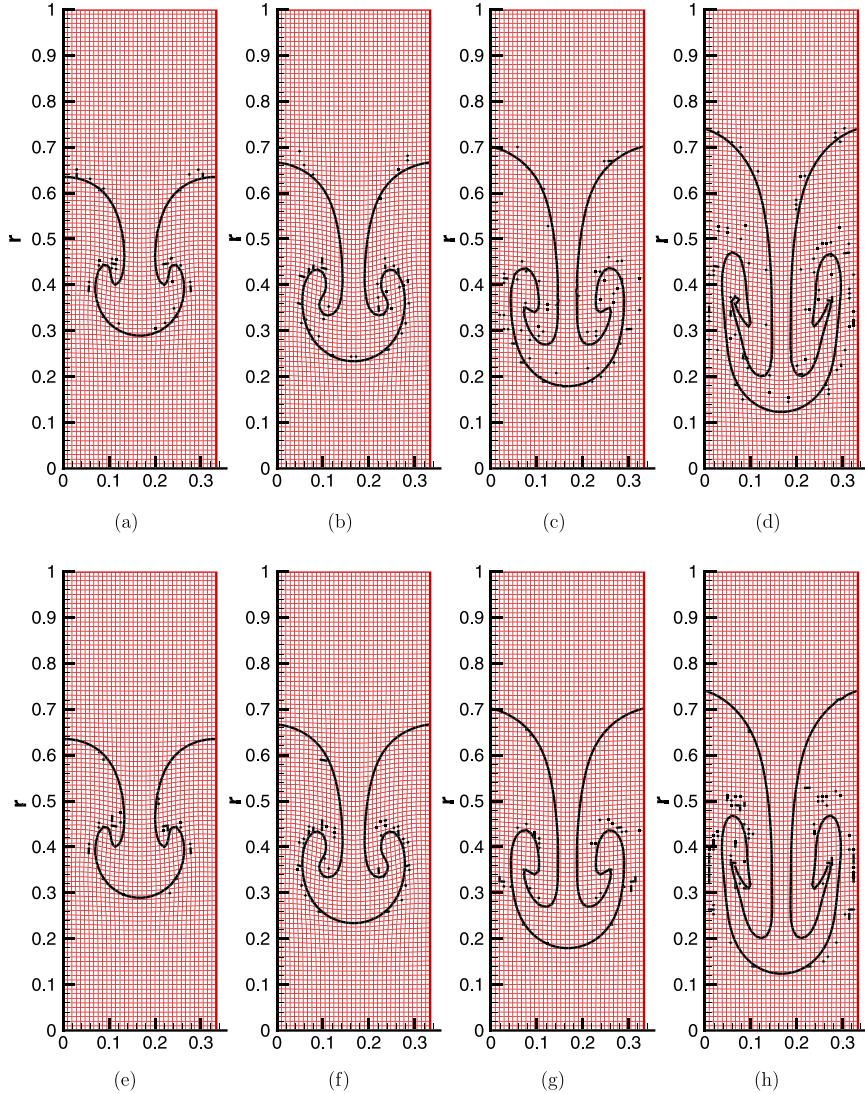


Fig. 14. Test 4.4. Rayleigh-Taylor instability. Meshes and interfaces at $t = 7, 8, 9, 10$ (from left to right). Results in subgraphs (a)–(d) are computed by method 1, while results in subgraphs (e)–(h) are computed by method 2.

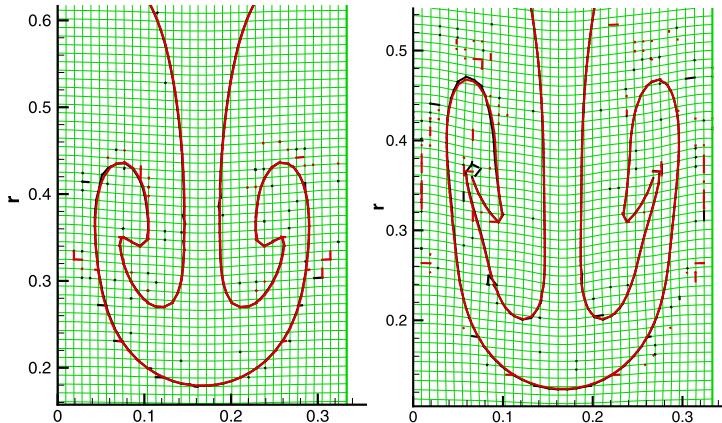


Fig. 15. Test 4.4. Rayleigh-Taylor instability. Meshes and interfaces at $t = 9, 10$ (from left to right). The interfaces computed by two methods are plotted on one mesh. The meshes are in green, while the interfaces computed by method 1 and method 2 are in black and red respectively.

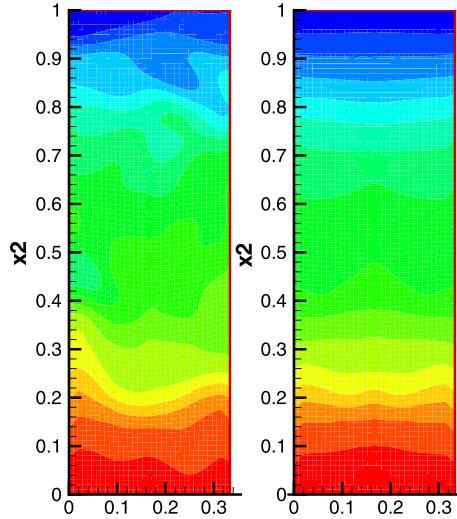


Fig. 16. Test 4.4. Rayleigh–Taylor instability. Pressure contours at $t = 10$. Left: results computed by method 1; Right: results computed by method 2.

mixed cells the two materials are indeed perfect gases with the same polytropic index $\gamma = 1.4$, and that we treat them as mixed cells to compare the numerical solutions with the analytical solutions. The analytical solution of the problem is that the shock wave is located at the radius $r = 1$ at the time $t = 1$ and the peak density reaches 6.

We compute this problem using the cell-centered MMALE method in [17], and use method 1 and method 2 to reconstruct the interfaces. For method 2, the step size h is set to be $h = 0.003$. Rezoning and remapping were done only at five time instances $t = 0.2, 0.4, 0.6, 0.8, 1.0$. The meshes and the interfaces at time instances $t = 0.25, 0.5, 0.75, 1.0$ are plotted in Fig. 17. The density at centroids of all cells versus the distance of the centroids to the origin at $t = 1$ are plotted in Fig. 18, in which the solid line represents the analytical solution and dots represent the numerical solution. These results show that the two MoF methods performed rather satisfactorily and the calculated shock positions are both accurate. For this problem, the results computed by two methods are comparable.

4.6. Bubble Shock interaction problem

This test case corresponds to the interaction of shock wave with a cylindrical Helium bubble surrounded by air at rest [30,17]. The computational domain is $[0, 0.65] \times [-0.089, 0.089]$ and the initial mesh is a Cartesian grid with 260×72 rectangle cells. As shown in Fig. 19, the bubble is a disk defined by its center $(x_c, y_c) = (0.32, 0)$ and radius $r = 0.025$. A piston-like boundary condition is given at the right boundary (initially at $x = 0.65$) which moves inward with the velocity $(u^*, 0)$. And at the other boundaries, the boundary conditions for the solid wall are given. The Mach number of the incident shock wave is $M_s = 1.22$. The bubble and the air are initially at rest. The initial data for Helium and air are $(\rho_1, P_1) = (0.182, 10^5)$, $(\rho_2, P_2) = (1, 10^5)$, respectively. The molar masses and polytropic indices of Helium and air are $(M_1, \gamma_1) = (5.269 \times 10^{-3}, 1.648)$, $(M_2, \gamma_2) = (28.963 \times 10^{-3}, 1.4)$, respectively. Using the Rankine–Hugoniot relations, the x-velocity of the piston $u^* = -124.824$ can be obtained. The x-component of the incident shock velocity is $D_c = -456.482$. The incident shock wave hits the helium bubble at time $t_i = 668.153 \times 10^{-6}$. We compute this problem using the cell-centered MMALE method in [17], and use method 1 and method 2 to reconstruct the interfaces. For method 2, the step size h is set to be $h = 0.001$. The calculation termination time is $t_{final} = t_i + 674 \times 10^{-6} = 1342.153 \times 10^{-6}$. The computed meshes and interfaces at three time instances $t_1 = t_i + 245 \times 10^{-6} = 913.153 \times 10^{-6}$, $t_2 = t_i + 427 \times 10^{-6} = 1095.153 \times 10^{-6}$ and t_{final} are plotted in Fig. 20, together with the experimental Schlieren maps of these three time instances [32] for reference and comparison. The results in Fig. 20 agree well with the experimental Schlieren images in [32], and the numerical results in [30,17]. At time t_1 , the shock wave has struck the bubble. At later time instances, t_2 and t_{final} , the shape of bubble is strongly distorted. The results indicate that the two MoF methods performed rather well in general. In Fig. 21, the interfaces at the time t_{final} computed by two methods are plotted on one mesh. Fig. 21 shows that the interfaces in several mixed cells reconstructed by method 2 are more accurate than that reconstructed by method 1.

4.7. Triple point problem

This is a two-material problem [30] which corresponds to a three states two-dimensional Riemann problem in a rectangular domain displayed in Fig. 22. The initial domain $\Omega = [0, 7] \times [0, 3]$ is split into the following three sub-domains $\Omega_1 = [0, 1] \times [0, 3]$; $\Omega_2 = [1, 7] \times [0, 1.5]$ and $\Omega_3 = [1, 7] \times [1.5, 3]$. The sub-domain Ω_1 contains a high-pressure high-density gas whose initial state is $(\rho_1, p_1, \mathbf{U}_1) = (1, 1, \mathbf{0})$, where ρ , p , $\mathbf{U} = (u_x, u_y)$ are the density, the pressure and the velocity vector respectively. The sub-domain Ω_2 contains a low-pressure high-density gas whose initial state is $(\rho_2, p_2, \mathbf{U}_2) = (1, 0.1, \mathbf{0})$.

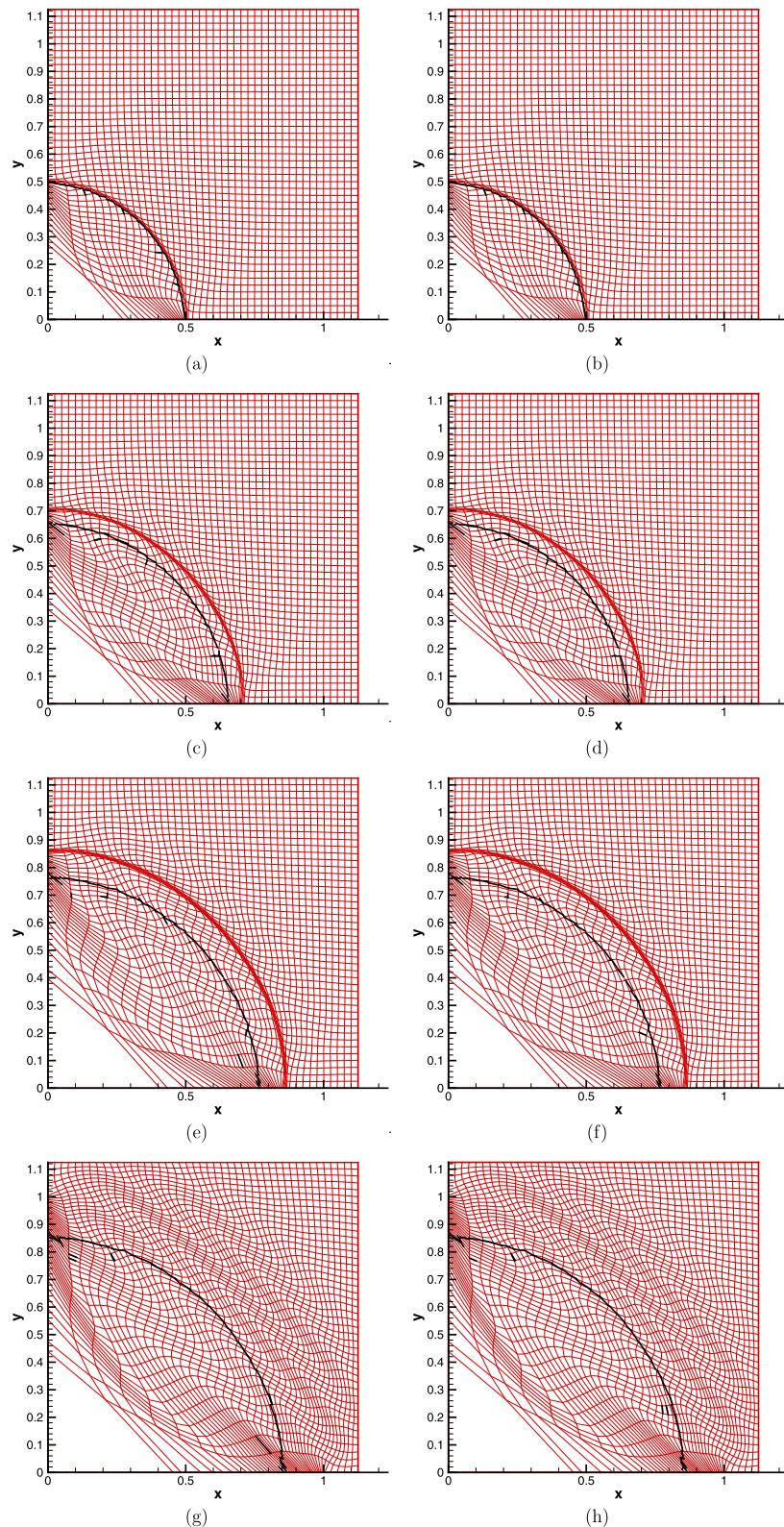


Fig. 17. Test 4.5. Sedov problem. Meshes and interfaces at time instances $t = 0.25, 0.5, 0.75, 1.0$ (from top to bottom). Left: results computed by method 1; Right: results computed by method 2.

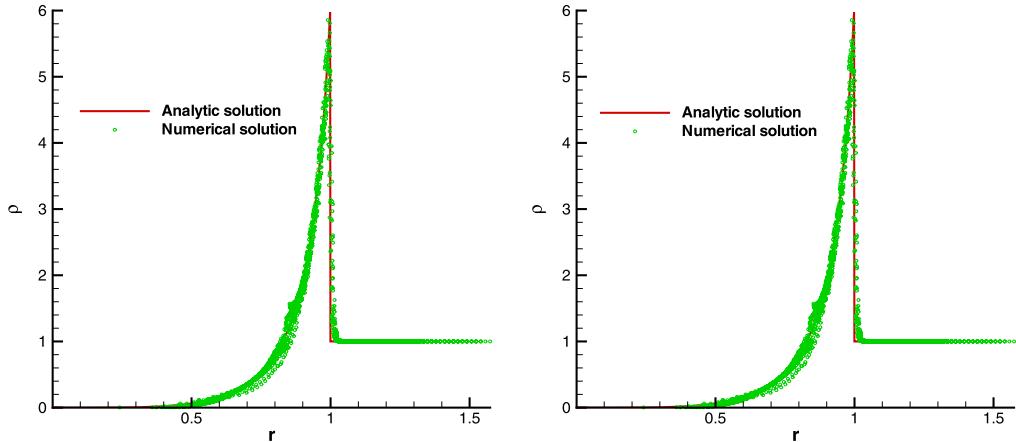


Fig. 18. Test 4.5. Sedov problem. The density at centroids of all cells versus the distance of the centroids to the origin at $t = 1.0$. Left: results computed by method 1; Right: results computed by method 2.

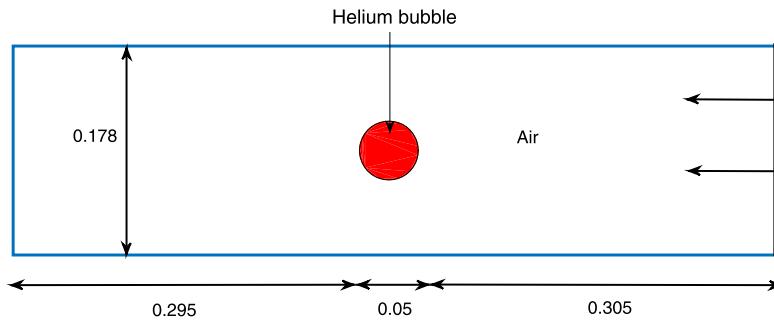


Fig. 19. Test 6. Bubble shock interaction. Geometrical data of the initial domain.

The sub-domain Ω_3 contains a low-pressure low-density gas whose initial state is $(\rho_3, p_3, \mathbf{U}_3) = (0.125, 0.1, \mathbf{0})$. The sub-domains Ω_1 and Ω_3 are filled with the same material characterized by the polytropic index $\gamma_1 = \gamma_3 = 1.5$ whereas the sub-domain Ω_2 is filled with a different material with $\gamma_2 = 1.4$. The boundary conditions are wall boundary conditions.

The final time of the simulation is $t_{final} = 8$. The initial mesh is paved with 70×30 rectangular cells. Due to the difference of the acoustic impedance, two shocks in domains Ω_2 and Ω_3 propagate with different speeds. This creates a strong shear along the initial contact discontinuity located at the interface between Ω_2 and Ω_3 . This shear produces a Kelvin–Helmholtz instability and a vortex formation occurs. We note that the Lagrangian computation of this problem fails before the vortex development due to mesh tangling [30]. Capturing the vortex is difficult when traditional ALE method is used. We compute this problem using the cell-centered MMALE method in [17], and use method 1 and method 2 to reconstruct the interfaces. For method 2, the step size h is set to be $h = 0.002$ and $h = 0.1$ respectively. The meshes and the interfaces at $t = 3$, $t = 5$ and $t = 8$ are shown in Fig. 23. It is easily seen that the results of method 2 are better than that of method 1, and that the results of method 2 with $h = 0.002$ are better than that of method 2 with $h = 0.1$. This test shows that for method 2 the step size h can not be selected to be too large. This is because that method 2 applies the properties of convex function, and that the selected value of h must ensure that $G(\theta)$ in section 3 is a convex function on corresponding neighborhood.

5. Conclusions

In this paper, a new algorithm is presented to achieve robust numerical optimization in the MoF method. When the MoF method is used on severely deformed meshes, the objective function often has multiple minimum value points. By using an efficient method for solving multiple roots of the nonlinear equation, these multiple minimum value points can be calculated. Then by comparing the values of the objective function at these points and the end points of the interval $[0, 2\pi]$, the global minimum value point of the objective function can be obtained. The advantage of the new algorithm is that it does not need to make an initial guess which should be carefully chosen in previous algorithms, and it can obtain the unique global minimum of the objective function rather than other local minima when it is used on severely distorted meshes. This advantage is gained at the cost of more computational expense compared to previous algorithms. This new

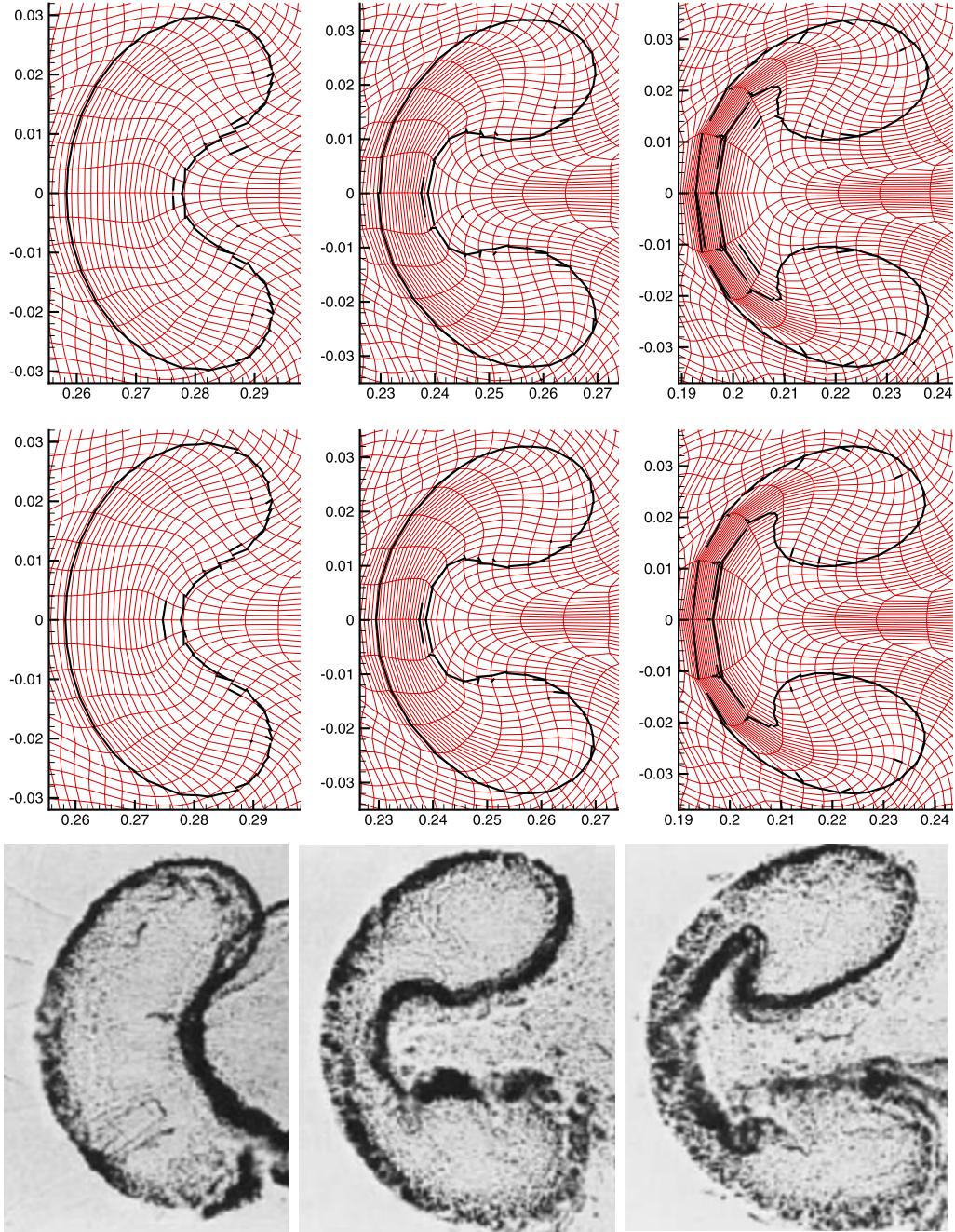


Fig. 20. Test 4.6. Bubble shock interaction. Meshes and interfaces computed by method 1 (top) and method 2 (middle), and Schlieren images from experimental data [32] (bottom) at three time instances t_1 , t_2 , t_{final} (from left to right).

algorithm can be generalized to 3D MoF method. Following the idea of finding global minimum of the objective function, we have developed a new 3D MoF method using a global optimization method. The manuscript is being prepared for this new 3D MoF method.

In the numerical results of tests 4.4–4.7, there are some mixed cells with small volume fractions near the material interfaces, and interface segments in the form of wisps are observed. This is due to small errors of the polygon-intersection algorithm in [17], just as the polyhedron-intersection algorithm in [16] which was improved in [18]. The polygon-intersection algorithm in [17] can be improved following the idea of the polyhedron-intersection algorithm in [18]. This will be our future work.

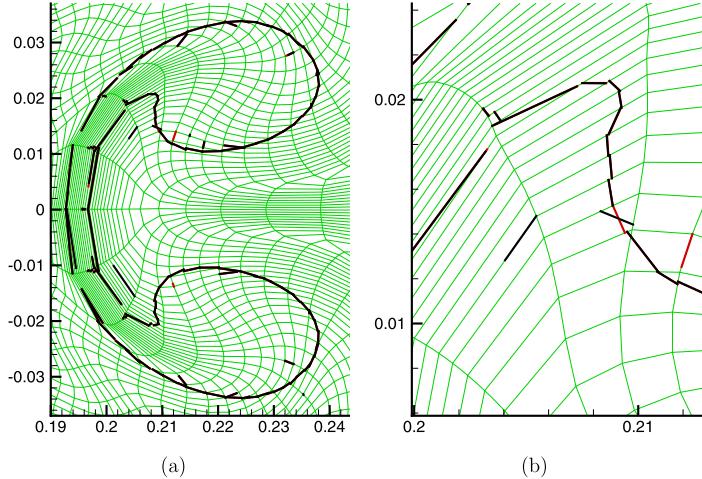


Fig. 21. Test 4.6. Bubble shock interaction. Meshes and interfaces at t_{final} . The interfaces computed by two methods are plotted on one mesh. The meshes are in green, while the interfaces computed by method 1 and method 2 are in black and red respectively. Subgraph *a* shows the whole bubble, and subgraph *b* is a zoomed local graph.

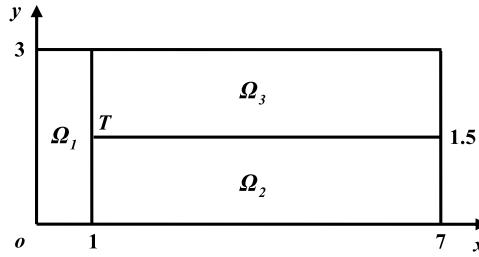


Fig. 22. Test 4.7. Triple point problem. Geometrical data of the initial domain.

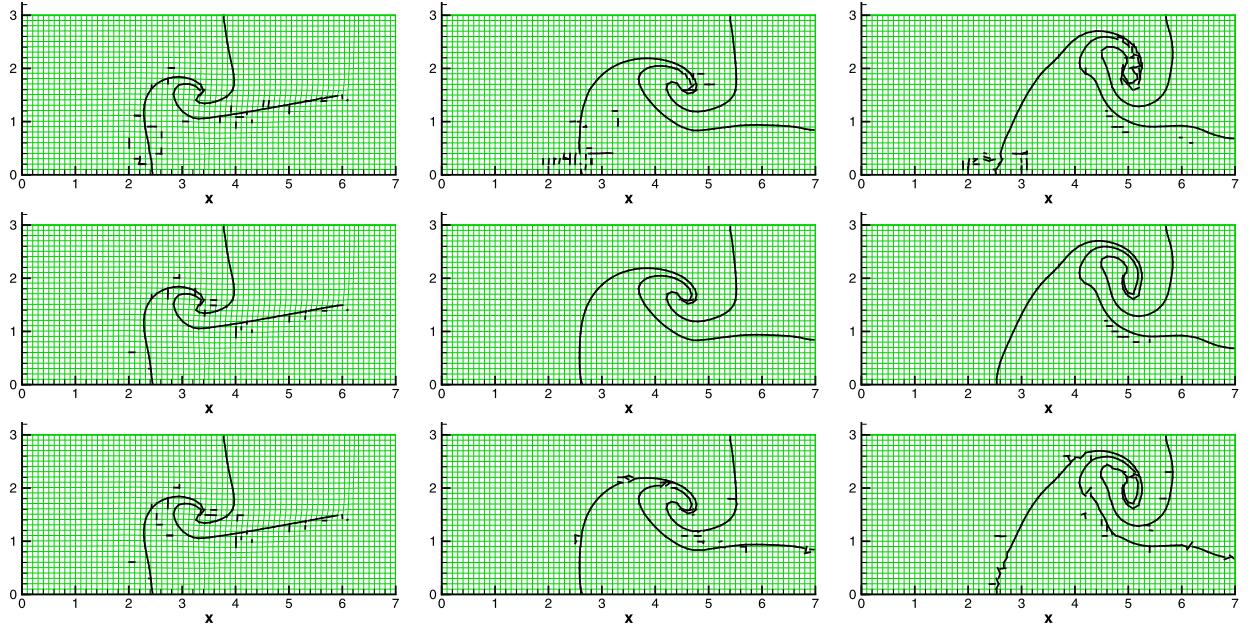


Fig. 23. Test 4.7. Triple point problem. Meshes and interfaces computed by method 1 (top) and method 2 with $h = 0.002$ (middle), and method 2 with $h = 0.1$ (bottom) at three time instances $t = 3, t = 5, t = 8$ (from left to right).

Acknowledgements

This research was sponsored by the National Nature Science Foundation of China (11571002, 11672047). The valuable comments of the anonymous referees are greatly appreciated.

References

- [1] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1) (1981) 201–225.
- [2] D.L. Youngs, Time-Dependent Multi-Material Flow with Large Fluid Distortion, *Numerical Methods in Fluid Dynamics*, Academic Press, 1982.
- [3] William J. Rider, Douglas B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (2) (1998) 112–152.
- [4] C.S. Wu, D.L. Young, H.C. Wu, Simulations of multidimensional interfacial flows by an improved volume-of-fluid method, *Int. J. Heat Mass Transf.* 60 (1) (2013) 739–755.
- [5] Wurigen Bo, John W. Grove, A volume of fluid method based ghost fluid method for compressible multi-fluid flows, *Comput. Fluids* 90 (90) (2014) 113–122.
- [6] C.W. Wang, T.G. Liu, B.C. Khoo, A real ghost fluid method for the simulation of multimedial compressible flow, *SIAM J. Sci. Comput.* 28 (1) (2006) 278–302.
- [7] Stanley Osher, James A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [8] Mark Sussman, Peter Smereka, Stanley Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1) (1994) 146–159.
- [9] James A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Natl. Acad. Sci. USA* 93 (4) (1996) 1591–1595.
- [10] Douglas Enright, Ronald Fedkiw, Joel Ferziger, Ian Mitchell, A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (1) (2002) 83–116.
- [11] Yucheng Zhao, Hamn Ching Chen, Cfd simulation of violent free surface flows by a coupled level-set and volume-of-fluid method, in: International Offshore and Polar Engineering Conference, 2013, pp. 2140–2151.
- [12] Vadim Dyadechko, Mikhail Shashkov, Moment-of-Fluid Interface Reconstruction, Technical Report, LA-UR-05-7571, Los Alamos National Laboratory, 2005.
- [13] Vadim Dyadechko, Mikhail Shashkov, Reconstruction of multi-material interfaces from moment data, *J. Comput. Phys.* 227 (11) (2008) 5361–5384.
- [14] Hyung Taek Ahn, Mikhail Shashkov, Adaptive moment-of-fluid method, *J. Comput. Phys.* 228 (8) (2009) 2792–2821.
- [15] Milan Kucharik, V. Garimella Rao, Samuel P. Schofield, Mikhail J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, *J. Comput. Phys.* 229 (7) (2010) 2432–2452.
- [16] Zupeng Jia, Jun Liu, Shudao 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 (1) (2013) 513–562.
- [17] Zupeng Jia, Yutao Sun, A 2D cell-centered MMALE method based on MOF interface reconstruction, *Chin. J. Comput. Phys.* 33 (5) (2016) 523–538.
- [18] Xiang Chen, Xiong Zhang, Zupeng Jia, A robust and efficient polyhedron subdivision and intersection algorithm for three-dimensional MMALE remapping, *J. Comput. Phys.* 338 (2017) 1–17.
- [19] Ryan N. Hill, Mikhail Shashkov, The symmetric moment-of-fluid interface reconstruction algorithm, *J. Comput. Phys.* 249 (3) (2013) 180–184.
- [20] Matthew Jemison, Mark Sussman, Mikhail Shashkov, Filament capturing with the multimaterial moment-of-fluid method, *J. Comput. Phys.* 285 (2015) 149–172.
- [21] H.K. Jinjal, Jyotirmay Banerjee, A Lagrangian–Eulerian advection scheme with moment-of-fluid interface reconstruction, *Numer. Heat Transf., Part B, Fundam.* 69 (6) (2016) 563–574.
- [22] Antoine Lemoine, S. Glockner, J. Breil, Moment-of-fluid analytic reconstruction on 2D Cartesian grids, *J. Comput. Phys.* 328 (2017) 131–139.
- [23] Xiang Chen, Xiong Zhang, An improved 2D MoF method by using high order derivatives, *J. Comput. Phys.* 349 (2017) 176–190.
- [24] William H. Press, Saul A. Teukolsky, Brian P. Flannery, Brian P. Flannery, *Numerical Recipes in FORTRAN (2nd ed.): The Art of Scientific Computing*, Cambridge University Press, 1992.
- [25] Haihe Li, Wansheng He, A method for solving more roots of nonlinear equation in large scope, *Chinese J. Gansu Sci.* 24 (3) (2012) 4–6.
- [26] Pierre Henri Maire, Remi Abgrall, Jerome Breil, Jean Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, *SIAM J. Sci. Comput.* 29 (4) (2007) 1781–1824.
- [27] Pierre Henri Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, *J. Comput. Phys.* 228 (7) (2009) 2391–2425.
- [28] Mikhail Shashkov, Closure models for multimaterial cells in arbitrary Lagrangian–Eulerian hydrocodes, *Int. J. Numer. Methods Fluids* 56 (8) (2008) 1497–1504.
- [29] Patrick M. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I-A framework for surface mesh optimization, *Int. J. Numer. Methods Eng.* 48 (3) (2000) 401–420.
- [30] Raphael Loubère, Pierre-Henri Maire, Mikhail Shashkov, Jerome Breil, Stephane Galera Reale, A reconnection-based arbitrary-Lagrangian–Eulerian method, *J. Comput. Phys.* 229 (12) (2010) 4724–4761.
- [31] L.I. Sedov, *Similarity and Dimensional Methods in Mechanics*, Academic Press, New York, 1959.
- [32] James J. Quirk, Smadar Karni, On the dynamics of a shock-bubble interaction, *J. Fluid Mech.* 318 (1) (1996) 129–163.