Journal of Computational Physics 384 (2019) 60–76

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A predicted-Newton's method for solving the interface positioning equation in the MoF method on general polyhedrons

Xiang Chen, Xiong Zhang *

School of Aerospace Engineering, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Article history:
Received 24 February 2018
Received in revised form 16 November 2018
Accepted 31 December 2018
Available online 22 February 2019

Keywords:
Interface reconstruction
Multi-material flow
Volume of fluid
Moment of fluid

ABSTRACT

Solving the interface positioning equation is the key procedure of the PLIC-VoF methods. Most of previous research only focused on the planar constant calculation and paid less attention to the relationship between the planar constant and the approximate interface orientation. The latter issue is important especially for the second order iteration based PLIC-VoF method, such as the MoF and LVIRA method. In these methods, the most accurate interface orientation is calculated through an iterative procedure, so the interface positioning equation has to be solved multiple times for the given volume fraction with different interface orientations. In this situation, if the incremental relation between the planar constant and the interface orientation is known, a predicted planar constant can be estimated. In this paper, we deduce the analytical partial derivatives of the planar constant with respect to the interface orientation and use them to predict the planar constant. A predicted-Newton's method is proposed to solve the interface positioning equation which takes the predicted planar constant as the initial guess. A great deal of numerical tests are also presented in this paper to verify the robustness of the new scheme. The efficiency of the proposed predicted-Newton's method is compared with the commonly used secant/bisection method by Ahn and Shashkov, and the numerical results indicate that the new method can reduce the iteration steps by 60% ∼ 66% in solving the interface positioning equation and reduce the CPU time by 32% ∼ 39% when implemented in the MoF method.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction and motivation

The PLIC (Piecewise Linear Interface Calculation) is one of the most famous interface reconstruction methods in the multi-material flow simulation. The basic idea of the PLIC method is to use a planar interface in a mixed cell to approximate the material interface which can be expressed as

$$\boldsymbol{n} \cdot \boldsymbol{x} + d = 0 \tag{1}$$

where $\boldsymbol{n} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$ is the normal of the approximate interface, (θ, φ) is the orientation of the interface and d is the planar constant. In the PLIC-VoF (Volume of Fluid) method, the approximate interface in Eq. (1) must fulfill the

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

https://doi.org/10.1016/j.jcp.2018.12.038
0021-9991/© 2019 Elsevier Inc. All rights reserved.

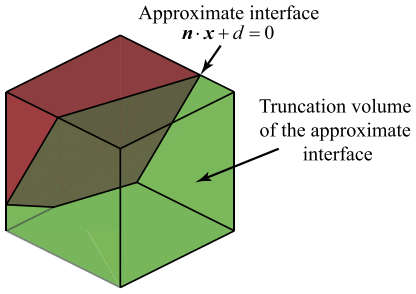

Fig. 1. The volume conservation requirement of the PLIC-VoF method. The truncation volume of the approximate interface (the green portion in the figure) must equal to the material volume in this mixed cell. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

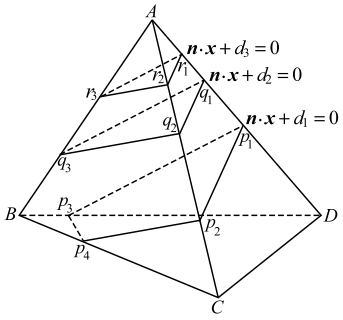

Fig. 2. The monotonicity of $V(d)$. The truncation volume $V(d)$ is a monotonous function of d, namely $V(d_3) > V(d_2) > V(d_1)$ in this figure.

volume conservation requirement, namely the truncation volume of the approximate interface must equal to the material's volume in the mixed cell as shown in Fig. 1.

The PLIC-VoF method involves two issues. The first one is the positioning process which calculate d with a given normal $\boldsymbol{n}$ and the second one is the orienting process which find the most accurate normal $\boldsymbol{n}^*$ to approximate the material interface best.

The first issue derives the interface positioning equation. As shown in Fig. 2, if the normal $\boldsymbol{n}$ of the interface is given and it points to the reference material, the truncation volume is a monotonic function of d with $V(d_{\min}) = 0,\ V(d_{\max}) = V_{\text{cell}}$. Moreover, V^{ref} must stay in $[0, V_{\text{cell}}]$, so there must be a unique d^* such that

$$V(d^*) = V^{\text{ref}} \tag{2}$$

where $d_{\min}$ and $d_{\max}$ are the minimal and maximal planar constants of $\boldsymbol{n}\cdot\boldsymbol{x}+d=0$ that pass through each cell vertex, $V(d^*)$ is the truncation volume of d^* and V^{ref} is the reference material volume in this mixed cell which is the input parameter of the PLIC-VoF method. Eq. (2) is called the interface positioning equation.

Eq. (2) can be solved directly only for regular polyhedrons and trivial combinations of $\boldsymbol{n}$ and volume fraction. However, for a general polyhedron, the explicit expression of $V(d)$ can not be obtained easily, so many methods have been developed, such as the analytical methods [1–6] and the iterative methods [7,8].

The basic ideas of the analytical methods are the same. They firstly bracket $[d_{\min}, d_{\max}]$ into intervals as

$$[d_{\min}, d_{\max}] = [d_0, d_1] \cup [d_1, d_2] \cup \cdots \cup [d_{N-1}, d_N] \tag{3}$$

where $d_0 = d_{\min}$, $d_{\max} = d_N$, d_i are the planar constants of the plane passing through each cell vertex and sorted ascendingly. Then, the target interval where d^* is located can be obtained by finding a k such that

$$V(d_{k-1}) \le V^{\text{ref}} \le V(d_k) \tag{4}$$

Finally, $V(d)$ in $[d_{k-1}, d_k]$ is either a linear, quadratic or cubic function with explicit expression and Eq. (2) can be solved analytically. The most time-consuming part of the analytical method is to calculate $V(d_k)$ and the polynomial parameters of $V(d)$ in $[d_{k-1}, d_k]$. Many schemes have been proposed to improve the efficiency of the analytical methods [1–3]. In recent year, the analytical methods for the non-convex polyhedron are also developed [5].

Another approach to solve Eq. (2) is the iterative method. In the iterative scheme, $V(d)$ will be calculated multiple times during the iteration until $\left|V(d) - V^{\text{ref}}\right|$ is smaller than the tolerance. The "Clipping and Capping" method [9,10] is usually used to calculate $V(d)$ which involves complex geometric analysis including segment-plane intersections, vertex reorganizations and special considerations for the non-convex polyhedrons [11]. The calculation of $V(d)$ is very expensive

so the major way to improve the efficiency of the iterative method is to reduce the number of iteration step. Comparing with the research about the analytical method, less attention is paid on the iterative method. The commonly used iterative method is the secant/bisection method proposed by Ahn and Shashkov [8]. The iterative method can be performed on both convex and non-convex polyhedrons directly, so it is suitable for the 3D MMALE (Multi-Material Arbitrary Lagrangian Eulerian) method where non-convex cells will naturally arise [11–13].

The second issue of the PLIC-VoF method is the normal calculation which classifies the PLIC-VoF method into different types. The early methods use the volume fraction gradient as the interface normal [7,14,15], but they are only first order accurate and even unable to precisely reconstruct a planar interface. To achieve second order accuracy, additional requirements must be applied. The LVIRA (Least square Volume Interface Reconstruction Algorithm) [16] estimates the normal by finding a linear interface, which will be extended outside the mix cell, to minimize the discrepancy of volume fraction in adjacent cells. The Swartz method [17,18] calculates the normal by finding a common linear interface for a pair of neighbor mix cells which rigorously satisfies their given volume fraction. More recently, Shashkov et al. proposed the MoF (Moment of Fluid) method [12,19] which makes use of the material centroid. The most accurate interface in the MoF method will minimize the discrepancy between the reference material centroid and the approximate centroid of the truncation volume. The orienting process of the MoF method does not need any information from neighbor cells, so it can be parallelized innately in the interface reconstruction phase. Moreover, a comparative study has been done by Kucharik et al. which indicates that the MoF method is more accurate than other second order PLIC-VoF methods [20]. Due to the various advantages of the MoF method, it has been developed and applied a lot in recent years [21–27].

The optimal normal of the second order PLIC-VoF method, such as the MoF and LVIRA method, is usually obtained by an iteration process, namely the trial interface orientation (θ, φ) will be changed until it converges to the optimal one. Therefore Eq. (2) must be solved multiple times with the same volume fraction and different interface orientations. However, the previous research about Eq. (2) only focus on the calculation of d with a given normal $\boldsymbol{n}$ and less attention is paid to the relation between d and $\boldsymbol{n}$ with a same V^{ref}. This issue is important because if $\partial d/\partial\theta$ and $\partial d/\partial\varphi$ are known, the planar constant at $(\theta+\Delta\theta, \varphi+\Delta\varphi)$ can be predicted as

$$d^{\text{predict}} = d(\theta,\varphi) + \frac{\partial d}{\partial\theta}\Delta\theta + \frac{\partial d}{\partial\varphi}\Delta\varphi \tag{5}$$

From our practical numerical experience, we know that roughly 90% of the $\Delta\theta$ and $\Delta\varphi$ are around 10^{-3} in the MoF method. So the predicted planar constant in Eq. (5) can be very close to the solution d^* and it can be regarded as a very good initial guess for the iterative method.

In this paper, the analytical expressions of $\partial d/\partial\theta$, $\partial d/\partial\varphi$ and $V'(d)$ are deduced. The cost of the derivative calculation is negligible because it only involves the known variables from the "Clipping and Capping" method. $V'(d)$ is used in a modified Newton's method which utilizes the monotonicity of $V(d)$ to ensure the convergence of the iteration. $\partial d/\partial\theta$ and $\partial d/\partial\varphi$ are used to predict the planar as Eq. (5) which will be regarded as the initial guess, so this new scheme is called the predicted-Newton's method. Eq. (5) combines the orienting process and the positioning process together, and to our best knowledge, it is a pioneering study in the PLIC-VoF method. The predicted-Newton's method can be used on an arbitrary polyhedron and it performs well even if a cell face is non-planar. A great deal of numerical tests are presented in this paper to verify the robustness of the new scheme. The efficiency of the predicted-Newton's method is also compared with the commonly used secant/bisection method by Ahn and Shashkov [8], and the numerical results indicate that new method can reduce the iteration steps by 60% ∼ 66% in solving the interface positioning equation and reduce the CPU time by 32% ∼ 39% when implemented in the MoF method.

The remaining part of the paper is organized as follows. In Section 2, we review the "centroid rotation rule" for an infinitesimal interface rotation and it is the foundation of the following derivation. Then, the partial derivatives of the planar constant with respect to θ and φ are deduced in Section 3 and the predicted-Newton's method is proposed in Section 4. Section 5 presents a large amount of numerical examples and Section 6 draws the conclusion.

2. Centroid rotation rule

In this section, we will introduce the "centroid rotation rule" which is derived from the volume conservation requirement. This rule was mentioned in our previous paper [26], and it will be rigorously proved for an arbitrary polyhedron in this section. The "centroid rotation rule" is the foundation for deducing the analytical partial derivatives of the planar constant.

For an arbitrary polyhedron, some faces may be non-planar. And in order to deal with the non-planarity, we introduce an auxiliary point at the face center and decompose the non-planar face into several triangles. Therefore, the polyhedron with non-planar faces can be represented by a polyhedron made up of triangle pieces.

Suppose $\boldsymbol{P}$ is the triangulated representation of an arbitrary polyhedron, and Γ_1, Γ_2 are two plane that fulfill the volume conservation requirement. Γ_1 and Γ_2 intersect $\boldsymbol{P}$ at p_i and q_i as shown in Fig. 3(a). For the sake of the clarity, the configuration of the entire $\boldsymbol{P}$ is not plotted and we only focus on the region near Γ_1 and Γ_2. The red lines represent the edges of $\boldsymbol{P}$. O_1O_2 is the intersection of Γ_1 and Γ_2 and $p_i, q_i\ i=1,2,\cdots,m$ locate on one side of O_1O_2 and $p_i, q_i\ i=m+1,m+2,\cdots,n$ locate on the other side of O_1O_2. The dihedral angle between these two interfaces is $\Delta\alpha$. Since $\Delta\alpha$ will tend to zero in the derivation, we can assume that $p_i, p_{i+1}, q_i, q_{i+1}$ locate on a same polyhedron face.

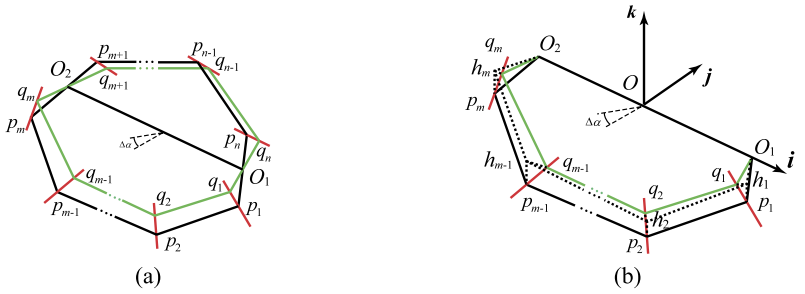

Fig. 3. The proof of the centroid rotation rule. Figure (a) shows two truncation faces Γ_1 and Γ_2 which fulfill the volume conservation requirement and figure (b) is the amplified view of figure (a).

Since Γ_1 and Γ_2 fulfill the volume conservation requirement, we have

$$V_{\Omega_1} = V_{\Omega_2} \tag{6}$$

where Ω_1 is the polyhedron $p_1 \cdots p_m q_1 \cdots q_m O_1 O_2$ and Ω_2 is the polyhedron $p_{m+1} \cdots p_n q_{m+1} \cdots q_n O_1 O_2$.

Fig. 3(b) is the amplified view of the polyhedron Ω_1 and a local coordinate system is constructed to calculate its volume. The origin of the local coordinate system O lies on the intersecting line O_1O_2 with axis $\boldsymbol{i}$ parallel to $\overrightarrow{O_2O_1}$, axis $\boldsymbol{k}$ parallel to the normal of interface $p_1p_2p_3$ and axis $\boldsymbol{j} = \boldsymbol{k} \times \boldsymbol{i}$. To calculate the volume, several auxiliary lines are created passing through the vertices p_i which are perpendicular to Γ_1 and intersect with Γ_2 at h_i as shown in Fig. 3(b). Since every face of $\boldsymbol{P}$ is triangle, $p_i, p_{i+1}, q_i, q_{i+1}$ must be on a same plane. Moreover, on the polyhedron Ω_1, segment p_ip_{i+1} and q_iq_{i+1} $(i = 1, 2, \cdots m-1)$ do not intersect, so h_i, h_{i+1} must locate on a same side of p_ip_{i+1} and q_iq_{i+1}.

The volume of Ω_1 is

$$V_{\Omega_1} = V_{C_1} - V_1^{\text{extra}} \tag{7}$$

where C_1 is the polyhedron $p_1p_2 \cdots p_m h_1 h_2 \cdots h_m O_1 O_2$ and

$$V_1^{\text{extra}} = V_{p_1q_1h_1O_1} + V_{p_mq_mh_mO_2} + \sum_{i=1}^{m-1} V_{p_ip_{i+1}q_iq_{i+1}h_ih_{i+1}} \tag{8}$$

C_1 is a column in the local coordinate system whose volume can be rewritten as

$$\begin{aligned} V_{C_1} &= \int_{C_1} \mathrm{d}x\mathrm{d}y\mathrm{d}z = \int_{C_1} \nabla \cdot (0,0,z)\,\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &= \sum_k \int_{\Pi_k} (0,0,z) \cdot \boldsymbol{n}_k \mathrm{d}S = \sum_k n_{kz} \int_{\Pi_k} z\mathrm{d}S \\ &= \cos(\Delta\alpha) \int_{h_1h_2\cdots h_mO_1O_2} z\mathrm{d}S \\ &= \tan(\Delta\alpha) \int_{p_1p_2\cdots p_mO_1O_2} y\mathrm{d}x\mathrm{d}y \end{aligned} \tag{9}$$

where Π_k is the k^{th} faces of C_1 and n_{kz} is the z-component of the normal to Π_k. The third line is due to the fact that at each face Π_k, either n_{kz} or z equals to zero except the face $h_1h_2 \cdots h_mO_1O_2$. The last equality is due to $z = \tan(\Delta\alpha)\, y$ in the local coordinate system.

Similarly, the volume of polyhedron Ω_2 can be rewritten as

$$V_{\Omega_2} = -\tan(\Delta\alpha) \int_{p_{m+1}p_{m+2}\cdots p_nO_1O_2} y\mathrm{d}x\mathrm{d}y + V_2^{\text{extra}} \tag{10}$$

where

$$V_2^{\text{extra}} = V_{p_{m+1}q_{m+1}h_{m+1}O_1} + V_{p_nq_nh_nO_2} + \sum_{i=m+1}^{n-1} V_{p_ip_{i+1}q_iq_{i+1}h_ih_{i+1}} \tag{11}$$

Substituting Eq. (7), Eq. (9) and Eq. (10) into Eq. (6), we have

$$\tan(\Delta\alpha)\int_{p_1p_2\cdots p_mO_1O_2} y\mathrm{d}x\mathrm{d}y+\tan(\Delta\alpha)\int_{p_{m+1}p_{m+2}\cdots p_nO_1O_2} y\mathrm{d}x\mathrm{d}y=V_1^{\text{extra}}+V_2^{\text{extra}} \tag{12}$$

namely

$$\int_{\Gamma_1} y\mathrm{d}x\mathrm{d}y=\frac{V_1^{\text{extra}}+V_2^{\text{extra}}}{\tan(\Delta\alpha)} \tag{13}$$

We have proved that the magnitude of each small volume in V_1^{extra} and V_2^{extra} is $O\left(\Delta\alpha^2\right)$ in our previous paper [26]. Therefore as $\Delta\alpha\to 0$, we have

$$\lim_{\Delta\alpha\to 0}\int_{\Gamma_1} y\mathrm{d}x\mathrm{d}y=0 \tag{14}$$

Eq. (14) indicates that in order to fulfill the volume conservation requirement, the axis of an infinitesimal rotation must pass through the centroid of the truncated face. This conclusion is called the "centroid rotation rule" and it can be expressed in another way as

$$\lim_{\Delta\alpha\to 0} s=0 \tag{15}$$

where s is the distance from the centroid of the truncated face to the rotation axis.

The derivation above is invariant with respect to the shape of the truncation face. For a non-convex polyhedron, Γ_1 can be a non-convex polygon like Fig. 8(b) or be composed of several isolated small polygons like Fig. 8(c), and the derivation is also applicable.

It should be mentioned that special attentions must be paid if the truncation face is coincident with one polyhedron face. In this situation, the definition of the truncation face is ambiguous and its centroid depends on the rotation direction. There will be a centroid jump in this case and the specific analysis is presented in Appendix A.

3. The partial derivatives of the planar constant

With a given material volume V^{ref}, the planar constant $d(\theta,\varphi)$ is a function of the interface orientation (θ,φ). According to the definition of the partial derivatives, we have

$$\frac{\partial d}{\partial\theta}=\lim_{\Delta\theta\to 0}\frac{d(\theta+\Delta\theta,\varphi)-d(\theta,\varphi)}{\Delta\theta} \tag{16}$$

$$\frac{\partial d}{\partial\varphi}=\lim_{\Delta\varphi\to 0}\frac{d(\theta,\varphi+\Delta\varphi)-d(\theta,\varphi)}{\Delta\varphi} \tag{17}$$

where $\Delta\theta$ and $\Delta\varphi$ denote infinitesimal rotations of the interface.

Let us take the partial derivative with respect to θ as an example. Fig. 4 shows the two related interfaces Γ_1 $(p_1p_2p_3)$ and Γ_2 $(q_1q_2q_3)$, which are defined by

$$\Gamma_1:\boldsymbol{n}(\theta,\varphi)\cdot\boldsymbol{x}+d(\theta,\varphi)=0$$

and

$$\Gamma_2:\boldsymbol{n}(\theta+\Delta\theta,\varphi)\cdot\boldsymbol{x}+d(\theta+\Delta\theta,\varphi)=0$$

respectively. In Fig. 4, C is the centroid of the truncated face Γ_1, l is the distance from C to the interface Γ_2, and s is the distance from C to the rotation axis O_1O_2. We will firstly assume a finite $\Delta\theta$ and then let it approaches to zero to calculate the partial derivative.

The centroid C lies on the interface Γ_1 so that

$$\boldsymbol{n}(\theta,\varphi)\cdot\boldsymbol{x}_C+d(\theta,\varphi)=0 \tag{18}$$

On the contrary, $\delta=\boldsymbol{n}(\theta+\Delta\theta,\varphi)\cdot\boldsymbol{x}_C+d(\theta+\Delta\theta,\varphi)\neq 0$ because C does not lie on the interface Γ_2. However, its absolute value gives the distance l from C to Γ_2, namely

$$\left|\boldsymbol{n}(\theta+\Delta\theta,\varphi)\cdot\boldsymbol{x}_C+d(\theta+\Delta\theta,\varphi)\right|=l=s\sin(\Delta\theta) \tag{19}$$

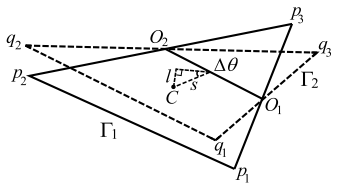

Fig. 4. The finite size rotation of the interface.

Supposing δ is positive, we have

$$\boldsymbol{n}(\theta+\Delta\theta,\varphi)\cdot\boldsymbol{x}_C+d(\theta+\Delta\theta,\varphi)=s\sin(\Delta\theta) \tag{20}$$

namely

$$d(\theta+\Delta\theta,\varphi)=s\sin(\Delta\theta)-\boldsymbol{n}(\theta+\Delta\theta,\varphi)\cdot\boldsymbol{x}_C \tag{21}$$

The case with a negative δ can be analyzed in the same way.

Invoking Eq. (21) and Eq. (18), we have

$$\frac{d(\theta+\Delta\theta,\varphi)-d(\theta,\varphi)}{\Delta\theta}=-\frac{\boldsymbol{n}(\theta+\Delta\theta,\varphi)-\boldsymbol{n}(\theta,\varphi)}{\Delta\theta}\cdot\boldsymbol{x}_C+s\frac{\sin(\Delta\theta)}{\Delta\theta} \tag{22}$$

Substituting Eq. (22) into Eq. (16) gives

$$\frac{\partial d}{\partial\theta}=-\boldsymbol{x}_C\cdot\frac{\partial\boldsymbol{n}}{\partial\theta}+\lim_{\Delta\theta\to0}s\lim_{\Delta\theta\to0}\frac{\sin(\Delta\theta)}{\Delta\theta}=-\boldsymbol{x}_C\cdot\frac{\partial\boldsymbol{n}}{\partial\theta}+\lim_{\Delta\theta\to0}s \tag{23}$$

According to the "centroid rotation rule" and Eq. (15), s approaches to zero as $\Delta\theta\to0$ and $\boldsymbol{n}$ can be rewritten as the unit normal vector in spherical coordinates as $\boldsymbol{n}=(\sin\theta\cos\varphi,\sin\theta\sin\varphi,\cos\theta)$, therefore

$$\frac{\partial d}{\partial\theta}=-\boldsymbol{x}_C\cdot\frac{\partial\boldsymbol{n}}{\partial\theta}=-\boldsymbol{x}_C\cdot(\cos\theta\cos\varphi,\cos\theta\sin\varphi,-\sin\theta) \tag{24}$$

Similarly, the partial derivative with respect to φ can be obtained as

$$\frac{\partial d}{\partial\varphi}=-\boldsymbol{x}_C\cdot\frac{\partial\boldsymbol{n}}{\partial\varphi}=-\boldsymbol{x}_C\cdot(-\sin\theta\sin\varphi,\sin\theta\cos\varphi,0) \tag{25}$$

Note that the "Clipping and Capping" method will calculate vertices of the truncated interface, so $\boldsymbol{x}_C$ can be easily obtained with very little extra expense.

The derivation of Eq. (24) and Eq. (25) are invariant with respect to the shape of the polyhedron and the truncation face, so Eq. (24) and Eq. (25) are suitable for arbitrary polyhedrons as long as no centroid jump occurs. However, it should be mentioned again that the centroid jump occurs if the truncation face is coincident with one polyhedron face, and it will lead to the discontinuity of $\frac{\partial d}{\partial\theta}$ and $\frac{\partial d}{\partial\varphi}$. The specific analysis and the influence on the iterative method are presented in Appendix A.

In summary, the partial derivative of the planar constant with respect to θ and φ can be calculated by the following steps:

1. Solve the interface positioning equation Eq. (2) and calculate the vertices of the truncated face;
2. Calculate the centroid of the truncated face $\boldsymbol{x}_C$;
3. Calculate the partial derivatives using Eq. (24) and Eq. (25).

4. The predicted Newton's iterative method

The Newton's method is a good choice for solving Eq. (2) but it also brings several problems in practice. Firstly, the n^{th} iteration step of the Newton's method is

$$d^{n+1}=d^n-\frac{V(d^n)}{V'(d^n)} \tag{26}$$

which involves the derivative $V'(d^n)$ and it must be obtained efficiently. Secondly, the potential divergence of the Newton's method must be avoided. Finally, an appropriate initial guess should be used because it will influence the efficiency of the Newton's method significantly.

In this section, we solved the three problems mentioned above and propose the predicted-Newton's method.

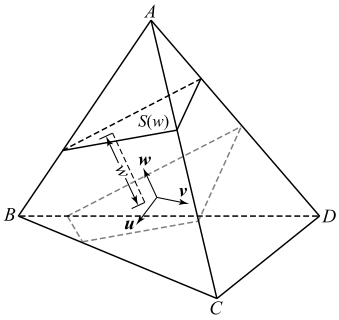

Fig. 5. The coordinate transformation to calculate the derivative of $V(d)$.

4.1. The analytical expression of $V'(d)$

The different value of the planar constant d in Eq. (1) means the plane moving along the normal $\boldsymbol{n}$, therefore a local coordinate system can be constructed as shown in Fig. 5. The $\boldsymbol{w}$ axis is parallel to $\boldsymbol{n}$, and the other two axes are arbitrary. In the local coordinate system, the truncation volume $V(d)$ can be calculated as

$$V(d) = \int_{d_{\min}}^{d} S(w)\,\mathrm{d}w \tag{27}$$

where $S(w)$ is the area of the truncated face at w. According to the definition of the derivative, the derivative can be rewritten as

$$\begin{aligned} V'(d) &= \lim_{\Delta d \to 0} \frac{V(d+\Delta d) - V(d)}{\Delta d} \\ &= \lim_{\Delta d \to 0} \frac{\int_{d_{\min}}^{d+\Delta d} S(w)\,\mathrm{d}w - \int_{d_{\min}}^{d} S(w)\,\mathrm{d}w}{\Delta d} \\ &= \lim_{\Delta d \to 0} \frac{\int_{d}^{d+\Delta d} S(w)\,\mathrm{d}w}{\Delta d} \end{aligned} \tag{28}$$

According to the mean value theorem of integral, Eq. (28) can be rewritten as

$$V'(d) = \lim_{\Delta d \to 0} \frac{S(\xi)\,\Delta d}{\Delta d} = \lim_{\Delta d \to 0} S(\xi) \tag{29}$$

where $\xi \in [d, d+\Delta d]$.

It is obvious that $S(\xi)$ converges to $S(d)$ as $\Delta d \to 0$, so we have

$$V'(d) = S(d) \tag{30}$$

The vertices of the truncated face are obtained when calculating $V\left(d^n\right)$, so $V'\left(d^n\right) = S\left(d^n\right)$ can be obtained directly with negligible extra cost.

The above derivation is also valid for the non-convex polyhedron when the truncated face has multiple components. The verification of the Newton's method using Eq. (30) is presented in Section 5.3.

4.2. Convergence consideration

Note that $V(d)$ is a piecewise linear, quadratic or cubic function with C1 discontinuity [1,3], so the traditional Newton's method may be divergent. To this end, we modify the traditional Newton's method by utilizing the monotonicity of $V(d)$ to ensure the convergence. In practice, the interval where the solution is located will be tracked during the iteration. And once the d^{n+1} from Eq. (26) is out of the interval, the cubic Hermite interpolation will be applied to calculate d^{n+1}. The process of the modified Newton's method is shown in Algorithm 1.

In the improved scheme, the length of the interval will decrease at every step which guarantees the convergence of the iteration. In practice, the cubic Hermite interpolation modification appears seldom, so its influence on the efficiency can be neglected.

Algorithm 1: The absolutely converged Newton's method.

Input V^{ref}, $\boldsymbol{n}$, $d_{\min}$, $d_{\max}$, d^{initial} with $V(d_{\min})=0$ and $V(d_{\max})=V_{\text{cell}}$
Output d
$d = d^{\text{initial}}$
Calculate $V(d)$ and $S(d)$ by the Clipping and Capping method
While $\left|V(d)-V^{\text{ref}}\right| > \varepsilon$
  if $V(d) > V^{\text{ref}}$
    $d_{\max} = d\ V_{\max} = V(d)\ S_{\max} = S(d)$
  else
    $d_{\min} = d\ V_{\min} = V(d)\ S_{\min} = S(d)$
  end
  $d = d - \dfrac{V(d)}{S(d)}$
  if $d \notin [d_{\min}, d_{\max}]$
    Calculate the cubic function $H(x)$ from Hermite interpolation in $[d_{\min}, d_{\max}]$
    $d =$ the root of $H(x)=0$
  end
  Calculate $V(d)$ and $S(d)$ by the Clipping and Capping method
end
return d

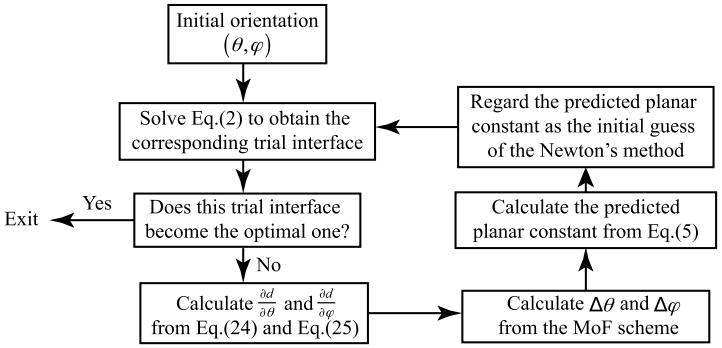

Fig. 6. The flow chart of the MoF method using the predicted-Newton's method to solve the interface positioning equation.

4.3. The predicted Newton's method

The initial guess is very important for the Newton's method and Eq. (5) is a good selection whenever the interface orientation (θ, φ) is changed. Since the initial guess from the Eq. (5), Eq. (24) and Eq. (25) is the second order prediction of the planar constant, the new scheme is called the predicted-Newton's method. The predicted-Newton's method can be incorporated with the MoF method and the flow chat is shown in Fig. 6.

It should be mentioned that if Eq. (2) is solved for the first time, thus no information could be used to predict the planar constant, the cubic Hermite interpolation is used to calculate the initial guess. The cubic polynomial $H(d)$ can be constructed by

$$\begin{aligned} H(d_{\min}) &= -V^{\text{ref}} \\ H(d_{\max}) &= V^{\text{cell}} - V^{\text{ref}} \\ H'(d_{\min}) &= S(d_{\min}) = 0 \\ H'(d_{\max}) &= S(d_{\max}) = 0 \end{aligned} \tag{31}$$

and the root of $H(d) = 0$ can be regarded as the initial guess of the Newton's method.

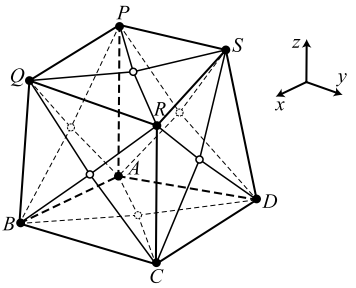

Fig. 7. A typical hexahedron. The solid dots represent the vertexes, whereas the hollow dots represent the face centers which are used to decompose each face into four triangles.

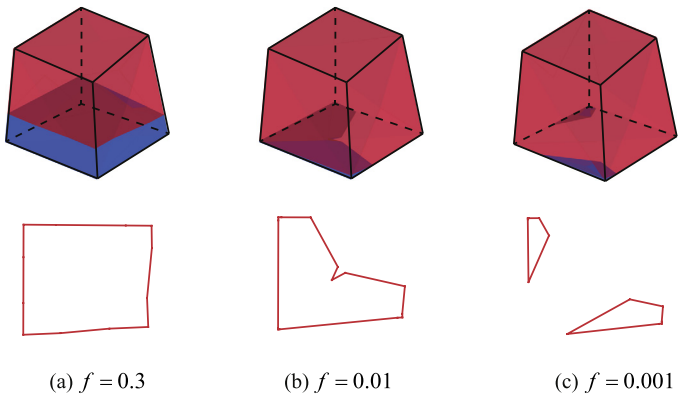

Fig. 8. The different types of a non-convex polyhedron subdivision. The volume fraction in figure (a), (b) and (c) is 0.3, 0.01 and 0.001 respectively.

5. Numerical result

A large number of numerical experiments are presented in this section. In this section, firstly, the analytical derivatives in Eq. (24) and Eq. (25) are verified on a polyhedron with non-planar faces. Then the efficiency and the robustness will be compared between the predicted-Newton's method and the secant/bisection method used by Ahn and Shashkov [8]. In their method, the secant method is stabilized with the bisection iteration to guarantee the convergence. Finally, the predicted-Newton's method is implemented in the MoF method to show the efficiency improvement in practice. The CPU used in the numerical study is Intel Core i7-3770 3.4 GHz and the compiler is Visual Studio 2012 with O2 optimization.

5.1. A typical polyhedron

In this subsection, a typical polyhedron as shown in Fig. 7 is studied to verify the partial derivatives and to examine the convergence rate and the efficiency improvement. The vertexes of the polyhedron are $A\,(0,0,0)$, $B\,(1.1,0,0)$, $C\,(1,1.2,0)$, $D\,(0,1.3,0.2)$, $P\,(0,0,1.2)$, $Q\,(1,0,1.1)$, $R\,(1,1.2,1.1)$ and $S\,(0,1,1.3)$, which do not lie on the same plane in every face. Each face is decomposed into four triangles by the dashed lines as Ahn et al. (2007) [12], so the cell is represented by a 24-face polyhedron with 14 vertexes. This type of polyhedron is very common in the MMALE (Multi-Material Arbitrary Lagrangian Eulerian) method. Moreover, different volume fraction f in this cell may lead to different type of subdivisions.

Suppose the orientation of the interface is $(\theta, \varphi) = (-0.01\pi, 0.7\pi)$, which is very close to the orientation of face $ABCD$. Fig. 8 shows the polyhedron subdivisions at $f = 0.3$, $f = 0.01$ and $f = 0.001$. The first row shows the 3D view of the subdivisions while the second row shows the shape of the truncation faces. In case (a), the truncation face does not pass through the non-planar face, so it is similar to the situations for a convex polyhedron subdivision. As the volume fraction decreases, the truncation face may pass through the non-planar face like cases (b) and (c). These weird subdivision types may appear if the volume faction is close to 0 or 1 and the orientation of the interface is close to the orientation of the non-planar face.

In summary, Fig. 8 contains three typical polyhedron subdivision situations. In the following subsections, we will verify the expressions in Eq. (24) and Eq. (25), study the convergence rate of our predicted-Newton's method and compare the efficiency with the secant/bisection method on this cell with different orientations and volume fractions. We believe that these tests could cover various polyhedron subdivision types and demonstrate the applicability of our new method for arbitrary polyhedrons.

Table 1
The discrepancy between the numerical and analytical partial derivatives for the three subdivision types in Fig. 8.

	Type (a)		Type (b)		Type (c)	
$\Delta\theta$ or $\Delta\varphi$	$\lvert\Delta d_\theta\rvert$	$\lvert\Delta d_\varphi\rvert$	$\lvert\Delta d_\theta\rvert$	$\lvert\Delta d_\varphi\rvert$	$\lvert\Delta d_\theta\rvert$	$\lvert\Delta d_\varphi\rvert$
10^{-3}	1.41×10^{-4}	2.90×10^{-6}	6.96×10^{-4}	7.68×10^{-7}	3.55×10^{-3}	1.08×10^{-5}
10^{-4}	1.41×10^{-5}	2.90×10^{-7}	6.96×10^{-5}	7.64×10^{-8}	3.51×10^{-4}	1.08×10^{-6}
10^{-5}	1.41×10^{-6}	2.90×10^{-8}	6.96×10^{-6}	7.63×10^{-9}	3.51×10^{-5}	1.08×10^{-7}
10^{-6}	1.41×10^{-7}	2.86×10^{-9}	6.96×10^{-7}	7.67×10^{-10}	3.51×10^{-6}	1.08×10^{-8}
10^{-7}	1.32×10^{-8}	1.42×10^{-9}	6.93×10^{-8}	9.84×10^{-12}	3.51×10^{-7}	1.10×10^{-9}
10^{-8}	4.34×10^{-9}	4.19×10^{-9}	5.67×10^{-9}	5.96×10^{-11}	3.66×10^{-8}	2.52×10^{-9}
10^{-9}	1.23×10^{-8}	3.75×10^{-8}	7.06×10^{-9}	2.42×10^{-8}	1.19×10^{-8}	1.81×10^{-9}

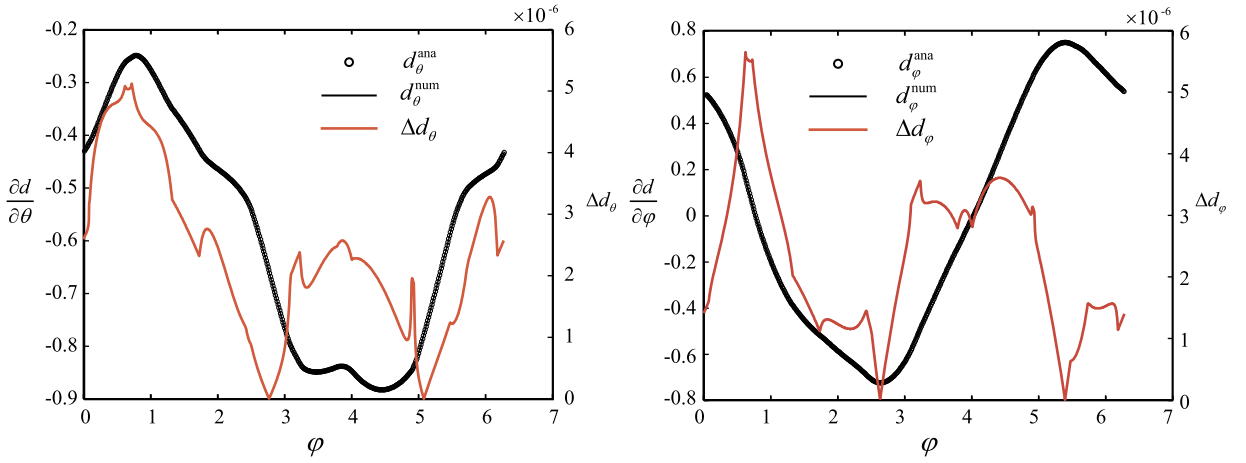

Fig. 9. Comparison of the numerical and analytical derivatives at a line.

5.2. Partial derivative verification

In this subsection, the expressions in Eq. (24) and Eq. (25) are verified on the polyhedron shown in Fig. 7. Denote the derivatives in Eq. (24) and Eq. (25) as d_θ^{ana} and d_φ^{ana} and finite difference partial derivatives defined as

$$d_\theta^{\text{num}} = \frac{d(\theta+\Delta\theta,\varphi) - d(\theta,\varphi)}{\Delta\theta} \tag{32}$$

$$d_\varphi^{\text{num}} = \frac{d(\theta,\varphi+\Delta\varphi) - d(\theta,\varphi)}{\Delta\varphi} \tag{33}$$

The results of d_θ^{num} and d_φ^{num} depend on the increments $\Delta\theta$ and $\Delta\varphi$ which will be more accurate as the increments decrease. On the contrary, the analytical derivatives d_θ^{ana} and d_φ^{ana} are independent of $\Delta\theta$ and $\Delta\varphi$. Therefore, we will examine the discrepancy between the analytical derivatives and the numerical derivatives by gradually decreasing $\Delta\theta$ and $\Delta\varphi$ to see whether the numerical derivatives converge to the analytical derivatives.

The results of the three subdivision types in Fig. 8 are shown in Table 1 where $\Delta d_\theta = d_\theta^{\text{num}} - d_\theta^{\text{ana}}$ and $\Delta d_\varphi = d_\varphi^{\text{num}} - d_\varphi^{\text{ana}}$. It can be seen that if the $\Delta\theta$ or $\Delta\varphi$ decreases from 10^{-3} to 10^{-6}, the numerical partial derivatives converge to the analytical partial derivatives at first order convergence rate. Therefore, the expressions in Eq. (24) and Eq. (25) are the exact partial derivatives for every polyhedron subdivision types. However, if the $\Delta\theta$ or $\Delta\varphi$ continues to decrease, the convergence rate will be gradually broken due to the round-off error in calculating $d(\theta+\Delta\theta,\varphi) - d(\theta,\varphi)$ and $d(\theta,\varphi+\Delta\varphi) - d(\theta,\varphi)$.

Then, in order to verify the analytical partial derivatives in a large scope, we calculate the numerical and analytical derivatives at the line $\theta = 0.37\pi$ and $\varphi = 0 \sim 2\pi$ with volume fraction $f = 0.3$. The increments $\Delta\theta$ and $\Delta\varphi$ are equal to 10^{-5} in the numerical derivatives. According to the Taylor's expansion, $\Delta d_\theta \approx \frac{1}{2}\frac{\partial^2 d}{\partial\theta^2}\Delta\theta$ and $\Delta d_\varphi \approx \frac{1}{2}\frac{\partial^2 d}{\partial\varphi^2}\Delta\varphi$, so we also plot the curves of Δd_θ and Δd_φ to show the properties of the second order partial derivatives. The results are shown in Fig. 9. The analytical partial derivatives and the numerical derivatives are almost exactly coincident which verifies the expression in Eq. (24) and Eq. (25). Moreover, we find that the Δd_θ and Δd_φ are not smooth functions and the nonsmoothness may occur when the truncated face pass through the vertex of the polyhedron.

In summary, these numerical studies verify the expressions in Eq. (24) and Eq. (25) as the accurate partial derivatives of the planar constant for both convex and non-convex polyhedrons.

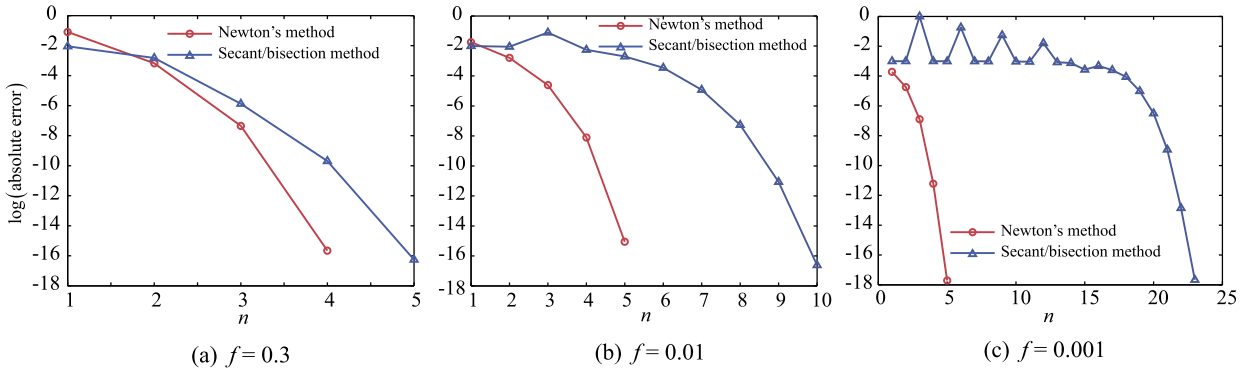

Fig. 10. The convergence rate comparison of the Newton's method without prediction and the secant/bisection method. The volume fractions of these three tests are 0.3, 0.01 and 0.001, respectively. The tolerance of the iteration for both Newton's method and the secant/bisection method is 10^{-15}.

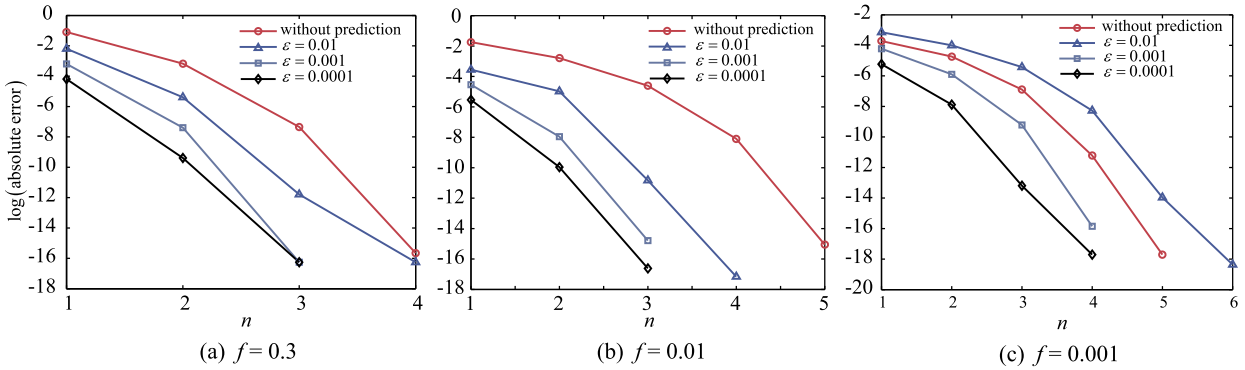

Fig. 11. The convergence rates of the predicted-Newton's method. The volume fractions of these three tests are 0.3, 0.01 and 0.001, respectively. For each test, three different $\Delta\theta$ and $\Delta\varphi$ are used to calculate the predicted planar constant. The tolerance of the iteration for these tests is also 10^{-15}.

5.3. The convergence rate comparison for solving the interface positioning equation

In this subsection, we will compare the convergence rate of the predicted-Newton's method and the secant/bisection method [8] for solving the interface positioning equation. The three subdivision types shown in Fig. 8 are used in this subsection again. The numerical tests are divided into two steps. The first step is solving the interface positioning equation on the polyhedron shown in Fig. 7 with $(\theta, \varphi) = (-0.01\pi, 0.7\pi)$ and $f = 0.3, 0.01, 0.001$. Because this is the first time to solve the interface positioning equation, the planar constant can not be predicted so the initial guess of the Newton's method is calculated from the cubic Hermite interpolation. The tolerance of the iteration is chosen as 10^{-15} for both Newton's method and the secant/bisection method.

Fig. 10 shows the absolute error $\log\left(\left|V\left(d^n\right) - V^*\right|\right)$ at the n^{th} step. Fig. 10(a) corresponds to the case (a) in Fig. 8. In this situation, the performance of the Newton's method and the secant/bisection method is comparable and the Newton's method is a little faster. Fig. 10(b) corresponds to the case (b) in Fig. 8. In this situation, the oscillation appears in the secant/bisection method at the beginning 4 steps but the Newton's method performs very well. The secant/bisection method takes 10 steps to reach the tolerance while the Newton's method only needs 5 steps. Fig. 10(c) corresponds to the case (c) in Fig. 8. In this situation, a much more severe oscillation appears in the secant/bisection method while the Newton's method is also very robust.

The second step of this numerical study will show the improvement from the predicted planar constant. The orientation (θ, φ) changes to $(\theta + \Delta\theta, \varphi + \Delta\varphi)$ where $\Delta\theta = \Delta\varphi = \pi\varepsilon$ and the volume fraction f also equals to 0.3, 0.01 and 0.001, respectively. The planar constants at (θ, φ) are known from the previous numerical study, so the initial guess can be predicted from Eq. (5).

Fig. 11 shows the absolute error at the n^{th} step by the predicted-Newton's method for different ε. The numerical results demonstrate that the predicted planar constant is more accurate as ε decreasing and iteration step could also be reduced by using the predicted planar constant as the initial guess.

To compare the different methods in a large scope, 1000 different volume fractions evenly in $[0, 1]$ are tested. In this test, Eq. (2) is solved 1000 times with different V^{ref} by the secant/bisection method, Newton's method without prediction and the predicted Newton's method with different ε. Table 2 records the maximal, minimal and the average iterations steps of the different methods. The results indicate that the Newton's method is more efficient and robust that the secant/bisection method and the prediction of the planar constant is effective for the Newton's method.

Table 2
The maximal, minimal and the average iterations steps of the different methods. Labels SM, NM and PNM(ε) represent the secant/bisection method, Newton's method and the predicted-Newton's method with different ε.

	SM	NM	PNM(0.01)	PNM(0.001)	PNM(0.0001)
$N_{\min}$	5	3	3	3	2
$N_{\max}$	24	6	6	4	3
N_{average}	6.917	4.086	3.165	3.004	2.077

In summary, the numerical studies in this subsection demonstrate that the predicted-Newton's method proposed in this paper is effective. The convergence rate and the robustness of the Newton's method is higher than the secant/bisection method. Moreover, an appropriate predicted planar constant will further improve the Newton's method in two aspects. It not only improves the efficiency by reducing the iteration steps but also improves the robustness because the predicted planar constant makes the Newton's method more easy to converge and avoid the oscillations appearing in the secant/bisection method.

5.4. Parameterized tests about continuous rotation of an interface

In this example, the interface will be rotated continuously with a constant volume fraction f, namely the interface orientation (θ, φ) changes from $(0, 0)$ to $(\pi, 2\pi)$ uniformly in N steps. Whenever the orientation is changed, the interface positioning equation will be resolved, so there is N different interface positioning equations to be solved for each f. This process is very similar to the iteration based second order PLIC-VoF method. The numerical study is also performed on the polyhedron shown in Fig. 7.

We use the predicted Newton's method and the secant/bisection method [8] to solve these series of interface positioning equations and compare their efficiency. The tolerance of the iteration is chosen as $\varepsilon = 10^{-14}$ for both Newton's method and the secant/bisection method. For a given volume fraction f, N interface positioning equations will be solved. And we use the average number of iteration step to measure the efficiency as

$$\psi(f) = \frac{1}{N}\sum_{i=1}^{N} q_i(f) \tag{34}$$

where $q_i(f)$ is the number of iteration steps required for solving the i^{th} interface positioning equation with volume fraction f.

The interface orientation increments at each rotation are $\Delta\theta = \pi/N$ and $\Delta\varphi = 2\pi/N$, so the predicted planar constant will be more accurate as N increase. Therefore, $\psi(f)$ will decrease when using the predicted Newton's method as N increase.

Two numerical experiments are performed in this section. In the first numerical experiment, 1000 different volume fractions f, which distribute evenly in $[0, 1]$, are tested. This test aims to compare the efficiency of different methods and show the contribution from the planar constant prediction. In the second numerical experiment, f will be very closed to 0 or 1, namely $f = \left[10^{-3}, 10^{-4}, \cdots, 10^{-13}\right]$ and $f = \left[1 - 10^{-3}, 1 - 10^{-4}, \cdots, 1 - 10^{-13}\right]$. Therefore, $V'(d)$ will also be very closed to 0 which brings difficulty to both secant/bisection method and the Newton's method. The secant/bisection method requires many iterations and may fail to converge if $f \to 0^+$ or $f \to 1^-$, so the bisection method is recommended [8]. In this section, we will study the behavior of the proposed predicted-Newton's method in this situation.

5.4.1. Numerical results of the Test 1

Fig. 12 shows the curves of $\psi(f)$ for the secant/bisection method, the Newton's method without prediction, the predicted-Newton's method with different N. It can be seen that the iteration efficiency of the Newton's method is significantly higher than the secant/bisection method. Moreover, an appropriate predicted planar can further improve the efficiency of the Newton's method. As expected, $\psi(f)$ decreases as N increasing in the predicted Newton's method which indicates that the predicted planar constant by Eq. (5) is effective.

Table 3 gives the average value of $\psi(f)$ for the secant/bisection method, Newton's method without prediction, and the predicted-Newton's method with different N. The results also indicate that the Newton's method performs better than the secant/bisection method and the efficiency can be further improved with smaller $\Delta\theta$ and $\Delta\varphi$. From our practical numerical experience we know that roughly 90% of the $\Delta\theta$ and $\Delta\varphi$ are around 10^{-3} in the MoF method, therefore the predicted Newton's method can reduce the number of iteration step by about 60% in practice.

5.4.2. Numerical results of Test 2

Fig. 13 shows the value of $\psi(f)$ at $f \to 0^+$ and $f \to 1^-$ for the Newton's method without prediction and the predicted-Newton's method with different N. Firstly, all the methods are able to find d^* which verify the robustness of the root finding algorithm in Section 4. Secondly, as $f \to 0^+$ and $f \to 1^-$, $V'(d^*)$ is very close to 0. So the accuracy of the initial guess

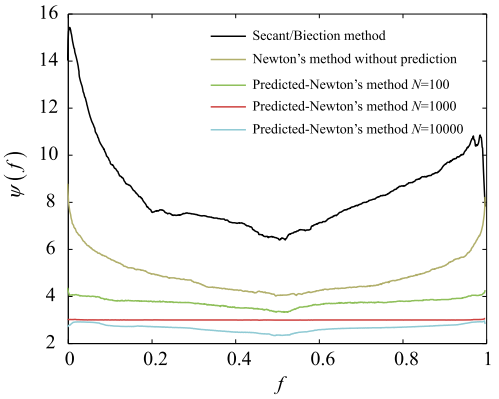

Fig. 12. The $\psi(f)$ curves of different methods. The average iteration steps of the secant/bisection method, Newton's method without prediction and the predicted-Newton's method with different $\Delta\theta$ and $\Delta\varphi$ are compared in the figure. The results indicate that the prediction of the planar constant is effective and d^{predict} is more accurate with smaller $\Delta\theta$ and $\Delta\varphi$.

Table 3
The average value of $\psi(f)$ for different situations.

Algorithm	Average value of $\psi(f)$
Secant/Bisection method	8.42
Newton's method without prediction	4.89
Predicted-Newton's method $N = 100$	3.74
Predicted-Newton's method $N = 1000$	3.00
Predicted-Newton's method $N = 10000$	2.66

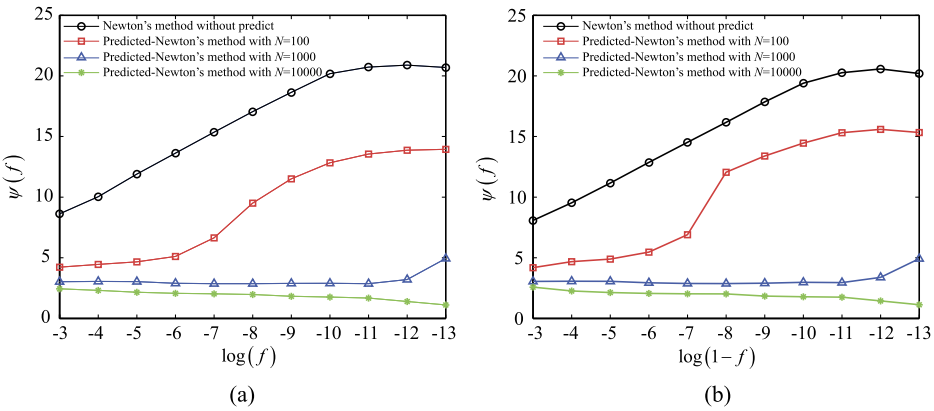

Fig. 13. The $\psi(f)$ curves of the predicted-Newton's method at $f \to 0^+$ and $f \to 1^-$.

is very important and it will significantly influence the efficiency of the iteration. If no predicted planar constant is used, the iteration will fall into the cubic Hermite interpolation easily so it takes the largest number of step to find d^* as shown in Fig. 13. However, if the predicted initial guess is introduced, $\psi(f)$ will decrease remarkably with increasing N. This is because an accurate initial guess can make the iteration stay in Newton's method rather than fall into the cubic Hermite interpolation. Finally, comparing the results in Fig. 12 and Fig. 13, the predicted planar constant is more necessary when $f \to 0^+$ and $f \to 1^-$.

5.5. The MoF interface reconstruction

In this subsection, the interface of a block and the sphere are reconstructed on different grids with increasing resolutions. The computational domain is $[0,1]^3$ with N^3 $(N = 4, 8, 16, 32, 64)$ equispaced hexahedral meshes. The MoF method is used in the orienting process while the secant/bisection method and the predicted-Newton's method are used to solve the interface positioning equation, respectively. The tolerance of solving the interface positioning equation is 10^{-14} for both methods. Fig. 14 lists the reconstructed interface using five levels of refined hexahedral meshes. The results show that the reconstructed interface converges to the true interface as mesh refines. The quantitatively comparison about the accuracy and the efficiency are presented in the following.

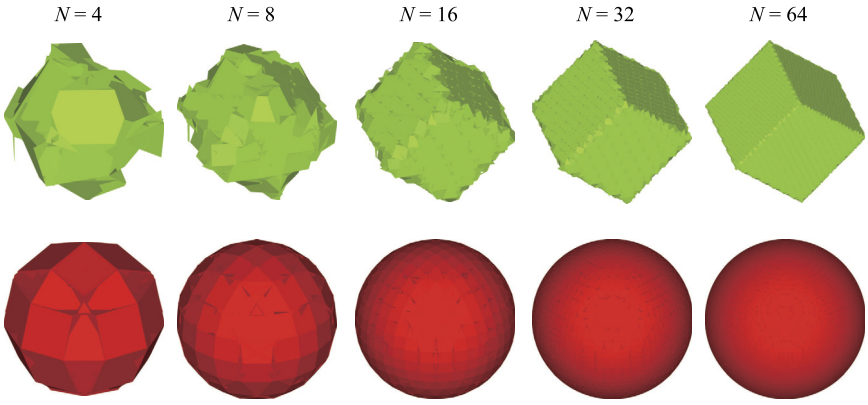

Fig. 14. The reconstructed interface on different grid with increasing resolutions.

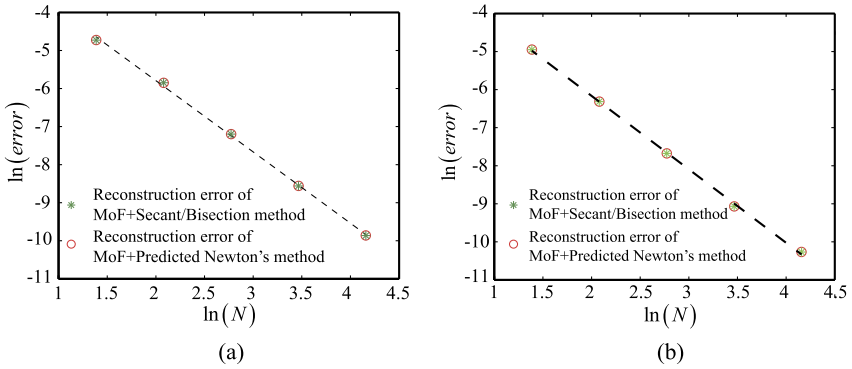

Fig. 15. The convergence rate of the reconstruction error. Figure (a) is the result of the brick reconstruction with the convergence rate equals to 1.93. Figure (b) is the result of the sphere reconstruction with the convergence equals to 1.97.

Table 4
The efficiency comparison of the secant/bisection method and the predicted-Newton's method in the MoF implementation.

Grid resolution	$N = 64$		$N = 32$		$N = 16$	
	block	sphere	block	sphere	block	sphere
I_{total} of the secant/bisection method	1.5×10^6	2.0×10^6	3.5×10^5	4.8×10^5	9.5×10^4	1.3×10^5
I_{total} of the predicted-Newton's method	4.9×10^5	6.7×10^5	1.4×10^5	1.8×10^5	3.4×10^4	4.5×10^4
T_{CPU} of the secant/bisection method (s)	3.20	4.59	0.87	1.20	0.25	0.33
T_{CPU} of the predicted-Newton's method (s)	2.04	2.81	0.58	0.76	0.17	0.22

Firstly, the convergence rate of the reconstruction error is calculated and compared with results from Ahn and Shashkov [8]. The reconstruction error is defined the same as Ahn and Shashkov, namely

$$e = \sum_i |V(\boldsymbol{T}_i \cup \boldsymbol{C}_i) - V(\boldsymbol{T}_i \cap \boldsymbol{C}_i)| \tag{35}$$

where $\boldsymbol{T}_i$ is the region of the true material in cell i and $\boldsymbol{C}_i$ is the region of the reconstructed material in cell i. Fig. 15 shows the reconstruction error on different grids by MoF+secant/bisection method and MoF+predicted-Newton's method. The results of the two schemes are almost the same and the convergence rate of the reconstruction error is similar to that recorded by Ahn and Shashkov. Therefore, from the perspective of their results, the secant/bisection method and the predicted-Newton's method are equivalent.

However, the efficiency of the predicted-Newton's method is significantly higher than the secant/bisection method. Table 4 lists the total number of iterations for solving the interface positioning equation I_{total} and the CPU time T_{CPU} used in the interface reconstruction. The numerical results indicate that 60% ∼ 66% of the iteration step can be saved by using the predicted-Newton's method which matches well with the conclusion in the subsection above. Moreover, the CPU time can be saved by 32% ∼ 39% when using the predicted-Newton's method. The lower CPU time improvement is the due to an extra truncation operation is performed after solving Eq. (2) to obtain the necessary variables required in the MoF such as the truncation centroid and the derivatives of the objective function [26]. We use the profiling tool to see the cost of each

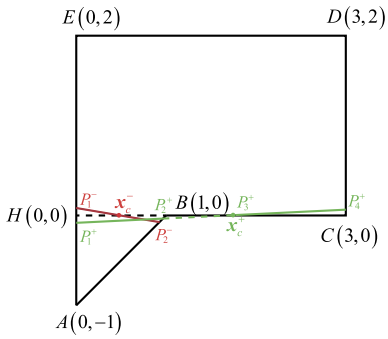

Fig. 16. The degenerated truncation of a polygon. HB is the interface. The red segment is the infinitesimal anticlockwise rotation of the interface while the green segments are the infinitesimal clockwise rotation of the interface. $\boldsymbol{x}_c^-$ and $\boldsymbol{x}_c^+$ are the centroid of the red and green segments, respectively.

part and find that the CPU time used in the extra calculation of the MoF method is about the half of the CPU time used in solving Eq. (2) by the predicted-Newton's method. Therefore a lower overall efficiency improvement is presented in practice.

6. Conclusion

In this paper, the interface positioning equation is studied from another perspective by focusing on the relation between the planar constant and the interface orientation for a given volume fraction. The analytical partial derivatives of the planar constant with respect to the interface orientation are deduced by the geometry analysis. These partial derivatives can be used to predict the planar constant whenever the interface orientation is changed which is useful for the iterative methods to solve the interface positioning equation. Moreover, an improved Newton's method, which utilizes the monotonicity of $V(d)$ to guarantee the convergence, is proposed and it takes the predicted planar constant as the initial guess. The predicted-Newton's method is implemented on a generalized hexahedron cell and the numerical studies indicate that it performs well even if a cell face is non-planar. Finally, the efficiency of the predicted Newton's method is compared with the commonly used secant/bisection method proposed by Ahn and Shashkov. The numerical results indicate that when implementing in the MoF method, the number of iteration step can be reduced by 60% ∼ 66% in solving the interface positioning equation and the CPU time can be reduced by 32% ∼ 39%.

Acknowledgements

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

Appendix A. The analysis of the degenerated truncation

The degenerated truncation means the truncation face is coincident with one polyhedron face. For the sake of clarity, we take a 2D problem as an example. As shown in Fig. 16, the normal of the interface is $\boldsymbol{n} = (1, 0)$, namely $\theta = \pi/2$, and the reference volume faction is 1/13. Therefore, the planar constant $d = 0$ and the interface HB is coincident with the edge BC. It should be mentioned that the degenerated truncation is almost impossible in practice because it only occurs when the volume fraction and the orientation equal to some certain values.

The numerical left and right derivatives can be calculated by finite difference as

$$\begin{aligned} d'_{\text{num}}\left(\theta^-\right) &= \frac{d(\theta) - d(\theta - \Delta\theta)}{\Delta\theta} = 0.5 \\ d'_{\text{num}}\left(\theta^+\right) &= \frac{d(\theta + \Delta\theta) - d(\theta)}{\Delta\theta} = 1.5501 \end{aligned} \tag{A.1}$$

at $\theta = \pi/2$ with $\Delta\theta = 10^{-7}$. $d'_{\text{num}}\left(\theta^-\right)$ does not equal to $d'_{\text{num}}\left(\theta^+\right)$ in the degenerated truncation and we will explain it as follows.

According to Eq. (24), the left and right analytical derivative can be written as

$$\begin{aligned} d'_{\text{ana}}\left(\theta^-\right) &= -\boldsymbol{x}_c^- \cdot \boldsymbol{n}'(\theta) \\ d'_{\text{ana}}\left(\theta^+\right) &= -\boldsymbol{x}_c^+ \cdot \boldsymbol{n}'(\theta) \end{aligned} \tag{A.2}$$

However, because HB is coincident with BC, the subdivision type at θ^- and θ^+ are different. As shown in Fig. 16, the red segment $P_1^- P_2^-$ is the interface at θ^- which belongs to the type (a) in Fig. 8. Segment $P_1^+ P_2^+$ and $P_3^+ P_4^+$ are the interface at θ^+ which belongs to the type (c) in Fig. 8. Obviously, the centroid of interface at θ^- and θ^+, namely $\boldsymbol{x}_c^-$ and $\boldsymbol{x}_c^+$, are

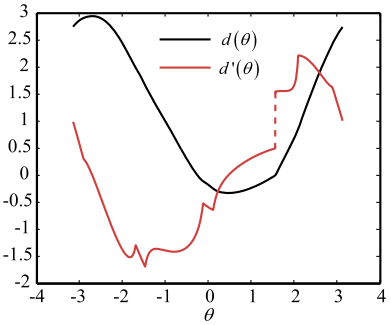

Fig. 17. The curves of $d(\theta)$ and $d'(\theta)$ and $d'(\theta)$ is discontinuous at $\theta = \pi/2$.

different which leads to the discontinuity observed in Eq. (A.1). Specifically, $\boldsymbol{x}_c^- = (0.5, 0)$ and $\boldsymbol{x}_c^+ = \left(4 - \sqrt{6}, 0\right)$ which can be obtain by solving

$$\boldsymbol{x}_c^- = \frac{\int_H^B \boldsymbol{x} \mathrm{d}l}{|BH|} \tag{A.3}$$

$$\boldsymbol{x}_c^+ = \frac{\int_H^B \boldsymbol{x} \mathrm{d}l + \int_{\boldsymbol{x}_c^+}^C \boldsymbol{x} \mathrm{d}l}{|BH| + \left|\boldsymbol{x}_c^+ C\right|} \tag{A.4}$$

Substituting $\boldsymbol{x}_c^-$ and $\boldsymbol{x}_c^+$ in to Eq. (A.2) and compare the result in Eq. (A.1), we have $d'_{\text{num}}\left(\theta^-\right) = d'_{\text{ana}}\left(\theta^-\right)$ and $d'_{\text{num}}\left(\theta^+\right) = d'_{\text{ana}}\left(\theta^+\right)$.

Therefore, Eq. (24) and Eq. (25) are also applicable for the degenerated truncation. But in this situation, $\boldsymbol{x}_c$ depends on the orientation direction which will lead to the discontinuity of the derivatives. Fig. 17 shows the curves about $d(\theta)$ and $d'(\theta)$ and as our expectation, $d'(\theta)$ is discontinuous at $\theta = \pi/2$.

The discontinuity of $d'(\theta)$ will decrease the accuracy of the predicted planar constant in the proposed predicted-Newton's method if θ and $\theta + \Delta\theta$ locate on the opposite side of the discontinuity. In this situation, more iterations are required to find d^*. However, it just produces minor influence on the interface reconstruction process because the degenerated truncation is almost impossible in practice.

References

[1] S. Diot, M.M. Francois, An interface reconstruction method based on an analytical formula for 3D arbitrary convex cells, J. Comput. Phys. 305 (2016) 63–74.
[2] J. Lopez, J. Hernandez, Analytical and geometrical tools for 3D volume of fluid method in general grids, J. Comput. Phys. 227 (2008) 5939–5948.
[3] J. Lopez, J. Hernandez, P. Gomez, F. Faura, A new volume conservation enforcement method for PLIC reconstruction in general convex grids, J. Comput. Phys. 316 (2016) 338–359.
[4] V. Dyadechko, M. Shashkov, Moment-of-Fluid Interface Reconstruction, Tech. Rep. LA-UR-05-7571, Los Alamos National Laboratory, Los Alamos, 2005.
[5] J. Lopez, P. Gomez, C. Zanzi, F. Faura, J. Hernandez, Application of non-convex analytical and geometric tools to a PLIC-VOF method, in: Proceedings of the ASME International Mechanical Engineering Congress and Exposition, vol. 7, Amer Soc Mech Engineers, ISBN 978-0-7918-5061-9, 2017, pp. 11–17.
[6] X. Yang, A.J. James, Analytic relations for reconstructing piecewise linear interfaces in triangular and tetrahedral grids, J. Comput. Phys. 214 (2006) 41–54.
[7] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[8] H.T. Ahn, M. Shashkov, Multi-Material Interface Reconstruction on Generalized Polyhedral Meshes, Tech. Rep. LA-UR-07-0656, Los Alamos National Laboratory, 2007.
[9] H.T. Ahn, M. Shashkov, Geometric algorithms for 3D interface reconstruction, in: Proceedings of the 16th International Meshing Roundtable, Springer, Berlin Heidelberg, 2008, pp. 405–422.
[10] M.B. Stephenson, H.N. Christiansen, A polyhedron clipping and capping algorithm and a display system for three dimensional finite element method, ACM SIGGRAPH Comput. Graph. 9 (1975) 1–16.
[11] X. Chen, X. Zhang, Z. Jia, A robust and efficient polyhedron subdivision and intersection algorithm for three-dimensional MMALE remapping, J. Comput. Phys. 338 (2017) 1–17.
[12] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096–2132.
[13] 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 of MOF interface reconstruction, J. Comput. Phys. 236 (2013) 513–562.
[14] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, 1982, pp. 273–285.
[15] C.W. Hirt, B.D. Nichols, Volume of fluid method for the dynamics of free boundary, J. Comput. Phys. 39 (1981) 201–255.
[16] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithm for tracking material interface, J. Comput. Phys. 199 (2004) 465–502.
[17] B. Swartz, The second-order sharpening of blurred smooth borders, Math. Comput. 186 (1989) 675–714.
[18] S.J. Mosso, B.K. Swartz, D.B. Kothe, R.C. Ferrell, A parallel volume-tracking algorithm for unstructured meshes, in: P. Schiano, A. Ecer, J. Periaux, N. Satofuka (Eds.), Parallel Computational Fluid Dynamic Algorithms and Results Using Advanced Computers, Elsevier Science, 1997, pp. 368–375.
[19] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361–5384.

[20] M. Kucharik, R.V. Garimella, S.P. Schofield, M.J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulation, J. Comput. Phys. 229 (2010) 2432–2452.
[21] H.R. Anbarlooei, K. Mazaheri, Moment of fluid interface reconstruction method in multi-material arbitrary Lagrangian Eulerian algorithm, Comput. Methods Appl. Mech. Eng. 198 (2009) 3782–3794.
[22] E. Kikinzon, Y. Kuznetsov, K. Lipnikov, M. Shashkov, Approximate static condensation algorithm for solving multi-material diffusion problems on meshes non-aligned with material interfaces, J. Comput. Phys. 347 (2017) 416–436.
[23] M. Jemison, M. Sussman, M. Shashkov, Filament capturing with the Multimaterial Moment-of-Fluid method, J. Comput. Phys. 285 (2015) 149–172.
[24] M. Friess, J. Breil, P. Maire, M. Shashkov, A multi-material CCALE-MOF approach in cylindrical geometry, Commun. Comput. Phys. 15 (2014) 330–364.
[25] X. Chen, X. Zhang, An improved 2D MoF method by using high order derivatives, J. Comput. Phys. 349 (2017) 176–190.
[26] X. Chen, X. Zhang, An improved 3D MoF method based on analytical partial derivatives, J. Comput. Phys. 326 (2016) 156–170.
[27] A. Lemoine, S. Glockner, J. Breil, Moment-of-fluid analytic reconstruction on 2D Cartesian grids, J. Comput. Phys. 328 (2017) 131–139.