

An improved continuity-preserving interface reconstruction method for multi-material flow

Shengping Liu^a, Heng Yong^a, Shaodong Guo^{a,*}, Yiqing Shen^{b,c}, Guoxi Ni^a

^aInstitute of Applied Physics and Computational Mathematics, Beijing 100094, China

^bState Key Laboratory of High Temperature Gas Dynamics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China

^cSchool of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:

Received 10 February 2021

Accepted 8 April 2021

Available online 23 April 2021

Keywords:

Dynamic programming interface reconstruction

Continuity-preserving

Volume-preserving

ABSTRACT

The dynamic programming interface reconstruction (DPIR) method introduced by Dumas et al. [1] is a volume-preserving and continuous interface reconstruction method. It is a two-step method, which comprises of an optimized step and a correction step. At first, in the optimized step, it minimizes a target function by the dynamic programming method to obtain a continuous interface. Then, it corrects the interface in each mixed cell to preserve the conservative of the volume fraction. However, only the difference of volume fraction is considered, and the interface normal is neglected in the target function. These make it easy to obtain different optimal results in the optimized step, and hence the resulting continuous interfaces always suffer from oscillations (i.e., the 'wave effects' [1]). In this paper, to suppress the continuous interfaces' oscillations in the optimized step and improve its accuracy, we constructed a non-dimensional target function based on the moment-of-fluid method's objective function, and also proposed a new correction method. Finally, several numerical tests are performed to show the new method's superiority over the original one of Dumas et al. [1].

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

There is a variety of interface reconstruction methods for multi-material flows. Among all of them, the volume-of-fluid (VoF) method [2] can preserve the mass of each component, and for a wide range of applications, the importance of mass conservation on a discrete level outbalances all the disadvantages associated with VoF methods [3]. In the VoF method, an approximate piecewise linear interface ($\vec{n} \cdot \vec{x} + d = 0$) in the mixed cell is constructed in two steps, determining the direction of the interface (i.e., the normal \vec{n}) and calculating the location of the interface (i.e., the constant d). The constant d in the second step can be uniquely calculated as long as \vec{n} and volume fraction are given [3,4]. Hence, the estimation of \vec{n} is much more critical, it will significantly influence the accuracy of interface reconstruction [5]. There are many different algorithms to derive the interface normal [2–4,6–11]. The moment-of-fluid (MoF) method [3] is one of the most accurate approaches among these VoF methods [5], and it can resolve interface details as small as the mixed cell itself [3].

However, it is well known that these VoF methods suffer from discontinuity of interface shape across the cell boundaries. To

overcome this deficiency, Dumas et al. [1] introduced a volume-preserving and continuous interface reconstruction method, which is denoted as DPIR. It is a two-step method containing an optimized step and a correction step. In the optimized step, a continuous interface can be obtained by minimizing a target function with the dynamic programming (DP) method. And then, in the correction step, the interface in each mixed cell is corrected to preserve the conservative of the volume fraction (i.e., guarantee the mass conservation). This target function is quite important for the DPIR method since it will significantly influence the accuracy of the reconstructed continuous interface in the optimized step and the robustness in the correction step. As the numerical results have shown [1], the continuous interface obtained by the target function suffers from oscillation (i.e., the 'wave effects' observed in [1]). To solve this problem, Dumas et al. [1] introduced a tunable penalty term to the target function. Recently, Chollet et al. [12] found that this penalty term may have very different scales with the target function and added a second penalty term to fix this problem.

As our numerical tests showed, both the penalty terms [1,12] are insufficient to suppress the artificial oscillation in the continuous interfaces. Furthermore, we notice that the penalty terms introduced in [1] and [12] for the target function have different dimension units. As is well known, all equations must be dimensionally consistent (also called dimensional homogeneity), i.e.,

* Corresponding author.

E-mail address: guo_shaodong@iapcm.ac.cn (S. Guo).

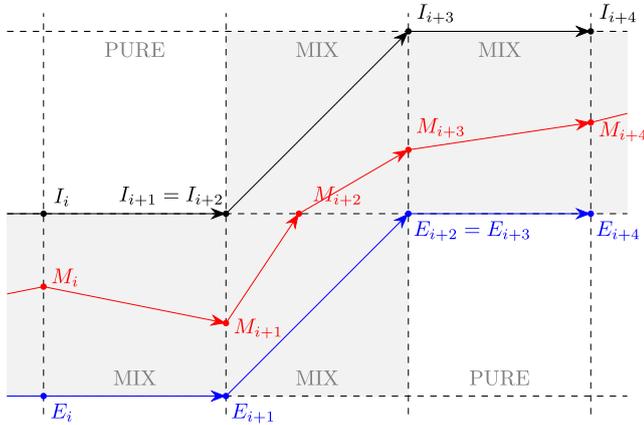


Fig. 1. The interface curve (red points M_i) with the internal (black point I_i) and external (blue points E_i) curves. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the units should be the same on both sides of an equation, otherwise, some spurious phenomena will be generated (see [13] for an example).

In addition, the target function [1] only bases on the volume fractions and does not consider the effect of the interface normal \bar{n} , it is difficult to maintain the uniqueness of the continuous interface, and hence an oscillatory interface may be generated. In this paper, to suppress oscillations of the continuous interfaces, a non-dimensional target function, which contains both the effects of the interface normal and volume fractions, is designed. And a correction step is also proposed to improve its accuracy and robustness. Finally, several tests are presented to show the superiority of the new method over the original one [1].

This paper is organized as follows: the DPIR method is briefly introduced in Section 2. Our new method is given in Section 3. In Section 4, several tests are presented to demonstrate the good performance of the new method. Concluding remarks are given in Section 5.

2. DPIR method

DPIR method is a two-step method. In the optimized step, dynamic programming (DP) is used to minimize a target function to obtain the continuous interface. And in the correction step, a local correction is made on each mixed cell to preserve the volume conservation.

2.1. The optimized step

First, the interface is assumed to be piecewise linear in each cell. As shown in Fig. 1, the minimization problem consists of finding a finite number of points ($\vec{M}_i, 0 \leq i \leq N$) located on the edges of mixed cells. The points \vec{M}_i are bounded in the external and internal points, i.e., E_i and I_i . By taking $\vec{M}_i = x_i \cdot \vec{I}_i + (1 - x_i) \cdot \vec{E}_i$, with $0 < x_i < 1$, Dumas et al. [1] designed a target function as follows,

$$f = \sum_{i=0}^{N-1} f_i, \quad f_i(x_i, x_{i+1}) = |\text{vol}(\vec{M}_i, \vec{M}_{i+1}) - \text{vol}_r|^p \quad (1)$$

where, vol_r is the exact volume fraction of the internal material (here, the materials separated by the interface are denoted as internal and external ones), and $\text{vol}(\vec{M}_i, \vec{M}_{i+1})$ is the one computed through the polygon ($\vec{I}_i - \vec{M}_i - \vec{M}_{i+1} - \vec{I}_{i+1}$). And the power $p = 2$ is suggested [1]. Since the function f_i have two continuous variables x_i and x_{i+1} , in order to implement the dynamic programming

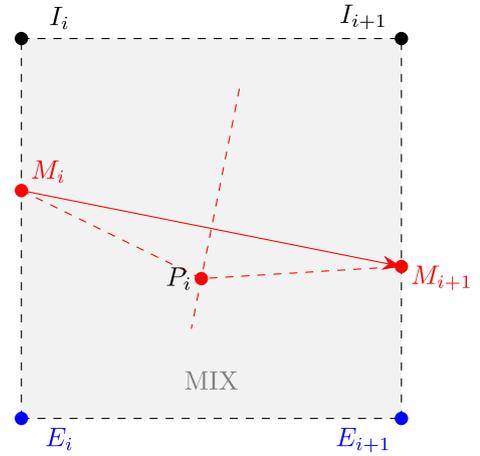


Fig. 2. The local correction of volume fraction.

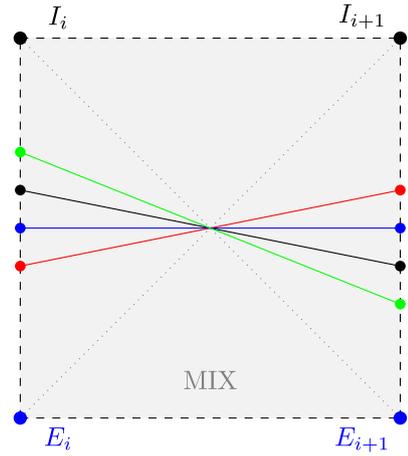


Fig. 3. Several optimal solutions of the target function (1) in this mixed cell.

method, Dumas et al. discrete each x_i into a finite set of points, i.e.,

$$x_i \in (0, 1) \Rightarrow x_i \in (x^1, x^2, \dots, x^{N_k-1}), x^j = \frac{j}{N_k}, \quad (2)$$

and then by fixing the value of $x_0 = x_N$, they can optimize the others as the same as the shortest path problem. After iterating the optimized solutions for all the possible values of $x_0 = x_N$, they get the optimized solution, i.e., an optimal continuous interface made up of the points $\vec{M}_0 - \vec{M}_1 - \dots - \vec{M}_N$. Please refer to [1] for more details.

However, the continuous interface obtained by Eq. (1) often oscillates between mixed cells. To fix this problem, Dumas et al. add a penalty term to the target function (1),

$$f_i(x_i, x_{i+1}) = |\text{vol}(\vec{M}_i, \vec{M}_{i+1}) - \text{vol}_r|^p + \lambda \cdot \|\vec{M}_{i+1} - \vec{M}_i\| \quad (3)$$

where, $\|\vec{M}_{i+1} - \vec{M}_i\|$ is the distance between points \vec{M}_i and \vec{M}_{i+1} . As their tests showed, the values of λ have a great influence on the results, and a weak penalization with $\lambda = 0.01$ is suggested [1].

In [12], Chollet et al. add another penalty term to the target function (3),

$$f_i(x_i, x_{i+1}) = |\text{vol}(\vec{M}_i, \vec{M}_{i+1}) - \text{vol}_r|^p + \lambda \cdot \|\vec{M}_{i+1} - \vec{M}_i\| + \frac{|\text{vol}_r - \text{vol}(\vec{M}_i, \vec{M}_{i+1})|}{\text{vol}(\vec{M}_i, \vec{M}_{i+1})} \quad (4)$$

If the calculated volume fraction $\text{vol}(\vec{M}_i, \vec{M}_{i+1})$ is too small with respect to the correction part ($|\text{vol}_r - \text{vol}(\vec{M}_i, \vec{M}_{i+1})|$), the second

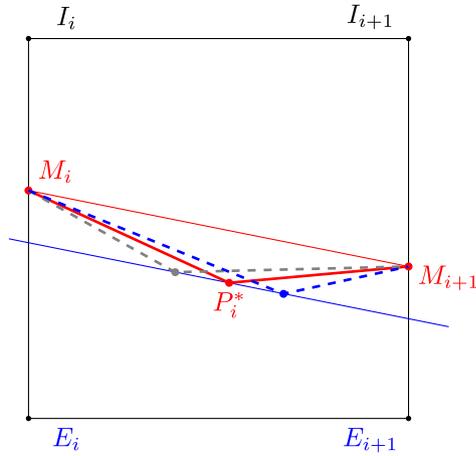


Fig. 4. The new local correction method.

penalty term will be a much big number, and hence it will not be chosen by the minimization.

However, we notice that the unit of the second part (unit of distance) for the target function in Eqs. (3) and (4) differs from the others (unit of 1). As mentioned previous, dimensional inconsistent equations may result in spurious solutions, hence this penalty term may limit its applications.

2.2. The correction step

This step is to make the remaining error ($|\text{vol}_r - \text{vol}(\vec{M}_i, \vec{M}_{i+1})|$) in volume fractions disappear after the optimized step. It is performed locally by adding a control point in each mixed cell. As shown in Fig. 2, with the point \vec{P}_i , lying on the perpendicular bisector of each segment $\vec{M}_i - \vec{M}_{i+1}$, the corresponding volume fraction defined by the triangle $(\vec{M}_i - \vec{P}_i - \vec{M}_{i+1})$ will be equal to the remaining error. First, we can obtain the height of this triangle with the remaining error, then we can get the location of \vec{P}_i by geometric relation.

To obtain a curved interface for specific cases (circle, for instance), Chollet et al. [12] introduced a curved interface correction method. They also suggested moving the control point along the

direction of the cell center instead of the perpendicular bisector. This may ensure more space available, and thus make the correction step more robust. Please refer to [12] for more information.

3. The new method

We find out that without the interface normal, the VoF method is no longer guaranteed uniqueness, and hence the target function Eq. (1) may have multiple solutions. For example, as the mixed cell ($\text{vol}_r = 0.5$) shown in Fig. 3, for all these lines $|\text{vol}(\vec{M}_i, \vec{M}_{i+1}) - \text{vol}_r|^p = 0$ are satisfied, and hence any of these lines may be regarded as optimal solution of the function (1). That is why the resulting interface suffers from oscillations.

In this section, to overcome this problem, we design a non-dimensional target function to suppress the oscillations.

3.1. The new target function

We noticed that the moment-of-fluid method [3] could obtain piecewise linear interfaces by solving the optimal solution of the objective function,

$$g(\theta) = \|\vec{x}_c(\theta) - \vec{x}_c^*\|_2^2 \tag{5}$$

where, θ is the polar angle which can specify the interface normal \vec{n} . $\vec{x}_c(\theta)$ and \vec{x}_c^* are the centroids of the calculated and exact material's volume, respectively. As the analysis has shown in [3], the optimal solution of (5) is unique. And as a side benefit, the amount of information carried by the volumes and centroids is sufficient to define the piecewise linear interface without exchanging data between the neighbor cells [3]. It means that the objective function (5) can be evaluated by a function of x_i and x_{i+1} , and hence be easily implemented in the DPIR method as the original function.

Considering the influence of the volume fraction and the objective function (5), a new dimensionless target function is designed as follows,

$$f = \sum_{i=0}^{N-1} f_i, \quad f_i(x_i, x_{i+1}) = \left(\frac{\|\vec{x}_c(\vec{M}_i, \vec{M}_{i+1}) - \vec{x}_c^*\|^2}{A_r} \cdot \frac{1}{\text{vol}_r} \right)^p \tag{6}$$

where $p = 1/5$, $\vec{x}_c(\vec{M}_i, \vec{M}_{i+1})$ and \vec{x}_c^* are the calculated and reference centroids of the target material, A_r and vol_r are the volume

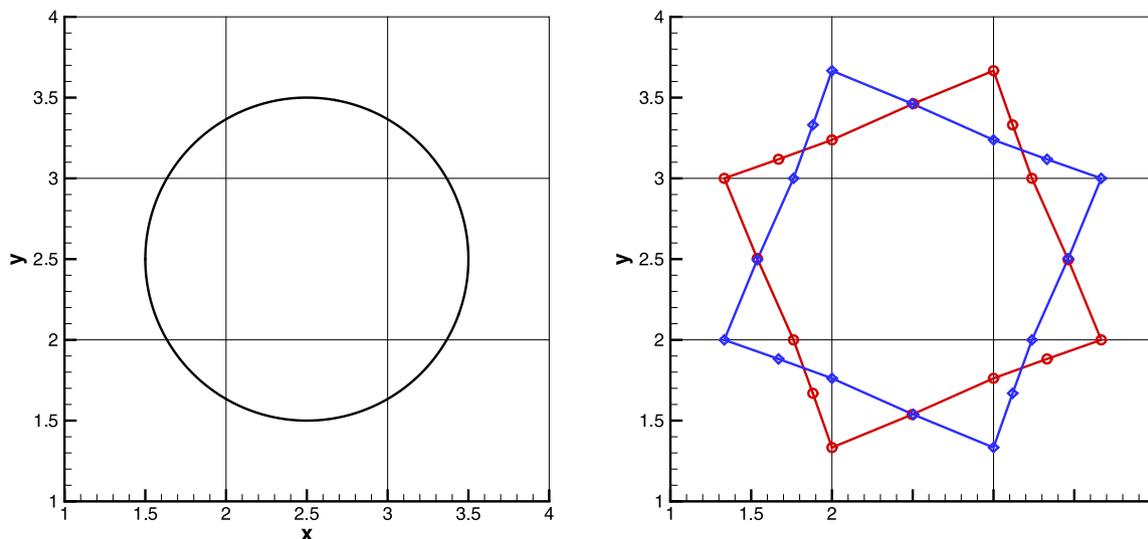


Fig. 5. Left: the initial mesh and interface; Right: two possible solutions of DPIR-O.

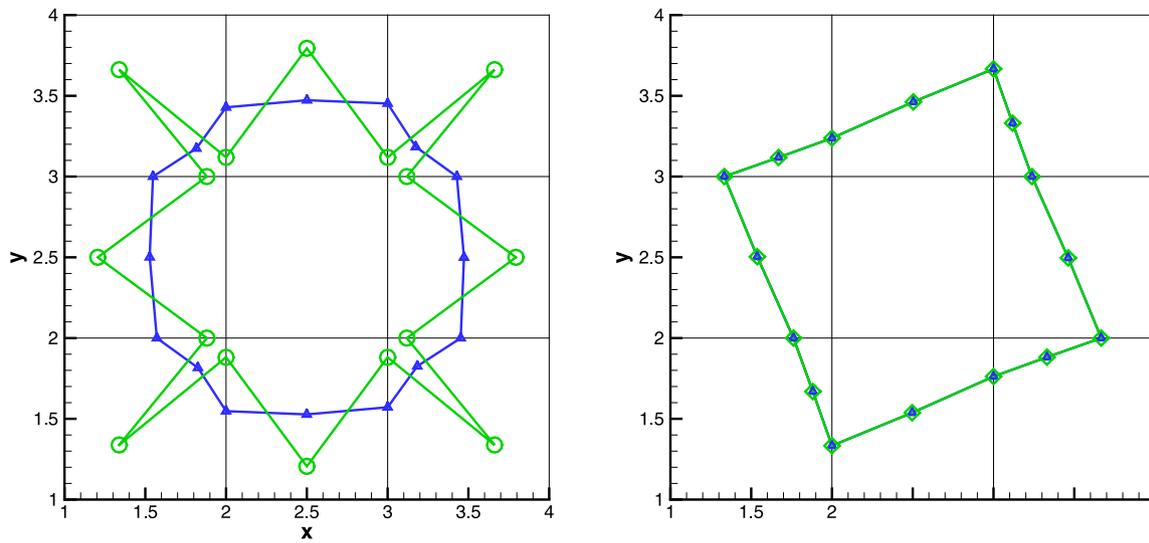


Fig. 6. Left: the solutions of DPIP-P1; Right: the possible solutions of DPIP-P2. (Blue- $\lambda = 0.01$, Green- $\lambda = 0.5$).

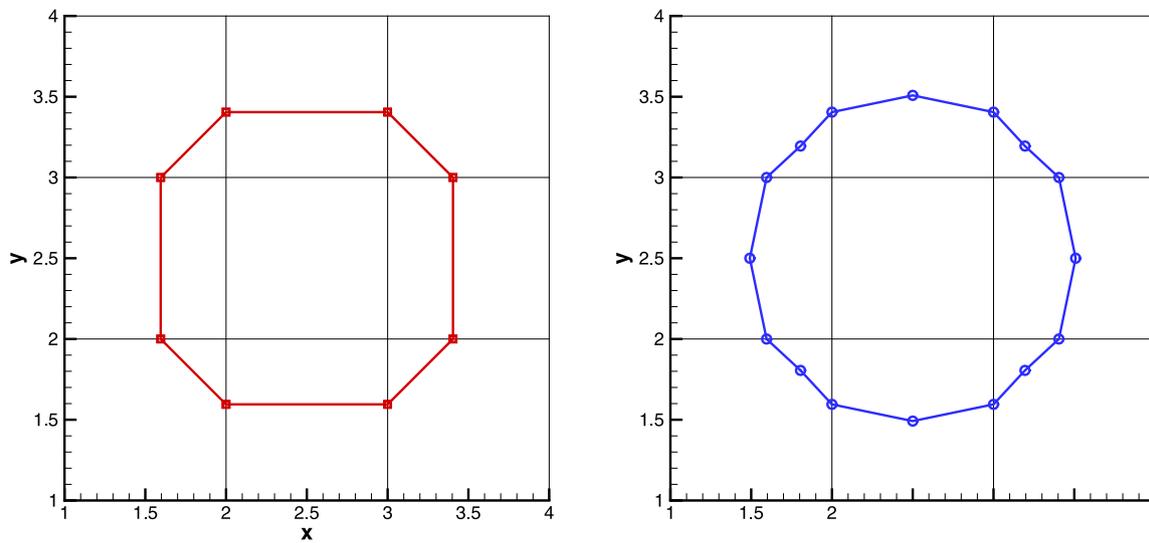


Fig. 7. The solution of DPIP-N. Left: the optimized interface in the first step; Right: the corrected interface in the second step.

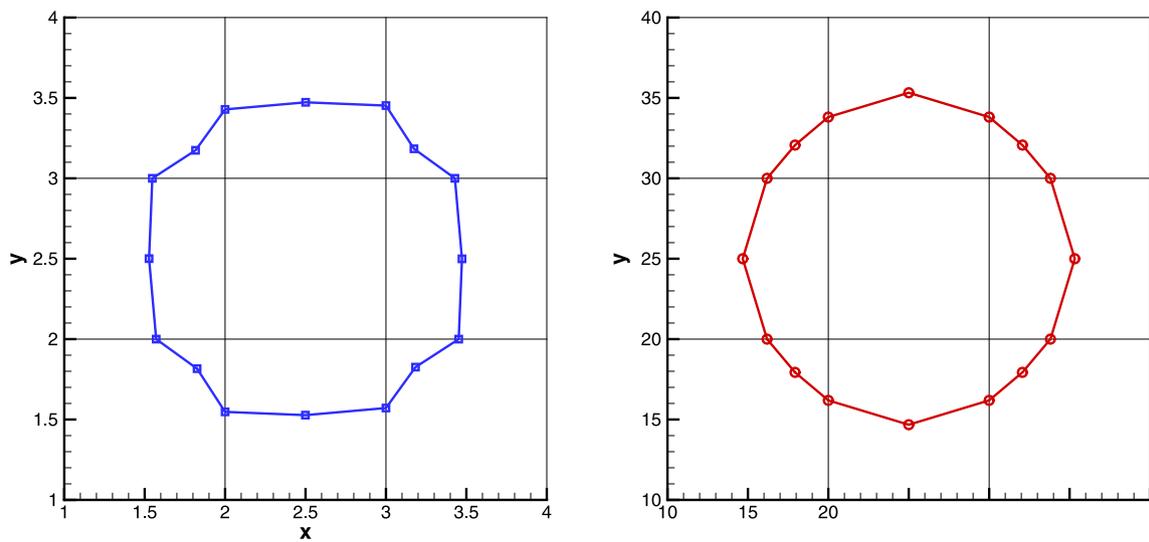


Fig. 8. Left: the solution on the original mesh; Right: the solution on the amplified mesh. (DPIP-P1 with $\lambda = 0.01$ is used).

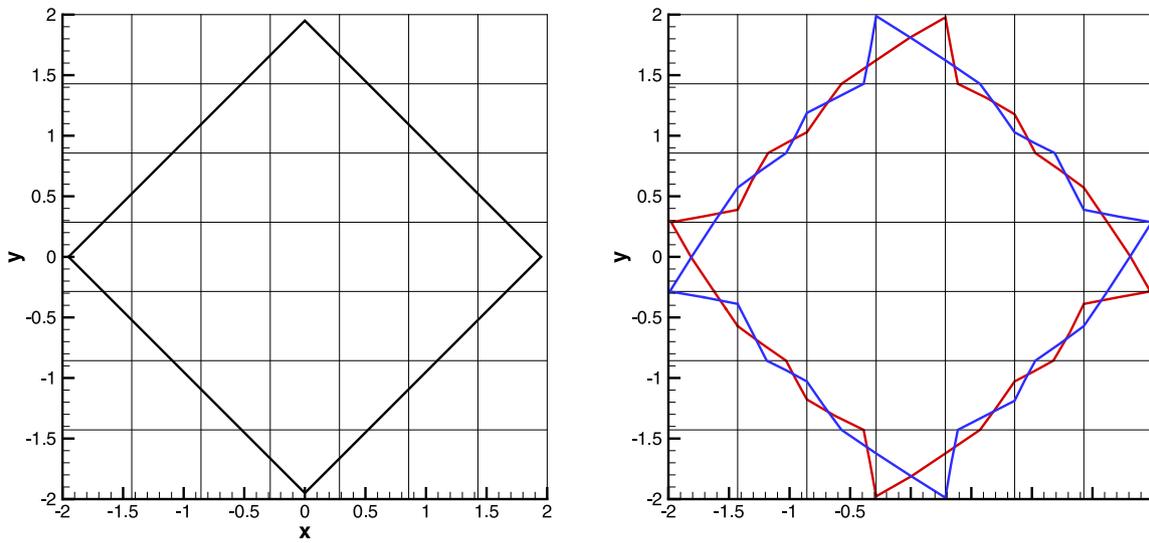


Fig. 9. Left: the initial mesh and interface; Right: two possible solutions of DPIR-O.

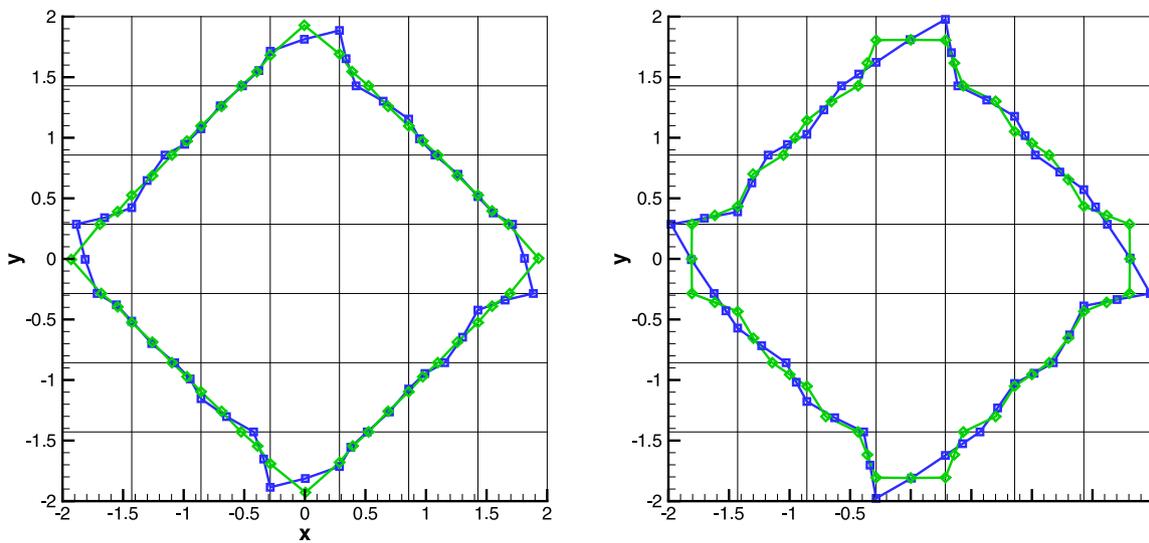


Fig. 10. Left: the possible solutions of DPIR-P1; Right: the possible solutions of DPIR-P2. (Blue- $\lambda = 0.01$, Green- $\lambda = 0.5$).

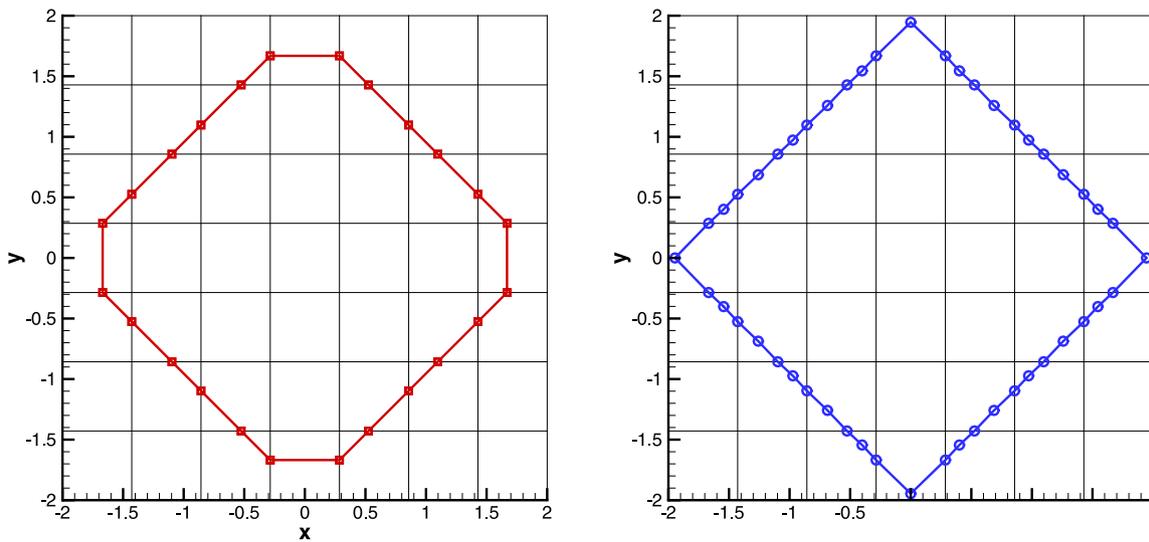


Fig. 11. The solution of DPIR-N. Left: the optimized interface in the first step; Right: the corrected interface in the second step.

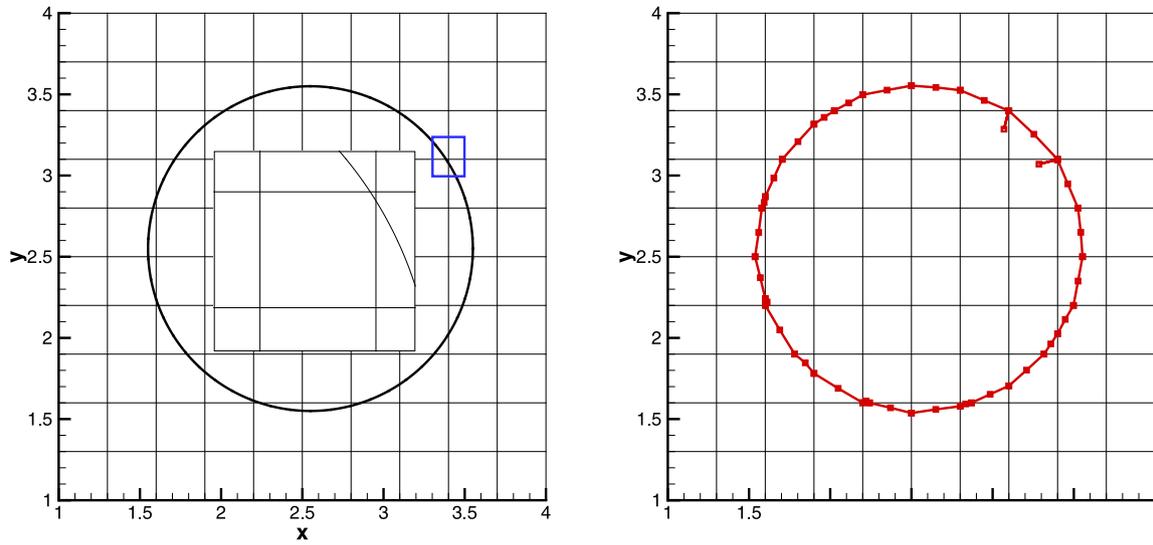


Fig. 12. Left: the initial mesh and interface; Right: the solution of DPIR-P1.

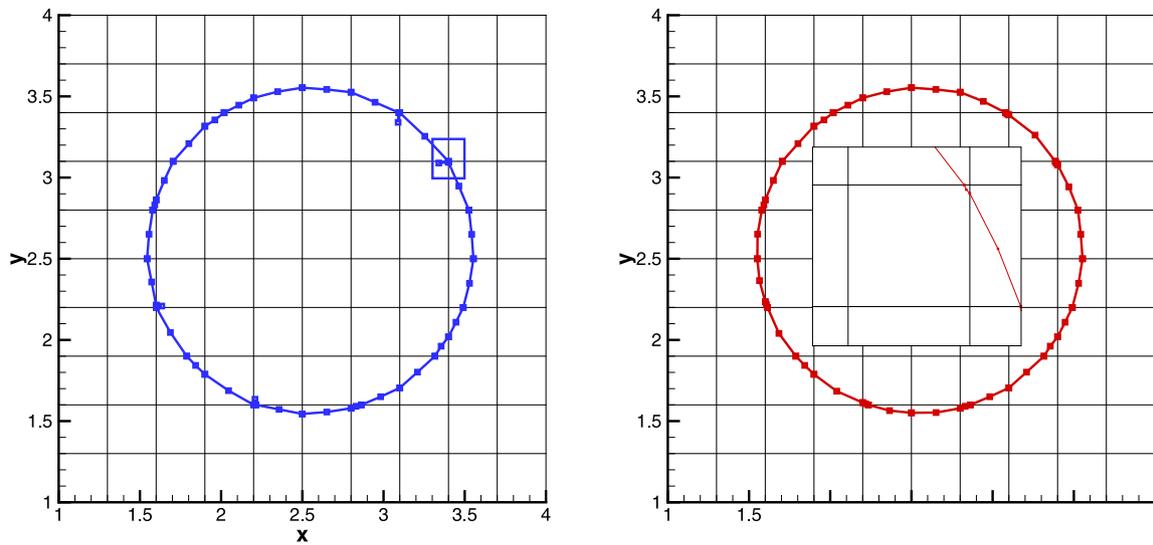


Fig. 13. Left: the solutions of DPIR-N1; Right: the solutions of DPIR-N.

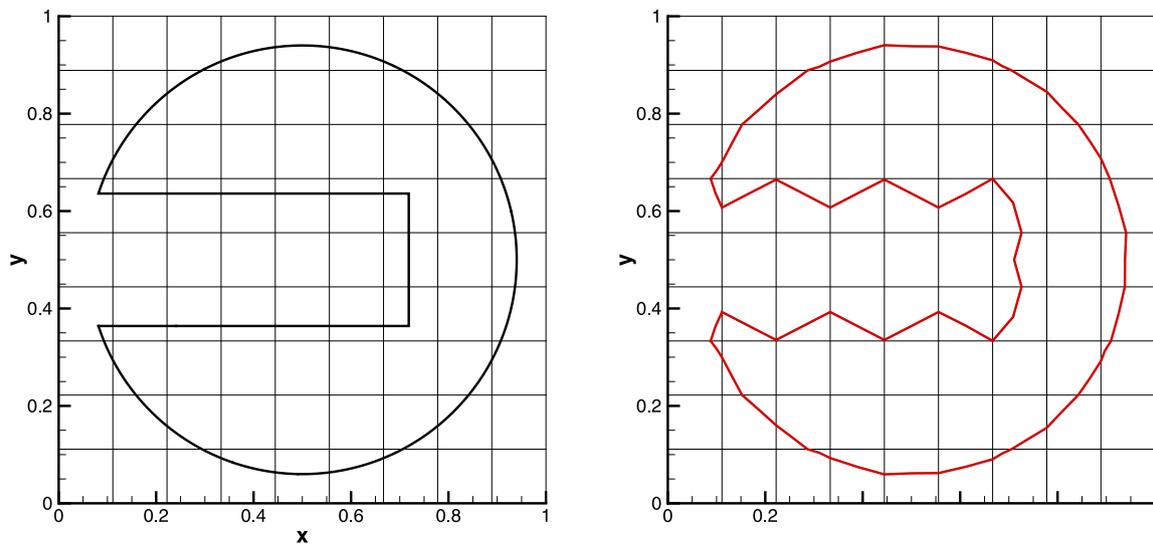


Fig. 14. Left: the initial mesh and interface; Right: the solution of DPIR-P1.

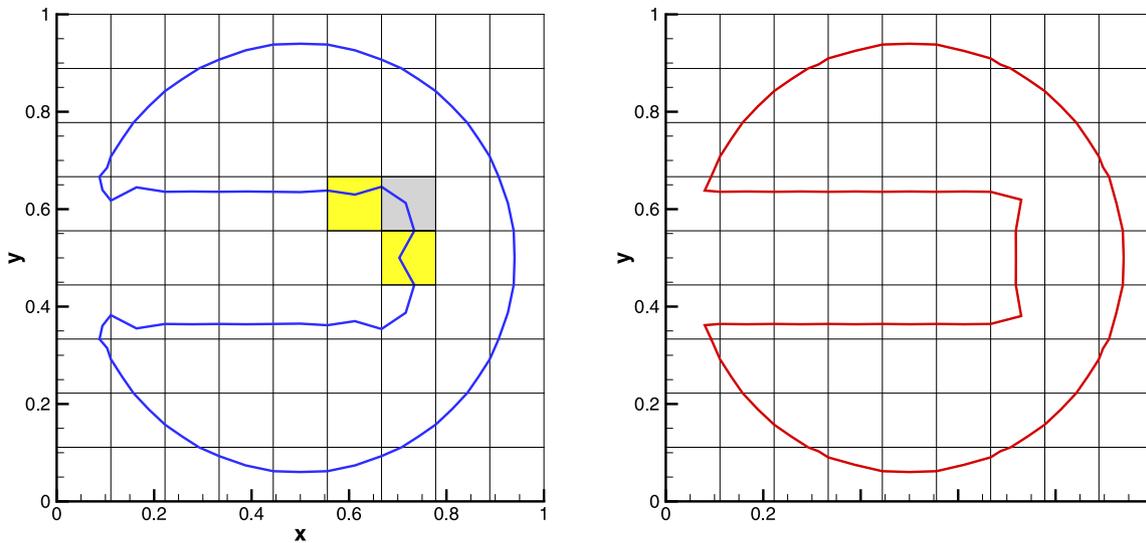


Fig. 15. Left: the solutions of DPIP-N2; Right: the solutions of DPIP-N.

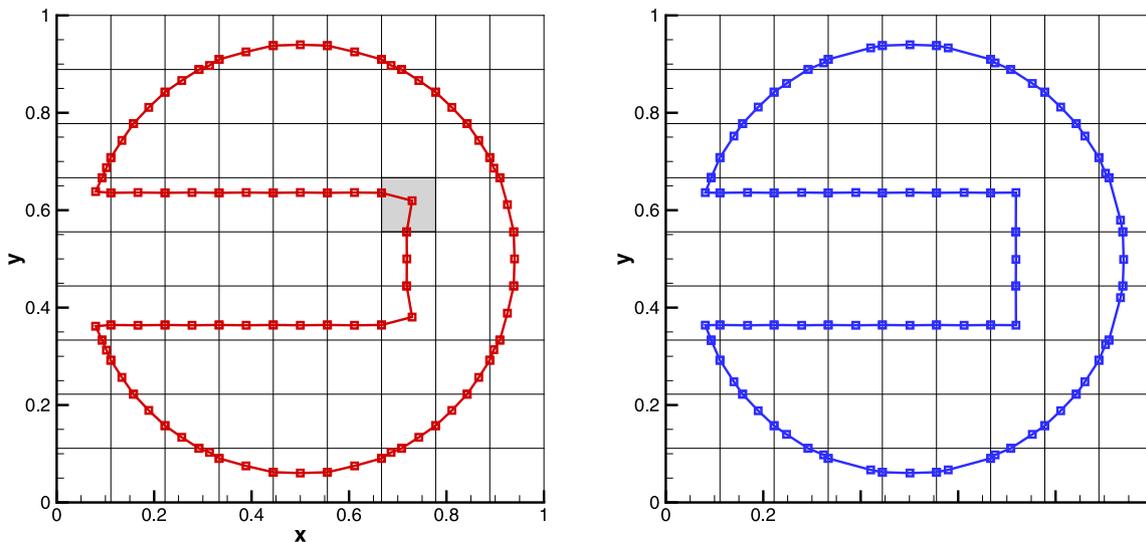


Fig. 16. The solutions of DPIP-N. Left: with simple correction method [1]; Right: with the new correction method.

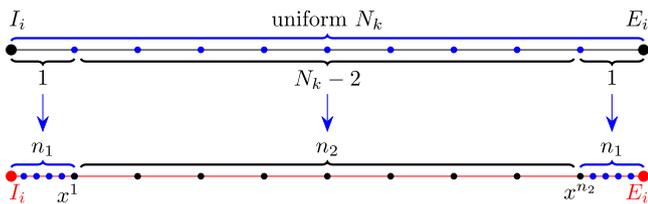


Fig. 17. A small change in the discretization of N_k .

and volume fraction of the target material. And this target material is always the one with a smaller fraction,

$$\begin{cases} (\bar{x}_c^*(\bar{M}_i, \bar{M}_{i+1}), \bar{x}_c^*, A_r, \text{vol}_r) \text{ refers to the internal material,} \\ \text{if } \text{vol}_l < \text{vol}_E \\ (\bar{x}_c^*(\bar{M}_i, \bar{M}_{i+1}), \bar{x}_c^*, A_r, \text{vol}_r) \text{ refers to the external material,} \\ \text{otherwise.} \end{cases} \quad (7)$$

where, vol_l and vol_E is the exact volume fraction of the internal and external materials. For example, if $\text{vol}_l < \text{vol}_E$,

$\bar{x}_c^*(\bar{M}_i, \bar{M}_{i+1}), \bar{x}_c^*, A_r$ will be calculated through the polygon $(\bar{I}_i - \bar{M}_i - \bar{M}_{i+1} - \bar{I}_{i+1})$ as shown in Fig. 2.

It is easy to know that, the optimized solution of Eq. (6) is composed of a set of approximate solutions of (5). Especially for a piecewise linear interface, due to its uniqueness, the optimized solution is identical to the results of the MoF method and hence avoid the ‘wave effects’. Although in the present state of our knowledge, it is difficult to verify the uniqueness property for a curved interface, various numerical tests show that the new function is sufficient to suppress the oscillatory of the continuous interfaces.

3.2. The new local correction method

Unlike the simple correction method in [1], we try to find an optimal correction point \bar{P}_i^* in the second step. As shown in Fig. 4, it is easy to know that the correction points \bar{P}_i and \bar{P}_i^* should be located on the solid blue line, which is parallel with the line $\bar{M}_i - \bar{M}_{i+1}$. As same as the MoF method, the objective function $g(\theta)$ (5) is used to find the optimal correction point on the blue line, and the only difference is that in a DPIP method, the mixed cell is separated by $\bar{M}_i - \bar{P}_i^* - \bar{M}_{i+1}$ instead of a linear line.

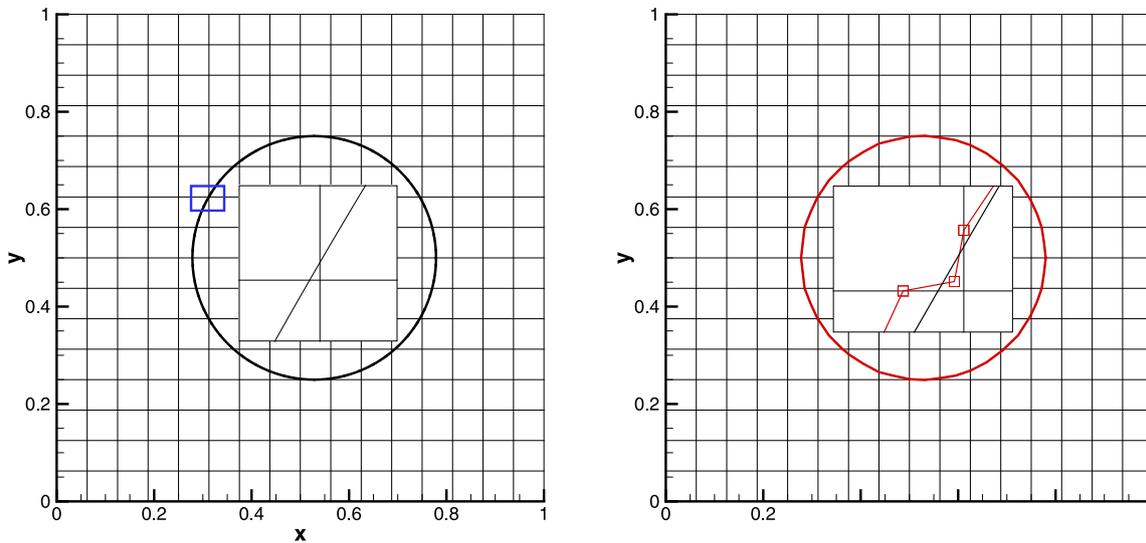


Fig. 18. Left: the initial mesh and interface; Right: the solution of DPIR-N with uniform $N_k = 60$.

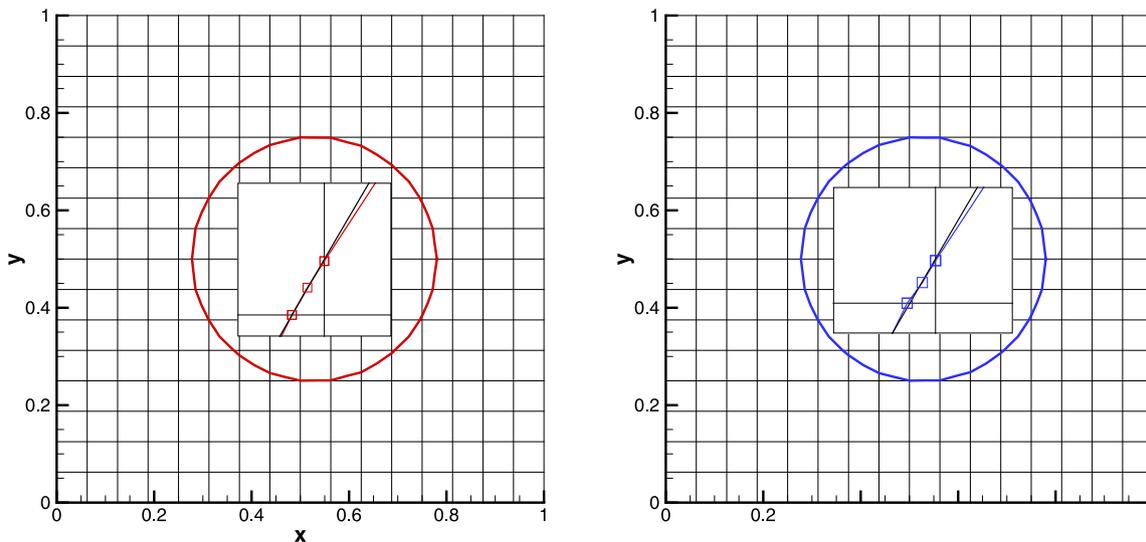


Fig. 19. Left: the solutions of DPIR-N with uniform $N_k = 260$; Right: the solutions of DPIR-N with new discretization method of $N_k = 60$.

4. Numerical examples

In this section, we first discuss the benefits of our new target function and the advantages of the new correction method. Then, we present several tests to show the superiority of the new method. For simplicity, the original DPIR method with target function (1) is denoted as DPIR-O. The ones with penalty terms (3) and (4) are denoted as DPIR-P1 and DPIR-P2, respectively. And the improved one in this work is called DPIR-N.

4.1. The benefits of the new target function

In this subsection, several cases are used to show the advantages of the new target function, and for easy of comparison between different target functions, the simple correction method of [1] is implemented.

4.1.1. Case 1 circle

First, we test the behavior of these functions on a circle interface (radius=1, center=(2.5,2.5)) with a uniform mesh. Figs. 5–7 show the results. As mentioned above, the results of DPIR-O are non-uniqueness, and both the red and blue interfaces can be the

optimal solution (Fig. 5). And, Fig. 6 shows that the parameter λ in the first penalty term has a great influence on DPIR-P1. Although the second penalty term limits its influence, the solutions of DPIR-P2 are virtually the same as the ones of DPIR-O (For simplicity, only one of its optimal solutions is plotted). As shown in Fig. 7, the optimal solution of our new target function is unique, and its correction interface is much better than the others.

To show the drawback of the penalty term (3), which has a different dimension unit with the others, Fig. 8 compares the solutions of DPIR-P1 with $\lambda = 0.01$ on the original case and a 10 times amplified case (a circle interface with radius=10, center=(25,25)). It can be seen that the two solutions do not maintain the principle of dynamic similarity, i.e., this method will generate different solutions if different dimensionless quantities (for instance, the unit of length, millimeter or centimeter) are used for a fluid dynamics problem. The behavior of the new DPIR-N method for this magnified problem is virtually the same as the original ones in Fig. 7, hence the results are not shown.

4.1.2. Case 2 square

The square interface (rectangular (-1.95,0) - (0,-1.95) - (1.95,0) - (0,1.95)) with a uniform grid is used. The solutions are shown

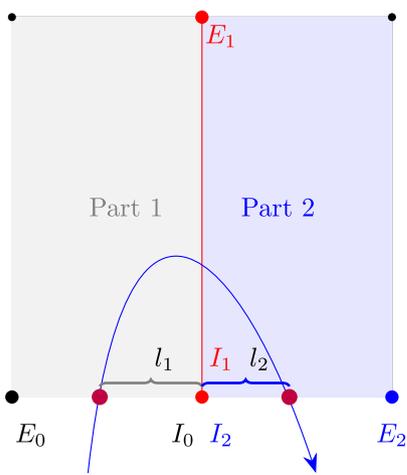


Fig. 20. Split the filament cell.

in Figs. 9 - 11. As Figs. 9 and 10 illustrate, the first penalty term (3) can reduce the ‘wave effects’, and the result of $\lambda = 0.5$ (the green one) is very close to the exact square interface. The same as the previous case, the second penalty term diminished the impact of the first one.

As shown in Fig. 11, once the reference interface in a mixed cell is linear, it will be reconstructed exactly, hence, the optimized solution of DPIR-N can reconstruct those mixed cells quite well, after implementing the simple correction step, its solutions preserve the square pretty well and free of oscillation.

Since it is quite difficult to decide a perfect value of λ for complex problems, $\lambda = 0.01$ will be used for all the cases below, as recommended in [1]. The DPIR-O will not be studied below. The behavior of DPIR-P2 is quite similar to DPIR-P1, hence it will not be discussed in the following too.

4.1.3. Case 3 circle

The circle interface (radius=1 and center=(2.55,2.55)) on a uniform mesh is used to further test the new function’s behavior on mixed cells that contain large fractions of internal material, i.e., $vol_I \gg vol_E$. As shown in Fig. 12, the DPIR-P1 method generates a spurious interface in this mixed cell. For comparison, we give the result of DPIR-N1 in Fig. 13. In the DPIR-N1 method, we only

use the internal material for all the mixed cells in Eq. (6), i.e., $(\bar{x}_c^*(\bar{M}_i, \bar{M}_{i+1}), \bar{x}_c^*, A_r, vol_r)$ is computed only by the internal material. It can be found out that, with the role of $1/vol_r$ in (6), the use of a small fraction can raise the corresponding mixed cell’s weight in the whole target function, and obtain a smaller remaining error in the optimized step for this mixed cell. And hence, the DPIR-N can get a smoother interface than the DPIR-N1.

4.1.4. Case 4 combined

The combined interface on a unit uniform mesh is tested to show the advantage of the power parameter in (6). The initial interface and solution of DPIR-P1 are plotted in Fig. 14. The solutions of DPIR-N2(the DPIR-N with power parameter $p = 1$) and DPIR-N are presented in Fig. 15. As we can see, with a parameter $p = 1/5$, it can reduce the influence of the trouble cell marked in gray on its neighbor cells marked in yellow. That is because, with the power function p , the value of the target function f_i of this trouble cell will be much larger than its neighbor’s, and hence it can decrease its influence during the optimized step. As we can see, the DPIR-N performs better and free of oscillation near this trouble cell.

4.2. The benefits of the new correction method

In this subsection, the combined interface in Case 4 is also used to show the benefits of the new correction method. The solutions are shown in Fig. 16. It can be seen that with the new correction method, the DPIR-N can resolve the trouble cell marked in gray much better.

4.3. Implementation of the DPIR method

A new discretization to reduce the computational cost and a strategy to handle the filament cells are introduced.

4.3.1. The new discretization method

As pointed out in [1], the complexity of the dynamic programming part of a DPIR method is $O(N \cdot N_k^3)$. Hence, a large value of N_k in Eq. (2) is much expensive. However, to solve the cell that contains a pretty large or small fraction of internal material, if a uniform discretization method is used, a large value of N_k will be needed to obtain a smooth interface in the correction step. To reduce the computational cost, we suggest using a small value of N_k and only refined the discretization points near the internal and

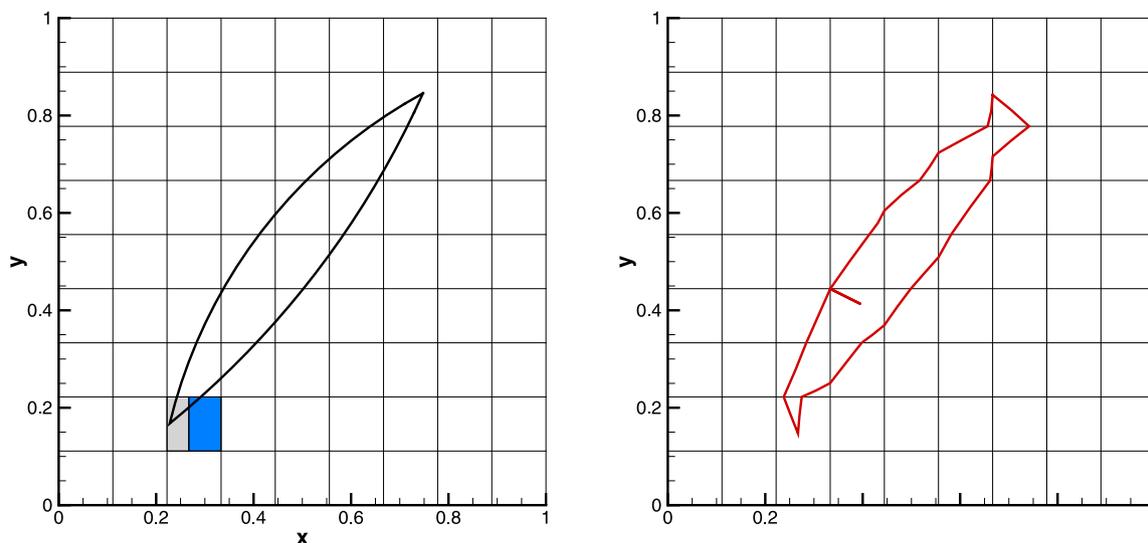


Fig. 21. Left: the initial mesh and interface; Right: the solution of DPIR-P1.

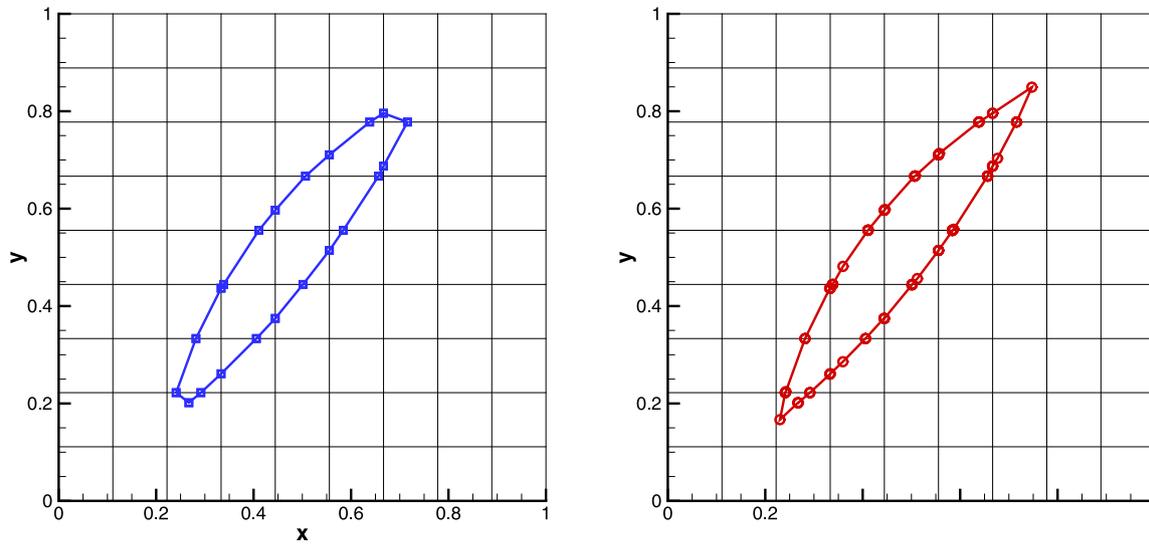


Fig. 22. The solution of DPIR-N. Left: the optimized interface in the first step; Right: the corrected interface in the second step.

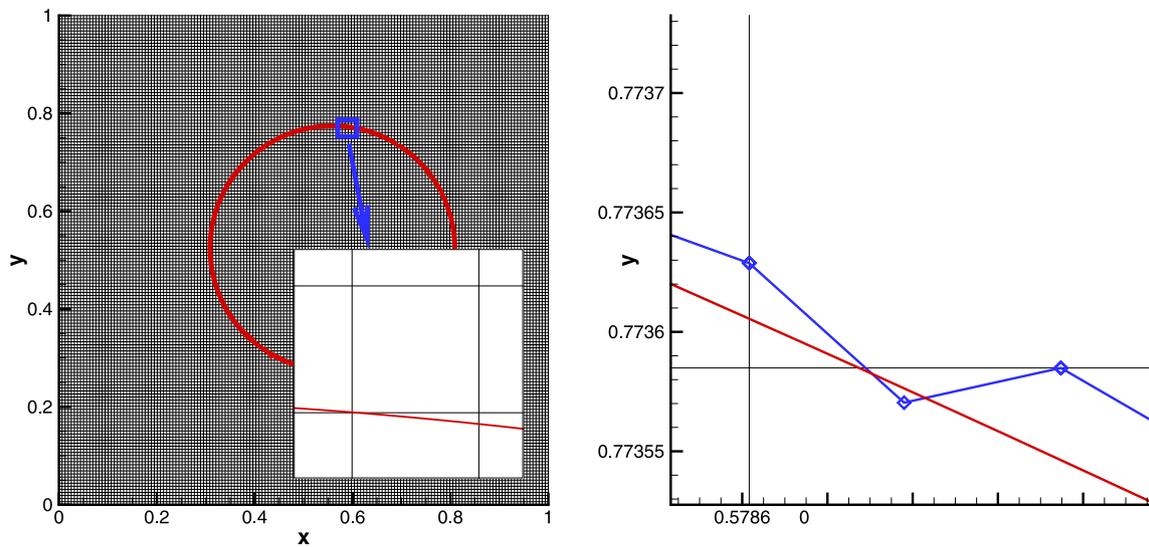


Fig. 23. Left: the initial mesh and the interface; Right: the correction interface of DPIR-P1 in the mixed cell.

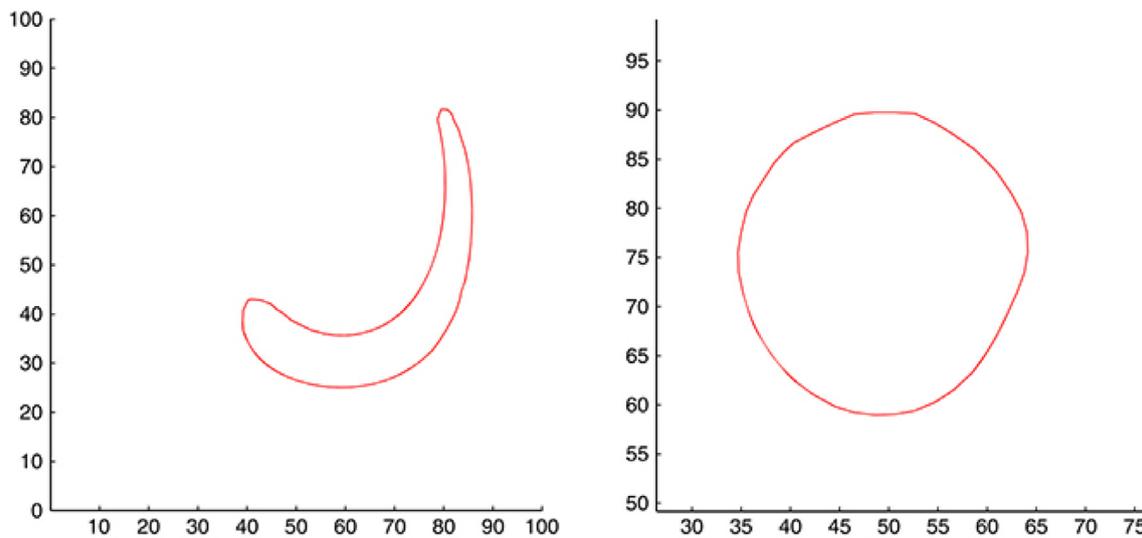


Fig. 24. The solutions of DPIR-P1 obtained by Dumas et al. [1]. Left: $T = 1$; Right: $T = 2$.

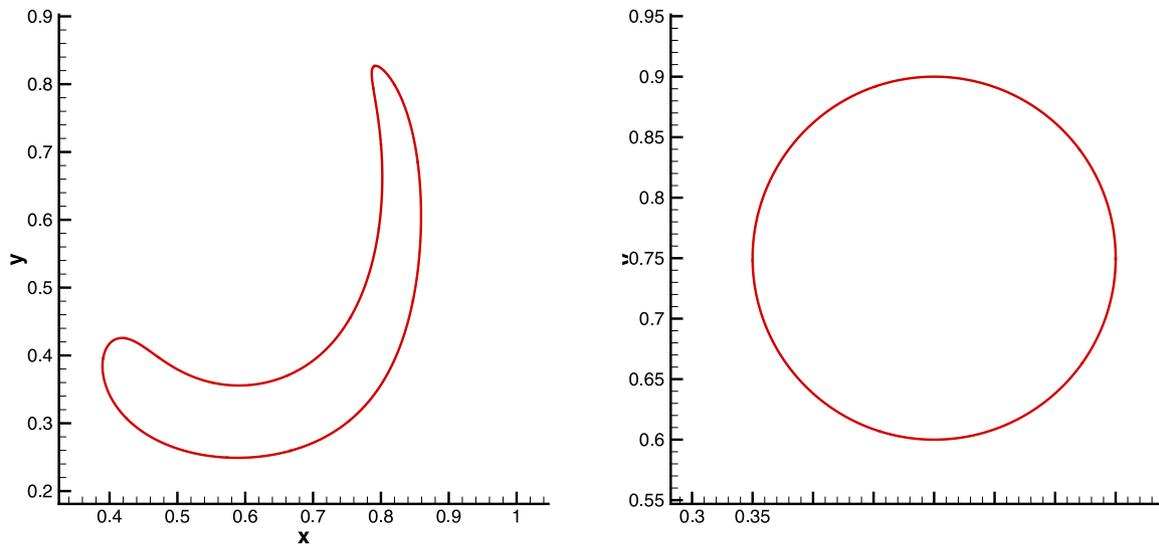


Fig. 25. The reference solutions. Left: $T = 1$; Right: $T = 2$.

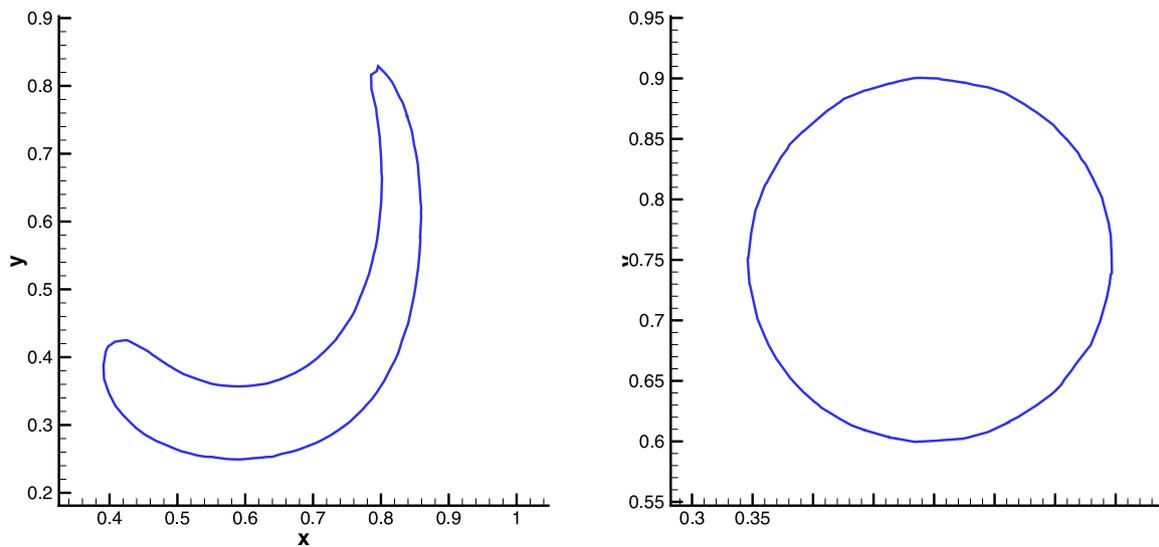


Fig. 26. The solutions of DPIR-N. Left: $T = 1$; Right: $T = 2$.

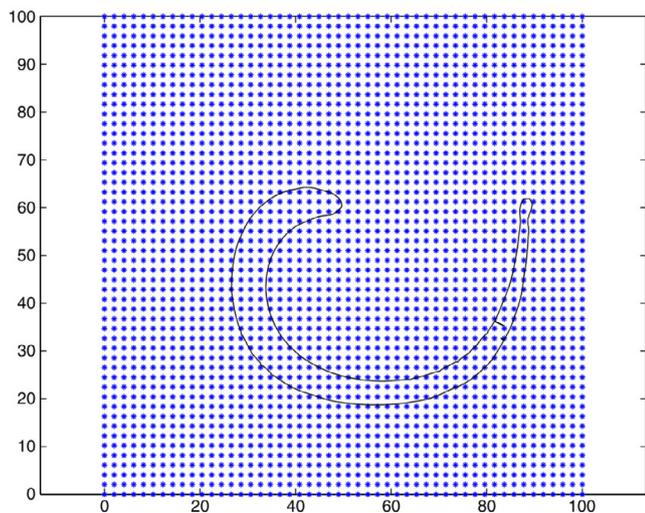


Fig. 27. The solution at $T = 1.011$ of DPIR-P1 obtained by Dumas et al. [1].

Table 1

CPU time with different N_k for DPIR-N.

N_k	CPU time
60	4.7176E-2
260	6.8108

external points (I_i and E_i) to obtain the smooth interface for cells with large or small fractions. See Fig. 17, we first discretize $x(2)$ into n_2 uniform parts, then divide segments $[I_i, x^1]$ and $[x^{n_2}, E_i]$ into n_1 parts.

As shown in Figs. 18 and 19, a circle interface (radius=0.25 and center=(0.5+1/35,0.5)) on a unit uniform mesh contains a mixed cell which has a very small fraction of internal material. This mixed cell may need a uniform discretization of $N_k = 260$ to achieve a smooth correction interface. While, with the new discretization method, $N_k = 60$ ($n_1 = 10$ and $n_2 = 40$) is sufficient. As shown in Table 1, with a small value of N_k it can save much time, and hence this new discretization method is used for all the DPIR methods below.

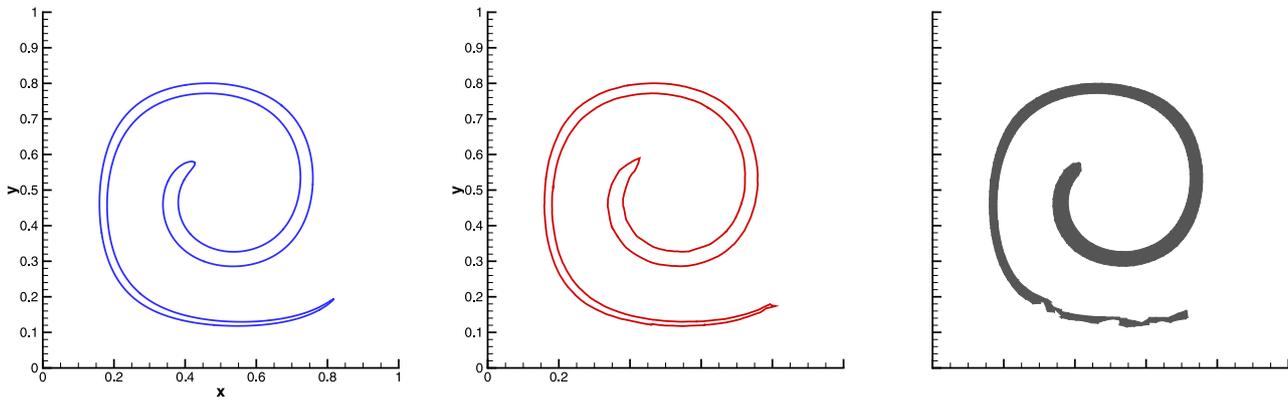


Fig. 28. The solutions at $T = 2$ with mesh of 50×50 . Left: tracking technique; Middle: DPIR-N; Right: MoF.

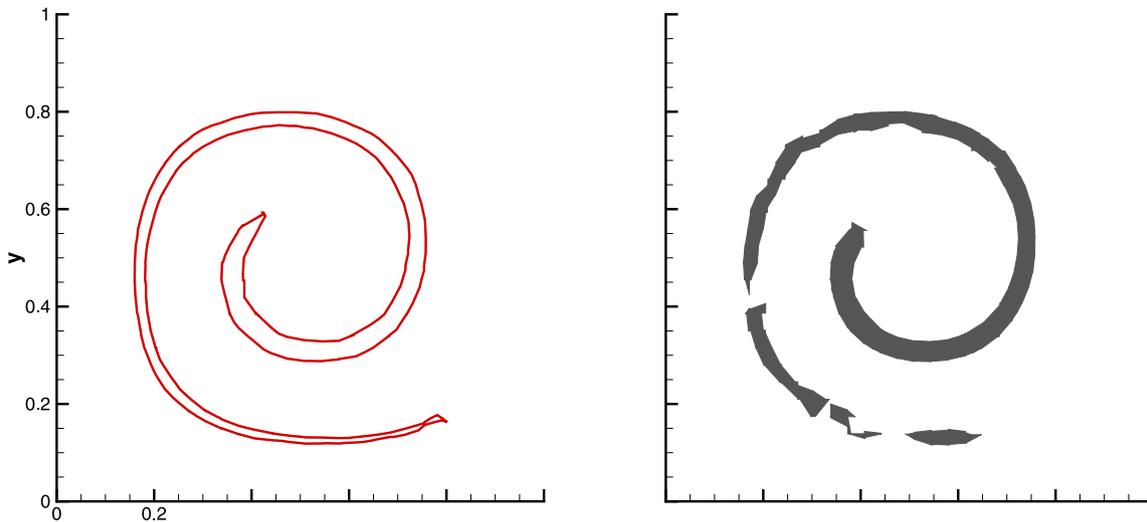


Fig. 29. The solutions at $T = 2$ with mesh of 30×30 . Left: DPIR-N; Right: MoF.

4.3.2. Strategy for filament cell

It is easy to know that, for the current DPIR method, it is not easy to set the external and internal points for those filament cells, in which the interface crosses the same cell boundary twice. Dumas et al. [1] suggest using the local refinement technique or relax the continuity property to overcome this difficulty. In this paper, instead of refining the filament cell, we split it into two parts to set the external and internal points. As shown in Fig. 20, the filament cell is split into two parts by adding the red line (for example, set $l_1 = l_2$), and then we can set the external and internal points for both parts.

A particular interface (Fig. 21) containing a filament cell is designed to test the DPIR-N method. The solutions are shown in Fig. 22. As we can see, the filament cell is divided into two parts (marked in gray and blue, respectively), and the DPIR-N method can solve this interface well.

4.4. Accuracy test

In the following tests, the DPIR-P1 method [1] with (3) and $\lambda = 0.01$ and our DPIR-N method with (6) are tested. The improved discretization with $N_k = 60$ is used. The original simple correction method is used for DPIR-P1, and the new local correction method is used for our DPIR-N.

Same as in [3], a circle interface (radius=0.25, center=(0.5+1/17, 0.5+1/41)) on a uniform grids ($[0, 1] \times [0, 1]$) is used to test the accuracy of these methods. The way to quantify the interface recon-

Table 2
Errors of DPIR-P1 and DPIR-N.

Mesh	DPIR-P1		DPIR-N	
	total	maximum	total	maximum
10×10	1.711E-3	2.065E-4	9.393E-4	1.008E-4
20×20	6.235E-4	3.333E-5	2.237E-4	1.556E-5
40×40	1.583E-4	6.718E-6	5.483E-5	1.841E-6
80×80	6.268E-5	1.519E-6	2.225E-5	4.748E-7
160×160	-	-	9.239E-6	3.126E-8

struction error is given in Section 4 of [3]. The total errors of the mixed cells and the maximum ones are shown in Table 2. As we can see, the errors of DPIR-N is much smaller. The DPIR-P1 method obtained a correction point out of the mixed cell with a mesh of 160×160 (See Fig. 23), and hence no results can be obtained.

4.5. Dynamic test

In this section, we tested the performance of the new method with a Lagrange+Remap technique. And we also compared our solutions with those obtained by Dumas et al. [1] in a similar condition.

4.5.1. Test1: Reversible interface

As in [1], the advection of a disc in a single vortex field is tested. It is a circle centered at (0.5,0.75) with radius=0.15 and ad-

vects with a velocity field:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\sin^2(\pi x)\sin(2\pi y) \\ \sin^2(\pi y)\sin(2\pi x) \end{pmatrix} \cos\left(\frac{\pi t}{2}\right) \quad (8)$$

on a unit uniform mesh of 50×50 .

The solutions of DPIR-P1 at $T = 1$ and $T = 2$ from Fig. 14 in [1] are shown in Fig. 24. The interfaces at $T = 1$ and $T = 2$ obtained with tracking technique [3] are given as the reference solutions (see Fig. 25), and the solutions of DPIR-N are plotted in Fig. 26. As we can see, both DPIR methods preserve the shape well.

4.5.2. Test2: Oneway interface

In the oneway test, the circle is advected with a velocity field:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \sin^2(\pi x)\sin(2\pi y) \\ -\sin^2(\pi y)\sin(2\pi x) \end{pmatrix} \quad (9)$$

on a unit domain. First, as in [1], a uniform mesh of 50×50 is used. The solutions of DPIR-P1 at $T = 1.011$ from Fig. 15 in [1] are shown in Fig. 27. The solution of DPIR-N at $T = 2$ is plotted in Fig. 28. And as a comparison, the solutions of MoF [3] and the tracking technique are also given. As mentioned in [1], the DPIR-P1 method failed near $T = 1.011$ and no results can be obtained after then. While, our DPIR-N still preserves a continuity interface and remains in one block after $T = 2$.

The solutions at $T = 2$ with a coarse mesh of 30×30 of DPIR-N and MoF are also given in Fig. 29. It can be seen that, even with a coarse grid, DPIR-N still preserves the continuity property well, but the interface of MoF has already separated.

5. Conclusion remarks

The continuous interfaces obtained by the DPIR methods developed in [1,12] suffer from 'wave effects'. That is because the target functions do not take into account the effect of the interface normal, resulting in multiple results in the optimized step. To overcome this problem, in this paper, we first design a non-dimensional target function to suppress the oscillation of the interface based on the objective function of the MoF method. Then we propose a new local correction method to improve its accuracy and robustness. The numerical results showed that the new continuity-

preserving DPIR method behaves better than the original one and can suppress the wave effects well.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research work was supported by the National Natural Science Foundation of China under Grants 11205016, 11671048, 11771055 and by CAEP under Grant 2014B0202031.

References

- [1] Dumas L, Ghidaglia J, Jaisson P, Motte R. A new volume-preserving and continuous interface reconstruction method for 2D multi-material flow. *Int J Numer Methods Fluids* 2017;85:48–66.
- [2] Benson DJ. Volume of fluid interface reconstruction methods for multi-material problems. *Appl Mech Rev* 2002;55:151–65.
- [3] Dyadechko V, Shashkov M. Moment-of-Fluid interface reconstruction. Tech Rep LA-UR-07-1537, Los Alamos National Laboratory, Los Alamos, October 2008.
- [4] Barth T, Jespersen D. The design and application of upwind schemes on unstructured meshes. In Proceedings of the 27th Aeroscience Meeting, Reno, NV, Jan 1989 AIAA.
- [5] Chen X, Zhang X. An improved 2D MoF method by using high order derivatives. *J Comput Phys* 2017;349:176–90.
- [6] Youngs D. Time-dependent multi-material flow with large fluid distortion. In KW Morton and MJ Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 273–285 Academic Press 1982.
- [7] Youngs D. An interface tracking method for a 3D eulerian hydrodynamics code. Technical Report 44/92/35, AWRE 1984.
- [8] Pilliod J, Puckett E. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J Comput Phys* 2004;199:465–502.
- [9] Swartz B. The second-order sharpening of blurred smooth borders. *Math Comput* 1989;52:675–714.
- [10] DeBar R. Fundamentals of the KRAKEN code. Technical Report UCID-17366, Lawrence Livermore National Laboratory 1974.
- [11] McMaster W. Computer codes for fluid-structure interactions. Technical Report UCRL-89724, Lawrence Livermore National Laboratory 1984.
- [12] Chollet I, Lissoni G, Corot T, Hoch P, et al TL. Curved interface reconstruction for 2D compressible multi-material flows. *ESAIM: Proceedings and Surveys*, EDP Sciences 2019. 1–10 hal-02496829
- [13] Liu S, Shen Y. Discontinuity-detecting method for a four-point stencil and its application to develop a third-order hybrid-WENO scheme. *J Sci Comput* 2019;81:1732–66.