



A High-Order Maximum-Principle-Satisfying Discontinuous Galerkin Method for the Level Set Problem

Fan Zhang¹ · Tiegang Liu² · Moubin Liu¹

Received: 13 May 2020 / Revised: 17 December 2020 / Accepted: 4 March 2021 / Published online: 19 March 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Level set (LS) method is a widely used interface capturing method. In the simulations of incompressible two-phase flows, in order to avoid discontinuities at interfaces, the LS function is usually taken as a smeared-out Heaviside function bounded on $[0, 1]$ and advected by a given velocity field \mathbf{u} obtained from the solution of the incompressible Navier-Stokes equations. In the incompressible limit $\nabla \cdot \mathbf{u} = 0$, the advection equation for the LS function can be written and discretized in conservative form. However, due to numerical errors, the resulting velocity field is in general not divergence free which leads to the solution of the advection equation in conservative form does not satisfy the maximum principle. To overcome this issue, in this work, we develop a high-order discontinuous Galerkin (DG) method to directly solve the advection equation for the LS function in non-conservative form. Moreover, we prove that by applying a linear scaling limiter, the proposed method together with a strong stability preserving (SSP) time discretization scheme can satisfy the strict maximum principle under a suitable CFL condition. Numerical simulations of several well-known benchmark problems, including the application to incompressible two-phase flows, are presented to demonstrate the high-order accuracy and maximum-principle-satisfying property of the proposed method.

Keywords Maximum principle · Discontinuous Galerkin method · Level set method · Incompressible two-phase flows

1 Introduction

Incompressible two-phase flows exist in a wide variety of natural processes and industrial applications, e.g., geophysical flows, water waves, biomechanics and many others. Numerical simulations of such flows are difficult because the interface separating different fluids must be accurately tracked or captured simultaneously with the evolution of the flow field [1,2]. Many methods have been developed for this purpose in literature. Among them, the level set (LS) method enjoys considerable popularity for its simplicity and flexibility [3–5]. In order

✉ Moubin Liu
mbliu@pku.edu.cn

¹ BIC-ESAT, College of Engineering, Peking University, Beijing 100871, P.R. China

² School of Mathematics, Beihang University, Beijing 100191, P.R. China

to avoid potential numerical oscillations around discontinuities, a smeared-out Heaviside function which is bounded on $[0, 1]$ is usually chosen as the LS function [6–8]. Under this case, the LS function takes zero in one fluid and one in the other, and it varies smoothly over the interface.

The LS method includes an advection equation that describes the evolution of the interface and a re-initialization equation that retains the profile and thickness of the interface. Numerical methods which are able to solve the advection equation with minimum numerical dissipation and dispersion for a nearly discontinuous solution are ideal candidates for the LS method. In this work, the discontinuous Galerkin (DG) method [9–11] is employed for the spatial discretization of the LS equation. However, it is well known that using high-order approximations in the DG discretization is prone to monotonicity violations. While small oscillations might be acceptable in some cases, in many others they can lead to unphysical values, e.g., values of the LS function outside of $[0, 1]$ may produce negative density in the context of incompressible two-phase simulations with large density ratio. Therefore, a high-order accurate scheme satisfying a strict maximum principle [12] in the sense that the numerical solution never goes out of the admissible set should be used in the computing.

In the simulations of incompressible two-phase flows, the LS function is advected by the velocity field \mathbf{u} obtained from the incompressible Navier-Stokes (INS) equations. In the incompressible limit, i.e., $\nabla \cdot \mathbf{u} = 0$, the advection equation for the LS function can be written in conservative form. Various maximum-principle-satisfying (MPS) numerical methods have been developed for solving such conservation laws. One important breakthrough was made by Zhang and Shu [13, 14] who proposed a uniformly high-order accurate MPS DG and weighted essentially non-oscillatory (WENO) schemes for scalar conservation laws. These high-order schemes achieve the strict maximum principle by applying a linear scaling limiter [15] at each stage of an explicit Runge-Kutta (RK) method or at each step of a multistep method. This technique was later generalized to positivity preserving high-order DG method for compressible Euler equations [16]. Another class of high-order parametrized maximum-principle-preserving (MPP) flux limiters was proposed by Xu *et al.* [17, 18] under a finite volume framework, which limits a high-order numerical flux towards a first-order monotone flux. In [19], Xiong *et al.* generalized the parametrized high-order MPP flux limiters for finite difference RK-WENO schemes with applications in inviscid incompressible flows. Extensions of the high-order MPS or MPP methods for solving convection-diffusion equations were also considered by many researchers, see for example [20–24].

For solutions containing discontinuities, in general, it is preferable to solve the conservative form of the advection equation rather than the non-conservative one. However, due to numerical errors, the velocity field obtained by solving the INS equations will slightly deviate from the divergence-free field. Under this case, the advection equation in conservation form does not imply the maximum principle [13] and this is the main difficulty in developing a high-order MPS method for solving the incompressible two-phase flows. In order to maintain a divergence-free velocity field, in [13, 14], a high-order DG method was developed for solving two-dimensional incompressible flows in the vorticity stream-function formulation [25].

The main object of this work is to construct a high-order MPS DG method for solving the LS problem in a given non-divergence-free velocity field such that the numerical solution never goes out of the range of the admissible set. First, by carefully constructing the numerical flux, we develop a high-order DG method, termed as non-conservative DG method, to directly solve the advection equation for the LS function in non-conservative form. Then, following the idea in [13, 14], we prove that by applying a linear scaling limiter, this non-conservative DG method together with a strong stability preserving (SSP) time discretization scheme can

satisfy the maximum principle under a suitable CFL condition. Generally, it needs to consider the following two steps:

1. To maintain the solution average in $[0, 1]$ during the temporal evolution;
2. To maintain the whole polynomial solution in $[0, 1]$ without losing accuracy.

The main challenge lies in retaining the cell averages in the next time step bounded in $[0, 1]$. As for the second step, the linear scaling limiter introduced in [15] can be applied to control the maximum/minimum of the original polynomial solutions. Numerical simulations of several well-known benchmark problems, including the application to incompressible two-phase flows, are presented to demonstrate the high-order accuracy and MPS property of the proposed non-conservative DG method.

This paper is organized as follows. Section 2 gives a brief introduction of the LS method. In Sect. 3, we develop a high-order DG method for solving the advection equation in non-conservative form and prove that the proposed method satisfies the maximum principle with arbitrary order accuracy. The high-order MPS DG method for solving the re-initialization equation is given in Sect. 4. Numerical tests for the proposed method, including examples from the incompressible two-phase flows, are shown in Sect. 5. Concluding remarks are given in Sect. 6.

2 Level Set Method

The LS method has been first developed by Osher and Sethian for computing the motion of two-phase flow [3]. In the standard LS method, the interface $\phi(\mathbf{x}, t)$ is defined to be a signed distance function, i.e.,

$$|\phi(\mathbf{x}, t)| = d(\mathbf{x}, t) = \min_{\mathbf{x}_I \in \Gamma(t)} |\mathbf{x} - \mathbf{x}_I|, \tag{1}$$

where t is time, $\Gamma(t)$ is the interface and \mathbf{x}_I is the location on the interface that is closest to the point \mathbf{x} . In practice, however, in order to avoid discontinuities at interface, the signed distance function $\phi(\mathbf{x}, t)$ is usually replaced with a smeared-out Heaviside function $\varphi(\mathbf{x}, t)$ defined as follows

$$\varphi(\mathbf{x}, t) = \frac{1}{2} \left(\tanh\left(\frac{\phi(\mathbf{x}, t)}{2\varepsilon}\right) + 1 \right), \tag{2}$$

where ε is a parameter that represents the thickness of the profile. Then, $\varphi(\mathbf{x}, t)$ takes the value 0 on one side of the interface and the value 1 on the other. The interface $\Gamma(t)$ is defined by the location of the $\varphi(\mathbf{x}, t) = 0.5$ iso-surface, i.e.,

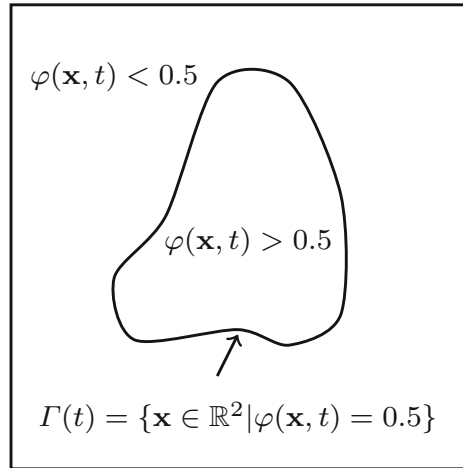
$$\Gamma(t) = \{\mathbf{x} | \varphi(\mathbf{x}, t) = 0.5\}. \tag{3}$$

Consider a moving interface $\Gamma(t)$ in two dimensional that bounds a region $\Omega \in \mathbb{R}^2$, see Fig. 1. Motion of the interface is achieved by solving the following advection equation

$$\begin{aligned} \frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi &= 0, \quad \forall (\mathbf{x}, t) \in \Omega \times [0, T], \\ \varphi(\mathbf{x}, 0) &= \varphi_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \end{aligned} \tag{4}$$

where $0 < T < \mathbb{R}$ is the final time, $\varphi : \Omega \times [0, T] \rightarrow \mathbb{R}$ is the LS function, $\mathbf{u} = (u, v) : \Omega \times [0, T] \rightarrow \mathbb{R}^2$ is the velocity field and $\varphi_0 : \Omega \rightarrow \mathbb{R}$ is the initial condition. Due to the existence of inevitable numerical errors or artificial diffusions together with velocity

Fig. 1 Explanation of the level set method



variations, the shape of $\varphi(\mathbf{x}, t)$ across the interface will be distorted when $\varphi(\mathbf{x}, t)$ is advected. As a result, a re-initialization step is necessary to retain the profile and thickness of the interface. This is performed by solving the following advection-diffusion equation [6]

$$\frac{\partial \varphi}{\partial \tau} + \nabla \cdot (\varphi(1 - \varphi)\widehat{\mathbf{n}}|_{\tau=0}) = \nabla \cdot (\varepsilon \nabla \varphi). \tag{5}$$

Here, τ is the pseudo-time and $\widehat{\mathbf{n}} = (\widehat{n}_x, \widehat{n}_y) = \nabla \varphi / \|\nabla \varphi\|$ is the interface normal vector. The term $\varphi(1 - \varphi)\widehat{\mathbf{n}}|_{\tau=0}$ is the compressive flux that aims at sharpening the profile, while $\varepsilon \nabla \varphi$ is the diffusion flux that maintains characteristic thickness ε and avoids discontinuities at the interface.

3 Maximum-Principle-Satisfying DG Method for the Advection Equation

3.1 Preliminaries

Before introducing the MPS DG method for the advection equation, we first introduce some notations that will help us to obtain the primal formulation. Let Ω_h be an approximation of the solution domain Ω and for simplicity we partition Ω_h by uniform rectangular cells such that $\Omega_h = \cup_{i=1}^{N_x} \cup_{j=1}^{N_y} I_{i,j}$ with $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, $\Delta x = (x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})$, $\Delta y = (y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})$ and $|I_{i,j}| = \Delta x \Delta y$. We define $\partial I_{i,j}$ as the cell interfaces of $I_{i,j}$ and we permanently associate it with a unit normal vector \mathbf{n}_e outward from the cell $I_{i,j}$.

We seek numerical solutions in the discontinuous piecewise polynomial space given as

$$V_h^k \stackrel{def}{=} \{\psi \in L^2(\Omega) : \psi|_{I_{i,j}} \in P^k(I_{i,j}), \forall I_{i,j} \in \Omega_h\}, \tag{6}$$

where P^k is the set of two-variable polynomials of degree equal or less than k . At each cell $I_{i,j}$, we have an approximate polynomial solution $\varphi_h(x, y) \in P^k$ with the cell average $\bar{\varphi}_{i,j}$ defined as

$$\bar{\varphi}_{i,j} = \frac{1}{|I_{i,j}|} \int_{I_{i,j}} \varphi_h(x, y) d\mathbf{x}. \tag{7}$$

Note that the functions in V_h^k can be double-valued on the cell interfaces $\partial I_{i,j}$. Hence, we denote φ_h^- and φ_h^+ as the left and right limits of φ_h , respectively, and denote the traces of φ_h on the four edges of $I_{i,j}$ as $\varphi_{i-\frac{1}{2},j}^+(y)$, $\varphi_{i+\frac{1}{2},j}^-(y)$, $\varphi_{i,j-\frac{1}{2}}^+(x)$ and $\varphi_{i,j+\frac{1}{2}}^-(x)$, respectively, as illustrated in Fig. 2(a). Then, all of the traces are single-variable polynomials of degree k .

The integrals in the DG formulation are approximated by quadratures with sufficient accuracy. In this work, we need to use two quadrature rules, namely the L -point Gauss-Legendre quadrature rule and the N -point Gauss-Lobatto quadrature rule. Specifically, the Gauss-Legendre quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ are denoted as

$$S_i^x = \{x_i^\alpha : \alpha = 1, \dots, L\}, \quad S_j^y = \{y_j^\alpha : \alpha = 1, \dots, L\}, \tag{8}$$

respectively, with quadrature weights w_α on interval $[-\frac{1}{2}, \frac{1}{2}]$ satisfying $\sum_{\alpha=1}^L w_\alpha = 1$. While the Gauss-Lobatto quadrature points are denoted as

$$\widehat{S}_i^x = \{\widehat{x}_i^\beta : \beta = 1, \dots, N\}, \quad \widehat{S}_j^y = \{\widehat{y}_j^\beta : \beta = 1, \dots, N\}, \tag{9}$$

with quadrature weights \widehat{w}_β on interval $[-\frac{1}{2}, \frac{1}{2}]$ satisfying $\sum_{\beta=1}^N \widehat{w}_\beta = 1$. For example, for $k = 2$, we use a three-point Gauss quadrature, i.e., $S^x = S^y = \{-\frac{\sqrt{15}}{10}, 0, \frac{\sqrt{15}}{10}\}$ with $w = \{\frac{5}{18}, \frac{8}{18}, \frac{5}{18}\}$ and $\widehat{S}^x = \widehat{S}^y = \{-\frac{1}{2}, 0, \frac{1}{2}\}$ with $\widehat{w} = \{\frac{1}{6}, \frac{4}{6}, \frac{1}{6}\}$. Then, by introducing the tensor product \otimes and defining the following points set $S_{i,j}$ [13]

$$S_{i,j} = (S_i^x \otimes \widehat{S}_j^y) \cup (\widehat{S}_i^x \otimes S_j^y), \tag{10}$$

see Fig. 2(b) for an illustration, the cell average $\bar{\varphi}_{i,j}$ can be approximated by

$$\bar{\varphi}_{i,j} = \sum_{\alpha=1}^L \sum_{\beta=1}^N w_\alpha \widehat{w}_\beta \varphi_h(x_i^\alpha, \widehat{y}_j^\beta). \tag{11}$$

Similarly, we also have

$$\bar{\varphi}_{i,j} = \sum_{\alpha=1}^L \sum_{\beta=1}^N w_\alpha \widehat{w}_\beta \varphi_h(\widehat{x}_i^\beta, y_j^\alpha). \tag{12}$$

In the following, unless otherwise specified, subscripts α and β, γ will be used for Gauss-Legendre and Gauss-Lobatto quadrature points, respectively.

3.2 Advection Equation with a Divergence-Free Velocity Field

In the context of a divergence-free velocity field, i.e., $\nabla \cdot \mathbf{u} = 0$, the advection equation for the LS function Eq. (4) can be written in conservative form

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\mathbf{u}\varphi) = 0. \tag{13}$$

Under this case, the exact solution of Eq. (13) is equivalent to that of Eq. (4) which satisfies a strict maximum principle, i.e., if

$$m = \min_{\mathbf{x}} \varphi_0(\mathbf{x}), \quad M = \max_{\mathbf{x}} \varphi_0(\mathbf{x}), \tag{14}$$

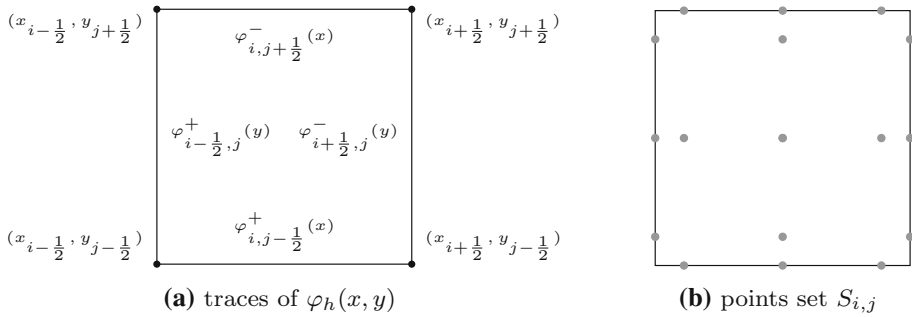


Fig. 2 The traces of solution and integral points in cell $I_{i,j}$

then $\varphi(\mathbf{x}, t) \in [m, M]$ for any \mathbf{x} and t . Here, $m = 0$ and $M = 1$ for the LS function Eq. (2). This property is also desired for numerical schemes solving Eq. (4) and Eq. (13), since the violation of this property can lead to instability and breakdown during the simulation [26].

Multiplying Eq. (13) by any test function $\psi \in V_h^k$, integrating over any cell $I_{i,j} \in \Omega_h$ and integrating by parts, we obtain the standard DG(P^k) method, i.e., we seek $\varphi_h \in V_h^k$ such that

$$\int_{I_{i,j}} (\partial_t \varphi_h) \psi \, d\mathbf{x} - \int_{I_{i,j}} \mathbf{F}_c(\varphi_h) \cdot \nabla \psi \, d\mathbf{x} + \int_{\partial I_{i,j}} \mathbf{n}_e \cdot \widehat{\mathbf{F}}_c(\varphi_h^-, \varphi_h^+) \psi \, ds = 0 \quad (15)$$

holds for any $\psi \in V_h^k$. Here, $\mathbf{F}_c(\varphi) \stackrel{def}{=} [f_c(\varphi), g_c(\varphi)] = \mathbf{u}\varphi$ and $\widehat{\mathbf{F}}_c(\cdot, \cdot)$ is any Lipschitz continuous monotone flux as defined in [27], e.g., the global Lax-Friedrichs flux defined by

$$\mathbf{n}_e \cdot \widehat{\mathbf{F}}_c(\varphi_h^-, \varphi_h^+) = \frac{1}{2} [\mathbf{n}_e \cdot (\mathbf{F}_c(\varphi_h^-) + \mathbf{F}_c(\varphi_h^+)) - a(\varphi_h^+ - \varphi_h^-)], \quad (16)$$

where $a = \max_{\mathbf{x}} |\mathbf{n}_e \cdot \mathbf{F}'_c(\varphi_h)| = \max_{\mathbf{x}} |\mathbf{n}_e \cdot \mathbf{u}_h|$ and \mathbf{u}_h is the given exact or approximate solution of the velocity field \mathbf{u} .

It is not natural for the DG method Eq. (15) to achieve a strict maximum principle in the sense that the numerical solution φ_h never goes out of the range $[m, M]$. To overcome this problem, Zhang and Shu [13,14] developed a uniformly high-order accurate DG method for the conservation laws satisfying a strict maximum principle which needs to consider the following two steps:

First, maintenance of the polynomial solution averages staying inside $[m, M]$. According to Eq. (15), the scheme satisfied by cell averages with the Euler forward time discretization can be written as

$$\begin{aligned} \overline{\varphi}_{i,j}^{n+1} &= \overline{\varphi}_{i,j}^n - \frac{\lambda_x}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \widehat{f}_c(\varphi_{i+\frac{1}{2},j}^-(y), \varphi_{i+\frac{1}{2},j}^+(y)) - \widehat{f}_c(\varphi_{i-\frac{1}{2},j}^-(y), \varphi_{i-\frac{1}{2},j}^+(y)) \, dy \\ &\quad - \frac{\lambda_y}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \widehat{g}_c(\varphi_{i,j+\frac{1}{2}}^-(x), \varphi_{i,j+\frac{1}{2}}^+(x)) - \widehat{g}_c(\varphi_{i,j-\frac{1}{2}}^-(x), \varphi_{i,j-\frac{1}{2}}^+(x)) \, dx, \end{aligned} \quad (17)$$

where the superscript n is the time level, $\lambda_x = \Delta t / \Delta x$, $\lambda_y = \Delta t / \Delta y$ and $\widehat{f}_c(\cdot, \cdot), \widehat{g}_c(\cdot, \cdot)$ are the one-dimensional numerical fluxes. Then, we have [13,14].

Lemma 1 Consider the scheme satisfied by the cell averages of the DG(P^k) method Eq. (17) for the advection equation in conservative form with a locally divergence-free velocity field.

If the approximate polynomials satisfy $\varphi_h^n(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$, then $\overline{\varphi}_{i,j}^{n+1} \in [m, M]$ under the CFL condition

$$a_x \lambda_x + a_y \lambda_y \leq \min\{\widehat{w}_1, \frac{1}{2k+1}\}. \tag{18}$$

Here, $a_x = \max |f'_c(\varphi)| = \|u\|_\infty$, $a_y = \max |g'_c(\varphi)| = \|v\|_\infty$ and $1/(2k+1)$ is the requirement of linear stability for the DG(P^k) method [10].

Remark 1 Notice that the establishment of Lemma 1 relies on the fact that the incompressibility $\nabla \cdot \mathbf{u}_h = 0$ holds everywhere inside the cell $I_{i,j}$, otherwise the conservative form Eq. (13) itself does not imply the maximum principle.

Second, maintenance of the whole polynomial solution staying inside $[m, M]$ without losing accuracy. To enforce this condition which is required by Lemma 1, we can employ a linear scaling limiter [15] to the original polynomial $\varphi_h^n(x, y)$ such that the modified polynomial $\widetilde{\varphi}_h^n(x, y) \in [m, M]$ for all $(x, y) \in S_{i,j}$, i.e.,

Lemma 2 For any cell $I_{i,j} \in \Omega_h$ with the cell average $\overline{\varphi}_{i,j}^n \in [m, M]$, by applying the following linear scaling limiter to the polynomial solution $\varphi_h^n(x, y)$

$$\widetilde{\varphi}_h^n(x, y) = \theta(\varphi_h^n(x, y) - \overline{\varphi}_{i,j}^n) + \overline{\varphi}_{i,j}^n, \quad \theta = \min\{|\frac{M - \overline{\varphi}_{i,j}^n}{M_{i,j} - \overline{\varphi}_{i,j}^n}|, |\frac{m - \overline{\varphi}_{i,j}^n}{m_{i,j} - \overline{\varphi}_{i,j}^n}|, 1\}, \tag{19}$$

where

$$M_{i,j} = \max_{(x,y) \in S_{i,j}} \varphi_h^n(x, y), \quad m_{i,j} = \min_{(x,y) \in S_{i,j}} \varphi_h^n(x, y), \tag{20}$$

then $\widetilde{\varphi}_h^n(x, y) \in [m, M]$ approximates $\varphi_h^n(x, y)$ with a $(k+1)$ -th order accuracy.

This completes the construction of the high-order MPS DG method for the advection equation in conservative form Eq. (13). Following a similar approach, we will develop a high-order MPS DG method for the advection equation in non-conservative form Eq. (4).

3.3 Advection Equation with a Non-Divergence-Free Velocity Field

In the simulation of incompressible two-phase flows, the velocity field is obtained by solving the INS equations. Although the exact solution of the INS equations is always divergence free, unfortunately, this is generally not strictly satisfied by the numerical solution due to the existence of numerical errors. In this situation, the advection equation in non-conservative form, i.e.,

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \tag{21}$$

is used and discretized in this work.

Similarly, multiplying Eq. (21) by any test function $\psi \in V_h^k$, integrating over any cell $I_{i,j} \in \Omega_h$ and then performing a formal integration by parts, we have

$$\int_{I_{i,j}} (\partial_t \varphi_h) \psi \, d\mathbf{x} - \int_{I_{i,j}} \varphi_h \nabla \cdot (\mathbf{u}_h \psi) \, d\mathbf{x} + \int_{\partial I_{i,j}} \mathbf{n}_e \cdot (\widehat{\mathbf{u}\varphi}) \psi \, ds = 0, \tag{22}$$

where $\mathbf{n}_e \cdot (\widehat{\mathbf{u}\varphi})$ is the numerical flux defined at the cell interfaces $\partial I_{i,j}$. In order to avoid computing the spatial derivatives with respect to \mathbf{u}_h , we perform an extra element-wise

integration by parts locally in $I_{i,j}$ to arrive at

$$\int_{I_{i,j}} (\partial_t \varphi_h) \psi \, d\mathbf{x} + \int_{I_{i,j}} (\mathbf{u}_h \cdot \nabla \varphi_h) \psi \, d\mathbf{x} - \int_{\partial I_{i,j}} \mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)} \psi \, ds + \int_{\partial I_{i,j}} \mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)} \psi \, ds = 0. \tag{23}$$

It should be noted that the second integration by parts is different from the first one since it is performed based on the local discretized DG polynomial solution within the cell $I_{i,j}$, i.e., no numerical flux is entailed which leads to

$$\mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)} = \begin{cases} \mathbf{n}_e \cdot (\mathbf{u}_h^- \varphi_h^-), & \text{in cell } I_L, \\ \mathbf{n}_e \cdot (\mathbf{u}_h^+ \varphi_h^+), & \text{in cell } I_R. \end{cases} \tag{24}$$

Here, I_L and I_R are the left and right cells of $\partial I_{i,j}$, respectively. Therefore, to fulfill the DG discretization Eq. (23), we only need to specify the formulation of the numerical flux $\mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)}$ which is crucial to the successful implementation of the DG method.

By taking $\varphi = 1$ in Eq. (23), it gives that the numerical flux $\mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)}$ should be coincide with $\mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)}$ [28,29]. Thus, we define the numerical flux as follows

$$\mathbf{n}_e \cdot \widehat{(\mathbf{u}\varphi)} \stackrel{def}{=} \mathbf{n}_e \cdot \widehat{\mathbf{F}}(\varphi_h^-, \varphi_h^+) = \begin{cases} \frac{1}{2} [(\mathbf{n}_e \cdot \mathbf{u}_h^-)(\varphi_h^- + \varphi_h^+) - a(\varphi_h^+ - \varphi_h^-)], & \text{in cell } I_L, \\ \frac{1}{2} [(\mathbf{n}_e \cdot \mathbf{u}_h^+)(\varphi_h^- + \varphi_h^+) - a(\varphi_h^+ - \varphi_h^-)], & \text{in cell } I_R. \end{cases} \tag{25}$$

Similarly, a is the global maximum value of $|\mathbf{n}_e \cdot \mathbf{u}_h|$. With this choice, the numerical flux Eq. (25) satisfies the following monotonicity property

$$\frac{\partial(\mathbf{n}_e \cdot \widehat{\mathbf{F}}(\varphi_h^-, \varphi_h^+))}{\partial \varphi_h^-} \geq 0, \quad \frac{\partial(\mathbf{n}_e \cdot \widehat{\mathbf{F}}(\varphi_h^-, \varphi_h^+))}{\partial \varphi_h^+} \leq 0. \tag{26}$$

Remark 2 When a continuous interfacial normal velocity is applied, the proposed method Eqs. (23–25) is coincided with the one given in [30] which was constructed for the Hamilton-Jacobi equation. Moreover, if the velocity field is also locally divergence-free, the proposed method is same as the standard DG method Eq. (15) associated with the flux Eq. (16).

Plugging Eq. (24) and Eq. (25) into Eq. (23), we can obtain the scheme satisfied by the cell averages of the DG(P^k) method with the Euler forward scheme

$$\begin{aligned} \bar{\varphi}_{i,j}^{n+1} &= \bar{\varphi}_{i,j}^n - \frac{\Delta t}{|I_{i,j}|} \int_{I_{i,j}} \mathbf{u}_h \cdot \nabla \varphi_h \, d\mathbf{x} \\ &+ \frac{\lambda_x}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} [u_{i+\frac{1}{2},j}^- \varphi_{i+\frac{1}{2},j}^-(y) - u_{i-\frac{1}{2},j}^+ \varphi_{i-\frac{1}{2},j}^+(y)] \\ &- [\widehat{f}(\varphi_{i+\frac{1}{2},j}^-(y), \varphi_{i+\frac{1}{2},j}^+(y)) - \widehat{f}(\varphi_{i-\frac{1}{2},j}^-(y), \varphi_{i-\frac{1}{2},j}^+(y))] \, dy \\ &+ \frac{\lambda_y}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} [v_{i,j+\frac{1}{2}}^- \varphi_{i,j+\frac{1}{2}}^-(x) - v_{i,j-\frac{1}{2}}^+ \varphi_{i,j-\frac{1}{2}}^+(x)] \\ &- [\widehat{g}(\varphi_{i,j+\frac{1}{2}}^-(x), \varphi_{i,j+\frac{1}{2}}^+(x)) - \widehat{g}(\varphi_{i,j-\frac{1}{2}}^-(x), \varphi_{i,j-\frac{1}{2}}^+(x))] \, dx. \end{aligned} \tag{27}$$

Here,

$$\widehat{f}(\cdot, \cdot) = (1, 0) \cdot \widehat{\mathbf{F}}(\cdot, \cdot), \quad \widehat{g}(\cdot, \cdot) = (0, 1) \cdot \widehat{\mathbf{F}}(\cdot, \cdot) \tag{28}$$

are the one-dimensional numerical fluxes in the x - and y - directions, respectively.

Then, for a piecewise-linear DG approximation, we have the following theorem

Theorem 1 Consider the scheme satisfied by the cell averages of the DG(P^1) method Eq. (27) for solving the advection equation in non-conservative form Eq. (21). If the approximate polynomials satisfy $\varphi_h^n(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$, then $\overline{\varphi}_{i,j}^{n+1} \in [m, M]$ under the CFL condition

$$a_x \lambda_x + a_y \lambda_y \leq \min\{\widehat{w}_1, \frac{1}{3}\}. \tag{29}$$

Here, $a_x = \max |f'(\varphi)| = \|u\|_\infty$ and $a_y = \max |g'(\varphi)| = \|v\|_\infty$.

Proof When using P^1 approximation, i.e., $k = 1$, the solution $\varphi_h^n(x, y)$ is a polynomial of degree no more than one with respect to its two variables. Thus, the surface integral in Eq. (27) can be expressed as

$$\int_{I_{i,j}} \mathbf{u}_h \cdot \nabla \varphi_h d\mathbf{x} = \int_{I_{i,j}} \left[u_h \frac{\varphi_{i+\frac{1}{2},j}^-(y) - \varphi_{i-\frac{1}{2},j}^+(y)}{\Delta x} + v_h \frac{\varphi_{i,j+\frac{1}{2}}^-(x) - \varphi_{i,j-\frac{1}{2}}^+(x)}{\Delta y} \right] d\mathbf{x}. \tag{30}$$

Substituting Eq. (30) into Eq. (27) and then using an L -point Gauss-Legendre quadrature rule and an N -point ($N = 2$) Gauss-Lobatto quadrature rule that are exact for polynomials of degree one, we have

$$\begin{aligned} \overline{\varphi}_{i,j}^{n+1} &= \gamma_x \widehat{w}_1 \sum_{\alpha=1}^L w_\alpha \left[\left(1 + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \bar{u}_{i,\alpha} - \frac{\lambda_x}{\gamma_x \widehat{w}_1} u_{i-\frac{1}{2},\alpha}^+ \right) \varphi_{i-\frac{1}{2},\alpha}^+ + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \widehat{f}(\varphi_{i-\frac{1}{2},\alpha}^-, \varphi_{i-\frac{1}{2},\alpha}^+) \right] \\ &+ \gamma_x \widehat{w}_2 \sum_{\alpha=1}^L w_\alpha \left[\left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_2} \bar{u}_{i,\alpha} + \frac{\lambda_x}{\gamma_x \widehat{w}_2} u_{i+\frac{1}{2},\alpha}^- \right) \varphi_{i+\frac{1}{2},\alpha}^- - \frac{\lambda_x}{\gamma_x \widehat{w}_2} \widehat{f}(\varphi_{i+\frac{1}{2},\alpha}^-, \varphi_{i+\frac{1}{2},\alpha}^+) \right] \\ &+ \gamma_y \widehat{w}_1 \sum_{\alpha=1}^L w_\alpha \left[\left(1 + \frac{\lambda_y}{\gamma_y \widehat{w}_1} \bar{v}_{\alpha,j} - \frac{\lambda_y}{\gamma_y \widehat{w}_1} v_{\alpha,j-\frac{1}{2}}^+ \right) \varphi_{\alpha,j-\frac{1}{2}}^+ + \frac{\lambda_y}{\gamma_y \widehat{w}_1} \widehat{g}(\varphi_{\alpha,j-\frac{1}{2}}^-, \varphi_{\alpha,j-\frac{1}{2}}^+) \right] \\ &+ \gamma_y \widehat{w}_2 \sum_{\alpha=1}^L w_\alpha \left[\left(1 - \frac{\lambda_y}{\gamma_y \widehat{w}_2} \bar{v}_{\alpha,j} + \frac{\lambda_y}{\gamma_y \widehat{w}_2} v_{\alpha,j+\frac{1}{2}}^- \right) \varphi_{\alpha,j+\frac{1}{2}}^- - \frac{\lambda_y}{\gamma_y \widehat{w}_2} \widehat{g}(\varphi_{\alpha,j+\frac{1}{2}}^-, \varphi_{\alpha,j+\frac{1}{2}}^+) \right]. \end{aligned} \tag{31}$$

Here, $\gamma_x = (a_x \lambda_x) / (a_x \lambda_x + a_y \lambda_y)$, $\gamma_y = (a_y \lambda_y) / (a_x \lambda_x + a_y \lambda_y)$,

$$\bar{u}_{i,\alpha} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_h(x, y_j^\alpha) dx, \quad \bar{v}_{\alpha,j} = \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} v_h(x_i^\alpha, y) dy, \quad 1 \leq \alpha \leq L, \tag{32}$$

and we use the properties that

$$\begin{aligned} \varphi_h^n(\widehat{x}_i^1, y_j^\alpha) &= \varphi_{i-\frac{1}{2},j}^+(y_j^\alpha) \stackrel{def}{=} \varphi_{i-\frac{1}{2},\alpha}^+, & \varphi_h^n(\widehat{x}_i^N, y_j^\alpha) &= \varphi_{i+\frac{1}{2},j}^-(y_j^\alpha) \stackrel{def}{=} \varphi_{i+\frac{1}{2},\alpha}^-, \\ \varphi_h^n(x_i^\alpha, \widehat{y}_j^1) &= \varphi_{i,j-\frac{1}{2}}^+(x_i^\alpha) \stackrel{def}{=} \varphi_{\alpha,j-\frac{1}{2}}^+, & \varphi_h^n(x_i^\alpha, \widehat{y}_j^N) &= \varphi_{i,j+\frac{1}{2}}^-(x_i^\alpha) \stackrel{def}{=} \varphi_{\alpha,j+\frac{1}{2}}^-. \end{aligned} \tag{33}$$

Next, based on Eq. (31), we introduce the following formulations

$$\begin{aligned}
 H_{x,\alpha}^{(1)}(\varphi_{i-\frac{1}{2},\alpha}^-, \varphi_{i-\frac{1}{2},\alpha}^+) &= (1 + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \bar{u}_{i,\alpha} - \frac{\lambda_x}{\gamma_x \widehat{w}_1} u_{i-\frac{1}{2},\alpha}^+) \varphi_{i-\frac{1}{2},\alpha}^+ + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \widehat{f}(\varphi_{i-\frac{1}{2},\alpha}^-, \varphi_{i-\frac{1}{2},\alpha}^+), \\
 H_{x,\alpha}^{(2)}(\varphi_{i+\frac{1}{2},\alpha}^-, \varphi_{i+\frac{1}{2},\alpha}^+) &= (1 - \frac{\lambda_x}{\gamma_x \widehat{w}_2} \bar{u}_{i,\alpha} + \frac{\lambda_x}{\gamma_x \widehat{w}_2} u_{i+\frac{1}{2},\alpha}^-) \varphi_{i+\frac{1}{2},\alpha}^- - \frac{\lambda_x}{\gamma_x \widehat{w}_2} \widehat{f}(\varphi_{i+\frac{1}{2},\alpha}^-, \varphi_{i+\frac{1}{2},\alpha}^+).
 \end{aligned}
 \tag{34}$$

Substituting the numerical flux Eq. (25) into Eq. (34), it gives

$$\begin{aligned}
 H_{x,\alpha}^{(1)}(\varphi_{i-\frac{1}{2},\alpha}^-, \varphi_{i-\frac{1}{2},\alpha}^+) &= \frac{1}{2} \frac{\lambda_x}{\gamma_x \widehat{w}_1} (u_{i-\frac{1}{2},\alpha}^+ + a_x) \varphi_{i-\frac{1}{2},\alpha}^- + [1 + \frac{1}{2} \frac{\lambda_x}{\gamma_x \widehat{w}_1} (2\bar{u}_{i,\alpha} - u_{i-\frac{1}{2},\alpha}^+ - a_x)] \varphi_{i-\frac{1}{2},\alpha}^+, \\
 H_{x,\alpha}^{(2)}(\varphi_{i+\frac{1}{2},\alpha}^-, \varphi_{i+\frac{1}{2},\alpha}^+) &= [1 - \frac{1}{2} \frac{\lambda_x}{\gamma_x \widehat{w}_2} (2\bar{u}_{i,\alpha} - u_{i+\frac{1}{2},\alpha}^- + a_x)] \varphi_{i+\frac{1}{2},\alpha}^- - \frac{1}{2} \frac{\lambda_x}{\gamma_x \widehat{w}_2} (u_{i+\frac{1}{2},\alpha}^- - a_x) \varphi_{i+\frac{1}{2},\alpha}^+.
 \end{aligned}
 \tag{35}$$

Since a two-point Gauss-Lobatto quadrature rule is applied, the solution average $\bar{u}_{i,\alpha}$ can be expressed as

$$\bar{u}_{i,\alpha} = \widehat{w}_1 u_{i-\frac{1}{2},\alpha}^+ + \widehat{w}_2 u_{i+\frac{1}{2},\alpha}^- = \frac{1}{2} (u_{i-\frac{1}{2},\alpha}^+ + u_{i+\frac{1}{2},\alpha}^-).
 \tag{36}$$

Then, it is easy to check that under the CFL condition Eq. (29), the formulations Eq. (35) are monotone with respect to their two arguments, i.e.,

$$\frac{\partial H_{x,\alpha}^{(1)}(\varphi_{i-\frac{1}{2},\alpha}^-, \varphi_{i-\frac{1}{2},\alpha}^+)}{\partial \varphi_{i-\frac{1}{2},\alpha}^\pm} \geq 0, \quad \frac{\partial H_{x,\alpha}^{(2)}(\varphi_{i+\frac{1}{2},\alpha}^-, \varphi_{i+\frac{1}{2},\alpha}^+)}{\partial \varphi_{i+\frac{1}{2},\alpha}^\pm} \geq 0.
 \tag{37}$$

Moreover, we have

$$\begin{aligned}
 H_{x,\alpha}^{(1)}(m, m) &= (1 + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \bar{u}_{i,\alpha}) m, & H_{x,\alpha}^{(1)}(M, M) &= (1 + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \bar{u}_{i,\alpha}) M, \\
 H_{x,\alpha}^{(2)}(m, m) &= (1 - \frac{\lambda_x}{\gamma_x \widehat{w}_2} \bar{u}_{i,\alpha}) m, & H_{x,\alpha}^{(2)}(M, M) &= (1 - \frac{\lambda_x}{\gamma_x \widehat{w}_2} \bar{u}_{i,\alpha}) M,
 \end{aligned}
 \tag{38}$$

which results in

$$\widehat{w}_1 H_{x,\alpha}^{(1)}(m, m) + \widehat{w}_2 H_{x,\alpha}^{(2)}(m, m) = m, \quad \widehat{w}_1 H_{x,\alpha}^{(1)}(M, M) + \widehat{w}_2 H_{x,\alpha}^{(2)}(M, M) = M.
 \tag{39}$$

Similar results can be obtained for the following formulations

$$\begin{aligned}
 H_{\alpha,y}^{(1)}(\varphi_{\alpha,j-\frac{1}{2}}^-, \varphi_{\alpha,j-\frac{1}{2}}^+) &= (1 + \frac{\lambda_y}{\gamma_y \widehat{w}_1} \bar{v}_{\alpha,j} - \frac{\lambda_y}{\gamma_y \widehat{w}_1} v_{\alpha,j-\frac{1}{2}}^+) \varphi_{\alpha,j-\frac{1}{2}}^+ + \frac{\lambda_y}{\gamma_y \widehat{w}_1} \widehat{g}(\varphi_{\alpha,j-\frac{1}{2}}^-, \varphi_{\alpha,j-\frac{1}{2}}^+), \\
 H_{\alpha,y}^{(2)}(\varphi_{\alpha,j+\frac{1}{2}}^-, \varphi_{\alpha,j+\frac{1}{2}}^+) &= (1 - \frac{\lambda_y}{\gamma_y \widehat{w}_2} \bar{v}_{\alpha,j} + \frac{\lambda_y}{\gamma_y \widehat{w}_2} v_{\alpha,j+\frac{1}{2}}^-) \varphi_{\alpha,j+\frac{1}{2}}^- - \frac{\lambda_y}{\gamma_y \widehat{w}_2} \widehat{g}(\varphi_{\alpha,j+\frac{1}{2}}^-, \varphi_{\alpha,j+\frac{1}{2}}^+).
 \end{aligned}
 \tag{40}$$

Thus, $\bar{\varphi}_{i,j}^{n+1}$ is a monotonically increasing function of all the arguments involved, i.e., $\varphi_{i\pm\frac{1}{2},\alpha}^\mp$ and $\varphi_{\alpha,j\pm\frac{1}{2}}^\mp$, which implies the strict maximum principle

$$m \leq \bar{\varphi}_{i,j}^{n+1} \leq M,
 \tag{41}$$

provided that $m \leq \varphi_h^n(x, y) \leq M$ for any $(x, y) \in S_{i,j}$. □

Now, we complete the first step in constructing a MPS DG(P^1) method for solving the advection equation in non-conservative form, i.e., maintaining the polynomial solution average $\bar{\varphi}_{i,j}^{n+1}$ falling in $[m, M]$. Next, in order to enforce the condition in Theorem 1, the same linear scaling limiter as given in Lemma 2 can be employed to modify $\varphi_h^n(x, y)$ such that the modified polynomial solution $\tilde{\varphi}_h^n(x, y) \in [m, M]$ for all $(x, y) \in S_{i,j}$ without losing accuracy.

Following a similar approach, the previous theorem can be extended to a general DG(P^k) method, which is given as follows

Theorem 2 Consider the scheme satisfied by the cell averages of the DG(P^k) method Eq. (27) for solving the advection equation in non-conservative form Eq. (21). If the approximate polynomials satisfy $\varphi_h^n(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$, then $\bar{\varphi}_{i,j}^{n+1} \in [m, M]$ under the CFL condition

$$a_x \lambda_x + a_y \lambda_y \leq \min\left\{\min_{\beta} \left\{ \frac{\hat{w}_{\beta}}{\sum_{\gamma=1}^N \hat{w}_{\gamma} |\partial_x h_x^{(\beta)}(\hat{x}_i^{\gamma}) \Delta x| + 1}, \frac{\hat{w}_{\beta}}{\sum_{\gamma=1}^N \hat{w}_{\gamma} |\partial_y h_y^{(\beta)}(\hat{y}_j^{\gamma}) \Delta y| + 1} \right\}, \frac{1}{2k+1}\right\}. \tag{42}$$

Here, $k \geq 1$, $a_x = \max |f'(\varphi)| = \|u\|_{\infty}$, $a_y = \max |g'(\varphi)| = \|v\|_{\infty}$ and

$$h_x^{(\beta)}(x) = \prod_{\substack{\gamma=1 \\ \gamma \neq \beta}}^N \frac{(x - \hat{x}_i^{\gamma})}{(\hat{x}_i^{\beta} - \hat{x}_i^{\gamma})}, \quad h_y^{(\beta)}(y) = \prod_{\substack{\gamma=1 \\ \gamma \neq \beta}}^N \frac{(y - \hat{y}_j^{\gamma})}{(\hat{y}_j^{\beta} - \hat{y}_j^{\gamma})} \tag{43}$$

The detailed proof of Theorem 2 is given in Appendix A for interested readers.

3.4 High-Order Temporal Discretization

The temporal discretizations in Eq. (15) and Eq. (23) are conducted by a strong stability preserving Runge-Kutta (SSP-RK) scheme [31]. For example, the DG(P^1) spatial discretization is employed together with the following second-order SSP-RK scheme

$$\begin{aligned} \varphi^{(1)} &= \varphi^n + \Delta t R(\varphi^n), \\ \varphi^{n+1} &= \frac{1}{2} \varphi^n + \frac{1}{2} [\varphi^{(1)} + \Delta t R(\varphi^{(1)})]. \end{aligned} \tag{44}$$

Here, $R(\varphi)$ is the spatial operator in either conservative or non-conservative form. While for the DG(P^2) spatial discretization, the third-order SSP-RK scheme is used to evolve the solution

$$\begin{aligned} \varphi^{(1)} &= \varphi^n + \Delta t R(\varphi^n), \\ \varphi^{(2)} &= \frac{3}{4} \varphi^n + \frac{1}{4} [\varphi^{(1)} + \Delta t R(\varphi^{(1)})], \\ \varphi^{n+1} &= \frac{1}{3} \varphi^n + \frac{2}{3} [\varphi^{(2)} + \Delta t R(\varphi^{(2)})]. \end{aligned} \tag{45}$$

We remark that since the multi-stage SSP-RK scheme can be regarded as a convex combinations of the Euler forward temporal discretization, the previous analysis keeps validity.

4 Maximum-Principle-Satisfying DG Method for the Re-initialization Equation

In order to maintain the shape of the hyperbolic tangent function and limit mass loss, a re-initialization procedure is carried out after every real time step. The DG discretization of the re-initialization Eq. (5) can be written as

$$\int_{I_{i,j}} (\partial_\tau \varphi_h) \psi d\mathbf{x} - \int_{I_{i,j}} (\mathbf{G}_c - \mathbf{G}_d) \cdot \nabla \psi d\mathbf{x} + \int_{\partial I_{i,j}} \mathbf{n}_e \cdot (\widehat{\mathbf{G}}_c - \widehat{\mathbf{G}}_d) \psi ds = 0. \tag{46}$$

Here, $\mathbf{G}_c = \varphi(1 - \varphi)\widehat{\mathbf{n}}|_{\tau=0}$ and $\mathbf{G}_d = \varepsilon \nabla \varphi$ are the compressive and diffusion terms, respectively. For the compressive part, we use the Lax-Friedrichs flux

$$\mathbf{n}_e \cdot \widehat{\mathbf{G}}_c(\varphi_h^-, \varphi_h^+) = \frac{1}{2} [\mathbf{n}_e \cdot (\mathbf{G}_c(\varphi_h^-) + \mathbf{G}_c(\varphi_h^+)) - a(\varphi_h^+ - \varphi_h^-)], \tag{47}$$

where a is the global maximum value of $|(\partial/\partial\varphi)(\mathbf{n}_e \cdot \widehat{\mathbf{G}}_c(\varphi_h))| = |\mathbf{n}_e \cdot (1 - 2\varphi_h)\widehat{\mathbf{n}}|_{\tau=0}|$, while for the diffusive part, we adopt the direct DG (DDG) flux [32,33]. Specifically, in the DG(P^2) formulation, the diffusive flux is defined as

$$\mathbf{n}_e \cdot \widehat{\mathbf{G}}_d(\varphi_h^-, \varphi_h^+) = \mathbf{n}_e \cdot (\varepsilon \widehat{\nabla \varphi}) = \varepsilon(\beta_0 \frac{[[\varphi]]}{\Delta} + \{\partial_{\mathbf{n}}\varphi\} + \beta_1 \Delta [[\partial_{\mathbf{n}}^2 \varphi]]). \tag{48}$$

Here, Δ is the characteristic length, $\partial_{\mathbf{n}}\varphi = \mathbf{n}_e \cdot \nabla \varphi$, $[[\cdot]]$ and $\{\cdot\}$ are the jump and average operators, respectively. (β_0, β_1) are constant coefficients need to be specified such that the DG method Eq. (46) can satisfy the maximum principle.

Eq. (46) is marched in pseudo-time τ by the third-order SSP-RK scheme Eq. (45). Theoretically, Eq. (46) should be solved to steady state. However, in practice we typically perform only a few, e.g., 10~20, iterations for improving efficiency. Before giving the sufficient condition for the cell averages of the DG method Eq. (46) to be bounded in $[m, M]$, we first introduce Lemma 3 [21].

Lemma 3 Consider the DG method associated with piecewise-quadratic polynomial approximations for solving the two-dimensional diffusion equation

$$\frac{\partial \varphi}{\partial t} = \nabla \cdot (\varepsilon \nabla \varphi), \quad (\mathbf{x}, t) \in \Omega \times (0, T). \tag{49}$$

Given $\varphi_h^n(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$, we have $\overline{\varphi}_{i,j}^{n+1} \in [m, M]$ provided that

$$\beta_0 \geq \frac{3}{2} - 4\beta_1, \quad \frac{1}{8} \leq \beta_1 \leq \frac{1}{4} \tag{50}$$

combined with a suitable CFL condition

$$\frac{\varepsilon \Delta t}{|I_{i,j}|} \leq \min \left\{ \frac{1}{6(\beta_0 + 8\beta_1 - 2)}, \frac{1}{6(1 - 4\beta_1)} \right\} \min \left\{ \frac{\lambda_x}{\lambda_x + \lambda_y}, \frac{\lambda_y}{\lambda_x + \lambda_y} \right\}. \tag{51}$$

Based on Lemma 3, we then have the following result

Theorem 3 Consider the scheme satisfied by the cell averages of the DG(P^2) method Eq. (46) for the re-initialization equation. If the approximate polynomials satisfy $\varphi_h^s(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$ and the coefficients (β_0, β_1) satisfy the condition Eq. (50), then $\overline{\varphi}_{i,j}^{s+1} \in$

$[m, M]$ under the CFL condition

$$\begin{cases} a_1\lambda_1 + a_2\lambda_2 \leq \frac{1}{16}, \\ \lambda_d \leq \min \left\{ \frac{1}{12(\beta_0 + 8\beta_1 - 2)}, \frac{1}{12(1 - 4\beta_1)} \right\} \min \left\{ \frac{\lambda_1}{\lambda_1 + \lambda_2}, \frac{\lambda_2}{\lambda_1 + \lambda_2} \right\}. \end{cases} \tag{52}$$

Here, s is the pseudo-time level, $\lambda_1 = \Delta\tau/\Delta x$, $\lambda_2 = \Delta\tau/\Delta y$, $a_1 = \|(1 - 2\varphi)\widehat{n}_x\|_\infty$, $a_2 = \|(1 - 2\varphi)\widehat{n}_y\|_\infty$ and $\lambda_d = \varepsilon\Delta\tau/|I_{i,j}|$.

Proof We present the proof in four steps:

- 1. *Split*: We split the average $\overline{\varphi}_{i,j}^{s+1}$ into two halves such that

$$\overline{\varphi}_{i,j}^{s+1} = \frac{1}{2}C_{i,j}^s + \frac{1}{2}D_{i,j}^s, \tag{53}$$

where the convection term is

$$C_{i,j} = \overline{\varphi}_{i,j}^s - \frac{2\Delta\tau}{|I_{i,j}|} \int_{\partial I_{i,j}} \mathbf{n}_e \cdot \widehat{\mathbf{G}}_c ds \tag{54}$$

and the diffusion term is

$$D_{i,j} = \overline{\varphi}_{i,j}^s + \frac{2\Delta\tau}{|I_{i,j}|} \int_{\partial I_{i,j}} \mathbf{n}_e \cdot \widehat{\mathbf{G}}_d ds. \tag{55}$$

It should be noted that this split is designed for convenient presentation purpose and may not lead to an optimal CFL condition [22].

- 2. *Convection*: Following the same arguments as in Theorem 2 and replacing Δt with $2\Delta\tau$, it gives that if $\varphi_h^s(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$, then $C_{i,j} \in [m, M]$ under the CFL condition

$$a_1\lambda_1 + a_2\lambda_2 \leq \frac{3}{8}\widehat{w}_1 = \frac{1}{16}. \tag{56}$$

- 3. *Diffusion*: Applying the results in Lemma 3 and replacing Δt with $2\Delta\tau$, it provides that with $\varphi_h^s(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$ and (β_0, β_1) satisfy the condition Eq. (50), we have $D_{i,j} \in [m, M]$ under the CFL condition

$$\lambda_d \leq \min \left(\frac{1}{12(\beta_0 + 8\beta_1 - 2)}, \frac{1}{12(1 - 4\beta_1)} \right) \min \left(\frac{\lambda_1}{\lambda_1 + \lambda_2}, \frac{\lambda_2}{\lambda_1 + \lambda_2} \right). \tag{57}$$

- 4. *Summation*: Plugging the results of $m \leq C_{i,j}, D_{i,j} \leq M$ in Eq. (53) gives

$$m \leq \overline{\varphi}_{i,j}^{s+1} \leq M. \tag{58}$$

□

Again, the bound of the approximate polynomial $\varphi_h^s(x, y)$ required in Theorem 3 is guaranteed by simply applying the linear scaling limiter Eq. (19).

Finally, taking the third-order MPS DG method as an example, we summary the solution algorithm for each time step from t^n to t^{n+1} as follows:

Algorithm 1 MPS DG method for the level set problem.

1. At time level t^n , apply the linear scaling limiter Eq. (19) to φ_h^n to obtain $\tilde{\varphi}_h^n \in [m, M]$.
2. Advance $\tilde{\varphi}_h^n$ by the non-conservative DG method Eq. (23) with the SSP-RK scheme Eq. (45) under the CFL condition Eq. (42) to obtain φ_h^{n+1} .
3. Update the normal vector $\hat{\mathbf{n}}|_{\tau=0}$ from φ_h^{n+1} .
4. Solve the re-initialization Eq. (5) iteratively by the DG method Eq. (46) with the SSP-RK scheme Eq. (45) under the CFL condition Eq. (52) to obtain $\tilde{\varphi}_h^{n+1}$.
5. Reset φ_h^{n+1} as $\tilde{\varphi}_h^{n+1}$.
6. Repeat step 1 to step 5 until the desired time level is reached.

Table 1 Convergence results of the conservative and non-conservative DG methods

Grid number	Conservative DG method				Non-conservative DG method			
	L^1 -error	Order	L^2 -error	Order	L^1 -error	Order	L^2 -error	Order
P^1								
10×10	4.50E-03		6.04E-03		4.49E-03		6.03E-03	
20×20	9.53E-04	2.24	1.23E-03	2.30	9.52E-04	2.24	1.23E-03	2.29
40×40	2.21E-04	2.11	2.64E-04	2.22	2.21E-04	2.11	2.64E-04	2.22
80×80	5.47E-05	2.01	6.24E-05	2.08	5.47E-05	2.01	6.24E-05	2.08
P^2								
10×10	2.88E-04		3.95E-04		2.87E-04		3.94E-04	
20×20	3.06E-05	3.23	4.32E-05	3.19	3.06E-05	3.23	4.31E-05	3.19
40×40	3.23E-06	3.24	4.51E-06	3.26	3.23E-06	3.24	4.50E-06	3.26
80×80	3.64E-07	3.15	4.89E-07	3.21	3.64E-07	3.15	4.89E-07	3.20

5 Numerical Tests

In this section, a number of benchmark test cases are carried out to illustrate the performance of the proposed method. For simplicity, in the following, the $DG(P^k)$ methods Eq. (15) with and without the MPS limiter for solving the advection equation in conservative form will be denoted as conservative MPS $DG(P^k)$ and conservative $DG(P^k)$ methods, respectively. While the $DG(P^k)$ methods Eq. (23) with and without the MPS limiter for solving the advection equation in non-conservative form will be denoted as non-conservative MPS $DG(P^k)$ and non-conservative $DG(P^k)$ methods, respectively. The union of these four methods will be denoted as $DG(P^k)$ method.

5.1 Convergence Test

We perform a convergence test on the high-order DG methods for solving the advection equations in both conservative and non-conservative forms. Consider an initial condition given by [34]

$$\varphi(\mathbf{x}, t = 0) = \tanh((y - 0.5)/0.25) \tag{59}$$

on a unit-sized domain with velocity

$$\mathbf{u}(\mathbf{x}, t) = (\sin(\pi x) \cos(\pi y) \sin(2\pi t), -\cos(\pi x) \sin(\pi y) \sin(2\pi t))^T. \tag{60}$$

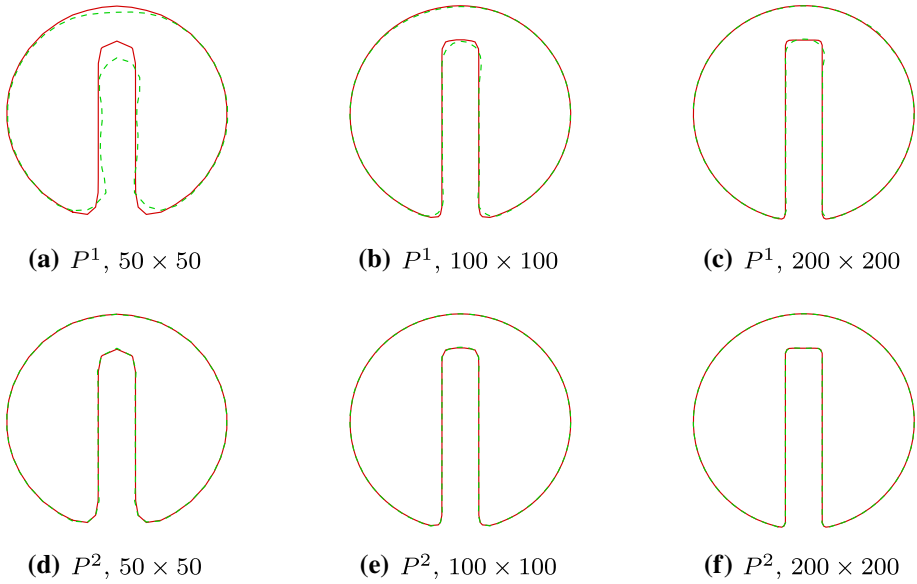


Fig. 3 Interface location for the Zalesak’s disk after one full rotation (red solid line: exact solution, green dashed line: non-conservative MPS DG method) (Color figure online)

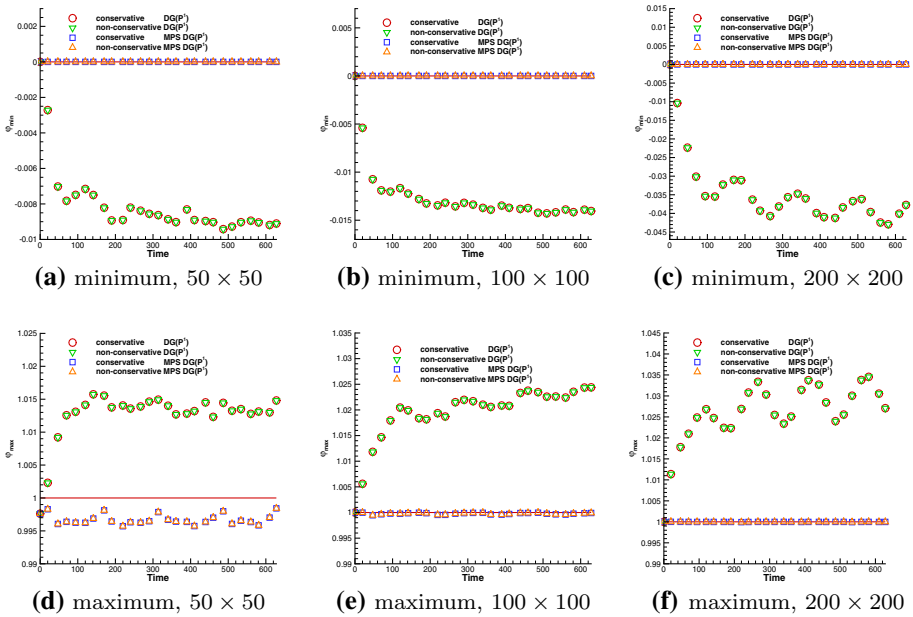


Fig. 4 Minimum and maximum values of $\bar{\varphi}$ in the Zalesak’s disk obtained by the $DG(P^1)$ method

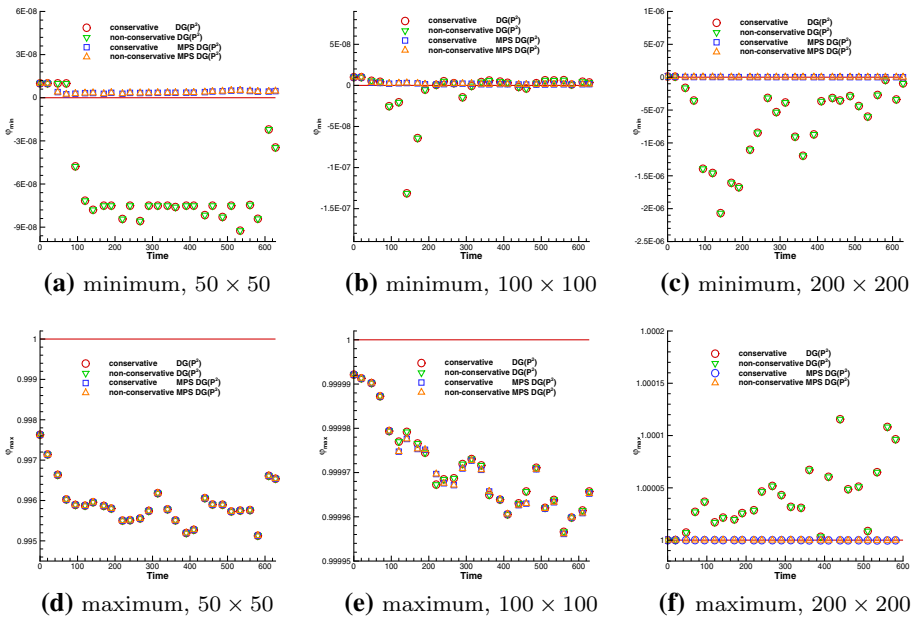


Fig. 5 Minimum and maximum values of \bar{v} in the Zalesak’s disk obtained by the DG(P^2) method

Since the velocity field is periodic and the problem is linear, the exact solution at $t = 1$ coincides with the initial condition.

The convergence results obtained by the conservative and non-conservative DG(P^k) methods are listed in Table 1. One can see that both these two methods can deliver the optimal $(k + 1)$ -th order of accuracy. Moreover, due to the velocity field used in this test problem is exactly divergence free, the resulting solutions of these two methods are nearly the same as stated in Remark 2.

5.2 Solid Body Rotation of a Notched Disk

The solid body rotation of a notched circle, also known as Zalesak’s disk [35], is one of the standard test problems for testing the ability of the level set methods to transport a complex geometry with sharp corners. The Zalesak’s disk is defined as follows

- Domain= $[0, 100]^2$;
- Radius=15, initial center=(50, 75);
- Slot width=5, slot length=25.

The constant vorticity velocity field is given by

$$\mathbf{u}(\mathbf{x}, t) = (\pi/314)(50 - y, x - 50)^T, \tag{61}$$

so that the disk completes one revolution every 628 time units. In order to test how mesh refinement and polynomial order affect the final shape of the notched disk, this problem will be solved by the DG(P^1) and DG(P^2) methods with three different meshes consisting of 50×50 , 100×100 and 200×200 grids.

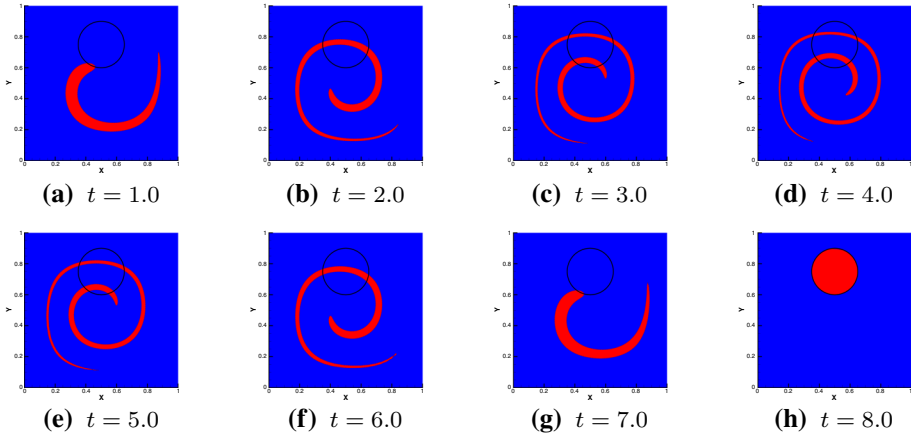


Fig. 6 Interface location for the circle in a deformation field obtained by the non-conservative MPS DG(P^2) method with 256×256 mesh grids

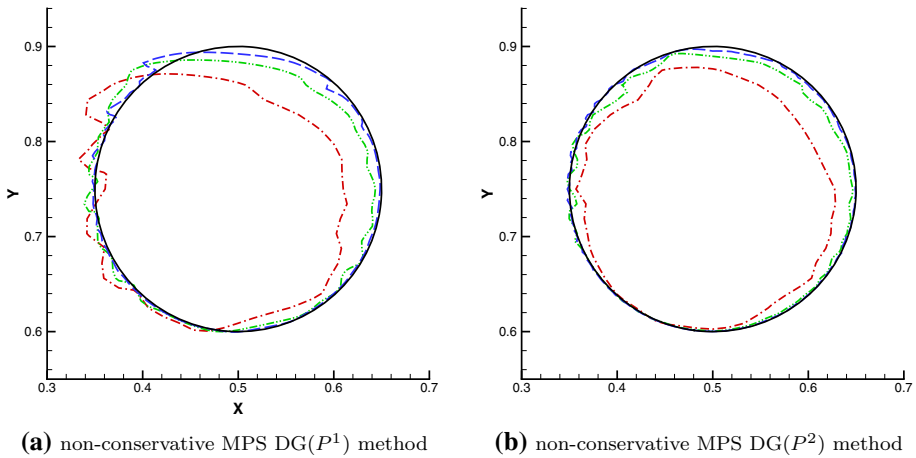


Fig. 7 Interface location for the circle in a deformation field at $t = T$ (black solid line: exact solution, red dash dotted line: 64×64 grids, green dash dotted dotted line: 128×128 grids, blue dashed line: 256×256 grids)

Figure 3 shows that compared with the DG(P^1) method, the DG(P^2) method greatly improves the shape of the notch after one period rotation, especially on the coarsest mesh. When the mesh is refined to 100×100 and 200×200 grids, the shape corners of the notched disk are well resolved by the DG(P^2) method and visually no difference can be discerned in comparison with the exact solution.

Next, we validate the maximum-principle-satisfying property of the proposed method. Time-variations of the minimum and maximum values of the cell average solutions are shown in Fig. 4 and 5 for the DG(P^1) and DG(P^2) methods, respectively. It can be observed that as the velocity field is divergence free, the solutions obtained by the conservative and non-conservative schemes are very close. Moreover, the numerical solutions of the DG(P^1) and DG(P^2) methods with the MPS limiter are strictly in the interval $[0, 1]$, while there are obvious overshoots and undershoots outside $[0, 1]$ for the results without the MPS limiter.

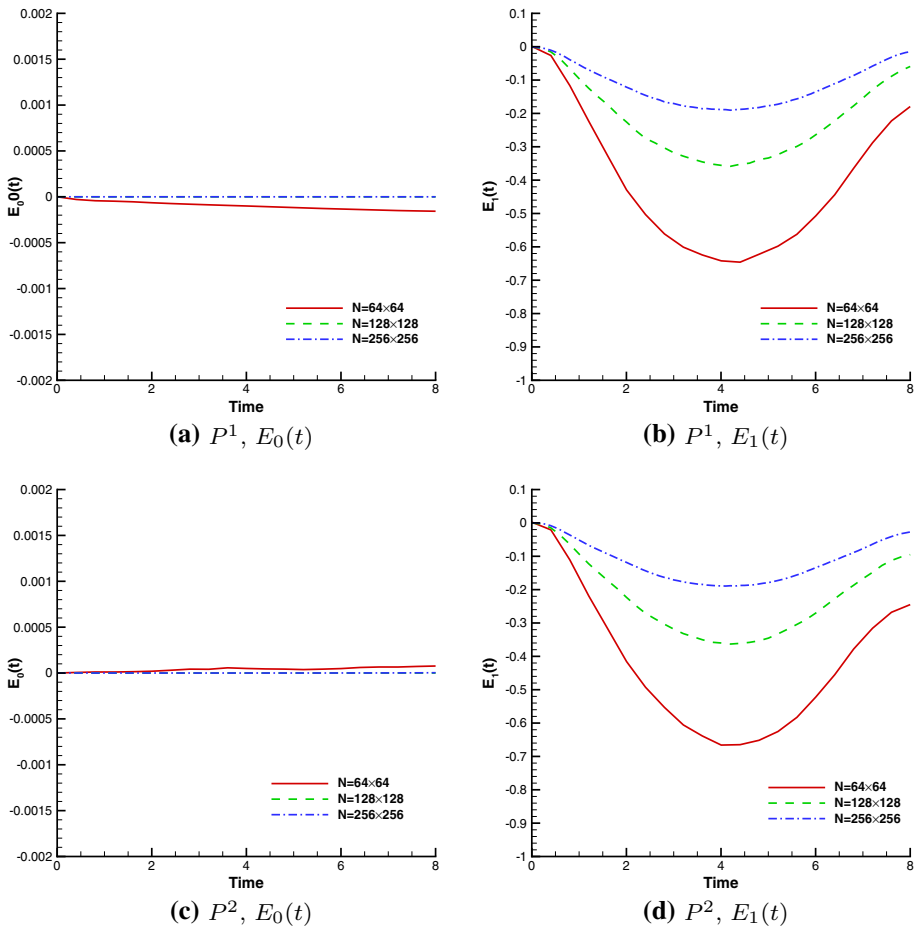


Fig. 8 Temporal evolution of the normalized mass errors for the circle in a deformation field obtained by the non-conservative MPS DG method (Color figure online)

This clearly demonstrates that the MPS limiter can preserve the strict maximum principle for the proposed DG(P^k) method for solving the advection equation in non-conservative form.

5.3 Single Vortex Deformation of a Circle

To test the ability of the proposed MPS DG methods to resolve and maintain ever thinner filaments, a circle in a deformation field problem, which was first introduced by Bell *et al.* [36] and then applied as a level set test problem by Enright *et al.* [37], is conducted. A circle of radius $r = 0.15$, initially centered at $(0.5, 0.75)^T$ is deformed inside a unit sized box under a solenoidal velocity

$$\begin{cases} u(\mathbf{x}, t) = -\sin^2(\pi x) \sin(2\pi y) \cos(\pi t/T), \\ v(\mathbf{x}, t) = +\sin^2(\pi y) \sin(2\pi x) \cos(\pi t/T), \end{cases} \quad (62)$$

where $T = 8$ is the time at which the flow returns to its initial state.

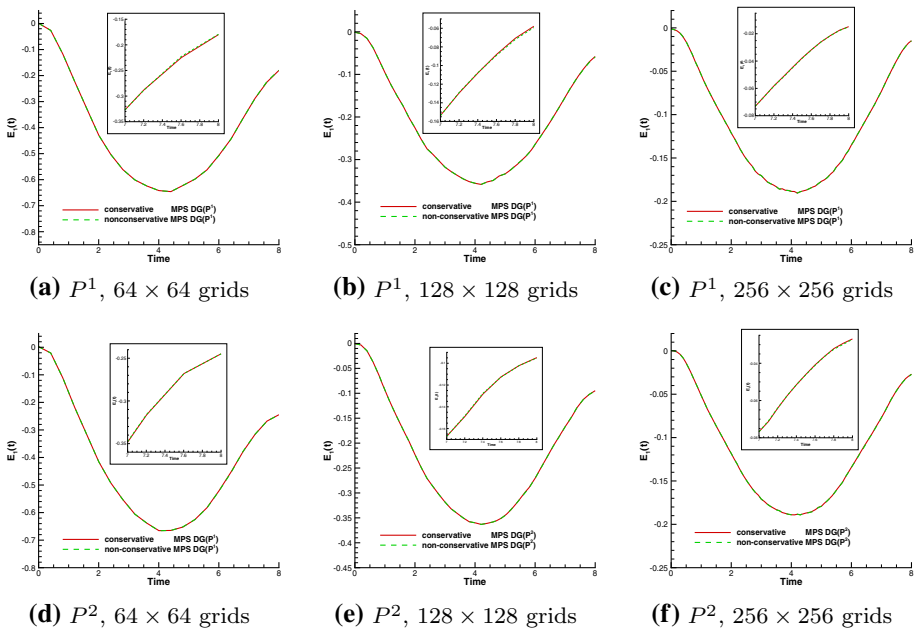


Fig. 9 Comparison of the normalized mass error $E_1(t)$ for the circle in a deformation field obtained by the conservative and non-conservative MPS DG methods

Fig. 6 shows the evolution of the circle shape in the deformation field obtained by the non-conservative MPS DG (P^2) method on a 256×256 mesh grid. It can be seen that the circle is stretched by the velocity field onto ever thinner filaments until $t = 4$. Then, as the velocity field reverses for another four time units, it pulls the filaments back into the initial circular shape [38]. We observe that during the entire stretching process, our method is able to capture very thin fluid filament.

Now, we turn our attention the convergence property of the proposed method, computations on three different mesh resolutions, refined from 64×64 , 128×128 to 256×256 mesh grids, are performed and compared in Fig. 7. It is clear to see that the recovered circle is very comparable to its initial shape when a fine mesh and a high-order method are applied. Next, for further quantitatively comparing the mass conservation property of the proposed method, we define the relative mass errors as follows

$$E_0(t) = \frac{\int_{\Omega} \varphi(\mathbf{x}, t) d\mathbf{x} - \int_{\Omega} \varphi(\mathbf{x}, 0) d\mathbf{x}}{\int_{\Omega} \varphi(\mathbf{x}, 0) d\mathbf{x}},$$

$$E_1(t) = \frac{\int_{\Omega} \varphi(\mathbf{x}, t) d\mathbf{x} - \int_{\Omega} \varphi(\mathbf{x}, 0) d\mathbf{x}}{\int_{\Omega} \varphi(\mathbf{x}, 0) d\mathbf{x}}, \quad \text{for } 0.5 \leq \varphi \leq 1. \tag{63}$$

The effect of mesh refinement on the mass conservative property of the proposed method is studied and the computed mass losses are plotted in Fig. 8. As we expect, on the finest mesh, the amount of the tail region that falls below mesh resolution and transfers into resolvable droplets is the least and therefore the conservation error around $t = 4$ is the smallest. For the coarser meshes, more of the tail region falls below mesh resolution and transfers into droplets, which leads to large curvature changes and significant loss of mass during the simulation.

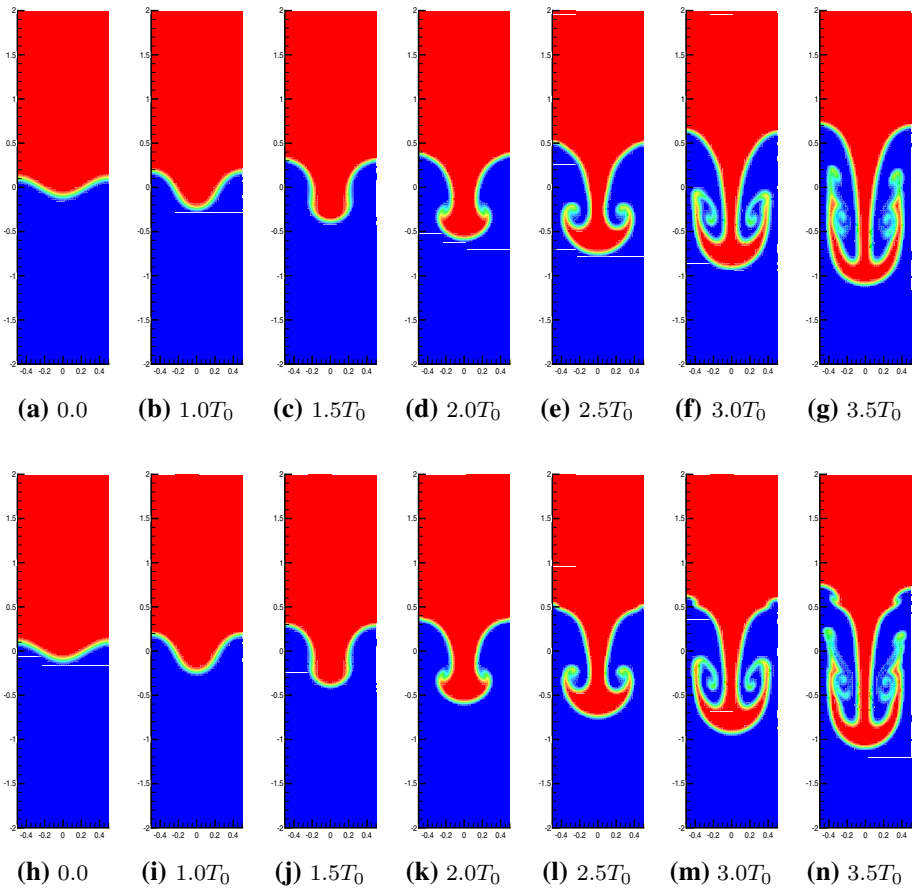


Fig. 10 Temporal evolution of the interface for the Rayleigh–Taylor instability at t^* obtained by the non-conservative MPS DG method ($At = 0.5$, $Re = 1000$, top: P^1 , bottom: P^2)

In the end, we compare the normalized mass error $E_1(t)$ obtained by the conservative and the non-conservative MPS DG methods. The comparisons are shown in Fig. 9. Again, as stated in Remark 2, since a smooth and divergence-free velocity field is applied in this test, the results obtained by these two methods are very close to each other which demonstrate the accuracy and effectiveness of the proposed non-conservative DG method.

5.4 Viscous Rayleigh–Taylor Instability

In this subsection, we combine our high-order MPS DG method for the LS function with an incompressible two-phase flow solver to illustrate the performance of the proposed method for a practical problem, namely the Rayleigh–Taylor instability [39–42]. The determination of the incompressible two-phase fluids requires the solution of the following INS equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= \frac{1}{\rho} (-\nabla p + \nabla \cdot [\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)]) + \mathbf{g}, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{64}$$

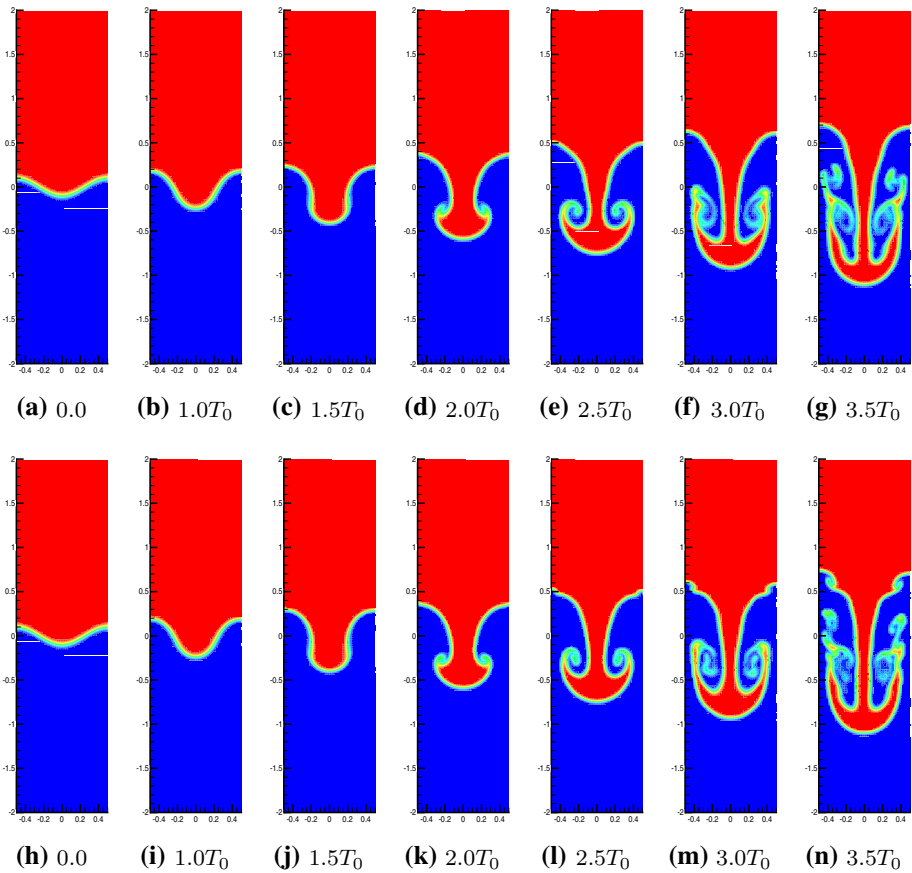


Fig. 11 Temporal evolution of the interface for the Rayleigh–Taylor instability at t^* obtained by the non-conservative MPS DG method ($At = 0.5$, $Re = 5000$, top: P^1 , bottom: P^2)

Here, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity vector, $p = p(\mathbf{x}, t)$ is the pressure and \mathbf{g} is a unit vector aligned with gravity. $\rho(\mathbf{x}, t)$ is the density calculated by

$$\rho(\mathbf{x}, t) = \rho_m + (\rho_M - \rho_m)\varphi(x, t), \tag{65}$$

where ρ_M and ρ_m ($\rho_M > \rho_m$) denote the densities of the heavier fluid and the light fluid, respectively, and $\mu > 0$ is the dynamic viscosity of the fluid (supposed to be constant in the whole domain). Initially, the heavier fluid superposed to the light fluid and the perturbed interface is described by

$$\eta(x) = -0.1d \cos(2\pi x/d), \quad d = 1. \tag{66}$$

The computing domain is $\Omega = (-d/2, d/2) \times (-2d, 2d)$. Non-slip boundary condition is enforced at the bottom and top walls and symmetry boundary condition is imposed on the two vertical sides. In this work, numerical discretization of the INS equations (64) is performed using a high-order DG method proposed in [43–45] for its simplicity in implementation.

The difficulty of this problem essentially depends on:

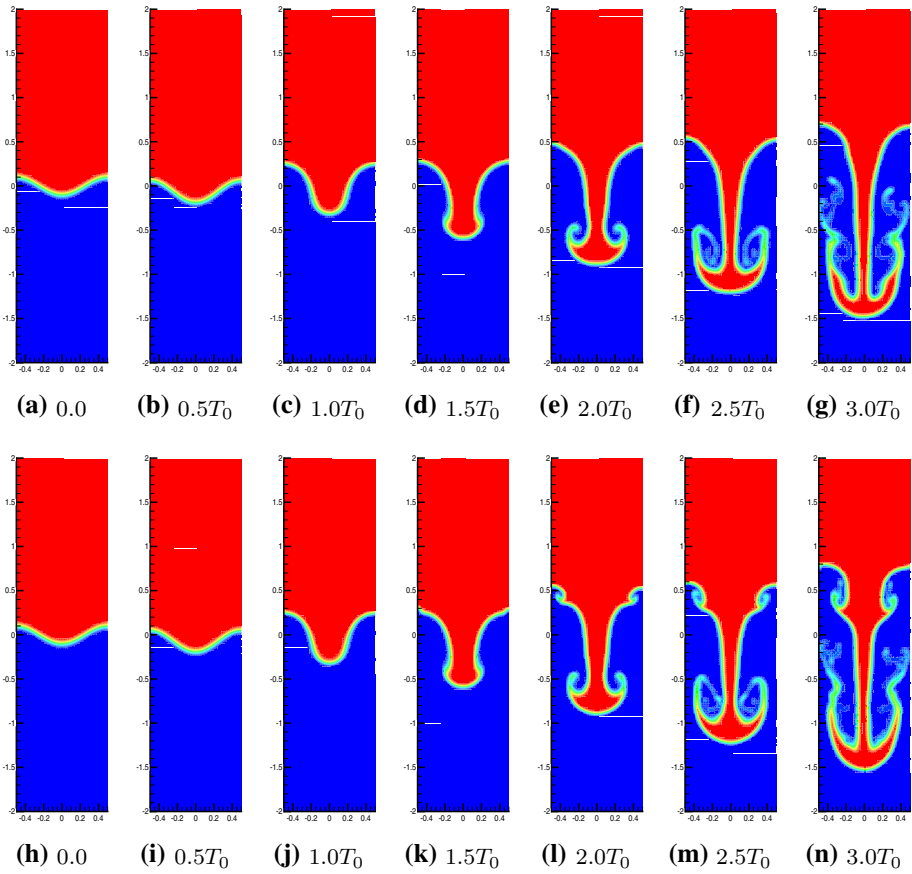


Fig. 12 Temporal evolution of the interface for the Rayleigh–Taylor instability at t^* obtained by the non-conservative MPS DG method ($At = 0.75$, $Re = 1000$, top: P^1 , bottom: P^2)

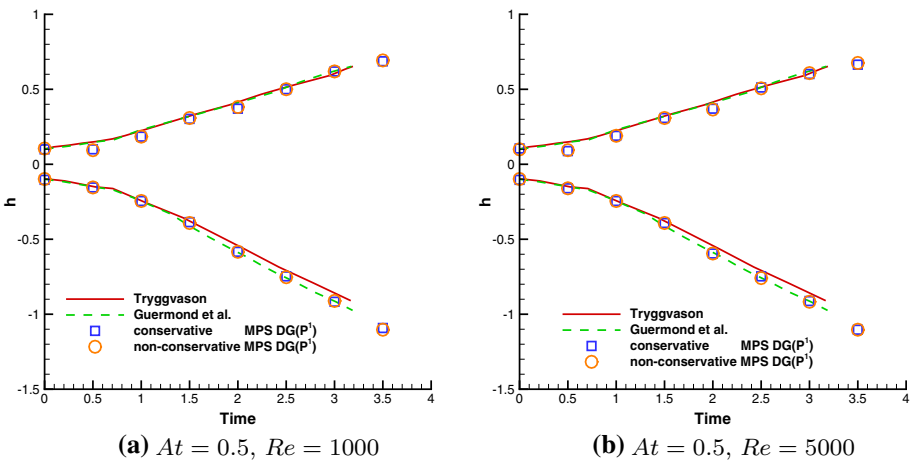


Fig. 13 Temporal evolution of the highest (top line) and the lowest (bottom line) interface locations for the Rayleigh–Taylor instability

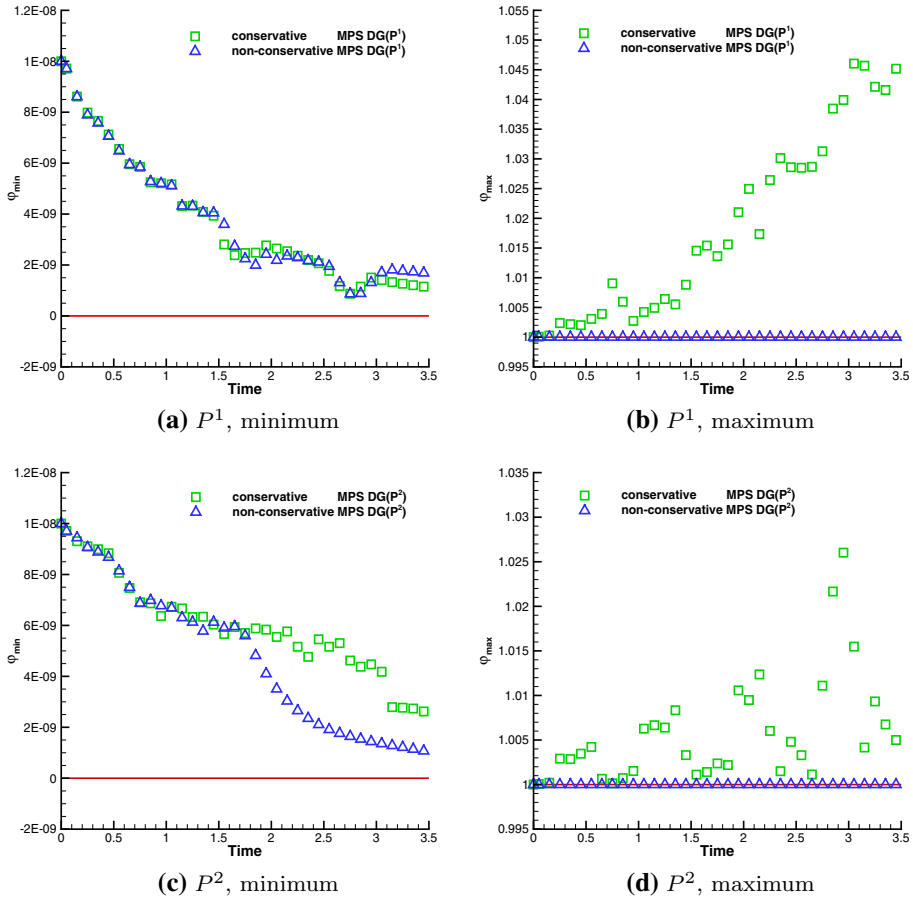


Fig. 14 Minimum and maximum values of $\bar{\varphi}_{i,j}$ for the Rayleigh–Taylor instability ($At = 0.5, Re = 1000$)

1. The density ratio between the heavy fluid and the light fluid, which is measured by the so-called Atwood number

$$At = \frac{\rho_M - \rho_m}{\rho_M + \rho_m}. \tag{67}$$

2. The viscosity of the fluid, which is measured by the Reynolds number

$$Re = \frac{\rho_m d^{3/2} g^{1/2}}{\mu}. \tag{68}$$

We compare the solutions at different Atwood and Reynolds numbers. A uniform mesh with 50×200 grids is employed for all the computations.

- *A low Atwood number problem at a low Reynolds number:* Setting $At = 0.5$ ($\rho_M = 3, \rho_m = 1$), we begin with a low Reynolds number case, $Re = 1000$. The temporal evolution of the interface at $t^* = (t/t_0)T_0 = (0.0, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5)T_0$, where $T_0 = \sqrt{At}$, is displayed in Fig. 10. The results are very close to those in [39,40], and we can only observe some slight difference nearly at the end of the simulation.

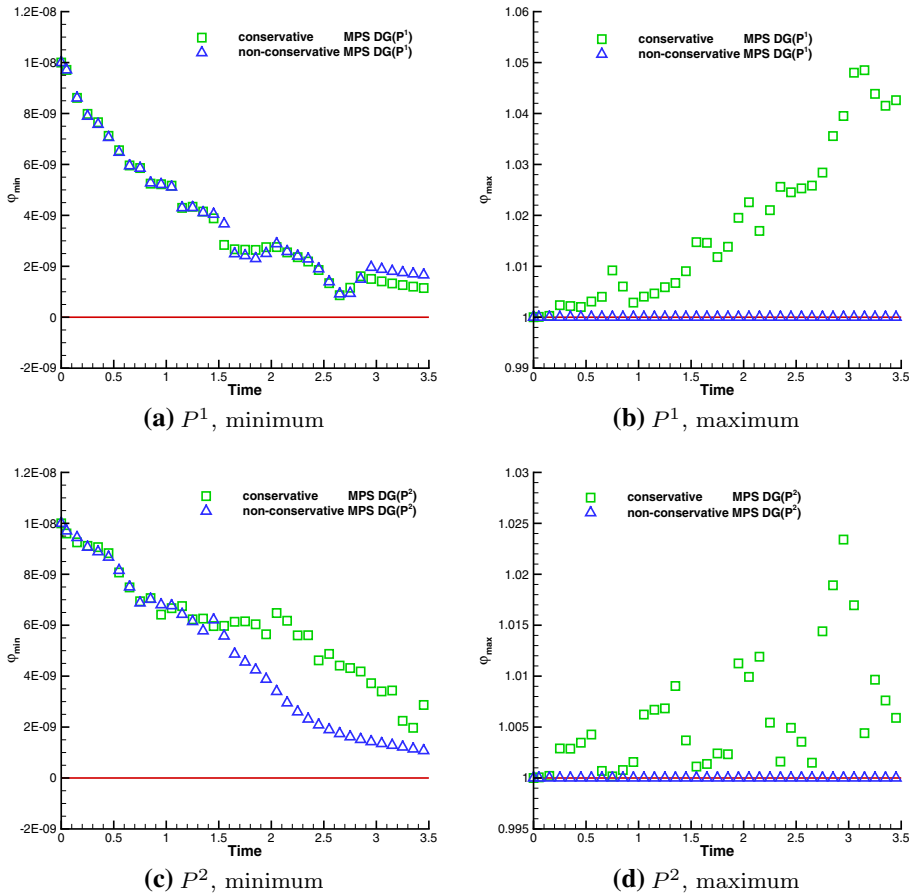


Fig. 15 Minimum and maximum values of $\bar{\varphi}_{i,j}$ for the Rayleigh–Taylor instability ($At = 0.5, Re = 5000$)

- *A low Atwood number problem at a high Reynolds number:* Setting $At = 0.5$ ($\rho_M \equiv 3, \rho_m = 1$) and $Re = 5000$. The temporal evolution of the interface at $t^* = (t/t_0)\sqrt{At} = (0.0, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5)T_0$ under this case is displayed in Fig. 11. We observe that when the Reynolds number increases, the velocity of the characteristic mushroom shape remains the same as that of the lower Reynolds case. The influence of increasing Reynolds number appears only in the shape of the rising counter-rotating vortices, which induces many different small structures for $t^* \geq 3.0T_0$.
- *A high Atwood number problem at a low Reynolds number:* Setting $At = 0.75$ ($\rho_M = 7, \rho_m = 1$) and $Re = 1000$. The temporal evolution of the interface at $t^* = (0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0)T_0$ under this case is displayed in Fig. 12. Compared with the above test, we can observe the similar structure and the global characteristic of the flow in the early stage. At the same time, we find that the heavy fluid falls faster compared with the low Atwood number problem. The simulation results coincide well with those presented in [41].

Next, to quantitatively measure the motion of the interface for the Rayleigh–Taylor instability problem, we track the highest and the lowest vertical locations of the interface obtained

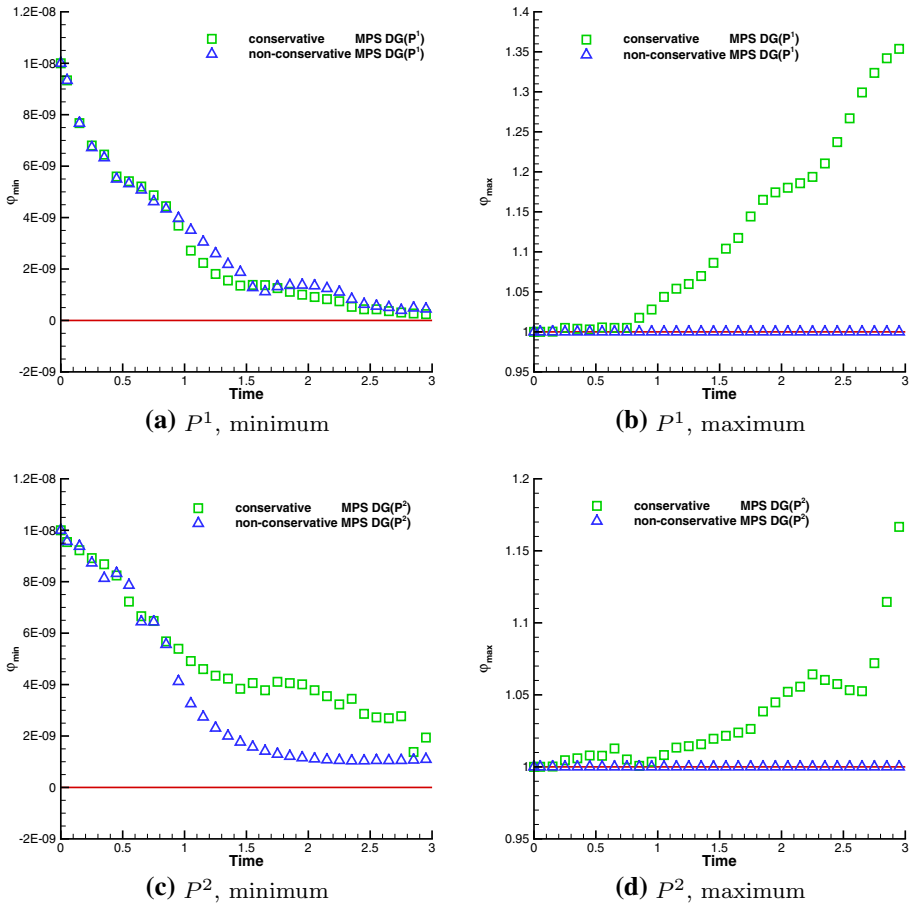


Fig. 16 Minimum and maximum values of $\bar{\varphi}_{i,j}$ for the Rayleigh–Taylor instability ($At = 0.75, Re = 1000$)

by the conservative and non-conservative MPS DG(P^1) methods and compare them with the available results [40,42]. The comparison is presented in Fig. 13 which clearly shows good agreement.

In the end, we validate the maximum preserving property of the proposed MPS DG method. In this test, due to numerical oscillations, the divergence of the discrete velocity field is not identically equal to zero. Thus, the conservative DG method can not satisfy the maximum principle even though implemented with the MPS limiter. However, the non-conservative MPS DG method can still keep the strict maximum principle and no value greater than $M = 1$ or smaller than $m = 0$ is encountered during the simulation as shown in Figs. 14, 15, 16. Thus, the high-order non-conservative MPS DG method can be used to solve the incompressible two-phase flows with large density ratio.

6 Concluding Remarks

In this paper, a general framework to construct high-order accurate maximum-principle-satisfying DG method has been developed for solving the level set problem with a

non-divergence-free velocity field. We proved that by applying a linear scaling limiter and a strong stability preserving time discretization scheme, the numerical solutions of the level set function obtained by the proposed DG method satisfy the strict maximum principle under a suitable CFL condition. Various numerical test cases, including the application to incompressible two-phase flows, have been presented to demonstrate the validity and accuracy of the proposed method. Generalizations to unstructured grids constitute our ongoing work.

Acknowledgements This work is supported by National Natural Science Foundation of China (No. 12001020) and China Postdoctoral Science Foundation (No. 2020M680176).

A Proof of Theorem 2

Consider an arbitrary order P^k approximation ($k \geq 1$). By using an N -point Gauss-Lobatto and an L -point Gauss-Legendre quadrature rules exact for single-variable polynomials of degree k , we can represent the solution $\varphi_h(x, y)$ along the line $y = y_j^\alpha$ ($1 \leq \alpha \leq L$) as

$$\varphi_h(x, y_j^\alpha) = \sum_{\beta=1}^N \varphi_{\widehat{\beta}, \alpha} \prod_{\gamma=1, \gamma \neq \beta}^N \frac{(x - \widehat{x}_i^\gamma)}{(\widehat{x}_i^\beta - \widehat{x}_i^\gamma)} \stackrel{def}{=} \sum_{\beta=1}^N \varphi_{\widehat{\beta}, \alpha} h^{(\beta)}(x). \tag{69}$$

Here, $\varphi_{\widehat{\beta}, \alpha} \stackrel{def}{=} \varphi_h(\widehat{x}_i^\beta, y_j^\alpha)$. Taking derivative once with respect to x results in

$$\partial_x \varphi_h(x, y_j^\alpha) = \sum_{\beta=1}^N \varphi_{\widehat{\beta}, \alpha} \frac{\sum_{\substack{\kappa=1 \\ \kappa \neq \beta}}^N \prod_{\substack{s=1 \\ s \neq \beta, \kappa}}^N (x - \widehat{x}_i^s)}{\prod_{\substack{\gamma=1 \\ \gamma \neq \beta}}^N (\widehat{x}_i^\beta - \widehat{x}_i^\gamma)} \stackrel{def}{=} \sum_{\beta=1}^N \varphi_{\widehat{\beta}, \alpha} h_x^{(\beta)}(x). \tag{70}$$

Similarly, we have

$$\partial_y \varphi_h(x_i^\alpha, y) = \sum_{\beta=1}^N \varphi_{\alpha, \widehat{\beta}} \frac{\sum_{\substack{\kappa=1 \\ \kappa \neq \beta}}^N \prod_{\substack{s=1 \\ s \neq \beta, \kappa}}^N (y - \widehat{y}_j^s)}{\prod_{\substack{\gamma=1 \\ \gamma \neq \beta}}^N (\widehat{y}_j^\beta - \widehat{y}_j^\gamma)} \stackrel{def}{=} \sum_{\beta=1}^N \varphi_{\alpha, \widehat{\beta}} h_y^{(\beta)}(y). \tag{71}$$

Here, $\varphi_{\alpha, \hat{\beta}} \stackrel{def}{=} \varphi_h(x_i^\alpha, \hat{y}_j^\beta)$. Then, the surface integral in Eq. (27) can be further written as

$$\begin{aligned} & \frac{\Delta t}{|I_{i,j}|} \int_{I_{i,j}} \mathbf{u}_h \cdot \nabla \varphi_h d\mathbf{x} \\ &= \lambda_x \sum_{\alpha=1}^L w_\alpha \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_h(x, y_j^\alpha) \partial_x \varphi_h(x, y_j^\alpha) dx + \lambda_y \sum_{\alpha=1}^L w_\alpha \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} v_h(x_i^\alpha, y) \partial_y \varphi_h(x_i^\alpha, y) dy \\ &= \lambda_x \sum_{\alpha=1}^L \sum_{\gamma=1}^N w_\alpha \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} \left(\sum_{\beta=1}^N \varphi_{\widehat{\beta}, \alpha} h_x^{(\beta)}(\widehat{x}_i^\gamma) \right) \Delta x + \lambda_y \sum_{\alpha=1}^L \sum_{\gamma=1}^N w_\alpha \widehat{w}_\gamma v_{\alpha, \widehat{\gamma}} \left(\sum_{\beta=1}^N \varphi_{\alpha, \widehat{\beta}} h_y^{(\beta)}(\widehat{y}_j^\gamma) \right) \Delta y. \end{aligned} \tag{72}$$

Substituting Eq. (72) into the scheme Eq. (27), it gives

$$\begin{aligned} & \overline{\varphi}_{i,j}^{n+1} \\ &= \gamma_x \sum_{\alpha=1}^L w_\alpha \widehat{w}_1 \left[\left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_1} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} h_x^{(1)}(\widehat{x}_i^\gamma) \right) \Delta x - \frac{\lambda_x}{\gamma_x \widehat{w}_1} u_{i-\frac{1}{2}, \alpha}^+ \varphi_{i-\frac{1}{2}, \alpha}^+ + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \widehat{f}(\varphi_{i-\frac{1}{2}, \alpha}^-, \varphi_{i-\frac{1}{2}, \alpha}^+) \right] \\ &+ \gamma_x \sum_{\alpha=1}^L \sum_{\beta=2}^{N-1} w_\alpha \widehat{w}_\beta \left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_\beta} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} h_x^{(\beta)}(\widehat{x}_i^\gamma) \right) \Delta x \varphi_{\widehat{\beta}, \alpha} \\ &+ \gamma_x \sum_{\alpha=1}^L w_\alpha \widehat{w}_N \left[\left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_N} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} h_x^{(N)}(\widehat{x}_i^\gamma) \right) \Delta x + \frac{\lambda_x}{\gamma_x \widehat{w}_N} u_{i+\frac{1}{2}, \alpha}^- \varphi_{i+\frac{1}{2}, \alpha}^- - \frac{\lambda_x}{\gamma_x \widehat{w}_N} \widehat{f}(\varphi_{i+\frac{1}{2}, \alpha}^-, \varphi_{i+\frac{1}{2}, \alpha}^+) \right] \\ &+ \gamma_y \sum_{\alpha=1}^L w_\alpha \widehat{w}_1 \left[\left(1 - \frac{\lambda_y}{\gamma_y \widehat{w}_1} \sum_{\gamma=1}^N \widehat{w}_\gamma v_{\alpha, \widehat{\gamma}} h_y^{(1)}(\widehat{y}_j^\gamma) \right) \Delta y - \frac{\lambda_y}{\gamma_y \widehat{w}_1} v_{\alpha, j-\frac{1}{2}}^+ \varphi_{\alpha, j-\frac{1}{2}}^+ + \frac{\lambda_y}{\gamma_y \widehat{w}_1} \widehat{g}(\varphi_{\alpha, j-\frac{1}{2}}^-, \varphi_{\alpha, j-\frac{1}{2}}^+) \right] \\ &+ \gamma_y \sum_{\alpha=1}^L \sum_{\beta=2}^{N-1} w_\alpha \widehat{w}_\beta \left(1 - \frac{\lambda_y}{\gamma_y \widehat{w}_\beta} \sum_{\gamma=1}^N \widehat{w}_\gamma v_{\alpha, \widehat{\gamma}} h_y^{(\beta)}(\widehat{y}_j^\gamma) \right) \Delta y \varphi_{\alpha, \widehat{\beta}} \\ &+ \gamma_y \sum_{\alpha=1}^L w_\alpha \widehat{w}_N \left[\left(1 - \frac{\lambda_y}{\gamma_y \widehat{w}_N} \sum_{\gamma=1}^N \widehat{w}_\gamma v_{\alpha, \widehat{\gamma}} h_y^{(N)}(\widehat{y}_j^\gamma) \right) \Delta y + \frac{\lambda_y}{\gamma_y \widehat{w}_N} v_{\alpha, j+\frac{1}{2}}^- \varphi_{\alpha, j+\frac{1}{2}}^- - \frac{\lambda_y}{\gamma_y \widehat{w}_N} \widehat{g}(\varphi_{\alpha, j+\frac{1}{2}}^-, \varphi_{\alpha, j+\frac{1}{2}}^+) \right]. \end{aligned} \tag{73}$$

Let us introduce the following formal formulations

$$\begin{aligned} H_{x, \alpha}^{(1)} &= \left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_1} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} h_x^{(1)}(\widehat{x}_i^\gamma) \right) \Delta x - \frac{\lambda_x}{\gamma_x \widehat{w}_1} u_{i-\frac{1}{2}, \alpha}^+ \varphi_{i-\frac{1}{2}, \alpha}^+ + \frac{\lambda_x}{\gamma_x \widehat{w}_1} \widehat{f}(\varphi_{i-\frac{1}{2}, \alpha}^-, \varphi_{i-\frac{1}{2}, \alpha}^+), \\ H_{x, \alpha}^{(\beta)} &= \left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_\beta} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} h_x^{(\beta)}(\widehat{x}_i^\gamma) \right) \Delta x \varphi_{\widehat{\beta}, \alpha}, \quad 2 \leq \beta \leq N-1 \\ H_{x, \alpha}^{(3)} &= \left(1 - \frac{\lambda_x}{\gamma_x \widehat{w}_N} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma}, \alpha} h_x^{(N)}(\widehat{x}_i^\gamma) \right) \Delta x + \frac{\lambda_x}{\gamma_x \widehat{w}_N} u_{i+\frac{1}{2}, \alpha}^- \varphi_{i+\frac{1}{2}, \alpha}^- - \frac{\lambda_x}{\gamma_x \widehat{w}_N} \widehat{f}(\varphi_{i+\frac{1}{2}, \alpha}^-, \varphi_{i+\frac{1}{2}, \alpha}^+). \end{aligned} \tag{74}$$

Plugging the expression of $\widehat{f}(\cdot, \cdot)$ Eq. (25) into Eq. (74), it gives

$$\begin{aligned}
 H_{x,\alpha}^{(1)} &= \left[1 - \frac{\lambda_x}{\gamma_x \widehat{w}_1} \left(\sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma},\alpha} h_x^{(1)}(\widehat{x}_i^\gamma) \Delta x + \frac{1}{2} u_{i-\frac{1}{2},\alpha}^+ + \frac{1}{2} a_x \right) \right] \varphi_{i-\frac{1}{2},\alpha}^+ + \frac{1}{2} \frac{\lambda_x}{\gamma_x \widehat{w}_1} (u_{i-\frac{1}{2},\alpha}^+ + a_x) \varphi_{i-\frac{1}{2},\alpha}^-, \\
 H_{x,\alpha}^{(\beta)} &= \left[1 - \frac{\lambda_x}{\gamma_x \widehat{w}_\beta} \sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma},\alpha} h_x^{(\beta)}(\widehat{x}_i^\gamma) \Delta x \right] \varphi_{\beta,\alpha}, \quad 2 \leq \beta \leq N-1 \\
 H_{x,\alpha}^{(3)} &= \left[1 - \frac{\lambda_x}{\gamma_x \widehat{w}_N} \left(\sum_{\gamma=1}^N \widehat{w}_\gamma u_{\widehat{\gamma},\alpha} h_x^{(N)}(\widehat{x}_i^\gamma) \Delta x - \frac{1}{2} u_{i+\frac{1}{2},\alpha}^- + \frac{1}{2} a_x \right) \right] \varphi_{i+\frac{1}{2},\alpha}^- - \frac{1}{2} \frac{\lambda_x}{\gamma_x \widehat{w}_N} (u_{i+\frac{1}{2},\alpha}^- - a_x) \varphi_{i+\frac{1}{2},\alpha}^+.
 \end{aligned} \tag{75}$$

Then, it is easy to verify that under the CFL condition Eq. (42), $H_{x,\alpha}^{(1)}$, $H_{x,\alpha}^{(\beta)}$ ($2 \leq \beta \leq N-1$) and $H_{x,\alpha}^{(N)}$ are monotonically increasing with respect to their arguments, i.e.,

$$\frac{\partial H_{x,\alpha}^{(1)}(\varphi_{i-\frac{1}{2},\alpha}^-, \varphi_{i-\frac{1}{2},\alpha}^+)}{\partial \varphi_{i-\frac{1}{2},\alpha}^\pm} \geq 0, \quad \frac{\partial H_{x,\alpha}^{(\beta)}(\varphi_{\beta,\alpha})}{\partial \varphi_{\beta,\alpha}} \geq 0, \quad \frac{\partial H_{x,\alpha}^{(N)}(\varphi_{i+\frac{1}{2},\alpha}^-, \varphi_{i+\frac{1}{2},\alpha}^+)}{\partial \varphi_{i+\frac{1}{2},\alpha}^\pm} \geq 0. \tag{76}$$

Moreover, we have

$$\begin{aligned}
 \widehat{w}_1 H_{x,\alpha}^{(1)}(m, m) + \sum_{\beta=2}^{N-1} \widehat{w}_\beta H_{x,\alpha}^{(\beta)}(m) + \widehat{w}_3 H_{x,\alpha}^{(N)}(m, m) &= m, \\
 \widehat{w}_1 H_{x,\alpha}^{(1)}(M, M) + \sum_{\beta=2}^{N-1} \widehat{w}_\beta H_{x,\alpha}^{(\beta)}(M) + \widehat{w}_3 H_{x,\alpha}^{(N)}(M, M) &= M.
 \end{aligned} \tag{77}$$

Similar results can be obtained for

$$\begin{aligned}
 H_{\alpha,y}^{(1)} &= \left[1 - \frac{\lambda_y}{\gamma_y \widehat{w}_1} \sum_{\gamma=1}^N \widehat{w}_\gamma v_{\alpha,\widehat{\gamma}} h_y^{(1)}(\widehat{y}_j^\gamma) \Delta y - \frac{\lambda_y}{\gamma_y \widehat{w}_1} v_{\alpha,j-\frac{1}{2}}^+ \right] \varphi_{\alpha,j-\frac{1}{2}}^+ + \frac{\lambda_y}{\gamma_y \widehat{w}_1} \widehat{g}(\varphi_{\alpha,j-\frac{1}{2}}^-, \varphi_{\alpha,j-\frac{1}{2}}^+), \\
 H_{\alpha,y}^{(2)} &= \left[1 - \frac{\lambda_y}{\gamma_y \widehat{w}_\beta} \sum_{\gamma=1}^N \widehat{w}_\gamma v_{\alpha,\widehat{\gamma}} h_y^{(\beta)}(\widehat{y}_j^\gamma) \Delta y \right] \varphi_{\alpha,\widehat{\beta}}, \\
 H_{\alpha,y}^{(3)} &= \left[1 - \frac{\lambda_y}{\gamma_y \widehat{w}_N} \sum_{\gamma=1}^N \widehat{w}_\gamma v_{\alpha,\widehat{\gamma}} h_y^{(N)}(\widehat{y}_j^\gamma) \Delta y + \frac{\lambda_y}{\gamma_y \widehat{w}_N} v_{\alpha,j+\frac{1}{2}}^- \right] \varphi_{\alpha,j+\frac{1}{2}}^- - \frac{\lambda_y}{\gamma_y \widehat{w}_N} \widehat{g}(\varphi_{\alpha,j+\frac{1}{2}}^-, \varphi_{\alpha,j+\frac{1}{2}}^+).
 \end{aligned} \tag{78}$$

Therefore, $\overline{\varphi}_{i,j}^{n+1} \in [m, M]$ under the CFL condition Eq. (42) since it is a convex combination of all the points values involved.

References

1. Bahbah, C., Khalloufi, M., Larcher, A., Mesri, Y., Coupez, T., Valette, R., Hachem, E.: Conservative and adaptive level-set method for the simulation of two-fluid flows. *Comput. Fluids* **191**, 104223 (2019)

2. Yang, X., James, A.J., Lowengrub, J., Zheng, X., Cristini, V.: An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *J. Comput. Phys.* **217**, 364–394 (2006)
3. Osher, S.J., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**, 12–49 (1988)
4. Sethian, J.A., Smereka, P.: Level set methods for fluid interfaces. *Annu. Rev. Fluid Mech.* **35**, 341–372 (2003)
5. Osher, S., Fedkiw, R.P.: Level set methods: an overview and some recent results. *J. Comput. Phys.* **169**, 463–502 (2001)
6. Olsson, E., Kreiss, G.: A conservative level set method for two phase flow. *J. Comput. Phys.* **210**, 225–246 (2005)
7. Olsson, E., Kreiss, G., Zahedi, S.: A conservative level set method for two phase flow II. *J. Comput. Phys.* **225**, 785–807 (2007)
8. Owkes, M., Desjardins, O.: A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. *J. Comput. Phys.* **249**, 275–302 (2013)
9. W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Los Alamos Scientific Laboratory Report, LA-UR-73-479, 1973
10. Cockburn, B., Shu, C.-W.: TVD Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math. Comput.* **52**, 411–435 (1989)
11. Cockburn, B., Shu, C.-W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* **16**, 173–261 (2001)
12. Dafermos, C.M.: Hyperbolic conservation laws in continuum physics. Springer, New York (2000)
13. Zhang, X., Shu, C.-W.: On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.* **229**, 3091–3120 (2010)
14. Zhang, X., Shu, C.-W.: Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. *Proc. R. Soc. A* **467**, 2752–2776 (2011)
15. Liu, X.-D., Osher, S.: Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes I. *SIAM J. Numer. Anal.* **33**(2), 760–779 (2011)
16. Zhang, X., Shu, C.-W.: On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *J. Comput. Phys.* **229**, 8919–8934 (2010)
17. Liang, C., Xu, Z.: Parametrized maximum principle preserving flux limiters for high order schemes solving multi-dimensional scalar hyperbolic conservation laws. *J. Sci. Comput.* **58**, 41–60 (2014)
18. Xu, Z.: Parametrized maximum principle preserving flux limiters for high order scheme solving hyperbolic conservation laws: one-dimensional scalar problem. *Math. Comput.* **83**, 2213–2238 (2014)
19. Xiong, T., Qiu, J.-M., Xu, Z.: A parametrized maximum principle preserving flux limiter for finite difference RK-WENO schemes with applications in incompressible flows. *J. Comput. Phys.* **252**, 310–331 (2013)
20. Zhang, Y., Zhang, X., Shu, C.-W.: Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection-diffusion equations on triangular meshes. *J. Comput. Phys.* **234**, 295–315 (2013)
21. Chen, Z., Huang, H., Yan, J.: Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes. *J. Comput. Phys.* **308**, 198–217 (2016)
22. Yu, H., Liu, H.L.: Third order maximum-principle-satisfying DG schemes for convection-diffusion problems with anisotropic diffusivity. *J. Comput. Phys.* **391**, 14–36 (2019)
23. Du, J., Yang, Y.: Maximum-principle-preserving third-order local discontinuous Galerkin method for convection-diffusion equations on overlapping meshes. *J. Comput. Phys.* **377**, 117–141 (2019)
24. Xiong, T., Qiu, J., Xu, Z.: High order maximum principle preserving discontinuous Galerkin method for convection-diffusion equations. *SIAM J. Sci. Comput.* **37**, A583–A608 (2015)
25. Liu, J.-G., Shu, C.-W.: A high-order discontinuous Galerkin method for 2D incompressible flows. *J. Comput. Phys.* **160**, 577–59 (2000)
26. Li, M., Dong, H., Hu, B., Xu, L.: Maximum-principle-satisfying and positivity-preserving high order central DG methods on unstructured overlapping meshes for two-dimensional hyperbolic conservation laws. *J. Sci. Comput.* **79**, 1361–1388 (2019)
27. Crandall, M., Majda, A.: Monotone difference approximations for scalar conservation laws. *Math. Comput.* **34**, 1–21 (1980)
28. Cheng, J., Zhang, F., Liu, T.G.: A discontinuous Galerkin method for the simulation of compressible gas-gas and gas-water two-medium flows. *J. Comput. Phys.* **403**, 109059 (2020)
29. Cheng, J., Zhang, F., Liu, T.G.: A quasi-conservative discontinuous Galerkin method for solving five equation model of compressible two-medium flows. *J. Sci. Comput.* **85**, 12 (2020)

30. Cheng, Y., Shu, C.-W.: A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *J. Comput. Phys.* **223**, 398–415 (2007)
31. Shu, C.-W.: Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.* **9**, 1073–1084 (1988)
32. Liu, H., Yan, J.: The direct discontinuous Galerkin (DDG) methods for diffusion problems. *SIAM J. Numer. Anal.* **47**, 475–698 (2009)
33. Liu, H., Yan, J.: The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections. *Commun. Comput. Phys.* **8**, 541–564 (2010)
34. Anderson, R., Dobrev, V., Kolev, T., Kuzmin, D., Quezada de Luna, M., Rieben, R., Tomov, V.: High-order local maximum principle preserving (MPP) discontinuous Galerkin finite element method for the transport equation. *J. Comput. Phys.* **334**, 102–124 (2017)
35. Zalesak, S.T.: Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31**, 335–362 (1979)
36. Bell, J.B., Colella, P., Glaz, H.M.: A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **85**, 257–283 (1989)
37. Enright, D., Fedkiw, R., Ferziger, J., Mitchell, I.: A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* **183**, 83–116 (2002)
38. Owkes, M., Desjardins, O.: A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. *J. Comput. Phys.* **249**, 275–30 (2013)
39. Guermond, J.L., Salgado, A.: A splitting method for the incompressible flows with variable density based on a pressure Poisson equation. *J. Comput. Phys.* **228**, 2834–2846 (2009)
40. Guermond, J.L., Quartapelle, L.: A projection FEM for variable density incompressible flows. *J. Comput. Phys.* **165**, 167–188 (2000)
41. Li, Y., Mei, J.Q., Ge, J.T., Shi, F.: A new fractional time-stepping method for variable density incompressible flows. *J. Comput. Phys.* **242**, 124–137 (2013)
42. Tryggvason, G.: Numerical simulations of the Rayleigh-Taylor instability. *J. Comput. Phys.* **75**, 235–282 (1988)
43. Zhang, F., Cheng, J., Liu, T.G.: A direct discontinuous Galerkin method for the incompressible Navier-Stokes equations on arbitrary grids. *J. Comput. Phys.* **380**, 269–294 (2019)
44. Zhang, F., Cheng, J., Liu, T.G.: A high-order discontinuous Galerkin method for the incompressible Navier-Stokes equations on arbitrary grids. *Int. J. Numer. Meth. Fluids* **90**, 217–246 (2019)
45. Zhang, F., Cheng, J., Liu, T.G.: A reconstructed discontinuous Galerkin method for incompressible flows on arbitrary grids. *J. Comput. Phys.* **418**, 109580 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.