Journal of Computational Physics 289 (2015) 181–195

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Positivity preserving high-order local discontinuous Galerkin method for parabolic equations with blow-up solutions

Li Guo [a,b], Yang Yang [b,*]

[a] School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, PR China
[b] Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931, United States

ARTICLE INFO

Article history:
Received 13 October 2014
Received in revised form 17 January 2015
Accepted 18 February 2015
Available online 2 March 2015

Keywords:
Positivity-preserving
Blow-up solutions
Local discontinuous Galerkin method
Dirichlet boundary conditions

ABSTRACT

In this paper, we apply positivity-preserving local discontinuous Galerkin (LDG) methods to solve parabolic equations with blow-up solutions. This model is commonly used in combustion problems. However, previous numerical methods are mainly based on a second order finite difference method. This is because the positivity-preserving property can hardly be satisfied for high-order ones, leading to incorrect blow-up time and blow-up sets. Recently, we have applied discontinuous Galerkin methods to linear hyperbolic equations involving δ-singularities and obtained good approximations. For nonlinear problems, some special limiters are constructed to capture the singularities precisely. We will continue this approach and study parabolic equations with blow-up solutions. We will construct special limiters to keep the positivity of the numerical approximations. Due to the Dirichlet boundary conditions, we have to modify the numerical fluxes and the limiters used in the schemes. Numerical experiments demonstrate that our schemes can capture the blow-up sets, and high-order approximations yield better numerical blow-up time.

1. Introduction

In this paper, we will develop and analyze the positivity-preserving high-order local discontinuous Galerkin methods for parabolic equations with reaction terms in one space dimension

$$\begin{aligned} &u_t = (u^\alpha)_{xx} + s(u), \quad x \in [x_a, x_b], \\ &u_0(x) = u(x,0) \geq 0, \quad x \in [x_a, x_b], \end{aligned} \tag{1}$$

as well as its two dimensional extension, where $\alpha \geq 1$ is a parameter and $s(u) \geq 0$. We consider Dirichlet boundary conditions

$$u(x_a, t) = g_1(t), \quad \text{and} \quad u(x_b, t) = g_2(t).$$

The source term $s(u)$ is chosen to be superlinear, i.e. $s(u) = u^m$ with $m \geq 1$. The initial condition $u = u_0(x)$ is assumed to be non-negative. By a maximum-principle, the exact solution $u \geq 0$ for $t \in (0, T)$. Here $(0, T)$ is the maximal time interval

* Corresponding author.
E-mail addresses: liguo@mtu.edu, lili2010@mail.ustc.edu.cn (L. Guo), yyang7@mtu.edu (Y. Yang).

http://dx.doi.org/10.1016/j.jcp.2015.02.041
0021-9991/

of existence of the solution u. The time T may be finite or infinite. When T is infinite, we say that the solution u exists globally. When T is finite, the solution u develops a singularity in finite time, namely

$$\lim_{t\to T^-} \|u(\cdot,t)\|_\infty = \infty,$$

where $\|u\|_\infty$ is the standard L^∞-norm of u on $[x_a, x_b]$. In this case, we say that the solution u blows up in finite time and the time T is called the blow-up time of the solution u, and $\{x \in [x_a, x_b] | u(x,t) \to \infty \text{ as } T \to T^-\}$ is called the blow-up set. In this paper, we assume the boundary is not included in the blow-up set. Therefore, the exact solution at the boundary should be smaller than the values nearby.

We can regard (1) as a mathematical model of combustion, and the unknown variable u can be interpreted as the temperature. We can rewrite (1) as

$$\begin{aligned} &u_t = (a(u)u_x)_x + s(u), \quad x \in [x_a, x_b], \\ &u(x,0) \geq 0, \quad x \in [x_a, x_b], \end{aligned} \tag{2}$$

with $a(u) = \alpha u^{\alpha-1}$, which can be considered as a nonlinear heat conduction coefficient of the medium. If $\alpha = 1$, (2) reduces to the well-studied semilinear heat equation [4]. Such problems have been investigated by many authors and existence and uniqueness of a classical solution have been proved (see e.g. [26,34,6,5,20,21,29,25,41] for some recent works). Under some assumptions, it is also shown that the classical solution blows up in finite time and the blow-up time has been estimated. Moreover, many numerical methods are also been studied in [11,8,9,28,13,22,35,2,7,31,10,1,27]. However, to the best knowledge of the authors, all the previous methods are based on second-order finite difference methods, which guarantee the positivity of the numerical approximations automatically under some special requirements for the meshes. For high-order methods, strong oscillations near the singularities may send physically positive quantities negative, and negative coefficient of the diffusion term will make the numerical solution blow up immediately, leading to wrong blow-up time and blow-up sets. In this paper, we would like to apply high-order positivity-preserving local discontinuous Galerkin methods to such problems, and numerically study the blow-up time, blow-up locations and the behavior of the solutions before blow-up. We will also use numerical experiments to demonstrate the significance of the positivity-preserving technique.

The study of the blow-up solutions is important and challenging. Since the exact solutions may not be smooth or bounded at finite time, the numerical schemes might yield poor approximations. Recently, we [38] have applied discontinuous Galerkin (DG) methods to obtain good approximations for PDEs with δ-singularity, one special unbounded singularity. In [38], we proved the superconvergence results for linear hyperbolic equations with singular initial data and singular source terms. Moreover, several numerical examples were also given in [38–40] to demonstrate the advantages of the DG scheme in approximating δ-singularities for both linear and nonlinear hyperbolic equations. In this paper, we will continue this approach, and use the high-order DG methods to solve parabolic PDEs with blow-up solutions.

The DG method was first introduced in 1973 by Reed and Hill [30], in the framework of neutron linear transport. Later, the method was applied by Johnson and Pitkäranta to a scalar linear hyperbolic equation and the L^p-norm error estimate was proved [24]. Subsequently, Cockburn et al. developed Runge–Kutta discontinuous Galerkin (RKDG) methods for hyperbolic conservation laws in a series of papers [16,15,14,17]. In [18], Cockburn and Shu first introduced the LDG method to solve the convection–diffusion equation. Their idea was motivated by Bassi and Rebay [3], where the compressible Navier–Stokes equations were successfully solved.

The idea of the positivity-preserving technique we would like to apply in this paper is different from the one used before in [43–45]. In [45], the authors construct second-order positivity-preserving DG schemes and argued that it was impossible to construct a third-order positivity-preserving DG scheme following the same approach. Recently, in [36], the author applied the flux limiter and constructed maximum-principle-satisfying finite difference methods, which further generalized to high-order positivity-preserving DG methods for convection diffusion problems. The DG method applied in [36] is based on Ultra-weak DG [12]. We will consider LDG method and study the positivity-preserving technique. Another related work can be found in [42], where the positivity-preserving technique for porous medium equations have been analyzed. However, only a second-order scheme was considered and the technique can hardly be extended to high-order schemes. Another crucial problem is the boundary treatment for the LDG method. Numerical experiments demonstrate that the general LDG method will degenerate accuracy when solving problems with Dirichlet boundary conditions. Following [19], we would like to add a penalty term in the flux at the boundary. With the penalty term, the convergence rate is optimal. Moreover, we will also theoretically prove that the penalty term is required for the positivity-preserving technique.

The organization of this paper is as follows. In Section 2, we will present the positivity-preserving high-order LDG methods in one and two space dimensions, and the implementation of the Dirichlet boundary condition. Some numerical experiments will be given in Section 3. We will end in Section 4 with some concluding remarks and remarks on future work.

2. Positivity-preserving high-order local discontinuous Galerkin methods

In this section, we present the positivity-preserving technique applied to the LDG methods for parabolic equations subject to Dirichlet boundary conditions. We first introduce the LDG methods and discuss how to enforce the Dirichlet boundary conditions. Then we describe how to apply the positivity-preserving flux limiter to the numerical fluxes in the scheme.

We discuss the Euler-forward time discretization, and the high-order ones are straightforward extendable. The flux limiter will be applied to guarantee the positivity of the numerical cell averages at time level $n+1$, provided the numerical approximations are positive at time level n. However, the numerical approximations may not be positive at time level $n+1$. Therefore, we would like to modify the numerical approximation with some slope limiters, keeping the cell averages untouched. We will show the procedure in one and two space dimensions.

2.1. Local discontinuous Galerkin method

In this subsection, we will develop the LDG method in one space dimension and study (2). To construct the LDG scheme, we consider a partition for the spatial domain $[x_a, x_b]$ given as

$$x_a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = x_b, \tag{3}$$

and define

$$I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$$

to be the cell. For simplicity, we use uniform mesh only, and the mesh size is denoted as Δx. The finite element space is given as

$$V_h = \left\{ u \in P^k(I_j) | j = 1, 2, \cdots, N \right\},$$

where $P^k(I_j)$ denotes the space of polynomials of degree at most k on the interval I_j. To construct the LDG scheme for (2), we introduce an auxiliary variable

$$q = a^\star(u)u_x, \quad \text{with} \quad a^\star(u) = \sqrt{a(u)},$$

then (2) can be written into a first-order system

$$u_t = (a^\star(u)q)_x + s(u), \tag{4}$$

$$q = (g(u))_x, \tag{5}$$

where $g(u) = \int^u a^\star(\tau)d\tau$. For simplicity, we omit the subscript h in Subsections 2.1 and 2.2, and use u and q for u_h and q_h, respectively. Then the general formulation of the LDG scheme is to find $u, q \in V_h$, such that for any test function $v, w \in V_h$

$$\int_{I_j} u_t v dx = -\int_{I_j} a^\star(u)q v_x dx + \widehat{(a^\star(u)q}v^-)_{j+\frac{1}{2}} - \widehat{(a^\star(u)q}v^+)_{j-\frac{1}{2}} + \int_{I_j} s(u)v dx, \tag{6}$$

$$\int_{I_j} q w dx = -\int_{I_j} g(u) w_x dx + \widehat{(g(u)}w^-)_{j+\frac{1}{2}} - \widehat{(g(u)}w^+)_{j-\frac{1}{2}}, \tag{7}$$

where $\widehat{a^\star(u)q}$ and $\widehat{g(u)}$ in (6)–(7) are the numerical fluxes which are defined at the cell interfaces. In this paper, we consider alternating fluxes only, i.e.

$$\widehat{(a^\star(u)q)}_{j+\frac{1}{2}} = \left(\frac{[g(u)]}{[u]}q^+\right)_{j+\frac{1}{2}} + \frac{C_{j+\frac{1}{2}}}{\Delta x}[u]_{j+\frac{1}{2}}, \quad \widehat{(g(u))}_{j+\frac{1}{2}} = g(u^-_{j+\frac{1}{2}}), \tag{8}$$

or

$$\widehat{(a^\star(u)q)}_{j+\frac{1}{2}} = \left(\frac{[g(u)]}{[u]}q^-\right)_{j+\frac{1}{2}} + \frac{C_{j+\frac{1}{2}}}{\Delta x}[u]_{j+\frac{1}{2}}, \quad \widehat{(g(u))}_{j+\frac{1}{2}} = g(u^+_{j+\frac{1}{2}}), \tag{9}$$

where $u^+_{j+\frac{1}{2}}$ is the right limit of u at $x = x_{j+\frac{1}{2}}$. Likewise for $u^-_{j+\frac{1}{2}}$, $q^+_{j+\frac{1}{2}}$ and $q^-_{j+\frac{1}{2}}$. Moreover, we denote $[u]_{j+\frac{1}{2}} = u^+_{j+\frac{1}{2}} - u^-_{j+\frac{1}{2}}$ as the jump of u at the cell interface $x_{j+\frac{1}{2}}$. $C_{j+\frac{1}{2}}$ is the coefficient for the penalty term. Due to the boundary condition, we have to take $u^-_{\frac{1}{2}} = g_1(t)$ and $u^+_{N+\frac{1}{2}} = g_2(t)$, i.e. we have to take (8) for $j = 0$ and (9) for $j = N$. In this paper, we would like to use (8) for all the other j's, and in practice, we can take $C_{j+\frac{1}{2}} = 0$, $j = 0, 1, \cdots, N-1$ (see Lemma 2.1). However, if we choose $C_{j+\frac{1}{2}} = 0$ for all j, the scheme is problematic. In [19], the author studied the steady-state problem, and argued that it is necessary to have all the C's to be zero expect the one at $x = x_{N+\frac{1}{2}}$, otherwise the numerical solution is not solvable. For time-dependent problems, even though the numerical solutions can be solved without the penalty term, numerical experiments demonstrated that, at $x = x_{N+\frac{1}{2}}$, the first-order numerical approximations are inconsistent with the exact solutions, even though they are sufficiently smooth. For high-order approximations, we cannot observe the inconsistency, but the order of accuracy may not be optimal. On this other hand, with the penalty, we will use numerical

experiments to demonstrate the optimality of the error estimates. Following the same idea in [19], we want $C_{N+\frac{1}{2}} > 0$. Therefore, the flux at the right boundary ($x = x_{N+\frac{1}{2}} = x_b$) is different from those elsewhere. We will discuss how to choose the parameter later.

2.2. Positivity-preserving technique

In this subsection, we will demonstrate the positivity-preserving technique. For simplicity we do not consider the contribution from the source, and assume $s(u) = 0$. We can make such an assumption because the source will never make negative contribution to the positivity of the cell averages for explicit time discretizations. The whole procedure can be divided into three steps. First, we need to prove the positivity of the first-order approximations. Then apply the flux limiter to high-order approximations to obtain positive cell averages. Finally, we modify the numerical approximations, keeping the cell averages untouched. With Euler forward time discretization, the first-order LDG scheme can be written as

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}((\widehat{a^\star(u^n)q^n})_{j+\frac{1}{2}} - (\widehat{a^\star(u^n)q^n})_{j-\frac{1}{2}}),$$
$$q_j^n = \frac{1}{\Delta x}(\widehat{g(u^n)}_{j+\frac{1}{2}} - \widehat{g(u^n)}_{j-\frac{1}{2}}).$$

Here Δt is the time mesh size. u_j^n, a constant, is the numerical approximation at time level n in cell I_j. Likewise for q_j^n. For simplicity, if we consider the numerical approximations at time level n, then the correspondence index will be omitted. If we choose the penalty parameters to be large enough, then there exists sufficient small Δt such that the first-order approximations are positive. The result is given below.

Lemma 2.1. *We can take $C_{N+\frac{1}{2}}$ and $C_{N-\frac{1}{2}}$ satisfy (12) and (13), and $\lambda = \frac{\Delta t}{\Delta x^2}$ satisfies (10), (11) and (14), then the numerical approximation from the first order scheme is positive.*

Proof. We consider three different cases. For convenience, we define $u_0 = g_1(t)$ and $u_{N+1} = g_2(t)$.

1. For $j = 1, 2, \cdots, N-2$, $\widehat{g(u)}_{j-\frac{1}{2}} = g(u_{j-1})$ and $q_j = \frac{1}{\Delta x}(g(u_j) - g(u_{j-1}))$. Therefore,
$$\begin{aligned} u_j^{n+1} &= u_j + \frac{\Delta t}{\Delta x}\left(\frac{g(u_{j+1}) - g(u_j)}{u_{j+1} - u_j} q_{j+1} - \frac{g(u_j) - g(u_{j-1})}{u_j - u_{j-1}} q_j\right) \\ &= u_j + \lambda\left(\frac{(g(u_{j+1}) - g(u_j))^2}{u_{j+1} - u_j} - \frac{(g(u_j) - g(u_{j-1}))^2}{u_j - u_{j-1}}\right) \\ &= u_j + \lambda\left(f_{j+\frac{1}{2}}^2(u_{j+1} - u_j) - f_{j-\frac{1}{2}}^2(u_j - u_{j-1})\right) \\ &= \lambda f_{j+\frac{1}{2}}^2 u_{j+1} + \left(1 - \lambda(f_{j+\frac{1}{2}}^2 + f_{j-\frac{1}{2}}^2)\right)u_j + \lambda f_{j-\frac{1}{2}}^2 u_{j-1}, \end{aligned}$$
where
$$f_{j+\frac{1}{2}} = \frac{g(u_{j+1}) - g(u_j)}{u_{j+1} - u_j}$$
is positive. Therefore, if we take
$$\lambda(f_{j+\frac{1}{2}}^2 + f_{j-\frac{1}{2}}^2) \le 1, \tag{10}$$
we have $u_j^{n+1} \ge 0$.
2. For $j = N-1$, $C_{j+\frac{1}{2}}$ may not be zero. Follow the same analysis above, we have
$$\begin{aligned} u_j^{n+1} &= \lambda f_{j+\frac{1}{2}}^2 u_{j+1} + \left(1 - \lambda(f_{j+\frac{1}{2}}^2 + f_{j-\frac{1}{2}}^2)\right)u_j + \lambda f_{j-\frac{1}{2}}^2 u_{j-1} + \lambda C_{j+\frac{1}{2}}(u_{j+1} - u_j) \\ &= \lambda\left(f_{j+\frac{1}{2}}^2 + C_{j+\frac{1}{2}}\right)u_{j+1} + \left(1 - \lambda(f_{j+\frac{1}{2}}^2 + f_{j-\frac{1}{2}}^2 + C_{j+\frac{1}{2}})\right)u_j + \lambda f_{j-\frac{1}{2}}^2 u_{j-1}. \end{aligned}$$
Therefore, if we take
$$\lambda(f_{j+\frac{1}{2}}^2 + f_{j-\frac{1}{2}}^2 + C_{j+\frac{1}{2}}) \le 1, \tag{11}$$
we have $u_j^{n+1} \ge 0$.

3. For $j = N$, the analysis is quite different.

$$q_N = \frac{g(u_{N+1}) - g(u_{N-1})}{\Delta x} = \frac{1}{\Delta x}\left(f_{N+\frac{1}{2}}(u_{N+1} - u_N) + f_{N-\frac{1}{2}}(u_N - u_{N-1})\right)$$

and

$$\begin{aligned} u_N^{n+1} &= u_N + \lambda\left(\left(f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}}\right)q_N\Delta x + C_{N+\frac{1}{2}}(u_{N+1} - u_N) - C_{N-\frac{1}{2}}(u_N - u_{N-1})\right) \\ &= A_{N+1}u_{N+1} + A_N u_N + A_{N-1}u_{N-1}, \end{aligned}$$

where

$$\begin{aligned} A_{N+1} &= \lambda\left(f_{N+\frac{1}{2}}(f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}}) + C_{N+\frac{1}{2}}\right), \\ A_N &= 1 - \lambda\left((f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}})^2 + C_{N-\frac{1}{2}} + C_{N+\frac{1}{2}}\right), \\ A_{N-1} &= \lambda\left(C_{N-\frac{1}{2}} - f_{N-\frac{1}{2}}(f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}})\right). \end{aligned}$$

Therefore, we have to take

$$C_{N-\frac{1}{2}} \geq f_{N-\frac{1}{2}}(f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}}), \tag{12}$$

$$C_{N+\frac{1}{2}} \geq -f_{N+\frac{1}{2}}(f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}}), \tag{13}$$

and

$$\lambda\left((f_{N+\frac{1}{2}} - f_{N-\frac{1}{2}})^2 + C_{N-\frac{1}{2}} + C_{N+\frac{1}{2}}\right) \leq 1, \tag{14}$$

to obtain $u_j^{n+1} > 0$. □

Remark 1. Since $f_{j+\frac{1}{2}}$ is positive, therefore in (12) and (13), we only need to have one nonzero parameter C. Based on the assumption that the blow-up never occurs at the boundary, the exact solution at the boundary should be bounded and is smaller than the values nearby. Therefore, we can take $C_{N-\frac{1}{2}} = 0$ and $C_{N+\frac{1}{2}}$ to be a bounded constant, and this will be verified by numerical experiments in Section 3.

Remark 2. If $\alpha = 1$, then $f_{j+\frac{1}{2}} = 1$ for all $j = 0, 1, \cdots, N$. Therefore, to construct positive numerical cell averages, we can take all the penalty parameters to be zero, and $\lambda \leq \frac{1}{2}$. However, due to the boundary effect, $C_{N+\frac{1}{2}} > 0$ is also required.

With the above lemma, we can proceed to construct positive numerical cell averages for high-order approximations. We take the test function $v = 1$ and divided by Δx on both sides of (6) to obtain the equation satisfied by the numerical cell averages $\bar{u}_j$

$$\bar{u}_j^{n+1} = \bar{u}_j^n + \frac{\Delta t}{\Delta x}(\hat{H}_{j+\frac{1}{2}} - \hat{H}_{j-\frac{1}{2}}), \tag{15}$$

where the flux $\hat{H} = \widehat{a^\star(u)q}$. The numerical cell averages computed form (15) might be negative, and we have to apply a positivity-preserving flux limiter on the flux $\hat{H}$. The idea is to apply Lemma 2.1 to construct a first-order numerical flux $\hat{h}$ based on the numerical cell average at time level n and $x = x_{j+\frac{1}{2}}$. The next step is to modify the numerical flux to obtain a new one

$$\tilde{H}_{j+\frac{1}{2}} = \theta_{j+\frac{1}{2}}(\hat{H}_{j+\frac{1}{2}} - \hat{h}_{j+\frac{1}{2}}) + \hat{h}_{j+\frac{1}{2}}. \tag{16}$$

The parameter $\theta_{j+\frac{1}{2}}$ is designed to ensure $\bar{u}_j^{n+1}$ to be positive. Actually, we can choose $\theta_{j+\frac{1}{2}}$ to be nonnegative, since if we take $\theta_{j+\frac{1}{2}} = 0$, then $\bar{u}_j^{n+1}$ is obtained from the first-order numerical scheme, which is proved to yield a positive numerical solution in Lemma 2.1. In [37], the author gave an idea to choose $\theta_{j+\frac{1}{2}}$. The goal is to have

$$\bar{u}_j^{n+1} = \bar{u}_j^n + \frac{\Delta t}{\Delta x}(\tilde{H}_{j+\frac{1}{2}} - \tilde{H}_{j-\frac{1}{2}}) > \varepsilon,$$

where $\varepsilon = 10^{-13}$ is a small positive number related to machine precision. The above equation further yields

$$\widetilde{\lambda}\theta_{j+\frac{1}{2}}F_{j+\frac{1}{2}} - \widetilde{\lambda}\theta_{j-\frac{1}{2}}F_{j-\frac{1}{2}} - \Gamma_j \geq 0, \tag{17}$$

where $\widetilde{\lambda} = \frac{\Delta t}{\Delta x}$, $F_{j\pm\frac{1}{2}} = \hat{H}_{j\pm\frac{1}{2}} - \hat{h}_{j\pm\frac{1}{2}}$ and $\Gamma_j = \varepsilon - (\bar{u}_j^n + \widetilde{\lambda}(\hat{h}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}}))$. The parameters $\theta_{j\pm\frac{1}{2}}$ can be obtained as follows [37]. We want

$$\theta_{j-\frac{1}{2}} \in [0, \Lambda_{-\frac{1}{2},I_j}], \quad \theta_{j+\frac{1}{2}} \in [0, \Lambda_{\frac{1}{2},I_j}],$$

where $\Lambda_{\pm\frac{1}{2},I_j}$ are designed to keep (17) correct.

- If $F_{j+\frac{1}{2}} \geq 0$ and $F_{j-\frac{1}{2}} \leq 0$, then
$$(\Lambda_{\frac{1}{2},I_j}, \Lambda_{-\frac{1}{2},I_j}) = (1, 1).$$
- If $F_{j+\frac{1}{2}} < 0$ and $F_{j-\frac{1}{2}} \leq 0$, then
$$(\Lambda_{\frac{1}{2},I_j}, \Lambda_{-\frac{1}{2},I_j}) = \Big(\min\Big(1, \frac{\Gamma_j}{\widetilde{\lambda} F_{j+\frac{1}{2}}}\Big), 1\Big).$$
- If $F_{j+\frac{1}{2}} \geq 0$ and $F_{j-\frac{1}{2}} > 0$, then
$$(\Lambda_{\frac{1}{2},I_j}, \Lambda_{-\frac{1}{2},I_j}) = \Big(1, \min\Big(1, -\frac{\Gamma_j}{\widetilde{\lambda} F_{j-\frac{1}{2}}}\Big)\Big).$$
- If $F_{j+\frac{1}{2}} < 0$ and $F_{j-\frac{1}{2}} > 0$, then
  - If Eq. (17) is satisfied with $(\theta_{j+\frac{1}{2}}, \theta_{j-\frac{1}{2}}) = (1, 1)$, then
  $$(\Lambda_{\frac{1}{2},I_j}, \Lambda_{-\frac{1}{2},I_j}) = (1, 1).$$
  - If not, then
  $$(\Lambda_{\frac{1}{2},I_j}, \Lambda_{-\frac{1}{2},I_j}) = \Big(\frac{\Gamma_j}{(\widetilde{\lambda} F_{j+\frac{1}{2}} - \widetilde{\lambda} F_{j-\frac{1}{2}})}, \frac{\Gamma_j}{(\widetilde{\lambda} F_{j+\frac{1}{2}} - \widetilde{\lambda} F_{j-\frac{1}{2}})}\Big).$$

The local parameter $\theta_{j+\frac{1}{2}}$ can be chosen as

$$\theta_{j+\frac{1}{2}} = \min(\Lambda_{\frac{1}{2},I_j}, \Lambda_{-\frac{1}{2},I_{j+1}}).$$

Follow the above steps, we have $\bar{u}_j^{n+1} > 0$, provided $u_j^n > 0$ where u_j^n is the polynomial at most degree k in the cell I_j at the time level n.

Finally, we have to modify the numerical approximations at time level $n+1$, since they may not be positive, even though the cell averages are positive. The idea is to use the modified $\tilde{u}_j^{n+1}$ to replace the numerical solution u_j^{n+1}, and $\tilde{u}_j^{n+1}$ can be obtained as follows. In the cell I_j

$$\tilde{u}_j^{n+1} = \bar{u}_j^{n+1} + \Theta_j(u_j^{n+1} - \bar{u}_j^{n+1}) \geq \varepsilon,$$

where

$$\Theta_j = \min\Big(\frac{\bar{u}_j^{n+1} - \varepsilon}{\bar{u}_j^{n+1} - m_j^{n+1}}, 1\Big),$$

with

$$m_j^{n+1} = \min_{x\in I_j} u_j^{n+1}(x).$$

In [44], the author argued that such a limiter does not degenerate accuracy.

2.3. High-order time discretizations

All the previous analyses are based on Euler-forward time discretization. However, we can also apply high-order Strong-Stability-Preserving (SSP) time discretizations to solve the ODE system $\mathbf{u}_t = \mathbf{L}(\mathbf{u})$. More details of these time discretizations can be found in [33,32,23]. In this paper, we use the third-order SSP Runge–Kutta method [33]

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + \Delta t \mathbf{L}(\mathbf{u}^n),\\
\mathbf{u}^{(2)} &= \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\left(\mathbf{u}^{(1)} + \Delta t \mathbf{L}(\mathbf{u}^{(1)})\right),\\
\mathbf{u}^{n+1} &= \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\left(\mathbf{u}^{(2)} + \Delta t \mathbf{L}(\mathbf{u}^{(2)})\right),
\end{aligned} \tag{18}$$

and the third-order SSP multi-step method [32]

$$\mathbf{u}^{n+1} = \frac{16}{27}\left(\mathbf{u}^n + 3\Delta t\mathbf{L}(\mathbf{u}^n)\right) + \frac{11}{27}\left(\mathbf{u}^{n-3} + \frac{12}{11}\Delta t\mathbf{L}(\mathbf{u}^{n-3})\right). \tag{19}$$

Since an SSP time discretization is a convex combination of Euler forwards, by using the limiter mentioned in Section 2.2, the numerical solutions obtained from the fully-discrete scheme are also positive.

2.4. Two dimensional case

In this subsection, we will develop the positivity-preserving LDG method in two space dimensions, and study the following problem:

$$\begin{aligned} &u_t = (u^\alpha)_{xx} + (u^\beta)_{yy} + u^m, \quad (x, y) \in [x_a, x_b] \times [y_a, y_b], \\ &u(x, y, 0) \geq 0, \quad (x, y) \in [x_a, x_b] \times [y_a, y_b]. \end{aligned} \tag{20}$$

We consider Dirichlet boundary conditions

$$u(x_a, y, t) = g_a^x(y, t), \quad u(x_b, y, t) = g_b^x(y, t), \quad u(x, y_a, t) = g_a^y(x, t), \quad u(x, y_b, t) = g_b^y(x, t).$$

The rectangular domain $[x_a, x_b] \times [y_a, y_b]$ can be discretized into $N_x \times N_y$ rectangular meshes

$$x_a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N_x-\frac{1}{2}} < x_{N_x+\frac{1}{2}} = x_b, \quad y_a = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N_y-\frac{1}{2}} < y_{N_y+\frac{1}{2}} = y_b, \tag{21}$$

with the cell $K_{i,j} = I_i \times J_j$, where $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and $J_j = [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. We also consider uniform meshes only, and the mesh size in x and y directions are denoted as Δx and Δy, respectively. For simplicity, if we consider the cell $K_{i,j}$, then the subscript will be omitted. We define the finite element space as

$$V_h = \{u_h : u_h|_K \in \mathcal{P}^k(K), 1 \leq i \leq N_x, 1 \leq j \leq N_y\},$$

where $\mathcal{P}^k(K)$ denotes the polynomials of degree at most k on the element K.

To construct the local discontinuous Galerkin method, firstly we need to rewrite (20) into a first-order system as follows

$$u_t = (a^\star(u)p)_x + (b^\star(u)q)_y + u^m, \tag{22}$$

$$p = (f(u))_x, \tag{23}$$

$$q = (g(u))_y, \tag{24}$$

where $a^\star(u) = \sqrt{\alpha u^{\alpha-1}}$, $b^\star(u) = \sqrt{\beta u^{\beta-1}}$, $f(u) = \int^u a^\star(s)ds$ and $g(u) = \int^u b^\star(s)ds$. In this subsection, we also denote the numerical solution u_h, p_h and q_h by u, p and q, respectively. Then the general formulation of the LDG scheme is to find $u, p, q \in V_h$, such that

$$\begin{aligned} \int_K u_t v dxdy = &- \int_K a^\star(u)pv_x dxdy + \int_{J_j} ((\widehat{a^\star(u)p}v^-)_{i+\frac{1}{2},j} - (\widehat{a^\star(u)p}v^+)_{i-\frac{1}{2},j})dy \\ &- \int_K b^\star(u)qv_y dxdy + \int_{I_i} ((\widehat{b^\star(u)q}v^-)_{i,j+\frac{1}{2}} - (\widehat{b^\star(u)q}v^+)_{i,j-\frac{1}{2}})dx, + \int_K u^m v dxdy \end{aligned} \tag{25}$$

$$\int_K pw dxdy = - \int_K f(u)w_x dxdy + \int_{J_j} ((\widehat{f(u)}w^-)_{i+\frac{1}{2},j} - (\widehat{f(u)}w^+)_{i-\frac{1}{2},j})dy, \tag{26}$$

$$\int_K q\tau dxdy = - \int_K g(u)\tau_y dxdy + \int_{I_i} ((\widehat{g(u)}\tau^-)_{i,j+\frac{1}{2}} - (\widehat{g(u)}\tau^+)_{i,j-\frac{1}{2}})dx, \tag{27}$$

for any test function $v, w, \tau \in P^k(K)$ and $i = 1, \cdots, N_x$, $j = 1, \cdots, N_y$. Here $\widehat{a^\star(u)p}$, $\widehat{b^\star(u)q}$, $\widehat{f(u)}$ and $\widehat{g(u)}$ in (25)–(27) are the numerical fluxes which are functions defined on the cell interfaces and should be designed to ensure stability. In this paper, the fluxes are chosen to be

$$(\widehat{a^\star(u)p})_{i+\frac{1}{2},j} = \left(\frac{[f(u)]}{[u]}p^+\right)_{i+\frac{1}{2},j} + \frac{C_{i+\frac{1}{2},j}}{\Delta x}[u]_{i+\frac{1}{2},j}, \quad (\widehat{f(u)})_{i+\frac{1}{2},j} = f(u^-_{i+\frac{1}{2},j}), \tag{28}$$

$$(\widehat{b^\star(u)q})_{i,j+\frac{1}{2}} = \left(\frac{[g(u)]}{[u]}q^+\right)_{i,j+\frac{1}{2}} + \frac{C_{i,j+\frac{1}{2}}}{\Delta y}[u]_{i,j+\frac{1}{2}}, \quad (\widehat{g(u)})_{i,j+\frac{1}{2}} = g(u^-_{i,j+\frac{1}{2}}), \tag{29}$$

or

$$(\widehat{a^\star(u)p})_{i+\frac{1}{2},j} = \left(\frac{[f(u)]}{[u]}p^-\right)_{i+\frac{1}{2},j} + \frac{C_{i+\frac{1}{2},j}}{\Delta x}[u]_{i+\frac{1}{2},j}, \quad (\widehat{f(u)})_{i+\frac{1}{2},j} = f(u^+_{i+\frac{1}{2},j}), \tag{30}$$

$$(\widehat{b^\star(u)q})_{i,j+\frac{1}{2}} = \left(\frac{[g(u)]}{[u]}q^-\right)_{i,j+\frac{1}{2}} + \frac{C_{i,j+\frac{1}{2}}}{\Delta y}[u]_{i,j+\frac{1}{2}}, \quad (\widehat{g(u)})_{i,j+\frac{1}{2}} = g(u^+_{i,j+\frac{1}{2}}), \tag{31}$$

where $u^+_{i+\frac{1}{2},j} = u(x^+_{i+\frac{1}{2}}, y)$ is the limit from the right cell $K_{i+1,j}$ and $u^+_{i,j+\frac{1}{2}} = u(x, y^+_{i+\frac{1}{2}})$ is the limit from the cell $K_{i,j+1}$. Likewise for all the other limits. Due to the Dirichlet boundary condition, we will use (28) and (29) for $i = 0, \cdots, N_x - 1$ and $j = 0, \cdots, N_y - 1$, and use (30) and (31) for $i = N_x$ and $j = N_y$, respectively. $C_{i+\frac{1}{2},j}$ and $C_{i,j+\frac{1}{2}}$ are non-negative constants. Similar to the one dimensional case, in practice most of the C's are zero, except those with $i = N_x - 1, N_x$ and $j = N_y - 1, N_y$. $[u]_{i+\frac{1}{2},j} = u^+_{i+\frac{1}{2},j} - u^-_{i+\frac{1}{2},j}$ and $[u]_{i,j+\frac{1}{2}} = u^+_{i,j+\frac{1}{2}} - u^-_{i,j+\frac{1}{2}}$ are the jump of u at the cell interface $\{x_{i+\frac{1}{2}}\} \times J_j$ and $I_i \times \{y_{j+\frac{1}{2}}\}$, respectively.

Then we will present the positivity-preserving technique. Since the source term u^m would not make negative contribution, we can assume the source to be zero for simplicity. We consider high-order LDG schemes only and want to keep the positivity of cell averages of u, i.e. $\forall i, j$

$$\bar{u}_{i,j} = \frac{1}{\Delta x \Delta y}\int_{K_{i,j}} u dx dy \geq 0.$$

The equation satisfied by the numerical cell averages is given as

$$\begin{aligned}\frac{d}{dt}\bar{u}_{i,j} &+ \frac{1}{\Delta x}\left(\frac{1}{\Delta y}\int_{I_j}\hat{H}(x_{i+\frac{1}{2}}, y)dy - \frac{1}{\Delta y}\int_{I_j}\hat{H}(x_{i-\frac{1}{2}}, y)dy\right)\\ &+ \frac{1}{\Delta y}\left(\frac{1}{\Delta x}\int_{I_i}\hat{G}(x, y_{j+\frac{1}{2}})dx - \frac{1}{\Delta x}\int_{I_i}\hat{G}(x, y_{j-\frac{1}{2}})dx\right) = 0,\end{aligned} \tag{32}$$

where the fluxes $\hat{H}(x_{i+\frac{1}{2}}, y) = -(\widehat{a^\star(u)p})_{i+\frac{1}{2},j}$ and $\hat{G}(x, y_{j+\frac{1}{2}}) = -(\widehat{b^\star(u)q})_{i,j+\frac{1}{2}}$. Actually p, q are functions of u since p, q can be computed from Eqs. (26)–(27). Therefore $\hat{H}$ and $\hat{G}$ are functions of u. Then we apply the positivity-preserving flux limiter on the fluxes $\hat{H}$ and $\hat{G}$.

Similar to the one dimensional case, we will apply Euler forward time discretization for (32) for simplicity,

$$\bar{u}^{n+1}_{i,j} = \bar{u}^n_{i,j} - \lambda_x(\hat{H}_{i+\frac{1}{2},j} - \hat{H}_{i-\frac{1}{2},j}) - \lambda_y(\hat{G}_{i,j+\frac{1}{2}} - \hat{G}_{i,j-\frac{1}{2}}), \tag{33}$$

where $\lambda_x = \frac{\Delta t}{\Delta x}$ and $\lambda_y = \frac{\Delta t}{\Delta y}$. $\hat{H}_{i+\frac{1}{2},j}$ is the integral of the numerical flux $\hat{H}(x_{i+\frac{1}{2}}, y)$ along the cell interface $\{x_{i+\frac{1}{2}}\} \times I_j$. Likewise for $\hat{G}_{i,j+\frac{1}{2}}$. We compute the integral by a six-point Gaussian quadrature.

The next step is to modify the numerical fluxes $\hat{H}_{i+\frac{1}{2},j}$ and $\hat{G}_{i,j+\frac{1}{2}}$ to be

$$\tilde{H}_{i+\frac{1}{2},j} = \theta_{i+\frac{1}{2},j}(\hat{H}_{i+\frac{1}{2},j} - \hat{h}_{i+\frac{1}{2},j}) + \hat{h}_{i+\frac{1}{2},j}, \tag{34}$$

$$\tilde{G}_{i,j+\frac{1}{2}} = \theta_{i,j+\frac{1}{2}}(\hat{G}_{i,j+\frac{1}{2}} - \hat{g}_{i,j+\frac{1}{2}}) + \hat{g}_{i,j+\frac{1}{2}}, \tag{35}$$

where $\hat{h}_{i+\frac{1}{2},j}$ and $\hat{g}_{i,j+\frac{1}{2}}$ are the first order numerical fluxes based on the numerical cell averages at time level n which can obtain similarly as the one dimensional case. If we choose the parameters in the penalty terms to be large and the time mesh size Δt to be small, then the numerical approximations from the first-order scheme are positive. The proof is the same as in Lemma 2.1 with some minor changes, so we omit it. The parameter $\theta_{i+\frac{1}{2},j}$ and $\theta_{i,j+\frac{1}{2}}$ are designed such that $\bar{u}^{n+1}_{i,j}$ is greater than ε, where ε is a small positive number related to machine precision, then we have

$$\theta_{i+\frac{1}{2},j}F_{i+\frac{1}{2},j} + \theta_{i-\frac{1}{2},j}F_{i-\frac{1}{2},j} + \theta_{i,j+\frac{1}{2}}F_{i,j+\frac{1}{2}} + \theta_{i,j-\frac{1}{2}}F_{i,j-\frac{1}{2}} - \Gamma_{i,j} \geq 0, \tag{36}$$

where

$$F_{i+\frac{1}{2},j} = -\lambda_x(\hat{H}_{i+\frac{1}{2},j} - \hat{h}_{i+\frac{1}{2},j}),$$

$$F_{i-\frac{1}{2},j} = \lambda_x(\hat{H}_{i-\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}),$$

$$F_{i,j+\frac{1}{2}} = -\lambda_y(\hat{G}_{i,j+\frac{1}{2}} - \hat{g}_{i,j+\frac{1}{2}}),$$
$$F_{i,j-\frac{1}{2}} = \lambda_y(\hat{G}_{i,j-\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}),$$

and

$$\Gamma_{i,j} = \varepsilon - (\bar{u}^n_{i,j} - \lambda_x(\hat{h}_{i+\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}) - \lambda_y(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}})).$$

For the parameter $\theta_{i,j\pm\frac{1}{2}}$ and $\theta_{i\pm\frac{1}{2},j}$ can be obtained as follows by the equality (36) and the details can be found in [36]. Assume

$$\theta_{i+\frac{1}{2},j} \in [0, \Lambda_{R,i,j}], \quad \theta_{i-\frac{1}{2},j} \in [0, \Lambda_{L,i,j}], \quad \theta_{i,j+\frac{1}{2}} \in [0, \Lambda_{U,i,j}], \quad \theta_{i,j-\frac{1}{2}} \in [0, \Lambda_{D,i,j}],$$

where $\Lambda_{R,i,j}$, $\Lambda_{L,i,j}$, $\Lambda_{U,i,j}$ and $\Lambda_{D,i,j}$ are designed based on (36).

The procedure to determine the parameters $\Lambda_{R,i,j}$, $\Lambda_{L,i,j}$, $\Lambda_{U,i,j}$ and $\Lambda_{D,i,j}$ are as follows.

- Calculate the values of $F_{i+\frac{1}{2},j}$, $F_{i-\frac{1}{2},j}$, $F_{i,j+\frac{1}{2}}$ and $F_{i,j-\frac{1}{2}}$. If all the values are positive, then we take $\Lambda_{R,i,j} = \Lambda_{L,i,j} = \Lambda_{U,i,j} = \Lambda_{D,i,j} = 1$, otherwise go to the next step.
- Pick the negative values among $F_{i+\frac{1}{2},j}$, $F_{i-\frac{1}{2},j}$, $F_{i,j-\frac{1}{2}}$ and $F_{i,j-\frac{1}{2}}$. According to the negative values, the parameters can be defined. For instance, if $F_{i+\frac{1}{2},j} < 0$, $F_{i-\frac{1}{2},j} < 0$ and $F_{i,j+\frac{1}{2}} \geq 0$, $F_{i,j-\frac{1}{2}} \geq 0$, then

$$\Lambda_{R,i,j}, \Lambda_{L,i,j} = \min\Big(\frac{\Gamma_{i,j}}{F_{i+\frac{1}{2},j} + F_{i-\frac{1}{2},j}}, 1\Big),$$
$$\Lambda_{U,i,j}, \Lambda_{D,i,j} = 1.$$

  Other cases are similar to define.

The local parameter $\theta_{i+\frac{1}{2},j}$ and $\theta_{i,j+\frac{1}{2}}$ can be determined to be

$$\theta_{i+\frac{1}{2},j} = \min(\Lambda_{R,i,j}, \Lambda_{L,i+1,j}),$$
$$\theta_{i,j+\frac{1}{2}} = \min(\Lambda_{U,i,j}, \Lambda_{D,i,j+1}).$$

Follow the above step, we have $\bar{u}^{n+1}_{i,j} > 0$ provided $u^n_{i,j} > 0$ where $u^n_{i,j}$ is the numerical approximation in $K_{i,j}$ at time level n.

Finally, we have to modify the numerical simulations at time level $n+1$. The idea is the same as the one dimensional case. We use the modified $\tilde{u}^{n+1}_{i,j}$ to replace the numerical solution $u^{n+1}_{i,j}$, and $\tilde{u}^{n+1}_{i,j}$ can be obtained as follows. In each cell $K_{i,j}$,

$$\tilde{u}^{n+1}_{i,j} = \bar{u}^{n+1}_{i,j} + \Theta_{i,j}(u^{n+1}_{i,j} - \bar{u}^{n+1}_{i,j}) \geq \varepsilon,$$

where

$$\Theta_{i,j} = \min\Big(\frac{\bar{u}^{n+1}_{i,j} - \varepsilon}{\bar{u}^{n+1}_{i,j} - m^{n+1}_{i,j}}, 1\Big),$$

with

$$m^{n+1}_{i,j} = \min_{(x,y)\in K_{i,j}} u^{n+1}_{i,j}(x, y).$$

3. Numerical results

In this section, we give some numerical experiments to illustrate the good performance of the preserving positivity LDG method. We first test the effect of the penalty term given in the numerical fluxes in (8) and (9), and then consider other examples with blow-up solutions. For the examples with blow-up, ε should be a relatively small parameter, one example could be $\varepsilon = 10^{-13}u_{\max}$, where $u_{\max}$ is the global maximum of the solution. Moreover, we also need to take steps to determine the blow-up time numerically. Following [27], the time steps are chosen to be

$$\Delta t = \Delta x^2 \min\Big(cfl, \frac{u_{\max}}{s(u_{\max})}\Big) \quad \text{and} \quad \Delta t = \min(\Delta x^2, \Delta y^2)\min\Big(cfl, \frac{u_{\max}}{s(u_{\max})}\Big),$$

for one and two space dimensions, respectively. The parameter cfl is the constant to ensure the stability for the diffusion term and in these experiments we let $cfl = 0.1, 0.05, 0.01$ with the degree of polynomials $k = 0, 1, 2$ respectively. And all the penalty parameters are 1. Since the source $s(u) = u^m$ with $m > 1$, the larger the u, the smaller the Δt, and we consider the numerical blow-up occurs when $\Delta t < 10^{-13}$. In this section, to plot the figures, we use the numerical cell average to be the point value at the middle point in each cell.

Table 1
Example 3.1, accuracy test with penalty at time $T = 0.1$.

	N	L^2 error	order	L^∞ error	order
P^0	10	1.17E−01	–	2.70E−01	–
	20	5.90E−02	0.99	1.37E−01	0.98
	40	2.95E−02	1.00	6.86E−02	1.00
	80	1.48E−02	1.00	3.43E−02	1.00
P^1	10	4.69E−03	–	1.28E−02	–
	20	1.13E−03	2.05	3.23E−03	1.99
	40	2.82E−04	2.01	8.07E−04	2.00
	80	7.05E−05	2.00	2.02E−04	2.00
P^2	10	1.12E−04	–	3.32E−04	–
	20	1.42E−05	2.99	4.18E−05	2.99
	40	1.77E−06	3.00	5.23E−06	3.00
	80	2.22E−07	3.00	6.54E−07	3.00

Table 2
Example 3.1, accuracy test without penalty at time $T = 0.1$.

	N	L^2 error	order	L^∞ error	order
P^0	10	1.32E−01	–	3.68E−01	–
	20	6.27E−02	1.07	1.85E−01	0.99
	40	3.05E−02	1.04	9.29E−02	1.00
	80	1.48E−02	1.00	3.43E−02	1.00
P^1	10	5.04E−02	–	2.20E−01	–
	20	1.80E−02	1.49	1.11E−01	1.00
	40	6.38E−03	1.50	5.57E−02	1.00
	80	2.25E−03	1.50	2.79E−02	1.00
P^2	10	3.45E−04	–	1.72E−03	–
	20	3.22E−05	3.43	2.16E−04	2.98
	40	3.11E−06	3.37	2.70E−05	3.00
	80	3.17E−07	3.30	3.38E−06	3.00

3.1. One-dimensional case

In this subsection, we numerically solve (2) with different parameters α and m. First, we give one example to test the accuracy and the effect of the penalty term of the scheme and then show some blow-up experiments.

Example 3.1. We take $\alpha = m = 1$ and consider the following problem

$$u_t = u_{xx} + (\pi^2 - 1)u, \quad x \in [0, 1],$$
$$u(x, 0) = \sin(\pi x), \quad x \in [0, 1],$$

with the homogeneous Dirichlet boundary condition. The exact solution is

$$u(x, t) = e^{-t} \sin(\pi x).$$

We compute the L^2- and L^∞-norm error estimates with different degrees of polynomials. Table 1 shows the result with penalty $C_{N+\frac{1}{2}} = 1$ and the positivity-preserving technique while Table 2 gives that computed directly without any special modification of the scheme. From Table 1, we can observe that the scheme with penalty can achieve optimal rate of convergence, and the positivity-preserving flux limiter does not degenerate the accuracy. However, the scheme without penalty is problematic. If we plot the result obtained from the first-order numerical scheme with $N = 80$, we can observe the phenomenon of inconsistency. In Fig. 1, we can observe a tale at $x = 1$, i.e. the numerical approximation at the right boundary is bigger than that in the cell next to it. Such a phenomenon disappears if we add the penalty to the scheme.

Example 3.2. We take $\alpha = 1$, $m = 2$ and study the following problem

$$u_t = u_{xx} + u^2, \quad x \in [0, 1],$$

with the initial condition $u_0(x) = 20 \sin(\pi x)$ and homogeneous Dirichlet boundary condition.

We use this example to demonstrate that high-order numerical scheme yields better numerical blow-up time. In this example, we use polynomials of degree $k = 0, 1, 2$ and compute the blow-up time, which has been given in Table 3. From the

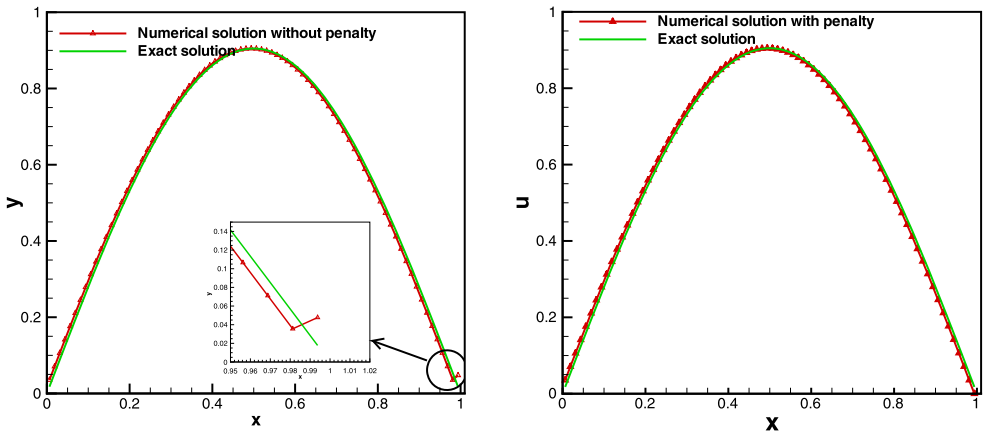

Fig. 1. Example 3.1: First-order numerical approximations with (right) and without (left) penalty. Other parameters are taken to be $N = 80$.

Table 3
Example 3.2, the blow-up time T with different meshes and different degrees of polynomials.

N	T		
	p^0	p^1	p^2
10	8.32162E−02	8.24457E−02	8.24406E−02
20	8.26315E−02	8.24391E−02	8.24375E−02
40	8.24856E−02	8.24376E−02	8.24374E−02
80	8.24493E−02	8.24374E−02	8.24373E−02
160	8.24399E−02	8.24371E−02	8.24371E−02

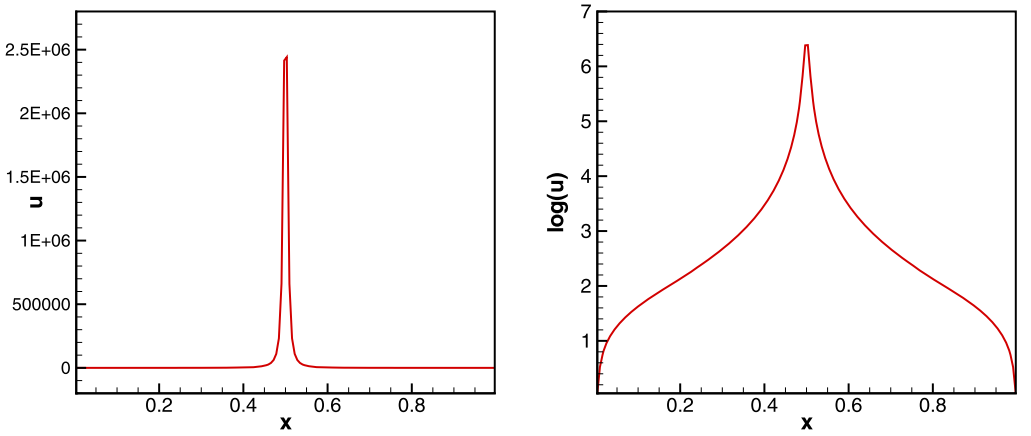

Fig. 2. Example 3.2: Numerical approximations u (left) vs. $\log(u)$ (right) at the blow-up time with P^2 polynomials and $N = 160$.

table we can observe that no matter which k we choose, the numerical blow-up time converges during the mesh refinement. For a fixed mesh size, the third scheme yields better numerical blow-up time. Especially, for $N > 20$, the numerical blow-up obtained from the third-order scheme do not vary much. This clearly shows that for high-order schemes, we can use a relatively coarse mesh to obtain accurate numerical blow-up time. In this example, we consider the reference blow-up time to be $T = 8.24371 \times 10^{-2}$. Clearly, the blow-up time computed from the first-order scheme is still somehow different from the reference one when $N = 160$. We also plot the numerical approximation at the blow-up time in Fig. 2, we can see that the blow-up set contains only one point $x = 0.5$.

Example 3.3. We take $\alpha = 1.5$, $m = 2$ and consider the following problem

$$u_t = u_{xx}^{1.5} + u^2, \quad x \in [-15, 15],$$
$$u(-15, t) = u(15, t) = 1, \quad t \in [0, T),$$
$$u(x, 0) = 15^2 - x^2 + 1, \quad x \in [-15, 15].$$

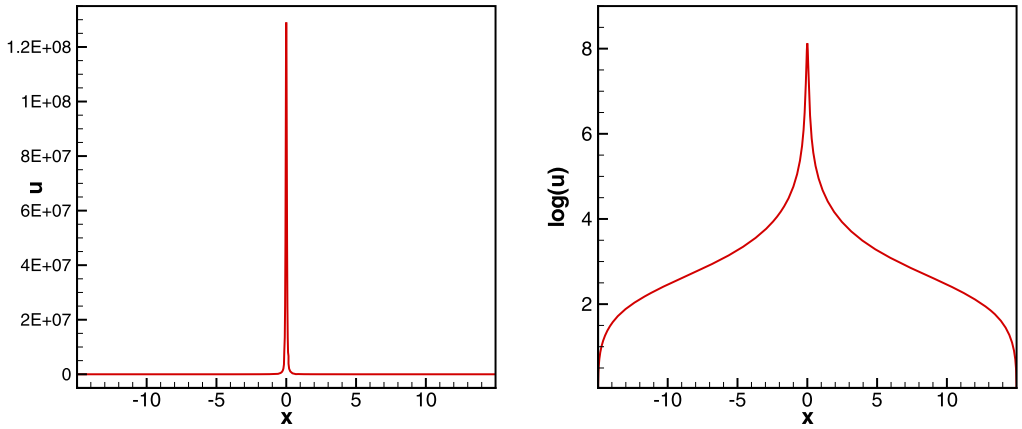

Fig. 3. Example 3.3: Numerical approximation u (left) vs. $\log(u)$ (right) at the blow-up time with P^2 polynomials and $N = 1280$.

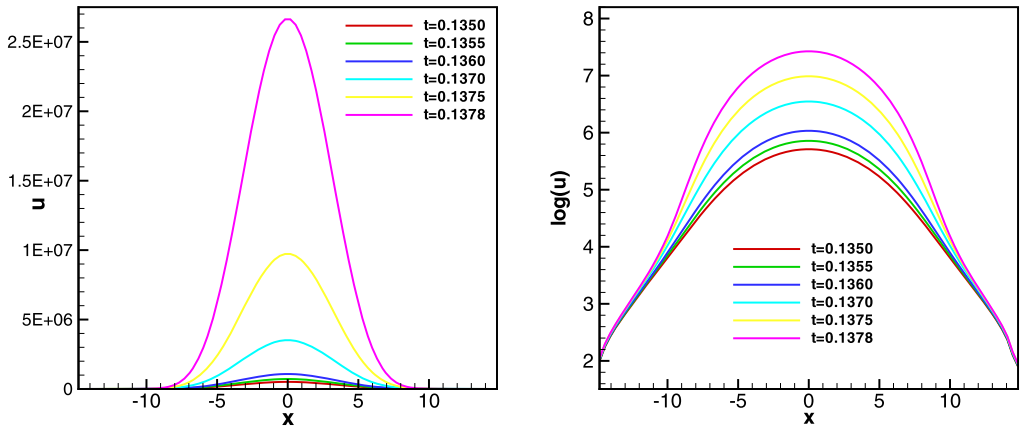

Fig. 4. Example 3.4: Left: The time evolution of the numerical approximation u (left) vs. $\log(u)$ (right) with P^2 polynomials and $N = 320$.

This example is used to check the necessity of the positivity-preserving technique. We first solve Example 3.3 based on the positivity-preserving LDG method. We use the third-order numerical scheme and take $N = 1280$. The numerical blow-up time is $T = 4.43243 \times 10^{-3}$. We also plot the numerical approximation at $T = 4.43243 \times 10^{-3}$ in Fig. 3, which shows the blow-up set is $x = 0$. However, without the positivity-preserving technique, the numerical approximation blows up at $T = 4.39810 \times 10^{-3}$, which is quite different the one given by the positivity-preserving LDG method.

In each the previous examples, the blow-up set contains one point only. However, our scheme can also be used to capture the phenomenon of regional blow-up.

Example 3.4. We take $\alpha = m = 1.5$, and study the following problem

$$u_t = u_{xx}^{1.5} + u^{1.5}, \quad x \in [-15, 15],$$
$$u(-15, t) = u(15, t) = 1, \quad t \in [0, T),$$
$$u(x, 0) = 15^2 - x^2 + 1, \quad x \in [-15, 15].$$

Different from the previous two, the two parameters α and m are the same in this example. In this case, the blow-up set is an interval and such a phenomenon is called regional blow-up. We would like to use the positivity-preserving LDG method to capture the blow-up set. Therefore, we plot the numerical approximation at the blow-up time in log scale. We also use the third-order numerical scheme and take $N = 320$. The blow-up time is $T = 0.1382$. From Fig. 4, we can find that the blow-up set is roughly at $(-9.42, 9.42) \approx (-3\pi, 3\pi)$.

3.2. Two-dimensional case

In this subsection, numerical experiments in two space dimensions are designed to show the performance of the positivity-preserving LDG scheme in solving (20). We take different sets of parameters α, β and m. First, we give one example to test the accuracy and then show some examples with blow-up solutions.

Table 4
Example 3.5, accuracy test at time $T = 0.1$.

	$N_x \times N_y$	L^2 error	order	L^∞ error	order
P^0	4×4	2.31E−01	–	5.31E−01	–
	8×8	1.44E−01	0.68	3.34E−01	0.67
	16×16	7.34E−02	0.97	1.71E−01	0.97
	32×32	3.69E−02	0.99	8.57E−02	0.99
P^1	4×4	5.23E−02	–	1.94E−01	–
	8×8	1.09E−02	2.26	4.10E−02	2.24
	16×16	2.67E−03	2.04	1.13E−02	1.85
	32×32	6.66E−04	2.00	2.91E−03	1.96
P^2	4×4	7.75E−03	–	4.64E−02	–
	8×8	8.00E−04	3.28	4.05E−03	3.52
	16×16	9.56E−05	3.06	5.13E−04	2.98
	32×32	1.17E−05	3.03	6.44E−05	2.99

Table 5
Example 3.6, the blow-up time T with different meshes and different degrees of polynomials.

$N_x \times N_y$	T		
	p^0	p^1	p^2
8×8	4.81935E−02	4.72074E−02	4.71102E−02
16×16	4.70453E−02	4.68389E−02	4.68125E−02
32×32	4.68009E−02	4.67537E−02	4.67468E−02
64×64	4.67440E−02	4.67328E−02	4.67311E−02
128×128	4.67303E−02	4.67277E−02	4.67272E−02

Example 3.5. We take $\alpha = \beta = m = 1$ and consider the following problem

$$u_t = u_{xx} + u_{yy} + (2\pi^2 - 1)u, \quad (x, y) \in [0, 1] \times [0, 1],$$

with the homogeneous Dirichlet boundary condition.

The exact solution is

$$u(x, y, t) = e^{-t} \sin(\pi x) \sin(\pi y).$$

We solve this example with polynomials of degree $k = 0, 1, 2$, and compute the error between the exact solution and the numerical solution in L^2- and L^∞-norms. From Table 4, we can observe optimal rate of convergence. This example illustrates that the flux limiter and the penalty term do not kill the accuracy.

Example 3.6. We take $\alpha = \beta = 1$, $m = 2$ and consider the following problem

$$u_t = u_{xx} + u_{yy} + u^2, \quad (x, y) \in [0, 1] \times [0, 1],$$

with the initial condition $u_0(x, y) = 40 \sin(\pi x) \sin(\pi y)$ and homogeneous Dirichlet boundary condition.

Table 5 shows the numerical blow-up time with different mesh sizes and different polynomial degrees. We can see that, for each fixed order of accuracy, the numerical blow-up time decreases and converges during mesh refinement. For each fixed mesh size, higher-order approximation yields better numerical blow-up time. Therefore, we can use a relatively coarse mesh to obtain accurate numerical blow-up time. Moreover, we also plot the numerical approximation at $T = 4.67272 \times 10^{-2}$. In Fig. 5, we can find the blow-up set contains only one point $(x, y) = (\frac{1}{2}, \frac{1}{2})$.

Example 3.7. We take $\alpha = 1$, $\beta = 1.5$, $m = 2$ and consider the following problem

$$u_t = u_{xx} + u_{yy}^{1.5} + u^2, \quad (x, y) \in [0, 1] \times [0, 1],$$

with the initial condition $u_0(x, y) = 200 \sin(\pi x) \sin(\pi y)$ and homogeneous Dirichlet boundary condition.

We use this example to test the positivity-preserving technique. We first solve the problem by using the positivity-preserving LDG method proposed in this paper. The numerical blow-up time is approximately $T = 1.82378 \times 10^{-2}$ with the P^2 polynomial and a uniform mesh with 64×64 rectangles. We also plot the numerical solution at $T = 1.82378 \times 10^{-2}$, and the result is given in Fig. 6. From the figure, we can observe that the blow-up set contains one point $(x, y) = (\frac{1}{2}, \frac{1}{2})$. It will blow up at $T = 1.22168 \times 10^{-6}$ which actually blow up after one time step without positivity-preserving limiter with

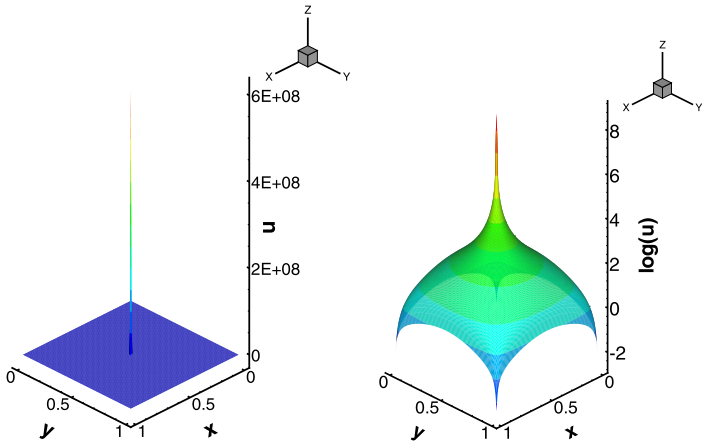

Fig. 5. Example 3.6: Numerical approximations u (left) vs. $\log(u)$ (right) at the blow-up time with P^2 polynomials and $N_x = N_y = 128$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

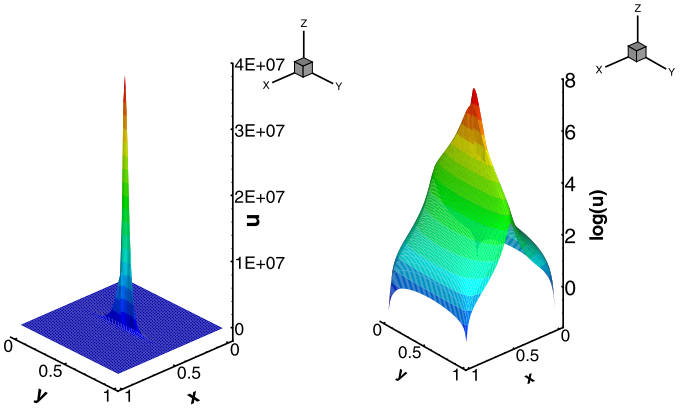

Fig. 6. Example 3.7: Numerical approximations u (left) vs. $\log(u)$ (right) at the blow-up time with P^2 polynomials and $N_x = N_y = 64$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

the same polynomial and mesh size. Different from Example 3.6, in this example, $\alpha \neq \beta$. Therefore, the decay rate in x- and y-directions are quite different. This can be observed from Fig. 6. If we observe from the x-axis, the singularity is much sharper.

4. Conclusion

In this paper, we developed the positivity-preserving high-order LDG method to solve parabolic equations with blow-up solutions subject to Dirichlet boundary conditions. The penalty terms are added to maintain the optimal convergence rates and special limiters are applied to obtain positive numerical approximations. Numerical experiments demonstrated that, without the positivity-preserving technique, the numerical blow-up time might be different from the correct one. Moreover, high-order numerical schemes yield better numerical blow-up time, and the schemes can also capture the blow-up set precisely.

References

[1] L.M. Abia, J.C. López-Marcos, J. Martinez, On the blow-up time convergence of semidiscretizations of reaction–diffusion equations, Appl. Numer. Math. 26 (1998) 399–414.
[2] G. Acosta, R.G. Durán, J.D. Rossi, An adaptive time step procedure for a parabolic problem with blow-up, Computing 68 (2002) 343–373.
[3] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.
[4] J. Bebernes, D. Eberly, Mathematical Problems from Combustion Theory, Springers-Verlag, 1989.
[5] T.K. Boni, On blow-up and asymptotic behavior of solutions to a nonlinear parabolic equation of second order, Asymptot. Anal. 21 (1999) 187–208.
[6] H. Brezis, et al., Blow-up of $u_t = u_{xx} + g(u)$ revisited, Adv. Differ. Equ. 1 (1996) 73–90.
[7] C. Bandle, H. Brunner, Numerical analysis of semilinear parabolic problems with blow-up solutions, Rev. R. Acad. Cienc. Exactas Fís. Nat. 88 (1994) 203–222.
[8] C. Bandle, H. Brunner, Blowup in diffusion equations: a survey, J. Comput. Appl. Math. 97 (1998) 3–22.

[9] C. Brändle, P. Groisman, J.D. Rossi, Fully discrete adaptive methods for a blow-up problem, Math. Models Methods Appl. Sci. 14 (2004) 1425–1450.
[10] C. Brändle, F. Quirós, J.D. Rossi, An adaptive numerical method to handle blow-up in a parabolic system, Numer. Math. 102 (2005) 39–59.
[11] C.J. Budd, W. Huang, R.D. Russell, Moving mesh methods for problems with blow-up, SIAM J. Sci. Comput. 17 (1996) 305–327.
[12] Y. Cheng, C.-W. Shu, A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives, Math. Comput. 77 (2008) 699–730.
[13] C.-H. Cho, On the finite difference approximation for blow-up solutions of the porous medium equation with a source, Appl. Numer. Math. 65 (2013) 1–26.
[14] B. Cockburn, S. Hou, C.-W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.
[15] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90–113.
[16] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.
[17] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[18] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[19] B. Cockburn, B. Dong, An analysis of the minimal dissipation local discontinuous Galerkin method for convection–diffusion problems, J. Sci. Comput. 32 (2007) 233–262.
[20] R. Ferreira, P. Groisman, J.D. Rossi, Numerical blow-up for a nonlinear problem with a nonlinear boundary condition, Math. Models Methods Appl. Sci. 12 (2002) 461–483.
[21] V.A. Galaktionov, J.L. Vazquez, The problem of blow-up in nonlinear parabolic equation, Discrete Contin. Dyn. Syst. 8 (2002) 399–433.
[22] P. Groisman, Totally discrete explicit and semi-implicit Euler methods for a blow-up problem in several space dimensions, Computing 76 (2006) 325–352.
[23] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (2001) 89–112.
[24] C. Johnson, J. Pitkäranta, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation, Math. Comput. 46 (1986) 1–26.
[25] F.S. Li, Global existence and uniqueness of weak solution to nonlinear viscoelastic full Marguerre–von Kármán shallow equations, Acta Math. Sin. Engl. Ser. 25 (2009) 2133–2156.
[26] I. Mai, K. Mochizuki, On blow-up of solutions for quasilinear degenerate parabolic equations, Publ. Res. Inst. Math. Sci. 27 (1991) 695–709.
[27] D. Nabongo, T.K. Boni, Blow-up for semidiscretization of a localized semilinear heat equation, J. Appl. Anal. 2 (2009) 173–204.
[28] F.K. N'Gohisse, T.K. Boni, Numerical blow-up for a nonlinear heat equation, Acta Math. Sin. Engl. Ser. 27 (2011) 845–862.
[29] P. Quittner, P. Souplet, Superlinear Parabolic Problems: Blow-up, Global Existence and Steady States, Birkhäuser Adv. Texts, Birkhäuser, Basel, 2007.
[30] W.H. Reed, T.R. Hill, Triangular mesh methods for the Neutron transport equation, Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.
[31] M.-N. Le Roux, Semidiscretization in time of nonlinear parabolic equations with blow-up of the solution, SIAM J. Numer. Anal. 31 (1994) 170–195.
[32] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.
[33] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[34] R. Suzuki, On blow-up sets and asymptotic behavior of interface of one dimensional quasilinear degenerate parabolic equation, Publ. Res. Inst. Math. Sci. 27 (1991) 375–398.
[35] T.K. Ushijima, On the approximation of blow-up time for solutions of nonlinear parabolic equations, Publ. Res. Inst. Math. Sci. 36 (2000) 613–640.
[36] T. Xiong, J.-M. Qiu, Z. Xu, A maximum principle preserving limiter for discontinuous Galerkin method with applications to convection–diffusion equations, submitted for publication.
[37] Z. Xu, Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem, Math. Comput. 83 (2014) 2213–2238.
[38] Y. Yang, C.-W. Shu, Discontinuous Galerkin methods for hyperbolic equations involving δ-singularities: negative-order norm error estimates and applications, Numer. Math. 124 (2013) 753–781.
[39] Y. Yang, D. Wei, C.-W. Shu, Discontinuous Galerkin method for Krause's consensus models and pressureless Euler equations, J. Comput. Phys. 252 (2013) 109–127.
[40] X. Zhao, Y. Yang, C. Seyler, A positivity-preserving semi-implicit discontinuous Galerkin scheme for solving extended magnetohydrodynamics equations, J. Comput. Phys. 278 (2014) 400–415.
[41] L. Zhang, Y.S. Ziang, Z. Zhou, Parabolic equation with VMO coefficients in generalized Morrey, Acta Math. Sin. Engl. Ser. 26 (2010) 117–130.
[42] Q. Zhang, Z.-L. Wu, Numerical simulation for porous medium equation by local discontinuous Galerkin finite element method, J. Sci. Comput. 38 (2009) 127–148.
[43] X. Zhang, C.-W. Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, J. Comput. Phys. 229 (2010) 3091–3120.
[44] X. Zhang, C.-W. Shu, On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes, J. Comput. Phys. 229 (2010) 8918–8934.
[45] Y. Zhang, X. Zhang, C.-W. Shu, Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection–diffusion equations on triangular meshes, J. Comput. Phys. 234 (2013) 295–316.