J Sci Comput (2009) 38: 127–148
DOI 10.1007/s10915-008-9223-7

Numerical Simulation for Porous Medium Equation by Local Discontinuous Galerkin Finite Element Method

Qiang Zhang · Zi-Long Wu

Received: 21 April 2008 / Revised: 23 June 2008 / Accepted: 27 June 2008 / Published online: 19 July 2008

Abstract In this paper we will consider the simulation of the local discontinuous Galerkin (LDG) finite element method for the porous medium equation (PME), where we present an additional nonnegativity preserving limiter to satisfy the physical nature of the PME. We also prove for the discontinuous $\mathbb{P}_0$ finite element that the average in each cell of the LDG solution for the PME maintains nonnegativity if the initial solution is nonnegative within some restriction for the flux's parameter. Finally, numerical results are given to show the advantage of the LDG method for the simulation of the PME, in its capability to capture accurately sharp interfaces without oscillation.

Keywords Local discontinuous Galerkin · Finite element · Porous medium equation · Nonnegativity preserving limiter

1 Introduction

In this paper we will consider the numerical simulation by the local discontinuous Galerkin (LDG) finite element method for the porous medium equation (PME), namely,

$$u_t = (u^m)_{xx}, \quad x \in \mathbb{R},\ t > 0, \tag{1}$$

in which m is a constant greater than one. This equation often occurs in nonlinear problems of heat and mass transfer, combustion theory, and flow in porous media, where u is either a concentration or a temperature required to be nonnegative. We assume in this paper that the

The research of Q. Zhang is supported by CNNSF grant 10301016.

Q. Zhang (✉)
Department of Mathematics, Nanjing University, Nanjing, Jiangsu Province, 210093, People's Republic of China
e-mail: qzh@nju.edu.cn

Z.-L. Wu
School of Mathematics and Science, Shijiazhuang University of Economics, Shijiazhuang, Hebei Province, 050031, People's Republic of China

initial-value for the PME (1), $u_0(x)$, is a bounded nonnegative continuous function. Then (1) falls into the standard nonlinear diffusion equation

$$u_t = (a(u)u_x)_x, \quad x \in \mathbb{R},\ t > 0; \tag{2}$$

with $a(u) = mu^{m-1} \geq 0$; however this equation is not strictly parabolic, since it degenerates at points where $u = 0$. Thus this is also called a degenerate parabolic equation.

In this case, the classical smooth solution may not always exist in general, even if the initial solution is smooth. It is necessary to consider the weak energy solution, whose behavior causes many difficulties for a good numerical simulation. For example, the weak solution may lose its classical derivative at some (interface) points, and the sharp interface of support may propagate with finite speed if the initial data have compact support. Enamored of these interesting facts, there have been many works on the simulation for the nonsmooth solution of the PME, for example, the finite different method by Graveleau and Jamet [12], the interface tracking algorithm by DiBenedetto and Hoff [11], and the relaxation scheme referred to in [15].

In this paper we pay particular attention to the finite element method, specifically, to the LDG method for the PME. The considered LDG method is a particular version of the discontinuous Galerkin (DG) method, which uses a completely discontinuous piecewise polynomial space for the numerical solution and the test functions. The first DG method was introduced in 1973 by Reed and Hill [19], in the framework of neutron transport (steady state linear hyperbolic equations). Then it was developed into the Runge-Kutta discontinuous Galerkin (RKDG) scheme by Cockburn et al. [3, 5, 7–9] for nonlinear hyperbolic systems. Later, the LDG method was introduced by Cockburn and Shu in [6] as an extension of the RKDG method to general convection-diffusion problems, or of the numerical scheme for the compressible Navier-Stokes equations proposed by Bassi and Rebay in [2]. For a fairly complete set of references on RKDG and LDG methods as well as their implementation and applications, see the review paper by Cockburn and Shu [10]; see also the review of the development of DG methods by Cockburn, Karniadakis and Shu [4].

The DG method possesses several properties to make it very attractive for practical computations, such as parallelization, adaptivity, and simple treatment of boundary conditions. The most important properties of this method is its strong stability and high-order accuracy; as a result, it is very good at capturing discontinuous jumps and sharp transient layers. Cockburn and Shu [6] proved that this scheme has a good L^2-stability for the standard nonlinear diffusion equation, including the PME. Furthermore, Xu and Shu [23] gave a quasi-optimal error estimate for the semi-discretized LDG method for such equations if the solution is smooth enough.

There are two main components in this paper. The first is the simulation by the LDG method for nonsmooth solutions of the PME. Based on the general implementation of the LDG method, we design a nonnegativity-preserving limiter to strengthen the physical relevancy of the numerical solutions. The given numerical results verify the above advantage of the LDG method that it has the ability to capture sharp interfaces accurately without or with very little numerical oscillation. As a comparison, the standard finite element method is also considered to show the numerical difficulties; see Sect. 2. The second component is an analysis for the nonnegativity preservation principle of the considered LDG method, i.e., the average in each cell of the LDG solution for the PME remains nonnegative, if the time step is small enough. This nice conclusion has been used for designing the nonnegativity-preserving limiter, although it is proved only for the discontinuous $\mathbb{P}_0$ finite element in this

paper. At the same time we will point out in this paper that we need to take the flux's parameter within some restriction for the nonnegativity preserving principle to hold for the PME, even though any choice of this parameter ensures the L^2-stability of the LDG method.

The rest content of this paper is organized as follows: In Sect. 2, we will show the numerical difficulties in the simulation for the PME, where the standard finite element method is used to resolve the Barenblatt solution. In Sect. 3, we will describe the detailed implementation of the general LDG method, together with the nonnegative-preserving limiter for the PME's simulation. A short analysis on the nonnegativity-preserving principle of the LDG method is given in Sect. 4, for the discontinuous $\mathbb{P}_0$ finite element. A few interesting numerical results are presented in Sect. 5, and the concluding remarks are given in Sect. 6.

2 Numerical Difficulties in the Simulation

We would like in this section to emphasize that adopting suitable numerical method is important to get good simulation results for the PME. We begin our study with the Barenblatt solution of the PME (1).

For any given m greater than one, the famous Barenblatt solution is defined by

$$B_m(x,t) = t^{-k}\left[\left(1 - \frac{k(m-1)}{2m}\frac{|x|^2}{t^{2k}}\right)_+\right]^{1/(m-1)}, \tag{3}$$

where $u_+ = \max(u, 0)$ and $k = (m+1)^{-1}$. This solution, for any time $t > 0$, has a compact support $[-\alpha_m(t), \alpha_m(t)]$ with the interface $|x| = \alpha_m(t)$ moving outward in a finite speed, where

$$\alpha_m(t) = \sqrt{\frac{2m}{k(m-1)}} \cdot t^k. \tag{4}$$

This property is the so-called finite speed of propagation of perturbations for degenerate parabolic equation. Barenblatt solution is very representative for PME to be a (weak) energy solution but not a classical solution, since there exists no derivative at the interface points $|x| = \alpha_m(t)$. To understand this clearly, we have plotted in Fig. 1 the pictures for

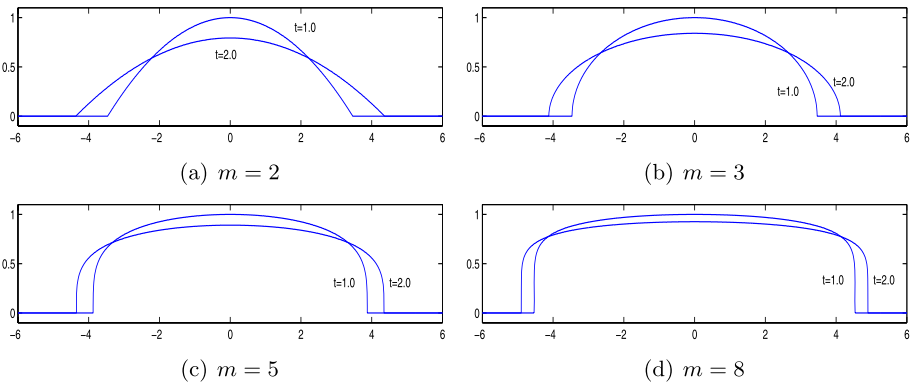
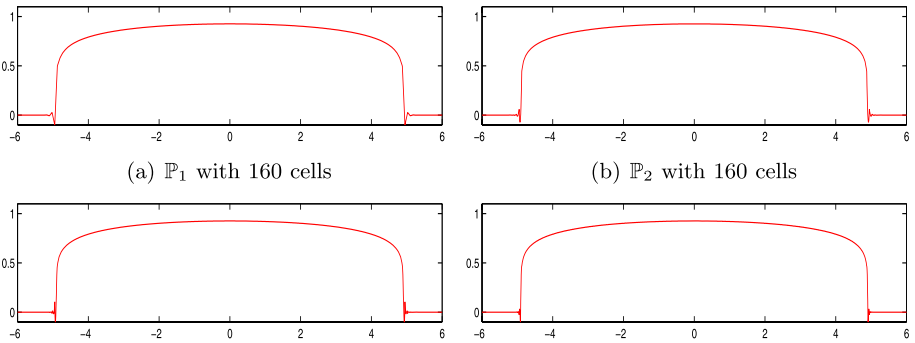


Fig. 1 Barenblatt solutions for the parameters $m = 2, 3, 5, 8$



(a) $\mathbb{P}_1$ with 160 cells (b) $\mathbb{P}_2$ with 160 cells

(c) $\mathbb{P}_3$ with 160 cells (d) $\mathbb{P}_2$ with 320 cells

Fig. 2 Numerical results for the Barenblatt solution by the PCSFE method: $t = 2$ and $m = 8$

the Barenblatt solution at time $t = 1$ and $t = 2$, respectively. The considered parameters are $m = 2, 3, 5$ and 8 from top-left to bottom-right. When the parameter m increases, the Barenblatt solution tends to vary more slowly inside its support, and it tends to be steeper near the interface of the support.

This kind of solutions will lead to difficulties in numerical simulation, for example, if we use the standard finite element (SFE) method. In this experiment, we use the conforming $\mathbb{P}_k$ finite element space on a uniform spatial mesh, that is to say, the piecewise polynomial of degree k is continuous at the element interface. Time is discretized by a predictor-correction algorithm, and the corresponding scheme is referred to as the PCSFE method. We begin the computation from $t = 1$ in order to obviate the singularity of the Barenblatt solution near $t = 0$. The boundary condition is $u(\pm 6, t) = 0$ for $t \geq 1$.

We present, in Fig. 2, some pictures for the PCSFE solutions of the Barenblatt solution. The plotting time is $t = 2$ and the parameter in the PME is $m = 8$; see Fig. 1(d) for the exact solution. In Figs. 2(a) to (c), we use the conforming finite element space from $\mathbb{P}_1$ to $\mathbb{P}_3$, respectively, all defined on the same mesh with 160 cells. The last Fig. 2(d) is plotted in comparison with Fig. 2(b), where only the mesh is refined from 160 cells to 320 cells, and the finite element space is still conforming $\mathbb{P}_2$.

From these pictures, we can see that the PCSFE method is not suitable for this simulation, especially for the non-smooth solution of the PME, since obvious oscillations appear near the interface, introducing negative solution values which are meaningless for physical interpretation. Moreover, it seems that such oscillations cannot be removed by raising the degree of finite element space and/or by refining spatial meshes. This numerical experiment motivates us to consider the LDG method, in order to capture accurately the sharp transient layer without oscillations.

3 Implementation of the LDG Method

In this section we describe in detail the implementation of the LDG method. To be more general, we consider the following convection-diffusion equation

$$u_t + \big[f(u) - a(u)u_x\big]_x = s(u), \quad x \in I = [x_a, x_b],\ t > 0, \tag{5a}$$

together with the initial condition

$$u(x, 0) = u_0(x), \quad x \in I, \tag{5b}$$

and some suitable boundary conditions, where $a(u) \geq 0$ is the diffusion viscosity, $f(u)$ is the convection flux, and $s(u)$ is the source term. For the PME (1), $a(u) = mu^{m-1}$, $f(u) = 0$ and $s(u) = 0$.

The discretization of the LDG method is obtained first by reformulating the equation (5a), as a nonlinear system of first order equations. By introducing a new variable $q = \sqrt{a(u)}\, u_x$, the resulting system is of the form

$$u_t + [f(u) - \sqrt{a(u)}q]_x = s(u), \quad x \in I,\ t > 0, \tag{6a}$$

$$q - g_x(u) = 0, \quad x \in I,\ t > 0, \tag{6b}$$

together with the same initial and boundary conditions as before, where $g(u) = \int^u \sqrt{a(u)}\, \mathrm{d}u$ is the diffusion flux for the auxiliary variable q.

The LDG method for (5) is then obtained by the DG discretization for the system (6), including the discretization of the spatial variables, the treatment to initial and boundary conditions, the time-marching and the slope limiter. See [6, 10] for more details.

Let $\mathbf{w} = (u, q)^t$, and define the flux function

$$\mathbf{h}(\mathbf{w}) = (\mathbf{h}_u(\mathbf{w}), \mathbf{h}_q(\mathbf{w}))^t = (f(u) - \sqrt{a(u)}q, -g(u))^t$$

for the simplification of notations.

3.1 The LDG Spatial Discretization

We first divide the domain into N cells with boundary points $x_a = x_{1/2} < x_{3/2} < \cdots < x_{N+1/2} = x_b$, and denote the cells by $I_j = (x_{j-1/2}, x_{j+1/2})$ and the cell size by $\Delta x_j = x_{j+1/2} - x_{j-1/2}$ for $j = 1, 2, \ldots, N$.

In the LDG method, the approximation solutions u_h and q_h, for any time $t \in (0, T]$, as well as the test function, belong to the discontinuous finite element space

$$V_h = \{v \in L^2[x_a, x_b] : v|_{I_j} \in \mathbb{P}_k(I_j),\ \forall I_j \in \mathcal{T}\}, \tag{7}$$

where $\mathbb{P}_k(I_j)$ denotes the space of polynomials in I_j of degree at most k. As usual, we denote by $w^{\pm}_{j+1/2} = w(x^{\pm}_{j+1/2}, t)$ the left and right limits of the discontinuous solution w at the boundary point $x_{j+1/2}$, and denote the corresponding average and jump by $\{\{w\}\}_{j+1/2} = \frac{1}{2}(w^+_{j+1/2} + w^-_{j+1/2})$ and $[[w]]_{j+1/2} = w^+_{j+1/2} - w^-_{j+1/2}$, respectively.

Following [6], we multiply smooth test function v_h and r_h with (6a) and (6b), respectively, and integrate by part in each cell. Then the LDG scheme is obtained after introducing the numerical fluxes at boundary points of each cell. The LDG solution $\mathbf{w}_h = (u_h, q_h)^t \in V_h \times V_h$, for any time $t \in (0, T]$, satisfies

$$\int_{I_j} u_{h,t} v_h \mathrm{d}x = \int_{I_j} [s(u_h)v_h + \mathbf{h}_u(\mathbf{w}_h)v_{h,x}]\mathrm{d}x - \hat{h}_{u,j+\frac{1}{2}}(\mathbf{w}_h)v^-_{h,j+\frac{1}{2}} + \hat{h}_{u,j-\frac{1}{2}}(\mathbf{w}_h)v^+_{h,j-\frac{1}{2}}, \tag{8a}$$

$$\int_{I_j} q_h r_h \mathrm{d}x = \int_{I_j} \mathbf{h}_q(u_h) r_{h,x} \mathrm{d}x - \hat{h}_{q,j+\frac{1}{2}}(\mathbf{w}_h)r^-_{h,j+\frac{1}{2}} + \hat{h}_{q,j-\frac{1}{2}}(\mathbf{w}_h)r^+_{h,j-\frac{1}{2}}, \tag{8b}$$

for any test function v_h and r_h in V_h. The initial condition is taken as $u_h(x, 0) = \mathbb{P}_h u_0(x)$, which is the local L^2-projection of initial condition $u_0(x)$, as the unique function in the finite element space V_h such that

$$\int_I u_0(x)v_h(x)\mathrm{d}x = \int_I \mathbb{P}_h u_0(x) v_h(x)\mathrm{d}x, \quad \forall v_h(x) \in V_h. \tag{8c}$$

The numerical flux $\hat{\mathbf{h}}(\mathbf{w}_h) = (\hat{h}_u(\mathbf{w}_h), \hat{h}_q(\mathbf{w}_h))^t$ plays an important role in ensuring the good performance of the LDG method. In general it depends on the two values at the interface point, i.e.,

$$\hat{\mathbf{h}}(\mathbf{w}_h)_{j+\frac{1}{2}} \equiv \hat{\mathbf{h}}(\mathbf{w}^-_{h,j+\frac{1}{2}}, \mathbf{w}^+_{h,j+\frac{1}{2}}).$$

Below we will omit the subscripts h and $j+\frac{1}{2}$ when there is no confusion. Following [6], we define the numerical flux as the sum of the convection flux and the diffusion flux. It reads

$$\hat{\mathbf{h}}(\mathbf{w}^-, \mathbf{w}^+) = \left(\hat{f}(u^-, u^+) - \frac{[[g(u)]]}{[[u]]}\{\{q\}\} - \gamma[[q]], -\{\{g(u)\}\} + \gamma[[u]] \right)^t, \tag{9}$$

where $(\hat{f}(u^-, u^+), 0)^t$ is termed as the convection numerical flux, and the remaining part in (9) is termed as the diffusion numerical flux. Here $\hat{f}(u^-, u^+)$ is any locally Lipschitz numerical flux consistent with the nonlinearity convection flux $f(u)$, satisfying the so-called E-flux property [17]. We refer to [10] for more details. The flux's parameter γ is locally Lipschitz in its arguments $u^{\pm}$, and tends to zero when the viscosity $a(u)$ disappears. If $[[u]] = 0$ we define $[[g(u)]]/[[u]] = \sqrt{a(u)}$.

The LDG method is very flexible to deal with boundary conditions. We just need to put correctly the boundary condition into the numerical boundary flux, without changing the framework of the LDG scheme. For example, for the Dirichlet boundary condition such as

$$u(x_a, t) = \alpha(t), \quad u(x_b, t) = \beta(t), \quad t > 0, \tag{10}$$

we define the numerical boundary flux at the domain boundary as follows:

$$\mathbf{h}_{x_a} = (\hat{f}(\alpha, u^+) - \sqrt{a(\alpha)}q^+, -g(\alpha)), \qquad \mathbf{h}_{x_b} = (\hat{f}(u^-, \beta) - \sqrt{a(\beta)}q^-, -g(\beta)). \tag{11}$$

In this paper $\mathbf{h}_{x_a} = \mathbf{h}_{x_b} = 0$, since only homogeneous Dirichlet boundary condition is considered for the PMEs.

Remark 3.1 The local property of the LDG methods is a result of the fact that the diffusion numerical flux depends only on the primal variable u_h. This allows us to solve locally the auxiliary variable q_h by the primal variable u_h, from the second equation (8b).

As a locally conservative scheme, the LDG scheme possesses many good properties. It is strongly stable and are high order accurate. The opposite signs before $\gamma[u]$ and $\gamma[q]$ in the numerical flux ensure the L^2-stability of the LDG method, for any choice of the flux's parameter γ. When the piecewise polynomials of degree k is used, the LDG method has at least k-order accuracy in the L^2-norm, and in many cases $(k+1)$-order accuracy can be achieved. For more details, see Cockburn and Shu [6, 10], and Xu and Shu [23].

3.2 Runge-Kutta Time-Marching

The combination of an explicit Runge-Kutta time stepping scheme together with the DG method for space discretization has been widely studied. In this paper we use the total variation diminishing explicit Runge-Kutta (TVD-ERK) method, first given by Shu and Osher [22]. It is a suitable convex combination of Euler-forward time-marching, and has been further developed in [13, 14] and termed as the explicit strong stability preserving (SSP) time discretization.

After the spatial discretization with a series of basis functions, the LDG scheme is equivalent to the first-order ODE system

$$\frac{\mathrm{d}u_h}{\mathrm{d}t} = \mathbb{M}^{-1}\mathrm{RHS}(u_h) = \mathcal{L}_h(u_h), \quad t \in (0, T), \tag{12}$$

where $\mathrm{RHS}(u_h)$ is resulted from the spatial discretization to the right-hand side of (8), and $\mathbb{M}$ is the mass matrix for the given basis functions. The inverse of $\mathbb{M}$ is easy to obtain, since $\mathbb{M}$ is a block diagonal matrix, even diagonal when a locally orthogonal basis is used.

In this paper we use the following third-order TVD-ERK time-marching to solve (12). The detail of the implementation of this algorithm is given by

$$u^{(1)} = u_h^n + \Delta t^n \mathcal{L}_h(u_h^n), \tag{13a}$$

$$u^{(2)} = \frac{3}{4}u_h^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t^n \mathcal{L}_h(u^{(1)}), \tag{13b}$$

$$u_h^{n+1} = \frac{1}{3}u_h^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t^n \mathcal{L}_h(u^{(2)}). \tag{13c}$$

For more details of the TVD-ERK schemes, we refer to [10, 13, 14].

With such a strategy, the time-step must be selected to satisfy a Courant-Friedrichs-Lewy (CFL) stability condition to ensure numerical stability, which in general depends on the size of the smallest element of the mesh, on the maximum convection speed and diffusion viscosity, and on the polynomial order of the approximation. In our case, the allowable time step Δt^n satisfies

$$\frac{\max|f'(u_h^n)|\Delta t^n}{\rho} + \frac{2\max a(u_h^n)\Delta t^n}{\rho^2} \le \mathrm{CFL}_{max}, \tag{14}$$

where $\rho = \min_{1\le j\le N}\Delta x_j$, and the maximum is taken over the interval covered by the numerical solution u_h^n. A table for the values CFL_{max} has been given by Cockburn and Shu[10], for piecewise polynomials of degree k, together with r-order TVD-ERK time-marching.

3.3 Slope Limiter

In the LDG method, the slope limiter is often used to strengthen numerical stability. The main ingredient of the limiter is that it only modifies the slope if necessary while maintaining the cell average, in order to control oscillation without sacrificing the conservation of the numerical solution. It is feasible to implement a slope limiter for discontinuous finite elements; however, it is not feasible to do so for the conforming finite elements.

In our simulation for the PME, we will use two successive limiters for the numerical solution u_h at each computational step. The first is the general slope limiter $\Lambda\Pi_h$, given in Cockburn and Shu [7, 10]; the second is our newly designed limiter for the nonnegativity constraint of the numerical solution, denoted by $P\Pi_h$ in this paper.

3.3.1 The First Slope Limiter $\Lambda\Pi_h$

We begin the description of the limiter $\Lambda\Pi_h$ from the piecewise linear solutions u_h such as

$$u_h|_{I_j} = \bar{u}_j + (x - x_j)u_{x,j}, \quad j = 1, 2, \ldots, N,$$

where $\bar{u}_j$ and $u_{x,j}$ are the average and slope of solution in the cell I_j, respectively. Then we can use a slope limiter $\Lambda\Pi_h^1$, due to Osher [18], to modify the slope $u_{x,j}$. The numerical solution in this cell is limited as

$$\Lambda\Pi_h^1 u_h|_{I_j} = \bar{u}_j + (x - x_j)\bar{m}\left(u_{x,j}, \frac{\bar{u}_j - \bar{u}_{j-1}}{h_j/2}, \frac{\bar{u}_{j+1} - \bar{u}_j}{h_j/2}\right), \tag{15}$$

where $\bar{m}$ is the modified *minmod* function given by

$$\bar{m}(a_1, a_2, a_3) = \begin{cases} a_1, & |a_1| \le \mu h^2, \\ s\min(|a_1|, |a_2|, |a_3|), & |a_1| > \mu h^2, \text{ and } s = \operatorname{sgn} a_i,\ i = 1, 2, 3, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

and $\mu \ge 0$ is a suitable constant to ensure total variation boundedness (TVB) and high order accuracy, see [21] for more details. In this paper we take $\mu = 1$ in all simulations. Moreover, formula (15) can be rewritten as follows:

$$\Lambda\Pi_h^1 u^-_{h,j+1/2} = \bar{u}_j + \bar{m}(u^-_{j+1/2} - \bar{u}_j, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j), \tag{17a}$$

$$\Lambda\Pi_h^1 v^+_{h,j-1/2} = \bar{u}_j - \bar{m}(\bar{u}_j - u^+_{j-1/2}, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j), \tag{17b}$$

which rely only on the cell averages of the numerical solution in adjacent cells, together with two values of the numerical solution at the endpoints of the considered cell. Hence the formula (17) is very convenient to use for piecewise high order polynomials.

The slope limiter for piecewise polynomials of degree $k \ge 2$ is based on this general formula (17) and the local L^2-projection. It reads as follows:

1. Compute $\Lambda\Pi_h^1 u^-_{h,j+1/2}$ and $\Lambda\Pi_h^1 u^+_{h,j-1/2}$ by using formula (17);
2. If $\Lambda\Pi_h^1 u^-_{h,j+1/2} = u^-_{h,j+1/2}$ and $\Lambda\Pi_h^1 u^+_{h,j-1/2} = u^+_{h,j-1/2}$, set $\Lambda\Pi_h u_h|_j = u_h|_{I_j}$;
3. If not, take $\Lambda\Pi_h u_h|_j = \Lambda\Pi_h^1 u_h^1$, where u_h^1 is the local L^2-projection of u_h into piecewise linear polynomials.

Remark 3.2 The boundary condition reflects the slope limiter also. Following [10], we can use (17) to define slope limiter similarly as above, by introducing two (ghost) element I_0 and I_{N+1}, and the corresponding average

$$\bar{u}_0 = 2\alpha(t) - \bar{u}_1, \qquad \bar{u}_{N+1} = 2\beta(t) - \bar{u}_N \tag{18}$$

for the considered Dirichlet boundary condition (10).

3.3.2 *The Second Slope Limiter* $P\Pi_h$

It is reasonable to the physical nature of the PME to expect the numerical solution to stay nonnegative for all time. That is to say, if the solution is negative at some points in one cell, it should be modified to become nonnegative. In this paper we design a new limiter $P\Pi_h$ to achieve this purpose, which depends solely on the numerical solution in the considered cell.

We begin the description of the limiter $P\Pi_h$ for piecewise linear polynomials. Now it is easy to judge whether a negative value emerges in the cell I_j, by the sign of solution at two endpoints. If both values are nonnegative, it is obvious that $u_h \ge 0$ in this cell I_j and the numerical solution needs no modification. Otherwise, we have to repair the *bad* solution to ensure nonnegativity. The details are given as follows.

1. Check if one of the values at two endpoints is negative. If so, take

$$P\Pi_h^1 u_h^1 = \begin{cases} [1 - 2h_j^{-1}(x - x_j)]\bar{u}_j, & \text{if } u^-_{h,j+\frac{1}{2}} < 0, \\ [1 + 2h_j^{-1}(x - x_j)]\bar{u}_j, & \text{if } u^+_{h,j-\frac{1}{2}} < 0. \end{cases} \tag{19}$$

2. Otherwise, set $P\Pi_h u_h|_{I_j} = u_h|_{I_j}$.

In either case, the absolute value of the slope does not increase. Consequently, the total variation of the numerical solution will not increase after this second slope limiting.

We remark that no discussion has been given for the case that both values of the numerical solution at the endpoints are negative. This is because such situation will not arise for our scheme, since the cell average in each cell of the numerical solution stays negative, a property which will be proved in the next section, for the discontinuous $\mathbb{P}_0$ finite element. However, we presume that this property is true for the piecewise polynomials with any degree, which is verified in the numerical results given in this paper. As a consequence, this second limiter does not take any effect for piecewise linear polynomials when the TVD limiter (15) with $\mu = 0$ in (16) is used, since it is carried out after the limiter $\Lambda\Pi_h$ which has already enforced a local maximum principle and hence the nonnegativity of the numerical solution. However, negative numerical solution values may appear in those cells where the cell averages in the cell itself and/or in the adjacent cells are close to zero, and the high-order polynomials have not been modified by the first slope limiter $\Lambda\Pi_h$ or they have been modified by the TVB limiter with $\mu > 0$.

The nonnegativity preserving limiter for higher order polynomials is based on the treatment above for linear polynomials. However, now we encounter the difficulty on how to detect negative values in the cell. Thus we scan the sign of the numerical solution at some preselected points which are distributed inside the considered cell, for example, two endpoints and the N_p Gaussian points used in the Gaussian numerical integration. If no negativity is detected at these points, it is acceptable to judge that the solution is nonnegative in this cell, so the solution stays the same as before. Otherwise, we project the solution into a linear polynomial, and modify it by the limiter $P\Pi_h^1$. This is enough for our purpose since the only values of the numerical solution that we use in the algorithm are those at two endpoints for numerical flux, and the Gaussian points, which are used for a numerical integration of the cell integrals in the scheme.

The details of the procedure now read as follows.

1. Scan the sign of the numerical solution u_h at two endpoints of the cell I_j, and the N_p Gaussian points inside the cell I_j;
2. If negative value emerges at least once, set $P\Pi_h u_h = P\Pi_h^1 u_h^1$, where u_h^1 is the local L^2-projection of the solution u_h into linear polynomial in this cell, then apply the second limiter (19);
3. Otherwise, set $P\Pi_h u_h|_{I_j} = u_h|_{I_j}$.

Remark 3.3 The above limiter is necessary in order to get good simulations. The general slope limiter is used to control the oscillation of numerical solution, and the nonnegativity preserving limiter is used to recover the nature of solution of degenerate problems. The computation maybe do not work without the later limiter, because there exist many square root to get.

4 Nonnegativity Preserving Principle

In this section we will discuss whether the cell average in each cell of the LDG solution has a nonnegativity preserving principle. To do that, we will consider the LDG method with piecewise constant approximation, namely, discontinuous $\mathbb{P}_0$ finite elements. Since there are little difference between piecewise polynomials of degree $k \geq 1$, we reckon this conclusion is also true for the piecewise polynomials of any degree. See the Remark 4.2.

We denote by u_j and q_j, the average of numerical solution u_h and q_h in the cell I_j, respectively. Then the scheme (8) can be rewritten in the explicit form

$$h_j u_{j,t} = \hat{h}_{u,j-\frac{1}{2}} - \hat{h}_{u,j+\frac{1}{2}}, \qquad h_j q_j = \hat{h}_{q,j-\frac{1}{2}} - \hat{h}_{q,j+\frac{1}{2}}, \tag{20a}$$

with the numerical fluxes

$$\hat{h}_{u,j+\frac{1}{2}} = -\frac{1}{2}\omega_{j+\frac{1}{2}}(q_{j+1}+q_j) - \gamma_{j+\frac{1}{2}}(q_{j+1}-q_j), \tag{20b}$$

$$\hat{h}_{q,j+\frac{1}{2}} = -\frac{1}{2}(g_{j+1}+g_j) + \gamma_{j+\frac{1}{2}}(u_{j+1}-u_j), \tag{20c}$$

where $g_j = g(u_j)$, and $\omega_{j+\frac{1}{2}} = \frac{g_{j+1}-g_j}{u_{j+1}-u_j}$ if $u_{j+1} \neq u_j$; otherwise $\omega_{j+\frac{1}{2}} = \sqrt{a(u_j)}$ if $u_{j+1} = u_j$.

Since the TVD-ERK algorithm is a convex combination of Euler forward time-marching, we just need to consider one step of Euler forward at time t^n below. After a series of simple manipulations, we can write the above scheme in a compact form

$$\Box u_j = A_2 u_{j+2} + A_1 u_{j+1} + A_0 u_j + A_{-1} u_{j-1} + A_{-2} u_{j-2}, \tag{21a}$$

where $\Box u_j = (u_j^{n+1} - u_j^n)/\Delta t$. Here and below the super-script n is omitted, and all terms on the right-hand side of (21a) is defined at time t^n with the coefficients defined by

$$A_2 = h_j^{-1} h_{j+1}^{-1} \mathcal{P}_{j+\frac{1}{2}} \mathcal{N}_{j+\frac{3}{2}}, \tag{21b}$$

$$A_1 = h_j^{-2}(\mathcal{N}_{j+\frac{1}{2}} - \mathcal{P}_{j-\frac{1}{2}})\mathcal{N}_{j+\frac{1}{2}} + h_j^{-1} h_{j+1}^{-1}(\mathcal{P}_{j+\frac{1}{2}} - \mathcal{N}_{j+\frac{3}{2}})\mathcal{P}_{j+\frac{1}{2}}, \tag{21c}$$

$$A_0 = -h_j^{-1} h_{j-1}^{-1} \mathcal{N}_{j-\frac{1}{2}}^2 - h_j^{-2}(\mathcal{N}_{j+\frac{1}{2}} - \mathcal{P}_{j-\frac{1}{2}})^2 - h_j^{-1} h_{j+1}^{-1} \mathcal{P}_{j+\frac{1}{2}}^2, \tag{21d}$$

$$A_{-1} = h_j^{-2}(\mathcal{P}_{j-\frac{1}{2}} - \mathcal{N}_{j+\frac{1}{2}})\mathcal{P}_{j-\frac{1}{2}} + h_j^{-1} h_{j-1}^{-1}(\mathcal{N}_{j-\frac{1}{2}} - \mathcal{P}_{j-\frac{3}{2}})\mathcal{N}_{j-\frac{1}{2}}, \tag{21e}$$

$$A_{-2} = h_j^{-1} h_{j-1}^{-1} \mathcal{N}_{j-\frac{1}{2}} \mathcal{P}_{j-\frac{3}{2}}, \tag{21f}$$

where

$$\mathcal{N}_{j+\frac{1}{2}} = \omega_{j+\frac{1}{2}} - \gamma_{j+\frac{1}{2}}, \qquad \mathcal{P}_{j+\frac{1}{2}} = \omega_{j+\frac{1}{2}} + \gamma_{j+\frac{1}{2}}. \tag{21g}$$

If all the coefficients in (21a) except A_0 are nonnegative, it is easy to obtain a strong local maximal-minimal principle when the time step Δt is small enough. As a consequence, the nonnegativity preserving principle for the LDG solution of the PME will hold in this case. However, we could not ensure the positivity of these coefficients in general. For example, $A_1 = \frac{1}{4} h_j^{-2} \omega_{j+\frac{1}{2}}(-\omega_{j-\frac{1}{2}} + 2\omega_{j+\frac{1}{2}} + \omega_{j+\frac{3}{2}})$, when $\gamma_{j+\frac{1}{2}} \equiv 0$ and a uniform mesh is used. This coefficient then depends on the numerical solution in three adjacent cells, making it difficult to consider its sign in general.

Before further analysis, we would like to consider a specific choice of the flux's parameter. In the following analysis, we take $\gamma_{j+\frac{1}{2}} = \theta_{j+\frac{1}{2}}\omega_{j+\frac{1}{2}}$, and demand $\theta_{j+\frac{1}{2}} = \theta$ to be a constant, just like what we have done in our simulations. It is also the choice that we have made in the numerical simulation. Then, (21a) yields that

$$\begin{aligned}\Box u_j = &\left[\frac{1}{4} - \theta^2\right] h_j^{-1}(h_{j+1}^{-1}\omega_{j+\frac{1}{2}} g_{j+2} + h_{j-1}^{-1}\omega_{j-\frac{1}{2}} g_{j-2}) \\ &+ \left[\frac{1}{4} - \theta^2\right] h_j^{-2}(\omega_{j+\frac{1}{2}} - \omega_{j-\frac{1}{2}})(g_{j+1} - g_{j-1}) \\ &+ [h_j^{-1}(2\theta^2 + \theta) + h_{j-1}^{-1}(2\theta^2 - \theta)] h_j^{-1}\omega_{j-\frac{1}{2}}(g_{j-1} - g_j) \\ &+ [h_j^{-1}(2\theta^2 - \theta) + h_{j+1}^{-1}(2\theta^2 + \theta)] h_j^{-1}\omega_{j+\frac{1}{2}}(g_{j+1} - g_j) \\ &- \left[\frac{1}{4} - \theta^2\right] h_j^{-1}(h_{j-1}^{-1}\omega_{j-\frac{1}{2}} + h_{j+1}^{-1}\omega_{j+\frac{1}{2}}) g_j \\ = &\,\Theta_1 + \Theta_2 + \Theta_3 + \Theta_4 + \Theta_5. \end{aligned} \tag{22}$$

In what follows we will use this formula to prove that the solution in each cell I_j is also nonnegative at t^{n+1}, if $u^n \geq 0$ and the time step is small enough. Moreover, we will point out that the parameter θ must be taken with some restriction for the nonnegativity preserving principle to hold; otherwise, the numerical solution may become negative, see the counterexample in Remark 4.1.

To obtain the nonnegativity of u_j^{n+1}, we are going to estimate the right-hand side of formula (22) for the nonnegative solution u^n. With this assumption, u_j, ω_j and g_j are all nonnegative for $j = 1, 2, \ldots, N$. For the PME, $a(u) = mu^{m-1}$ and $g(u) = \frac{2\sqrt{m}}{m+1} u^{\frac{m+1}{2}}$, both of which are increasing with respect to u. Then it follows that

$$\max_j \omega_{j+\frac{1}{2}} \leq \sqrt{\max a(u)}, \qquad \max_{j, u_j \neq 0} \frac{g_j}{u_j} \leq \frac{2}{m+1}\sqrt{\max a(u)}, \tag{23}$$

where the maximum is taken from the interval covered by the numerical solution u^n.

The following estimates depend on the status of u_j with respect to the numerical solution in adjacent cells. Three cases will be considered below.

Case 1. Suppose the solution u_j is a local minimum, namely, $u_j \geq 0$ and $u_{j\pm1} \geq u_j$. The relationship between u_j and $u_{j\pm2}$ is not restricted to get the nonnegativity preserving principle, since there always holds $\Theta_1 \geq 0$ for any case, provided $|\theta| \leq \frac{1}{2}$.

To estimate the second term Θ_2, we define the following function

$$H(u) = \frac{g(u) - g(u_j)}{u - u_j} = \frac{\int_{u_j}^{u} \sqrt{a(u)}\,\mathrm{d}u}{u - u_j}, \quad u \in \mathbb{R}^+,$$

and hence $\omega_{j\pm\frac{1}{2}} = H(u_{j\pm1})$. It is easy to prove that this function is increasing with respect to $u \geq 0$, since, for the PME, $\sqrt{a(u)}$ is an increasing function. Then it follows that Θ_2 is nonnegative, whether u_{j+1} is smaller than u_{j-1} or not.

Now we turn to consider the nonnegativity of Θ_4 and Θ_5. This does not hold in general for any $|\theta| \leq \frac{1}{2}$. The nonnegativity is also affected by the ratio of the mesh sizes of the two

adjacent cells, i.e., the constant θ needs to satisfy

$$\theta \in \Big[-\frac{1}{2}, \frac{\delta_{min}-1}{2(\delta_{min}+1)}\Big] \cup \{0\} \cup \Big[\frac{\delta_{max}-1}{2(\delta_{max}+1)}, \frac{1}{2}\Big], \tag{24}$$

where $\delta_{max} = \max\{\max_j h_j/h_{j-1}, 1\}$ and $\delta_{min} = \min\{\min_j h_j/h_{j-1}, 1\}$. This restriction ensures that all term in the square bracket of formula (22) are nonnegative. Thus both Θ_4 and Θ_5 are nonnegative if u_j is a local minimum. It is worthy to mention that this restriction is necessary; see the counterexample in Remark 4.1.

We estimate the last term Θ_5 in two cases.

Case 1(a). If this local minimal $u_j = 0$, then $g_j = 0$ and $\Theta_5 = 0$. Then the right-hand side of (22) is obviously not smaller than zero, which reveals that the solution in this cell will not decrease, consequently, it will be nonnegative at t^{n+1}.

Case 1(b). If this local minimum $u_j > 0$, we can not make sure whether this solution will increase or not. But we can make sure that it does not become negative if the time step is small enough. Since only the last term in formula (22) may be negative, we have

$$\Box u_j \geq -\frac{1}{4}h_j^{-1}(h_{j+1}^{-1}\omega_{j+\frac{1}{2}} + h_{j-1}^{-1}\omega_{j-\frac{1}{2}})\frac{g_j}{u_j}\cdot u_j \geq -\frac{\max a(u)}{(m+1)\rho^2}\cdot u_j,$$

where $\rho = \min_{j=1,\ldots,N}\Delta x_j$ is the minimum of cell sizes. Consequently,

$$u_j^{n+1} \geq \Big[1 - \frac{\lambda \max a(u)}{m+1}\Big]u_j \geq 0,$$

if the time step is small such that $\lambda \max a(u) \leq 1$; here $\lambda = \Delta t/\rho^2$ and $m > 1$. This restriction for the time step is independent of how small the numerical solution is.

Case 2. Suppose u_j is a local maximum, namely, $0 \leq u_{j\pm1} \leq u_j$. The analysis for this case follows the same line as that for Case 1(b). In this case, Θ_1 and Θ_2 are still nonnegative. The other terms in the right-hand side of (22) are negative, since the restriction (24) ensures that all term in the square bracket of formula (22) are nonnegative. Note that $2\theta^2 \pm \theta$ have different sign and both lie in $[-1, 1]$, since $|\theta| \leq \frac{1}{2}$. Thus we have that

$$\Theta_3 \geq -[h_j^{-1}(2\theta^2+\theta) + h_{j-1}^{-1}(2\theta^2-\theta)]h_j^{-1}\omega_{j-\frac{1}{2}}g_j \geq -\rho^{-2}\omega_{j-\frac{1}{2}}g_j,$$

$$\Theta_4 \geq -[h_j^{-1}(2\theta^2-\theta) + h_{j+1}^{-1}(2\theta^2+\theta)]h_j^{-1}\omega_{j+\frac{1}{2}}g_j \geq -\rho^{-2}\omega_{j+\frac{1}{2}}g_j,$$

$$\Theta_5 \geq -\frac{1}{4}\rho^{-2}(\omega_{j+\frac{1}{2}} + \omega_{j-\frac{1}{2}})g_j.$$

Summing up the above estimate, finally we have that

$$\Box u_j \geq -\frac{5}{4}\rho^{-2}(\omega_{j+\frac{1}{2}} + \omega_{j-\frac{1}{2}})g_j \geq -\frac{5\max a(u)}{(m+1)\rho^2}\cdot u_j,$$

thus the solution will not become negative if the time step is small such that $\lambda \max a(u) \leq \frac{1}{5}$.

Case 3. Suppose u_j is not a local maximum or minimum, namely, the three solution values are placed in a monotone fashion, either $u_{j+1} \geq u_j \geq u_{j-1}$ or in reverse. Note that $u_j > 0$ in either case. The analysis for these two cases are the same, so we will just take the former as an example. In this case, Θ_1, Θ_2 and Θ_4 are nonnegative. Thus, it follows from the analysis in Case 2 that

$$\Box u_j \geq \Theta_3 + \Theta_5 \geq -\frac{3\max a(u)}{(m+1)\rho^2} \cdot u_j,$$

and the solution will not become negative if the time step is small such that $\lambda \max a(u) \leq \frac{1}{3}$.

We can now conclude that the numerical solution in each cell will never become negative, if the time step is small such that $\lambda \max a(u) \leq \frac{1}{5}$. Summing up the above analysis, we have the following theorem.

Theorem 4.1 *Assume, in the LDG scheme* (8)*, that the flux's parameter* $\gamma = \theta\omega$ *with the constant* θ *under the restriction* (24)*. If the initial numerical solution is nonnegative, then the numerical solution given by the above LDG method with the discontinuous* $\mathbb{P}^0$ *finite element is nonnegative for all time, if the time step is small such that* $\lambda \max a(u) \leq \frac{1}{5}$.

Remark 4.1 The flux's parameter must be taken carefully for the nonnegativity preserving principle to hold, even for the discontinuous $\mathbb{P}_0$ finite element. To show that, we will consider the PME with the parameter $m = 3$ and give some counterexamples to destroy the nonnegativity.

Example 1 If the flux's parameter is taken as a constant, i.e., $\gamma_{j+\frac{1}{2}} = \gamma$, only $\gamma = 0$ is good for the simulation of the PME. Otherwise, the solution may become negative. For example, for a small positive constant γ, consider the solution defined on the uniform mesh with piecewise constants, $u_{j-2} = u_{j-1} = u_j = u_{j+1} = 0$ and $u_{j+2} = \gamma$. It follows from formula (21a) that $\Box u_j = (\frac{\sqrt{3}}{2} - 1)\gamma^2 < 0$, which implies that the solution in the cell I_j will become negative in the next time level.

Example 2 If the flux's parameter is taken in the form $\gamma = \theta\omega$, the nonnegativity preserving principle may be destroyed for certain non-uniform meshes. For example, consider the piecewise constant solution, $u_{j-2} = u_j = u_{j+2} = 0$ and $u_{j+1} = u_{j-1} = 1$. Let $\theta = \frac{1}{4}$ and the adjacent mesh sizes satisfy $h_{j+1} = 4h_j = 16h_{j-1}$, then $\Box u_j = -\frac{15}{128}h_j^{-2} < 0$. This implies that u_j^{n+1} is negative for any time step. Note that $\theta = \frac{1}{4} \notin [-\frac{1}{2}, 0] \cup [\frac{3}{10}, \frac{1}{2}]$, i.e. it does not satisfy the restriction (24).

Remark 4.2 In this paper we present the proof of the nonnegativity preserving property only for piecewise constant. However, we can not get that for the general piecewise polynomials, along the same line. Because it is more difficult now to copy with the relation between the boundary values and the cell average, which is trivial for piecewise constant. This proof for general case is our future work.

5 Numerical Simulation to the Porous Medium Equation

In this section we will present some numerical results to the PME, given by the LDG scheme. In all simulations, we use $\mathbb{P}_2$ finite element space defined on the uniform mesh. The flux's

parameter is $\gamma = \frac{1}{2}\omega$, and number of Gaussian points used in the nonnegativity preserving limiter is $N_p = 3$. For almost every experiments, we use the third-order TVD-ERK time-marching with CFL $= 0.02$.

The following numerical results are composed of two parts. In the first part we will show that the LDG scheme is good at simulating the energy solution of PME, especially the Barenblatt solution; in the second part we will use this effective method to verify some interesting phenomena for the PME.

5.1 Numerical Results to the Barenblatt Solution

We first present some numerical results to show the effectiveness of the LDG method. To do that, we begin our simulation for the Barenblatt solution of the PME (1), where the initial condition is taken as the Barenblatt solution at $t = 1$, and the boundary condition is $u(\pm 6, t) = 0$ for $t > 1$.

We divide the computation domain into $N = 320$ uniform cells, and plot, in Fig. 3, the numerical solution at $t = 2$. Here the square-box is the numerical solution (plotted one point per cell) and the solid line is the exact solution. The parameters for the PME are taken as $m = 2, 3, 5$ and 8, respectively, from top-left and bottom-right. Comparing with Figs. 1 and 2, this figure shows that the LDG scheme can simulate the Barenblatt solution, accurately and sharply, without noticeable oscillations near the interface.

We now pay more attention to the movement of the numerical interface, to check whether the LDG scheme has the ability to capture the true interface accurately. The position of the numerical interface is detected as follows: given a tolerance $\epsilon > 0$, for example $\epsilon = 0.0001$ in this paper, we scan the numerical solution from right to left, and find the first cell in which the average is greater than the tolerance ϵ. We then conclude the interface emerges in this cell, and define the *left* endpoint of this cell as the numerical interface.

We plot in Fig. 4 the evolution of the numerical interface for the Barenblatt solution, with four different parameters $m = 2, 3, 5$ and 8, from $t = 1$ to $t = 2$. Here the solid line is the position of the exact interface, and the square-box is the position of the numerical interface. This figure verifies that the LDG method is very accurate at capturing the moving interface.

At the end of this subsection, we point out that the accuracy is holding in the smooth part of the solution. The accuracy table (Table 1) is given for the Barenblatt solution of PME (1) with parameter $m = 8$, where the considered domain is $[-1.5, 1.5]$ and the final time

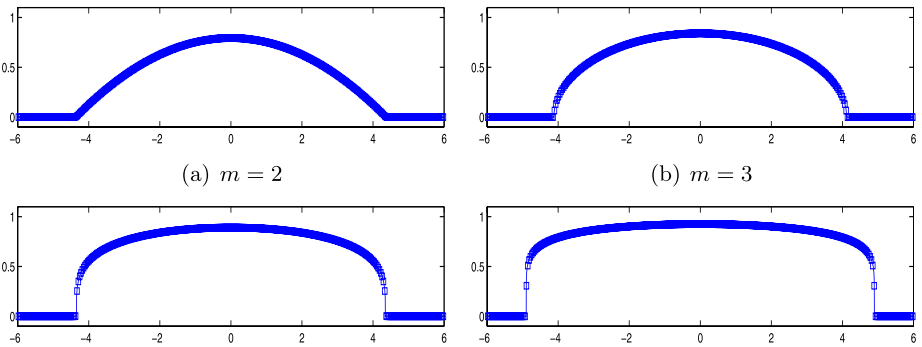


Fig. 3 Numerical results for the Barenblatt solution by the LDG method: $t = 2$

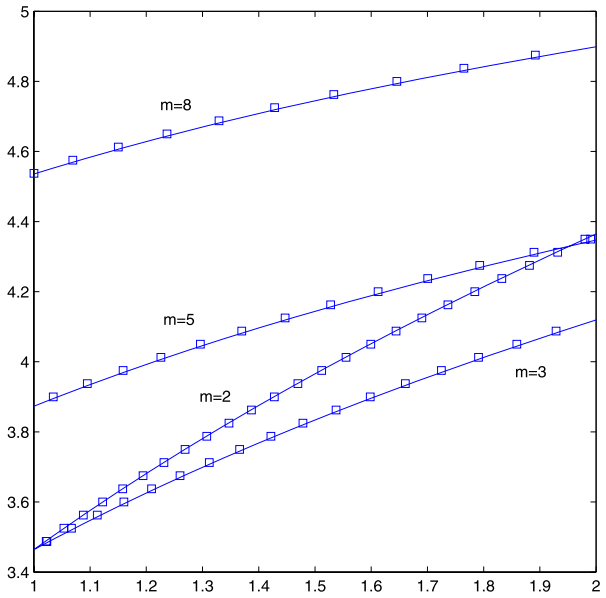


Fig. 4 Movement of the numerical interface for the Barenblatt solution: $m = 2, 3, 5, 8$

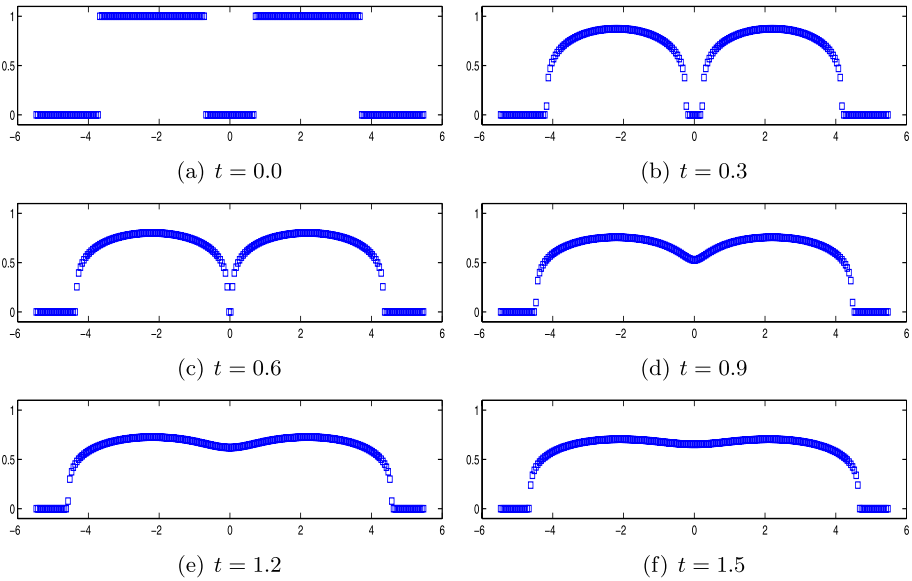


Fig. 5 Collision of the two-Box solution with the same height

is $t = 1.05$. It shows that the present method has formally high order accuracy for smooth solution, although there exists a sharp layer and the limiter is needed.

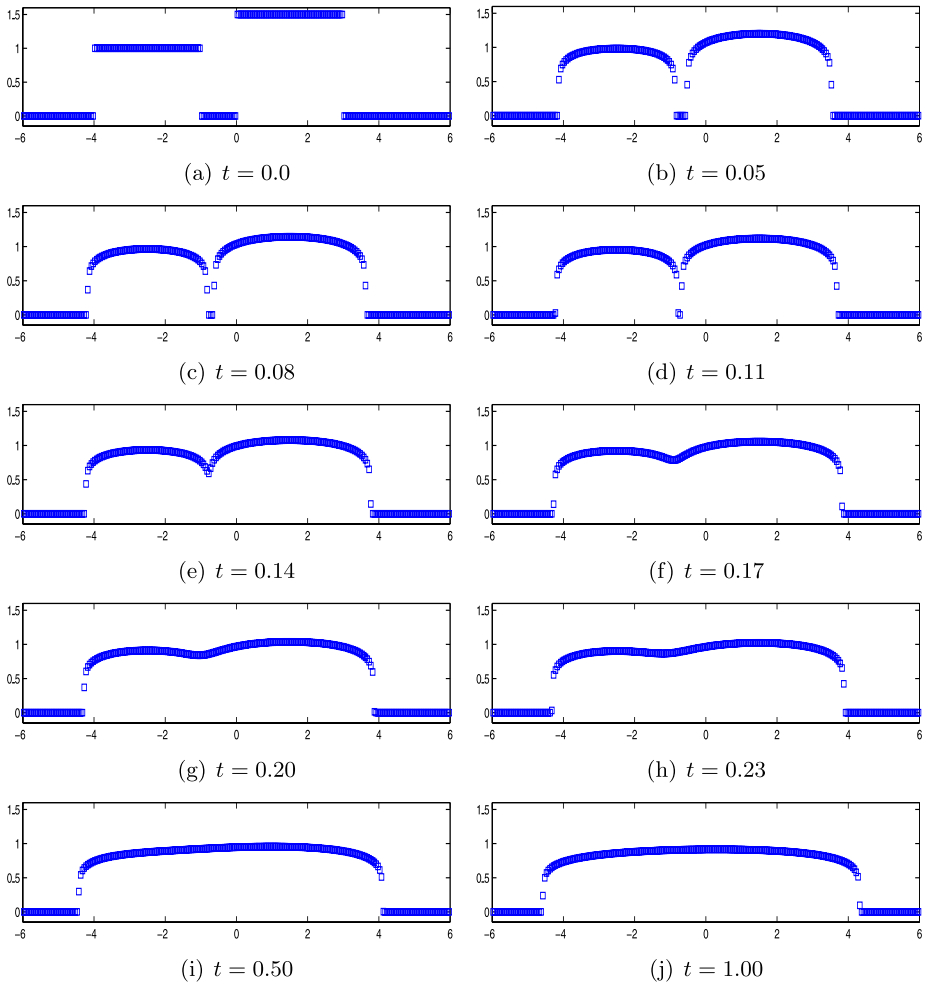


Fig. 6 Collision of the two-Box solution with different heights

Table 1 Accuracy for the smooth part of the solution

	# of cells	L_∞ error	L_∞ order	L_2 error	L_2 order
$m = 8$	40	2.80(–6)	–	8.40(–7)	–
	80	3.75(–7)	2.89	1.06(–7)	2.99
	160	4.87(–8)	2.95	1.32(–8)	3.00
	320	6.21(–9)	2.97	1.65(–9)	3.00
	640	7.84(–10)	2.99	2.07(–10)	3.00

5.2 Numerical Simulation for Additional Interesting Cases

After we have been convinced that the LDG scheme is very good at simulating the energy solutions of the PME, we will continue applying the LDG scheme in more simulation for

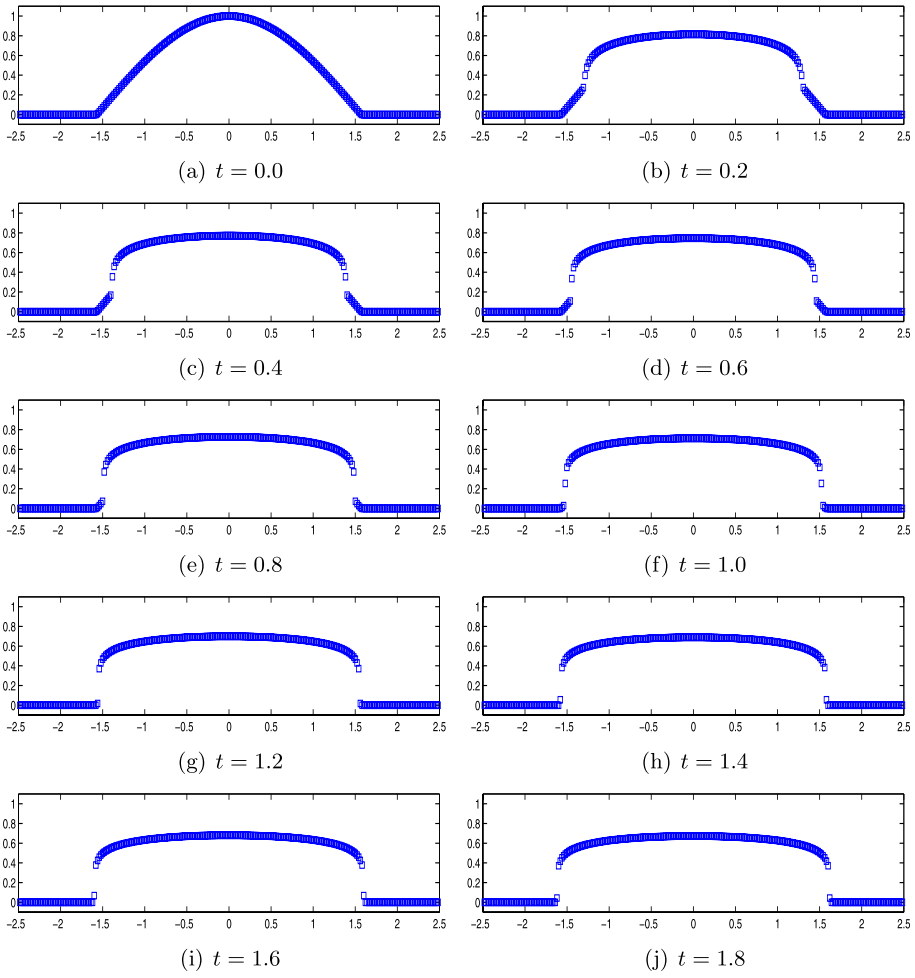


Fig. 7 Waiting-time phenomenon

the PME, to verify some interesting phenomena, such as the collision of two Box waves, the waiting time for the interface, and the splitting of the support.

Case 1. We first consider the collision of two Box solutions with the same or different heights. If the variable u is regarded as the temperature, this model case is used to describe how the temperature changes when two hot spots are suddenly put in the computation domain.

In Fig. 5 we plot the evolution of the numerical solution for the PME with $m = 5$. The initial condition is the two-Box solution with the same height, namely

$$u_0(x) = \begin{cases} 1, & \text{if } x \in (-3.7, -0.7) \cup (0.7, 3.7), \\ 0, & \text{otherwise;} \end{cases} \tag{25}$$

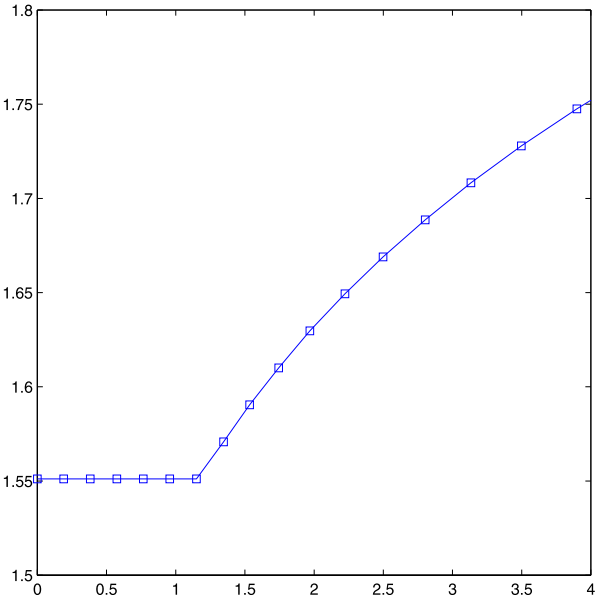


Fig. 8 Waiting-time phenomenon: the movement of the numerical interface from $t = 0$ to $t = 4$. The first raising square is pointed to the waiting time $t = 1.345293$

and the boundary condition is $u(\pm 5.5, t) = 0$ for all time. The computational domain is divided into $N = 220$ uniform cells with mesh size $\Delta x = 0.05$.

We also consider the collision of the two-Box solution with different heights. Plotted in Fig. 6 is the evolution of the numerical solution for the PME with the parameter $m = 8$. The initial condition is defined as

$$u_0(x) = \begin{cases} 1, & \text{if } x \in (-4, -1), \\ 1.5, & \text{if } x \in (0, 3), \\ 0, & \text{otherwise;} \end{cases} \tag{26}$$

and the boundary condition is $u(\pm 6, t) = 0$ for all time. The computational domain is divided into $N = 240$ uniform cells with mesh size $\Delta x = 0.05$.

From these simulations, we can see an analogous evolution whether the heights of the two boxes in the initial condition is the same or not. Two-Box solutions first move outward independently before the collision, then they join each other to make the temperature smooth, and finally the solution becomes almost constant in the common support.

Case 2. From the simulation for the Barenblatt solution and the Box solution, we can see that the interface of the support is moving outward immediately. But this is not always true for arbitrary initial conditions. Angenent proved in [1] that a waiting time exists in some cases, and the interface of the support does not move outward until the waiting time.

To verify this by the LDG method, we consider the PME with $m = 8$. The initial condition is defined as a fast-varying solution, namely,

$$u_0(x) = \begin{cases} \cos x, & \text{if } x \in (-\frac{\pi}{2}, \frac{\pi}{2}), \\ 0, & \text{otherwise;} \end{cases} \tag{27}$$

and the boundary condition is $u(\pm\pi, t) = 0$. Plotted in Fig. 7 is the evolution of the numerical simulation, where the computational domain is divided into $N = 320$ uniform cells.

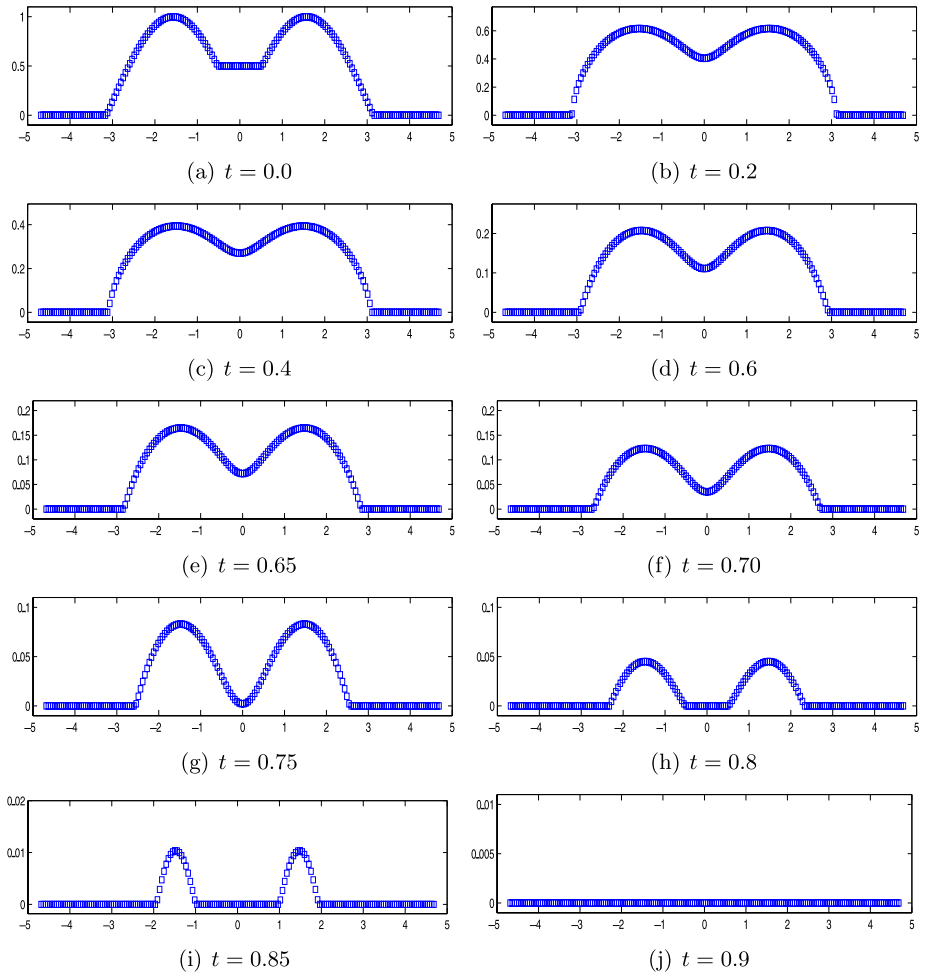


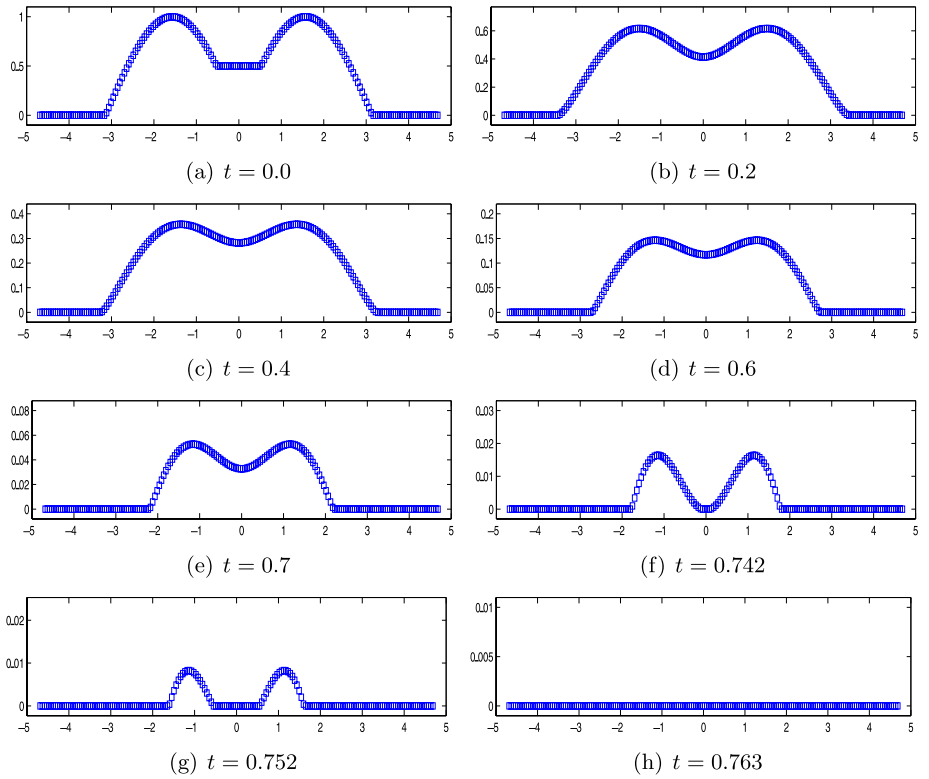
Fig. 9 Support splitting phenomenon (i): $m = 3$, $p = 0.1$

One can see that the interface does not move outward before the so-called waiting-time; see Fig. 7(a) to (g). After that, the interface point moves outward with a finite speed; see Fig. 7(h) and (j). It seems from the numerical results that the solution tends to the Barenblatt-type solution before the waiting-time. Whether this is true analytically or not is still an open problem, to our best knowledge.

To describe this phenomenon more clearly, we have plotted, in Fig. 8, the evolution of the numerical interface from $t = 0$ to $t = 4$. The first raising square is pointed to the waiting time $t = 1.345293$.

Case 3. In the last simulation, we consider some modification to the PME. For example, we have added a strong absorption and considered the following equation

$$\frac{\partial u}{\partial t} = (u^m)_{xx} - cu^p, \tag{28}$$



(a) $t = 0.0$ (b) $t = 0.2$

(c) $t = 0.4$ (d) $t = 0.6$

(e) $t = 0.7$ (f) $t = 0.742$

(g) $t = 0.752$ (h) $t = 0.763$

Fig. 10 Support splitting phenomenon (ii): $m = 1.5$, $p = 0.05$

where $0 < p < 1$ and $c > 0$ are two given constants. Subjecting to zero boundary condition, the steady state vanishes and the solution tends to zero as time goes to infinity. The interesting phenomenon is that in this case, absorption can cool the medium faster than diffusion supplies heat from the hot area, and the support shrinks and becomes disconnected, even if the initial value is positive in the support. Rosenau and Kamin [20] first found this "divorce of support" in the solutions, and Nakaki and Tomoeda [16] proved the existence of this phenomenon for the special case that $m + p = 2$ and $0 < p < 1$.

In this paper we would like, by numerical simulations using the LDG method, to find out whether this phenomenon still exists in other cases. To do that, we take the same initial condition

$$u_0(x) = \begin{cases} |\sin x|, & x \in (-\pi, -\pi/6) \cup (\pi/6, \pi), \\ 0.5, & x \in (-\pi/6, \pi/6), \\ 0, & \text{otherwise}; \end{cases} \tag{29}$$

and the boundary condition $u(\pm 3\pi/2, t) = 0$ for all time, and consider three groups of parameters in (28), namely, (i) $m = 3$, $p = 0.1$ and $c = 1$; (ii) $m = 1.5$, $p = 0.05$ and $c = 1$; and (iii) $m = 1.92$, $p = 0.08$ and $c = 1$. These setting are used for representations for different case in which $m + p$ is bigger than, small than, or equal to 2. It is worthy to mention that this phenomenon may not happen if the parameters are not given well.

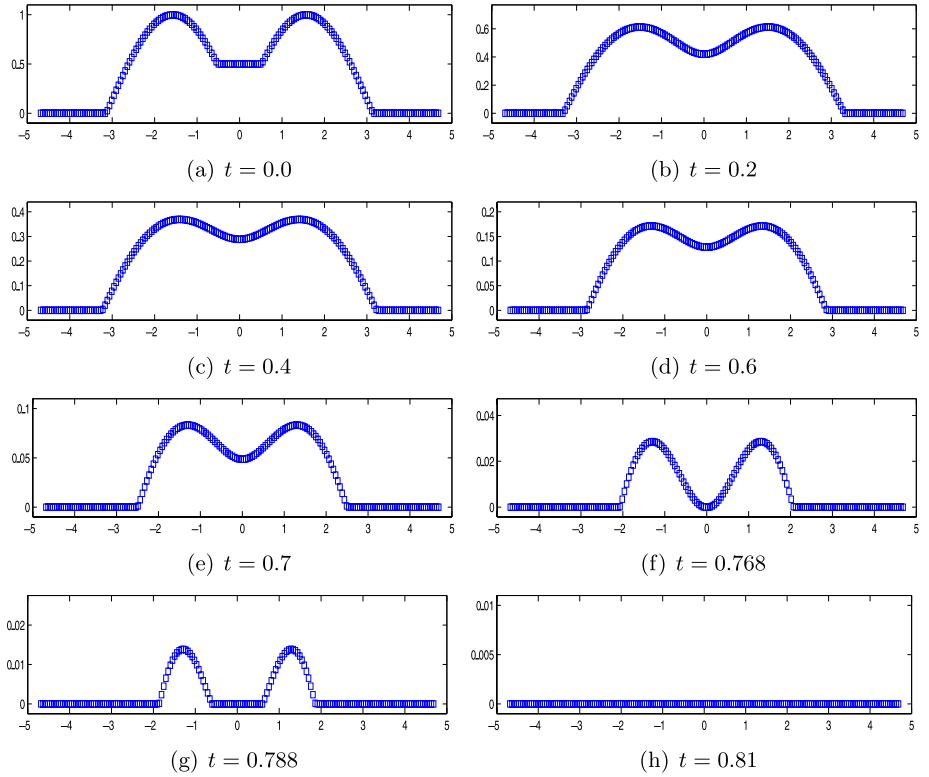


Fig. 11 Support splitting phenomenon (iii): $m = 1.92$, $p = 0.08$

The evolution of the numerical solution is plotted in Figs. from 9 to 11, respectively, for the above three groups of parameters; the computational domain is divided into $N = 180$ uniform cells, and the time step is 10^{-6}. We can clearly see the splitting of the support for these set of parameters. The two peaks and the bridge between them are dropping continuously as time goes on, and the support splitting phenomenon emerges. After that, the two branches of the split support leave each other, and the peaks decrease continuously until the solution finally tends to zero.

6 Conclusion

In this paper we consider the simulation of the PME by the LDG method. Numerical results show that the LDG method can give satisfying simulation results with a sharp and non-oscillatory numerical interface of the support. Moreover, the cell average in each cell of the LDG solution for the PME satisfies a nonnegativity preserving principle, which is only proved for the discontinuous $\mathbb{P}_0$ finite element in this paper. We are going to prove this property for general piecewise polynomials in the further work. Because of the discontinuous finite element space, we can use a nonnegativity preserving limiter to repair any negative solution which violates the physical nature. Finally, we use this LDG method to verify three interesting phenomena for the PME, namely the collision of the two-Box waves,

the waiting time for the interface, and the splitting of the support. The simulation for the two-dimensional porous medium equations is our ongoing project.

References

1. Angenent, S.: Analyticity of the interface of the porous medium equation after waiting time. Proc. Am. Math. Soc. **102**, 329–336 (1988)
2. Bassi, F., Rebay, S.: A high-order accurate discontinuous finite element method for the numerical solution of compressible Navier-Stokes equations. J. Comput. Phys. **131**, 267–279 (1997)
3. Cockburn, B., Hou, S., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. Math. Comput. **54**, 545–581 (1990)
4. Cockburn, B., Karniadakis, G.E., Shu, C.-W. (ed.): Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering, vol. 11. Springer, Berlin (2000)
5. Cockburn, B., Lin, S.Y., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. J. Comput. Phys. **84**, 90–113 (1989)
6. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin finite element method for convection-diffusion systems. SIAM J. Numer. Anal. **35**, 2440–2463 (1998)
7. Cockburn, B., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math. Comput. **52**, 411–435 (1989)
8. Cockburn, B., Shu, C.-W.: The Runge-Kutta local projection P^1-discontinuous Galerkin method for scalar conservation laws. RAIRO Anal. Numér. **25**, 337–361 (1991)
9. Cockburn, B., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998)
10. Cockburn, B., Shu, C.-W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. **16**, 173–261 (2001)
11. DiBenedetto, E., Hoff, D.: An interface tracking algorithm for the porous medium equation. Trans. Am. Math. Soc. **284**, 463–500 (1984)
12. Graveleau, J.L., Jamet, P.: A finite difference approach to some degenerate nonlinear parabolic equations. SIAM J. Appl. Math. **20**, 199–223 (1971)
13. Gottlieb, S., Shu, C.-W.: Total variation diminishing Runge-Kutta schemes. Math. Comput. **67**, 73–85 (1998)
14. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability preserving high order time discretization methods. SIAM Rev. **43**, 89–112 (2001)
15. Jin, S., Pareschi, L., Toscani, G.: Diffusive relaxation schemes for multiscale discrete-velocity kinetic equations. SIAM J. Numer. Anal. **35**(6), 2405–2439 (1998)
16. Nakaki, T., Tomoeda, K.: A finite difference scheme for some nonlinear diffusion equations in an absorbing medium: support splitting phenomena. SIAM J. Numer. Anal. **40**, 945–954 (2002)
17. Osher, S.: Riemann solvers, the entropy condition, and difference approximations. SIAM J. Numer. Anal. **21**, 217–235 (1984)
18. Osher, S.: Convergence of generalized MUSCL schemes. SIAM J. Numer. Anal. **22**, 947–961 (1984)
19. Reed, W.H., Hill, T.R.: Triangular mesh methods for the neutron transport equation. Los Alamos Scientific Laboratory report LA-UR-73-479, Los Alamos, NM (1973)
20. Rosenau, P., Kamin, S.: Thermal waves in an absorbing and convecting medium. Physica D **8**, 273–283 (1983)
21. Shu, C.-W.: TVB uniformly high-order schemes for conservation laws. Math. Comput. **49**, 105–121 (1987)
22. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
23. Xu, Y., Shu, C.-W.: Error estimates of the semi-discrete local discontinuous Galerkin method for nonlinear convection-diffusion and KdV equations. Comput. Methods Appl. Mech. Eng. **196**, 3805–3822 (2007)