Stability analysis and error estimates of local discontinuous Galerkin methods for convection-diffusion equations on overlapping meshes[1]

Jie Du[2], Yang Yang[3] and Eric Chung[4]

Abstract

Local discontinuous Galerkin (LDG) methods are popular for convection-diffusion equations. In LDG methods, we introduce an auxiliary variable p to represent the derivative of the primary variable u, and solve them on the same mesh. In this paper, we will introduce a new LDG method, and solve u and p on different meshes. The stability and error estimates will be investigated. The new algorithm is more flexible and flux-free for pure diffusion equations without introducing additional computational cost compared with the original LDG methods, since it is not necessary to solve each equation twice. Moreover, it is possible to construct third-order maximum-principle-preserving schemes based on the new algorithm [12]. However, one cannot anticipate optimal accuracy in some special cases. In this paper, we will find out the reason for accuracy degeneration which further leads to several alternatives to obtain optimal convergence rates. Finally, several numerical experiments will be given to demonstrate the stability and optimal accuracy of the new algorithm.

Key Words: Stability, Error estimates, Convection-diffusion equations, Local discontinuous Galerkin method, Overlapping meshes.

1   Introduction

In this paper, we aim to construct new local discontinuous Galerkin (LDG) schemes for solving the following nonlinear convection-diffusion equation

$$u_t + f(u)_x = b(u)_{xx}, \tag{1.1}$$

or equivalently

$$u_t + f(u)_x = (a^2(u)u_x)_x, \tag{1.2}$$

as well as their two-dimensional versions, where $a^2(u) = b'(u) \geq 0$. We also assume that $a(u) \geq 0$. The initial condition is given as $u(x,0) = u_0(x)$.

The discontinuous Galerkin (DG) method was first introduced in 1973 by Reed and Hill [24] in the framework of neutron linear transport. Subsequently, Cockburn et al. developed Runge-Kutta

---

[2]Yau Mathematical Sciences Center, Tsinghua University, Beijing, China. E-mail: jdu@mail.tsinghua.edu.cn

[3]Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931. E-mail: yyang7@mtu.edu

[4]Department of Mathematics, The Chinese University of Hong Kong, Hong Kong SAR. E-mail: tschung@math.cuhk.edu.hk

discontinuous Galerkin (RKDG) methods for hyperbolic conservation laws in a series of papers [7, 5, 6, 8]. In [9], Cockburn and Shu introduced the LDG method to solve the convection-diffusion equations. Their idea was motivated by Bassi and Rebay [1], where the compressible Navier-Stokes equations were successfully solved. As in traditional LDG methods, we introduce an auxiliary variable p to represent $a(u)u_x$ and thus can rewrite (1.2) into the following system of first order equations

$$\begin{cases} u_t + f(u)_x = (a(u)p)_x, \\ p = A(u)_x, \end{cases} \tag{1.3}$$

where $A(u) = \int^u a(t)\, dt$. Usually, u and p are solved on the same mesh.

The LDG method is one of the most important numerical methods for convection-diffusion equations. However, for some special convection-diffusion systems, such as chemotaxis model [19, 22] and miscible displacements in porous media [10, 11], the LDG methods are not easy to construct and analyze. In each of the two models, the convection term is the product of one of the primary variables and the derivatives of the another primary variable. Most of the well established numerical fluxes for the convection terms, such as the upwind fluxes, cannot be applied, since the coefficients of the convection terms turn out to be discontinuous after the spatial discretization. It is well known that hyperbolic equations with discontinuous coefficients are in general not well-posed [13, 18]. Therefore, the DG schemes may not be stable when applied to those model equations. Within the DG framework, there are three different ways to bridge this gap. Firstly, in [30, 15, 20] the authors combined the convection terms and diffusion terms together and obtain the optimal error estimates. The idea was motivated by Wang et. al. [25, 26, 27], where u_x and the jump of u across the cell interfaces were proved to be bounded by p. Moreover, to make the numerical solutions to be physically relevant, we have to add a very large penalty which depends on the numerical approximations of the derivatives of the primary variables [20, 16]. The second approach is to applied the flux-free numerical methods such as the Central DG (CDG) methods [21]. However, for CDG methods, we have to solve each equation in (1.3) on both the primary and dual meshes, which double the computational cost. The last idea is to apply the Staggered DG (SDG) methods [3]. However, the method requires some continuity of the numerical approximations, which is not easy to apply limiters. In this paper, we introduce a new LDG method, and solve u and p on the primitive and dual meshes, respectively, and do not require any continuity across the cell interfaces. Since p is continuous across the cell interfaces in the primitive mesh, we can apply the upwind fluxes for the convection term for the complicated systems discussed above.

Finally, the most significant advantage of the new algorithm is the construction of third-order maximum-principle-preserving (MPP) LDG methods. Recently, in [31], genuinely MPP high-order DG schemes for scalar equations and two-dimensional incompressible flows in vorticity-streamfunction formulation have been constructed. Subsequently, positivity-preserving high order DG schemes for compressible Euler equations were given in [32]. Later, the technique was applied to other hyperbolic systems, such as pressureless Euler equations [29], Extended MHD equations [34], relativistic hydrodynamics [23], etc, and the L^1 stability was demonstrated. For parabolic

equations, the extension was given in [33], where second-order MPP discontinuous Galerkin methods were demonstrated, and the construction of third-order schemes seem to be not straightforward. Later another approach based on the flux limiter were discussed in [28, 17]. In [2], the third-order MPP direct DG method was introduced. However, the scheme was not easy to implement and we need to add two penalty terms. In [12], we applied the new LDG method and constructed third-order MPP schemes. There is a mild penalty which does not depend on the numerical approximations. Since the dual meshes can be moved arbitrarily, we also showed that if the dual mesh agree with the primitive mesh, the penalty coefficient turns out to be infinity. Therefore, our algorithm does not violate the results given in [33].

In contrast to the above advantages, the new LDG method may not have optimal convergence rates when applied to the pure diffusion equations if the dual mesh is generated by the midpoint in each cell on the primitive mesh and piecewise odd order polynomials are applied. This is the main reason why in the SDG method the numerical approximations are required to be continuous across some of the cell interfaces. In this paper, we will theoretically study the stability and error estimates of new algorithm. We will demonstrate the reason for the accuracy degeneration and introduce several alternatives to gain the optimal convergence rates.

The organization of this paper is as follows. In Section 2, we construct the new LDG scheme for nonlinear convection-diffusion equations on overlapping meshes in one space dimension. The stability and error estimates will be given. The extension to problems in two space dimensions will be discussed in Section 3. Numerical experiments will be given in Section 4 to demonstrate the accuracy and good performance of the new LDG scheme. Finally, we will end in Section 5 with concluding remarks.

2 Numerical scheme for one-dimensional case

In this section, we will introduce the new LDG method for solving the one-dimensional nonlinear convection-diffusion equation (1.3) on overlapping meshes.

2.1 Overlapping meshes

Different from the LDG method introduced in [9], in which u and p are solved on the same mesh, our new method solves (1.3) on two meshes, as shown in Figure 2.1. For simplicity, we consider periodic boundary conditions and the algorithms for other boundary conditions will be discussed in the future.
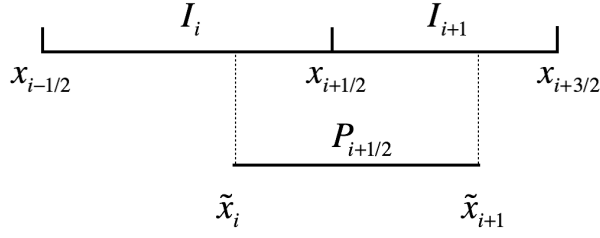
Figure 2.1: **Overlapping meshes**

We first define the primitive mesh on which the primary variable u is solved. It is just a decomposition of the computational domain $\Omega = [0, 1]$, which can be non-uniform. We denote the i-th cell as

$$I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \quad i = 1, \cdots, N_x.$$

The cell length and the cell center of I_i are denoted as

$$\triangle x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \qquad x_i = \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2},$$

respectively. We define $\Delta x = \max_i \Delta x_i$. In this paper, we consider regular meshes, i.e. there exists a positive constant $C \geq 1$ such that $\Delta x \leq C \min_i \Delta x_i$. Clearly, if $C = 1$, then the mesh is uniformly distributed.

Based on the primitive mesh, we move each cell center within the corresponding cell to obtain a new mesh called the P-mesh, which is used to solve the auxiliary variable p, i.e. in each cell I_i, we choose a point $\tilde{x}_i$ given as

$$\tilde{x}_i = x_i + \frac{\triangle x_i}{2} \xi_{i0}, \qquad \xi_{i0} \in [-1, 1], \quad i = 1, \cdots, N_x. \tag{2.1}$$

For simplicity, we consider ξ_{i0} to be a constant independent of i and denoted as $\xi_0 \in [-1, 1]$. It is easy to check $\tilde{x}_i \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. The $(i - \frac{1}{2})$-th cell of the dual mesh is defined as

$$P_{i-\frac{1}{2}} = [\tilde{x}_{i-1}, \tilde{x}_i], \quad i = 1, \cdots, N_x,$$

where we denote $\tilde{x}_0 = \tilde{x}_{N_x} - 1$. We further denote the cell length and the cell center of $P_{i-\frac{1}{2}}$ as

$$\triangle \tilde{x}_{i-\frac{1}{2}} = \tilde{x}_i - \tilde{x}_{i-1}, \qquad \tilde{x}_{i-\frac{1}{2}} = \frac{\tilde{x}_{i-1} + \tilde{x}_i}{2},$$

respectively. We can easily check that

$$\min\{\Delta x_{i-1}, \Delta x_i\} \leq \Delta \tilde{x}_{i-\frac{1}{2}} \leq \max\{\Delta x_{i-1}, \Delta x_i\},$$

and hence we have $\max_i \Delta x_{i-\frac{1}{2}} \leq \Delta x$. Due to the periodic boundary condition, we can also define $P_{\frac{1}{2}} = [0, \tilde{x}_1] \cup [\tilde{x}_{N_x}, 1]$. Therefore, we consider a polynomial on $P_{\frac{1}{2}}$ as a polynomial on $[\tilde{x}_0, \tilde{x}_1]$. We

define the dual mesh to be the P-mesh which consists of all these P cells. Notice that when $\xi_0 = 0$, we have $\tilde{x}_i = x_i$ and $P_{i+\frac{1}{2}} = [x_i, x_{i+1}]$. In this case, the cell interfaces of the dual mesh are exactly the cell centers of the primitive mesh. This kind of mesh is the most commonly used overlapping mesh, such as in the central discontinuous Galerkin (CDG) method [21]. When $\xi_0 = 1$, we have $\tilde{x}_i = x_{i+\frac{1}{2}}$ and hence the P-mesh is the same as the primitive mesh.

2.2 Norms

In this section, we proceed to define some norms to be used throughout the paper.

For any interval I, we define $\|u\|_I$ and $\|u\|_{\infty,I}$ to be the standard L^2- and L^∞-norm of u on I, respectively. For any natural number ℓ, we consider the norm of the Sobolev space $H^\ell(I)$, defined by

$$\|u\|_{\ell,I} = \left\{ \sum_{0 \le \beta \le \ell} \left\| \frac{\partial^\beta u}{\partial x^\beta} \right\|_I^2 \right\}^{\frac{1}{2}}.$$

For convenience, if I is the whole computational domain, then the corresponding subscript will be omitted.

Moreover, for any $u \in C(I_i)$, we define

$$\|u\|_{\Gamma_i} = |u^-_{i+\frac{1}{2}}| + |u^+_{i-\frac{1}{2}}|,$$

Similarly, for any $u \in C(P_{i-\frac{1}{2}})$, we define

$$\|u\|_{\Gamma_{i+\frac{1}{2}}} = |u^-_i| + |u^+_{i-1}|.$$

2.3 LDG method on overlapping meshes

In this section, we introduce the LDG methods for the following pure diffusion equation

$$\begin{cases} u_t = (a(u)p)_x, \\ p = A(u)_x, \end{cases} \tag{2.2}$$

where $A(u) = \int^u a(t)dt$. We define the finite element spaces to be

$$V_h := \{u_h : u_h|_{I_i} \in P^k(I_i),\ i = 1, \cdots, N_x\},$$
$$P_h := \{p_h : p_h|_{P_{i-\frac{1}{2}}} \in P^k(P_{i-\frac{1}{2}}),\ i = 1, \cdots, N_x\}.$$

By multiplying (2.2) with test functions and using the integration by parts, our new LDG method on overlapping meshes is defined as follows: find $(u_h, p_h) \in V_h \times P_h$, such that for any test functions $(v, w) \in V_h \times P_h$, we have

$$\int_{I_i} (u_h)_t v\, dx = -\int_{I_i} a(u_h) p_h v_x\, dx + \hat{a}_{i+\frac{1}{2}} \hat{p}_{i+\frac{1}{2}} v^-_{i+\frac{1}{2}} - \hat{a}_{i-\frac{1}{2}} \hat{p}_{i-\frac{1}{2}} v^+_{i-\frac{1}{2}}, \tag{2.3}$$

$$\int_{P_{i-\frac{1}{2}}} p_h w\, dx = -\int_{P_{i-\frac{1}{2}}} A(u_h) w_x\, dx + A(u_h(\tilde{x}_i)) w^-_i - A(u_h(\tilde{x}_{i-1})) w^+_{i-1}. \tag{2.4}$$

where $v_{i+\frac{1}{2}}^- = v^-(x_{i+\frac{1}{2}})$ and $w_i^- = w^-(\tilde{x}_i)$. Likewise for $v_{i-\frac{1}{2}}^+$ and w_{i-1}^+. For simplicity, we denote $v_{\frac{1}{2}}^- = v_{N_x+\frac{1}{2}}^-$ and $v_{N_x+\frac{1}{2}}^+ = v_{\frac{1}{2}}^+$. The numerical flux $\hat{a}$ at the point $x_{i+\frac{1}{2}}$ is taken as

$$\hat{a}_{i+\frac{1}{2}} = \frac{[A(u_h)]_{i+\frac{1}{2}}}{[u_h]_{i+\frac{1}{2}}}, \tag{2.5}$$

where $[s]_{i+\frac{1}{2}} := s_{i+\frac{1}{2}}^+ - s_{i+\frac{1}{2}}^-$ denotes the jump of a function s across the cell interface $x = x_{i+\frac{1}{2}}$. Similarly, we can also denote the jump of w across $x = \tilde{x}_i$ on the P-mesh as $[w]_i = w_i^+ - w_i^-$. Notice that p_h is continuous at the interfaces of the primitive cells and hence $p_h(x_{i+\frac{1}{2}})$ is well defined. We choose the numerical flux $\hat{p}_{i+\frac{1}{2}}$ as the value of p_h evaluated at $x = x_{i+\frac{1}{2}}$ with a penalty term

$$\hat{p}_{i+\frac{1}{2}} = p_h(x_{i+\frac{1}{2}}) + \frac{\alpha_{i+\frac{1}{2}}}{\triangle\tilde{x}_{i+\frac{1}{2}}}[u_h]_{i+\frac{1}{2}}. \tag{2.6}$$

Remark 2.1. *The parameter $\alpha_{i+\frac{1}{2}}$ in (2.6) is used for optimal rates of convergence and the maximum-principle-preserving technique introduced in [12].*

Finally, we would like to define

$$H_u(u_h, p_h, v) = -\sum_{i=1}^{N_x} \int_{I_i} a(u_h)p_h v_x dx - \sum_{i=1}^{N_x} \hat{a}_{i-\frac{1}{2}}\hat{p}_{i-\frac{1}{2}}[v]_{i-\frac{1}{2}}, \tag{2.7}$$

$$H_p(u_h, w) = -\sum_{i=1}^{N_x} \int_{P_{i-\frac{1}{2}}} A(u_h)w_x dx - \sum_{i=1}^{N_x} A(u_h(\tilde{x}_i))[w]_i, \tag{2.8}$$

which would be useful is the stability analysis and error estimates. With the above notations, the LDG scheme can be rewritten as

$$\int_\Omega (u_h)_t v dx = H_u(u_h, p_h, v), \tag{2.9}$$

$$\int_\Omega p_h w dx = H_p(u_h, w). \tag{2.10}$$

2.4 Stability analysis

In this section, we proceed to demonstrate the stability of the new LDG method. We would start with the following lemma.

Lemma 2.1. *Suppose H_u and H_p are defined in (2.7) and (2.8), respectively, then we have*

$$H_u(u_h, p_h, u_h) + H_p(u_h, p_h) = -\sum_{i=1}^{N_x} \frac{[A(u_h)]_{i-\frac{1}{2}}}{[u_h]_{i-\frac{1}{2}}} \frac{\alpha_{i-\frac{1}{2}}[u_h]_{i-\frac{1}{2}}^2}{\triangle\tilde{x}_{i-\frac{1}{2}}}. \tag{2.11}$$

Proof. Taking $w = p_h$ in (2.8) and using integration by parts, we obtain

$$H_p(u_h, p_h) = -\sum_{i=1}^{N_x} \int_{P_{i-\frac{1}{2}}} A(u_h)(p_h)_x dx - \sum_{i=1}^{N_x} A(u_h(\tilde{x}_i))[p_h]_i$$

$$= -\sum_{i=1}^{N_x} \int_{\tilde{x}_{i-1}}^{x_{i-\frac{1}{2}}} A(u_h)(p_h)_x dx - \sum_{i=1}^{N_x} \int_{x_{i-\frac{1}{2}}}^{\tilde{x}_i} A(u_h)(p_h)_x dx - \sum_{i=1}^{N_x} A(u_h(\tilde{x}_i))[p_h]_i$$

$$= \sum_{i=1}^{N_x} \int_{\tilde{x}_{i-1}}^{x_{i-\frac{1}{2}}} a(u_h)(u_h)_x p_h dx + \sum_{i=1}^{N_x} \int_{x_{i-\frac{1}{2}}}^{\tilde{x}_i} a(u_h)(u_h)_x p_h dx + \sum_{i=1}^{N_x} [A(u_h)]_{i-\frac{1}{2}} p_h(x_{i-\frac{1}{2}})$$

$$= \sum_{i=1}^{N_x} \int_{I_i} a(u_h)(u_h)_x p_h dx + \sum_{i=1}^{N_x} [A(u_h)]_{i-\frac{1}{2}} p_h(x_{i-\frac{1}{2}}). \tag{2.12}$$

Taking $v = u_h$ in (2.7), we obtain

$$H_u(u_h, p_h, u_h) = -\sum_{i=1}^{N_x} \int_{I_i} a(u_h) p_h (u_h)_x dx$$

$$- \sum_{i=1}^{N_x} \frac{[A(u_h)]_{i-\frac{1}{2}}}{[u_h]_{i-\frac{1}{2}}} \left(p_h(x_{i-\frac{1}{2}}) + \frac{\alpha_{i-\frac{1}{2}}}{\triangle \tilde{x}_{i-\frac{1}{2}}} [u_h]_{i-\frac{1}{2}} \right) [u_h]_{i-\frac{1}{2}}. \tag{2.13}$$

Summing (2.12) and (2.13), we have (2.11). $\qquad\square$

By the definition of $A(u)$ in (2.2), we can easily check $\frac{[A(u_h)]_{i+\frac{1}{2}}}{[u_h]_{j+\frac{1}{2}}} \geq 0$, which further leads to the L_2 stability of the LDG method on overlapping meshes. The proof is straightforward, so we omit it and only demonstrate the result as the following theorem.

Theorem 2.1. *The LDG method introduced in* (2.3) *and* (2.4) *is stable and*

$$\frac{1}{2} \frac{d}{dt} \|u_h\|^2 + \|p_h\|^2 \leq 0.$$

Remark 2.2. *For stability issue, we can take the penalty parameter $\alpha_{i-\frac{1}{2}} = 0$ for all $i = 1, 2, \cdots, N_x$. However, numerical experiments demonstrate that, this may lead to accuracy degeneration in some special cases.*

2.5   Error estimates

In this section, we proceed to the error estimates. For simplicity, we consider linear equations only, e.g. $a(u) = 1$ (A(u)=u), then (2.3) and (2.4) become

$$\int_{I_i} (u_h)_t v dx = -\int_{I_i} p_h v_x dx + \hat{p}_{i+\frac{1}{2}} v^-_{i+\frac{1}{2}} - \hat{p}_{i-\frac{1}{2}} v^+_{i-\frac{1}{2}}, \tag{2.14}$$

$$\int_{P_{i-\frac{1}{2}}} p_h w dx = -\int_{P_{i-\frac{1}{2}}} u_h w_x dx + u_h(\tilde{x}_i) w_i^- - u_h(\tilde{x}_{i-1}) w_{i-1}^+. \tag{2.15}$$

Moreover, H_u in (2.7) can also be written as

$$H_u(u_h, p_h, v) = -\sum_{i=1}^{N_x} \int_{I_i} p_h v_x dx - \sum_{i=1}^{N_x} \hat{p}_{i-\frac{1}{2}}[v]_{i-\frac{1}{2}}. \tag{2.16}$$

We denote the error as $e_u = u - u_h$ and $e_p = p - p_h$, then the error equations are

$$\int_{I_i} (e_u)_t v dx = -\int_{I_i} e_p v_x dx + \widehat{e_p}_{i+\frac{1}{2}} v_{i+\frac{1}{2}}^- - \widehat{e_p}_{i-\frac{1}{2}} v_{i-\frac{1}{2}}^+, \tag{2.17}$$

$$\int_{P_{i-\frac{1}{2}}} e_p w dx = -\int_{P_{i-\frac{1}{2}}} e_u w_x dx + e_u(\tilde{x}_i) w_i^- - e_u(\tilde{x}_{i-1}) w_{i-1}^+. \tag{2.18}$$

We first study some basic properties of the finite element space. Let us start with the classical inverse properties.

Lemma 2.2. *Assuming $u \in V_h$, then there exists a constant $C > 0$ independent of Δx and u such that for $\alpha \geq 1$*

$$\|\partial_x^\alpha u\|_{I_i} \leq C\Delta x_i^{-\alpha} \|u\|_{I_i}, \qquad \|u\|_{\Gamma_i} \leq C\Delta x_i^{-1/2} \|u\|_{I_i}.$$

Similarly, for any $u \in P_h$, there exists a constant $C > 0$ independent of Δx and u such that for $\alpha \geq 1$

$$\|\partial_x^\alpha u\|_{P_{i+\frac{1}{2}}} \leq C\Delta \tilde{x}_{i+\frac{1}{2}}^{-\alpha} \|u\|_{P_{i+\frac{1}{2}}} \qquad \|u\|_{\Gamma_{i+\frac{1}{2}}} \leq C\Delta \tilde{x}_{i+\frac{1}{2}}^{-1/2} \|u\|_{P_{i+\frac{1}{2}}}.$$

We also introduce the standard L^2 projection P_k^1 into V_h and P_k^2 into P_h by:

$$\int_{I_i} P_k^1 uv \ dx = \int_{I_i} uv \ dx \ , \ \forall v \in P^k(I_i), \quad \text{and} \quad \int_{P_{i+\frac{1}{2}}} P_k^2 uv \ dx = \int_{P_{i+\frac{1}{2}}} uv \ dx \ , \ \forall v \in P^k(P_{i+\frac{1}{2}}),$$

respectively. By the scaling argument, we obtain the following lemma [4].

Lemma 2.3. *Suppose the function $u(x) \in C^{k+1}(I_i)$, then there exists a positive constant C independent of Δx and u, such that*

$$\|u - P_k^1 u\|_{I_i} + \Delta x_i \|(u - P_k^1 u)_x\|_{I_i} + \Delta x_i^{\frac{1}{2}} \|u - P_k^1 u\|_{\infty, I_i} \leq C\Delta x_i^{k+1} \|u\|_{k+1, I_i}.$$

Moreover, if $u(x) \in C^{k+1}(P_{i+\frac{1}{2}})$, then there exists a positive constant C independent of Δx and u, such that

$$\|u - P_k^2 u\|_{P_{i+\frac{1}{2}}} + \Delta \tilde{x}_{i+\frac{1}{2}} \|(u - P_k^2 u)_x\|_{P_{i+\frac{1}{2}}} + \Delta \tilde{x}_{i+\frac{1}{2}}^{\frac{1}{2}} \|u - P_k^2 u\|_{\infty, P_{i+\frac{1}{2}}} \leq C\Delta \tilde{x}_{i+\frac{1}{2}}^{k+1} \|u\|_{k+1, P_{i+\frac{1}{2}}},$$

where $\|u\|_{k+1, I}$ is the standard H^{k+1}-norm over the interval I.

As the general treatment of the DG methods, we write the errors as

$$e_u = \eta_u - \xi_u \quad e_p = \eta_p - \xi_p,$$

where

$$\eta_u = u - P_k^1 u, \quad \xi_u = u_h - P_k^1 u, \quad \eta_p = p - P_k^2 p, \quad \xi_p = p_h - P_k^2 p.$$

With the above notations, we can rewrite the error equations (2.17) and (2.18) as

$$\int_{I_i} (\xi_u)_t v \, dx = \int_{I_i} e_p v_x \, dx - \widehat{e}_{p_{i+\frac{1}{2}}} v^-_{i+\frac{1}{2}} + \widehat{e}_{p_{i-\frac{1}{2}}} v^+_{i-\frac{1}{2}}, \tag{2.19}$$

$$\int_{P_{i+\frac{1}{2}}} \xi_p w \, dx = \int_{P_{i+\frac{1}{2}}} e_u w_x \, dx - e_u(\tilde{x}_{i+1}) w^-_{i+1} + e_u(\tilde{x}_i) w^+_i. \tag{2.20}$$

Now, we can state the main theorem.

Theorem 2.2. *Suppose the exact solution $u \in C^{k+2}(\Omega)$ and the finite element space is made up of piecewise polynomial of degree k in each cell. The numerical solutions satisfy (2.14) and (2.15). Then the error between the numerical and exact solutions satisfy*

$$\|e_u\| + \int_0^T \|e_p\| \, dt \leq C \Delta x^k,$$

where C is independent of Δx.

Proof. Sum up (2.19) and (2.20) with $v = \xi_u$ and $w = \xi_p$, and then sum up over i to obtain

$$
\begin{aligned}
\frac{1}{2}\frac{d}{dt}\|\xi_u\|^2 + \|\xi_p\|^2 &= \sum_{i=1}^{N_x} \int_{I_i} (\eta_p - \xi_p)(\xi_u)_x \, dx + \sum_{i=1}^{N_x} \left(\eta_{p_{i-\frac{1}{2}}} - \xi_{p_{i-\frac{1}{2}}} + \alpha_{i-\frac{1}{2}} \frac{[\eta_u - \xi_u]_{i-\frac{1}{2}}}{\Delta \tilde{x}_{i-\frac{1}{2}}} \right) [\xi_u]_{i-\frac{1}{2}} \\
&\quad + \sum_{i=1}^{N_x} \int_{P_{i-\frac{1}{2}}} (\eta_u - \xi_u)(\xi_p)_x \, dx + \sum_{i=1}^{N_x} (\eta_u - \xi_u)(\tilde{x}_i)[\xi_p]_i \\
&= \sum_{i=1}^{N_x} \int_{I_i} \eta_p(\xi_u)_x \, dx + \sum_{i=1}^{N_x} \left(\eta_{p_{i-\frac{1}{2}}} + \frac{\alpha_{i-\frac{1}{2}}[\eta_u]_{i-\frac{1}{2}}}{\Delta \tilde{x}_{i-\frac{1}{2}}} \right) [\xi_u]_{i-\frac{1}{2}} \\
&\quad + \sum_{i=1}^{N_x} \int_{P_{i-\frac{1}{2}}} \eta_u(\xi_p)_x \, dx + \sum_{i=1}^{N_x} \eta_u(\tilde{x}_i)[\xi_p]_i - H_u(\xi_u, \xi_p, \xi_u) - H_p(\xi_p, \xi_u) \\
&= R_1 + R_2 + R_3, \tag{2.21}
\end{aligned}
$$

where

$$R_1 = \sum_{i=1}^{N_x} \int_{I_i} \eta_p(\xi_u)_x \, dx + \sum_{i=1}^{N_x} \int_{P_{i-\frac{1}{2}}} \eta_u(\xi_p)_x \, dx,$$

$$R_2 = \sum_{i=1}^{N_x} \frac{\alpha_{i-\frac{1}{2}}[\eta_u]_{i-\frac{1}{2}}}{\Delta \tilde{x}_{i-\frac{1}{2}}} [\xi_u]_{i-\frac{1}{2}} - H_u(\xi_u, \xi_p, \xi_u) - H_p(\xi_p, \xi_u),$$

$$R_3 = \sum_{i=1}^{N_x} \eta_{p_{i-\frac{1}{2}}}[\xi_u]_{i-\frac{1}{2}} + \sum_{i=1}^{N_x} \eta_u(\tilde{x}_i)[\xi_p]_i.$$

9

Now we estimate R_i $i = 1, 2, 3$ term by term.

$$R_1 \leq \sum_{i=1}^{N_x} \left(\|\eta_p\|_{I_i} \|(\xi_u)_x\|_{I_i} + \|\eta_u\|_{P_{i-\frac{1}{2}}} \|(\xi_p)_x\|_{P_{i-\frac{1}{2}}} \right)$$

$$\leq \sum_{i=1}^{N_x} \left(\|\eta_p\|_{P_{i-\frac{1}{2}} \cup P_{i+\frac{1}{2}}} \|(\xi_u)_x\|_{I_i} + \|\eta_u\|_{I_{i-1} \cup I_i} \|(\xi_p)_x\|_{P_{i-\frac{1}{2}}} \right)$$

$$\leq C\Delta x^k \sum_{i=1}^{N_x} \left(\left(\|p\|_{k+1, P_{i-\frac{1}{2}}} + \|p\|_{k+1, P_{i+\frac{1}{2}}} \right) \|\xi_u\|_{I_i} + \left(\|u\|_{k+1, I_{i-1}} + \|u\|_{k+1, I_i} \right) \|\xi_p\|_{P_{i-\frac{1}{2}}} \right)$$

$$\leq C\Delta x^k \left(\|p\|_{k+1} \|\xi_u\| + \|u\|_{k+1} \|\xi_p\| \right). \tag{2.22}$$

where in the first step, we applied Cauchy-Schwarz inequality, in the second step we used Lemmas 2.2 and 2.3, and the last step follows from Cauchy-Schwarz inequality again. Applying Lemma 2.1, we obtain the estimate of R_2

$$R_2 \leq \sum_{i=1}^{N_x} \frac{\alpha_{i-\frac{1}{2}}}{\Delta \tilde{x}_{i-\frac{1}{2}}} \left([\eta_u]_{i-\frac{1}{2}} [\xi_u]_{i-\frac{1}{2}} - [\xi_u]_{i-\frac{1}{2}}^2 \right)$$

$$\leq C \sum_{i=1}^{N_x} \frac{\alpha_{i-\frac{1}{2}}}{\Delta x} [\eta_u]_{i-\frac{1}{2}}^2$$

$$\leq C \sum_{i=1}^{N_x} \alpha_{i-\frac{1}{2}} \Delta x^{2k} \left(\|u\|_{I_{i-1}}^2 + \|u\|_{I_i}^2 \right),$$

$$\leq C\Delta x^{2k}, \tag{2.23}$$

where in step 3, we applied Lemma 2.3, steps 2 and 4 follow from direct computation. Finally, we estimate R_3.

$$R_3 \leq \sum_{i=1}^{N_x} \|\eta_p\|_{\infty, P_{i-\frac{1}{2}}} (\|\xi_u\|_{\Gamma_{i-1}} + \|\xi_u\|_{\Gamma_i}) + \sum_{i=1}^{N_x} \|\eta_u\|_{\infty, I_i} (\|\xi_p\|_{\Gamma_{i-\frac{1}{2}}} + \|\xi_p\|_{\Gamma_{i+\frac{1}{2}}})$$

$$\leq C\Delta x^k \sum_{i=1}^{N_x} \left(\|p\|_{k+1, P_{i-\frac{1}{2}}} \left(\|\xi_u\|_{I_{i-1}} + \|\xi_u\|_{I_i} \right) + \|u\|_{k+1, I_i} \left(\|\xi_p\|_{P_{i-\frac{1}{2}}} + \|\xi_p\|_{P_{i+\frac{1}{2}}} \right) \right)$$

$$\leq C\Delta x^k \left(\|p\|_{k+1} \|\xi_u\| + \|u\|_{k+1} \|\xi_p\| \right), \tag{2.24}$$

where step 1 is straightforward, step 2 follows from Lemmas 2.2 and 2.3, and in the last step we applied the Cauchy-Schwarz inequality. Substitute (2.22)-(2.24) into (2.21) to obtain

$$\frac{1}{2} \frac{d}{dt} \|\xi_u\|^2 + \|\xi_p\|^2 \leq C\Delta x^{2k} + C\Delta x^k \left(\|\xi_u\| + \|\xi_p\| \right),$$

which further yields

$$\frac{d}{dt} \|\xi_u\|^2 + \|\xi_p\|^2 \leq C\Delta x^{2k} + \|\xi_u\|^2.$$

Finally, we can apply the Gronwall's inequality and complete the proof. $\square$

In Theorem 2.2, we only proved suboptimal convergence rate. Numerical experiments in Section 4 demonstrate that in some cases the order of accuracy is exactly k. To explain this point, we use uniform meshes and denote Δx as the mesh size for both the primitive and P-meshes. We consider the following steady state problem

$$p_x = 0, \quad p = u_x,$$

subject to periodic boundary condition. To make the problem well-posed, we need another assumption that $\int_\Omega u(x)\, dx = 0$. Then the numerical scheme turns out to be

$$0 = -\int_{I_i} p_h v_x dx + \hat{p}_{i+\frac{1}{2}} v^-_{i+\frac{1}{2}} - \hat{p}_{i-\frac{1}{2}} v^+_{i-\frac{1}{2}}, \tag{2.25}$$

$$\int_{P_{i+\frac{1}{2}}} p_h w dx = -\int_{P_{i+\frac{1}{2}}} u_h w_x dx + u_h(\tilde{x}_{i+1}) w^-_{i+1} - u_h(\tilde{x}_i) w^+_i. \tag{2.26}$$

We take $\xi_0 = 0$ in (2.1), i.e. the dual mesh is constructed by using the midpoint of the primitive mesh and $\alpha_{i-\frac{1}{2}} = 0$ for all $i = 1, 2, \cdots, N_x$. Moreover, we use P^1 polynomials and assume

$$u_h(x) = u_i^0 + u_i^1 L_i(x), \quad x \in I_i,$$

$$p_h(x) = p_{i+\frac{1}{2}}^0 + p_{i+\frac{1}{2}}^1 L_{i+\frac{1}{2}}(x), \quad x \in P_{i+\frac{1}{2}},$$

where $L_i(x)$ and $L_{i+\frac{1}{2}}(x)$ are the scaled Legendre polynomial in cell I_i and $P_{i+\frac{1}{2}}$, respectively. Take $v(x) = 1$ and $v(x) = L_i(x)$ in (2.25), respectively, to obtain

$$0 = p_{i+\frac{1}{2}}^0 - p_{i-\frac{1}{2}}^0, \quad 0 = -p_{i-\frac{1}{2}}^0 - \frac{1}{2}p_{i-\frac{1}{2}}^1 - p_{i+\frac{1}{2}}^0 + \frac{1}{2}p_{i+\frac{1}{2}}^1 + p_{i+\frac{1}{2}}^0 + p_{i-\frac{1}{2}}^0, \tag{2.27}$$

which further yield

$$p_{i+\frac{1}{2}}^0 = p_{i-\frac{1}{2}}^0, \quad p_{i+\frac{1}{2}}^1 = p_{i-\frac{1}{2}}^1.$$

Take $w(x) = 1$ and $w(x) = L_{i+\frac{1}{2}}(x)$ in (2.26), respectively to obtain

$$\Delta x p_{i+\frac{1}{2}}^0 = u_{i+1}^0 - u_i^0, \quad \frac{\Delta x}{3} p_{i+\frac{1}{2}}^1 = -\frac{1}{2}u_i^1 + \frac{1}{2}u_{i+1}^1. \tag{2.28}$$

Clearly, (2.27) and (2.28) are not uniquely solvable, one nontrivial solution could be $u_h(x) = L_i(x)$ in I_i and $p_h = 0$. It is easy to check that $\int_\Omega u_h\, dx = 0$. Therefore, it is impossible to construct the elliptic projection to prove the optimal convergence rate. Numerical experiments demonstrate that each of the following case would result in optimal convergence rates.

1. Use even order polynomials, i.e. $k = 0, 2, \cdots$;

2. Take $\xi_0 = 0$ with $\alpha \neq 0$;

3. Take $\xi_0 \neq 0$;

Besides the above, for convection-diffusion equations, we can always obtain optimal rates of convergence even though we take $\xi_0 = \alpha = 0$.

3 Numerical scheme for two-dimensional case

In this section, we will construct the LDG scheme on overlapping meshes in two space dimensions and study the following PDE over the domain $\Omega = [0, 1] \times [0, 1]$,

$$\begin{cases} u_t = (a(u)p)_x + (b(u)q)_y, \\ p = A(u)_x, \\ q = B(u)_y \end{cases} \tag{3.1}$$

subject to periodic boundary conditions, where $A(u) = \int^u a(t)dt$ and $B(u) = \int^u b(t)dt$.

We first define the primitive mesh for the primary variable u which is a regular rectangular decomposition of Ω. Let $0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N_x+\frac{1}{2}} = 1$ and $0 = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N_y+\frac{1}{2}} = 1$ be the grid points in x and y directions, respectively, and denote the i, j-th cell as

$$I_{ij} = I_i \times J_j, \quad i = 1, \cdots, N_x, \ j = 1, \cdots, N_y.$$

where $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and $J_j = [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. Moreover, we denote

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \quad x_i = \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2}, \quad \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \quad y_j = \frac{y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}}}{2}.$$

and

$$\Delta x = \max_i \Delta x_i, \quad \Delta y = \max_j \Delta y_j, \quad h = \max\{\Delta x, \Delta y\}$$

We also move each cell horizontally to obtain the P-mesh: $P_{i+\frac{1}{2},j} = [\tilde{x}_i, \tilde{x}_{i+1}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, where

$$\tilde{x}_i = x_i + \frac{\Delta x_i}{2}\xi_0, \qquad \xi_0 \in [-1, 1], \qquad i = 1, 2, \cdots, N_x, \tag{3.2}$$

with $\tilde{x}_0 = \tilde{x}_{N_x} - 1$. Similarly, we can define the Q-mesh: $Q_{i,j+\frac{1}{2}} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [\tilde{y}_j, \tilde{y}_{j+1}]$, where

$$\tilde{y}_j = y_j + \frac{\Delta y_j}{2}\eta_0, \qquad \eta_0 \in [-1, 1], \qquad j = 1, 2, \cdots, N_y, \tag{3.3}$$

with $\tilde{y}_0 = \tilde{y}_{N_y} - 1$. The P-mesh and Q-mesh are used to solve the auxiliary variables p and q, respectively. Similar to the problem in one space dimension, we can also define $P_{\frac{1}{2},j} = ([0, \tilde{x}_1] \cup [\tilde{x}_{N_x}, 1]) \times J_j$ and $Q_{j,\frac{1}{2}} = I_i \times ([0, \tilde{y}_1] \cup [\tilde{y}_{N_y}, 1])$.

We define the finite element spaces to be

$$V_h := \{u_h : u_h|_{I_{ij}} \in P^k(I_{ij}), \ i = 1, \cdots, N_x, \ j = 1, \cdots, N_y\},$$

$$P_h := \{p_h : p_h|_{P_{i+\frac{1}{2},j}} \in P^k(P_{i+\frac{1}{2},j}), \ i = 1, \cdots, N_x, \ j = 1, \cdots, N_y\},$$

$$Q_h := \{q_h : q_h|_{Q_{i,j+\frac{1}{2}}} \in P^k(Q_{i,j+\frac{1}{2}}), \ i = 1, \cdots, N_x, \ j = 1, \cdots, N_y\}.$$

Given $u \in V_h$, we denote $u^+_{i-\frac{1}{2},j}$, $u^-_{i+\frac{1}{2},j}$, $u^+_{i,j-\frac{1}{2}}$, $u^-_{i,j+\frac{1}{2}}$ to be the traces of u on the four edges of I_{ij}, respectively. Likewise for the traces of $P_{i+\frac{1}{2},j}$ along the vertical edges and those of $Q_{i,j+\frac{1}{2}}$ along the horizontal edges. Moreover, we use $[u] = u^+ - u^-$ and $\{u\} = \frac{1}{2}(u^+ + u^-)$ as the jump and average of u at the cell interfaces, respectively.

Now, we can introduce the LDG method on overlapping meshes: find $(u_h, p_h, q_h) \in V_h \times P_h \times Q_h$, such that for any test functions $(v, w, z) \in V_h \times P_h \times Q_h$, we have

$$\int_{I_{ij}} (u_h)_t v \, dx dy = - \int_{I_{ij}} a(u_h) p_h v_x \, dx dy + \int_{J_j} \hat{a}_{i+\frac{1}{2},j} \hat{p}_{i+\frac{1}{2},j} v^-_{i+\frac{1}{2},j} \, dy - \int_{J_j} \hat{a}_{i-\frac{1}{2},j} \hat{p}_{i-\frac{1}{2},j} v^+_{i-\frac{1}{2},j} \, dy,$$
$$- \int_{I_{ij}} b(u_h) q_h v_y \, dx dy + \int_{I_i} \hat{b}_{i,j+\frac{1}{2}} \hat{q}_{i,j+\frac{1}{2}} v^-_{i,j+\frac{1}{2}} \, dx - \int_{I_i} \hat{b}_{i,j-\frac{1}{2}} \hat{q}_{i,j-\frac{1}{2}} v^+_{i,j-\frac{1}{2}} \, dx,$$

(3.4)

$$\int_{P_{i+\frac{1}{2},j}} p_h w \, dx dy = - \int_{P_{i+\frac{1}{2},j}} A(u_h) w_x \, dx dy + \int_{J_j} A(u_h(\tilde{x}_{i+1})) w^-_{i+1} \, dy - \int_{J_j} A(u_h(\tilde{x}_i)) w^+_i \, dy, \quad (3.5)$$

$$\int_{Q_{i,j+\frac{1}{2}}} p_h z \, dx dy = - \int_{Q_{i,j+\frac{1}{2}}} B(u_h) z_y \, dx dy + \int_{I_i} B(u_h(\tilde{y}_{j+1})) z^-_{j+1} \, dx - \int_{I_i} B(u_h(\tilde{y}_j)) z^+_j \, dx. \quad (3.6)$$

The numerical flux $\hat{a}$ along the edge $x = x_{i+\frac{1}{2}}$ is taken as

$$\hat{a}_{i+\frac{1}{2},j} = \frac{[A(u_h)]_{i+\frac{1}{2},j}}{[u_h]_{i+\frac{1}{2},j}}$$

Similarly, the numerical flux $\hat{b}$ along the edge $y = y_{j+\frac{1}{2}}$ is taken as

$$\hat{b}_{i,j+\frac{1}{2}} = \frac{[B(u_h)]_{i,j+\frac{1}{2}}}{[u_h]_{i,j+\frac{1}{2}}},$$

where $[s]_{i+\frac{1}{2},j} := s^+_{i+\frac{1}{2},j} - s^-_{i+\frac{1}{2},j}$ denotes the jump of a function s across the cell boundary $\{x_{i+\frac{1}{2}}\} \times J_j$. Likewise for $[s]_{i,j+\frac{1}{2}}$. Moreover, we choose

$$\hat{p}_{i+\frac{1}{2},j} = p_h(x_{i+\frac{1}{2}}, y) + \frac{\alpha_{i+\frac{1}{2},j}}{\Delta \tilde{x}_{i+\frac{1}{2},j}} [u_h]_{i+\frac{1}{2},j}, \quad \hat{q}_{i,j+\frac{1}{2}} = q_h(x, y_{j+\frac{1}{2}}) + \frac{\alpha_{i,j+\frac{1}{2}}}{\Delta \tilde{x}_{i,j+\frac{1}{2}}} [u_h]_{i,j+\frac{1}{2}}$$

Following the same analyses for problems in one space dimension, we can obtain the stability analysis and error estimates. Therefore, we will skip the proof and only demonstrate the results in the following two theorems.

Theorem 3.1. *The LDG methods introduced in* (3.4)-(3.6) *is stable and*

$$\frac{1}{2} \frac{d}{dt} \|u_h\|^2 + \|p_h\|^2 + \|q_h\|^2 \leq 0.$$

Theorem 3.2. *Suppose the exact solution for linear parabolic equation* (3.1) *with* $a(u) = b(u) = 1$ *satisfies* $u \in C^{k+1}(\Omega)$ *and the finite element space is made up of piecewise polynomial of degree* k *in each cell. The numerical solutions satisfy* (3.4)-(3.6). *Then the error between the numerical and exact solutions satisfy*

$$\|u - u_h\| + \int_0^T \|p - p_h\| + \|q - q_h\| \, dt \leq Ch^k,$$

where C *is independent of* h.

4 Numerical examples

In this section, we will use numerical experiments to demonstrate the stability and the accuracy of the new LDG method on overlapping meshes. In all the numerical experiments, we use piecewise polynomials of degree $k = 1, 2, 3, 4$. If not otherwise stated we consider third-order SSP Runge-Kutta time discretization [14] with $\Delta t = 0.1\Delta x^2$ if $k = 1, 2$ and $\Delta t = 0.01\Delta x^2$ if $k = 3, 4$ to reduce the time error, and take the final time $T = 1$. Moreover, the random mesh is generated by randomly and independently perturbing each node in a uniform mesh by up to 20%.

Example 4.1. *We solve the following heat equation in one space dimension*

$$\begin{cases} u_t = u_{xx}, \\ u(x,0) = \sin(x), \end{cases} \quad x \in [0, 2\pi]. \tag{4.1}$$

Clearly, the exact solution is

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

We consider uniform mesh and take $\xi_0 = 0$ in (2.1), i.e. the dual mesh is generated by using the midpoint of the primitive mesh. We compute the error between the numerical and exact solutions and the results under the L^2- and L^∞-norms are given in Table 4.1. From the table, we can only

Table 4.1: Example 4.1: midpoint, uniform mesh.

k	number of cells	no penalty				$\alpha = 0.2$			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
1	10	3.36E-02	–	7.43E-02	–	8.05E-03	–	1.76E-02	–
	20	1.68E-02	1.00	3.73E-02	0.99	1.45E-03	2.47	2.84E-03	2.63
	40	8.42E-03	1.00	1.87E-02	1.00	3.00E-04	2.27	5.46E-04	2.38
	80	4.21E-03	1.00	9.34E-03	1.00	7.02E-05	2.10	1.40E-04	1.96
	160	2.10E-03	1.00	4.67E-03	1.00	1.72E-05	2.03	3.54E-05	1.99
2	10	3.05E-04	–	8.61E-04	–	2.56E-04	–	6.86E-04	–
	20	3.85E-05	2.99	1.11E-04	2.95	3.21E-05	3.00	8.99E-05	2.93
	40	4.83E-06	3.00	1.40E-05	2.99	4.02E-06	3.00	1.14E-05	2.98
	80	6.04E-07	3.00	1.75E-06	3.00	5.03E-07	3.00	1.43E-06	3.00
	160	7.55E-08	3.00	2.19E-07	3.00	6.28E-08	3.00	1.79E-07	3.00
3	10	8.53E-06	–	1.50E-05	–	8.61E-06	–	1.42E-05	–
	20	5.69E-07	3.91	1.12E-06	3.75	5.29E-07	4.02	8.80E-07	4.01
	40	4.30E-08	3.72	9.91E-08	3.49	3.29E-08	4.01	5.51E-08	4.00
	80	4.05E-09	3.41	1.08E-08	3.20	2.06E-09	4.00	3.45E-09	4.00
	160	4.55E-10	3.15	1.29E-09	3.06	1.28E-10	4.00	2.15E-10	4.00
4	10	6.55E-07	–	1.84E-06	–	2.28E-07	–	5.80E-07	–
	20	1.84E-08	5.16	5.19E-08	5.15	7.04E-09	5.02	1.95E-08	4.90
	40	5.60E-10	5.04	1.58E-09	5.03	2.19E-10	5.01	6.20E-10	4.97
	80	1.74E-11	5.01	4.92E-11	5.01	6.82E-12	5.00	1.95E-11	4.99
	160	5.43E-13	5.00	1.54E-12	5.00	2.13E-13	5.00	6.12E-13	4.99

observe suboptimal accuracy if k is an odd number and the penalty parameter $\alpha = 0$. To obtain optimal convergence rates, we can choose $\alpha \neq 0$ or use even order polynomials. We repeat the

Table 4.2: midpoint, random mesh (20%) .

k	number of cells	no penalty				$\alpha = 0.2$			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
	10	3.99E-02	–	1.19E-01	–	8.30E-03	–	2.22E-02	–
	20	1.82E-02	1.19	4.56E-02	1.45	1.51E-03	2.61	3.33E-03	2.90
1	40	9.13E-03	1.02	2.52E-02	0.88	3.50E-04	2.56	1.06E-03	2.00
	80	4.55E-03	1.08	1.50E-02	0.80	7.93E-05	1.95	2.49E-04	1.90
	160	2.28E-03	0.98	7.55E-03	0.98	1.95E-05	2.21	6.04E-05	2.23
	10	3.25E-04	–	1.34E-03	–	3.27E-04	—	1.05E-03	–
	20	4.66E-05	2.60	1.69E-04	2.77	3.74E-05	2.99	1.51E-04	2.66
2	40	5.64E-06	3.01	2.33E-05	2.82	5.04E-06	3.14	2.25E-05	2.99
	80	7.37E-07	3.00	3.20E-06	2.93	6.02E-07	3.22	2.24E-06	3.49
	160	9.50E-08	3.23	4.17E-07	3.21	7.92E-08	2.96	3.55E-07	2.69
	10	1.44E-05	–	5.43E-05	–	1.70E-05	–	4.17E-05	–
	20	1.22E-06	3.83	4.64E-06	3.81	7.59E-07	3.96	1.95E-06	3.90
3	40	1.05E-07	3.41	4.22E-07	3.34	5.58E-08	4.23	1.65E-07	4.01
	80	1.40E-08	2.97	6.75E-08	2.70	3.08E-09	4.22	1.07E-08	3.98
	160	1.57E-09	3.38	7.61E-09	3.37	1.93E-10	4.06	7.56E-10	3.89
	10	1.04E-06	–	3.75E-06	–	3.00E-07	—	1.20E-06	–
	20	3.67E-08	5.41	2.30E-07	4.52	1.23E-08	5.20	5.41E-08	5.05
4	40	1.12E-09	5.53	8.64E-09	5.21	3.24E-10	5.40	1.33E-09	5.49
	80	3.84E-11	5.01	2.99E-10	4.99	1.01E-11	4.84	6.59E-11	4.20
	160	1.16E-12	4.68	6.21E-12	5.18	3.30E-13	5.07	2.09E-12	5.10

same example with random meshes with results given in Table 4.2, we can observe exactly the same convergence rates discussed above.

Another possible way to recover optimal convergence rates would be taking $\xi_0 \neq 0$. In Tables 4.3 and 4.4, we take $\alpha = 0$, and choose $\xi_0 = 0.05$ which is close to 0 and $\xi_0 = \sqrt{3}/3$ which is away from 0. We can clearly observe optimal convergence rates. Moreover, the errors for $\xi_0 = \sqrt{3}/3$ is less than those for $\xi_0 = 0.05$.

Example 4.2. *We solve the following convection-diffusion equation*

$$\begin{cases} u_t + u_x = u_{xx}, \\ u(x,0) = \sin(x), \end{cases} \quad x \in [0, 2\pi]. \tag{4.2}$$

Clearly, the exact solution is

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

We consider random meshes only and use upwind fluxes for the convection term. We take $\xi_0 = \alpha = 0$ to test the accuracy and the results are given in Table 4.5. From the table, we can observe optimal convergence rates.

Example 4.3. *We solve the heat equation in two space dimensions:*

$$\begin{cases} u_t = u_{xx} + u_{yy}, \\ u(x,0) = \sin(x)\cos(y), \end{cases} \quad (x,y) \in [0, 2\pi] \times [0, 2\pi]. \tag{4.3}$$

Table 4.3: $\xi_0 = 0.05$ and $\xi_0 = \sqrt{3}/3$, uniform mesh.

k	number of cells	$\xi_0 = 0.05$				$\xi_0 = \sqrt{3}/3$			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
1	10	3.13E-02	–	7.06E-02	–	3.82E-03	–	9.09E-03	–
	20	1.29E-02	1.28	2.96E-02	1.25	9.56E-04	2.00	2.21E-03	2.04
	40	3.75E-03	1.78	8.72E-03	1.77	2.39E-04	2.00	5.53E-04	2.00
	80	8.46E-04	2.15	2.00E-03	2.12	5.98E-05	2.00	1.38E-04	2.00
	160	1.98E-04	2.10	4.72E-04	2.08	1.49E-05	2.00	3.46E-05	2.00
2	10	3.08E-04	–	9.12E-04	–	3.09E-04	–	1.03E-03	–
	20	3.88E-05	2.99	1.17E-04	2.96	3.76E-05	3.04	1.26E-04	3.03
	40	4.85E-06	3.00	1.47E-05	2.99	4.67E-06	3.01	1.57E-05	3.01
	80	6.07E-07	3.00	1.84E-06	3.00	5.83E-07	3.00	1.96E-06	3.00
	160	7.59E-08	3.00	2.30E-07	3.00	7.28E-08	3.00	2.44E-07	3.00
3	10	2.12E-05	–	7.02E-05	–	6.01E-05	–	2.02E-04	–
	20	3.36E-06	2.66	1.06E-05	2.72	3.85E-06	3.97	1.33E-05	3.93
	40	2.73E-07	3.62	8.54E-07	3.64	2.21E-07	4.12	7.73E-07	4.10
	80	1.74E-08	3.98	5.42E-08	3.98	1.33E-08	4.06	4.66E-08	4.05
	160	1.09E-09	3.99	3.40E-09	3.99	8.20E-10	4.02	2.88E-09	4.02
4	10	6.71E-07	–	1.96E-06	–	4.75E-07	–	1.45E-06	–
	20	1.87E-08	5.17	5.48E-08	5.16	1.48E-08	5.00	4.71E-08	4.94
	40	5.69E-10	5.04	1.67E-09	5.04	4.63E-20	5.00	1.49E-09	4.99
	80	1.77E-11	5.01	5.19E-11	5.01	1.45E-12	5.00	4.65E-11	5.00
	160	5.51E-13	5.00	1.62E-12	5.00	4.53E-13	5.00	1.46E-12	5.00

Table 4.4: $\xi_0 = 0.05$ and $\xi_0 = \sqrt{3}/3$, random mesh (20%).

k	number of cells	$\xi_0 = 0.05$				$\xi_0 = \sqrt{3}/3$			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
1	10	3.82E-02	–	1.06E-01	–	4.24E-03	–	1.41E-02	–
	20	1.38E-02	1.43	3.72E-02	1.47	1.08E-03	2.23	3.85E-03	2.11
	40	3.97E-03	1.80	1.15E-02	1.69	2.81E-04	2.23	1.30E-03	1.80
	80	9.28E-04	2.21	2.75E-03	2.18	7.05E-05	1.92	2.66E-04	2.20
	160	2.20E-04	2.12	7.76E-04	1.86	1.71E-05	2.12	7.70E-05	1.86
2	10	4.91E-04	–	1.61E-03	–	4.37E-04	–	2.05E-03	–
	20	5.71E-05	3.26	2.67E-04	2.72	5.29E-05	2.79	2.62E-04	2.72
	40	6.14E-06	3.19	3.05E-05	3.10	6.00E-06	3.97	3.53E-05	3.65
	80	7.45E-07	2.86	3.15E-06	3.08	8.31E-07	2.98	5.55E-06	2.78
	160	9.47E-08	3.03	4.89E-07	2.74	1.14E-07	2.62	9.08E-07	2.38
3	10	2.40E-05	–	9.34E-05	–	7.52E-05	–	3.19E-04	–
	20	5.36E-06	2.39	2.84E-05	1.89	4.67E-06	4.10	1.58E-05	4.43
	40	3.57E-07	3.93	1.76E-06	4.04	2.78E-07	4.56	1.11E-06	4.29
	80	2.79E-08	4.02	1.79E-07	3.60	1.82E-08	4.00	8.01E-08	3.86
	160	1.74E-09	3.90	1.32E-08	3.67	1.14E-09	4.07	6.02E-09	3.80
4	10	9.42E-07	–	3.06E-06	–	9.23E-07	–	4.27E-06	–
	20	4.14E-08	5.72	2.59E-07	4.52	3.26E-08	5.60	2.26E-07	4.92
	40	1.17E-09	5.37	9.91E-09	4.92	7.79E-10	5.19	5.85E-09	5.09
	80	3.47E-11	5.22	2.72E-10	5.34	2.58E-11	5.08	1.54E-10	5.42
	160	1.39E-12	4.59	9.81E-12	4.73	1.10E-12	4.73	6.05E-12	4.84

Table 4.5: Example 4.2: $\xi_0 = \alpha = 0$. random mesh.

number of cells	L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
			P^1				P^2	
10	8.00E-03	–	1.71E-02	–	2.78E-04	–	8.53E-04	–
20	2.08E-03	1.92	5.19E-03	1.70	4.25E-05	2.94	1.45E-04	2.78
40	5.46E-04	1.78	1.38E-03	1.76	5.73E-06	3.22	2.60E-05	2.77
80	1.39E-04	2.20	3.60E-04	2.16	7.29E-07	2.96	3.72E-06	2.79
160	3.52E-05	1.93	9.10E-05	1.93	9.65E-08	2.84	4.82E-07	2.87
			P^3				P^4	
10	1.49E-05	–	5.33E-05	–	2.67E-07	–	9.82E-07	–
20	9.62E-07	4.51	3.95E-06	4.29	1.35E-08	5.15	7.46E-08	4.45
40	6.47E-08	3.89	2.43E-07	4.02	6.08E-10	4.50	3.23E-09	4.56
80	3.77E-09	4.10	1.56E-08	3.96	2.14E-11	4.67	1.43E-10	4.34
160	2.36E-10	4.04	1.06E-09	3.92	9.57E-13	5.01	5.05E-12	5.40

Clearly, the exact solution is

$$u(x,t) = e^{-2t}\sin(x)\cos(y).$$

We first test the example with $\xi_0 = 0$ in (3.2), $\eta_0 = 0$ in (3.3) and $\alpha = \beta = 0$. The results are given in Table 4.6. From the table, we can only observe $(k+1)$-th order convergence rates for $k = 3$. To recover the optimal convergence rates, we choose other penalty parameters and the results are

Table 4.6: Example 4.3: $\xi_0 = \eta_0 = \alpha = \beta = 0$, different meshes.

k	number of cells	uniform mesh				random mesh			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
	10	2.20E-02	–	5.08E-02	–	2.25E-02	–	6.25E-02	–
	20	1.10E-02	1.01	2.46E-02	1.05	1.13E-02	0.99	3.44E-02	0.86
1	40	5.48E-03	1.00	1.22E-02	1.01	5.67E-03	1.00	1.75E-02	0.98
	80	2.74E-03	1.00	6.09E-03	1.00	2.83E-03	1.00	9.42E-03	0.89
	160	1.37E-03	1.00	3.04E-03	1.00	1.41E-03	1.00	4.72E-03	1.00
	10	4.18E-03	–	2.01E-02	–	4.25E-03	–	2.44E-02	–
	20	1.07E-03	1.97	5.24E-03	1.94	1.10E-03	1.96	7.27E-03	1.75
2	40	2.69E-04	1.99	1.32E-03	1.98	2.78E-04	1.98	1.96E-03	1.89
	80	6.74E-05	2.00	3.32E-04	2.00	6.94E-05	2.00	6.56E-04	1.58
	160	1.69E-05	2.00	8.31E-05	2.00	1.73E-05	2.00	1.54E-04	2.09
	10	6.50E-05	–	3.59E-04	–	8.32E-05	–	7.01E-04	–
	20	3.95E-06	4.04	2.25E-05	3.99	5.27E-06	3.98	5.45E-05	3.69
3	40	2.46E-07	4.01	1.42E-06	3.98	3.31E-07	3.99	3.67E-06	3.89
	80	1.54E-08	4.00	8.93E-08	4.00	2.05E-08	4.01	2.13E-07	4.11
	160	9.81E-10	3.97	5.62E-09	3.99	1.39E-09	3.89	1.53E-08	3.79
	10	1.47E-05	–	1.08E-04	–	1.63E-05	–	1.71E-04	–
	20	9.29E-07	3.98	7.11E-06	3.93	1.05E-06	3.96	1.26E-05	3.76
4	40	5.82E-08	4.00	4.46E-07	3.99	6.16E-08	4.09	7.57E-07	4.06
	80	3.64E-09	4.00	2.79E-08	4.00	3.15E-09	4.29	5.71E-08	3.73
	160	2.27E-10	4.00	1.75E-09	4.00	1.05E-10	4.90	1.75E-09	5.03

listed in Table 4.7. We use random meshes only and can observe that if both penalty parameters

are nonzero, we can obtain optimal rates of convergence. However, if only one of the penalty parameters is nonzero, we still have the accuracy degeneration if piecewise odd order polynomials are applied. Moreover, following Example 4.1, we also choose random meshes and take different

Table 4.7: Example 4.3: $\xi_0 = \eta_0 = 0$, different α and β, random mesh.

k	number of cells	$\alpha = 0,\ \beta = 0.2$				$\alpha = 0.2,\ \beta = 0.2$			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
1	10	1.86E-02	–	6.48E-02	–	1.07E-02	–	3.95E-02	–
	20	8.78E-03	1.08	3.09E-02	1.07	2.32E-03	2.21	9.68E-03	2.03
	40	4.43E-03	0.99	1.89E-02	0.71	5.40E-04	2.10	2.85E-03	1.77
	80	2.22E-03	0.99	9.76E-03	0.95	1.31E-04	2.05	7.09E-04	2.01
	160	1.11E-03	1.00	4.47E-03	1.13	3.25E-05	2.01	1.89E-04	1.91
2	10	1.79E-03	–	1.07E-02	–	1.21E-03	–	7.55E-03	–
	20	1.62E-04	3.47	1.10E-03	3.28	1.23E-04	3.29	9.67E-04	2.96
	40	1.62E-05	3.32	1.23E-04	3.15	1.41E-05	3.13	1.13E-04	3.10
	80	1.78E-06	3.19	1.45E-05	3.09	1.69E-06	3.07	1.38E-05	3.03
	160	2.14E-07	3.05	1.82E-06	3.00	2.09E-07	3.02	1.74E-06	2.99
3	10	8.02E-05	–	6.45E-04	–	7.36E-05	–	5.64E-04	–
	20	5.72E-06	3.81	5.76E-05	3.49	4.70E-06	3.97	4.92E-05	3.52
	40	4.92E-07	3.54	3.76E-06	3.94	2.95E-07	4.00	3.22E-06	3.94
	80	4.91E-08	3.33	3.32E-07	3.50	1.79E-08	4.04	1.76E-07	4.19
	160	5.85E-09	3.07	3.81E-08	3.12	1.11E-09	4.01	1.32E-08	3.74
4	10	5.38E-06	–	4.54E-05	–	4.66E-06	–	3.82E-05	–
	20	1.67E-07	5.01	1.96E-06	4.53	1.56E-07	4.90	1.76E-06	4.44
	40	5.14E-09	5.02	6.15E-08	4.99	4.94E-09	4.99	5.59E-08	4.98
	80	1.52E-10	5.08	1.87E-09	5.04	1.48E-10	5.06	1.73E-09	5.01
	160	4.73E-12	5.01	5.99E-11	4.96	4.60E-12	5.01	5.61E-11	4.95

values of ξ_0 and η_0. The results are provided in Table 4.8. We can observe that, if both of them are nonzero, the optimal convergence rates can be recovered. However, if one of them is zero, (e.g. $\xi_0 = 0$), the optimal rates cannot be obtained for $k = 1$.

Example 4.4. *We solve the convection-diffusion equation in two space dimensions*

$$\begin{cases} u_t + u_x + u_y = u_{xx} + u_{yy}, \\ u(x, y, 0) = \sin(x)\cos(y), \end{cases} \quad (x, y) \in [0, 2\pi] \times [0, 2\pi]. \tag{4.4}$$

Clearly, the exact solution is

$$u(x, y, t) = e^{-2t} \sin(x - t)\cos(y - t).$$

We consider random meshes only and use upwind fluxes for the convection term. We take $\xi_0 = \eta_0 = \alpha = \beta = 0$ to test the accuracy and the results are given in Table 4.9. From the table, we can observe optimal convergence rates.

5 Conclusion

In this paper, we have introduced a new LDG method on overlapping meshes. The scheme is stable but the order of accuracy may not be optimal. We demonstrated a potential reason for the accuracy

Table 4.8: Example 4.3: different ξ_0 and η_0, $\alpha = \beta = 0$, random mesh.

k	number of cells	$\xi_0 = 0$, $\eta_0 = 0.05$				$\xi_0 = 0.05$, $\eta_0 = 0.05$			
		L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
1	10	2.21E-02	–	6.73E-02	–	2.18E-02	–	6.63E-02	–
	20	1.07E-02	1.04	3.46E-02	0.96	1.01E-02	1.11	3.76E-02	0.82
	40	4.78E-03	1.17	1.84E-02	0.91	3.74E-03	1.43	1.71E-02	1.14
	80	2.11E-03	1.18	9.03E-03	1.02	9.24E-04	2.02	4.99E-03	1.78
	160	1.01E-03	1.06	4.31E-03	1.07	2.15E-04	2.10	1.26E-03	1.98
2	10	4.06E-03	–	2.48E-02	–	3.95E-03	–	2.46E-02	–
	20	9.46E-04	2.10	6.57E-03	1.92	8.18E-04	2.27	5.76E-03	2.09
	40	1.59E-04	2.57	1.20E-03	2.45	1.12E-04	2.87	8.30E-04	2.80
	80	1.76E-05	3.17	1.60E-04	2.91	1.12E-05	3.32	9.04E-05	3.20
	160	1.79E-06	3.30	1.68E-05	3.25	1.21E-06	3.22	1.00E-05	3.17
3	10	8.45E-05	–	7.48E-04	–	8.50E-05	–	7.06E-04	–
	20	5.70E-06	3.89	5.64E-05	3.73	6.02E-06	3.82	5.96E-05	3.57
	40	4.25E-07	3.75	3.81E-06	3.89	4.88E-07	3.62	4.53E-06	3.72
	80	2.59E-08	4.03	2.27E-07	4.07	3.11E-08	3.97	2.90E-07	3.97
	160	1.65E-09	3.98	1.59E-08	3.84	1.93E-09	4.01	2.03E-08	3.84
4	10	1.54E-05	–	1.71E-04	–	1.49E-05	–	1.66E-04	–
	20	8.58E-07	4.17	1.08E-05	3.99	7.10E-07	4.39	8.90E-06	4.22
	40	3.05E-08	4.81	4.25E-07	4.67	2.11E-08	5.07	2.81E-07	4.99
	80	7.83E-10	5.28	1.30E-08	5.03	5.34E-10	5.30	7.31E-09	5.26
	160	2.06E-11	5.25	3.46E-10	5.23	1.50E-11	5.16	2.08E-10	5.13

Table 4.9: Example 4.4: $\xi_0 = \eta_0 = \alpha = \beta = 0$. random mesh.

number of cells	L^2 norm	order	L^∞ norm	order	L^2 norm	order	L^∞ norm	order
	P^1				P^2			
10	1.01E-02	–	4.10E-02	–	1.04E-03	–	6.67E-03	–
20	2.69E-03	1.90	1.21E-02	1.76	1.43E-04	2.86	1.01E-03	2.73
40	6.86E-04	1.97	3.23E-03	1.90	1.87E-05	2.94	1.31E-04	2.94
80	1.72E-04	2.00	7.54E-04	2.10	2.38E-06	2.98	1.70E-05	2.95
160	4.30E-05	2.00	1.98E-04	1.93	3.00E-07	2.99	2.18E-06	2.96
	P^3				P^4			
10	7.36E-05	–	6.30E-04	–	4.66E-06	–	4.77E-05	–
20	4.85E-06	3.92	4.98E-05	3.66	1.65E-07	4.82	1.88E-06	4.67
40	3.10E-07	3.97	3.72E-06	3.74	5.35E-09	4.94	6.86E-08	4.78
80	1.90E-08	4.02	2.25E-07	4.05	1.64E-10	5.03	2.05E-09	5.06
160	1.20E-09	3.99	1.42E-08	3.98	5.15E-12	4.99	6.98E-11	4.88

degeneration and provided several alternatives to recover the optimal convergence rates.

References

[1] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, Journal of Computational Physics, 131 (1997), 267-279.

[2] Z. Chen, H. Huang and J. Yan, *Third order Maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes*, Journal of Computational Physics, 308 (2016), 198-217.

[3] E. Chung and C.S. Lee *A staggered discontinuous Galerkin method for the convection–diffusion equation*, Journal of Numerical Mathematics, 20 (2012), 1-31.

[4] P.G. Ciarlet. *Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.

[5] B. Cockburn, S. Hou and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case*, Mathematics of Computation, 54 (1990), 545-581.

[6] B. Cockburn, S.-Y. Lin and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems*, Journal of Computational Physics, 84 (1989), 90-113.

[7] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework*, Mathematics of Computation, 52 (1989), 411-435.

[8] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, Journal of Computational Physics, 141 (1998), 199-224.

[9] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time dependent convection-diffusion systems*, SIAM Journal on Numerical Analysis, 35 (1998), 2440-2463.

[10] J. Douglas, Jr., R.E. Ewing and M.F. Wheeler, *A time-discretization procedure for a mixed finite element approximation of miscible displacement in porous media*, R.A.I.R.O. Analyse numérique, 17 (1983), 249-256.

[11] J. Douglas, Jr., R.E. Ewing and M.F. Wheeler, *The approximation of the pressure by a mixed method in the simulation of miscible displacement*, R.A.I.R.O. Analyse numérique, 17 (1983), 17-33.

[12] J. Du and Y. Yang, *Maximum-principle-preserving third-order LDG method for convection-diffusion equations on overlapping meshes*, submitted.

[13] I.M. Gelfand, *Some questions of analysis and differential equations*, American Mathematical Society Translations, 26 (1963), 201-219.

[14] S. Gottlieb, C.-W. Shu and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), 89-112.

[15] H. Guo, F. Yu and Y. Yang, *Local discontinuous Galerkin method for incompressible miscible displacement problem in porous media*, Journal of Scientific Computing, 71 (2017), 615-633.

[16] H. Guo and Y. Yang, *Bound-preserving discontinuous Galerkin method for compressible miscible displacement problem in porous media*, SIAM Journal on Scientific Computing, 39 (2017), A1969-A1990.

[17] L. Guo and Y. Yang, *Positivity-preserving high-order local discontinuous Galerkin method for parabolic equations with blow-up solutions*, Journal of Computational Physics, 289 (2015), 181-195.

[18] A.E. Hurd and D.H. Sattinger, *Questions of existence and uniqueness for hyperbolic equations with discontinuous coefficients*, Transactions of the American Mathematical Society, 132 (1968), 159-174.

[19] E. F. Keller and L. A. Segel, *Initiation on slime mold aggregation viewed as instability*, Journal of Theoretical Biology, 26 (1970), 399-415.

[20] X. Li, C.-W. Shu and Y. Yang, *Local discontinuous Galerkin method for the Keller-Segel chemotaxis model*, Journal of Scientific Computing, 73 (2017), 943-967.

[21] Y. Liu, C.-W. Shu, E. Tadmor and M. Zhang, *Central local discontinuous Galerkin methods on overlapping cells for diffusion equations*, ESAIM: Mathematical Modelling and Numerical Analysis (M2AN), 45 (2011), 1009-1032.

[22] C. Patlak, *Random walk with persistence and external bias*, The bulletin of mathematical biophysics, 15 (1953), 311338.

[23] T. Qin, C.-W. Shu and Y. Yang, *Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics*, Journal of Computational Physics, 315 (2016), 323-347.

[24] W.H. Reed and T.R. Hill, *Triangular mesh methods for the Neutron transport equation*, Los Alamos Scientific Laboratory Report LA-UR-73-479, Los Alamos, NM, 1973.

[25] H. Wang, C.-W. Shu, and Q. Zhang, *Stability and error estimates of local discontinuous Galerkin methods with implicit-explicit time-marching for advection-diffusion problems*, SIAM Journal on Numerical Analysis, 53 (2015), 206-227.

[26] H. Wang, C.-W. Shu and Q. Zhang, *Stability analysis and error estimates of local discontinuous Galerkin methods with implicit-explicit time-marching for nonlinear convection-diffusion problems*, Applied Mathematics and Computation, 272 (2016), 237-258.

[27] H. Wang, S. Wang, Q. Zhang and C.-W. Shu, *Local discontinuous Galerkin methods with implicit-explicit time marching for multi-dimensional convectiondiffusion problems*, ESAIM: M2AN, 50 (2016), 1083-1105.

[28] T. Xiong, J.-M. Qiu and Z. Xu, *High order maximum-principle-preserving discontinuous Galerkin method for convection-diffusion equations*, SIAM Journal on Scientific Computing, 37 (2015), A583-A608.

[29] Y. Yang, D. Wei and C.-W. Shu, *Discontinuous Galerkin method for Krause's consensus models and pressureless Euler equations*, Journal of Computational Physics, 252 (2013), 109-127.

[30] F. Yu, H. Guo, N. Chuenjarern and Y. Yang, *Conservative local discontinuous Galerkin method for compressible miscible displacements in porous media*, Journal of Scientific Computing, 73 (2017), 1249-1275.

[31] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, Journal of Computational Physics, 229 (2010), 3091-3120.

[32] X. Zhang and C.-W. Shu, *On positivity preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, Journal of Computational Physics, 229 (2010), 8918-8934.

[33] Y. Zhang, X. Zhang and C.-W. Shu, *Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection-diffusion equations on triangular meshes*, Journal of Computational Physics, 234 (2013), 295-316.

[34] X. Zhao, Y. Yang and C. Seyler, *A positivity-preserving semi-implicit discontinuous Galerkin scheme for solving extended magnetohydrodynamics equations*, Journal of Computational Physics, 278 (2014), 400-415.