Journal of Computational Physics 307 (2016) 574–592

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Alternating evolution discontinuous Galerkin methods for convection–diffusion equations

Hailiang Liu *, Michael Pollack

Iowa State University, Mathematics Department, Ames, IA 50011, United States

ARTICLE INFO

Article history:
Received 22 April 2015
Received in revised form 9 September 2015
Accepted 8 December 2015
Available online 14 December 2015

Keywords:
Alternating evolution
Convection–diffusion equations
Discontinuous Galerkin

ABSTRACT

In this work, we propose a high order alternating evolution discontinuous Galerkin (AEDG) method to solve convection–diffusion equations. The difficulties related to numerical fluxes of DG methods for diffusion problems have been a major issue of investigation in the literature. The AEDG scheme presented here is based on an alternating evolution system for convection–diffusion equations, and therefore no numerical fluxes are needed in the scheme formulation. Moreover, the method is shown to be consistent, conservative and stable. Numerical experiments are provided to show the goodness of the proposed method.

1. Introduction

Nonlinear PDEs arise in many applications such as differential geometry, fluid mechanics, image processing, meteorology, mesh generation, optimal control, and optimal mass transport. As a result, the solution of each of these application problems critically depends on the solution of their underlying PDEs. Efficient and reliable high order numerical methods for computing the physically relevant solutions of these nonlinear PDEs are highly desired and challenging; see [10].

A grid-based discontinuous Galerkin (DG) method, called the alternating evolution discontinuous Galerkin (AEDG) method, has been recently developed in [18] for the Hamilton–Jacobi equation – a class of first order fully nonlinear PDEs. Our interest is to extend the AEDG methodology to solve second order PDEs. In this paper, we describe the designing principle of the AEDG scheme through the following model equation,

$$\partial_t \phi + \nabla_x \cdot f(\phi) = \Delta_x a(\phi), \quad \phi(x,0) = \phi_0(x), \quad x \in \mathbb{R}^d, \quad t > 0, \tag{1.1}$$

where $f(\phi)$ is a vector flux, and a is a smooth function satisfying $a' \geq 0$. These equations are also of great practical importance since they and variations model a variety of physical phenomena that appear in fluid mechanics, astrophysics, groundwater flow, traffic flow, semiconductor device simulation, and magneto-hydrodynamics, among many others.

The alternating evolution framework was originally introduced in [14], in which the hyperbolic conservation law, i.e. (1.1) with $a = 0$, is connected to an alternating evolution (AE) system

$$\partial_t u + \nabla_x \cdot f(v) = \frac{1}{\epsilon}(v - u), \tag{1.2a}$$

$$\partial_t v + \nabla_x \cdot f(u) = \frac{1}{\epsilon}(u - v). \tag{1.2b}$$

* Corresponding author.
E-mail addresses: hliu@iastate.edu (H. Liu), mpollack@iastate.edu (M. Pollack).

http://dx.doi.org/10.1016/j.jcp.2015.12.017
0021-9991/

The solution of this system when taking the same initial data $\phi_0(x)$ for two components will recover the entropy solution of the corresponding conservation law, independent of ϵ. The idea using the AE system as a numerical device has been elaborated further in [3,25,17,18]. More precisely, local AE finite volume schemes by Saran and Liu in [3,25] for hyperbolic conservation laws, a high resolution AE finite difference scheme by Liu, Pollack, and Saran in [17] for Hamilton–Jacobi equations, and a high order AEDG scheme by Liu and Pollack in [18] for Hamilton–Jacobi equations. This work generalizes these earlier developments to second order PDEs for the first time.

A distinct advantage of the existing AE schemes is that no numerical fluxes are needed in the scheme formulation. Instead, the communication with neighboring solution representatives ϕ^{SN} is through an AE formulation

$$\partial_t \phi + \nabla_x \cdot f(\phi^{SN}) = \Delta a(\phi^{SN}) + \frac{1}{\epsilon}(\phi^{SN} - \phi).$$

In this formulation, whenever a spatial derivative is evaluated, the neighboring polynomials will be used in the computation, since the evaluation of the solution at the interface is in the continuous region of the neighboring solution representatives. This reformulation is then sampled by each representative over alternating grids. The semi-discrete scheme for (1.1) in one dimensional case thus has the following form:

$$\int_{I_j} (\partial_t \Phi_j + \partial_x f(\Phi_j^{SN}) - \partial_x^2 a(\Phi_j^{SN}))\eta dx = \left(-[f(\Phi_j^{SN})]\eta + [\partial_x a(\Phi_j^{SN})]\eta - [a(\Phi_j^{SN})]\eta_x\right)\Big|_{x=x_j} + \frac{1}{\epsilon}\left(\int_{I_j} \Phi_j^{SN}\eta dx - \int_{I_j} \Phi_j \eta dx\right),$$

where x_j is the grid point in the cell I_j, in which numerical solution is denoted by Φ_j; Φ_j^{SN} are sampled from neighboring polynomials $\Phi_{j\pm1}$. The initial condition is taken as the L^2 projection of $\phi_0(x)$ into the relevant numerical solution space.

The model we consider is in conservative form, so one may also apply an existing numerical flux-based DG method. In fact, the discussion on the difficulties related to numerical fluxes of DG methods for diffusion problems has attracted a lot of attention in the literature. For example, we mention the method by Bassi and Rebay [6] for compressible Navier–Stokes equations, its generalization called the local discontinuous Galerkin (LDG) methods by Cockburn and Shu [8], the method by Baumann, Oden et al. [21,5], the recovery type DG method by van Leer and Nomura [16] (see also [24]), the DDG method by Liu and Yan [19,20], and the hybridizable discontinuous Galerkin (HDG) method of Cockburn et al. [7]. Also in the 1970s, Galerkin methods for elliptic and parabolic problems using discontinuous finite elements, called the interior penalty (IP) methods, were independently introduced and studied; see, e.g., [1,4,26]. We refer to [2] for a unified analysis of DG methods for elliptic problems and background references for the IP methods.

In this article we have two objectives: (i) to present the construction of the AEDG method; (ii) to verify the stability of the method and illustrate its numerical performance. The construction for (i) is based on sampling the AE system of the convection–diffusion equation by a polynomial representative near each grid point, and it is carried out by allowing the neighboring polynomials to overlap. It is similar to the central DG methods [15,22,23] in the sense that whenever a spatial derivative is evaluated, the neighboring polynomials (or the other representatives in the central DG scheme) are used. The closely related methods are the two versions of the central local discontinuous Galerkin (LDG) methods for diffusion problems recently introduced in [23], where the equation is rewritten as a system of first-order equations and two polynomial representatives are used on two sets of overlapping meshes. Therefore, in the one dimensional case, four unknowns are needed to be resolved on each computational cell. The present AEDG method involves only one unknown near each grid point, independent of the spatial dimension.

The stability analysis for (ii) is subtle. Even for the linear diffusion case, the stability property is not obvious from the scheme formulation. We overcome this difficulty by a special regrouping of mixed terms combined with the use of some inverse inequalities. The difference between this work, and the work in [23] is more delicate than the difference between some direct DG methods such as those in [9,19,20] and the LDG method in [8]. Roughly speaking, in direct DG methods for diffusion, stability can be enforced by tuning 'penalty' parameters in the numerical flux, but there is no such mechanism to use in the AEDG formulation.

Regarding the numerical performance, the advantage of the AEDG scheme compared to the central LDG schemes in [23] is more pronounced in the multi-dimensional case. The AE framework results in an easy formulation of the scheme where the solution spatial derivatives are computed using the closest neighboring polynomials.

For the model equation (1.1), the AEDG method provides an alternative algorithm to other existing numerical flux-based direct DG methods, see e.g., [16,9,19,20], with comparable computational costs. Another advantage of the AEDG method is its potential for solving second order fully nonlinear PDEs, for which most numerical flux-based DG methods are no longer applicable in a straightforward manner.

The article is organized as follows: in Section 2, we formulate the AEDG method for the one-dimensional convection–diffusion equations. In Section 3, properties of scheme conservation and stability are analyzed for the linear convection–diffusion equation and, as a result, the bound for the parameter ϵ is justified. In Section 4, the scheme is extended to the

multi-dimensional case. Numerical results for both the one and two-dimensional problems presented in Section 5 illustrate the goodness of the proposed method. Concluding remarks are given in Section 6.

2. Alternating evolution DG methods

Our numerical schemes consist of a semi-discrete formulation based on sampling of the AE system on alternating grids and a fully discrete version by using an appropriate Runge–Kutta solver.

2.1. The AEDG formulation

We develop an AE discontinuous Galerkin (AEDG) method for the one-dimensional convection diffusion equation

$$\partial_t \phi + \partial_x f(\phi) = \partial_x^2 (a(\phi)), \tag{2.1}$$

subject to initial data $\phi_0(x)$. We divide the spatial domain to form a uniform grid with grid points $\{x_j\}$. We denote $I_j = (x_{j-1}, x_{j+1})$ for $j = 2, \ldots, N-1$, $I_1 = (x_1, x_2)$ and $I_N = (x_{N-1}, x_N)$, with mesh as $\Delta x = x_{j+1} - x_j$. Here the uniform grids are taken only for easy presentation, nonuniform grids work as well, though for nonuniform grids x_j is no longer a center point in I_j. Centered at each grid $\{x_j\}$, the numerical approximation is a polynomial $\Phi|_{I_j} = \Phi_j(x) \in P^k$, where P^k denotes a linear space of all polynomials of degree at most k:

$$P^k := \{p \mid p(x)|_{I_j} = \sum_{0 \le i \le k} a_i (x - x_j)^i, \quad a_i \in \mathbb{R}\}.$$

We denote $v(x^{\pm}) = \lim_{\epsilon \to 0\pm} v(x + \epsilon)$, and $v_j^{\pm} = v(x_j^{\pm})$. The jump at x_j is $[v]_j = v(x_j^+) - v(x_j^-)$. The AE system for (2.1) is

$$\begin{aligned} \partial_t u + \partial_x f(v) - \partial_x^2 (a(v)) &= \frac{1}{\epsilon}(v - u), \\ \partial_t v + \partial_x f(u) - \partial_x^2 (a(u)) &= \frac{1}{\epsilon}(u - v). \end{aligned}$$

Integrating this AE formulation over I_j against the test function $\eta \in P^k(I_j)$, the semi-discrete AEDG scheme is to find $\Phi|_{I_j} \in P^k$ such that

$$\begin{aligned} \int_{I_j} (\partial_t \Phi_j + \partial_x f(\Phi_j^{SN}) - \partial_x^2 a(\Phi_j^{SN}))\eta dx = &\left(-[f(\Phi_j^{SN})]\eta + [\partial_x a(\Phi_j^{SN})]\eta - [a(\Phi_j^{SN})]\eta_x\right)\Big|_{x=x_j} \\ &+ \frac{1}{\epsilon}\left(\int_{I_j} \Phi_j^{SN}\eta dx - \int_{I_j} \Phi_j \eta dx\right), \end{aligned} \tag{2.2}$$

where Φ_j^{SN} are sampled from neighboring polynomials $\Phi_{j\pm1}$ in the following way:

$$\begin{aligned} \int_{I_j} \partial_x^2 a(\Phi_j^{SN})\eta dx &= \int_{x_{j-1}}^{x_j} \partial_x^2 a(\Phi_{j-1})\eta dx + \int_{x_j}^{x_{j+1}} \partial_x^2 a(\Phi_{j+1})\eta dx, \\ \int_{I_j} \partial_x f(\Phi_j^{SN})\eta dx &= \int_{x_{j-1}}^{x_j} \partial_x f(\Phi_{j-1})\eta dx + \int_{x_j}^{x_{j+1}} \partial_x f(\Phi_{j+1})\eta dx, \\ \int_{I_j} \Phi_j^{SN}\eta dx &= \int_{x_{j-1}}^{x_j} \Phi_{j-1}\eta dx + \int_{x_j}^{x_{j+1}} \Phi_{j+1}\eta dx, \\ [f(\Phi_j^{SN})]\Big|_{x=x_j} &= f(\Phi_{j+1}(x_j^+)) - f(\Phi_{j-1}(x_j^-)), \\ [\partial_x a(\Phi_j^{SN})]\Big|_{x=x_j} &= \partial_x a(\Phi_{j+1}(x_j^+)) - \partial_x a(\Phi_{j-1}(x_j^-)), \\ [a(\Phi_j^{SN})]\Big|_{x=x_j} &= a(\Phi_{j+1}(x_j^+)) - a(\Phi_{j-1}(x_j^-)). \end{aligned}$$

Note that the terms at x_j are needed since

$$\Phi_j^{SN} = \begin{cases} \Phi_{j-1}(x) & x_{j-2} < x < x_j \\ \Phi_{j+1}(x) & x_j < x < x_{j+2} \end{cases}$$

are not required to be continuous at x_j, hence $\lim_{\delta\to 0}(\int_{x_j-\delta}^{x_j+\delta} \partial_x f(\Phi_j^{SN}) - \partial_x^2 a(\Phi_j^{SN}))\eta dx$ does not vanish, instead it can be approximated by

$$[f(\Phi_j^{SN})]\eta - [\partial_x a(\Phi_j^{SN})]\eta + [a(\Phi_j^{SN})]\eta_x.$$

To update each grid-centered polynomial element Φ, we write the compact form of the semi-discrete scheme

$$\frac{d}{dt}\int_{I_j} \Phi_j \eta dx = L[\Phi_j; \Phi_j^{SN}, \eta], \tag{2.3}$$

where

$$L[\Phi_j; \Phi_{j\pm1}, \eta] = \frac{1}{\epsilon}\int_{I_j} (\Phi_j^{SN} - \Phi_j)\eta dx - \int_{I_j} \partial_x f(\Phi_j^{SN})\eta dx + \int_{I_j} \partial_x^2 a(\Phi_j^{SN})\eta dx + \left(-[f(\Phi_j^{SN})]\eta + [\partial_x a(\Phi_j^{SN})]\eta - [a(\Phi_j^{SN})]\eta_x\right)\Big|_{x=x_j}. \tag{2.4}$$

For cells near the boundary we treat with care. For a computational domain $[a, b]$ with $x_1 = a$, $x_N = b$ and $\Delta x = (b-a)/(N-1)$, the two equations near boundary may be given as

$$\int_{x_1}^{x_2} (\partial_t \Phi_1 + \partial_x f(\Phi_2) - \partial_x^2 a(\Phi_2))\eta dx = \frac{1}{2}\left(-[f(\Phi_1^{SN})]\eta + [\partial_x a(\Phi_1^{SN})]\eta - [a(\Phi_1^{SN})]\eta_x\right)\Big|_{x=x_1} + \frac{1}{\epsilon}\int_{x_1}^{x_2} (\Phi_2 - \Phi_1)\eta dx, \tag{2.5}$$

$$\int_{x_{N-1}}^{x_N} (\partial_t \Phi_N + \partial_x f(\Phi_{N-1}) - \partial_x^2 a(\Phi_{N-1})\eta dx = \frac{1}{2}\left(-[f(\Phi_N^{SN})]\eta + [\partial_x a(\Phi_N^{SN})]\eta - [a(\Phi_N^{SN})]\eta_x\right)\Big|_{x=x_N} + \frac{1}{\epsilon}\int_{x_{N-1}}^{x_N} (\Phi_{N-1} - \Phi_N)\eta dx. \tag{2.6}$$

At the left boundary, $\eta \in P^k(x_1, x_2)$, and on the right boundary, $\eta \in P^k(x_{N-1}, x_N)$, so that the length of the interval is Δx instead of $2\Delta x$.

For the Dirichlet boundary condition, $\phi(a,t) = g_1(t)$ and $\phi(b,t) = g_2(t)$, we modify (2.5) by setting

$$\begin{aligned} [f(\Phi_1^{SN})] &= f(\Phi_2(x_1^+)) - f(g_1(t)), \\ [a(\Phi_1^{SN})] &= a(\Phi_2(x_1^+)) - a(g_1(t)), \\ [\partial_x a(\Phi_1^{SN})] &= 0. \end{aligned}$$

We also modify (2.6) by setting

$$\begin{aligned} [f(\Phi_N^{SN})] &= f(g_2(t)) - f(\Phi_{N-1}(x_N^-, t)), \\ [a(\Phi_N^{SN})] &= a(g_2(t)) - a(\Phi_{N-1}(x_N^-, t)), \\ [\partial_x a(\Phi_N^{SN})] &= 0. \end{aligned}$$

In the case of periodic boundary conditions, $\Phi_N(x)$ is regarded to be identical to $\Phi_1(x)$, so $[a(\Phi)]$ can be computed as $a(\Phi_2(x_1^+)) - a(\Phi_{N-1}(x_N^-))$ at $x = x_1, x_N$, so can other jump terms.

Remark 2.1. In the case of diffusion of the form $\partial_x(\nu(\phi)\partial_x\phi)$, only a slight modification is needed. For example, $[\partial_x a(\Phi)] = [\nu(\Phi)\partial_x\Phi]$, and $[a(\Phi)]$ may be replaced by $\{\nu(\Phi)\}[\Phi]$, where $\{\nu(\Phi)\}$ denotes the average of $\nu(\Phi)$ crossing the center points x_j.

The fully discrete scheme follows from applying an appropriate Runge–Kutta solver to (2.3). We summarize the algorithm as follows.

Algorithm:

1. Initialization: in any cell I_j, compute the initial data by the local L^2-projection
$$\int_{I_j} (\Phi^0 - \phi_0)\eta dx = 0, \quad \eta \in P^k(I_j). \tag{2.7}$$
2. Alternating evaluation: take polynomials $\Phi^n_{j\pm1}(x) = \Phi^n|_{I_{j\pm1}}$, and then sample in I_j to get $L[\Phi^n_j; \Phi^n_{j\pm1}, \eta]$ as defined in (2.4).
3. Evolution: obtain Φ^{n+1} from Φ^n by some Runge–Kutta type procedure to solve the ODE system (2.3).

In the fully AEDG schemes, ϵ is chosen such that the stability condition,
$$\Delta t < \epsilon \le Q\,(\Delta x)^2 \tag{2.8}$$
is satisfied. The choice of Q depends on the order of the scheme and the diffusion. We suggest the following formula for linear diffusion in which $a(\phi)$ is replaced with $\beta\phi$ from (2.1):
$$\Delta t < \epsilon \le \frac{(\Delta x)^2}{\beta(k+1)^2\left(3(k+1)^2-1\right)}. \tag{2.9}$$
This numerical range for ϵ is justified by both Lemma 2.2 for the lower bound, and the stability analysis in Section 3 for the upper bound, though (2.9) is less restrictive than that obtained from the stability analysis. In the numerical results for $k=1$ and $k=2$, all ϵ's used fall within this range. Note that the scheme is in general inconsistent with the diffusion for the P^0 case, due to the vanishing of all $a(\Phi^{SN}_j)$ related terms in the scheme formulation. For the linear diffusion, ϵ can be taken as the upper bound in (2.9), i.e., $\frac{(\Delta x)^2}{2\beta}$, so to recover the scheme consistency.

To make this point clear, we discuss in detail why further restriction on ϵ is needed when taking P^0 polynomials. In such a case, we can rewrite (2.2) as
$$\frac{d}{dt}\Phi_j = -\frac{1}{2\Delta x}\left(f(\Phi^n_{j+1}) - f(\Phi^n_{j-1})\right) + \frac{1}{\epsilon}\left(\frac{\Phi^n_{j-1}+\Phi^n_{j+1}}{2} - \Phi^n_j\right). \tag{2.10}$$
Note that for $k=0$ case, the diffusion related terms in (2.2) vanish. A consistency check using the Taylor expansion shows that for the smooth solution $\phi(x,t)$,
$$\frac{1}{\epsilon}\left(\frac{\phi(x_{j-1},t^n)+\phi(x_{j+1},t^n)}{2} - \phi(x_j,t^n)\right) = \frac{(\Delta x)^2}{2\epsilon}(\partial_x^2\phi(x_j,t^n) + O(\Delta x)^2).$$
Hence it is necessary to choose $\epsilon = \frac{(\Delta x)^2}{2\beta}$ so that the semi-discrete scheme (2.10) is consistent with
$$\partial_t\phi + \partial_x f(\phi) = \beta\partial_x^2\phi. \tag{2.11}$$
Furthermore, for the forward Euler time discretization of (2.10),
$$\frac{\Phi^{n+1}_j - \Phi^n_j}{\Delta t} = -\frac{1}{2\Delta x}\left(f(\Phi^n_{j+1}) - f(\Phi^n_{j-1})\right) + \frac{1}{\epsilon}\left(\frac{\Phi^n_{j-1}+\Phi^n_{j+1}}{2} - \Phi^n_j\right), \tag{2.12}$$
we have the following statements.

Lemma 2.2. *(2.12) is consistent with (2.11) if any only if*
$$\epsilon = \frac{(\Delta x)^2}{2\beta}.$$
For small Δx such that $\max|f'| \le \frac{2\beta}{\Delta x}$, (2.12) preserves the maximum principle, i.e.,
$$\min_j \Phi^n_j \le \Phi^{n+1}_j \le \max_j \Phi^n_j \tag{2.13}$$
if and only if
$$\frac{\Delta t}{\epsilon} \le 1. \tag{2.14}$$

Proof. Equation (2.12) can be rewritten as

$$\begin{aligned}\Phi_j^{n+1} &= \left(1-\frac{\Delta t}{\epsilon}\right)\Phi_j^n + \frac{\Delta t}{2\epsilon}\left(\Phi_{j+1}^n + \Phi_{j-1}^n\right) - \frac{\lambda}{2}\left(f(\Phi_{j+1}^n) - f(\Phi_{j-1}^n)\right),\\ &= \left(1-\frac{\Delta t}{\epsilon}\right)\Phi_j^n + \frac{1}{2}\left(\frac{\Delta t}{\epsilon} - \lambda f_j'\right)\Phi_{j+1}^n + \frac{1}{2}\left(\frac{\Delta t}{\epsilon} + \lambda f_j'\right)\Phi_{j-1}^n,\end{aligned}$$

where $\lambda = \Delta t/\Delta x$, $f_j' = \partial_u f(\cdot)$ with $(\cdot)$ being an immediate value between Φ_{j-1}^n and Φ_{j+1}^n. For small Δx as specified, we know that $\max|f'| \le \frac{\Delta x}{\epsilon} = \frac{2\beta}{\Delta x}$. Thus, the discrete maximum principle (2.13) holds if (2.14) is met. We can also show that (2.14) is also necessary for (2.13) to hold. Consider the case $\Phi_{j_0}^0 = 1$ for some j_0 and $\Phi_j^0 = 0$ for $j \neq j_0$. The discrete maximum principle (2.13) requires $\Phi_j^1 \ge \min \Phi_j^0 = 0$. This when combined with the fact that $\Phi_{j_0}^1 = (1 - \frac{\Delta t}{\epsilon})$ leads to (2.14). □

2.2. Spatial discretization

We now present the AEDG spatial discretization with polynomial elements of degree k.
Let $\{\phi_l(\xi)\}_{l=1}^{k+1}$ be the basis in the master domain $[-1,1]$, then in each cell I_j we can express

$$\Phi_j(x) = \sum_{l=1}^{k+1} a_j^l \phi_l(\xi) =: \phi^\top(\xi) a_j, \quad \xi = \frac{x - x_j}{\Delta x},$$

then

$$\Phi_{j\pm1}(x) = \sum_{l=1}^{k+1} a_{j\pm1}^l \phi_l(\xi \mp 1) = \phi^\top(\xi \mp 1) a_{j\pm1},$$

and

$$\partial_\xi \Phi_{j\pm1}(x) = \frac{1}{\Delta x}\sum_{l=2}^{k+1} a_{j\pm1}^l (l-1)\phi_l(\xi \mp 1) = \partial_\xi \phi^\top(\xi \mp 1) a_{j\pm1}.$$

A simple calculation of (2.2) gives

$$M\dot{a}_j = -\frac{1}{\epsilon} M a_j + \frac{1}{\epsilon} L_j, \tag{2.15}$$

where

$$M = \Delta x \int_{-1}^{1} \phi(\xi)\phi^\top(\xi) d\xi, \quad L_j = C_{j-1} + D_{j+1},$$

with

$$\begin{aligned}C_{j-1} &= \Delta x \int_{-1}^{0} \phi(\xi)\left[\phi^\top(\xi+1)a_{j-1} - \frac{\epsilon}{h}\partial_\xi f(\phi^\top(\xi+1)a_{j-1}) + \frac{\epsilon}{h}\partial_\xi^2 a(\phi^\top(\xi+1)a_{j-1})\right] d\xi\\ &\quad + \epsilon f(\phi^\top(1)a_{j-1})\phi(0) - \frac{\epsilon}{h}\partial_\xi a(\phi^\top(1)a_{j-1})\phi(0) + \frac{\epsilon}{h} a(\phi^\top(1)a_{j-1})\partial_\xi\phi(0),\\ D_{j+1} &= \Delta x \int_{0}^{1} \phi(\xi)\left[\phi^\top(\xi-1)a_{j+1} - \frac{\epsilon}{h}\partial_\xi f(\phi^\top(\xi-1)a_{j+1}) + \frac{\epsilon}{h}\partial_\xi^2 a(\phi^\top(\xi-1)a_{j+1})\right] d\xi\\ &\quad - \epsilon f(\phi^\top(-1)a_{j+1})\phi(0) + \frac{\epsilon}{h}\partial_\xi a(\phi^\top(-1)a_{j+1})\phi(0) - \frac{\epsilon}{h} a(\phi^\top(-1)a_{j+1})\partial_\xi\phi(0).\end{aligned}$$

In the above evaluation, nonlinear terms are handled by using some quadrature of high accuracy.

2.3. Temporal discretization

We now turn to time discretization of (2.15). Let $\{t^n\}$, $n = 0, 1, \ldots, K$ be a uniform partition of the time interval $[0, T]$. Let $\Phi^0 = P\phi_0$ be the piecewise L^2 projection of $\phi_0(x)$ defined in (2.7), and $A = [a_1, \cdots, a_N]^\top$, then

$$\dot{A} = G(A).$$

We use the third order explicit SSP Runge–Kutta method [11] for time discretization. In details, let $A^{(0)} = A^n$ be the solution at time level n,

$$\begin{aligned} A^{(1)} &= A^{(0)} + \Delta t G(A^{(0)}), \\ A^{(2)} &= \frac{3}{4}A^{(0)} + \frac{1}{4}\left(A^{(1)} + \Delta t G(A^{(1)})\right), \\ A^{n+1} &= \frac{1}{3}A^{(0)} + \frac{2}{3}\left(A^{(2)} + \Delta t G(A^{(2)})\right). \end{aligned} \tag{2.16}$$

3. Properties of the AEDG scheme

With the addition of jump terms from two neighboring polynomials, the AEDG scheme (2.2) is still consistent with (2.1). In fact, if we consider the interval (x_j, x_{j+1}), two overlapping polynomials Φ_j and Φ_{j+1} are numerical approximations to the same solution $\phi(x,t)|_{(x_j,x_{j+1})}$. Assume $\phi(x,t)$ is smooth. We replace the numerical solution in (2.2) with the exact solution $\phi(x,t)$, we see that the left hand side becomes

$$\int_{I_j} (\partial_t \phi(x,t) + \partial_x f(\phi(x,t)) - \partial_x^2 a(\phi(x,t)))\eta dx = 0,$$

and the right hand side can be evaluated as

$$(-[f(\phi(x,t))]\eta + [\partial_x a(\phi(x,t))]\eta - [a(\phi(x,t))]\eta_x)\Big|_{x=x_j} + \frac{1}{\epsilon}\left(\int_{I_j} \phi(x,t)\eta dx - \int_{I_j} \phi(x,t)\eta dx\right) = 0.$$

Hence the AEDG scheme is consistent with (2.1).

We next show that the semi-discrete AEDG scheme is also conservative and stable.

Theorem 3.1. *Let Φ be computed from the AEDG scheme for the convection–diffusion equation with periodic boundary conditions. Then*

$$\frac{d}{dt}\left(\sum_{j=1}^{N-1}\int_{x_j}^{x_{j+1}} \frac{\Phi_{j+1} + \Phi_j}{2} dx\right) = 0.$$

Proof. Set $J(w) = f(w) - a(w)_x$. Taking $\eta = 1$, $\Phi_{2i} =: u$ in (2.3) for $j = 2i$, and $\Phi_{2i+1} =: v$ in (2.3) for $j = 2i+1$, respectively, we have

$$\begin{aligned} \frac{d}{dt}\int_{I_{2i}} u dx + \int_{I_{2i}} J_x(v) dx + [J(v)]|_{x_{2i}} &= \frac{1}{\epsilon}\left(\int_{I_{2i}} v dx - \int_{I_{2i}} u dx\right), \\ \frac{d}{dt}\int_{I_{2i+1}} v dx + \int_{I_{2i+1}} J_x(u) dx + [J(u)]|_{x_{2i+1}} &= \frac{1}{\epsilon}\left(\int_{I_{2i+1}} u dx - \int_{I_{2i+1}} v dx\right). \end{aligned}$$

Note that

$$[a,b] = \cup_{i=1}^{N/2-1} I_{2i} + [x_{N-1}, x_N] = [x_1, x_2] + \cup_{i=1}^{N/2-1} I_{2i+1},$$

we sum over all index and use (2.5)–(2.6) to obtain

$$\frac{d}{dt}\int_a^b u dx + \left(\sum_{i=1}^{N/2-1}\int_{I_{2i}} + \int_{x_{N-1}}^{x_N}\right) J_x(v)dx + \sum_{i=1}^{N/2-1}[J(v)]|_{x_{2i}} + \frac{1}{2}[J(v)]|_{x_N} = \frac{1}{\epsilon}\int_a^b (v-u)dx,$$

$$\frac{d}{dt}\int_a^b v dx + \left(\int_{x_1}^{x_2} + \sum_{i=1}^{N/2-1}\int_{I_{2i+1}}\right) J_x(u)dx + \frac{1}{2}[J(u)]|_{x_1} + \sum_{i=1}^{N/2-1}[J(u)]|_{x_{2i+1}} = \frac{1}{\epsilon}\int_a^b (u-v)dx.$$

Adding these two relations up, we obtain

$$\frac{d}{dt}\int_a^b (u+v)dx + J_1 + J_2 = 0,$$

where the terms in J_1 and J_2 are expressed as follows:

$$\begin{aligned} J_1 &= \left(\sum_{i=1}^{N/2}\int_{x_{2i-1}}^{x_{2i}} + \sum_{i=1}^{N/2-1}\int_{x_{2i}}^{x_{2i+1}}\right)[J_x(v) + J_x(u)]dx \\ &= -\left(\sum_{i=1}^{N/2-1}[J(u)]|_{x_{2i+1}} + \sum_{i=1}^{N/2-1}[J(v)]|_{x_{2i}}\right) - J(\Phi_1)(x_1) + J(\Phi_{N-1})(x_N-) \\ &\quad - J(\Phi_2)(x_1^+) + J(\Phi_N)(x_N) \end{aligned}$$

and

$$\begin{aligned} J_2 &= \left(\sum_{i=1}^{N/2-1}[J(u)]|_{x_{2i+1}} + \sum_{i=1}^{N/2-1}[J(v)]|_{x_{2i}}\right) \\ &\quad + \frac{1}{2}(J(\Phi_2)(x_1^+) - J(\Phi_0)(x_1^-)) + \frac{1}{2}(J(\Phi_{N+1})(x_N^+) - J(\Phi_{N-1})(x_N^-)). \end{aligned}$$

Using the periodic boundary conditions $\Phi_1(x_1) = \Phi_N(x_N)$, and $\Phi_0(x_1^-) = \Phi_{N-1}(x_N^-)$, $\Phi_{N+1}(x_N^+) = \Phi_2(x_1^+)$, we have

$$J_1 + J_2 = 0.$$

This concludes the conservation property. □

We now present some useful bounds which will be used for the stability analysis.

Lemma 3.1. *Let $I = [a, b]$ be an interval of length $|I| = b - a$, and $v \in P^m(I)$, then*

$$|v(b)| \le (m+1)|I|^{-1/2}\|v\|_{L^2(I)}, \tag{3.1a}$$

$$\|\partial_x v\|_{L^2(I)} \le (m+1)\sqrt{m(m+2)}|I|^{-1}\|v\|_{L^2(I)}. \tag{3.1b}$$

Proof. Rescaling so that $I = [-1, 1]$. We consider the normalized Legendre polynomial vector basis $\{\psi_i\}_{i=1}^{m+1}$, satisfying $\int_{-1}^1 \psi_i(\xi)\psi_l(\xi)d\xi = \delta_{il}$ for $1 \le i, l \le m+1$, and

$$\psi_i(1) = \sqrt{\frac{2i-1}{2}}, \quad \partial_\xi \psi_i(1) = \sqrt{\frac{2i-1}{2}}\frac{i(i-1)}{2}, \quad i = 1, \cdots m+1. \tag{3.2}$$

Set $\psi = (\psi_1, \cdots, \psi_{m+1})^\top$ and $v(\xi) = \sum_{i=1}^{m+1} a_i\psi_i(\xi)$, then

$$\int_{-1}^1 |v|^2 d\xi = a^T\left(\int_{-1}^1 \psi\psi^T d\xi\right)a = \sum_{j=1}^{m+1}|a_j|^2 = |a|^2.$$

On the other hand, $v^2(\xi) = a^T\psi(\xi)(\psi(\xi))^T a \le |\psi(\xi)|^2|a|^2$, so that

$$v^2(1) \le |a|^2|\psi(1)|^2 = |a|^2 \sum_{i=1}^{m+1} \frac{2i-1}{2} = |a|^2(m+1)^2/2.$$

Using integration by parts and $\int_{-1}^{1} \psi_j(\xi)\partial_\xi^2\psi_j(\xi) = 0$ we have

$$\begin{aligned}\int_{-1}^{1} v_\xi^2 d\xi &\le |a|^2 \sum_{j=2}^{m+1} \int_{-1}^{1} |\partial_\xi \psi_j(\xi)|^2 d\xi = 2|a|^2 \sum_{j=2}^{m+1} \psi_j(1)\partial_\xi\psi_j(1)\\ &= \frac{1}{2}|a|^2 \sum_{j=2}^{m+1}(2j-1)j(j-1) = \frac{1}{2}|a|^2 \sum_{j=2}^{m+1}(2j^3-3j^2-j)\\ &= \frac{1}{4}|a|^2 m(m+1)^2(m+2).\end{aligned}$$

The desired results follow. □

Remark 3.2. These estimates are known as inverse inequalities. Through a simple argument we show here how the explicit bounds can be obtained. The first bound is well known, see e.g. [27]. The second inequality may be found, say in [28, Lemma 3.3] with a slightly bigger bounding constant.

We next show that the semi-discrete AEDG scheme is L^2 stable for linear problems.

Theorem 3.2. *Let Φ be computed from the AEDG scheme for linear convection–diffusion equations with periodic boundary conditions*

$$\partial_t\phi + \alpha\partial_x\phi = \beta\partial_x^2\phi.$$

Then

$$\frac{d}{dt}\left(\sum_{j=1}^{N-1}\int_{x_j}^{x_{j+1}} \frac{\Phi_{j+1}^2 + \Phi_j^2}{2} dx\right) \le -\beta \sum_{j=1}^{N-1}\int_{x_j}^{x_{j+1}} \frac{(\partial_x\Phi_{j+1})^2 + (\partial_x\Phi_j)^2}{2} dx,$$

provided $\epsilon \le Q\,\Delta x^2$ with

$$Q = \frac{1}{\beta(k+1)^2(17(k+1)^2-1)}. \tag{3.3}$$

Proof. Taking $\eta = \Phi_{2i} =: u$ in (2.3) for $j = 2i$, and $\eta = \Phi_{2i+1} =: v$ in (2.3) for $j = 2i+1$, respectively, it gives

$$\begin{aligned}&\frac{d}{dt}\int_{I_{2i}} \frac{u^2}{2} dx + \int_{I_{2i}} J_x(v)u dx + ([J(v)]u + [a(v)]u_x)|_{x_{2i}} = \frac{1}{\epsilon}\left(\int_{I_{2i}} uv dx - \int_{I_{2i}} u^2 dx\right),\\ &\frac{d}{dt}\int_{I_{2i+1}} \frac{v^2}{2} dx + \int_{I_{2i+1}} J_x(u)v dx + ([J(u)]v + [a(u)]v_x)|_{x_{2i+1}} = \frac{1}{\epsilon}\left(\int_{I_{2i+1}} uv dx - \int_{I_{2i+1}} v^2 dx\right).\end{aligned}$$

Note that

$$[a,b] = \cup_{i=1}^{N/2-1} I_{2i} + [x_{N-1}, x_N] = [x_1, x_2] + \cup_{i=1}^{N/2-1} I_{2i+1},$$

we sum over all index and use (2.5)–(2.6) to obtain

$$\begin{aligned}\frac{d}{dt}\int_a^b \frac{u^2}{2} dx + &\left(\sum_{i=1}^{N/2-1}\int_{I_{2i}} + \int_{x_{N-1}}^{x_N}\right) J_x(v)u dx + \sum_{i=1}^{N/2-1} ([J(v)]u + [a(v)]u_x)|_{x_{2i}}\\ &+ \frac{1}{2}([J(v)]u + [a(v)]u_x)|_{x_N} = \frac{1}{\epsilon}\int_a^b (uv - u^2) dx,\end{aligned}$$

$$\frac{d}{dt}\int_a^b \frac{v^2}{2}dx+\left(\int_{x_1}^{x_2}+\sum_{i=1}^{N/2-1}\int_{I_{2i+1}}\right) J_x(u)v dx+\frac{1}{2}\left([J(u)]v+[a(u)]v_x\right)|_{x_1}$$
$$+\sum_{i=1}^{N/2-1}\left([J(u)]v+[a(u)]v_x\right)|_{x_{2i+1}}=\frac{1}{\epsilon}\int_a^b (uv-v^2)dx.$$

Adding these two relations up, we obtain

$$\frac{d}{dt}\int_a^b \frac{u^2+v^2}{2}dx+J_1+J_2=-\frac{1}{\epsilon}\int_a^b (u-v)^2dx,$$

where all terms in J_1 and J_2 are expressed as follows:

$$J_1=\left(\sum_{i=1}^{N/2}\int_{x_{2i-1}}^{x_{2i}}+\sum_{i=1}^{N/2-1}\int_{x_{2i}}^{x_{2i+1}}\right)[J_x(v)u+J_x(u)v]dx.$$

Note that for $J(w)=\alpha w-\beta w_x$ we have

$$J_x(v)u+J_x(u)v=(J(uv))_x+2\beta u_x v_x=(J(uv))_x-\beta(u_x-v_x)^2+\beta(u_x^2+v_x^2).$$

We see that

$$J_1=-\sum_{i=2}^{N-1}[J(uv)]|_{x_i}-J(uv)(x_1^+)+J(uv)(x_N^-)-\beta\int_a^b (u_x-v_x)^2dx+\beta\int_a^b (u_x^2+v_x^2)dx.$$

For the computation of J_2, we first note that

$$\begin{aligned}[J(v)]u+[a(v)]u_x&=[\alpha v-\beta v_x]u+[\beta v]u_x\\&=[\alpha uv-\beta uv_x-\beta u_x v+\beta u_x v+\beta u_x v]\\&=[J(uv)]+2\beta[v]u_x.\end{aligned}$$

Then, using periodic boundary conditions we can verify that

$$J(uv)(x_1^-)=J(uv)(x_N^-),\quad J(uv)(x_N^+)=J(uv)(x_1^+),$$

hence

$$J_2=\sum_{i=2}^{N-1}[J(uv)]|_{x_i}+2\beta\sum_{i=1}^{N/2-1}\left([v]u_x|_{x_{2i}}+[u]v_x|_{x_{2i+1}}\right)+J(uv)(x_1^+)-J(uv)(x_N^-)$$
$$+\beta(\Phi_2(x_1^+)-\Phi_{N-1}(x_N^-))\Phi_1'(x_1)+\beta(\Phi_2(x_1^+)-\Phi_{N-1}(x_N^-))\Phi_N'(x_N).$$

Using the periodic boundary condition $\Phi_1(x_1)=\Phi_N(x_N)$, we have

$$J_1+J_2=-\beta\int_a^b (u_x-v_x)^2dx+\beta\int_a^b (u_x^2+v_x^2)dx+2\beta\sum_{i=1}^{N/2-1}\left([v]u_x|_{x_{2i}}+[u]v_x|_{x_{2i+1}}\right)$$
$$+\beta(\Phi_2(x_1^+)-\Phi_{N-1}(x_N^-))\Phi_1'(x_1)+\beta(\Phi_2(x_1^+)-\Phi_{N-1}(x_N^-))\Phi_N'(x_N).$$

By Lemma 3.1, we have for $\tau_j=(x_j,x_{j+1})$,

$$|u_x|_{x_{2i}}\le(k+1)h^{-1/2}\min\{\|u_x\|_{L^2(\tau_{2i})},\|u_x\|_{L^2(\tau_{2i-1})}\},$$
$$|v_x|_{x_{2i+1}}\le(k+1)h^{-1/2}\min\{\|v_x\|_{L^2(\tau_{2i})},\|v_x\|_{L^2(\tau_{2i+1})}\},$$
$$|[u-v]|_{x_i}\le(k+1)h^{-1/2}\left(\|u-v\|_{L^2(\tau_{i-1})}+\|u-v\|_{L^2(\tau_i)}\right).$$

Using the fact that $ab \le \delta a^2 + b^2/(4\delta)$, we can bound, for example, one term by

$$\begin{aligned}\left|2\beta \sum_{i=1}^{N/2-1} [v]u_x|_{x_{2i}}\right| &= 2\beta \sum_{i=1}^{N/2-1} [v-u]u_x|_{x_{2i}} \\ &\le 2\beta \sum_{i=1}^{N/2-1} \frac{(k+1)^2}{h}\left(\|v-u\|_{L^2(\tau_{2i-1})} + \|v-u\|_{L^2(\tau_{2i})}\right)\|u_x\|_{L^2(\tau_{2i})} \\ &\le 2\beta\left(\frac{2(k+1)^4}{h^2}\left(\int_{x_1}^{x_{N-2}} + \int_{x_2}^{x_{N-1}}\right)(v-u)^2dx + \frac{1}{4}\int_{x_2}^{x_{N-1}} u_x^2dx\right).\end{aligned}$$

Similarly,

$$\left|2\beta \sum_{i=1}^{N/2-1} [u]v_x|_{x_{2i+1}}\right| \le 2\beta\left(\frac{2(k+1)^4}{h^2}\left(\int_{x_2}^{x_{N-1}} + \int_{x_3}^{x_N}\right)(u-v)^2dx + \frac{1}{4}\int_{x_2}^{x_{N-1}} v_x^2dx\right).$$

The terms at two end points using $\Phi_1(x_1) = \Phi_N(x_N)$ are

$$\begin{aligned}&\beta(\Phi_2(x_1^+) - \Phi_1(x_1) + \Phi_N(x_N) - \Phi_{N-1}(x_N^-))(\Phi_1'(x_1) + \Phi_N'(x_N)) \\ &= \beta((u-v)(x_1^+) - (u-v)(x_N^-))(v_x(x_1) + u_x(x_N)) \\ &\le \frac{\beta(k+1)^2}{h}(\|u-v\|_{L^2(\tau_1)} + \|u-v\|_{L^2(\tau_{N-1})})(\|v_x\|_{L^2(\tau_1)} + \|u_x\|_{L^2(\tau_{N-1})}) \\ &\le 2\beta\left(\frac{(k+1)^4}{h^2}\left(\int_{x_1}^{x_2} + \int_{x_{N-1}}^{x_N}\right)(u-v)^2dx + \frac{1}{4}\left(\int_{x_1}^{x_2} v_x^2dx + \int_{x_{N-1}}^{x_N} u_x^2dx\right)\right).\end{aligned}$$

Thus all terms involving $[v]u_x = [v-u]u_x$ and $[u]v_x = [u-v]v_x$ may be bounded from below by

$$-\frac{\beta}{2}\int_a^b (u_x^2 + v_x^2)dx - 16(k+1)^4\beta h^{-2}\int_a^b (u-v)^2dx,$$

leading to

$$J_1 + J_2 \ge \frac{\beta}{2}\int_a^b (u_x^2 + v_x^2)dx - \beta\int_a^b (u_x - v_x)^2dx - 16(k+1)^4\beta h^{-2}\int_a^b (u-v)^2dx.$$

Hence using (3.1b) in Lemma 3.1 we have

$$\int_a^b (u_x - v_x)^2dx = \sum_{j=1}^{N-1}\int_{x_j}^{x_{N-1}} (u_x - v_x)^2dx \le \frac{k(k+1)^2(k+2)}{h^2}\int_a^b (u-v)^2dx.$$

Hence if $\epsilon \le Q h^2$ with Q as defined in (3.3), then

$$\frac{d}{dt}\int_a^b \frac{u^2+v^2}{2}dx + \frac{\beta}{2}\int_a^b (u_x^2 + v_x^2)dx \le \left(\frac{1}{Q h^2} - \frac{1}{\epsilon}\right)\int_a^b (u-v)^2dx \le 0,$$

which ends the proof. □

4. Multi-dimensional case

We can construct AEDG schemes for multi-dimensional convection–diffusion equations

$$\partial_t\phi + \nabla_x \cdot f(\phi) = \Delta a(\phi), \quad \phi(x,0) = \phi_0(x), \quad x \in \mathbb{R}^d, \quad t > 0.$$

Let $\{x_\alpha\}$ be uniformly distributed grids in $\mathbb{R}^d$. Consider I_α be a hypercube centered at x_α with vertices at $x_{\alpha\pm1}$ where the number of vertices is 2^d.

Centered at each grid $\{x_\alpha\}$, the numerical approximation is a polynomial $\Phi|_{I_\alpha} = \Phi_\alpha(x) \in P_r$, where P_r denotes a linear space of all polynomials of degree at most r in all x_i:

$$P_r := \{p \mid p(x)|_{I_\alpha} = \sum_{0\le|\beta|\le r} a_\beta (x - x_\alpha)^\beta, \quad 1 \le i \le d, \quad a_\beta \in \mathbb{R}\}.$$

Sampling the AE system in I_α, we obtain the semi-discrete AEDG scheme

$$\int_{I_\alpha} (\partial_t \Phi_\alpha + \nabla_x f(\Phi_\alpha^{SN}) - \Delta a(\Phi_\alpha^{SN}))\eta dx = \frac{1}{\epsilon}\int_{I_\alpha} (\Phi_\alpha^{SN} - \Phi_\alpha)\eta dx + B, \tag{4.1}$$

where

$$B = \sum_{j=1}^{d} \int_{I_\alpha} \left(-[f(\Phi_j^{SN})]\eta(x) + [\partial_x a(\Phi_j^{SN})]\eta(x) - [a(\Phi_j^{SN})]\eta_x\right)\delta(x_j - x_{\alpha_j})dx, \tag{4.2}$$

and Φ_α^{SN} are sampled from neighboring polynomials. The choice of Φ_α^{SN} is not unique, and we shall take

$$\Phi_\alpha^{SN} \in \text{span}\{\Phi_{\alpha\pm e_j}\}_{j=1}^d.$$

In the two dimensional case with $\alpha = (j,k)$, the convection–diffusion equation is written as

$$\partial_t\phi + \partial_x f(\phi) + \partial_y g(\phi) = \partial_x^2(a(\phi)) + \partial_y^2(b(\phi)).$$

The semi-discrete scheme in two dimensional case thus has the following form:

$$\begin{aligned}
&\int_{x_{j-1}}^{x_{j+1}}\int_{y_{k-1}}^{y_{k+1}} \left(\partial_t\Phi_{j,k} + \partial_x f(\Phi_{j,k}^{SN}) + \partial_y g(\Phi_{j,k}^{SN}) - \partial_x^2 a(\Phi_{j,k}^{SN}) - \partial_y^2 b(\Phi_{j,k}^{SN})\right)\eta dy dx \\
&= \frac{1}{\epsilon}\int_{x_{j-1}}^{x_{j+1}}\int_{y_{k-1}}^{y_{k+1}} \left(\Phi_{j,k}^{SN} - \Phi_{j,k}\right)dxdy - B,
\end{aligned}$$

where the B term can be computed as

$$\begin{aligned}
B = &-\int_{y_{k-1}}^{y_{k+1}} \left(f(\Phi_{j+1,k}(x_j^+, y)) - f(\Phi_{j-1,k}(x_j^-, y))\right)\eta(x_j, y)dy \\
&-\int_{x_{j-1}}^{x_{j+1}} \left(g(\Phi_{j,k+1}(x, y_k^+)) - g(\Phi_{j,k-1}(x, y_k^-))\right)\eta(x, y_k)dx \\
&+\int_{y_{j-1}}^{y_{k+1}} \left(\partial_x a(\Phi_{j+1,k}(x_j^+, y)) - \partial_x a(\Phi_{j-1,k}(x_j^-, y))\right)\eta(x_j, y)dy \\
&+\int_{x_{j-1}}^{x_{j+1}} \left(\partial_y b(\Phi_{j,k+1}(x, y_k^+)) - \partial_y b(\Phi_{j,k-1}(x, y_k^-))\right)\eta(x, y_k)dx \\
&-\int_{y_{k-1}}^{y_{k+1}} \left(a(\Phi_{j+1,k}(x_j^+, y)) - a(\Phi_{j-1,k}(x_j^-, y))\right)\partial_x\eta(x_j, y)dy \\
&-\int_{x_{j-1}}^{x_{j+1}} \left(b(\Phi_{j,k+1}(x, y_k^+)) - b(\Phi_{j,k-1}(x, y_k^-))\right)\partial_y\eta(x, y_k)dx.
\end{aligned}$$

Similar to the one dimensional case, other terms are as follows:

$$\int_{x_{j-1}}^{x_{j+1}}\int_{y_{k-1}}^{y_{k+1}} \left(\partial_x f(\Phi_{j,k}^{SN}) + \partial_y g(\Phi_{j,k}^{SN})\right)\eta dydx = \int_{x_j}^{x_{j+1}}\int_{y_k}^{y_{k+1}} \left(\partial_x f(\Phi_{j+1,k}) + \partial_y g(\Phi_{j,k+1})\right)\eta(x,y)dydx$$
$$+ \int_{x_j}^{x_{j+1}}\int_{y_{k-1}}^{y_k} \left(\partial_x f(\Phi_{j+1,k}) + \partial_y g(\Phi_{j,k-1})\right)\eta(x,y)dydx$$
$$+ \int_{x_{j-1}}^{x_j}\int_{y_k}^{y_{k+1}} \left(\partial_x f(\Phi_{j-1,k}) + \partial_y g(\Phi_{j,k+1})\right)\eta(x,y)dydx$$
$$+ \int_{x_{j-1}}^{x_j}\int_{y_{k-1}}^{y_k} \left(\partial_x f(\Phi_{j-1,k}) + \partial_y g(\Phi_{j,k-1})\right)\eta(x,y)dydx.$$

A similar computation is used for $\int_{x_{j-1}}^{x_{j+1}}\int_{y_{k-1}}^{y_{k+1}} \left(-\partial_x^2 a(\Phi_{j,k}^{SN}) - \partial_y^2 b(\Phi_{j,k}^{SN})\right)\eta dydx$. An average of two neighboring polynomials $\Phi_{j\pm1,k}$ and $\Phi_{j,k\pm1}$ will be used to evaluate $\int_{x_{j-1}}^{x_{j+1}}\int_{y_{k-1}}^{y_{k+1}} \Phi^{SN}\eta dx$, that is

$$\int_{x_{j-1}}^{x_{j+1}}\int_{y_{k-1}}^{y_{k+1}} \Phi^{SN}\eta dx = \frac{1}{2}\int_{x_j}^{x_{j+1}}\int_{y_k}^{y_{k+1}} (\Phi_{j+1,k} + \Phi_{j,k+1})\eta(x,y)dydx$$
$$+ \frac{1}{2}\int_{x_j}^{x_{j+1}}\int_{y_{k-1}}^{y_k} (\Phi_{j+1,k} + \Phi_{j,k-1})\eta(x,y)dydx$$
$$+ \frac{1}{2}\int_{x_{j-1}}^{x_j}\int_{y_k}^{y_{k+1}} (\Phi_{j-1,k} + \Phi_{j,k+1})\eta(x,y)dydx$$
$$+ \frac{1}{2}\int_{x_{j-1}}^{x_j}\int_{y_{k-1}}^{y_k} (\Phi_{j-1,k} + \Phi_{j,k-1})\eta(x,y)dydx.$$

Boundary conditions are taken the same as in [18].

Remark 4.1. For non-uniform distributed grids, one needs to identify a small neighborhood for each x_α, labeled as I_α, so that each I_α overlaps with neighboring $I_{\alpha'}$, which may not be hypercubes. This set-up corresponds to unstructured meshes.

5. Numerical tests

In this section, we present some numerical tests for the proposed AEDG method. The numerical examples selected are all classical benchmark tests.

We compute the errors as

$$\|e\|_{L^1} = \sum_i \left[\sum_j \int_{K_i^j} |\Phi_i(x) - \phi^{\text{exact}}(x)|dx\right], \tag{5.1}$$

$$\|e\|_{L^\infty} = \max_i \max_j |\Phi_i(x_j) - \phi^{\text{exact}}(x_j)|, \tag{5.2}$$

Table 1
L^1 and L^∞ (using (5.4)) comparison for Example 5.1 at time $T = 0.1$ using P^1 polynomials with $\epsilon = 0.25\Delta x^2$.

$N-1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.20712213		0.06704600	
20	AEDG	0.05198972	1.994	0.01761377	1.928
40	AEDG	0.01258497	2.046	0.00448945	1.972
80	AEDG	0.00303877	2.050	0.00112454	1.997
160	AEDG	7.42929214e−04	2.032	2.78120329e−04	2.015
320	AEDG	1.93290051e−04	1.942	6.62448737e−05	2.069

Table 2
L^1 and L^∞ (using (5.4)) comparison for Example 5.1 at time $T = 0.1$ using P^2 polynomials with $\epsilon = 0.1\Delta x^2$.

$N-1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.006640223		3.08370626e−04	
20	AEDG	7.72579812e−04	3.103	2.13740410e−05	3.850
40	AEDG	8.73207007e−05	3.145	1.39809940e−06	3.934
80	AEDG	9.35862884e−06	3.221	8.84156929e−08	3.983
160	AEDG	9.96171400e−07	3.231	5.53549756e−09	3.997
320	AEDG	1.43151157e−07	2.798	3.46298434e−10	3.998

Table 3
L^1 and L^∞ (using (5.2)) comparison for Example 5.1 at time $T = 0.1$ using P^1 polynomials with $\epsilon = 0.25\Delta x^2$.

$N-1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.20712213		0.04349875	
20	AEDG	0.05198972	1.994	0.01106009	1.975
40	AEDG	0.01258497	2.046	0.00278477	1.989
80	AEDG	0.00303877	2.050	6.97309010e−04	1.997
160	AEDG	7.42929214e−04	2.032	1.74393438e−04	1.999
320	AEDG	1.93290051e−04	1.942	4.36037822e−05	1.999

Table 4
L^1 and L^∞ (using (5.2)) comparison for Example 5.1 at time $T = 0.1$ using P^2 polynomials with $\epsilon = 0.1\Delta x^2$.

$N-1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.006640223		0.00690727	
20	AEDG	7.72579812e−04	3.103	8.80195411e−04	2.972
40	AEDG	8.73207007e−05	3.145	1.10010792e−04	3.000
80	AEDG	9.35862884e−06	3.221	1.37433998e−05	3.000
160	AEDG	9.96171400e−07	3.231	1.71513748e−06	3.002
320	AEDG	1.43151157e−07	2.798	2.14242249e−07	3.001

where the indices i correspond to the original mesh with $x \in K_i = [x_{i-1/2}, x_{i+1/2}] \subset I_i$, j indicates a finer mesh K_i^j such that $\cup_{j=1}^N K_i^j = K_i$, and $\phi^{\text{exact}}(x)$ is the exact solution, or the reference solution.

Example 5.1. We consider the linear equation

$$\partial_t \phi + \partial_x \phi = \beta \partial_x^2 \phi \tag{5.3}$$

with initial data

$$\phi(x, 0) = \sin(x)$$

on $[0, 2\pi]$ with periodic boundary conditions. The exact solution is $\phi(x, t) = \exp(-\beta t)\sin(x - t)$. We take $\beta = 0.001$ and the final time $T = 0.1$. We expect to observe $(k+1)$-th order of accuracy for P^k polynomials.

We point out that for DG methods superconvergence is often observed at some special points. To see this, we measure the error in L^∞-norm by using only grid points,

$$\|e\|_{L^\infty} = \max_i |\Phi_i(x_i) - \phi^{\text{exact}}(x_i)|, \tag{5.4}$$

as shown in Tables 1 and 2, superconvergence of order 4 in L^∞ for P^2 polynomials is indeed observed as shown in Table 2, see also Table 8.

When more sample points are used in the L^∞ norm as defined in (5.2), we no longer observe superconvergence, only the desired $(k+1)$-th order for P^k polynomials, as seen in Tables 3 and 4.

Table 5
L^1 and L^∞ (using (5.2)) comparison for Example 5.1 at time $T = 0.1$ using P^1 polynomials with $\epsilon = 0.05\Delta x^2$.

$N-1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.23708181		0.08716931	
20	AEDG	0.06046435	1.971	0.02384041	1.870
40	AEDG	0.01516808	1.995	0.00610681	1.964
80	AEDG	0.00384927	1.978	0.00153262	1.994
160	AEDG	0.00101861	1.917	3.7920552e−04	2.014
320	AEDG	2.8898900e−04	1.817	9.03325526e−05	2.069

Table 6
L^1 and L^∞ (using (5.2)) comparison for Example 5.1 at time $T = 0.1$ using P^1 polynomials with $\epsilon = 0.01\Delta x^2$.

$N-1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.56733894		0.18503204	
20	AEDG	0.16182547	1.809	0.05427548	1.769
40	AEDG	0.04278443	1.919	0.01416051	1.938
80	AEDG	0.01123781	1.928	0.00357067	1.987
160	AEDG	0.00298459	1.912	8.84491331e−04	2.013
320	AEDG	7.7069789e−04	1.953	2.10761844e−04	2.069

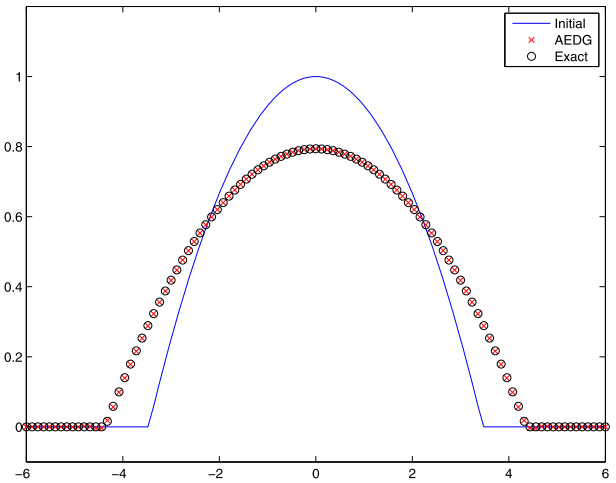

Fig. 1. AEDG scheme for Example 5.2 with $m = 2$ using P^2 polynomials and $N = 101$ at $T = 1$.

We next test the effect of using different values of ϵ satisfying (2.8). We take $\epsilon = 0.05\Delta x^2, 0.01\Delta x^2$ as can be seen in Tables 5 and 6. We observe 2nd order of accuracy in both L^1 and L^∞ norms with the actual error increasing as ϵ decreases.

Example 5.2. The typical porous medium equation in one dimensional case can be written as follows:

$$\partial_t \phi = \partial_x^2(\phi^m), \quad m > 1. \tag{5.5}$$

We test the proposed scheme using the Barenblatt solution:

$$B_m(x,t) = t^{-k}\left[1 - \frac{k(m-1)}{2m}\frac{|x|^2}{t^{2k}}\right]_+^{\frac{1}{m-1}}, \quad k = \frac{1}{m+1}, \tag{5.6}$$

which is an exact solution to (5.5) with compact support. We compute the numerical solution with initial data $B_m(x,1)$ and zero boundary conditions, up to final time $T = 1$; for both initial data (solid line) and numerical solutions (x's) see Fig. 1 for $m = 2$ and Fig. 2 for $m = 3$, both using polynomials of degree 2. A sharp resolution of the solution at the corner is observed for $m = 2$, but we see a bit undershoots at the corner for $m = 3$. In our numerical solutions we do not use any limiter, which when used can prevent numerical oscillations.

Example 5.3. Another standard numerical test is the scalar convection–diffusion Buckley–Leverett equation, which is a model often used in reservoir simulations (two-phase flow), see e.g., [12,13]:

$$\partial_t \phi + \partial_x f(\phi) = \beta \partial_x(\nu(\phi)\partial_x \phi), \tag{5.7}$$

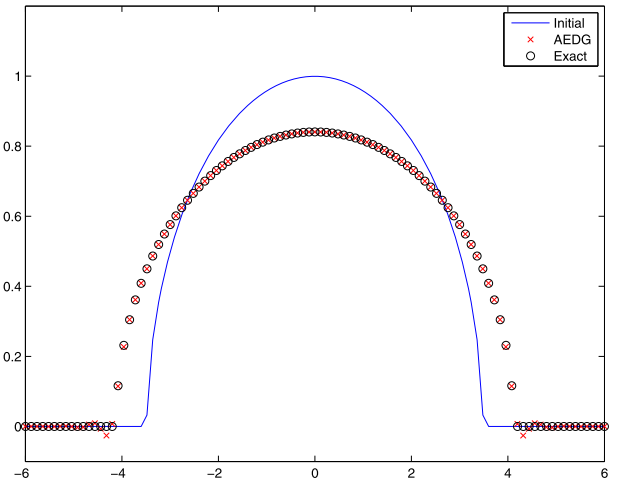

Fig. 2. AEDG scheme for Example 5.2 with $m = 3$ using P^2 polynomials and $N = 101$ at $T = 1$.

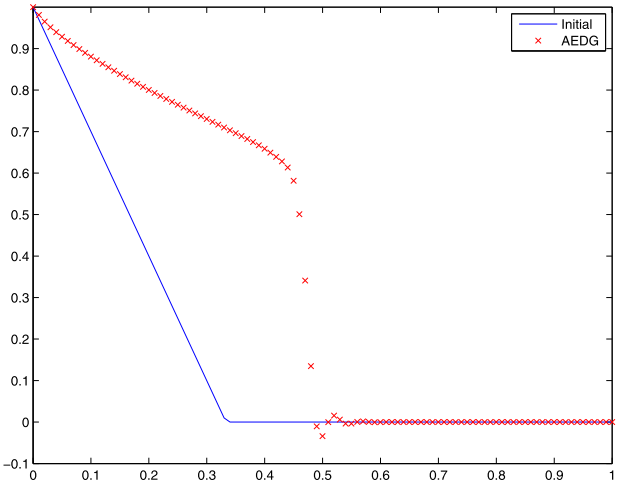

Fig. 3. AEDG scheme for Example 5.3 at $T = 0.2$ using P^2 polynomials and $N = 101$.

where f has an s-shape:

$$f(\phi) = \frac{\phi^2}{\phi^2 + (1-\phi)^2},$$

and

$$\nu(\phi) = 4\phi(1-\phi)1_{0\le\phi\le 1}.$$

The numerical solution using polynomials of degree 2 is computed with $\beta = 0.01$, and initial and boundary conditions as

$$\phi(x,0) = (1-3x)1_{0\le x\le 1/3}, \quad \phi(0,t) = 1, \quad \phi(1,t) = 0.$$

The final time is $T = 0.2$. Both the initial data and the numerical solution computed by the AEDG scheme for grid points with $N = 101$ are presented in Fig. 3. No exact solution to this model problem is available, but if compared with the numerical solutions reported in [13], our solutions seem to converge to the correct entropy solution, except for some oscillations near the shock, see Fig. 3. We point out that no limiter is used in our numerical solutions, yet some minmod-like limiters are used in [13] to prevent numerical oscillations.

Example 5.4. We test the 2D accuracy by solving

$$\partial_t\phi + \partial_x\phi + \partial_y\phi = \beta\Delta\phi, \quad \phi(x,y,0) = \sin(2\pi(x+y)) \tag{5.8}$$

on $[0,1]^2$ with periodic boundary conditions. The exact solution is

$$\phi(x,y,t) = \exp(-8\pi^2\beta t)\sin(2\pi(x+y-2t)).$$

Table 7
L^1 and L^∞ (using (5.4)) comparison for Example 5.4 at time $T = 0.1$ using P^1 polynomials.

$N_x - 1 = N_y - 1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.09313247		0.29243473	
20	AEDG	0.02820924	1.723	0.09990699	1.549
40	AEDG	0.00770212	1.872	0.02750247	1.861
80	AEDG	0.00199733	1.947	0.00705871	1.962
160	AEDG	5.07171319e−04	1.977	0.00177683	1.990
320	AEDG	1.27687688e−04	1.989	4.45133305e−04	1.996

Table 8
L^1 and L^∞ (using (5.4)) comparison for Example 5.4 at time $T = 0.1$ using P^2 polynomials.

$N_x - 1 = N_y - 1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.01715563		0.01194954	
20	AEDG	0.00194596	3.140	0.00115476	3.371
40	AEDG	2.05262495e−04	3.244	7.62286095e−05	3.921
80	AEDG	2.18258015e−05	3.233	4.81607720e−06	3.984
160	AEDG	2.90353143e−06	2.910	3.01949014e−07	3.995

Table 9
L^1 and L^∞ (using (5.2)) comparison for Example 5.4 at time $T = 0.1$ using P^1 polynomials.

$N_x - 1 = N_y - 1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.09313247		0.29201165	
20	AEDG	0.02820924	1.723	0.09946215	1.553
40	AEDG	0.00770212	1.872	0.02705763	1.878
80	AEDG	0.00199733	1.947	0.00661387	2.032
160	AEDG	5.07171319e−04	1.977	0.00133189	2.312

Table 10
L^1 and L^∞ (using (5.2)) comparison for Example 5.4 at time $T = 0.1$ using P^2 polynomials.

$N_x - 1 = N_y - 1$	Scheme	L^1 error	L^1 order	L^∞ error	L^∞ order
10	AEDG	0.01715563		0.02645451	
20	AEDG	0.00194596	3.140	0.00338011	2.968
40	AEDG	2.05262495e−04	3.244	4.10763010e−04	3.040
80	AEDG	2.18258015e−05	3.233	5.07698157e−05	3.016
160	AEDG	2.90353143e−06	2.910	6.36046021e−06	2.996

We take $\beta = 0.0001$ and the final time $T = 0.1$. The results are shown in Tables 7 and 8 with L^∞ norm defined in (5.4) and show that we get the desired order for P^1 and P^2 polynomials. Again superconvergence of order 4 in L^∞ for P^2 polynomials is observed as shown in Table 8.

We then take the L^∞ norm to be the same as in (5.2). Superconvergence is no longer observed as seen in Tables 9 and 10.

Example 5.5. We test the two dimensional porous medium equation of the form

$$\partial_t \phi = \Delta(\phi^2) \tag{5.9}$$

on $[-1, 1]^2$ with periodic boundary conditions. The initial condition is taken as

$$\phi(x, y, 0) = 1_{[-0.5,0.5]^2}.$$

We compute our numerical solution up to time $T = 0.02$, see the contour plot in Fig. 4. We can see that numerical solutions remain positive, and no oscillations are observed, see also the section view along line $x = y$ in Fig. 5.

6. Concluding remarks

In this paper we have proposed a simple alternating evolution discontinuous Galerkin (AEDG) method for solving convection–diffusion equations. The AEDG schemes avoid the choice of numerical fluxes and provide high order approximating polynomials near each grid point. The scheme parameter is bigger than the time step, but less than $Q\,\Delta x^2$ under which the stability of the scheme is established. In addition, since the method does not rely on the divergence form of the underlying PDEs, it can be applied to fully nonlinear second order PDEs, which is under our current investigation.

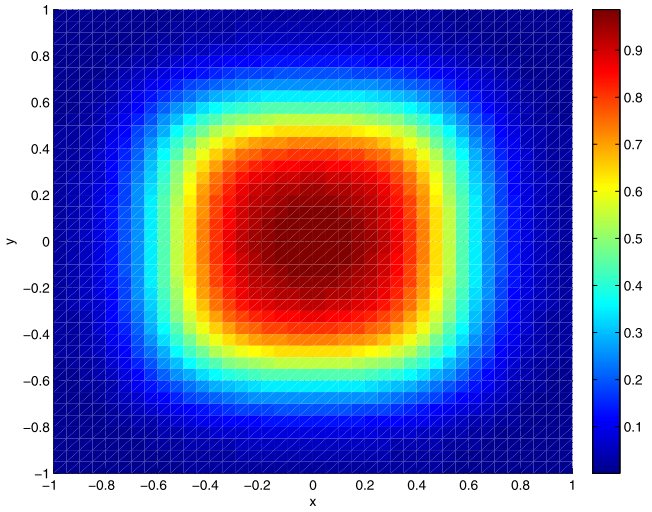

Fig. 4. Contour plot for Example 5.5 at time $T = 0.02$ with $N_x = N_y = 41$ using P^2 polynomials.

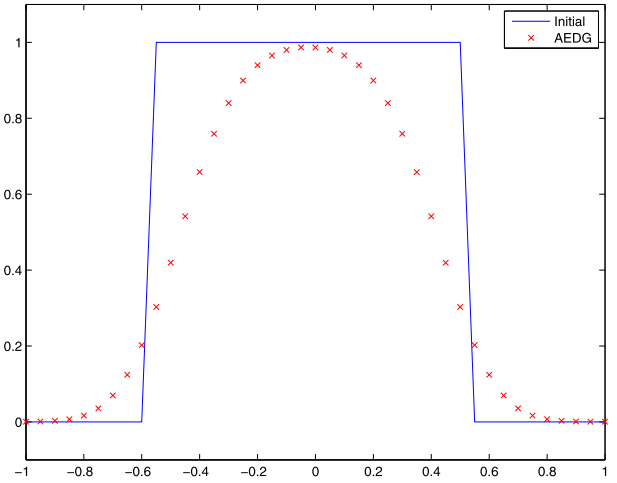

Fig. 5. AEDG plot along the line $y = x$ at time $T = 0.02$ with $N_x = N_y = 41$ using P^2 polynomials.

Acknowledgements

This work was supported by the National Science Foundation under Grant DMS1312636 and by NSF Grant RNMS (KI-Net) 1107291.

References

[1] D.N. Arnold, An interior penalty finite element method with discontinuous elements, SIAM J. Numer. Anal. 19 (4) (1982) 742–760.
[2] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2001/02) 1749–1779 (electronic).
[3] Ahmed Haseena, High-resolution alternating evolution schemes for hyperbolic conservation laws and Hamilton–Jacobi equations, PhD thesis, Department of Mathematics, Iowa State University, 2008.
[4] G.A. Baker, Finite element methods for elliptic equations using nonconforming elements, Math. Comput. 31 (1977) 45–59.
[5] C.E. Baumann, J.T. Oden, A discontinuous hp finite element method for convection–diffusion problems, Comput. Methods Appl. Mech. Eng. 175 (3–4) (1999) 311–341.
[6] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (2) (1997) 267–279.
[7] B. Cockburn, B. Dong, J. Guzman, M. Restelli, R. Sacco, A hybridizable discontinuous Galerkin method for steady-state convection–diffusion–reaction problems, SIAM J. Sci. Comput. 31 (2009) 3827–3846.
[8] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[9] Y.-D. Cheng, C.-W. Shu, A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives, Math. Comput. 77 (262) (2008) 699–730.

[10] X. Feng, R. Glowinski, M. Neilan, Recent developments in numerical methods for fully nonlinear second order partial differential equations, SIAM Rev. 55 (2) (2013) 205–267.
[11] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.
[12] K.H. Karlsen, K. Brusdal, H.K. Dahle, S. Evje, K.-A. Lie, The corrected operator splitting approach applied to a nonlinear advection–diffusion problem, Comput. Methods Appl. Mech. Eng. 167 (1998) 239–260.
[13] A. Kurganov, E. Tadmor, New high resolution central schemes for nonlinear conservation laws and convection–diffusion equations, J. Comput. Phys. 160 (2000) 241–282.
[14] H. Liu, An alternating evolution approximation to systems of hyperbolic conservation laws, J. Hyperbolic Differ. Equ. 5 (2) (2008) 1–27.
[15] Y. Liu, Central schemes on overlapping cells, J. Comput. Phys. 209 (1) (2005) 82–104.
[16] B. van Leer, S. Nomura, Discontinuous Galerkin for diffusion, in: Proceedings of 17th AIAA Computational Fluid Dynamics Conference, 2005, 2005–5108.
[17] H. Liu, M. Pollack, H. Saran, Alternating evolution schemes for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 35 (1) (2013) 122–149.
[18] H. Liu, M. Pollack, Alternating evolution discontinuous Galerkin methods for Hamilton–Jacobi equations, J. Comput. Phys. 258 (1) (2014) 31–46.
[19] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems, SIAM J. Numer. Anal. 47 (1) (2009) 675–698.
[20] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections, Commun. Comput. Phys. 8 (3) (2010) 541–564.
[21] J.T. Oden, I. Babuska, C.E. Baumann, A discontinuous hp finite element method for diffusion problems, J. Comput. Phys. 146 (2) (1998) 491–519.
[22] Y.-J. Liu, C.-W. Shu, E. Tadmor, M.-P. Zhang, L^2 stability analysis of the central discontinuous Galerkin method and a comparison between the central and regular discontinuous Galerkin methods, ESAIM: Math. Model. Numer. Anal. 42 (4) (2008) 593–607.
[23] Y. Liu, C.W. Shu, E. Tadmor, M. Zhang, Central local discontinuous Galerkin methods on overlapping cells for diffusion equations, ESAIM: Math. Model. Numer. Anal. 45 (6) (2011) 1009–1032.
[24] M. van Raalte, B. van Leer, Bilinear forms for the recovery-based discontinuous Galerkin method for diffusion, Commun. Comput. Phys. 5 (2009) 683–693.
[25] H. Saran, H. Liu, Formulation and analysis of alternating evolution schemes for conservation laws, SIAM J. Sci. Comput. 33 (2011) 3210–3240.
[26] M.F. Wheeler, An elliptic collocation-finite element method with interior penalties, SIAM J. Numer. Anal. 15 (1978) 152–161.
[27] T. Warburton, J.S. Hesthaven, On the constants in hp-finite element trace inequalities, Comput. Methods Appl. Mech. Eng. 192 (2003) 2765–2773.
[28] T. Warburton, T. Hagstrom, Taming the CFL number for discontinuous Galerkin methods on structured meshes, SIAM J. Numer. Anal. 46 (6) (2008) 3151–3180.