MATHEMATICS OF COMPUTATION
Volume 81, Number 280, October 2012, Pages 1929–1950
S 0025-5718(2012)02586-5
Article electronically published on March 2, 2012

ACCURACY-ENHANCEMENT OF DISCONTINUOUS GALERKIN SOLUTIONS FOR CONVECTION-DIFFUSION EQUATIONS IN MULTIPLE-DIMENSIONS

LIANGYUE JI, YAN XU, AND JENNIFER K. RYAN

Abstract. Discontinuous Galerkin (DG) methods exhibit "hidden accuracy" that makes superconvergence of this method an increasing popular topic to address. Previous investigations have focused on the superconvergent properties of ordinary differential equations and linear hyperbolic equations. Additionally, superconvergence of order $k+\frac{3}{2}$ for the convection-diffusion equation that focuses on a special projection using the upwind flux was presented by Cheng and Shu. In this paper we demonstrate that it is possible to extend the smoothness-increasing accuracy-conserving (SIAC) filter for use on the multi-dimensional linear convection-diffusion equation in order to obtain $2k+m$ order of accuracy, where m depends upon the flux and takes on the values 0, $\frac{1}{2}$, or 1. The technique that we use to extract this hidden accuracy was initially introduced by Cockburn, Luskin, Shu, and Süli for linear hyperbolic equations and extended by Ryan et al. as a smoothness-increasing accuracy-conserving filter. We solve this convection-diffusion equation using the local discontinuous Galerkin (LDG) method and show theoretically that it is possible to obtain $\mathcal{O}(h^{2k+m})$ in the negative-order norm. By post-processing the LDG solution to a linear convection equation using a specially designed kernel such as the one by Cockburn et al., we can compute this same order accuracy in the L^2-norm. Additionally, we present numerical studies that confirm that we can improve the LDG solution from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+1})$ using alternating fluxes and that we actually obtain $\mathcal{O}(h^{2k+2})$ for diffusion-dominated problems.

1. Introduction

Discontinuous Galerkin (DG) methods exhibit "hidden accuracy" that makes the superconvergence of this method an increasing popular topic to address. Previous investigations have focused on the superconvergent properties of ordinary differential equations, linear hyperbolic equations, and using a special projection of the solution [1, 4, 3, 23]. This paper shows that by using an alternative convolution kernel approach we can obtain accuracy of order $2k+m$ for the multi-dimensional

Received by the editor September 26, 2010 and, in revised form, April 26, 2011 and July 1, 2011.

2010 *Mathematics Subject Classification.* Primary 65M60; Secondary 35K10, 35L02.

Key words and phrases. discontinuous Galerkin method, convection-diffusion equations, negative-order norm error estimates, filtering, post-processing, accuracy enhancement.

The research of the second author was supported by NSFC grant No.10971211, No. 11031007, FANEDD No. 200916, FANEDD of CAS, NCET No. 09-0922 and the Fundamental Research Funds for the Central Universities. Additional support was provided by the Alexander von Humboldt-Foundation while the author was in residence at Freiburg University, Germany.

time-dependent linear convection equation,

$$u_t + \sum_{i=1}^{d} a_i\, u_{x_i} + a_o u - \epsilon \Delta u = 0, \quad (\boldsymbol{x}, t) \in \Omega \times (0, T], \tag{1.1}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \tag{1.2}$$

where $a_i,\ i = 0, \ldots, d$ are constant, $\epsilon \geq 0$, and k is the highest degree polynomial used in the approximation and $m = 0,\ \frac{1}{2},\ 1$, depends upon the flux. For simplicity, we always consider the domain Ω to have unit length in each coordinate direction.

In order to solve this problem computationally, we implement the local discontinuous Galerkin (LDG) method [6, 8, 9]. This technique uses an approximation space consisting of piecewise polynomials of degree less than or equal to k. However, to deal with the higher-order derivative term, we implement an alternative auxiliary formulation common in differential equations. The LDG method is a particularly nice method that is suitable for unstructured meshes as well as parallelization due to the properties of the approximation space and the choice of the numerical flux.

Superconvergence of convection-diffusion equations has been addressed by Adjerid and Klauser in [1]. They demonstrated that it was possible to obtain $k{+}2$ order of accuracy at the Radau points for convection dominated differential equations. The same order of accuracy is obtained for diffusion dominated problems for the derivative of the solution at the roots of the derivative of the Radau polynomials. Additionally, superconvergence for the convection-diffusion partial differential equation that focuses on a special projection using the upwind flux was presented by Cheng and Shu in [4]. This superconvergence was of the order $k + \frac{3}{2}$. Celiker and Cockburn found superconvergence of order $2k{+}1$ for the numerical traces at the nodes of the mesh for conservative solutions [3]. Recently, Zhang, Xie, and Zhang [23] built upon the work of Celiker and Cockburn and relate the leading terms of the discretization errors of the minimal dissipation LDG method to the right and left Radau polynomials, as well as to the Legendre polynomials for the consistent DG method. However, our method focuses on extracting this "extra" accuracy from the solution by using a specially designed convolution kernel to obtain superconvergence.

In this paper, we provide a theoretical and computational extension of this convolution kernel approach to obtain superconvergence for the multi-dimensional linear convection-diffusion equation that improves the convergence rate from order $k{+}1$ for the LDG method to $2k{+}m$, $m = 0,\ \frac{1}{2},\ 1$, for the filtered LDG method, where m depends upon the flux. This convolution kernel falls under the area of smoothness-increasing accuracy-conserving (SIAC) filters studied by Ryan, Kirby et al. [17]. This technique was initially introduced for discontinuous Galerkin approximations by Cockburn, Luskin, Shu, and Süli for linear hyperbolic equations [7]. It is based upon previous work by Mock and Lax [12] and Bramble and Schatz [2]. We solve this convection-diffusion equation using the LDG method and show theoretically that it is possible to obtain $\mathcal{O}(h^{2k+1})$ in the negative-order norm, provided an alternating flux is chosen. By applying the SIAC filter to the LDG solution of a linear convection-diffusion equation, we can compute this same order accuracy in the L^2-norm. Additionally, we present numerical studies that confirm that we can improve the LDG solution from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+1})$ and that we actually obtain $\mathcal{O}(h^{2k+2})$ for diffusion-dominated problems.

The paper is organized as follows. In Section 2, we introduce the LDG method and SIAC filtering as well as the relevant notation that will be required for the proof of our method. In Section 3 we prove the negative-order norm estimates for the multi-dimensional linear convection-diffusion equation. These results are confirmed numerically in Section 4.

2. Notations, definitions and projections

We begin by defining the necessary notations used in the proof of accuracy enhancement of discontinuous Galerkin solutions for linear convection-diffusion equations. This is done for projections and interpolations for the finite element spaces used in the error analysis.

2.1. **Tessellation and function spaces.**

Let $\mathcal{T}_h$ denote a tessellation of the domain Ω with shape-regular elements K. Let Γ denote the union of the boundary faces of elements $K \in \mathcal{T}_h$, i.e., $\Gamma = \bigcup_{K\in\mathcal{T}_h} \partial K$, and $\Gamma_0 = \Gamma\backslash\partial\Omega$.

In order to describe the flux functions define e to be a face shared by the "left" and "right" elements K_L and K_R (we refer to [21] and [20] for a proper definition of "left" and "right" in our context). The normal vectors ν_L and ν_R on edge e point exterior to K_L and K_R, respectively. If ψ is a function on K_L and K_R, but possibly discontinuous across e, let ψ^L denote $(\psi|_{K_L})|_e$ and ψ^R denote $(\psi|_{K_R})|_e$, the left and right traces, respectively.

Let $\mathcal{Q}^k(K)$ be the space of tensor product polynomials of degree at most $k \geq 0$ on $K \in \mathcal{T}_h$ in each variable. The finite element spaces are denoted by

$$\begin{aligned} V_h =& \Big\{\varphi \in L^2(\Omega): \quad \varphi|_K \in \mathcal{Q}^k(K), \quad \forall K \in \mathcal{T}_h\Big\}, \\ \Sigma_h =& \Big\{\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)^T \in (L^2(\Omega))^d: \quad \eta_l|_K \in \mathcal{Q}^k(K), \;\; l = 1\cdots d, \quad \forall K \in \mathcal{T}_h\Big\}. \end{aligned}$$

For purposes of the negative-order norm estimates, it is also allowable to consider the space

$$\begin{aligned} V_h =& \Big\{\varphi \in L^2(\Omega): \quad \varphi|_K \in \mathcal{P}^k(K), \quad \forall K \in \mathcal{T}_h\Big\}, \\ \Sigma_h =& \Big\{\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)^T \in (L^2(\Omega))^d: \quad \eta_l|_K \in \mathcal{P}^k(K), \;\; l = 1\cdots d, \quad \forall K \in \mathcal{T}_h\Big\}, \end{aligned}$$

where $\mathcal{P}^k(K)$ is the usual polynomial space. For the one-dimensional case, we have $\mathcal{Q}^k(K) = \mathcal{P}^k(K)$. Note that functions in V_h and Σ_h are allowed to have discontinuities across element interfaces. Furthermore, we note that this flexibility in the choice of the function spaces ($\mathcal{P}^k$ or $\mathcal{Q}^k$) applies to the negative-order norm error estimates of the LDG solution, but that the post-processing kernel is applied in a tensor product fashion and therefore for the superconvergence extraction by the kernel we require $\mathcal{Q}^k$ polynomials.

We use the following notation for the L^2-norm in Ω and on the boundary

$$\|\eta\|_\Omega = \left(\int_\Omega \eta^2 d\boldsymbol{x}\right)^{\frac{1}{2}}, \quad \|\eta\|_{\partial\Omega} = \left(\int_{\partial\Omega} \eta^2 ds\right)^{\frac{1}{2}}, \tag{2.1}$$

and define the ℓ-norm in Ω as

$$\|\eta\|_{\ell,\Omega} = \left(\sum_{|\alpha|\leq\ell} \|D^\alpha \eta\|_\Omega^2\right)^{\frac{1}{2}}, \quad \ell > 0. \tag{2.2}$$

Without loss of generality, we use the notation $\|\eta\|_\ell$ instead of $\|\eta\|_{\ell,\Omega}$. We also use the following inner product notation

$$(w,v)_\Omega = \sum_K \int_K wv\,dK, \quad (\boldsymbol{q},\boldsymbol{p})_\Omega = \sum_K \int_K \boldsymbol{q}\cdot\boldsymbol{p}\,dK. \tag{2.3}$$

Lastly, given $\ell > 0$, we define the negative-order norm on the domain Ω as

$$\|\eta\|_{-\ell,\Omega} = \sup_{\Phi\in C_0^\infty(\Omega)} \frac{(\eta,\Phi)_\Omega}{\|\Phi\|_{\ell,\Omega}}. \tag{2.4}$$

We define the following notation for the difference quotients

$$\partial_{h,j} v(\boldsymbol{x}) = \frac{1}{h}\left(v\left(\boldsymbol{x}+\frac{1}{2}h\boldsymbol{e}_j\right) - v\left(\boldsymbol{x}-\frac{1}{2}h\boldsymbol{e}_j\right)\right), \tag{2.5}$$

here $\boldsymbol{e}_j$ is the multi-index whose jth component is 1 and all others 0. For any multi-index $\alpha = (\alpha_1,\cdots,\alpha_d)$ we set the αth-order difference quotient

$$\partial_h^\alpha v(\boldsymbol{x}) = (\partial_{h,1}^{\alpha_1}\cdots\partial_{h,d}^{\alpha_d})v(\boldsymbol{x}). \tag{2.6}$$

2.2. **Projection and interpolation properties.** For the accuracy enhancement of the DG solution to be effective, it is necessary to perform an L^2-projection of the initial function. In what follows, we will consider the standard L^2-projection P for scalar functions and Π for vector-valued functions. Denote $\Omega = \bigcup K$ as the domain. K is any element in our mesh with $\Gamma = \bigcup \partial K$ being the sum of all the boundary elements ∂K. It is well known that (cf. [5])

$$\begin{aligned} \|\eta^e\|_\Omega + h\|\eta^e\|_{L^\infty(\Omega)} + h^{\frac{1}{2}}\|\eta^e\|_\Gamma &\le Ch^{k+1}\|\eta\|_{k+1,\Omega}, \\ \|\boldsymbol{p}^e\|_\Omega + h\|\boldsymbol{p}^e\|_{L^\infty(\Omega)} + h^{\frac{1}{2}}\|\boldsymbol{p}^e\|_\Gamma &\le Ch^{k+1}\|\boldsymbol{p}\|_{k+1,\Omega}, \end{aligned} \tag{2.7}$$

where $\eta^e = P\eta - \eta$ and $\boldsymbol{p}^e = \Pi\boldsymbol{p} - \boldsymbol{p}$. We note that the positive constant C is independent of h. These properties of the projection will be used to estimate the negative-order norm in Lemmas 3.3–3.5.

2.3. **Regularity for the convection-diffusion equation.** An essential ingredient in the error analysis is given by the regularity result:

Lemma 2.1. *For any time t we have the property for equation* (1.1)

$$\|u(\boldsymbol{x},t)\|_{\ell,\Omega} \le C\|u(\boldsymbol{x},0)\|_{\ell,\Omega} \tag{2.8}$$

and

$$\left(\int_0^T \|u(\boldsymbol{x},t)\|^2_{\ell+1,\Omega}\,dt\right)^{1/2} \le C\|u(\boldsymbol{x},0)\|_{\ell,\Omega}, \tag{2.9}$$

for $\ell \ge 0$, where C is a constant depends on a, ε and final time T.

Remark 2.1. The proof of Lemma 2.1 is trivial and can be found by simply applying the method of separation of variables.

2.4. **The LDG method for convection-diffusion equations.** In this section, we consider the LDG method for convection-diffusion equations in Ω,

$$u_t + \sum_{i=1}^{d} a_i\, u_{x_i} + a_o u - \epsilon \Delta u = 0, \quad (\boldsymbol{x}, t) \in \Omega \times (0, T], \tag{2.10}$$

with $\epsilon \geq 0$ and smooth initial conditions,

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}). \tag{2.11}$$

We assume that periodic boundary conditions are given.

In order to define the LDG scheme, we begin by defining $\mu = \sqrt{\epsilon}$ and rewriting the equations using an auxiliary formulation,

$$u_t + \sum_{i=1}^{d} a_i\, u_{x_i} + a_o u - \mu \nabla\cdot \boldsymbol{q} = 0, \tag{2.12a}$$

$$\boldsymbol{q} - \mu \nabla u = 0. \tag{2.12b}$$

The approximate solution, $(u_h(\boldsymbol{x}, t), \boldsymbol{q}_h(\boldsymbol{x}, t))$, given by the LDG method are sought in the finite element space V_h and Σ_h. That is, for any $\psi \in V_h$ and $\boldsymbol{\phi} \in \Sigma_h$, $(u_h, \boldsymbol{q}_h)$ satisfies

$$\int_K (u_h)_t \psi \, dK - \sum_{i=1}^{d} \int_K u_h (a_i\, \psi_{x_i})\, dK + \int_K a_o u_h \psi \, dK + \int_K \mu \boldsymbol{q}_h \cdot \nabla \psi \, dK$$

$$+ \sum_{i=1}^{d} \int_{\partial K} a_i \nu_i \widetilde{u_h} \psi \, ds - \int_{\partial K} \mu \widehat{\boldsymbol{q}_h} \cdot \boldsymbol{\nu} \psi \, ds = 0, \tag{2.13a}$$

$$\int_K \boldsymbol{q}_h \cdot \boldsymbol{\phi} \, dK + \int_K \mu u_h \nabla \cdot \boldsymbol{\phi} \, dK - \int_{\partial K} \mu \widehat{u_h} \boldsymbol{\nu} \cdot \boldsymbol{\phi} \, ds = 0, \tag{2.13b}$$

where $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d)$ is the unit outward normal vector for the integration domain. The "hat" and "tilde" terms in (2.13) in the cell boundary terms obtained from integration by parts are the so-called "numerical fluxes", which are single-valued functions defined on the edges. These fluxes are designed based on different guiding principles for the given partial differential equation in order to ensure stability [20]. For the numerical experiments, the choice of $\widetilde{u_h}$ is chosen to satisfy the upwind condition depending on the sign of a_i, $i = 1, \ldots, d$. Without lost of generality, we assume $a_i \geq 0$, $i = 1, \ldots, d$, which gives

$$\widetilde{u_h} = u_h^L \tag{2.14}$$

and $\widehat{u_h}$, $\widehat{\boldsymbol{q}_h}$ are chosen to be alternating fluxes, i.e.,

$$\widehat{u_h} = u_h^L, \quad \widehat{\boldsymbol{q}_h} = \boldsymbol{q}_h^R \tag{2.15}$$

or

$$\widehat{u_h} = u_h^R, \quad \widehat{\boldsymbol{q}_h} = \boldsymbol{q}_h^L. \tag{2.16}$$

Summing (2.13) over K, we get

$$((u_h)_t, \psi)_\Omega + B_1(\boldsymbol{q}_h, u_h; \psi) = 0, \tag{2.17}$$

$$(\boldsymbol{q}_h, \boldsymbol{\phi})_\Omega + B_2(u_h; \boldsymbol{\phi}) = 0, \tag{2.18}$$

where the bilinear forms B_1 and B_2 are defined as

$$\begin{aligned} B_1(\boldsymbol{q}_h, u_h; \psi) &= -\sum_{i=1}^{d}(u_h, a_i\psi_{x_i})_\Omega + (u_h, a_o\psi)_\Omega + (\boldsymbol{q}_h, \mu\nabla\psi)_\Omega \\ &\quad + \sum_K \left(\sum_{i=1}^{d} \int_{\partial K} a_i\nu_i\widehat{u_h}\psi\, ds - \int_{\partial K} \mu\widehat{\boldsymbol{q}_h}\cdot \boldsymbol{\nu}\psi\, ds \right), \\ B_2(u_h; \boldsymbol{\phi}) &= (u_h, \mu\nabla\cdot \boldsymbol{\phi})_\Omega - \sum_K \int_{\partial K} \mu\widehat{u_h}\boldsymbol{\nu}\cdot \boldsymbol{\phi}\, ds. \end{aligned}$$

Although this paper focuses on the use of scheme (2.13), we can easily obtain a different method to solve the convection-diffusion equation depending on the choice of numerical fluxes [19, 22]. The choice of the flux will give different accuracy results that affect the estimates for negative-order norm. Therefore, we concentrate on the general results for the convection-diffusion equation where the flux is not specified. However, to complete our superconvergence result we will need the following lemma:

Lemma 2.2. *If $u, \boldsymbol{q}$ and $u_h, \boldsymbol{q}_h$ are solutions to* (2.12) *and* (2.13)*, respectively, then for $T > 0$ we have*

$$(2.19) \qquad \max_t \|u - u_h\|_\Omega + \left(\int_0^T \|\boldsymbol{q} - \boldsymbol{q}_h\|_\Omega^2\, dt \right)^{1/2} \leq Ch^{k+m},$$

where C is a constant independent of h that depends on $\|u_0\|_{H^{k+1}}$ and T. m is some constant that depends on the choice of numerical flux in (2.13)*.*

Remark 2.2. For Lemma 2.2, $m \geq 0$ and can be taken to be 0, $\frac{1}{2}$ or 1. In this paper, we consider the case where $m = 1$, this is for the choice of fluxes given in (2.14) and (2.15) or (2.16). We refer the reader to [19] and [22] for details.

2.5. **Smoothness-increasing accuracy-conserving filters.** In order to obtain the superconvergence of the LDG method in the L^2-norm, we implement a smoothness-increasing accuracy-enhancing (SIAC) filter. This filter works by essentially inducing smoothness in the field by convolving the solution against a specially chosen kernel,

$$(2.20) \qquad u_h^\star = K_h^{2k+1,k+1} \star u_h.$$

In this formulation, $u_h^\star$ is the filtered solution, u_h is the LDG solution calculated at the final time, and $K_h^{2k+1,k+1}(\boldsymbol{x}) = K^{2k+1,k+1}(\boldsymbol{x}/h)/h^d$ is the convolution kernel. This kernel allows the extraction of hidden accuracy that is detected by the negative-order norm. It is a linear combination of B-splines of order $k+1$ obtained by convolving the characteristic function over the interval $(-\frac{1}{2}, \frac{1}{2})$ with itself k times. Using B-splines makes this kernel computationally efficient, provided the mesh is uniform, as the kernel is translation invariant and is locally supported in at most $2k + 2$ elements. The one-dimensional convolution kernel is of the form

$$(2.21) \qquad K_h^{2k+1,k+1}(x) = \frac{1}{h} \sum_{\gamma=-k}^{k} c_\gamma^{2k+1,k+1} \psi^{(k+1)} \left(\frac{x}{h} - \gamma \right),$$

and, given an arbitrary $\boldsymbol{x} = (x_1, \dots, x_d) \in R^d$, we set

$$(2.22) \qquad \psi^{(k+1)}(\boldsymbol{x}) = \psi^{(k+1)}(x_1) \cdots \psi^{(k+1)}(x_d).$$

The kernel for the multidimensional space considered is of the form

$$K_h^{2k+1,k+1}(\boldsymbol{x}) = \sum_{\gamma\in Z^d} \mathbf{c}_\gamma^{2k+1,k+1}\psi^{(k+1)}(\boldsymbol{x}-\gamma). \tag{2.23}$$

The coefficients, $\mathbf{c}_\gamma^{2k+1,k+1}$, are tensor products of the one-dimensional coefficients. These one-dimensional coefficients are chosen such that $K^{2(k+1),k+1} * p = p$ for polynomials p of degree $2k$.

The original use of this kernel was by Mock and Lax [12] and Bramble and Schatz [2] for continuous finite element solutions. Cockburn, Luskin, Shu, and Süli [7] specifically addressed linear hyperbolic equations using the discontinuous Galerkin method and demonstrate that it is feasible to raise the order of accuracy of the DG solution from k+1 to $2k$+1. Below we discuss the kernel properties as well as the computational efficiency.

2.5.1. *Kernel Properties.* This convolution kernel has the two following important properties that are used to show its accuracy enhancing capabilities:

Proposition 2.3 (Bramble and Schatz [2]). *Let $\Omega_0 + 2supp(K_h^{2k+1,k+1}(x)) \subset\subset \Omega_1 \subset \Omega$ then*

$$\|u - K_h^{2k+1,k+1} \star u\|_{0,\Omega_0} \leq Ch^{2k+1}|u|_{0,\Omega_1}, \tag{2.24}$$

where C depends solely on $\Omega_0, \Omega_1, d, k, c_\gamma^{2k+1,k+1}$, and is independent of h.

The second important property allows us to express derivatives of the convolution with the kernel in terms of simple difference quotients.

Proposition 2.4 (Bramble and Schatz [2]). *Let $\Omega_0 + 2supp(K_h^{2k+1,k+1}(x)) \subset\subset \Omega_1 \subset \Omega$ and α be any multi-index with $\alpha_i \leq k+1, i = 1,\ldots,d$. Then for any fixed integer s (positive or negative) we have*

$$\|D^\alpha(K_h^{2k+1,k+1} \star u)\|_{s,\Omega_0} \leq C\|\partial_h^\alpha u\|_{s,\Omega_1}, \textit{for all}\, u \in H^s(\Omega_1), \tag{2.25}$$

where C depends solely on $\Omega_0, \Omega_1, d, k, c_\gamma^{2k+1,k+1}$, and is independent of h.

Using Propositions 2.3 and 2.4 we state an approximation result which allows the kernel to extract the hidden accuracy in the negative-order norm.

Theorem 2.5 (Bramble and Schatz [2]). *For $T > 0$, let u be the exact solution of the problem* (2.10). *Let $\Omega_0 + 2supp(K_h^{2k+1,k+1}(x)) \subset\subset \Omega_1 \subset \Omega$, where U is any approximation to u, then*

$$\begin{aligned}\|u(T) - K_h^{2k+1,k+1} \star U\|_{0,\Omega_0} &\leq C_1 h^{2k+1}|u|_{2k+1,\Omega_1}\\ &+ C_2 \sum_{|\alpha|\leq k+1} \|\partial_h^\alpha(u-U)\|_{-(k+1),\Omega_1}.\end{aligned}$$

where C_1 and C_2 depends solely on $\Omega_0, \Omega_1, d, k, c_\gamma^{2k+1,k+1}$, independent of h.

Remark 2.3. For our problem we only consider periodic boundary conditions and we take $U = u_h$ to be an approximation obtained by using the LDG method. We can obtain the estimation for the whole domain, i.e., $\Omega_0 = \Omega$ by considering $\Omega\backslash\Omega_0$ as the interior part of a new period.

We neglect to provide the details of this proof, which is shown in [7] and [2] as well as that of Ryan and Cockburn [13] and Thomée [18], where a similar result was presented for the derivatives of the post-processed solution for linear hyperbolic equations.

2.5.2. *Computational Efficiency.* We emphasize that this accuracy enhancement is achieved by post-processing the approximate solution only once, at the end of the computation, at $t = T$. This makes for efficient computation of the post-processed approximation. Additionally, it is computationally efficient due to the locally compact support property and can further be improved by choosing specific points within an element as evaluation points, such as the gauss points. This allows the post-processed solution to be evaluated through small matrix-vector operations.

For further illustration of the computational efficiency, let us investigate the one-dimensional computation. Using the uniform mesh assumption the exact evaluation of $u_h^*(x)$ is evaluated by using (2.20) and (2.21). This allows us to write the post-processed solution as

$$u_h^*(x) = \sum_{m=-k'}^{k'} \sum_{l=0}^{k} u_{j+m}^{(l)}(T)\, C(m,l,k,x), \tag{2.26}$$

for $x \in I_j$, where I_j is the element of the mesh that the evaluation point lies. In this equation, $k' = \lceil (3k+1)/2 \rceil$, and $C(m,l,k,x)$ is a polynomial of degree $2k+1$ depending on the evaluation point, x. We note that $u_{j+m}^{(l)}(T)$ are the coefficients in the LDG approximation at the final time. The post-processing coefficients, $C(m,l,k,x)$, of the post-processing matrix are given by

$$C(m,l,k,x) = \frac{1}{h} \sum_{\gamma=-k}^{k} c_\gamma^{2k+1,k+1} \int_{I_{j+m}} \psi^{(k+1)}\left(\frac{y-x}{h} - \gamma\right) \varphi_{j+m}^{(l)}(y)\, dy \tag{2.27}$$

where $\varphi_{j+m}^{(l)}$ represents the basis of the approximation that are contained in V_h. As mentioned previously, we can choose to evaluate the post-processed solution at specific points within an element, such as the Gauss-Legendre points. Then the post-processing matrix values are the same for every element and only need to be computed once. We note that this post-processing operation can be computed in $\mathcal{O}(N)$ operations as given in [10]. See [15, 17, 10] for efficient computation in two-dimension.

3. Theoretical error estimates

In this section, we show that for a given time T, the approximate solution, $u_h(T)$, converges with higher order in the L^2-norm when convolved with a specially designed kernel.

Theorem 3.1. *Let u_h be the approximate solution to the linear convection-diffusion equation* (2.10) *given by the LDG method* (2.13). *Let $K_h^{2k+1,k+1}$ be a convolution kernel consisting of 2k+1 B-splines of order k+1 such that it reproduces polynomials of degree 2k* (2.21). *Assuming that the initial data, u_o, is smooth enough, we then obtain the error estimate*

$$\|u - u_h^\star\|_\Omega \leq \mathsf{C}(u_o,T)\, h^{2k+m}, \tag{3.1}$$

where $u_h^\star = K_h^{2k+1,k+1} \star u_h$ and $m = 0, \frac{1}{2}$ or 1.

We accomplish this by showing higher-order convergence in the negative-order norm. Then, as noted in [7], because we are dealing with a linear equation, we then have

$$\|\partial_h^\alpha(u-u_h)\|_{-\ell,\Omega} \le C\|\partial_h^\alpha u_0\|_{\ell,\Omega}\, h^{2k+m}. \tag{3.2}$$

Therefore, as long as this increased accuracy can be demonstrated in the negative-order norm, then we have $\mathcal{O}(h^{2k+m})$ accuracy for the post-processed solution in the L^2-norm. This is shown by combining Theorem 2.5 and as well as the proof of higher-order convergence in the negative-order norm, which then allows this convolution kernel to extract the hidden accuracy in the LDG solution.

3.1. **Negative order norm estimates in multi-dimensions.**

In this section we will give the detailed proof for:

Theorem 3.2. *Let u_h be the approximate solution to the linear convection-diffusion equation* (2.10) *given by the LDG method* (2.13)*. Assuming that the initial data, u_o, is smooth enough, we then obtain the error estimate in negative-order norm*

$$\|u-u_h\|_{-(k+1),\Omega} \le \mathsf{C}(u_o,T)\, h^{2k+m} \tag{3.3}$$

where $\mathsf{C}(u_o,T)$ depends on the $\|u_o\|_{k+1,\Omega}$ and T, independent of h, and $m=0,\frac{1}{2}$ or 1, depending on the choice of numerical fluxes.

The main idea behind the proof is to use the dual argument. This approach is clear when we consider the definition of the negative-order norm. That is, given that $\ell>0$, we wish to estimate

$$\|u(T)-u_h(T)\|_{-\ell,\Omega} = \sup_{\Phi\in C_0^\infty(\Omega)} \frac{(u(T)-u_h(T),\Phi)_\Omega}{\|\Phi\|_{\ell,\Omega}}, \tag{3.4}$$

for the result in Theorem 3.2 to be valid. We follow the analysis in [7].

Our dual equation is defined as: find a function φ such that $\varphi(\cdot,t)$ is one-periodic for all $t\in[0,T]$ and

$$\varphi_t + \sum_{i=1}^{d} a_i\varphi_{x_i} - a_o\varphi + \epsilon\Delta\varphi = 0, \quad \Omega\times[0,T) \tag{3.5}$$

$$\varphi(\boldsymbol{x},T) = \Phi(\boldsymbol{x}). \tag{3.6}$$

Note that if we multiply (2.10) by φ and (3.5) by u, we have

$$\frac{d}{dt}(u,\varphi)_\Omega = 0. \tag{3.7}$$

This relation allows us to estimate the term $(u(T)-u_h(T),\Phi)_\Omega$ appearing in the definition of the negative norm (3.4). That is,

$$(u(T)-u_h(T),\Phi)_\Omega = (u(0)-u_h(0),\varphi(0))_\Omega - \int_0^T((u_h)_t,\varphi)_\Omega dt - \int_0^T(u_h,\varphi_t)_\Omega dt.$$

We can rewrite this following [7] by adding and subtracting the piecewise polynomial function $\chi\in V_h$. We use this along with the continuity of φ, the periodic boundary conditions, definition of the dual equation and the method in order to write

$$\begin{aligned} &((u_h)_t,\varphi)_\Omega + (u_h,\varphi_t)_\Omega \\ &\quad = ((u_h)_t,\varphi-\chi)_\Omega + B_1(\boldsymbol{q}_h,u_h;\varphi-\chi) + (u_h,\varphi_t)_\Omega - B_1(\boldsymbol{q}_h,u_h;\varphi). \end{aligned} \tag{3.8}$$

Let us consider more closely the term $(u_h, \varphi_t)_\Omega - B_1(\boldsymbol{q}_h, u_h; \varphi)$, which is corresponding to the second-order derivative term and is the main different part comparing to the derivation in [7]. Using the definition of B_1 and the continuity of φ and the periodic nature of the boundary conditions, the modified expression for $B_1(\boldsymbol{q}_h, u_h; \varphi)$ is then

$$\begin{aligned} B_1(\boldsymbol{q}_h, u_h; \varphi) &= -\sum_{i=1}^{d}(u_h, a_i\varphi_{x_i})_\Omega + (a_o u_h, \varphi)_\Omega + (\boldsymbol{q}_h, \mu\nabla\varphi)_\Omega \\ &= (\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega + (\boldsymbol{q}_h, \mu\Pi(\nabla\varphi))_\Omega \\ &\quad - \sum_{i=1}^{d}(u_h, a_i\varphi_{x_i})_\Omega + (a_o u_h, \varphi)_\Omega, \end{aligned}$$

where $\Pi(\nabla\varphi)$ is the L^2-projection of $\nabla\varphi$ onto the piecewise polynomial space Σ_h. Next, we use the equation for $\boldsymbol{q}_h$, (2.18), and take $\boldsymbol{\phi} = \Pi(\nabla\varphi)$. This gives

$$\begin{aligned} B_1(\boldsymbol{q}_h, u_h; \varphi) =& (\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega - \mu B_2(u_h; \Pi(\nabla\varphi)) \\ & - \sum_{i=1}^{d}(u_h, a_i\varphi_{x_i})_\Omega + (a_o u_h, \varphi)_\Omega, \end{aligned}$$

so that

$$\begin{aligned} (u_h, \varphi_t)_\Omega - B_1(\boldsymbol{q}_h, u_h; \varphi) =& (u_h, \varphi_t + \sum_{i=1}^{d} a_i\varphi_{x_i} - a_o\varphi)_\Omega \\ & - (\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega + \mu B_2(u_h; \Pi(\nabla\varphi)). \end{aligned}$$

Recall the definition of the dual equation (3.5)

$$\varphi_t + \sum_{i=1}^{d} a_i\varphi_{x_i} - a_o\varphi = -\epsilon\Delta\varphi.$$

Using this, combined with the above, we have

$$\begin{aligned} (u_h, \varphi_t)_\Omega - B_1(\boldsymbol{q}_h, u_h; \varphi) =& - (u_h, \epsilon\Delta\varphi)_\Omega + \mu B_2(u_h; \nabla\varphi) \\ & - (\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega - \mu B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi)). \end{aligned}$$

We now use the definition of B_2 to obtain

$$-(u_h, \epsilon\Delta\varphi)_\Omega + \mu B_2(u_h; \nabla\varphi) = -\sum_K \int_{\partial K} \epsilon\widehat{u_h}\boldsymbol{\nu}\cdot \nabla\varphi \, ds = 0,$$

where the last equality stems from the continuity of $\nabla\varphi$ and the periodic boundary conditions of φ. We remind the reader that the numerical fluxes are single valued and are given by equation (2.14) together with (2.15) or (2.16). Because we assume that we have periodic boundary conditions, the boundary fluxes will also cancel. Finally, (3.8) becomes

$$\begin{aligned} ((u_h)_t, \varphi)_\Omega + (u_h, \varphi_t)_\Omega =& ((u_h)_t, \varphi - \chi)_\Omega + B_1(\boldsymbol{q}_h, u_h; \varphi - \chi) \\ & - (\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega - \mu B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi)). \end{aligned}$$

Therefore we have the estimate

$$\begin{aligned}
&(u(T) - u_h(T), \Phi)_\Omega \\
&= (u(0) - u_h(0), \varphi(0))_\Omega - \int_0^T ((u_h)_t, \varphi)_\Omega dt - \int_0^T (u_h, \varphi_t)_\Omega dt \\
&= (u(0) - u_h(0), \varphi(0))_\Omega - \int_0^T \big(((u_h)_t, \varphi - \chi)_\Omega + B_1(\boldsymbol{q}_h, u_h; \varphi - \chi)\big) dt \\
&\quad + \int_0^T \big((\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega + \mu B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi))\big) dt \\
&= \Theta_1 + \Theta_2 + \Theta_3,
\end{aligned}$$

where

$$\begin{aligned}
\Theta_1 &= (u(0) - u_h(0), \varphi(0))_\Omega, \\
\Theta_2 &= -\int_0^T \big(((u_h)_t, \varphi - \chi)_\Omega + B_1(\boldsymbol{q}_h, u_h; \varphi - \chi)\big) dt, \\
\Theta_3 &= \int_0^T \big((\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega + \mu B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi))\big) dt.
\end{aligned}$$

We now give separate estimates for Θ_1, Θ_2, and Θ_3.

Lemma 3.3 (Estimating the first term: projection)**.** *There exists a positive constant C_1, independent of h, such that*

$$|\Theta_1| \le C_1 h^{2k+2} \|u_0\|_{k+1} \|\varphi(0)\|_{k+1}. \tag{3.9}$$

Proof. In practice we usually choose $u_h(\boldsymbol{x}, 0) = Pu_0(\boldsymbol{x})$ which is the standard L^2-projection of the initial function. This gives the following for Θ_1:

$$\begin{aligned}
\Theta_1 = (u(0) - u_h(0), \varphi(0))_\Omega &= (u_0 - Pu_0, \varphi(0))_\Omega \\
&= (u_0 - Pu_0, \varphi(0) - P\varphi(0))_\Omega,
\end{aligned}$$

where the last equality is due to the property of the L^2-projection. Using the Cauchy-Schwarz inequality, we then have

$$|\Theta_1| \le \|u_0 - Pu_0\|_\Omega \|\varphi(0) - P\varphi(0)\|_\Omega \le C_1 h^{2k+2} \|u_0\|_{k+1} \|\varphi(0)\|_{k+1}. \qquad \square$$

Lemma 3.4 (Estimating the second term: residual)**.** *There exists a positive constant C_2, independent of h, such that*

$$\begin{aligned}
|\Theta_2| \le C_2 h^k &\left(\left(\int_0^T \|u_h - u\|_\Omega^2 dt \right)^{1/2} \right. \\
&\left. + \left(\int_0^T \|\boldsymbol{q}_h - \boldsymbol{q}\|_\Omega^2 dt \right)^{1/2} \right) \left(\int_0^T \|\varphi\|_{k+1}^2 dt \right)^{1/2}.
\end{aligned} \tag{3.10}$$

Proof. Θ_2 is defined as

$$\Theta_2 = -\int_0^T \left(((u_h)_t, \varphi - \chi)_\Omega + B_1(\boldsymbol{q}_h, u_h; \varphi - \chi) \right) dt.$$

Let $\chi = P\varphi$ and consider the terms inside the integral. We then have that

$$((u_h)_t, \varphi - P\varphi)_\Omega = 0$$

and

$$\begin{aligned}
B_1(\boldsymbol{q}_h, u_h; \varphi - P\varphi) = {} & -\sum_{i=1}^{d}(u_h, a_i(\varphi - P\varphi)_{x_i})_\Omega \\
& + (u_h, a_o(\varphi - P\varphi))_\Omega + (\boldsymbol{q}_h, \mu\nabla(\varphi - P\varphi))_\Omega \\
& + \sum_K \left(\sum_{i=1}^{d} \int_{\partial K} a_i \nu_i \widetilde{u_h}(\varphi - P\varphi) ds - \int_{\partial K} \mu \widehat{\boldsymbol{q}_h} \cdot \boldsymbol{\nu}(\varphi - P\varphi) ds \right) \\
= {} & \sum_{i=1}^{d}(a_i(u_h)_{x_i}, (\varphi - P\varphi))_\Omega + (a_o u_h, \varphi - P\varphi)_\Omega - (\mu\nabla\cdot \boldsymbol{q}_h, (\varphi - P\varphi))_\Omega \\
& - \sum_K \left(\sum_{i=1}^{d} \int_{\partial K} a_i \nu_i u_h(\varphi - P\varphi) ds - \int_{\partial K} \mu \boldsymbol{q}_h \cdot \boldsymbol{\nu}(\varphi - P\varphi) ds \right) \\
& + \sum_K \left(\sum_{i=1}^{d} \int_{\partial K} a_i \nu_i \widetilde{u_h}(\varphi - P\varphi) ds - \int_{\partial K} \mu \widehat{\boldsymbol{q}_h} \cdot \boldsymbol{\nu}(\varphi - P\varphi) ds \right) \\
= {} & \sum_K \left(\sum_{i=1}^{d} \int_{\partial K} a_i \nu_i (\widetilde{u_h} - u_h)(\varphi - P\varphi) ds - \int_{\partial K} \mu (\widehat{\boldsymbol{q}_h} - \boldsymbol{q}_h) \cdot \boldsymbol{\nu}(\varphi - P\varphi) ds \right) \\
= {} & \sum_K \sum_{i=1}^{d} \int_{\partial K} a_i \nu_i (\widetilde{u_h} - u + u - u_h)(\varphi - P\varphi) ds \\
& - \sum_K \int_{\partial K} \mu (\widehat{\boldsymbol{q}_h} - \boldsymbol{q} + \boldsymbol{q} - \boldsymbol{q}_h) \cdot \boldsymbol{\nu}(\varphi - P\varphi) ds.
\end{aligned}$$

Using an inverse inequality, we can estimate Θ_2 by

$$\begin{aligned}
|\Theta_2| & \le h^{-1} \int_0^T \hat{C}_2 \|u_h - u\|_\Omega \|\varphi - P\varphi\|_\Omega dt + h^{-1} \int_0^T \mu \|\boldsymbol{q}_h - \boldsymbol{q}\|_\Omega \|\varphi - P\varphi\|_\Omega dt \\
& \le C_2\, h^k \left(\left(\int_0^T \|u_h - u\|_\Omega^2 dt \right)^{1/2} + \left(\int_0^T \|\boldsymbol{q}_h - \boldsymbol{q}\|_\Omega^2 dt \right)^{1/2} \right) \left(\int_0^T \|\varphi\|_{k+1}^2 dt \right)^{1/2}.
\end{aligned}$$

□

Lemma 3.5 (Estimating the third term: consistency)**.** *There exists a positive constant* C_3, *independent of* h, *such that*

$$|\Theta_3| \le C_3\, h^k \left(\int_0^T \|u_h - u\|_\Omega^2 dt \right)^{1/2} \left(\int_0^T \|\varphi\|_{k+2}^2 dt \right)^{1/2}. \tag{3.11}$$

Proof. Recall that Θ_3 is given by

$$\Theta_3 = \int_0^T \left((\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega + \mu B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi)) \right) dt.$$

Using the property of the L^2-projection, we know that $(\boldsymbol{q}_h, \mu(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega = 0$ so that we are left with the part of the expression containing B_2. Applying the definition of B_2, we have

$$B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi)) = (u_h, \nabla\cdot(\nabla\varphi - \Pi(\nabla\varphi)))_\Omega - \sum_K \int_{\partial K} \widehat{u_h}\boldsymbol{\nu}\cdot(\nabla\varphi - \Pi(\nabla\varphi))ds$$

$$= -(\nabla u_h, (\nabla\varphi - \Pi(\nabla\varphi)))_\Omega + \sum_K \int_{\partial K} (u_h - \widehat{u_h})\boldsymbol{\nu}\cdot(\nabla\varphi - \Pi(\nabla\varphi))ds$$

$$= \sum_K \int_{\partial K} (u_h - u + u - \widehat{u_h})\boldsymbol{\nu}\cdot(\nabla\varphi - \Pi(\nabla\varphi))ds,$$

which gives the estimate

$$\begin{aligned}&|B_2(u_h; \nabla\varphi - \Pi(\nabla\varphi))|\\ &\quad \le C_a h^{-1/2}\|u_h - u\|_\Omega C_b h^{k+1/2}\|\nabla\varphi\|_{k+1} \le C_3 h^k \|u_h - u\|_\Omega \|\varphi\|_{k+2}.\end{aligned}$$

Therefore, we have

$$|\Theta_3| \le C_3\, h^k \left(\int_0^T \|u_h - u\|_\Omega^2 dt\right)^{1/2} \left(\int_0^T \|\varphi\|_{k+2}^2 dt\right)^{1/2}. \qquad \square$$

Combining Lemmas 3.3, 3.4, 3.5, we can now estimate the numerator used in the negative-order norm:

$$\begin{aligned}&(u(T) - u_h(T), \Phi)_\Omega = \Theta_1 + \Theta_2 + \Theta_3\\ &\le C_1 h^{2k+2}\|u_0\|_{k+1}\|\varphi(0)\|_{k+1} + C_3 h^k \left(\int_0^T \|u_h - u\|_\Omega^2 dt\right)^{1/2} \left(\int_0^T \|\varphi\|_{k+2}^2 dt\right)^{1/2}\\ &+ C_2 h^k \left(\left(\int_0^T \|u_h - u\|_\Omega^2 dt\right)^{1/2} + \left(\int_0^T \|\boldsymbol{q}_h - \boldsymbol{q}\|_\Omega^2 dt\right)^{1/2}\right) \left(\int_0^T \|\varphi\|_{k+1}^2 dt\right)^{1/2}\\ &\le C_1 h^{2k+2}\|u_0\|_{k+1}\|\Phi\|_{k+1} + C_3 h^{2k+m} \left(\int_0^T \|\varphi\|_{k+2}^2 dt\right)^{1/2}\\ &+ C_2 h^{2k+m} \left(\int_0^T \|\varphi\|_{k+1}^2 dt\right)^{1/2}.\end{aligned}$$

Notice that for the second and third terms, the convergence depends on the fluxes (cf. Lemma 2.2). Therefore, after using Lemma 2.1 we have the estimate for the negative norm given by

$$\|u(T) - u_h(T)\|_{-(k+1),\Omega} = \sup_{\Phi \in C_0^\infty(I)} \frac{(u(T) - u_h(T), \Phi)_\Omega}{\|\Phi\|_{k+1,I}} \le Ch^s,$$

where $s = \min(2k+2, 2k+m)$. For our purposes, we have chosen an alternating flux with $m = 1$. This same argument for the divided differences of the error, $(\partial_h^\alpha(u(T) - u_h(T)), \Phi)_\Omega$, can be repeated to obtain the same convergence result.

4. Numerical studies

In this section, we present numerical results confirming that we can indeed improve on the convergence rate of the DG solution from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+1})$ for the linear convection-diffusion equation. We consider both convection-dominated and diffusion-dominated flows in one- and two-dimensions. All the examples are calculated by using the LDG method with an alternating flux with $m = 1$ as in [8] on a uniform (quadrilateral) mesh. We use an L^2-projection of the initial conditions and calculate our errors using a six-point Gaussian quadrature. We use the standard third-order SSP Runge-Kutta scheme [11] and take the time-step such that the spatial errors dominate. In each example, we can clearly see that we get at least $\mathcal{O}(h^{2k+1})$ after post-processing. We note that the usual convection-diffusion equation in the one-dimensional case was presented in [7] showing the expected improvement to $2k+1$ order of accuracy and therefore we neglect this example below. The superconvergence of the pure convection equation is well documented [7, 15] and therefore we neglect to present examples in this paper.

We note that the theoretical estimates presented hold for a uniform mesh assumption. This gives the translation invariance of the post-processing kernel, and this kernel only relies on (nearest) neighboring information. However, in [10] it was demonstrated that higher-order information can also be extracted when the mesh is nonuniform and the nonuniformity is given by an analytic function. This would require re-evaluating the kernel for each post-processing evaluation point, making the computational complexity of the post-processing step increase. However, we note that the application of the post-processor is done only at the final time and therefore the computational costs are neglegiable compared with calculation of the LDG solution.

Example 4.1. We begin by presenting the linear scalar heat equation with a smooth solution on the domain $\Omega = [0, 2\pi]$:

$$u_t = u_{xx}, \quad \Omega \times (0, T], \quad u(x, 0) = \sin(x), \quad x \in \Omega \tag{4.1}$$

with periodic boundary conditions. The errors are presented in Table 4.1 and Figure 4.1 and are computed at time $T = 2$.

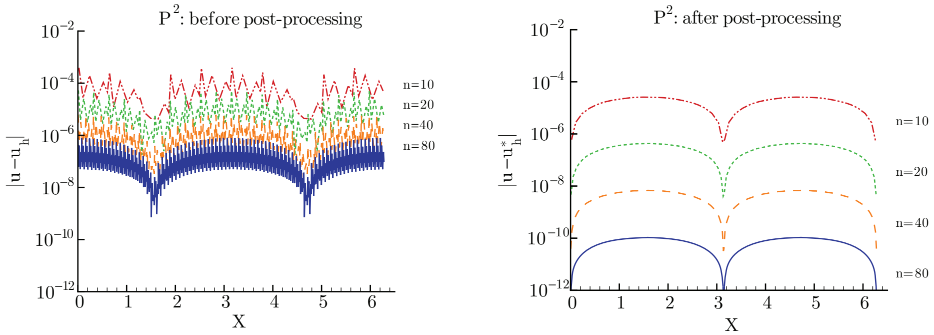

FIGURE 4.1. Plot of pointwise errors in log scale before (left) and after post-processing (right) for the linear heat equation solved at time $T = 2$ using the LDG method. The SIAC filter reduces the oscillations in the LDG solution and improves the smoothness and accuracy.

In Table 4.1, we can clearly see that for the linear heat equation we can improve on the LDG scheme from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+2})$ may be due to the superconvergence of the scheme for the heat equation. The theory only guarantees $\mathcal{O}(h^{2k+1})$ superconvergence, however, we speculate for diffusion-dominated flows the order of

Table 4.1. L^2- and L^∞-errors before and after post-processing for the linear heat equation solved at time $T = 2$ using the LDG method. Although the theory only guarantees $\mathcal{O}(h^{2k+1})$ convergence in the L^2-norm, we clearly see $\mathcal{O}(h^{2k+2})$ in both the L^2- and L^∞-norms.

	Before post-processing				After post-processing			
Mesh	L^2-error	order	L^∞-error	order	L^2-error	order	L^∞-error	order
$\mathcal{P}^1$								
10	5.80E-003	–	7.69E-003	–	1.61E-004	–	2.62E-004	–
20	1.44E-003	2.01	1.92E-003	2.00	1.03E-005	3.96	1.66E-005	3.97
40	3.60E-004	2.00	4.83E-004	1.99	6.50E-007	3.99	1.05E-006	3.99
80	9.01E-005	2.00	1.21E-004	2.00	4.07E-008	4.00	6.60E-008	4.00
160	2.25E-005	2.00	3.02E-005	2.00	2.55E-009	4.00	4.13E-009	4.00
$\mathcal{P}^2$								
10	2.91E-004	–	3.89E-004	–	1.83E-005	–	2.58E-005	–
20	3.63E-005	3.00	4.97E-005	2.97	3.02E-007	5.92	4.30E-007	5.91
40	4.54E-006	3.00	6.25E-006	2.99	4.79E-009	5.98	6.81E-009	5.98
80	5.67E-007	3.00	7.82E-007	3.00	7.49E-011	6.00	1.07E-010	6.00
160	7.09E-008	3.00	9.78E-008	3.00	1.17E-012	6.00	1.67E-012	6.00
$\mathcal{P}^3$								
10	1.12E-005	–	1.31E-005	–	2.18E-006	–	3.08E-006	–
20	7.01E-007	4.00	8.16E-007	4.01	9.31E-009	7.87	1.32E-008	7.87
40	4.38E-008	4.00	5.14E-008	3.99	3.72E-011	7.97	5.26E-011	7.97
80	2.74E-009	4.00	3.22E-009	4.00	1.46E-013	7.99	2.07E-013	7.99
160	1.71E-010	4.00	2.08E-010	3.95	5.71E-016	8.00	8.08E-016	8.00

convergence will be closer to $2k+2$. Additionally, we see significant improvement in the magnitude of the errors. Plots of the pointwise errors are given in Figure 4.1. The SIAC filter clearly works to rid the LDG errors of oscillations and improve the order of accuracy.

Example 4.2. We next present a linear scalar convection-diffusion equation with convection-dominated flow. We assume we have a smooth solution on the domain $\Omega = [0, 2\pi]$. The equation we consider is

$$u_t + u_x = 0.01u_{xx}, \quad \Omega \times (0, T], \quad u(x, 0) = \sin(x), \quad x \in \Omega, \tag{4.2}$$

with periodic boundary conditions. The errors again are computed at $T = 2$ and the results are presented in Table 4.2 and Figure 4.2.

In this example we detect the effect after post-processing if we choose a larger convection coefficient and a smaller diffusion coefficient. We notice that the results are very similar to the linear wave equation, where we obtain improvement to order $2k+1$ after post-processing.

Example 4.3. The last of the one-dimensional examples that we present is the nonlinear viscous Burgers equation with forcing function on the domain $\Omega = [0, 2\pi]$:

$$u_t + \left(\frac{u^2}{2}\right)_x = \epsilon u_{xx} + f(x, t), \quad \Omega \times (0, T], \quad u(x, 0) = \sin(x), \quad x \in \Omega. \tag{4.3}$$

TABLE 4.2. L^2- and L^∞-errors before and after post-processing for the linear convection-diffusion equation with small diffusion coefficient solved at time $T = 2$ using the LDG method. We clearly see improvement to $\mathcal{O}(h^{2k+1})$ after post-processing.

	Before post-processing				After post-processing			
Mesh	L^2-error	order	L^∞-error	order	L^2-error	order	L^∞-error	order
$\mathcal{P}^1$								
10	4.18E-002	–	4.72E-002	–	5.79E-003	–	8.42E-003	–
20	1.05E-002	2.00	1.30E-002	1.85	6.68E-004	3.11	9.54E-004	3.14
40	2.61E-003	2.00	3.39E-003	1.95	7.95E-005	3.07	1.13E-004	3.07
80	6.53E-004	2.00	8.62E-004	1.97	9.65E-006	3.04	1.37E-005	3.04
160	1.63E-004	2.00	2.17E-004	1.99	1.18E-006	3.02	1.68E-006	3.03
$\mathcal{P}^2$								
10	2.09E-003	–	2.77E-003	–	1.50E-004	–	2.11E-004	–
20	2.62E-004	3.00	3.60E-004	2.94	2.76E-006	5.76	3.92E-006	5.75
40	3.28E-005	3.00	4.53E-005	2.99	5.29E-008	5.71	7.50E-008	5.71
80	4.11E-006	3.00	5.67E-006	3.00	1.12E-009	5.56	1.59E-009	5.56
160	5.13E-007	3.00	7.09E-007	3.00	2.65E-011	5.40	3.75E-011	5.40
$\mathcal{P}^3$								
10	8.11E-005	–	9.41E-005	–	1.58E-005	–	2.24E-005	–
20	5.08E-006	4.00	5.95E-006	3.98	6.77E-008	7.87	9.58E-008	7.87
40	3.17E-007	4.00	3.73E-007	4.00	2.72E-010	7.96	3.84E-010	7.96
80	1.98E-008	4.00	2.33E-008	4.00	1.08E-012	7.98	1.52E-012	7.98
160	1.25E-009	4.00	1.46E-009	3.95	4.27E-015	7.98	6.03E-015	7.98

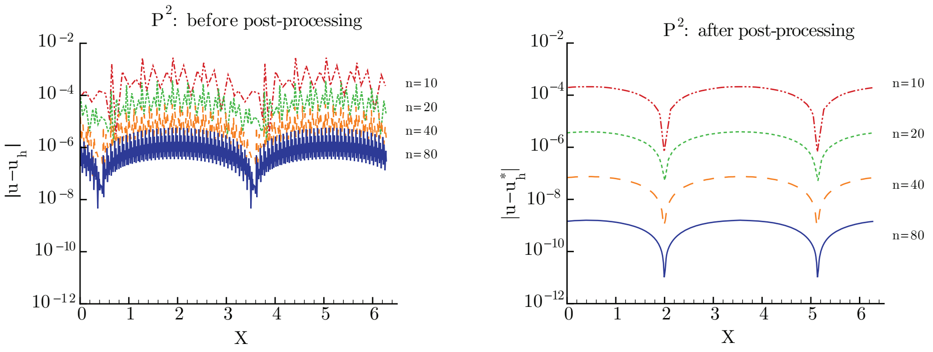

FIGURE 4.2. Plot of pointwise errors in log scale before (left) and after post-processing (right) for the linear convection-diffusion solved at time $T = 2$ using the LDG method. The SIAC filter reduces the oscillations in the LDG solution and improves the smoothness and accuracy.

Here we take $\epsilon = 1$ and $f(x, t) = 0.5 \sin(2x)e^{-2\epsilon t}$ with boundary conditions that are periodic. The exact solution for this equation is $u(x, t) = \sin(x)e^{-\epsilon t}$. We present the errors computed at final time $T = 2$ in Table 4.3 and Figure 4.3.

TABLE 4.3. L^2- and L^∞-errors before and after post-processing for the nonlinear viscous Burgers equation with forcing function solved at time $T = 2$ using the LDG method. We clearly see improvement to $\mathcal{O}(h^{2k+1})$ after post-processing.

	Before post-processing				After post-processing			
Mesh	L^2-error	order	L^∞-error	order	L^2-error	order	L^∞-error	order
$\mathcal{P}^1$								
10	2.30E-003	–	7.32E-003	–	4.33E-004	–	6.48E-004	–
20	5.74E-004	2.00	1.87E-003	1.97	4.52E-005	3.26	6.63E-005	3.29
40	1.43E-004	2.00	4.76E-004	1.97	5.00E-006	3.18	7.23E-006	3.20
80	3.59E-005	2.00	1.20E-004	1.99	5.85E-007	3.10	8.38E-007	3.11
160	8.98E-006	2.00	3.01E-005	2.00	7.07E-008	3.05	1.01E-007	3.06
$\mathcal{P}^2$								
10	1.15E-004	–	3.85E-004	–	1.93E-005	–	2.73E-005	–
20	1.44E-005	3.00	4.95E-005	2.96	3.36E-007	5.84	4.77E-007	5.84
40	1.81E-006	3.00	6.23E-006	2.99	5.85E-009	5.84	8.32E-009	5.84
80	2.26E-007	3.00	7.81E-007	3.00	1.09E-010	5.75	1.54E-010	5.75
160	2.83E-008	3.00	9.78E-008	3.00	2.22E-012	5.61	3.15E-012	5.61
$\mathcal{P}^3$								
10	4.45E-006	–	1.31E-005	–	2.18E-006	–	3.08E-006	–
20	2.79E-007	4.00	8.16E-007	4.01	9.32E-009	7.87	1.32E-008	7.87
40	1.75E-008	4.00	5.14E-008	3.99	3.73E-011	7.97	5.27E-011	7.97
80	1.09E-009	4.00	3.22E-009	4.00	1.47E-013	7.99	2.08E-013	7.99
160	6.83E-011	4.00	2.01E-010	4.00	5.79E-016	7.99	8.19E-016	7.99

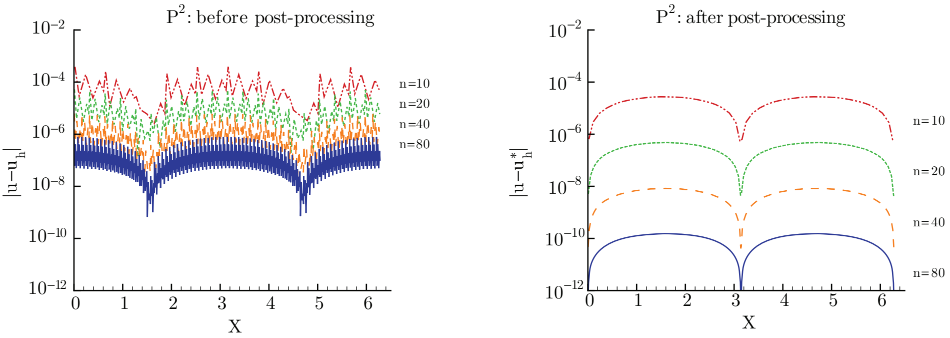

FIGURE 4.3. Plot of pointwise errors in log scale before (left) and after post-processing (right) for the nonlinear viscous Burgers equation with forcing function solved at time $T = 2$ using the LDG method. The SIAC filter reduces the oscillations in the LDG solution and improves the smoothness and accuracy.

This example is calculated by using the LDG method [8]. We note that the theory presented in this paper does not cover this example. However, we can clearly see in Table 4.3 that we improve the errors in both L^2 and L^∞ from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+1})$ after post-processing. In Figure 4.3, we see that the oscillations in the errors of the LDG solutions are reduced and the order is improved.

We now present a series of two-dimensional numerical examples on domain $\Omega = [0, 2\pi] \times [0, 2\pi]$ confirming our theoretical results.

Example 4.4. Table 4.4 presents the L^2- and L^∞-errors before and after post-processing for the two-dimensional linear heat equation,

$$u_t = u_{xx} + u_{yy}, \quad \Omega \times (0, T], \tag{4.4}$$

$$u(x, y, 0) = \sin(x)\sin(y), \quad x \in \Omega, \tag{4.5}$$

with periodic boundary conditions. The errors are computed at time $T = 2$ using the $\mathcal{Q}^k$ polynomial basis.

TABLE 4.4. L^2- and L^∞-errors before and after post-processing for the two-dimensional linear heat equation solved at time $T = 2$ using the LDG method. Although the theory only guarantees $\mathcal{O}(h^{2k+1})$ convergence in the L^2-norm, we clearly see $\mathcal{O}(h^{2k+2})$ in both the L^2- and L^∞-norms.

	Before post-processing				After post-processing			
Mesh	L^2-error	order	L^∞-error	order	L^2-error	order	L^∞-error	order
$\mathcal{Q}^1$								
10×10	3.09E-004	–	2.03E-003	–	4.45E-005	–	9.72E-005	–
20×20	7.77E-005	1.99	5.16E-004	1.97	2.42E-006	4.20	5.42E-006	4.17
40×40	1.94E-005	2.00	1.31E-004	1.98	1.38E-007	4.13	3.13E-007	4.11
80×80	4.86E-006	2.00	3.27E-005	2.00	8.23E-009	4.07	1.87E-008	4.06
$\mathcal{Q}^2$								
10×10	1.56E-005	–	6.31E-005	–	3.55E-006	–	7.11E-006	–
20×20	1.96E-006	2.99	7.52E-006	3.07	5.83E-008	5.93	1.16E-007	5.93
40×40	2.45E-007	3.00	8.94E-007	3.07	9.19E-010	5.99	1.85E-009	5.98
80×80	3.06E-008	3.00	1.09E-007	3.04	1.44E-011	6.00	2.90E-011	6.00
$\mathcal{Q}^3$								
10×10	6.01E-007	–	3.36E-006	–	4.18E-007	–	8.35E-007	–
20×20	3.78E-008	3.99	2.18E-007	3.94	1.78E-009	7.87	3.56E-009	7.87
40×40	2.36E-009	4.00	1.39E-008	3.97	7.12E-012	7.97	1.42E-011	7.97
80×80	1.48E-010	4.00	8.71E-010	4.00	2.80E-014	7.99	5.60E-014	7.99

Although the theory only guarantees us convergence of order $2k+1$, again we clearly see the $\mathcal{O}(h^{2k+2})$ convergence in both the L^2- and L^∞-norms for the two-dimensional linear heat equation after post-processing. Proving the $2k+2$ superconvergence theoretically is currently being in investigated.

Example 4.5. The next two-dimensional example that we present is a convection-dominated example,

$$u_t + u_x + u_y = 0.01u_{xx} + 0.01u_{yy}, \quad \Omega \times (0, T], \tag{4.6}$$

$$u(x, y, 0) = \sin(x)\sin(y), \quad x \in \Omega. \tag{4.7}$$

TABLE 4.5. L^2- and L^∞-errors before and after post-processing for the convection-dominated equation, $u_t + u_x + u_y = 0.01u_{xx} + 0.01u_{yy}$, solved at time $T = 2$ using the $\mathcal{Q}^k$-polynomial approximations. We clearly see improved order of convergence over the LDG solution.

	Before post-processing				After post-processing			
Mesh	L^2-error	order	L^∞-error	order	L^2-error	order	L^∞-error	order
$\mathcal{Q}^1$								
10×10	1.13E-002	–	7.24E-002	–	2.57E-003	–	5.29E-003	–
20×20	3.54E-003	1.68	2.36E-002	1.62	1.56E-004	4.04	3.39E-004	3.97
40×40	9.98E-004	1.83	6.72E-003	1.81	9.34E-006	4.06	2.06E-005	4.04
80×80	2.54E-004	1.97	1.71E-003	1.97	5.64E-007	4.05	1.25E-006	4.04
$\mathcal{Q}^2$								
10×10	7.13E-004	–	1.79E-003	–	1.87E-004	–	3.74E-004	–
20×20	9.44E-005	2.92	3.41E-004	2.39	3.07E-006	5.93	6.14E-006	5.93
40×40	1.27E-005	2.90	4.58E-005	2.90	4.85E-008	5.98	9.74E-008	5.98
80×80	1.60E-006	2.98	5.69E-006	3.01	7.58E-010	6.00	1.53E-009	6.00
$\mathcal{Q}^3$								
10×10	2.63E-005	–	1.14E-004	–	2.19E-005	–	4.38E-005	–
20×20	1.86E-006	3.82	9.73E-006	3.55	9.35E-008	7.87	1.87E-007	7.87
40×40	1.23E-007	3.92	7.13E-007	3.77	3.73E-010	7.97	7.47E-010	7.97
80×80	7.75E-009	3.99	4.55E-008	3.97	1.47E-012	7.99	2.93E-012	7.99

Where again periodic boundary conditions are used and the errors are computed at $T = 2$. The errors for the L^2- and L^∞-norms are presented in Table 4.5 using the $\mathcal{Q}^k$ polynomial approximation. For this convection-dominated example, we clearly see errors of the order $2k+1$, as the theory predicts.

Example 4.6. The last example that we present is when the convection and diffusion coefficients are equal. That is,

$$u_t + u_x + u_y = u_{xx} + u_{yy}, \quad \Omega \times (0, T], \tag{4.8}$$

$$u(x, y, 0) = \sin(x + y), \quad x \in \Omega, \tag{4.9}$$

where periodic boundary conditions are used. The errors for L^2 and L^∞ are computed at $T = 2$ using the $\mathcal{P}^k$ polynomial basis and are presented in Table 4.6. This example is similar to the one-dimensional case. If we denote $z = x + y$, then we can convert the equation to the one-dimensional case. The numerical test confirms this by showing similar $2k + 1$ order error as in [7].

TABLE 4.6. L^2- and L^∞-errors before and after post-processing for $u_t + u_x + u_y = u_{xx} + u_{yy}$, $I \times (0,T)$, solved at time $T = 2$. Errors for two-dimensional $\mathcal{P}^k$ polynomial basis.

	Before post-processing				After post-processing			
Mesh	L^2-error	order	L^∞-error	order	L^2-error	order	L^∞-error	order
				$\mathcal{P}^1$				
10×10	8.33E-004	–	2.31E-003	–	6.74E-004	–	9.54E-004	–
20×20	1.70E-004	2.29	7.50E-004	1.63	8.83E-005	2.93	1.25E-004	2.93
40×40	3.92E-005	2.12	2.09E-004	1.84	1.11E-005	2.99	1.58E-005	2.99
80×80	9.59E-006	2.03	5.48E-005	1.93	1.39E-006	3.00	1.97E-006	3.00
				$\mathcal{P}^2$				
10×10	6.46E-005	–	4.99E-004	–	1.43E-005	–	2.02E-005	–
20×20	8.35E-006	2.95	6.58E-005	2.92	3.88E-007	5.21	5.49E-007	5.21
40×40	1.07E-006	2.97	8.49E-006	2.95	1.10E-008	5.14	1.56E-008	5.14
80×80	1.35E-007	2.98	1.08E-006	2.98	3.26E-010	5.08	4.62E-010	5.08
				$\mathcal{P}^3$				
10×10	5.17E-006	–	4.78E-005	–	6.85E-007	–	9.69E-007	–
20×20	3.38E-007	3.94	3.22E-006	3.89	3.50E-009	7.61	4.95E-009	7.61
40×40	2.16E-008	3.97	2.09E-007	3.94	2.14E-011	7.36	3.02E-011	7.36
80×80	1.37E-009	3.98	1.33E-008	3.98	1.85E-013	6.85	2.61E-013	6.85

5. Concluding remarks

By implementing a smoothness-increasing accuracy-conserving filter, we can provably improve the quality of LDG solutions from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+m})$. The proof of these results for the multi-dimensional linear convection-diffusion equations requires investigating the negative-order norm of the LDG method. For a uniform mesh, the kernel then extracts this hidden accuracy out of the LDG solution to obtain order $2k+m$ accuracy in the L^2-norm of the filtered solution. We theoretically demonstrated that we obtain $\mathcal{O}(h^{2k+m})$ in the negative-order norm and numerically confirmed that we do indeed see improved convergence in both the L^2- and L^∞-norms after filtering. These results can easily be extended to include multi-dimensional systems, provided the matrix is diagonalizable.

We note that the $2k+m$ order seen in the negative-order norm is an inherent property for the discontinuous Galerkin solution. It does not rely on the mesh assumption and the special kernel that we use. However, the proofs of the accuracy extracting capabilities rely heavily on the translation invariance of the mesh. We speculate that in order to fully see the $2k+1$ order accuracy on unstructured triangular meshes, the convolution kernel must be redesigned. Additionally, the proofs of the higher order accuracy for the negative-order norm do not rely on using the $\mathcal{Q}^k$ polynomial space, however the applicability of the kernel does. These related issues remain a challenging open problem.

Furthermore, in this paper we only consider periodic boundary conditions. For other types of boundary conditions, it is possible to estimate the negative-order norm which captures the superconvergent points of the numerical methods. In order to post-process this type of solution we would need the one-sided kernel in [14, 16].

Last, we point out that our numerical experiments indicate that the post-processing technique can be applied to nonlinear convection-diffusion equations. This is the subject of current research.

References

1. S. Adjerid and A. Klauser, *Superconvergence of discontinuous finite element solutions for transient convection-diffusion problems*, Journal on Scientific Computing, **22-23** (2005), pp. 5-24. MR2142188 (2006a:65120)
2. J.H. Bramble and A.H. Schatz, *Higher order local accuracy by averaging in the finite element method*, Mathematics of Computation, **31** (1977), pp. 94-111. MR0431744 (55:4739)
3. F.Celiker and B. Cockburn, *Superconvergence of the numerical traces of discontinuous Galerkin and Hybridized methods for convection-diffusion problems in one space dimension*, Mathematics of Computation, **76** (2007), pp. 67-96. MR2261012 (2008e:65225)
4. Y. Cheng and C.-W. Shu, *Superconvergence of discontinuous Galerkin and local discontinuous Galerkin schemes for linear hyperbolic and convection-diffusion equations in one space dimension*, SIAM Journal on Numerical Analysis, **47** (2010), pp. 4044-4072. MR2585178 (2011e:65187)
5. P. Ciarlet, *The finite element method for elliptic problem*, North Holland, 1975. MR0520174 (58:25001)
6. B. Cockburn, *Discontinuous Galerkin methods for methods for convection-dominated problems*, in *High-order methods for computational physics*, T.J. Barth and H. Deconinck, editors, Lecture Notes in Computational Science and Engineering, volume 9, Springer, 1999, pp. 69-224. MR1712278 (2000f:76095)
7. B. Cockburn, M. Luskin, C.-W. Shu, E. Süli, *Enhanced accuracy by post-processing for finite element methods for hyperbolic equations*, Mathematics of Computation, **72** (2003), pp. 577-606. MR1954957 (2004g:65129)
8. B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM Journal on Numerical Analysis, **35** (1998), pp. 2440-2463. MR1655854 (99j:65163)
9. B. Cockburn and C.-W. Shu, *Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems*, Journal on Scientific Computing, **16** (2001), pp. 173-261. MR1873283 (2002i:65099)
10. S. Curtis, R. M. Kirby, J. K. Ryan and C.-W. Shu, *Postprocessing for the discontinuous Galerkin method over nonuniform meshes*, SIAM Journal on Scientific Computing, **30** (2007), pp. 272-289. MR2377442 (2009a:65249)
11. S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability preserving high-order time discretization methods*, SIAM Review, **43** (2001), pp. 89–112. MR1854647 (2002f:65132)
12. M.S. Mock and P.D. Lax, *The computation of discontinuous solutions of linear hyperbolic equations*, Communications on Pure and Applied Mathematics, **18** (1978), pp. 423-430. MR0468216 (57:8054)
13. J.K. Ryan and B. Cockburn, *Local Derivative Post-processing for the discontinuous Galerkin method*, Journal of Computational Physics, **228** (2009), pp. 8642-8664. MR2558770 (2010j:65192)
14. J.K. Ryan and C.-W. Shu, *One-sided post-processing for the discontinuous Galerkin methods*, Methods and Applications of Analysis, **10** (2003), pp. 295-307. MR2074753
15. J.K. Ryan, C.-W. Shu, and H. Atkins, *Extension of a post-processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem*, SIAM Journal on Scientific Computing, **26** (2005), pp. 821-843. MR2126114 (2005m:65222)
16. P. van Slingerland, J.K. Ryan, and C.W. Vuik, *Position-Dependent Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering for Accuracy for Improving discontinuous Galerkin solutions*, SIAM Journal on Scientific Computing, **33** 2011, pp. 802-825
17. M. Steffan, S. Curtis, R.M. Kirby, and J.K. Ryan, *Investigation of smoothness enhancing accuracy-conserving filters for improving streamline integration through discontinuous fields*, IEEE-TVCG, **14** (2008), pp. 680-692.
18. V. Thomée, *High order local approximations to derivatives in the finite element method*, Mathematics of Computation, **31** (1977), pp. 652-660. MR0438664 (55:11572)

19. Y. Xu and C.-W. Shu, *Error estimates of the semi-discrete local discontinuous Galerkin method for nonlinear convection-diffusion and KdV equations*, Computer Methods in Applied Mechanics and Engineering, **196** (2007), pp. 3805-3822. MR2340006 (2009e:65139)
20. Y. Xu and C.-W. Shu, *Local discontinuous Galerkin methods for high-order time-dependent partial differential equations*, Communications in Computational Physics, **7** (2010), pp. 1-46. MR2673127 (2011g:65204)
21. J. Yan and C.-W. Shu, *A local discontinuous Galerkin method for KdV type equations*, SIAM Journal on Numerical Analysis, **40** (2002), pp. 769-791. MR1921677 (2003e:65181)
22. Q. Zhang and C.-W. Shu, *Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin methods for scalar conservation laws*, SIAM Journal on Numerical Analysis, **42** (2004), pp. 641-666. MR2084230 (2005h:65149)
23. Z. Zhang, Z. Xie and Z. Zhang, *Superconvergence of discontinuous Galerkin methods for convection-diffusion problems*, Journal on Scientific Computing, **41** (2009), pp. 70-93. MR2540105 (2010k:65141)

Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands.

Current address: Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, P.R. China.

E-mail address: jlyue@mail.ustc.edu.cn

Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China

E-mail address: yxu@ustc.edu.cn

Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands

E-mail address: J.K.Ryan@tudelft.nl