Journal of Computational and Applied Mathematics 294 (2016) 275–296

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Smoothness-Increasing Accuracy-Conserving (SIAC) filters for derivative approximations of discontinuous Galerkin (DG) solutions over nonuniform meshes and near boundaries

X. Li [a], J.K. Ryan [b,*], R.M. Kirby [c], C. Vuik [a]

[a] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, Netherlands
[b] School of Mathematics, University of East Anglia, Norwich, UK
[c] School of Computing, University of Utah, Salt Lake City, Utah, USA

ARTICLE INFO

Article history:
Received 15 December 2014
Received in revised form 10 June 2015

Keywords:
Discontinuous Galerkin method
Post-processing
SIAC filtering
Superconvergence
Nonuniform meshes
Boundaries

ABSTRACT

Accurate approximations for the derivatives are usually required in many application areas such as biomechanics, chemistry and visualization applications. With the help of Smoothness-Increasing Accuracy-Conserving (SIAC) filtering, one can enhance the derivatives of a discontinuous Galerkin solution. However, current investigations of derivative filtering are limited to uniform meshes and periodic boundary conditions, which do not meet practical requirements. The purpose of this paper is twofold: to extend derivative filtering to nonuniform meshes and propose position-dependent derivative filters to handle filtering near the boundaries. Through analyzing the error estimates for SIAC filtering, we extend derivative filtering to nonuniform meshes by changing the scaling of the filter. For filtering near boundaries, we discuss the advantages and disadvantages of two existing position-dependent filters and then extend them to position-dependent derivative filters, respectively. Further, we prove that with the position-dependent derivative filters, the filtered solutions can obtain a better accuracy rate compared to the original discontinuous Galerkin approximation with arbitrary derivative orders over nonuniform meshes. One- and two-dimensional numerical results are provided to support the theoretical results and demonstrate that the position-dependent derivative filters, in general, enhance the accuracy of the solution for both uniform and nonuniform meshes.

1. Introduction

In many cases, one can argue persuasively that the changes in values of a function are often more import than the values themselves, such as exhibited by streamline integration of fields. Therefore, an accurate derivative approximation is often required in many areas such as biomechanics, optimization, chemistry and visualization applications. However, computing derivatives of discontinuous Galerkin approximations is challenging because the DG solution only has weak continuity at element boundaries. This means that the strong form of derivatives for a DG solution technically does not hold at element boundaries, and computing the derivative directly does not always produce accurate results. For example, naive and careless use of the derivatives of the discontinuous Galerkin solution directly to streamline integration can produce inconsistent

* Corresponding author.
E-mail addresses: X.Li-2@tudelft.nl (X. Li), Jennifer.Ryan@uea.ac.uk (J.K. Ryan), kirby@cs.utah.edu (R.M. Kirby), C.Vuik@tudelft.nl (C. Vuik).

http://dx.doi.org/10.1016/j.cam.2015.08.011
0377-0427/

results with the exact solution [1]. Once derivatives are needed near the boundaries, the difficulty increases since the solution often has less regularity in those regions.

In order to obtain accurate approximations for the derivatives of discontinuous Galerkin (DG) solutions, this paper focuses on using the so-called Smoothness-Increasing Accuracy-Conserving class of filters. As the name implies, SIAC filtering can increase the smoothness of DG solutions, and this smoothness-increasing property helps to enhance the accuracy of derivatives of DG solutions. Before giving context to what we will propose, we first review the currently existing SIAC and derivative SIAC filters. The source of SIAC filtering is considered to be the superconvergence extraction technique introduced by [2] for finite element solutions of elliptic problems. The extension for DG methods of linear hyperbolic equations given in [3]. In an ideal situation, by applying SIAC filtering, the accuracy order of the filtered DG approximation can improve from $k+1$ to $2k+1$. The concept of the derivative filter was introduced in [4] for finite element methods and [5] for DG methods. With the derivative filter, the filtered solution maintains an accuracy order of $2k+1$ regardless of the derivative order. However, the previous investigations of derivative filtering have two major limitations: the requirement of a uniform meshes and periodic boundary conditions.

The purpose of this paper is to overcome these two limitations. We propose position-dependent derivative filters to approximate the derivatives of the discontinuous Galerkin solution over nonuniform meshes and near boundaries. Our main contributions are:

Nonuniform meshes. Filtering over nonuniform meshes has always been a significant challenge for SIAC filtering since the $2k+1$ accuracy order is no longer guaranteed in general. Most of previous work for nonuniform meshes (such as [6–8]) only considered a particular family of nonuniform meshes, smoothly-varying meshes. Among these works, only [7] mentioned derivatives over nonuniform meshes. It discussed the challenges of derivative filtering over nonuniform meshes and presented preliminary results concerning smooth-varying meshes. In this paper, we propose a method for arbitrary nonuniform meshes: using the scaling $H = h^{\mu}$ for filtering over nonuniform meshes. We cannot guarantee that the derivative filtering can improve the derivatives of DG solutions to accuracy order of $2k+1$, but we prove that a higher convergence rate (compared to DG solution) is still obtained. Further, the numerical examples suggest that the accuracy is improved once the mesh is sufficiently refined.

Boundaries. First, we point out that previously there was no derivative filter that could be used near boundaries except for periodic meshes. Without considering derivatives, there are three existing position-dependent filters that can be used to handle boundary regions, see [8–10]. Two of them, [9,10], are constructed by only using central B-splines. They showed good performance over uniform meshes. The position-dependent filter recently introduced in [8] was aimed at nonuniform meshes. It uses $2k+1$ central B-splines and an extra general B-spline. The results in [8] suggested that adding the extra general B-spline improves the performance of the position-dependent filter over nonuniform meshes compared to using only central B-splines. In this paper, we extend the position-dependent filter [10] (referred to as the SRV filter) and the new position-dependent filter [8] (referred to as the RLKV filter) to position-dependent derivative filters. Then, we discuss the advantages and disadvantages of these position-dependent derivative filters over uniform and nonuniform meshes. For nonuniform meshes, we prove that by using the position-dependent derivative filtering, the convergence rate of the derivatives of the DG solution can be improved. Numerical comparisons over uniform and nonuniform meshes also demonstrate that the derivative filtered solutions are more accurate than the derivatives of DG approximations.

Our new contributions are:

- Testing the position-dependent derivative filters for uniform meshes, which has never been done before;
- Applying the symmetric and position-dependent derivative filters over different nonuniform meshes.

This paper is organized as follows. In Section 2, we first review properties of the discontinuous Galerkin solution and its derivatives. Then, we introduce the symmetric and position-dependent filters. Lastly, we show the symmetric derivative filter over uniform meshes. The symmetric and position-dependent derivative filters over nonuniform meshes are presented in Section 3 with theoretical error estimations. Numerical results over uniform and nonuniform one-dimensional meshes are given in Section 4. Nonuniform two-dimensional quadrilateral meshes are considered in Section 5. We present conclusions in Section 6.

2. Background

In this section, we first briefly review the basic features of discontinuous Galerkin methods and the properties related to derivatives. Then, we present the relevant background of Smoothness-Increasing Accuracy-Conserving filters.

2.1. DG approximation and its derivatives

Consider a multi-dimensional linear hyperbolic equation

$$u_t + \sum_{i=1}^{d} A_i u_{x_i} + A_0 u = 0, \qquad u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad (\mathbf{x}, t) \in \Omega \times (0, T], \quad \Omega \subset \mathbb{R}^d, \tag{2.1}$$

where the initial condition $u_0(\mathbf{x})$ is a sufficiently smooth function and the coefficients A_i are constants. The details of discontinuous Galerkin methods for hyperbolic equations can be found in [11,12]. Here, we skip the details and write the DG approximation directly as

$$u_h(\mathbf{x}, t) = \sum_{|\ell|=0}^{k} u_K^{\ell}(t)\varphi_K^{\ell}(\mathbf{x}), \quad \text{for } x \in K,$$

where k is the highest degree of approximation polynomial, and φ_K^{ℓ} is the multi-dimensional piecewise polynomial basis function of degree ℓ, polynomial inside element K and zero outside K, $\ell = (\ell_1, \ldots, \ell_d)$ is a multi-index. Here we choose scaled Legendre polynomials for simplicity. The mesh elements K can be either uniform or nonuniform, and the mesh size is represented as h_K (h for uniform meshes).

A well-known feature of the DG approximation is that the use of a piecewise polynomial basis leads to superconvergence. However, by using a piecewise polynomial basis, the solution u_h is discontinuous at the interface of the elements, which is the main challenge to defining the global derivatives of the DG approximation. Alternatively, we can define the local derivatives of u_h in the interior of each element by

$$\partial_x^{\alpha} u_h(\mathbf{x}, t) = \sum_{|\ell|=0}^{k} u_K^{\ell}(t)\partial_x^{\alpha}\varphi_K^{\ell}(\mathbf{x}), \quad \text{for } \mathbf{x} \in K, \tag{2.2}$$

where $\alpha = (\alpha_1, \ldots, \alpha_d)$ is an arbitrary multi-index. Eq. (2.2) is an approximation of $\partial_x^{\alpha} u$, but it is only a local derivative, and the convergence rate in Eq. (2.3) is not satisfied. Unlike the general DG approximation, which has the convergence rate of $k+1$, with each successive derivative, one order of accuracy is lost,

$$\left\| \partial_x^{\alpha} u - \partial_x^{\alpha} u_h \right\|_0 \sim \mathcal{O}(h^{k+1-|\alpha|}). \tag{2.3}$$

This means $|\alpha| \leq k$, since the $(k+1)$th derivative is just zero. In order to obtain a global and accurate derivative of the DG approximation, we can increase its smoothness by using SIAC filtering. Since the filtered solutions are more smooth and have higher accuracy order, this filtering technique allows us to obtain derivatives for $|\alpha| > k$. We emphasize that the ability of calculating derivatives for $|\alpha| > k$ is a unique benefit of using SIAC filtering.

2.2. SIAC filter

Smoothness-Increasing Accuracy-Conserving filtering is based on a postprocessing technique first demonstrated by Bramble and Schatz [2] for finite element methods to obtain a better approximation. Later, in [3], this technique was extended to DG methods. In the one-dimensional case, the filtered approximation $u_h^{\star}$ is given by

$$u_h^{\star}(x) = K_h^{(r+1,k+1)} \star u_h(x), \tag{2.4}$$

and

$$\|u - u_h^{\star}\|_0 \leq Ch^{2k+1},$$

where the symmetric filter $K^{(r+1,k+1)}$ is a linear combination of the central B-splines,

$$K^{(r+1,k+1)}(x) = \sum_{\gamma=0}^{r} c_{\gamma}^{(r+1,k+1)} \psi^{(k+1)}\left(x + \frac{r}{2} - \gamma\right) \tag{2.5}$$

and the scaled filter K_h is given by formula $K_h(x) = \frac{1}{h}K(\frac{x}{h})$. In [2,3], the number of B-splines is $r+1 = 2k+1$. The coefficients $c_{\gamma}^{(r+1,k+1)}$ are calculated by requiring the filter to reproduce the polynomials by convolution

$$K^{(r+1,k+1)} \star p = p, \quad p = 1, x, \ldots, x^r.$$

Here, the central B-splines are constructed recursively by

$$\psi^{(1)}(x) = \chi_{[-1/2,1/2)}(x),$$
$$\psi^{(\ell+1)}(x) = \psi^{(1)} \star \psi^{(\ell)}, \quad \ell \geq 1.$$

We note that this symmetric filter is suitable only for the interior region, $(r+k+1)/2$ elements away from the boundaries, since it uses symmetric information around the evaluated point. When filtering near the domain boundaries, we need to use position-dependent filters.

In the multi-dimensional case, the filter is a tensor product of the one-dimensional filters (2.5) and can be written as

$$K_h^{(r+1,k+1)}(\mathbf{x}) = \prod_{i=1}^{d} K_h^{(r+1,k+1)}(x_i), \qquad \mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d,$$

with the scaled filter $K_h^{(r+1,k+1)}(\mathbf{x}) = \frac{1}{h^d} K^{(r+1,k+1)}\left(\frac{\mathbf{x}}{h}\right)$.

2.3. Position-dependent filter

There are three position-dependent filters that have been introduced in previous work. Two of the three, [9,10], use only central B-splines. These central B-spline filters have similar structures. Here we only discuss the one with better performance, [10], referred to as the SRV filter, in this section. The last position-dependent filter use central B-splines, and an extra noncentral B-spline. It was recently introduced in [8], and the error of the filtered solution near the boundary is reduced over nonuniform meshes compared to the SRV filter. In the following context, we refer to position-dependent filter [8] as the RLKV filter.

2.3.1. SRV filter

The SRV filter using $4k+1$ central B-splines was introduced in [10] for uniform meshes. This boundary filter demonstrated better behavior in terms of error than the original position-dependent filter which uses $2k+1$ given by Ryan and Shu in [9]. It changes its support according to the location of the point being filtered. For example, at the left boundary, a translation of the filter should be done so that the support of the filter remains inside the domain (up to the left boundary). The SRV filter for filtering near the boundaries can then be written as

$$K^{(4k+1,k+1)}(x) = \sum_{\gamma=0}^{4k} c_\gamma^{(4k+1,k+1)} \psi^{(k+1)}\left(x - x_\gamma(\bar{x})\right), \tag{2.6}$$

where x_γ depends on the location of the evaluation point $\bar{x}$ and is given by

$$x_\gamma(\bar{x}) = -2k + \gamma + \lambda(\bar{x}),$$

with

$$\lambda(\bar{x}) = \begin{cases} \min\left\{0, -\dfrac{5k+1}{2} + \dfrac{\bar{x} - x_{left}}{h}\right\}, & \bar{x} \in \left[x_{left}, \dfrac{x_{left} + x_{right}}{2}\right), \\ \max\left\{0, \dfrac{5k+1}{2} + \dfrac{\bar{x} - x_{right}}{h}\right\}, & \bar{x} \in \left[\dfrac{x_{left} + x_{right}}{2}, x_{right}\right]. \end{cases} \tag{2.7}$$

Here x_{left} and x_{right} are the left and right boundaries, respectively. In the interior, the symmetric filter uses $2k+1$ central B-splines. In order to provide a smooth transition between the SRV filter and the symmetric filter, a convex combination was used:

$$u_h^\star(x) = \theta(x)\left(K_h^{(2k+1,k+1)} \star u_h\right)(x) + (1 - \theta(x))\left(K_h^{(4k+1,k+1)} \star u_h\right)(x), \tag{2.8}$$

where $\theta(x) \in \mathcal{C}^{k-1}$ such that $\theta = 1$ in the interior and $\theta = 0$ in the boundary regions. The SRV filter showed good performance over uniform meshes in [10] when a multi-precision package was used. However, recent work [8] showed that the SRV filter was not suitable for nonuniform meshes.

2.3.2. RLKV filter

The performance of the SRV filter strongly depends on three conditions: the problem is linear, the mesh is uniform and the computations are carried out with a multi-precision package (at least quadruple precision). When one of these conditions is not satisfied, the good performance of the SRV filter can no longer be guaranteed. In order to overcome the aforementioned limitations, the RLKV filter was proposed in [8]. This RLKV filter uses $2k+1$ central B-splines and an extra general B-spline. Near the left boundary, the RLKV filter has the formula

$$K_{\mathbf{T}}^{(2k+2,k+1)}(x) = \underbrace{\sum_{\gamma=0}^{2k} c_\gamma^{(2k+2,k+1)} \psi_{\mathbf{T}(\gamma)}^{(k+1)}(x)}_{\text{Position-dependent filter with } 2k+1 \text{ central B-splines}} + \underbrace{c_{2k+1}^{(2k+2,k+1)} \psi_{\mathbf{T}(2k+1)}^{(k+1)}(x)}_{\text{General B-spline}}, \tag{2.9}$$

where $\mathbf{T}$ is the knot matrix defined in [8]. This knot matrix is a $(2k+2) \times (k+2)$ matrix such that each row, $\mathbf{T}(i) = [T(i,0), \ldots, T(i,k+1)]$ $(i = 0, \ldots, 2k+1)$, of the matrix is a knot sequence with $k+2$ elements that are used to create the B-spline $\psi_{\mathbf{T}(\gamma)}^{(k+1)}(x)$. Let $\mathbf{T}(i, j_0 : j_1)$ represents the knot sequence $[T(i,j_0), T(i,j_0+1), \ldots, T(i,j_1)]$, then the B-spline can be constructed by the following recurrence relation [13]:

$$\psi_{\mathbf{T}(i,0:\ell)}^{(\ell)}(x) = \begin{cases} \chi_{[T(i,0),T(i,1))} & \text{for } \ell = 1; \\ \dfrac{x - T(i,0)}{T(i,\ell-1) - T(i,0)} \psi_{\mathbf{T}(i,0:\ell-1)}^{(\ell-1)}(x) + \left(1 - \dfrac{x - T(i,1)}{T(i,\ell) - T(i,1)}\right) \psi_{\mathbf{T}(i,1:\ell)}^{(\ell-1)}(x) & \text{for } \ell > 1. \end{cases}$$

We note that the definition above provides more flexibility than the previous central B-spline notation, where $\mathbf{T}(i)$ are sequences of equidistant knots. Assume that we want to filter the point $\bar{x}$ which is located near the left boundary, then $\mathbf{T}$ is given by,

$$T(i,j) = \begin{cases} -3k-1+j+i+\dfrac{\bar{x}-x_{left}}{h}, & 0 \le i \le 2k,\ 0 \le j \le k+1; \\ \dfrac{\bar{x}-x_{left}}{h} + \min\{j-1,0\}, & i = 2k+1,\ 0 \le j \le k+1; \end{cases} \tag{2.10}$$

similarly near the right boundary the knot sequence is given by

$$T(i,j) = \begin{cases} \dfrac{\bar{x}-x_{right}}{h} + \max\{j-k,0\}, & i = 0,\ 0 \le j \le k+1; \\ i+j-1+\dfrac{\bar{x}-x_{right}}{h}, & 1 \le i \le 2k+1,\ 0 \le j \le k+1. \end{cases} \tag{2.11}$$

More details of this position-dependent filter can be found in [8].

2.4. Symmetric derivative filter: uniform meshes

Smoothness-Increasing Accuracy-Conserving filtering is named after its improvement of the smoothness of the filtered approximation. Using the filter in Eq. (2.5), the filtered solution is a $\mathcal{C}^{k-1}$ function. One can see that the smoothness is significantly improved from the original weakly continuous solution. By taking advantage of the improved smoothness, we can obtain better derivative approximations.

Derivative filtering over uniform meshes was introduced in [5,4]. In these papers, the authors identified two ways to calculate the derivatives. The first method is a direct calculation of the derivatives of the filtered solution (2.4). The convergence rate of the filtered solution is higher than the derivatives of the DG approximation itself, but the accuracy order decreases and oscillations in the error increase with each successive derivative. The second method is employed to maintain a fixed accuracy order regardless of the derivative order. In order to calculate the αth derivative of the DG approximation without losing any accuracy order, we have to use higher-order central B-splines to construct the filter,

$$K^{(r+1,k+1+\alpha)}(x) = \sum_{\gamma=0}^{r} c_\gamma^{(r+1,k+1+\alpha)} \psi^{(k+1+\alpha)}\left(x + \frac{r}{2} - \gamma\right). \tag{2.12}$$

Notice that the order of the B-splines is now $k+1+\alpha$ instead of $k+1$ in (2.5), and the filtered solution becomes a $\mathcal{C}^{k-1+\alpha}$ function. Then we can write the αth derivative of the symmetric kernel as $\frac{d^\alpha}{dx^\alpha} K_h^{(r+1,k+1+\alpha)}(x) = \partial_h^\alpha \tilde{K}_h^{(r+1,k+1,\alpha)}$, where

$$\tilde{K}_h^{(r+1,k+1,\alpha)} = \sum_{\gamma=0}^{r} c_\gamma^{(r+1,k+1+\alpha)} \psi_h^{(k+1)}\left(x + \frac{r}{2} - \gamma\right).$$

By the property of convolution,

$$\partial_x^\alpha u_h^\star = \partial_x^\alpha \left(K_h^{(r+1,k+1+\alpha)} \star u_h\right) = \left(\frac{d^\alpha}{dx^\alpha} K_h^{(r+1,k+1+\alpha)}\right) \star u_h = \left(\partial_h^\alpha \tilde{K}_h^{(r+1,k+1,\alpha)}\right) \star u_h. \tag{2.13}$$

For uniform meshes, [5] showed the filtered solution (2.13) has $2k+1$ superconvergence rate regardless of the derivative order α

$$\|\partial_x^\alpha u - \partial_x^\alpha u_h^\star\|_0 \sim \mathcal{O}(h^{2k+1}).$$

Unfortunately, these methods are limited to uniform meshes. For nonuniform meshes, SIAC filtering becomes complicated, and derivative SIAC filtering is more difficult. If we naively apply the same derivative filtering technique over nonuniform meshes, we will lose accuracy from $\mathcal{O}(h^{2k+1})$ to $\mathcal{O}(h^{k+1-\alpha})$ since over nonuniform meshes the divided differences of the DG solution no longer have the superconvergence property. In the following section, we will address nonuniform meshes by adjusting the scaling of the SIAC filter.

3. Derivative filter: nonuniform meshes and near boundaries

3.1. Symmetric derivative filter: nonuniform meshes

A brief introductory description of symmetric derivative filtering over nonuniform meshes can be found in [7]. It discusses the challenges of symmetric derivative filtering over nonuniform meshes and gives preliminary results for smoothly-varying meshes (an affine mapping of a uniform mesh [6]). In order to develop derivative filtering for arbitrary nonuniform meshes, we first present some useful lemmas.

Lemma 3.1 (*Bramble and Schatz [2]*). *Let $\Omega_0 \subset\subset \Omega_1$ and s be an arbitrary but fixed nonnegative integer. Then for $u \in H^s(\Omega_1)$, there exists a constant C such that*

$$\|u\|_{0,\Omega_0} \leq C \sum_{|\alpha|\leq s} \|D^\alpha u\|_{-s,\Omega_1}.$$

Lemma 3.2. *Let u be the exact solution to the linear hyperbolic equation*

$$\begin{aligned} &u_t + \sum_{i=1}^{d} A_i u_{x_i} + A_0 u = 0, \quad \mathbf{x} \in \Omega \times (0, T], \\ &u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \end{aligned} \tag{3.1}$$

where the initial condition $u_0(\mathbf{x})$ is a sufficiently smooth function and the coefficients A_i are constants. Here, $\Omega \subset \mathbb{R}^d$. Let u_h be the DG approximation over a nonuniform mesh with periodic boundary conditions. Denote $\Omega_0 \subset\subset \Omega_1 \subset\subset \Omega$, $\ell \geq k+1$. The negative order norm estimation of $u - u_h$ satisfies,

$$\|(u - u_h)(T)\|_{-\ell,\Omega_1} \leq Ch^{2k+1},$$

and

$$\|\partial_H^\alpha (u - u_h)(T)\|_{-\ell,\Omega_0} \leq C_\alpha h^{2k+1} H^{-|\alpha|},$$

where $\alpha = (\alpha_1, \ldots, \alpha_d)$ is an arbitrary multi-index and H is the scaling of the divided difference operator ∂_H^α.

Proof. The proof of the negative order norm estimation was given in [3] and the divided difference estimation was presented as a hypotheses. The proof is trivial and therefore we only give a proof for $d = 1$ case.

Set Ω_0 such that $\Omega_0 + \left[-\frac{|\alpha|H}{2}, \frac{|\alpha|H}{2}\right] \subset \Omega_1$. Consider the first divided difference, by the definition of the negative order norm, we have

$$\begin{aligned} \|\partial_H (u - u_h)\|_{-\ell,\Omega_0} &= \sup_{\Phi \in C_0^\infty(\Omega_0)} \left(\frac{\left((u - u_h)\left(x + \frac{H}{2}\right), \Phi\right) - \left((u - u_h)\left(x - \frac{H}{2}\right), \Phi\right)}{H\|\Phi\|_{\ell,\Omega_0}} \right), \\ &\leq \sup_{\Phi \in C_0^\infty(\Omega_0)} \frac{\left((u - u_h)\left(x + \frac{H}{2}\right), \Phi\right)}{H\|\Phi\|_{\ell,\Omega_0}} + \sup_{\Phi \in C_0^\infty(\Omega_0)} \frac{\left((u - u_h)\left(x - \frac{H}{2}\right), \Phi\right)}{H\|\Phi\|_{\ell,\Omega_0}}, \\ &\leq \frac{2}{H}\|u - u_h\|_{-\ell,\Omega_1}. \end{aligned}$$

By induction, we have

$$\|\partial_H^\alpha (u - u_h)(T)\|_{-\ell,\Omega_0} \leq C_\alpha h^{2k+1} H^{-|\alpha|},$$

where $C_\alpha = 2^{|\alpha|}C$. The proof is similar for $d > 1$ case. □

Lemma 3.2 demonstrates the optimal accuracy order estimation of the divided differences of the DG approximation in the sense that the nonuniform mesh is arbitrary [3,14].

Theorem 3.3. *Under the same conditions as in Lemma 3.2, let $K^{(r+1,k+1+\alpha)}$ be the symmetric derivative filter given in (2.12). Denote $\Omega_0 + 2\mathrm{supp}(K_H^{(r+1,k+1+\alpha)}) \subset\subset \Omega_1 \subset\subset \Omega$. Then, for general nonuniform meshes, we have*

$$\|\partial_x^\alpha u - \partial_x^\alpha \left(K_H^{(r+1,k+1+\alpha)} \star u_h\right)\|_{0,\Omega_0} \leq Ch^{\frac{r+1}{r+k+2+\alpha}(2k+1)},$$

where $H = h^\mu$ and $\mu = \frac{2k+1}{r+k+2+\alpha}$.

Proof. Set $\Omega_{1/2}$ such that

$$\Omega_0 + \mathrm{supp}(K_H^{(r+1,k+1+\alpha)}) \subset \Omega_{1/2}, \quad \text{and} \quad \Omega_{1/2} + \mathrm{supp}(K_H^{(r+1,k+1+\alpha)}) \subset \Omega_1.$$

By applying Lemmas 3.1 and 3.2, we have

$$\begin{aligned} \left\|\partial_x^\alpha u - \partial_x^\alpha \left(K_H^{(r+1,k+1+\alpha)} \star u_h\right)\right\|_{0,\Omega_0} &\leq \left\|\partial_x^\alpha u - K_H^{(r+1,k+1+\alpha)} \star \partial_x^\alpha u\right\|_{0,\Omega_0} + \left\|\partial_x^\alpha \left(K_H^{(r+1,k+1+\alpha)} \star (u - u_h)\right)\right\|_{0,\Omega_0} \\ &\leq C_0 H^{r+1} + C_1 \sum_{|\beta|\leq k+1} \left\|\partial_x^{\alpha+\beta} \left(K_H^{(r+1,k+1+\alpha)} \star (u - u_h)\right)\right\|_{-(k+1),\Omega_{1/2}} \end{aligned}$$

$$
\begin{aligned}
&= C_0 H^{r+1} + C_1 \sum_{|\beta|\le k+1} \left\| \left(\tilde{K}_H^{(r+1,k+1-\beta,\alpha+\beta)} \star \partial_H^{\alpha+\beta}(u-u_h) \right) \right\|_{-(k+1),\Omega_{1/2}} \\
&= C_0 H^{r+1} + C_1 \sum_{|\beta|\le k+1} \left\| \tilde{K}_H^{(r+1,k+1-\beta,\alpha+\beta)} \right\|_{L^1} \left\| \partial_H^{\alpha+\beta}(u-u_h) \right\|_{-(k+1),\Omega_1} \\
&\le C_0 H^{r+1} + C_2 h^{2k+1} H^{-(k+1+\alpha)}.
\end{aligned}
$$

Let the scaling $H = h^{\mu}$ such that

$$H^{r+1} = h^{2k+1} H^{-(k+1+\alpha)}.$$

We then have that $\mu = \frac{2k+1}{r+k+2+\alpha}$ and

$$\left\| \partial_x^{\alpha} u - \partial_x^{\alpha} \left(K_H^{(r+1,k+1+\alpha)} \star u_h \right) \right\|_{0,\Omega_0} \le C h^{\frac{r+1}{r+k+2+\alpha}(2k+1)}. \quad \square$$

Remark 3.1 (*Discussion of the Number of B-Splines*)**.** The filter given in (2.12) uses $(r+1)$ B-splines. Theorem 3.3 implies that by increasing the value of r, one can increase the value of $\frac{r+1}{r+k+2+\alpha}$, and then approximate the superconvergence rate $2k+1$ as close as we want and regardless of the derivative order α. However, increasing the value of r presents a serious inconvenience for computational implementation. For example, while $r \gg 2k$, a multi-precision package is required during the computation process, [8]. Another disadvantage is that the support size of the filter, $(r+k+1+\alpha)h^{\mu}$, increases with r [3]. The increased support size means the convolution involves more DG elements and that the computational cost increases as well. For nonderivative filtering, we usually keep $r = 2k$, but there is another consideration for derivative filtering. We notice that the accuracy order decreases with the derivative order α if we keep $r = 2k$. One solution is to eliminate the negative effect of the derivative order α is to use $r = 2(k+\alpha)$ instead of $r = 2k$. However, our experience shows that the benefit of using $r = 2(k+\alpha)$ is limited. It slightly improves the accuracy and smoothness, but increases the computational cost. In this paper, we will focus on using $r = 2k$ for nonuniform meshes.

3.2. Position-dependent derivative filter

In the previous section, we discussed the symmetric derivative filtering over nonuniform meshes. As we mentioned before, in order to handle boundary regions, we need to use position-dependent filters. In this section, we extend two position-dependent filters to position-dependent derivative filters.

3.2.1. Derivative SRV filter

Since the SRV filter uses only central B-splines, we can easily extend it to the derivative SRV filter by increasing the order of B-splines from $k+1$ in (2.6) to $k+1+\alpha$

$$K^{(4k+1,k+1+\alpha)}(x) = \sum_{\gamma=0}^{4k} c_{\gamma}^{(4k+1,k+1+\alpha)} \psi^{(k+1+\alpha)}(x - x_{\gamma}), \tag{3.2}$$

and adjusting the shift function $\lambda(\bar{x})$ (2.7) to

$$\lambda(\bar{x}) = \begin{cases} \min\left\{0, -\dfrac{5k+1+\alpha}{2} + \dfrac{\bar{x} - x_{left}}{h}\right\}, & \bar{x} \in \left[x_{left}, \dfrac{x_{left}+x_{right}}{2}\right), \\ \max\left\{0, \dfrac{5k+1+\alpha}{2} + \dfrac{\bar{x} - x_{right}}{h}\right\}, & \bar{x} \in \left[\dfrac{x_{left}+x_{right}}{2}, x_{right}\right]. \end{cases} \tag{3.3}$$

For example, Fig. 3.1 shows the derivative SRV filters with $k = 2$ for the first and second derivatives at the left boundary.

Remark 3.2. The theoretical analysis of the derivative SRV filter remains the same as Theorem 3.3 with $r = 4k$. The difference between the derivative SRV filter and the symmetric derivative filter is the scaling $H = h^{\mu}$. The scaling of the derivative SRV filter is $H = h^{\frac{2k+1}{5k+2+\alpha}}$, which is much larger than the scaling of the symmetric derivative filter, $H = h^{\frac{2k+1}{3k+2+\alpha}}$.

3.2.2. Derivative RLKV filter

For the RLKV filter, we need to shift the $2k+1$ central B-splines and then change the extra general B-spline according to the derivative order α. To complete these changes, we have to change the knot sequence (2.10), which is used only for

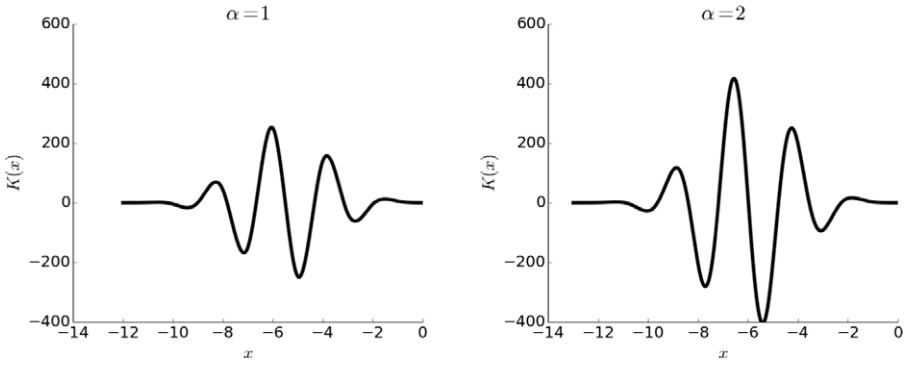

Fig. 3.1. The derivative SRV filters (first and second derivatives) before convolution at the left boundary with $k = 2$ and scaling $H = 1$. The boundary is represented by $x = 0$.

the DG approximation u_h without derivatives. For the derivative RLKV filter near the left boundary (similar for the right boundary), we need to redistribute the knots in the knot matrix $\mathbf{T}$ to meet the derivative requirement by

$$T(\gamma, j) = \begin{cases} -3k - 1 - \alpha + j + \gamma + \dfrac{\bar{x} - x_{left}}{h}, & 0 \le \gamma \le 2k,\ 0 \le j \le k + 1 + \alpha; \\ \dfrac{\bar{x} - x_{left}}{h} + \min\{j - \alpha, 0\}, & \gamma = 2k + 1,\ 0 \le j \le k + 1 + \alpha, \end{cases} \tag{3.4}$$

and the position-dependent derivative filter is given by

$$K_{\mathbf{T}}^{(2k+2,k+1+\alpha)}(x) = \underbrace{\sum_{\gamma=0}^{2k} c_{\gamma}^{(2k+2,k+1)} \psi_{\mathbf{T}(\gamma)}^{(k+1+\alpha)}(x)}_{\text{Position-dependent filter with } 2k+1 \text{ central B-splines}} + \underbrace{c_{2k+1}^{(2k+2,k+1+\alpha)} \psi_{\mathbf{T}(2k+1)}^{(k+1+\alpha)}}_{\text{General B-spline}}, \tag{3.5}$$

Remark 3.3. It is necessary to use a B-spline of order $k+1+\alpha$ instead of $k+1$ when $\alpha > k$. In formula (3.5), if we keep the order of B-spline as $k+1$, when $\alpha > k$ the knot sequence $\mathbf{T}(2k+1)$ becomes a uniformly spaced knot sequence, and then the general B-spline $\psi_{\mathbf{T}(2k+1)}^{(k+1+\alpha)}$ added at the boundary reduces to a central B-spline. Then, the purpose of adding a special B-spline at the boundary fails, and this special B-spline is needed to place more weight on the filtered point.

We note that the derivative RLKV filter allows us to approximate arbitrary order of derivatives near boundaries theoretically. For example, Fig. 3.2 shows the derivative RLKV filters with $k = 2$ for the first and second derivatives at the left boundary. Compared to the derivative SRV filter in Fig. 3.1, the derivative RLKV filter clearly has reduced support and magnitude (range from -400 to 600 versus -4 to 6).

Theorem 3.4. *Under the same conditions as in Lemma 3.2, let $K_{\mathbf{T}}^{(2k+2,k+1+\alpha)}$ be the derivative RLKV filter (3.5). We have*

$$\left\| \partial_x^{\alpha} u - \partial_x^{\alpha} \left(K_{H\mathbf{T}}^{(2k+2,k+1+\alpha)} \star u_h \right) \right\|_{0,\Omega_0} \le C h^{\mu(2k+2)},$$

where $H = h^{\mu}$, $\mu = \frac{2k+1}{3k+3+\alpha}$.

Proof.

$$\begin{aligned} \left\| \partial_x^{\alpha} u - \partial_x^{\alpha} \left(K_{H\mathbf{T}}^{(2k+2,k+1+\alpha)} \star u_h \right) \right\|_{0,\Omega_0} \le\ & C_0 H^{2k+2} + \left\| \partial_x^{\alpha} \left(\sum_{\gamma=0}^{2k} c_{\gamma} \psi_{H\mathbf{T}(\gamma)}^{(k+1+\alpha)} \star (u - u_h) \right) \right\|_{0,\Omega_0} \\ & + \left\| \partial_x^{\alpha} \left(c_{2k+1} \psi_{H\mathbf{T}(2k+1)}^{(k+1+\alpha)} \star (u - u_h) \right) \right\|_{0,\Omega_0}. \end{aligned}$$

For the second term on the left side of the above inequality, which only involves central B-splines, similar to Theorem 3.3, we have

$$\left\| \partial_x^{\alpha} \left(\sum_{\gamma=0}^{2k} c_{\gamma} \psi_{H\mathbf{T}(\gamma)}^{(k+1+\alpha)} \star (u - u_h) \right) \right\|_{0,\Omega_0} \le C_1 h^{2k+1} H^{-(k+1+\alpha)}.$$

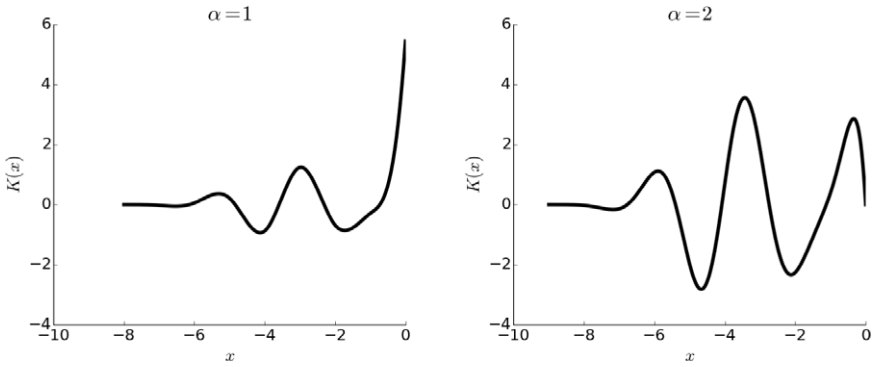

Fig. 3.2. The derivative RLKV filters (first and second derivatives) before convolution at the left boundary with $k = 2$ and scaling $H = 1$. The boundary is represented by $x = 0$.

For the third term with a general B-spline, we have

$$\begin{aligned}
\left\| \partial_x^\alpha \left(c_{2k+1} \psi_{H\mathbf{T}(2k+1)}^{(k+1+\alpha)} \star (u - u_h) \right) \right\|_{0,\Omega_0} &\le C_2 \sum_{\beta \le k+1} \left\| c_{2k+1} \left(\frac{d^{\alpha+\beta}}{dx^{\alpha+\beta}} \psi_{H\mathbf{T}(2k+1)}^{(k+1)} \right) \star (u - u_h) \right\|_{-(k+1),\Omega_{1/2}} \\
&\le C_2 \sum_{\beta \le k+1} \left\| c_{2k+1} \left(\frac{d^{\alpha+\beta}}{dx^{\alpha+\beta}} \psi_{H\mathbf{T}(2k+1)}^{(k+1)} \right) \right\|_{L^1} \|u - u_h\|_{-(k+1),\Omega_1} \\
&\le C_3 \sum_{\beta \le k+1} H^{-(\alpha+\beta)} \left\| \left(\frac{d^{\alpha+\beta}}{dx^{\alpha+\beta}} \psi_{\mathbf{T}(2k+1)}^{(k+1)} \right) \right\|_{L^1} \|u - u_h\|_{-(k+1),\Omega_1} \\
&\le C_4 h^{2k+1} H^{-(k+1+\alpha)},
\end{aligned}$$

where

$$\Omega_0 + \operatorname{supp}(K_{H\mathbf{T}}^{(2k+2,k+1+\alpha)}) \subset \Omega_{1/2}, \quad \text{and} \quad \Omega_{1/2} + \operatorname{supp}(K_{H\mathbf{T}}^{(2k+2,k+1+\alpha)}) \subset \Omega_1.$$

Then, we have

$$\left\| \partial_x^\alpha u - \partial_x^\alpha \left(K_{H\mathbf{T}}^{(2k+2,k+1+\alpha)} \star u_h \right) \right\|_{0,\Omega_0} \le C_0 H^{2k+2} + C_5 h^{2k+1} H^{-(k+1+\alpha)}.$$

Similar to the symmetric filter case in Theorem 3.3, we require the scaling H satisfies $H^{2k+2} = h^{2k+1} H^{-(k+1+\alpha)}$ and finally, we have

$$\left\| \partial_x^\alpha u - \partial_x^\alpha \left(K_{H\mathbf{T}}^{(2k+2,k+1+\alpha)} \star u_h \right) \right\|_{0,\Omega_0} \le C h^{\mu(2k+2)},$$

where $H = h^\mu$ and $\mu = \frac{2k+1}{3k+3+\alpha}$. □

Remark 3.4 (*Discussion of Support Size of the Filters*)**.** Theorems 3.3 and 3.4 give convergence rates of the symmetric and position-dependent derivative filters, respectively. One can easily verify that the convergence rates are better than calculating the derivatives of the DG approximation directly, $k + 1 - \alpha$. For the scaling $H = h^\mu$, for convenience we let the degree $k \to \infty$, then the symmetric derivative filter and derivative RLKV filters have the scaling $H = h^{2/3}$ and the derivative SRV filter has the scaling $H = h^{2/5}$. In order to show the difference of support size of the different filters, we present Fig. 3.3 to show a direct comparison. From Fig. 3.3, we can see that the SRV filter requires a much larger support size than the RLKV filter. The large support size usually will lead to computational problems (increased flop counts, round-off error, etc.).

However, we notice that the scaling $H = h^\mu$ is still quite large compared to h. A large support usually has negative effects on the accuracy over coarse meshes. Let the domain be $\Omega = [0, 1]$ and $h = 1/N$, where N is the number of elements. In order to guarantee the conclusions in Theorems 3.3 and 3.4, we must choose N large enough so that the support size of filters is less than the domain size, which requires

$$(r + k + \alpha + 1)h^\mu \le 1 \Longrightarrow N \ge (r + k + \alpha + 1)^{1/\mu},$$

here $r = 2k$ for the symmetric and derivative RLKV filters and $r = 4k$ for the derivative SRV filter. Table 3.1 gives the minimum number of elements for different filters. We note that for the SRV filter, the required number of elements is always too large. This is one important reason that the SRV filter performs poorly over nonuniform meshes in the numerical examples. Once N is smaller than the minimum requirement given in Table 3.1, we have to rescale the filter by using scaling

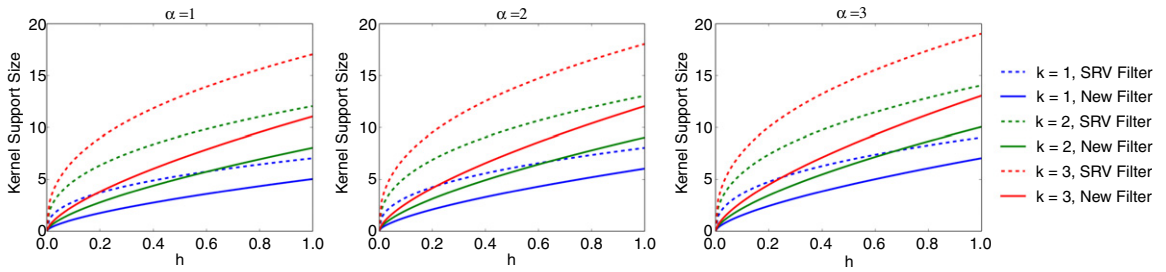

Fig. 3.3. Comparison of support sizes of the derivative SRV filter and the derivative RLKV filters. The symmetric derivative filter has the same support size as the derivative RLKV filter.

Table 3.1
The minimum requirement of element number N according to the derivative order α. Here, N_1 is used for the symmetric and derivative RLKV filters and N_2 is used for the derivative SRV filter.

N_1 / N_2	$\alpha=0$	$\alpha=1$	$\alpha=2$	$\alpha=3$	$\alpha=4$	$\alpha=5$
$k=1$	8 / 89	12 / 130	15 / 182	19 / 243	23 / 317	27 / 402
$k=2$	19 / 402	23 / 499	27 / 610	32 / 734	37 / 872	42 / 1024
$k=3$	32 / 1024	37 / 1192	42 / 1375	47 / 1574	53 / 1789	59 / 2021

$H = 1/(r+k+\alpha+1)$. However, this rescaling technique normally has negative effects on the accuracy order. There are two ways by which to overcome these drawbacks. One is to keep the order of B-splines as $k+1$ as the nonderivative filter and then calculate the derivatives directly as mentioned in [9]. This method can decrease the support size of the filters from $(r+k+\alpha+1)H$ to $(r+k+1)H$. In the next section, we will present a numerical comparison of using the order of B-splines of $k+1$ and $k+1+\alpha$. The other way will be presented in our future work; it will give an alternative version of Lemma 3.2 according to the given nonuniform mesh. It allows us to choose a smaller scaling H (or larger μ) than that in Theorems 3.3 and 3.4.

4. Numerical results

In the previous section, we proposed two position-dependent derivative filters and investigated how to choose the proper scaling of the filters over nonuniform meshes. We also proved that the filtered solutions have better accuracy order and smoothness compared to the original DG approximations regardless of the derivative order α. We now turn to the numerical examples of the position-dependent derivative filtering. The aims of this section are:

1. Testing the position-dependent derivative filters (the SRV and RLKV filter) for uniform meshes, which has never been done before;
2. Applying the symmetric and position-dependent derivative filters over different nonuniform meshes;
3. Comparing the derivative filters with different order B-splines. For convenience, we introduce the following notation:
   - the derivative of the filtered solution, $\partial_x^\alpha u_h^\star$. This filtered solution uses the standard filter and then takes the derivative.
   - the filtered derivative, $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$, which uses the higher order derivative filter $K_H^{(r+1,k+1+\alpha)}$ for filtering the DG solution.

We note that the DG approximation makes sense only when $\alpha \leq k$. In addition, the derivative of the filtered solution $\partial_x^\alpha u_h^\star$ loses the wanted accuracy order when $\alpha > k$ ($u_h^\star = K_H^{r+1,k+1} \star u_h$ is a C^{k-1} function only). Therefore, we mainly present comparison examples with $\alpha \leq k$ in this section. When $\alpha > k$, we only present the results of the filtered derivative $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$, and we point out that the filtered solution $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ has a theoretical meaning for an arbitrary α, but the accuracy deteriorates with each successive derivative. However, we also note that once $\alpha > k$, the nonuniform meshes have to be sufficiently refined in order to see the accuracy improvement. Because of these reasons, we only present the $\alpha = k+1$ case for nonuniform meshes. Also, since the symmetric derivative filter is applied in the interior region of each example, we do not present it separately.

Remark 4.1. For the following numerical examples:

- when the number of elements is less than the minimum requirement in Table 3.1, a rescaling technique is used;
- quadruple precision is used for the SRV filter, and double precision is used for the RLKV filter and all two-dimensional examples;

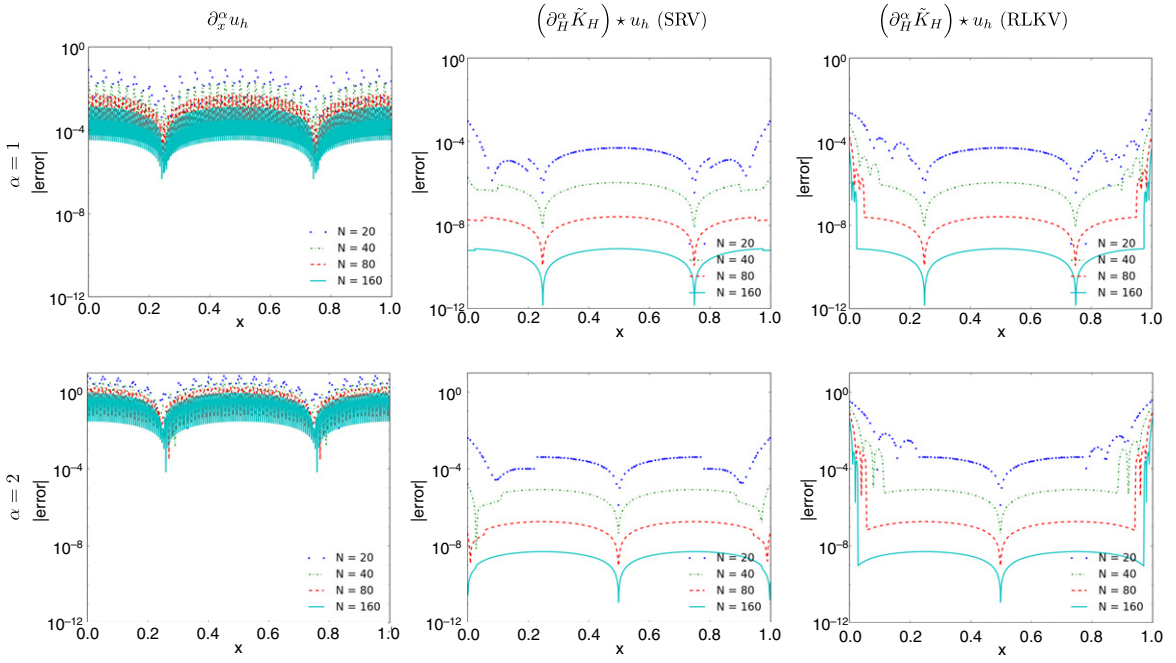

Fig. 4.1. Comparison of the point-wise errors in log scale of the first and second derivatives of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions (the SRV and RLKV filters) for the linear convection equation (4.1), over uniform meshes. The B-spline order is $k+1+\alpha$, the filter scaling is taken as $H = h$, and $k = 2$.

- the blending function $\theta(x)$ in (2.8) is no longer needed for the RLKV filter (see [8]), therefore the function $\theta(x)$ is not used in the following examples.

4.1. Uniform mesh

Before approaching nonuniform meshes, we first apply the position-dependent derivative filters over uniform meshes. Here we present results of using both the SRV filter and the RLKV filter since each of them has its advantages over uniform meshes that we address in the following examples. Consider a linear convection equation

$$\begin{aligned} &u_t + u_x = 0, \quad x \in [0, 1], \\ &u(x, 0) = \sin(2\pi x), \end{aligned} \tag{4.1}$$

at time $T = 1$ with periodic boundary conditions. For uniform meshes, we can also use scaling $H = h^\mu$ and obtain results as Theorems 3.3 and 3.4 described. However, according to the analysis in [3,8,10], in order to maximize the benefits of using central B-splines over uniform meshes, we choose the uniform mesh size, h, as the filter scaling. Here, we compare the derivatives of the DG approximation, the filtered solutions (the SRV and RLKV filter) with using B-splines of order $k+1+\alpha$ (Table 4.1 and Fig. 4.1) and using B-splines of order $k+1$ (Table 4.2 and Fig. 4.2). From the tables, we can see that the filtered solutions $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ and $\partial_x^\alpha u_h^\star$ have better accuracy compared to the original DG solutions.

With the scaling $H = h$, the SRV filter clearly has an advantage for uniform meshes. Because the SRV filter is constructed using only central B-splines, it was proven to have $2k+1$ accuracy order regardless of the derivative order α for linear equations over uniform meshes in [9]. In Tables 4.1 and 4.2, the SRV filter shows much better accuracy compared to the RLKV filter near the boundaries, especially when α is large. For the RLKV position-dependent derivative filter, we notice that the filtered solutions only have accuracy of order $k+1-\alpha$ in Tables 4.1 and 4.2. This is because we use scaling $H = h$ instead of scaling $H = h^\mu$ in Theorem 3.4. We note that if using a multi-precision package is acceptable, then the SRV filter is more advantageous for the accuracy order. However, if only double precision is available during computation (for example, GPU computing), then in order to avoid round-off error, the RLKV filter is a better choice, see [8]. However, when $\alpha > k$, the optimal choice is still the SRV filter with B-splines of order $k+1+\alpha$ because only this filter does not lose the accuracy with each successive derivative.

We note that the derivative of filtered solution $\partial_x^\alpha u_h^\star$ also performs well near boundaries for uniform meshes. However, for the derivative order $\alpha > k$, we still need to use higher-order B-splines to construct the derivative filters. Figs. 4.1 and 4.2 present the point-wise error plots in log scale using the DG approximation of degree $k = 2$. After filtering, the filtered approximations are much smoother than the DG solution, but in order to reduce oscillations in the interior regions, we still have to use B-splines of order $k+1+\alpha$.

Table 4.1
L^2- and L^∞-errors for the αth derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions (the SRV and RLKV filters) for the linear convection Eq. (4.1), over uniform meshes. The B-spline order is $k+1+\alpha$ and the filter scaling is taken as $H=h$.

Mesh	$\partial_x^\alpha u_h$				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (SRV)				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$(RLKV)			
	L^2 error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order
$\alpha=1$												
$\mathcal{P}^1$												
20	4.62E−01	–	1.22E+00	–	1.43E−02	–	4.41E−02	–	1.22E−02	–	2.07E−02	–
40	2.32E−01	0.99	6.22E−01	0.98	1.55E−03	3.20	2.61E−03	4.08	1.55E−03	2.97	4.60E−03	2.17
80	1.16E−01	1.00	3.12E−01	0.99	1.91E−04	3.02	2.74E−04	3.25	2.04E−04	2.92	1.20E−03	1.94
160	5.82E−02	1.00	1.56E−01	1.00	2.37E−05	3.01	3.36E−05	3.03	2.84E−05	2.84	3.01E−04	1.99
320	2.91E−02	1.00	7.81E−02	1.00	2.96E−06	3.01	4.18E−06	3.01	4.22E−06	2.75	7.50E−05	2.00
$\mathcal{P}^2$												
20	2.19E−02	–	7.97E−02	–	1.40E−04	–	9.00E−04	–	4.78E−04	–	3.09E−03	–
40	5.48E−03	2.00	2.01E−02	1.98	6.69E−07	7.71	1.91E−06	8.88	8.14E−05	2.55	6.83E−04	2.18
80	1.37E−03	2.00	5.05E−03	2.00	1.69E−08	5.31	2.52E−08	6.24	1.44E−05	2.50	1.68E−04	2.02
160	3.43E−04	2.00	1.26E−03	2.00	5.13E−10	5.04	7.37E−10	5.09	2.54E−06	2.50	4.20E−05	2.00
320	8.56E−05	2.00	3.16E−04	2.00	2.14E−11	4.58	3.04E−11	4.60	4.50E−07	2.50	1.05E−05	2.00
$\mathcal{P}^3$												
20	6.55E−04	–	2.80E−03	–	2.41E−06	–	1.59E−05	–	6.24E−06	–	2.50E−05	–
40	8.20E−05	3.00	3.53E−04	2.99	2.10E−09	10.16	3.89E−09	11.99	1.04E−07	5.91	7.61E−07	5.04
80	1.02E−05	3.00	4.42E−05	3.00	9.95E−12	7.72	1.63E−11	7.90	2.18E−09	5.58	2.99E−08	4.67
160	1.28E−06	3.00	5.53E−06	3.00	1.10E−13	6.50	1.62E−13	6.65	9.58E−11	4.51	1.78E−09	4.07
320	1.60E−07	3.00	6.92E−07	3.00	8.98E−15	3.62	1.27E−14	3.67	4.24E−12	4.50	1.11E−10	4.00
$\alpha=2$												
$\mathcal{P}^2$												
20	2.67E+00	–	6.96E+00	–	7.20E−04	–	4.12E−03	–	6.42E−02	–	3.71E−01	–
40	1.34E+00	1.00	3.50E+00	0.99	5.90E−06	6.93	1.73E−05	7.89	2.27E−02	1.50	1.74E−01	1.09
80	6.70E−01	1.00	1.75E+00	1.00	1.29E−07	5.51	1.83E−07	6.57	8.03E−03	1.50	8.67E−02	1.00
160	3.35E−01	1.00	8.78E−01	1.00	3.55E−09	5.19	5.02E−09	5.19	2.84E−03	1.50	4.33E−02	1.00
320	1.67E−01	1.00	4.39E−01	1.00	1.39E−10	4.67	1.97E−10	4.67	1.00E−03	1.50	2.17E−02	1.00
$\mathcal{P}^3$												
20	1.34E−01	–	4.78E−01	–	6.13E−05	–	3.87E−04	–	6.48E−04	–	4.84E−03	–
40	3.36E−02	2.00	1.21E−01	1.99	2.35E−08	11.35	3.59E−08	13.39	1.58E−05	5.36	1.49E−04	5.03
80	8.40E−03	2.00	3.02E−02	2.00	1.03E−10	7.83	1.48E−10	7.93	1.39E−06	3.51	1.73E−05	3.10
160	2.10E−03	2.00	7.56E−03	2.00	8.46E−13	6.93	1.20E−12	6.95	1.23E−07	3.50	2.16E−06	3.00
320	5.25E−04	2.00	1.89E−03	2.00	5.70E−14	3.89	8.06E−14	3.89	1.09E−08	3.50	2.70E−07	3.00
$\alpha=3$												
$\mathcal{P}^3$												
20	1.64E+01	–	4.16E+01	–	3.68E−04	–	2.26E−03	–	1.74E−02	–	1.09E−01	–
40	8.19E+00	1.00	2.09E+01	0.99	1.93E−07	10.90	8.61E−07	11.36	3.07E−03	2.50	2.54E−02	2.10
80	4.10E+00	1.00	1.05E+01	1.00	7.68E−10	7.97	1.30E−09	9.38	5.43E−04	2.50	6.42E−03	1.99
160	2.05E+00	1.00	5.24E+00	1.00	5.93E−12	7.02	8.96E−12	7.18	9.60E−05	2.50	1.61E−03	2.00
320	1.02E+00	1.00	2.62E+00	1.00	8.50E−13	2.80	2.53E−11	−1.50	1.70E−05	2.50	4.02E−04	2.00

Remark 4.2. For uniform meshes, we choose to use the scaling $H=h$ instead of the scaling $H=h^\mu$ in Theorem 3.3. This is because for uniform meshes, the scaling $H=h$ can provide a better accuracy order of $2k+1$ compared to the conclusion in Theorem 3.3, especially in the interior region. Also, the SRV filter benefits of the scaling $H=h$ in the boundary region once quadruple precision is used. If the scaling $H=h^\mu$ is used for uniform meshes, the accuracy order will decrease and the error magnitude will increase in the interior region. However, the RLKV filter will have better accuracy order in the boundary region, and the error magnitude will improve once the mesh is sufficiently refined.

4.1.1. Smoothly-varying mesh

As mentioned in [6,8], there is a particular family of nonuniform meshes, smoothly-varying meshes for which the filter behaves similar to the uniform case. In [8], the authors proved that the filtered solutions (both the SRV and RLKV filters) have a similar performance over smoothly-varying meshes compared to uniform meshes. However, it would be of practical interests to show the performance of the position-dependent derivative filters over smoothly-varying meshes, especially for smoothly increasing/decreasing meshes.

Consider a linear convection equation with Dirichlet boundary conditions

$$u_t + u_x = 0, \quad x \in [0,1], \tag{4.2}$$

Table 4.2
L^2- and L^∞-errors for the αth derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions (the SRV and RLKV filters) for the linear convection equation (4.1), over uniform meshes. The B-spline order is $k+1$ and the filter scaling is taken as $H = h$.

Mesh	$\partial_x^\alpha u_h$				$\partial_x^\alpha u_h^\star$ (SRV)				$\partial_x^\alpha u_h^\star$(RLKV)			
	L^2 error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order
$\alpha = 1$												
$\mathcal{P}^1$												
20	4.62E−01	–	1.22E+00	–	1.25E−02	–	2.52E−02	–	1.45E−02	–	6.89E−02	–
40	2.32E−01	0.99	6.22E−01	0.98	1.53E−03	3.03	2.25E−03	3.48	1.91E−03	2.92	1.28E−02	2.43
80	1.16E−01	1.00	3.12E−01	0.99	1.91E−04	3.01	2.72E−04	3.05	2.63E−04	2.86	2.64E−03	2.28
160	5.82E−02	1.00	1.56E−01	1.00	2.38E−05	3.00	3.38E−05	3.01	3.84E−05	2.78	5.90E−04	2.16
320	2.91E−02	1.00	7.81E−02	1.00	2.96E−06	3.00	4.22E−06	3.00	5.95E−06	2.69	1.39E−04	2.09
$\mathcal{P}^2$												
20	2.19E−02	–	7.97E−02	–	5.03E−05	–	3.23E−04	–	5.66E−05	–	2.84E−04	–
40	5.48E−03	2.00	2.01E−02	1.98	5.38E−07	6.55	9.68E−07	8.38	1.05E−06	5.75	8.34E−06	5.09
80	1.37E−03	2.00	5.05E−03	2.00	1.51E−08	5.16	2.22E−08	5.44	2.25E−08	5.55	2.87E−07	4.86
160	3.43E−04	2.00	1.26E−03	2.00	4.83E−10	4.96	6.93E−10	5.00	7.49E−10	4.91	1.39E−08	4.37
320	8.56E−05	2.00	3.16E−04	2.00	2.10E−11	4.53	2.98E−11	4.54	3.22E−11	4.54	8.03E−10	4.12
$\mathcal{P}^3$												
20	6.55E−04	–	2.80E−03	–	9.62E−07	–	6.45E−06	–	4.03E−06	–	1.64E−05	–
40	8.20E−05	3.00	3.53E−04	2.99	1.34E−09	9.48	2.51E−09	11.33	3.56E−08	6.83	2.59E−07	5.98
80	1.02E−05	3.00	4.42E−05	3.00	6.65E−12	7.66	1.09E−11	7.85	4.55E−10	6.29	6.34E−09	5.35
160	1.28E−06	3.00	5.53E−06	3.00	9.66E−14	6.11	1.41E−13	6.27	1.95E−11	4.55	3.50E−10	4.18
320	1.60E−07	3.00	6.92E−07	3.00	8.92E−15	3.44	1.26E−14	3.48	8.61E−13	4.50	2.17E−11	4.01
$\alpha = 2$												
$\mathcal{P}^2$												
20	2.67E+00	–	6.96E+00	–	1.94E−04	–	5.41E−04	–	1.14E−03	–	1.11E−02	–
40	1.34E+00	1.00	3.50E+00	0.99	1.19E−05	4.03	2.50E−05	4.44	7.49E−05	3.92	7.18E−04	3.95
80	6.70E−01	1.00	1.75E+00	1.00	7.42E−07	4.00	1.49E−06	4.07	6.31E−06	3.57	8.11E−05	3.15
160	3.35E−01	1.00	8.78E−01	1.00	4.64E−08	4.00	9.73E−08	3.93	5.45E−07	3.53	9.89E−06	3.04
320	1.67E−01	1.00	4.39E−01	1.00	2.90E−09	4.00	6.16E−09	3.98	4.76E−08	3.52	1.22E−06	3.02
$\mathcal{P}^3$												
20	1.34E−01	–	4.78E−01	–	5.93E−06	–	4.03E−05	–	1.24E−04	–	1.02E−03	–
40	3.36E−02	2.00	1.21E−01	1.99	1.04E−08	9.15	1.91E−08	11.05	2.22E−07	9.12	2.66E−06	8.58
80	8.40E−03	2.00	3.02E−02	2.00	5.43E−11	7.58	1.20E−10	7.31	3.92E−10	9.15	7.91E−09	8.39
160	2.10E−03	2.00	7.56E−03	2.00	7.45E−13	6.19	1.70E−12	6.15	5.52E−12	6.15	1.03E−10	6.27
320	5.25E−04	2.00	1.89E−03	2.00	5.65E−14	3.72	9.20E−14	4.21	1.67E−13	5.05	5.58E−12	4.20
$\alpha = 3$												
$\mathcal{P}^3$												
20	1.64E+01	–	4.16E+01	–	3.87E−05	–	2.55E−04	–	4.23E−04	–	3.61E−03	–
40	8.19E+00	1.00	2.09E+01	0.99	4.36E−07	6.47	1.12E−06	7.83	4.26E−06	6.63	5.85E−05	5.95
80	4.10E+00	1.00	1.05E+01	1.00	1.48E−08	4.88	3.47E−08	5.01	8.67E−08	5.62	1.26E−06	5.54
160	2.05E+00	1.00	5.24E+00	1.00	4.68E−10	4.98	1.09E−09	5.00	3.73E−09	4.54	6.67E−08	4.24
320	1.02E+00	1.00	2.62E+00	1.00	1.48E−11	4.98	7.85E−11	3.79	1.65E−10	4.50	4.15E−09	4.01

$$u(x, 0) = \sin(2\pi x),$$
$$u(0, t) = \sin(-2\pi t),$$

at time $T = 1$ over a smoothly decreasing mesh defined in [8]:

$$x = \xi - 0.2(\xi - 1)\xi,$$

where x is the smoothly decreasing mesh variable and ξ is the variable of the uniform mesh over domain $[0, 1]$.

Similar to the uniform mesh case, we choose the local mesh size as the filter scaling, $H = \Delta x_j$ according to [8]. Here, in order to save space, we present results for the filtered derivative $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ only. The L^2 and L^∞ errors are presented in Table 4.3 with the first three derivatives over the above smoothly decreasing mesh. The respective point-wise error plots ($k = 2$ case) are given in Fig. 4.3. We point out one phenomenon that is very different to the uniform mesh case. The RLKV filter provides better accuracy compared to the SRV filter near the boundaries, see Table 4.3. However, both the SRV and RLKV filters can improve the accuracy of the original DG solutions once the mesh is sufficiently refined.

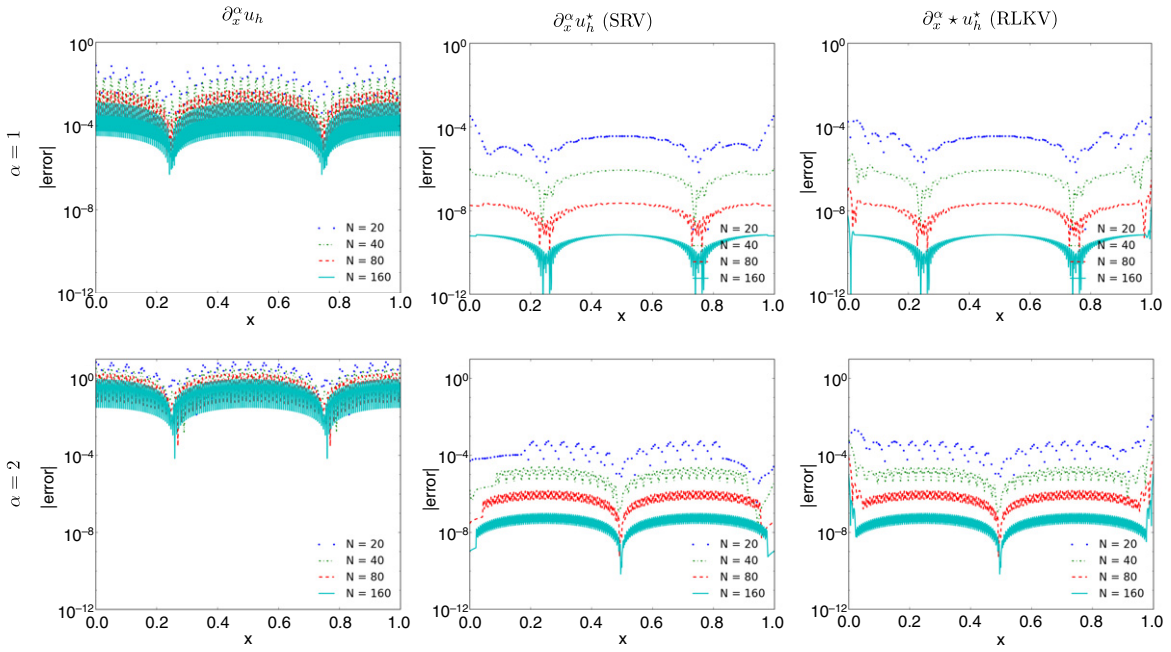

Fig. 4.2. Comparison of the point-wise errors in log scale of the first derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions (the SRV and RLKV filters) for the linear convection equation (4.1), over uniform meshes. The B-spline order is $k+1$, the filter scaling is taken as $H = h$, and $k = 2$.

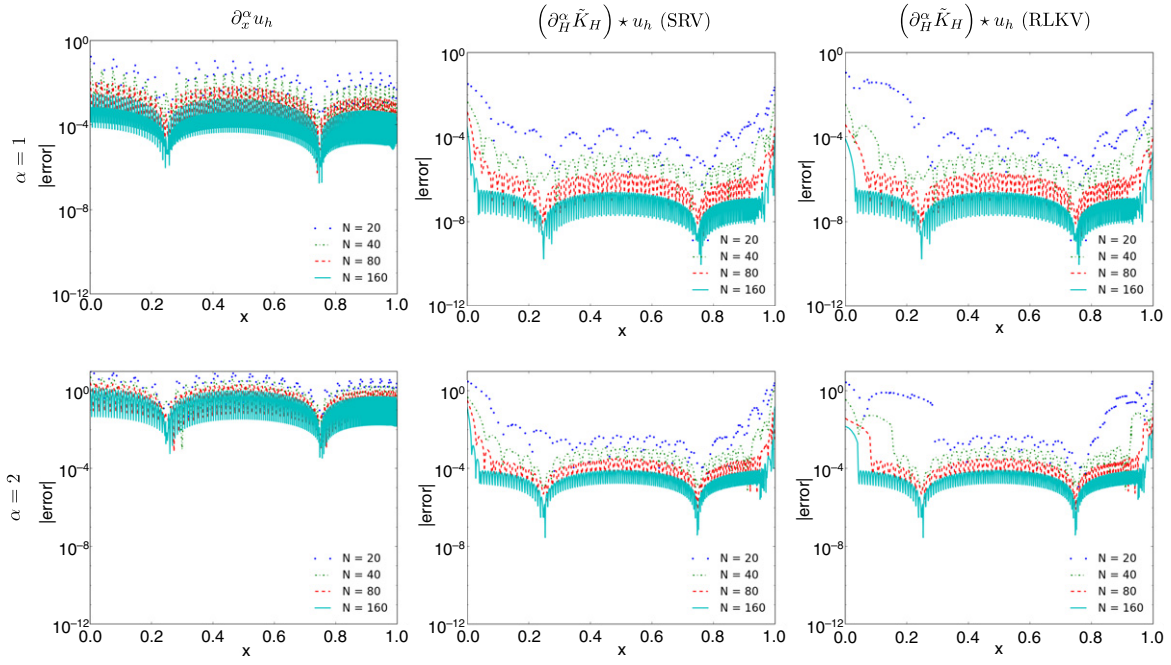

Fig. 4.3. Comparison of the point-wise errors in log scale of the first and second derivatives of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions (the SRV and RLKV filters) for the linear convection equation with Dirichlet boundary condition (4.2), over smoothly decreasing meshes. The B-spline order is $k+1+\alpha$, the filter scaling is taken as $H = \Delta x_j$, and $k = 2$.

4.2. Nonuniform mesh

Now we show the main results of this paper: the position-dependent derivative filtering over arbitrary nonuniform meshes. Before proceeding further, we first give the numerical setting of nonuniform meshes. In order to generate a more general format for nonuniform meshes, we use a random number generator to design the following two meshes.

Table 4.3
L^2- and L^∞-errors for the αth derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions (the SRV and RLKV filters) for the linear convection equation with Dirichlet boundary conditions (4.2), over smoothly decreasing meshes. The B-spline order is $k+1+\alpha$ and the filter scaling is taken as $H=\Delta x_j$.

Mesh	$\partial_x^\alpha u_h$				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (SRV)				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$(RLKV)			
	L^2 error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order
$\alpha=1$												
$\mathcal{P}^1$												
20	5.20E−01	–	1.63E+00	–	3.08E−02	–	1.57E−01	–	2.97E−02	–	1.63E−01	–
40	2.60E−01	1.00	8.31E−01	0.97	3.85E−03	3.00	2.33E−02	2.75	5.36E−03	2.47	4.38E−02	1.90
80	1.30E−01	1.00	4.14E−01	1.01	5.79E−04	2.74	5.28E−03	2.14	5.85E−04	3.20	5.22E−03	3.07
160	6.50E−02	1.00	2.08E−01	0.99	8.60E−05	2.75	7.98E−04	2.73	8.76E−05	2.74	7.47E−04	2.81
320	3.25E−02	1.00	1.04E−01	1.00	1.66E−05	2.37	1.08E−04	2.89	1.74E−05	2.33	1.37E−04	2.45
$\mathcal{P}^2$												
20	3.05E−02	–	1.73E−01	–	4.77E−03	–	3.21E−02	–	1.74E−02	–	1.13E−01	–
40	7.63E−03	2.00	4.47E−02	1.95	4.91E−04	3.28	4.50E−03	2.83	3.22E−04	5.76	3.40E−03	5.05
80	1.91E−03	2.00	1.13E−02	1.98	1.06E−04	2.22	1.51E−03	1.57	3.74E−05	3.11	3.78E−04	3.17
160	4.78E−04	2.00	2.83E−03	1.99	1.43E−05	2.88	2.88E−04	2.40	6.44E−06	2.54	8.38E−05	2.17
320	1.19E−04	2.00	7.09E−04	2.00	1.46E−06	3.30	4.14E−05	2.80	1.13E−06	2.51	2.16E−05	1.95
$\mathcal{P}^3$												
20	1.14E−03	–	6.86E−03	–	5.93E+00	–	5.04E+01	–	1.46E−02	–	9.61E−02	–
40	1.43E−04	3.00	8.53E−04	3.01	4.46E−05	17.02	7.69E−04	16.00	7.90E−06	10.85	7.09E−05	10.40
80	1.78E−05	3.00	1.07E−04	3.00	3.51E−06	3.67	5.11E−05	3.91	6.78E−08	6.86	8.36E−07	6.41
160	2.23E−06	3.00	1.33E−05	3.00	2.10E−07	4.06	4.43E−06	3.53	1.84E−09	5.21	2.88E−08	4.86
320	2.78E−07	3.00	1.66E−06	3.00	6.14E−09	5.10	1.83E−07	4.60	6.12E−11	4.91	1.54E−09	4.22
$\alpha=2$												
$\mathcal{P}^2$												
20	2.99E+00	–	1.02E+01	–	4.84E−01	–	2.97E+00	–	5.26E−01	–	2.91E+00	–
40	1.50E+00	1.00	5.22E+00	0.97	1.16E−01	2.07	1.74E+00	0.77	4.35E−02	3.60	3.51E−01	3.05
80	7.49E−01	1.00	2.62E+00	0.99	2.44E−02	2.24	3.27E−01	2.42	6.47E−03	2.75	5.53E−02	2.67
160	3.74E−01	1.00	1.31E+00	1.00	8.30E−03	1.56	1.42E−01	1.20	1.95E−03	1.73	1.43E−02	1.96
320	1.87E−01	1.00	6.58E−01	1.00	1.77E−03	2.23	4.55E−02	1.64	6.60E−04	1.56	8.25E−03	0.79
$\mathcal{P}^3$												
20	1.82E−01	–	8.31E−01	–	1.46E+03	–	1.24E+04	–	7.38E+01	–	3.08E+02	–
40	4.56E−02	2.00	2.16E−01	1.94	6.52E−03	17.78	7.99E−02	17.25	5.15E−03	13.81	4.61E−02	12.71
80	1.14E−02	2.00	5.39E−02	2.00	9.28E−04	2.81	1.40E−02	2.52	3.61E−05	7.16	4.62E−04	6.64
160	2.85E−03	2.00	1.35E−02	2.00	1.17E−04	2.99	2.34E−03	2.58	4.14E−06	3.12	5.19E−05	3.16
320	7.13E−04	2.00	3.36E−03	2.00	7.19E−06	4.02	2.04E−04	3.52	2.51E−07	4.05	5.04E−06	3.36
$\alpha=3$												
$\mathcal{P}^3$												
20	1.83E+01	–	5.46E+01	–	8.91E+03	–	6.73E+04	–	4.85E+05	–	2.93E+06	–
40	9.15E+00	1.00	2.78E+01	0.97	4.48E+00	10.96	7.35E+01	9.84	1.17E+00	18.67	1.65E+01	17.44
80	4.58E+00	1.00	1.39E+01	1.00	4.23E−01	3.41	9.44E+00	2.96	4.53E−02	4.68	8.75E−01	4.24
160	2.29E+00	1.00	6.96E+00	0.99	5.94E−02	2.83	1.12E+00	3.07	4.69E−03	3.27	9.29E−02	3.23
320	1.14E+00	1.00	3.48E+00	1.00	7.78E−03	2.93	2.08E−01	2.43	5.93E−04	2.98	1.09E−02	3.09

Mesh 4.3. The first nonuniform mesh that we consider is defined by

$$x_{\frac{1}{2}}=0,\qquad x_{N+\frac{1}{2}}=1,\qquad x_{j+\frac{1}{2}}=\left(j+b\cdot r_{j+\frac{1}{2}}\right)h,\quad j=1,\ldots,N-1$$

where $\left\{r_{j+\frac{1}{2}}\right\}_{j=1}^{N-1}$ are random numbers between $(-1,1)$, and $b\in(0,0.5]$ is a constant number. The size of the element $\Delta x_j=x_{j+\frac{1}{2}}-x_{j-\frac{1}{2}}$ is between $((1-2b)h,(1+2b)h)$. In order to save space, in this paper we present an example with $b=0.4$ only, other values of b such as 0.1, 0.2 and 0.45 were also calculated.

Mesh 4.4. The second nonuniform mesh is more irregular than the first one. We distribute the element interface by $x_{j+\frac{1}{2}}$, $j=1,\ldots,N-1$ randomly for the entire domain and require only

$$\Delta x_j=x_{j+\frac{1}{2}}-x_{j-\frac{1}{2}}\geq b\cdot h,\quad j=0,\ldots,N.$$

In this paper, we only present $b=0.5$ case for this mesh, other values of b such as 0.6, 0.8, etc. were also calculated.

Table 4.4

L^2- and L^∞-errors for the first derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solution $\partial_x^\alpha u_h^\star$ and $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (with the SRV filter) for the linear convection equation (4.1), over Mesh 4.3. The filter scaling is taken as $H = h^{2/5}$ near boundaries and $H = h^{2/3}$ for interior regions (where the symmetric filter is applied).

Mesh	$\partial_x^\alpha u_h$				$\partial_x^\alpha u_h^\star$ (SRV)				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$(SRV)			
	L^2 error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order
$\alpha = 1$												
$\mathcal{P}^1$												
20	5.48E−01	–	1.76E+00	–	2.85E−01	–	1.49E+00	–	5.91E−01	–	2.54E+00	–
40	2.82E−01	0.96	1.05E+00	0.74	2.63E−01	0.11	1.57E+00	−0.08	4.90E−01	0.27	2.37E+00	0.10
80	1.37E−01	1.05	4.98E−01	1.08	2.11E−01	0.32	1.52E+00	0.05	4.17E−01	0.24	2.38E+00	−0.00
160	6.72E−02	1.02	2.57E−01	0.96	1.29E−01	0.71	1.18E+00	0.36	3.08E−01	0.43	2.30E+00	0.05
320	3.38E−02	0.99	1.30E−01	0.98	3.12E−02	2.05	3.75E−01	1.65	9.55E−02	1.69	9.77E−01	1.24
$\mathcal{P}^2$												
20	3.56E−02	–	2.01E−01	–	1.15E−02	–	8.74E−02	–	3.30E−02	–	1.35E−01	–
40	8.96E−03	1.99	5.80E−02	1.79	2.32E−03	2.31	1.90E−02	2.20	4.21E−03	2.97	2.41E−02	2.49
80	1.96E−03	2.20	1.16E−02	2.32	2.08E−03	0.16	1.49E−02	0.35	3.70E−03	0.19	2.34E−02	0.05
160	4.86E−04	2.01	3.88E−03	1.58	1.68E−03	0.30	1.49E−02	−0.00	3.16E−03	0.23	2.35E−02	−0.01
320	1.32E−04	1.88	8.95E−04	2.11	1.36E−03	0.30	1.50E−02	−0.01	2.63E−03	0.27	2.36E−02	−0.01
$\mathcal{P}^3$												
20	1.53E−03	–	1.10E−02	–	1.33E−02	–	7.20E−02	–	4.03E−02	–	2.58E−01	–
40	2.10E−04	2.86	1.72E−03	2.68	7.05E−05	7.56	3.40E−04	7.73	4.44E−05	9.83	3.07E−04	9.72
80	2.27E−05	3.21	1.80E−04	3.26	4.84E−06	3.86	3.58E−05	3.25	6.48E−06	2.78	4.29E−05	2.84
160	2.72E−06	3.06	2.52E−05	2.84	3.30E−06	0.55	2.89E−05	0.31	5.84E−06	0.15	4.53E−05	−0.08
320	3.42E−07	2.99	3.22E−06	2.97	2.71E−06	0.28	2.91E−05	−0.01	4.97E−06	0.23	4.55E−05	−0.01

We remark that we have tested various differential equations over both kinds of nonuniform meshes: Meshs 4.3 and 4.4. However, the filtered approximations have similar performances since the performance mainly depends on the mesh. In order to save space, we choose to present one equation for each nonuniform mesh.

4.2.1. Comparison of the SRV filter and RLKV filter over nonuniform Mesh

In [8], the authors showed that the SRV filter has worse performance compared to the RLKV filter over nonuniform meshes for filtering the solution itself. We also mentioned that the enormous support size of the SRV filter causes problems: we have to rescale the SRV filter to fit the domain size. Hence, we cannot guarantee neither the accuracy order nor error reduction. Table 3.1 shows the minimum requirement of the number of elements for the SRV filter, and we can see that it is difficult to satisfy. Based on these deficiencies, we conclude that the SRV filter is not suitable for approximating derivatives over nonuniform meshes. However, in order to provide a complete view of the position-dependent derivative filters, we still give one example of using the SRV filter for the first derivative over Mesh 4.3. Table 4.4 shows that with the SRV filter, the filtered solutions (no matter what order of B-splines is used) are even worse than the derivative of the DG approximation. In the rest of this section, we focus on testing the RLKV filter over nonuniform meshes.

4.2.2. Linear equation over Mesh 4.3

For Mesh 4.3, we present results for the linear convection equation (4.1) with the first, second and third derivatives. The L^2 and L^∞ norm errors are given in Table 4.5 and Fig. 4.4 shows the point-wise error in log scale. When $\alpha \leq k$, both the derivative of filtered solution $\partial_x^\alpha u_h^\star$ and the filtered derivative $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ have better accuracy and convergence rates than the original DG approximation. The filtered derivative $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ shows better smoothness and theoretically has a better accuracy order than the derivative of the filtered solution $\partial_x^\alpha u_h^\star$ when $\alpha \leq k$, but $\partial_x^\alpha u_h^\star$ has better accuracy near the boundaries. For smoothness, the results are similar to the uniform mesh case; $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ has fewer oscillations compared to the DG solution and $\partial_x^\alpha u_h^\star$. Furthermore, we point out that by using higher-order B-splines we can disregard the requirement that $\alpha \leq k$.

For the point-wise error plots given in Fig. 4.4, the middle column is the filtered approximation $\partial_x^\alpha u_h^\star$, which shows more oscillations than the $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$, especially in the interior regions. We note, however that the support size of the filter that uses a higher-order B-spline increases with the derivative order α and it slightly increases the computational cost. Near the boundaries, the filtered solutions have a larger error magnitude than those in the interior region. Because near the boundaries we cannot obtain symmetric information around the filtered points, and the general B-spline has less regularity compared to the central B-spline. We note that the coarse meshes (such as $N = 20$ or even $N = 40$) are not sufficient to

Table 4.5

L^2- and L^∞-errors for the αth derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions $\partial_x^\alpha u_h^\star$ and $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (with the RLKV filter) for the linear convection equation (4.1), over Mesh 4.3. The filter scaling is taken as $H = h^{2/3}$.

Mesh	$\partial_x^\alpha u_h$				$\partial_x^\alpha u_h^\star$ (RLKV)				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$(RLKV)			
	L^2 error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order
$\alpha = 1$												
$\mathcal{P}^1$												
20	5.48E−01	–	1.76E+00	–	4.19E−02	–	1.52E−01	–	5.36E−02	–	9.92E−02	–
40	2.82E−01	0.96	1.05E+00	0.74	8.18E−03	2.36	3.36E−02	2.18	1.14E−02	2.23	4.02E−02	1.30
80	1.37E−01	1.05	4.98E−01	1.08	1.89E−03	2.11	6.12E−03	2.46	2.19E−03	2.38	8.22E−03	2.29
160	6.72E−02	1.02	2.57E−01	0.96	4.93E−04	1.94	2.09E−03	1.55	3.51E−04	2.64	1.44E−03	2.51
320	3.38E−02	0.99	1.30E−01	0.98	1.46E−04	1.76	6.15E−04	1.76	5.04E−05	2.80	2.35E−04	2.61
$\mathcal{P}^2$												
20	3.56E−02	–	2.01E−01	–	3.13E−02	–	9.60E−02	–	5.84E−02	–	1.32E−01	–
40	8.96E−03	1.99	5.80E−02	1.79	3.22E−04	6.60	1.26E−03	6.25	1.04E−03	5.82	2.45E−03	5.75
80	1.96E−03	2.20	1.16E−02	2.32	7.59E−05	2.08	4.26E−04	1.57	1.78E−04	2.54	6.73E−04	1.86
160	4.86E−04	2.01	3.88E−03	1.58	5.28E−06	3.85	3.78E−05	3.49	1.46E−05	3.61	6.65E−05	3.34
320	1.32E−04	1.88	8.95E−04	2.11	3.20E−07	4.04	2.60E−06	3.86	8.71E−07	4.07	4.83E−06	3.78
$\mathcal{P}^3$												
20	1.53E−03	–	1.10E−02	–	4.08E−03	–	1.23E−02	–	5.34E−03	–	1.42E−02	–
40	2.10E−04	2.86	1.72E−03	2.68	8.01E−04	2.35	2.63E−03	2.22	2.89E−03	0.88	7.96E−03	0.83
80	2.27E−05	3.21	1.80E−04	3.26	4.79E−06	7.38	2.39E−05	6.78	3.10E−06	9.87	1.44E−05	9.11
160	2.72E−06	3.06	2.52E−05	2.84	3.62E−07	3.73	2.03E−06	3.56	9.36E−07	1.73	4.15E−06	1.79
320	3.42E−07	2.99	3.22E−06	2.97	9.64E−09	5.23	6.87E−08	4.89	2.71E−08	5.11	1.51E−07	4.78
$\alpha = 2$												
$\mathcal{P}^1$												
20	–	–	–	–	–	–	–	–	1.94E+00	–	9.16E+00	–
40	–	–	–	–	–	–	–	–	2.63E−01	2.89	1.51E+00	2.60
80	–	–	–	–	–	–	–	–	3.42E−02	2.94	1.99E−01	2.93
160	–	–	–	–	–	–	–	–	6.39E−03	2.42	2.11E−02	3.23
320	–	–	–	–	–	–	–	–	2.19E−03	1.54	8.55E−03	1.30
$\mathcal{P}^2$												
20	3.16E+00	–	9.99E+00	–	3.19E−01	–	1.83E+00	–	3.42E−01	–	1.80E+00	–
40	1.60E+00	0.98	5.79E+00	0.79	2.87E−02	3.47	2.05E−01	3.16	1.00E−01	1.77	5.59E−01	1.69
80	7.57E−01	1.08	2.60E+00	1.16	1.11E−03	4.69	1.21E−02	4.07	5.25E−03	4.25	3.72E−02	3.91
160	3.78E−01	1.00	1.52E+00	0.78	5.10E−04	1.12	5.79E−03	1.07	2.07E−04	4.66	1.96E−03	4.24
320	1.96E−01	0.94	7.38E−01	1.04	2.65E−05	4.26	2.02E−04	4.85	8.54E−06	4.60	4.75E−05	5.37
$\mathcal{P}^3$												
20	2.15E−01	–	1.12E+00	–	2.11E−02	–	1.39E−01	–	2.21E−02	–	1.21E−01	–
40	5.70E−02	1.92	3.46E−01	1.70	9.02E−03	1.23	6.45E−02	1.11	2.23E−02	−0.01	1.25E−01	−0.06
80	1.31E−02	2.12	7.71E−02	2.17	2.40E−04	5.23	2.13E−03	4.92	1.15E−03	4.28	7.69E−03	4.03
160	3.17E−03	2.05	2.05E−02	1.91	3.76E−06	6.00	4.21E−05	5.66	2.06E−05	5.80	1.71E−04	5.49
320	7.98E−04	1.99	5.27E−03	1.96	5.25E−08	6.16	7.68E−07	5.78	2.93E−07	6.14	3.10E−06	5.79
$\alpha = 3$												
$\mathcal{P}^2$												
20	–	–	–	–	–	–	–	–	4.98E+00	–	2.58E+01	–
40	–	–	–	–	–	–	–	–	1.01E+00	2.31	5.55E+00	2.22
80	–	–	–	–	–	–	–	–	3.06E−02	5.04	2.73E−01	4.35
160	–	–	–	–	–	–	–	–	3.25E−03	3.24	2.95E−02	3.21
320	–	–	–	–	–	–	–	–	1.64E−03	0.99	1.66E−02	0.83
$\mathcal{P}^3$												
20	1.95E+01	–	5.77E+01	–	2.74E−01	–	1.78E+00	–	3.40E−01	–	2.21E+00	–
40	9.94E+00	0.97	3.54E+01	0.70	7.63E−02	1.84	5.36E−01	1.73	3.45E−01	−0.02	2.12E+00	0.06
80	4.81E+00	1.05	1.67E+01	1.08	1.59E−03	5.59	1.51E−02	5.15	5.12E−03	6.07	3.62E−02	5.87
160	2.37E+00	1.02	8.62E+00	0.96	3.54E−04	2.17	4.15E−03	1.87	1.81E−04	4.83	1.73E−03	4.39
320	1.19E+00	0.99	4.37E+00	0.98	1.42E−05	4.64	2.12E−04	4.29	6.44E−06	4.81	7.71E−05	4.49

use the position-dependent derivative filter, the filtered solution may have worse accuracy compared to the original DG approximation.

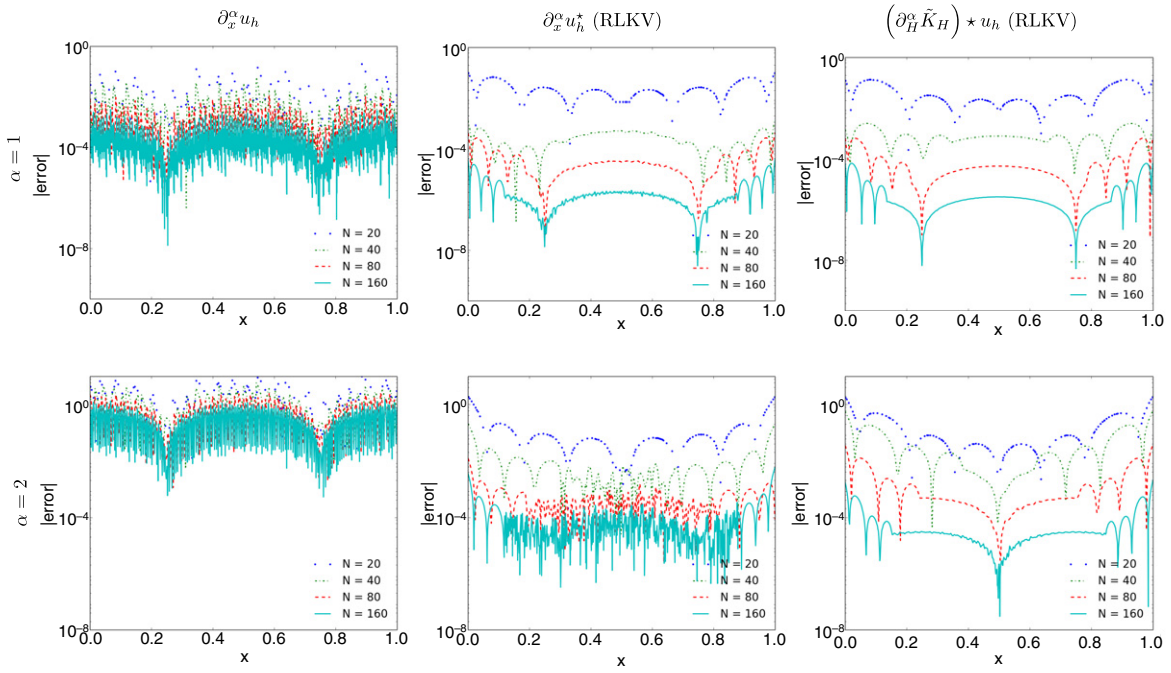

Fig. 4.4. Comparison of the point-wise errors in log scale of the first and second derivatives of DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions $\partial_x^\alpha u_h^\star$ and $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (with the RLKV filter) for the linear convection equation (4.1) over Mesh 4.3. The filter scaling is taken as $H = h^{2/3}$, and $k = 2$.

4.2.3. Variable coefficient equation over Mesh 4.4

After testing the linear convection equation (4.1), we move to a variable coefficient equation with periodic boundary conditions,

$$\begin{aligned} &u_t + (au)_x = f, \quad (x, t) \in [0, 1] \times (0, T] \\ &u(x, 0) = \sin(2\pi x), \end{aligned} \tag{4.3}$$

where the variable coefficient $a(x, t) = 2 + \sin(2\pi(x + t))$ and the right side term $f(x, t)$ are chosen to make the exact solution

$$u(x, t) = \sin(2\pi(x - t)).$$

As with the linear convection example, we present the L^2 and L^∞ errors in Table 4.6 with the first three derivatives over Mesh 4.4. The respective point-wise error plots ($k = 2$ case) are shown in Fig. 4.5. The results are similar to the results for the constant coefficient case. In order to save space we no longer repeat the descriptions, which are similar. However, we still want to point out one phenomenon. In this variable coefficient case, the filtered solution $\partial_x^\alpha u_h^\star$ shows somewhat better accuracy than the filtered solution $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ near the boundaries when $\alpha \leq k$. This performance suggests that when $\alpha \leq k$ we can consider not increasing the order of the B-splines, although it causes more oscillations in the error.

Remark 4.5. Here we summarize the consequences of using B-splines of order $k + 1$ compared to using normal order $k + 1 + \alpha$; they are the following:

- it can give better accuracy near the boundaries;
- it can give better accuracy in the interior regions (when $\alpha \ll k$), but it damages the smoothness of filtered solution (more oscillations);
- it has a smaller support size;
- it allows the use of the symmetric filter over a larger area; and
- it requires $\alpha \leq k$.

Remark 4.6. We notice that in Tables 4.5 and 4.6, the accuracy order is smaller than the conclusion in Theorem 3.4. The reason is twofold:

- The first reason is the effect of the boundary region. The error magnitude of the filtered solution in the interior region is much better than the error magnitude in the boundary region. Therefore, the accuracy order in the L^2 norm appears to be unstable and the numbers are smaller than the theoretical expectation. If we look at Figs. 4.4 and 4.5 (right column), the convergence rates are stable respective to boundary and interior regions separately.

Table 4.6

L^2- and L^∞-errors for the αth derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions $\partial_x^\alpha u_h^\star$ and $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (with the RLKV filter) for variable coefficient equation (4.3), over Mesh 4.4. The filter scaling is taken as $H = h^{2/3}$.

Mesh	$\partial_x^\alpha u_h$				$\partial_x^\alpha u_h^\star$ (RLKV)				$\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$(RLKV)			
	L^2 error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order	L^2error	Order	L^∞error	Order
$\alpha = 1$												
$\mathcal{P}^1$												
20	5.73E−01	–	2.04E+00	–	4.28E−02	–	9.17E−02	–	4.04E−02	–	8.51E−02	–
40	2.76E−01	1.05	9.98E−01	1.03	1.29E−02	1.73	6.62E−02	0.47	1.47E−02	1.46	5.53E−02	0.62
80	1.53E−01	0.85	6.35E−01	0.65	3.44E−03	1.91	1.31E−02	2.34	2.72E−03	2.43	7.98E−03	2.79
160	7.16E−02	1.10	3.43E−01	0.89	1.01E−03	1.76	5.63E−03	1.22	6.26E−04	2.12	1.82E−03	2.13
320	4.07E−02	0.82	2.46E−01	0.48	9.81E−04	0.05	8.33E−03	−0.57	6.37E−04	−0.02	3.11E−03	−0.77
$\mathcal{P}^2$												
20	6.60E−02	–	4.27E−01	–	3.38E−02	–	1.10E−01	–	6.16E−02	–	1.42E−01	–
40	1.27E−02	2.38	9.22E−02	2.21	3.17E−04	6.73	1.26E−03	6.45	9.68E−04	5.99	2.46E−03	5.85
80	2.12E−03	2.58	1.27E−02	2.87	7.71E−05	2.04	4.28E−04	1.55	1.78E−04	2.44	6.61E−04	1.89
160	6.40E−04	1.73	5.66E−03	1.16	5.33E−06	3.85	3.71E−05	3.53	1.47E−05	3.60	6.54E−05	3.34
320	2.48E−04	1.37	2.99E−03	0.92	4.67E−07	3.51	2.70E−06	3.78	8.72E−07	4.08	4.93E−06	3.73
$\mathcal{P}^3$												
20	2.95E−03	–	2.29E−02	–	4.28E−03	–	1.29E−02	–	5.72E−03	–	1.52E−02	–
40	2.88E−04	3.35	2.84E−03	3.01	8.01E−04	2.42	2.61E−03	2.31	2.89E−03	0.98	7.97E−03	0.93
80	3.95E−05	2.87	3.84E−04	2.89	4.82E−06	7.38	2.36E−05	6.79	3.10E−06	9.87	1.41E−05	9.15
160	4.74E−06	3.06	6.15E−05	2.64	3.62E−07	3.73	2.05E−06	3.53	9.36E−07	1.73	4.15E−06	1.76
320	1.28E−06	1.89	2.14E−05	1.52	9.64E−09	5.23	6.95E−08	4.88	2.71E−08	5.11	1.51E−07	4.78
$\alpha = 2$												
$\mathcal{P}^1$												
20	–	–	–	–	–	–	–	–	2.45E+00	–	1.14E+01	–
40	–	–	–	–	–	–	–	–	4.39E−01	2.48	2.61E+00	2.12
80	–	–	–	–	–	–	–	–	6.57E−02	2.74	2.23E−01	3.55
160	–	–	–	–	–	–	–	–	3.48E−02	0.92	1.13E−01	0.98
320	–	–	–	–	–	–	–	–	5.99E−02	−0.78	2.80E−01	−1.31
$\mathcal{P}^2$												
20	3.89E+00	–	1.40E+01	–	3.34E−01	–	2.36E+00	–	3.39E−01	–	2.26E+00	–
40	1.78E+00	1.13	6.49E+00	1.10	2.93E−02	3.51	2.23E−01	3.40	1.03E−01	1.72	5.84E−01	1.95
80	7.66E−01	1.22	2.54E+00	1.35	1.11E−03	4.72	9.19E−03	4.60	5.31E−03	4.28	3.73E−02	3.97
160	4.10E−01	0.90	1.86E+00	0.45	2.28E−04	2.28	1.26E−03	2.87	1.86E−04	4.84	1.62E−03	4.52
320	2.28E−01	0.85	1.36E+00	0.45	2.00E−04	0.19	1.38E−03	−0.14	1.39E−05	3.74	8.73E−05	4.22
$\mathcal{P}^3$												
20	2.89E−01	–	1.67E+00	–	2.16E−02	–	1.63E−01	–	1.84E−02	–	1.13E−01	–
40	6.24E−02	2.21	4.20E−01	1.99	9.03E−03	1.26	6.49E−02	1.33	2.22E−02	−0.28	1.25E−01	−0.15
80	1.80E−02	1.79	1.27E−01	1.72	2.35E−04	5.27	2.09E−03	4.96	1.15E−03	4.28	7.60E−03	4.04
160	4.12E−03	2.13	3.67E−02	1.79	3.66E−06	6.00	4.28E−05	5.61	2.06E−05	5.80	1.73E−04	5.46
320	1.51E−03	1.45	1.79E−02	1.03	1.06E−07	5.10	7.61E−07	5.81	2.93E−07	6.14	3.11E−06	5.80
$\alpha = 3$												
$\mathcal{P}^2$												
20	–	–	–	–	–	–	–	–	6.75E+00	–	4.36E+01	–
40	–	–	–	–	–	–	–	–	1.18E+00	2.51	7.23E+00	2.59
80	–	–	–	–	–	–	–	–	2.36E−02	5.65	2.19E−01	5.04
160	–	–	–	–	–	–	–	–	7.55E−03	1.64	7.38E−02	1.57
320	–	–	–	–	–	–	–	–	1.28E−03	2.57	7.79E−03	3.24
$\mathcal{P}^3$												
20	2.04E+01	–	6.86E+01	–	1.68E+00	–	1.10E+01	–	2.37E+00	–	1.56E+01	–
40	9.76E+00	1.06	3.31E+01	1.05	9.36E−02	4.17	6.06E−01	4.18	3.52E−01	2.75	2.16E+00	2.85
80	5.39E+00	0.86	2.14E+01	0.63	4.87E−04	7.59	3.77E−03	7.33	5.15E−03	6.10	3.73E−02	5.86
160	2.52E+00	1.10	1.15E+01	0.89	1.55E−04	1.65	1.84E−03	1.04	1.76E−04	4.87	1.73E−03	4.43
320	1.39E+00	0.86	7.30E+00	0.66	6.97E−05	1.16	5.24E−04	1.81	6.60E−06	4.74	7.91E−05	4.45

- The second reason is that the Meshs 4.3 and 4.4 are randomly generated. There is no strict refinement relation among the meshes. Therefore, the accuracy order is affected by the randomness of the meshes, and a very stable accuracy order is

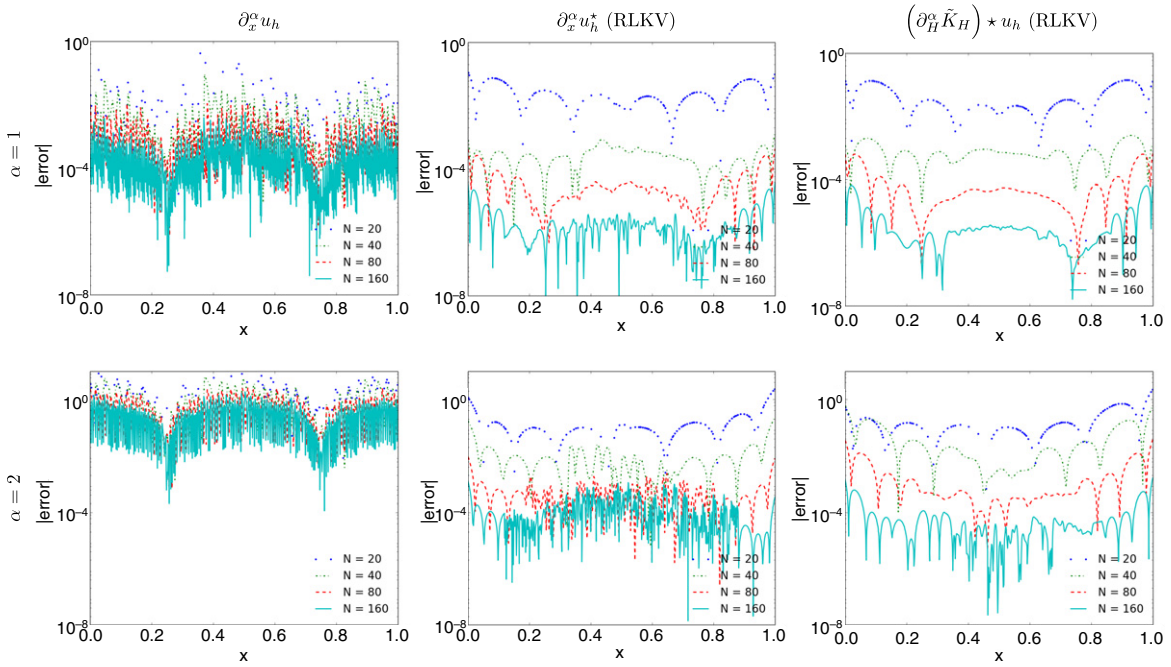

Fig. 4.5. Comparison of the point-wise errors in log scale of the first and second derivative of the DG approximation $\partial_x^\alpha u_h$ together with the two filtered solutions $\partial_x^\alpha u_h^\star$ and $\left(\partial_H^\alpha \tilde{K}_H\right) \star u_h$ (with the RLKV filter) for variable coefficient equation (4.3) over Mesh 4.4. The filter scaling is taken as $H = h^{2/3}$, and $k = 2$.

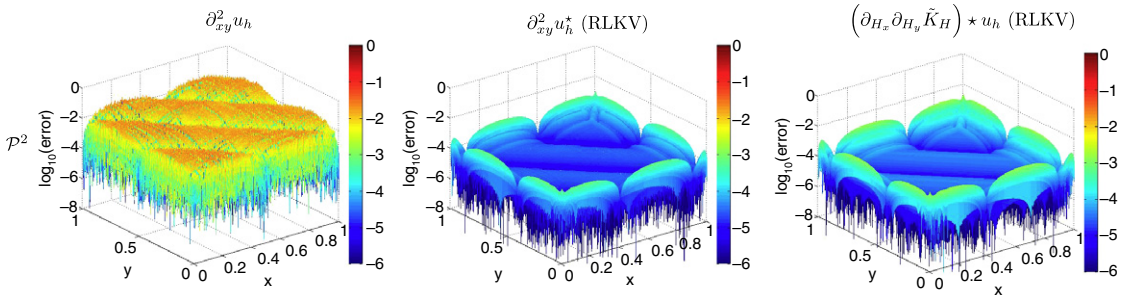

Fig. 5.1. Comparison of the point-wise errors in log scale of the cross-derivative DG approximation $\partial_{xy}^2 u_h$ together with the filtered solutions for the two-dimensional linear advection equation (5.1) over Mesh 4.3 (2D, $N = 160 \times 160$).

not observed. One can see that once the effect caused by randomness becomes smaller in the two-dimensional example, the accuracy order becomes stable, see Tables 5.1 and 5.2.

5. Two-dimensional example

For the two-dimensional example, we consider a 2D version of the linear convection equation

$$u_t + u_x + u_y = 0, \quad (x, y) \in [0, 1] \times [0, 1], \qquad (5.1)$$
$$u(x, y, 0) = \sin(2\pi x + 2\pi y),$$

at time $T = 1$ with periodic boundary conditions. The nonuniform meshes we used are the 2D quadrilateral extension of Meshs 4.3 and 4.4. Here, we show the cross-derivative ∂_{xy}^2, the first derivatives ∂_x and ∂_y are omitted as they are similar to the 1D results. We give the L^2 and L^∞ error in Tables 5.1–5.2 and the point-wise error plots in Figs. 5.1–5.2. We note that the filtered accuracy error seems slightly worse than the DG approximation over coarse meshes, because near the boundary regions we need sufficiently refined meshes to show the advantage of the position-dependent filter. Once the mesh is sufficiently refined, we see better results. We also note that although we require a relatively refined mesh for boundary regions, the results in the interior regions behave better (see the point-wise error plots).

Table 5.1
L^2- and L^∞-errors for the cross-derivative DG approximation $\partial^2_{xy} u_h$ together with the filtered solutions for the two-dimensional linear convection equation (5.1) over Mesh 4.3 (2D).

Mesh	$\partial^2_{xy} u_h$				$\partial^2_{xy} u_h^\star$ (RLKV)				$(\partial_{H_x}\partial_{H_y}\tilde{K}_H) \star u_h$ (RLKV)			
	L^2 error	Order	L^∞ error	Order	L^2 error	Order	L^∞ error	Order	L^2 error	Order	L^∞ error	Order
$\mathcal{P}^1$												
20×20	5.47E+00	–	2.32E+01	–	–	–	–	–	1.32E+00	–	1.25E+01	–
40×40	2.71E+00	1.01	1.33E+01	0.81	–	–	–	–	1.97E−01	2.75	1.80E+00	2.79
80×80	1.33E+00	1.03	6.39E+00	1.06	–	–	–	–	2.81E−02	2.81	2.56E−01	2.81
160×160	6.62E−01	1.00	3.38E+00	0.92	–	–	–	–	4.08E−03	2.78	3.39E−02	2.92
$\mathcal{P}^2$												
20×20	3.48E−01	–	2.49E+00	–	4.68E−01	–	3.44E+00	–	5.66E−01	–	3.59E+00	–
40×40	8.16E−02	2.09	7.13E−01	1.80	2.65E−02	4.14	3.63E−01	3.24	5.38E−02	3.40	6.81E−01	2.40
80×80	1.93E−02	2.08	1.81E−01	1.98	1.38E−03	4.26	1.96E−02	4.22	2.83E−03	4.25	4.03E−02	4.08
160×160	4.79E−03	2.01	4.53E−02	2.00	6.86E−05	4.33	8.74E−04	4.48	1.44E−04	4.30	1.84E−03	4.45
$\mathcal{P}^3$												
20×20	1.54E−02	–	1.47E−01	–	4.11E−02	–	2.79E−01	–	4.06E−02	–	2.63E−01	–
40×40	1.75E−03	3.13	2.29E−02	2.68	1.14E−02	1.85	1.27E−01	1.13	2.45E−02	0.73	2.04E−01	0.37
80×80	2.00E−04	3.13	2.51E−03	3.19	2.42E−04	5.56	4.26E−03	4.90	5.30E−04	5.53	8.51E−03	4.58
160×160	2.47E−05	3.02	3.58E−04	2.81	4.91E−06	5.62	8.59E−05	5.63	1.08E−05	5.62	1.81E−04	5.56

Table 5.2
L^2- and L^∞-errors for the cross-derivative DG approximation $\partial^2_{xy} u_h$ together with the filtered solutions for the two-dimensional linear advection equation (5.1) over Mesh 4.4 (2D).

Mesh	$\partial^2_{xy} u_h$				$\partial^2_{xy} u_h^\star$ (RLKV)				$(\partial_{H_x}\partial_{H_y}\tilde{K}_H) \star u_h$ (RLKV)			
	L^2 error	Order	L^∞ error	Order	L^2 error	Order	L^∞ error	Order	L^2 error	Order	L^∞ error	Order
$\mathcal{P}^1$												
20×20	6.03E+00	–	2.94E+01	–	–	–	–	–	1.53E+00	–	1.27E+01	–
40×40	3.20E+00	0.91	1.95E+01	0.59	–	–	–	–	2.72E−01	2.50	2.07E+00	2.62
80×80	1.61E+00	0.99	1.09E+01	0.84	–	–	–	–	5.44E−02	2.32	4.01E−01	2.37
160×160	7.39E−01	1.12	5.18E+00	1.07	–	–	–	–	8.35E−03	2.70	1.37E−01	1.55
$\mathcal{P}^2$												
20×20	5.60E−01	–	7.00E+00	–	4.73E−01	–	3.48E+00	–	5.68E−01	–	3.59E+00	–
40×40	1.68E−01	1.73	2.65E+00	1.40	2.67E−02	4.15	3.67E−01	3.24	5.34E−02	3.41	6.88E−01	2.38
80×80	3.85E−02	2.13	5.30E−01	2.32	1.36E−03	4.29	1.94E−02	4.24	2.98E−03	4.16	4.12E−02	4.06
160×160	7.27E−03	2.41	1.03E−01	2.37	7.90E−05	4.11	1.05E−03	4.21	1.54E−04	4.27	1.89E−03	4.44
$\mathcal{P}^3$												
20×20	4.02E−02	–	3.66E−01	–	4.12E−02	–	2.79E−01	–	4.05E−02	–	2.64E−01	–
40×40	7.60E−03	2.40	1.03E−01	1.83	1.14E−02	1.86	1.29E−01	1.12	2.45E−02	0.72	2.06E−01	0.36
80×80	7.71E−04	3.30	1.68E−02	2.61	2.42E−04	5.55	4.21E−03	4.93	5.30E−04	5.53	8.40E−03	4.62
160×160	5.76E−05	3.74	1.31E−03	3.69	4.91E−06	5.62	8.65E−05	5.60	1.08E−05	5.62	1.82E−04	5.51

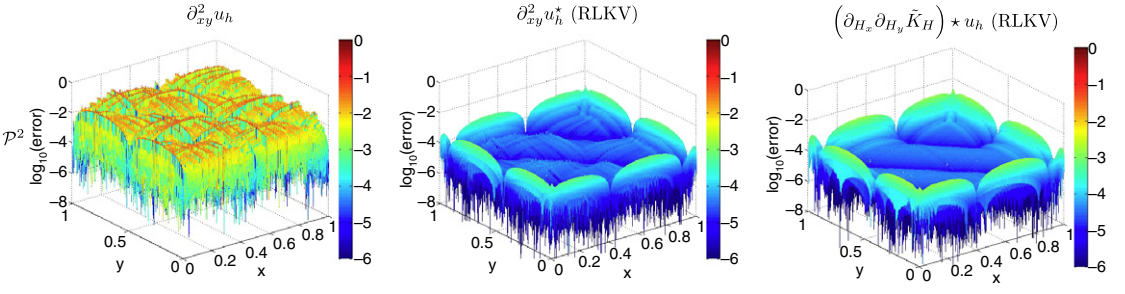

Fig. 5.2. Comparison of the point-wise errors in log scale of the cross-derivative DG approximation $\partial^2_{xy} u_h$ together with the filtered solutions for the two-dimensional linear advection equation (5.1) over Mesh 4.4 (2D, $N = 160 \times 160$).

6. Conclusion

In this paper, we have proposed two position-dependent derivative filters (the SRV and RLKV filter), to approximate the derivatives of the discontinuous Galerkin solutions over uniform and nonuniform meshes. These position-dependent

derivative filters allow us to obtain more accurate derivatives of the DG solutions compared to calculating the derivatives of DG solutions directly. The derivative SRV filter uses $4k + 1$ central B-splines, and obtains a convergence rate of $2k + 1$ over uniform meshes regardless of derivative order. The derivative RLKV filter uses $2k + 1$ central B-splines and an extra general B-spline, where the general B-spline relies on the derivative order α. We have proved that the derivative RLKV filter has accuracy order of $\mu(2k + 2)$ when using a filter scaling $H = h^{\mu}$ ($\mu \approx 2/3$). Additionally, we are able, for the first time, to extend the symmetric derivative filter to nonuniform meshes. Through numerical examples, we compared the derivative SRV and RLKV filter over uniform and nonuniform meshes. We demonstrated that once the required conditions are satisfied the derivative SRV filter has a better performance over uniform meshes compared to the derivative RLKV filter. However, for nonuniform meshes, only the derivative RLKV filter can maintain its performance and improve the accuracy of the DG approximations. Also, we compared derivative filters with different order of B-splines: order $k + 1$ and order $k + 1 + \alpha$. Numerical results indicate that using B-splines of order $k + 1$ may improve the accuracy of the filtered solution near the boundaries. For interior regions where the symmetric derivative filtering is applied, using B-splines of order $k + 1 + \alpha$ shows better accuracy and smoothness. Lastly, we point out that for given nonuniform meshes there may exist a better scaling that allows us to get better results. Our future work will concentrate on investigating better methods for filtering over nonuniform meshes and extending the position-dependent derivative filtering to unstructured triangular meshes.

Acknowledgments

The first and second authors are sponsored by the Air Force Office of Scientific Research (AFOSR), Air Force Material Command, USAF, under grant number FA8655-09-1-3017. The third author is sponsored in part by the Air Force Office of Scientific Research (AFOSR), Computational Mathematics Program (Program Manager: Dr. Fariba Fahroo), under grant number FA9550-12-1-0428. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] M. Steffen, S. Curtis, R.M. Kirby, J.K. Ryan, Investigation of smoothness-increasing accuracy-conserving filters for improving streamline integration through discontinuous fields, IEEE Trans. Vis. Comput. Graphics 14 (3) (2008) 680–692.
[2] J.H. Bramble, A.H. Schatz, Higher order local accuracy by averaging in the finite element method, Math. Comp. 31 (137) (1977) 94–111.
[3] B. Cockburn, M. Luskin, C.-W. Shu, E. Süli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations, Math. Comp. 72 (242) (2003) 577–606.
[4] V. Thomée, High order local approximations to derivatives in the finite element method, Math. Comp. 31 (1977) 652–660.
[5] J.K. Ryan, B. Cockburn, Local derivative post-processing for the discontinuous Galerkin method, J. Comput. Phys. 228 (23) (2009) 8642–8664.
[6] S. Curtis, R.M. Kirby, J.K. Ryan, C.-W. Shu, Postprocessing for the discontinuous Galerkin method over nonuniform meshes, SIAM J. Sci. Comput. 30 (1) (2008) 272–289.
[7] J. Ryan, Local Derivative Post-processing: Challenges for a non-uniform mesh, Delft University of Technology Report 10–18.
[8] J. Ryan, X. Li, R.M. Kirby, C. Vuik, One-sided position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering over uniform and non-uniform meshes, J. Sci. Comput. 64 (3) (2015) 773–817.
[9] J. Ryan, C.W. Shu, On a one-sided post-processing technique for the discontinuous Galerkin methods, Methods Appl. Anal. 10 (2) (2003) 295–308.
[10] P. van Slingerland, J.K. Ryan, C. Vuik, Position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering for improving discontinuous Galerkin solutions, SIAM J. Sci. Comput. 33 (2) (2011) 802–825.
[11] B. Cockburn, An introduction to the discontinuous Galerkin method for convection-dominated problems, in: A. Quarteroni (Ed.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, in: Lecture Notes in Mathematics, vol. 1697, Springer, Berlin, Heidelberg, 1998, pp. 151–268.
[12] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.
[13] C. de Boor, A Practical Guide to Splines, Springer-Verlag, New York, 2001.
[14] M.S. Mock, P.D. Lax, The computation of discontinuous solutions of linear hyperbolic equations, Comm. Pure Appl. Math. 31 (4) (1978) 423–430.