Journal of Computational Physics 228 (2009) 8642–8664

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Local derivative post-processing for the discontinuous Galerkin method

Jennifer K. Ryan [a,*], Bernardo Cockburn [b]

[a] Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands
[b] School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 26 December 2008
Received in revised form 5 June 2009
Accepted 21 August 2009
Available online 1 September 2009

MSC:
65M60

Keywords:
Accuracy enhancement
Post-processing
Discontinuous Galerkin method
Hyperbolic equations

ABSTRACT

Obtaining accurate approximations for derivatives is important for many scientific applications in such areas as fluid mechanics and chemistry as well as in visualization applications. In this paper we discuss techniques for computing accurate approximations of high-order derivatives for discontinuous Galerkin solutions to hyperbolic equations related to these areas. In previous work, improvement in the accuracy of the numerical solution using discontinuous Galerkin methods was obtained through post-processing by convolution with a suitably defined kernel. This post-processing technique was able to improve the order of accuracy of the approximation to the solution of time-dependent symmetric linear hyperbolic partial differential equations from order $k+1$ to order $2k+1$ over a uniform mesh; this was extended to include one-sided post-processing as well as post-processing over non-uniform meshes. In this paper, we address the issue of improving the accuracy of approximations to derivatives of the solution by using the method introduced by Thomée [19]. It consists in simply taking the αth-derivative of the convolution of the solution with a sufficiently smooth kernel. The order of convergence of the approximation is then *independent* of the order of the derivative, $|\alpha|$. We also discuss an efficient way of computing the approximation which does not involve differentiation but the application of simple finite differencing. Our results show that the above-mentioned approximations to the αth-derivative of the exact solution of linear, multidimensional symmetric hyperbolic systems obtained by the discontinuous Galerkin method with polynomials of degree k converge with order $2k+1$ regardless of the order $|\alpha|$ of the derivative.

1. Introduction

Obtaining accurate approximations of derivatives is important in many applications arising in such areas as continuum mechanics, fluid mechanics and chemistry. It is also important in visualization for volume classification shading [14]. Motivated by this variety of practical applications, we study the problem of obtaining accurate approximations of high-order derivatives of the solution having computed an approximation of the solution *only*.

In [2], a technique to enhance the approximation to the solution was introduced in the framework of finite element methods for second-order elliptic problems. It consisted of a simple convolution with a kernel whose support had a diameter of the order of the element size; locally uniform meshes were required. It allowed for the post-processed approximation to obtain the same order of convergence of the negative-order norms as the error. Thus, if polynomials of degree k were used to define the numerical approximation, the post-processed solution converged with order $2k$. This approach was later applied to discontinuous Galerkin methods for hyperbolic problems in [5,6] for uniform meshes, in [15,16] for piecewise-uniform meshes, and in [12] for non-uniform meshes. When polynomials of degree k are used, the post-processed solution was

* Corresponding author. Tel.: +31 15 27 89755.
E-mail addresses: J.K.Ryan@tudelft.nl (J.K. Ryan), cockburn@math.umn.edu (B. Cockburn).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.08.017

shown to converge with order $2k+1$ in the first two cases. In [15], approximations for the derivatives were obtained by simply differentiating the above-mentioned convolution. However, the corresponding order of convergence eventually began to diminish as the order of the derivative increased. In this paper, we show how to avoid this unpleasant phenomenon by the application of the approach proposed by Thomée [19]; it results in approximate derivatives converging with order $2k+1$ independently of the order of the derivative. We do this in the framework of discontinuous Galerkin approximations for hyperbolic problems.

The outline of this paper is as follows. In the next section, we introduce our model problem, namely, the one-dimensional transport equation, as well as its discontinuous Galerkin approximation in space. In Section 3, we discuss Thomée's technique for derivative post-processing and prove that the approximate derivatives converge with order $2k+1$ regardless of the order of the derivative. We then describe how to efficiently implement the corresponding post-processors. We end by briefly indicating how to extend these results to general linear, multidimensional hyperbolic systems. In Section 4, we present numerical results demonstrating the performance of the post-processing. We end with some concluding remarks.

2. The model problem and its discretization

The discontinuous Galerkin method continues to gain in popularity due to flexibility in adaptivity, simple treatment of boundary conditions, ability to handle complicated geometries, and a local nature that makes it easy to implement in parallel. It has particular advantages when it is used to solve hyperbolic conservation laws and is well studied [7,8,4,3,9–11]. For the sake of simplicity, we take as a model problem the simple transport equation

$$u_t + (au)_x = 0 \quad \text{in } \mathbb{R}\times(0,T), \tag{2.1a}$$
$$u(\cdot, t=0) = u_o(\cdot) \quad \text{on } \mathbb{R}, \tag{2.1b}$$

where a is a positive constant.

We start by defining a uniform mesh $I_i = (x_i - \frac{h}{2}, x_i + \frac{h}{2}), i \in \mathbb{Z}$, where h is the uniform element size. Next, we choose an approximation space, V_h, to be the space of piecewise polynomials of degree less than or equal to k on each interval I_i. The approximate solution u_h given by the DG method is taken in the space V_h at each time, and is set equal to the L^2-projection of the initial data u_o at time $t=0$. For $t>0$, it is determined as the solution of the formulation

$$\int_{I_i} (u_h)_t\, v\, \mathrm{d}x = \int_{I_i} a u_h v_x\, \mathrm{d}x - a\hat{u}_{i+1/2} v^-_{i+1/2} + a\hat{u}_{i-1/2} v^+_{i-1/2},$$

for all $v \in V_h$. Here $\hat{u}_{i+1/2} := u_h\left(x^-_{i+1/2}\right)$ is nothing but the classic upwinding numerical trace. The time integration is performed using a third order Strong Stability Preserving Runge–Kutta scheme [13,18].

The discontinuous Galerkin approximation u_h will produce a $(k+1)$th order accurate approximation for sufficiently smooth initial data u_o. However, as pointed out above, the post-processed solution

$$u^*(x) = \left(K_h^{v,\ell} \star u_h(\cdot, T)\right)(x),$$

where the convolution kernel is of the form

$$K_h^{v,\ell}(x) = \frac{1}{h}\sum_{\gamma=-k}^{k} c_\gamma^{v,\ell}\, \psi^{(\ell)}\left(\frac{x}{h} - \gamma\right),$$

and T is the final time of the numerical approximation, converges with order $2k+1$, again, when the initial data is smooth enough; see [5,6,15,16,12]. For the discontinuous Galerkin approximation, v is taken to be $2k+2$, and $\ell = k+1$.

The kernel $K_h^{v,\ell}$ is such that $K_h^{v,\ell} \star u = u$ for polynomials u of degree $2k$; this is the only condition the coefficients c_γ must verify. Note that the kernel is supported in at most $2k+2$ elements; this renders the evaluation of the convolution computationally efficient. Finally, note that the function $\psi^{(\ell)}$ is the B-spline obtained by convolving the characteristic function of the interval $(-\frac{1}{2}, \frac{1}{2})$ with itself $\ell - 1$ times. As a consequence, the mapping $x \mapsto \left(K_h^{v,\ell} \star u_h(\cdot, T)\right)(x)$ is a $\mathcal{C}^{k-1}(\mathbb{R})$-function. We can thus take

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{v,\ell} \star u_h(\cdot, T)\right)(x),$$

as an approximation to $\frac{\mathrm{d}^\alpha u}{\mathrm{d}x^\alpha}(x,T)$ for $\alpha \leqslant k-1$, or even for $\alpha = k$; see [16]. However, the order of accuracy of this approximation eventually begins to decrease with the order of the derivative; moreover, the oscillations in the error increase. Next we show that this can be avoided by using the approach devised by Thomée [19].

3. Post-processing for derivative information

3.1. Derivative of the post-processed solution

The first approach was presented in [15]. This method calculates directly the derivative of the post-processed solution. That is,

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{v,\ell}\star u_h(\cdot,T)\right)=\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(\frac{1}{h}\sum_{\gamma=-k}^{k}c_\gamma^{v,\ell}\int_{-\infty}^{\infty}\psi^{(\ell)}\Big(\frac{x-y}{h}-\gamma\Big)u_h(y,T)\mathrm{d}y\right). \tag{3.1}$$

The support width of the post-processor is $2k'+1$, where $k'=\lceil\frac{3k+1}{2}\rceil$. This support width does not increase with taking higher derivatives, but the accuracy decreases and oscillations in the error increase. For the first derivative over a uniform mesh, we obtain the same $(2k+1)$th order accuracy that we obtained for the solution. For the second derivative, we obtain $(2k)$th order accuracy. In general we obtain $(2k+2-\alpha)$th-order accuracy for the α derivative.

3.2. Thomée's approach

In [19], Thomée found that the function

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell}\star u_h(\cdot,T)\right), \tag{3.2a}$$

is an approximation to $\frac{\mathrm{d}^\alpha u}{\mathrm{d}x^\alpha}(x,T)$ whose order or convergence is independent of α. Here, the kernel $K_h^{\alpha+v,\ell}$ is of the form

$$K_h^{\alpha+v,\ell}(x)=\frac{1}{h}\sum_{\gamma\in\mathbb{Z}}c_\gamma^{\alpha+v,\ell}\psi^{(\alpha+v)}\Big(\frac{x}{h}-\gamma\Big), \tag{3.2b}$$

and is such that

$$K_h^{\alpha+v,\ell}\star u=u\quad\text{for polynomials of degree } v-1. \tag{3.2c}$$

In other words, to maintain the order of convergence independent of the order of the derivative, it is enough to work with a kernel defined in term of *smoother* B-splines.

He also noted that, since

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\psi^{\alpha+v}=\partial_h^\alpha\psi^v,$$

where $\partial_h v(x):=(v(x+h/2)-v(x-h/2))/h$, we have that

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell}\star u_h(\cdot,T)\right)=\widetilde{K}_h^{\alpha,v,\ell}\star\partial_h^\alpha u_h, \tag{3.3a}$$

$$\widetilde{K}_h^{\alpha,v,\ell}(x):=\frac{1}{h}\sum_{\gamma\in\mathbb{Z}}c_\gamma^{\alpha+v,\ell}\psi^{(\ell)}\Big(\frac{x}{h}-\gamma\Big). \tag{3.3b}$$

This implies that, once we compute the convolution of translations of the B-spline $\psi^{(\ell)}$ with u_h, the above approximation can be readily computed for any α.

3.3. Theoretical considerations

Here, we obtain an estimate of the L^2-error of the approximation

$$e_\alpha(x,T):=\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}u(x,T)-\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}(K_h^{\alpha+v,\ell}\star u_h(\cdot,T))(x),$$

on a domain Ω_0. The result is the following.

Theorem 1. *Let u_h be the approximate solution given by the DG method for the model problem (2.1). Assume that the initial data u_o is very smooth. Then*

$$\|e_\alpha(\cdot,T)\|_{L^2(\Omega_0)}\leqslant Ch^{\min\{v,2k+1,k+\ell+1\}},$$

where C depends upon the smoothness of the solution.

This result states that the order of convergence of $2k+1$ is achieved provided we take $v\geqslant 2k+1$ and $\ell\geqslant k$.

Proof. We follow Thomée [19]. For the sake of clarity, we drop the argument (x,T); so instead of writing $e_\alpha(x,T)$, we simply write e_α. We have

$$e_\alpha=\left(\frac{\mathrm{d}^\alpha u}{\mathrm{d}x^\alpha}-\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell}\star u\right)\right)+\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell}\star(u-u_h)\right)=T_1+T_2,$$

where

$$T_1:=\frac{\mathrm{d}^\alpha u}{\mathrm{d}x^\alpha}-K_h^{\alpha+v,\ell}\star\frac{\mathrm{d}^\alpha u}{\mathrm{d}x^\alpha},$$

$$T_2:=\left(\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}K_h^{\alpha+v,\ell}\right)\star(u-u_h),$$

by a well-known property of the convolution. Hence

$$\|e_\alpha\|_{L^2(\Omega_0)} \leqslant \|T_1\|_{L^2(\Omega_0)} + \|T_2\|_{L^2(\Omega_0)}.$$

Next, we estimate each of the terms on the right-hand side. To do that, we use very different properties of the post-processor. Indeed, the first term is estimated by using the *approximation* properties of the kernel $K_h^{\alpha+v,\ell}$, that is, that it reproduces by convolution polynomials of degree $v-1$. This term will be bounded only in terms of the smoothness of the exact solution u. On the other hand, the second term is estimated by using the *smoothness* of the kernel, that is, this is defined in terms of the B-spline $\psi^{(\alpha+v)}$. This term will be bounded in terms of negative-order norms of the approximation $u-u_h$.

Let us estimate the first term. If the function u is sufficiently smooth, for example, if its derivatives up to the order $\alpha+v$ are in $L^2(\Omega_1)$ where Ω_1 strictly contains Ω_0, then the term T_1 can be shown to be of the order of h^v by property (3.2c):

$$\|T_1\|_{L^2(\Omega_0)} \leqslant Ch^v \left\| \frac{\mathrm{d}^{\alpha+v}u}{\mathrm{d}x^{\alpha+v}} \right\|_{L^2(\Omega_1)}.$$

The estimate of T_2 is more delicate. By (3.3a),

$$T_2 = \widetilde{K}_h^{\alpha,v,\ell} \star \left(\partial_h^\alpha (u-u_h)\right),$$

and, after a few technicalities described in detail in [19], we get

$$\|T_2\|_{L^2(\Omega_0)} \leqslant C \sum_{|\beta|\leqslant \ell} \|\partial_h^{\alpha+\beta}(u-u_h)\|_{H^{-\ell}(\Omega_0)}.$$

As pointed out above, this term can be estimated in terms of negative-order norms of the error $u-u_h$. Note that until now, we have not used any information about the approximation u_h. In the special case under consideration, we have that

$$\|T_2\|_{L^2(\Omega_0)} \leqslant Ch^{\min\{2k+1,k+\ell+1\}},$$

where C depends upon the smoothness of the solution. In [5], the order of convergence proven was only $\min\{2k+1, k+\ell+1/2\}$. However, for our model problem, it is not difficult to prove that 1/2 can be replaced by 1. As a consequence, we have that

$$\|e_\alpha\|_{L^2(\Omega_0)} \leqslant Ch^{\min\{v,2k+1,k+\ell+1\}}.$$

This completes the proof. □

3.4. Implementation details for general kernels

Now that we have shown that we can obtain the same order of accuracy for the derivative of the post-processed solution as we can for the post-processed solution itself, we concentrate on some implementation details for the derivative post-processor.

Here we *assume* that we know the coefficients defining the convolution kernel and obtain a convenient expression for our approximation to the αth-derivative of u. The expression is written in terms of the following representation of the approximate solution at time $t \in [0,T]$:

$$u_h(x,t) = \sum_{i=1}^{N} \sum_{m=0}^{k} u_m^i(t)\phi_m\left(\frac{x}{h} - i\right),$$

where $\{\phi_m\}_{m=0}^k$ is a basis of the space of polynomials of degree k on the interval $[-1/2, 1/2]$.

Proposition 1

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell} \star u_h(\cdot,T)\right) = \sum_{i=1}^{N} \sum_{m=0}^{k} u_m^i(T)\Phi_m^{\alpha,v,\ell}\left(\frac{x}{h} - i\right),$$

where

$$\Phi_m^{\alpha,v,\ell}(\eta) := \sum_{\gamma\in\mathbb{Z}} c_\gamma^{\alpha+v,\ell}\partial_h^\alpha \theta_m^\ell(\eta-\gamma),$$

$$\theta_m^\ell(\zeta) := \int_{-1/2}^{1/2} \psi^{(\ell)}(\zeta - z)\phi_m(z)\mathrm{d}z.$$

Proof. By the definition of the convolution kernel (3.3), we have

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell} \star u_h(\cdot,T)\right) = \widetilde{K}_h^{\alpha,v,\ell} \star \partial_h^\alpha u_h(\cdot,T) = \frac{1}{h}\sum_{\gamma\in\mathbb{Z}} c_\gamma^{\alpha+v,\ell}\psi^{(\ell)}\left(\frac{\cdot}{h} - \gamma\right) \star \partial_h^\alpha u_h(\cdot,T),$$

and inserting the expression for the approximate solution,

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell}\star u_h(\cdot,T)\right)=\sum_{\gamma\in\mathbb{Z}}c_\gamma^{\alpha+v,\ell}\sum_{i=1}^{N}\sum_{m=0}^{k}u_m^i(T)\partial_h^\alpha C_{i,m,\gamma}^\ell(x),$$

where

$$C_{i,m,\gamma}^\ell(x):=\left(\psi^{(\ell)}\left(\frac{\cdot}{h}-\gamma\right)\star\phi_m\left(\frac{\cdot}{h}-i\right)\right)(x)=\int_{\mathbb{R}}\psi^{(\ell)}\left(\frac{x-y}{h}-\gamma\right)\phi_m\left(\frac{y}{h}-i\right)\mathrm{d}y=\int_{-1/2}^{1/2}\psi^{(\ell)}\left(\frac{x}{h}-z-i-\gamma\right)\phi_m(z)\mathrm{d}z,$$

and so

$$C_{i,m,\gamma}^\ell(x)=\theta_m^\ell\left(\frac{x}{h}-i-\gamma\right).$$

This implies that

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+v,\ell}\star u_h(\cdot,T)\right)=\sum_{i=1}^{N}\sum_{m=0}^{k}u_m^i(T)\sum_{\gamma\in\mathbb{Z}}c_\gamma^{\alpha+v,\ell}\partial_h^\alpha\theta_m^\ell\left(\frac{x}{h}-i-\gamma\right),$$

and the result follows. This completes the proof. □

Let us find a more convenient way of evaluating the function $\Phi_m^{\alpha,v,\ell}$. Since we have

$$\Phi_m^{\alpha,v,\ell}(\eta)=\sum_{\gamma\in\mathbb{Z}}c_\gamma^{\alpha+v,\ell}\partial_h^\alpha\theta_m^\ell(\eta-\gamma)=\sum_{\gamma\in\mathbb{Z}}c_\gamma^{\alpha+v,\ell}h^{-\alpha}\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^j\theta_m^\ell\left(\eta-\gamma+\frac{\alpha}{2}-j\right),$$

we get that

$$\Phi_m^{\alpha,v,\ell}(\eta)=\sum_{\mu+\frac{\alpha}{2}\in\mathbb{Z}}d_\mu^{\alpha,v,\ell}\theta_m^\ell(\eta-\mu),\tag{3.4a}$$

$$d_\mu^{\alpha,v,\ell}:=h^{-\alpha}\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^j c_{\mu+\frac{\alpha}{2}-j}^{\alpha+v,\ell}.\tag{3.4b}$$

We see that for odd-order derivatives the function θ_m^ℓ must be evaluated at the points of the form $\eta+\frac{1}{2}+i$ for $i\in\mathbb{Z}$ whereas for even-order derivatives it has to be evaluated at points of the form $\eta+i$ for $i\in\mathbb{Z}$. For odd derivatives, we perform a change of variables in order to avoid having to calculate the approximation at the element boundaries.

Note also that the function θ_m^ℓ is a piecewise polynomial of degree $\ell+m$ whose support is the interval $[-\ell/2-1,\ell/2+1]$. It is possible to evaluate the integral defining this function by using Gauss quadratures exact for polynomials of degree $\ell+m$. Of course, special care has to be taken to use them across the points in which the function changes its polynomial representation. However, the results contained in this paper use an exact integral calculation.

We also remind the reader that it is possible to evaluate the B-spline $\psi^{(\ell)}$ using the recurrence relation

$$\psi^{(\ell+1)}(x)=\frac{1}{\ell}\left[\left(x+\frac{\ell+1}{2}\right)\psi^{(\ell)}\left(x+\frac{1}{2}\right)+\left(\frac{\ell+1}{2}-x\right)\psi^{(\ell)}\left(x-\frac{1}{2}\right)\right]\quad\text{for }\ell\geqslant 1,\tag{3.5a}$$

where

$$\psi^{(1)}=\begin{cases}1, & x\in\left[-\frac{1}{2},\frac{1}{2}\right],\\ 0, & \text{otherwise;}\end{cases}\tag{3.5b}$$

see [17].

3.5. The computation of even kernels

Next, we describe an implementation of Thomée's [19] computation of kernels $K_h^{\alpha,v,\ell}$ in the case $\ell=2k+1$ where the approximation polynomial degree, k, is even; they turn out to be even functions. Although non-even kernels are useful in the context of piecewise-uniform meshes and for post-processing up to the boundary, see [15,16], an easy and systematic way of computing them, like the one we present next for even kernels, remains to be developed.

Proposition 2. *The non-zero coefficients of the kernel $K_h^{\alpha,v,2k+1}$ of the form (3.2b) satisfying (3.2c) are*

$$c_\gamma^{\alpha+v,2k+1}:=(-1)^\gamma\sum_{j=|\gamma|}^{k-1}\binom{2j}{j-\gamma}(-4)^j\xi_j^{\alpha+v,2k+1}\quad\text{for }|\gamma|<k.$$

The values $\left\{\xi_j^{\alpha+v,2k+1}\right\}_{j=0}^{k-1}$ are computed by using the recurrence

$$\xi_j^{n+1,2k+1} := \sum_{m=0}^{j} \xi_{j-m}^{n,2k+1}\xi_m^{1,2k+1} \quad \text{for } j = 0,\dots,k-1,$$

where $\left\{\xi_j^{1,2k+1}\right\}_{j=0}^{k-1}$ *are the non-zero coefficients of the Taylor polynomial of order* $2k-2$ *of the function* $(\arcsin\tau/\tau)$ *around* $\tau = 0$*, that is,*

$$\left(\frac{\arcsin\tau}{\tau}\right) = \sum_{j=1}^{k-1} \xi_j^{1,2k+1}\tau^{2j} + \mathcal{O}(\tau^{2k}).$$

Note that, since $c_{-\gamma}^{\alpha+v,2k+1} = c_{\gamma}^{\alpha+v,2k+1}$, we have that the kernel $K_h^{\alpha+v,2k+1}$ is an even function, as claimed.

Proof. By Lemma 1 in [19], we only have to show that

$$\sum_{j=0}^{k-1} \xi_j^{\alpha+v,2k+1}\left(\sin\frac{\sigma}{2}\right)^{2j} = \sum_{\gamma=-k+1}^{k-1} c_\gamma^{\alpha+v,\ell}\,\mathrm{e}^{-i\sigma\gamma},$$

where $\left\{\xi_j^{\alpha+v,2k+1}\right\}_{j=0}^{k-1}$ are the non-zero coefficients of the Taylor polynomial of order $2k-2$ of the function $(\arcsin\tau/\tau)^{\alpha+v}$ around $\tau = 0$. But

$$\begin{aligned}
\sum_{j=0}^{k-1} \xi_j^{\alpha+v,2k+1}\left(\sin\frac{\sigma}{2}\right)^{2j} &= \sum_{j=0}^{k-1} \xi_j^{\alpha+v,2k+1}\left(\frac{\mathrm{e}^{i\frac{\sigma}{2}} - \mathrm{e}^{-i\frac{\sigma}{2}}}{2i}\right)^{2j} = \sum_{j=0}^{k-1} \xi_j^{\alpha+v,2k+1}(-4)^{-j}\sum_{m=0}^{2j}\binom{2j}{2j-m}(-1)^m\,\mathrm{e}^{-i\sigma(j-m)} \\
&= \sum_{j=0}^{k-1} \xi_j^{\alpha+v,2k+1}(-4)^{-j}\sum_{\gamma=-j}^{j}\binom{2j}{j+\gamma}(-1)^m\,\mathrm{e}^{-i\sigma\gamma} \\
&= \sum_{\gamma=-k+1}^{k-1}\left(\sum_{j=|\gamma|}^{k-1} \xi_j^{\alpha+v,2k+1}(-4)^{-j}\binom{2j}{j+\gamma}(-1)^{j-\gamma}\right)\mathrm{e}^{-i\sigma\gamma} = \sum_{\gamma=-k+1}^{k-1} c_\gamma^{\alpha+v,\ell}\mathrm{e}^{-i\sigma\gamma}.
\end{aligned}$$

It remains to prove the statement about the coefficients $\left\{\xi_j^{n,2p+1}\right\}_{j=0}^{k-1}$. We proceed by induction on n. For $n = 1$ the statement is true by construction. Now assume that the statement is true for n and let us prove it also holds for $n+1$. We have

$$\begin{aligned}
\left(\frac{\arcsin\tau}{\tau}\right)^{n+1} &= \left(\frac{\arcsin\tau}{\tau}\right)^{n}\left(\frac{\arcsin\tau}{\tau}\right) = \left(\sum_{j=0}^{\infty}\chi_j^n\tau^{2j}\right)\left(\sum_{j=0}^{\infty}\chi_j^1\tau^{2j}\right) = \sum_{j=0}^{k-1}\left(\sum_{m=0}^{j}\chi_{j-m}^n\chi_m^1\right)\tau^{2j} + \mathcal{O}(\tau^{2p}) \\
&= \sum_{j=0}^{p-1}\left(\sum_{m=0}^{j}\xi_{j-m}^{n,2k+1}\xi_m^{1,2k+1}\right)\tau^{2j} + \mathcal{O}(\tau^{2k}),
\end{aligned}$$

Table 4.1
The L^2-errors for the first derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the linear convection equation with sine initial conditions. $\partial_x(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x u_h$		$\partial_x(K \star u_h)$		$\widetilde{K} \star \partial_h u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Linear convection equation $u_t + u_x = 0$						
First derivatives						
$\mathbb{P}^1$						
40	3.6955E−02	–	4.8111E−04	–	6.1640E−04	–
60	2.4663E−02	1.00	1.4230E−04	3.00	1.7998E−04	3.04
80	1.8504E−02	1.00	5.9967E−05	3.00	7.5334E−05	3.03
100	1.4805E−02	1.00	3.0681E−05	3.00	3.8383E−05	3.02
$\mathbb{P}^2$						
40	8.7198E−04	2.00	1.5408E−07	5.28	2.4823E−06	4.98
60	3.8764E−04	2.00	1.8998E−08	5.16	3.2752E−07	5.00
80	2.1807E−04	2.00	4.4039E−09	5.13	7.7761E−08	5.00
100	1.3957E−04	2.00	1.4399E−09	5.01	2.5485E−08	5.00
$\mathbb{P}^3$						
40	1.3046E−05	–	2.9092E−10	–	1.8998E−08	–
60	3.8662E−06	3.00	1.2051E−11	7.85	1.1176E−09	6.99
80	1.6305E−06	3.00	1.3729E−12	7.55	1.4946E−10	6.99
100	8.3518E−07	3.00	2.9673E−13	6.86	3.1371E−11	7.00

by the inductive hypothesis, and so

$$\left(\frac{\arcsin\tau}{\tau}\right)^{n+1} = \sum_{j=0}^{k-1} \xi_j^{n+1,2k+1}\tau^{2j} + \mathcal{O}(\tau^{2k}),$$

by the recurrence relation. This completes the proof. □

3.6. Practical implementation

Although the previous sections discuss general implementation, in practice we can choose specific points within an element in which to evaluate the post-processed solution and its derivatives, such as the Gauss points, and perform the integration exactly. Since the convolution kernel is translation invariant for a uniform mesh, we compute the post-processing matrix once and store the information for future use. The computational cost is then negligible.

Table 4.2
The L^2-errors for the second derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the linear convection equation with sine initial conditions. $\partial_x^2(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h^2 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^2 u_h$		$\partial_x^2(K \star u_h)$		$\widetilde{K} * \partial_h^2 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Linear convection equation $u_t + u_x = 0$						
Second derivatives						
$\mathbb{P}^1$						
40	7.0711E−01	–	1.1623E−03	–	4.8828E−04	–
60	7.0711E−01	–	4.9224E−04	2.13	1.4367E−04	3.02
80	7.0711E−01	–	2.7193E−04	2.06	6.0382E−05	3.01
100	7.0711E−01	–	1.7255E−04	2.04	3.0844E−05	3.01
$\mathbb{P}^2$						
40	3.3919E−02	–	3.3093E−07	–	2.0432E−07	–
60	2.2618E−02	1.00	6.1531E−08	4.15	2.3419E−08	5.34
80	1.6965E−02	1.00	1.9124E−08	4.06	5.1904E−09	5.24
100	1.3572E−02	1.00	7.7776E−09	4.03	1.6459E−09	5.15
$\mathbb{P}^3$						
40	8.5100E−04	–	2.9392E−10	–	6.5174E−10	–
60	3.7831E−04	2.00	1.2599E−11	7.77	2.6211E−11	7.93
80	2.1278E−04	2.00	1.5210E−12	7.35	2.7934E−12	7.78
100	1.3621E−04	2.00	3.4276E−13	6.68	5.3526E−13	7.40

Table 4.3
The L^2-errors for the third derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the linear convection equation with sine initial conditions. $\partial_x^3(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h^3 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^3 u_h$		$\partial_x^3(K \star u_h)$		$\widetilde{K} * \partial_h^3 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Linear convection equation $u_t + u_x = 0$						
Third derivatives						
$\mathbb{P}^1$						
40	3.3547E−01	–	3.6955E−02	–	6.9814E−04	–
60	3.3547E−01	–	2.4663E−02	1.00	2.0344E−04	3.04
80	3.3547E−01	–	1.8504E−02	1.00	8.5055E−05	3.03
100	3.3547E−01	–	1.4805E−02	1.00	4.3304E−05	3.03
$\mathbb{P}^2$						
40	7.3474E−02	–	1.0413E−05	–	3.6668E−06	–
60	5.9998E−02	0.50	3.0844E−06	3.00	4.8441E−07	4.99
80	5.1962E−02	0.50	1.3011E−06	3.00	1.1506E−07	5.00
100	4.6477E−02	0.50	6.6617E−07	3.00	3.7719E−08	5.00
$\mathbb{P}^3$						
40	1.1638E−02	–	1.9562E−09	–	2.9704E−08	–
60	7.7595E−03	1.00	2.5500E−10	5.03	1.7491E−09	6.99
80	5.8198E−03	1.00	6.0458E−11	5.00	2.3398E−10	6.99
100	4.6559E−03	1.00	1.9808E−11	5.00	4.9118E−11	7.00

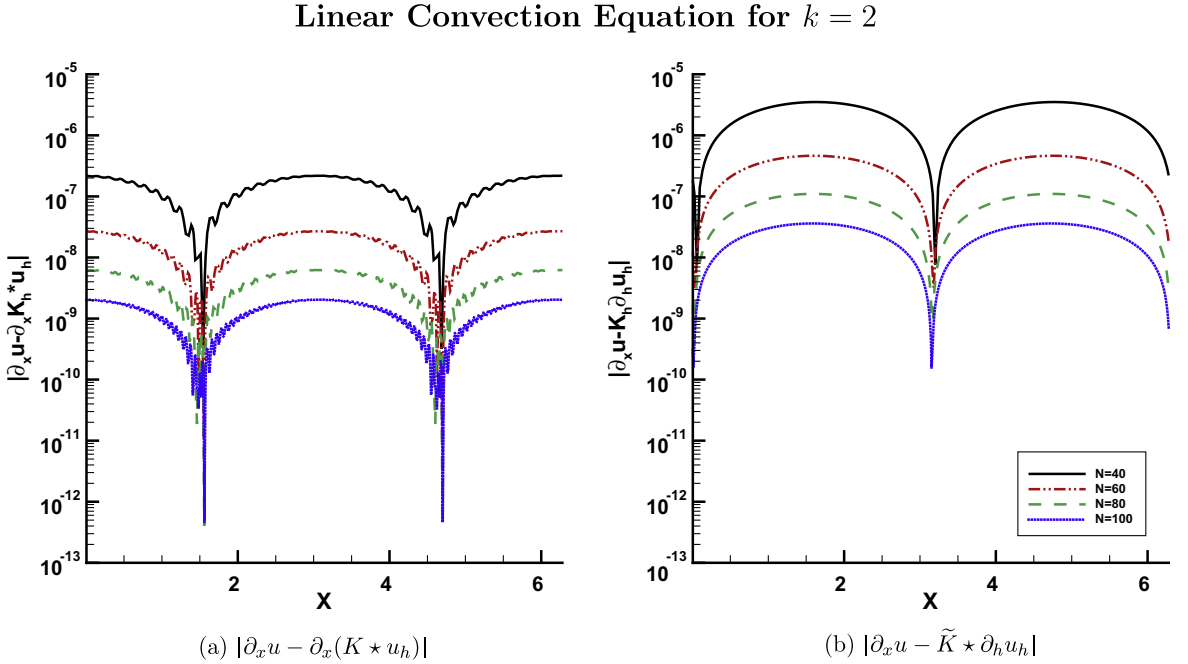
Alternatively, as discussed in [15], the computational implementation of the post-processed solution uses small matrix–vector multiplications with the post-processed solution consists of a sum of these small matrix–vector multiplications. Computing the derivative of the post-processed solution is similar. This can be extended to computation of the derivatives as presented in Proposition 1. That is, to find the derivative of the post-processed solution, we have

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(K_h^{\alpha+\nu,\ell} \star u_h(\cdot,T)\right) = \sum_{i=1}^{N}\sum_{m=0}^{k} u_m^i(T)\frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\left(\sum_{\gamma=-k}^{k} C_{i,m,\gamma}^{\ell}\left(\frac{x}{h}-i-\gamma\right)\right), \tag{3.6}$$

where $\sum_{\gamma=-k}^{k} C_{i,m,\gamma}^{\ell}\left(\frac{x}{h}-i-\gamma\right)$ is the post-processed matrix evaluated at the chosen points within the element. The general formula is given by

Table 4.4
The L^2-errors for the fourth derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the linear convection equation with sine initial conditions. $\partial_x^4(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h^4 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^4 u_h$		$\partial_x^4(K \star u_h)$		$\widetilde{K} * \partial_h^4 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Linear convection equation $u_t + u_x = 0$						
Fourth derivatives						
$\mathbb{P}^1$						
40	–	–	–	–	4.9896E−04	–
60	–	–	–	–	1.4578E−04	3.03
80	–	–	–	–	6.1053E−05	3.03
100	–	–	–	–	3.1119E−05	3.02
$\mathbb{P}^2$						
40	–	–	7.7623E−04	–	2.8322E−07	–
60	–	–	3.4507E−04	2.00	3.0383E−08	5.51
80	–	–	1.9411E−04	2.00	6.4322E−09	5.40
100	–	–	1.2424E−04	2.00	1.9717E−09	5.30
$\mathbb{P}^3$						
40	–	–	8.5394E−08	–	1.235E−09	–
60	–	–	1.6870E−08	4.00	4.8581E−11	7.98
80	–	–	5.3381E−09	4.00	4.8795E−12	7.99
100	–	–	2.1865E−09	4.00	8.4144E−13	7.88

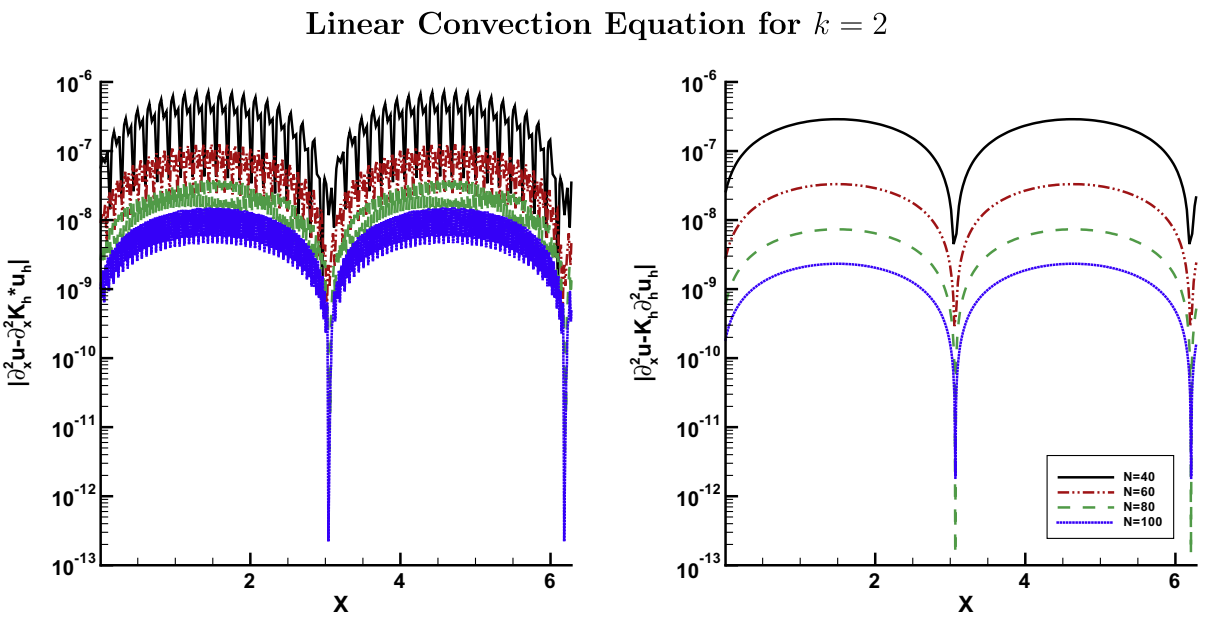


(a) $|\partial_x u - \partial_x(K \star u_h)|$ (b) $|\partial_x u - \widetilde{K} \star \partial_h u_h|$

Fig. 4.1. Pointwise errors for the post-processed first derivatives for $\mathbb{P}^2$-polynomial approximation to the linear convection equation with sine initial conditions.

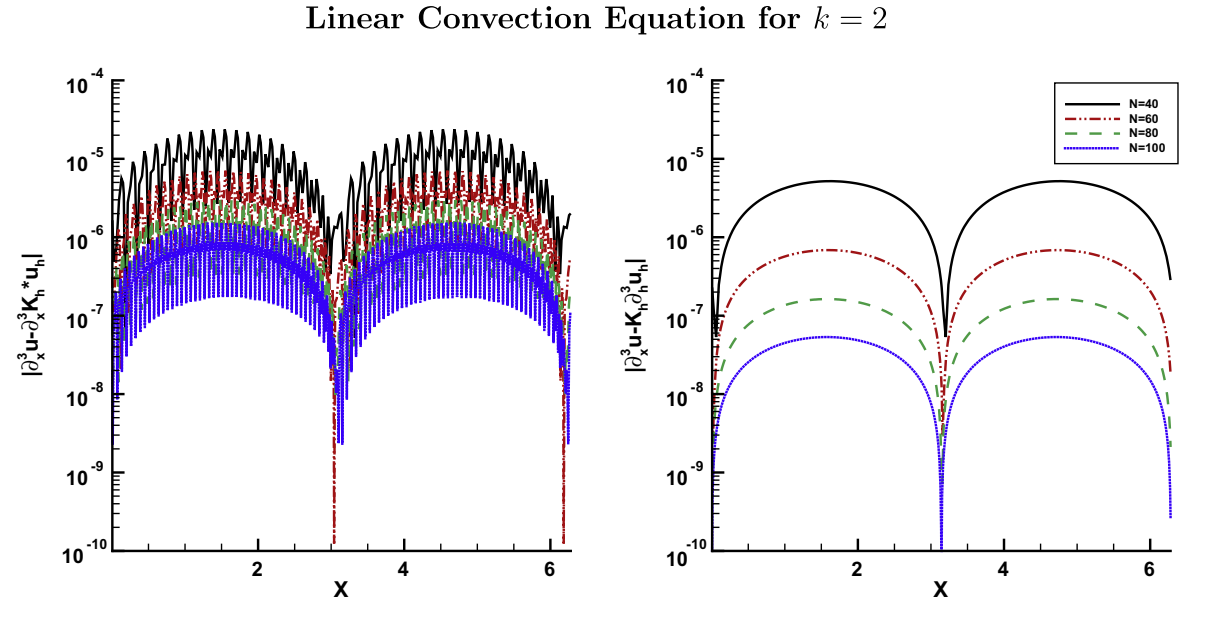
$$C^{\ell}_{i,m,\gamma}(x) = c_{\gamma}\int_{-1/2}^{1/2} \psi^{(\ell)}\left(\frac{x}{h} - z - i - \gamma\right)\phi_m(z)\mathrm{d}z. \tag{3.7}$$

This is a polynomial of degree $2k+1-\alpha$. The support of this method remains the same as for the post-processed solution itself, that is, $2k'+1$ where $k' = \lceil \frac{3k+1}{2} \rceil$.



(a) $|\partial_x^2 u - \partial_x^2(K \star u_h)|$ (b) $|\partial_x^2 u - \widetilde{K} \star \partial_h^2 u_h|$

Fig. 4.2. Pointwise errors for the post-processed second derivatives for $\mathbb{P}^2$-polynomial approximation to the linear convection equation with sine initial conditions.



(a) $|\partial_x^3 u - \partial_x^3(K \star u_h)|$ (b) $|\partial_x^3 u - \widetilde{K} \star \partial_h^3 u_h|$

Fig. 4.3. Pointwise errors for the post-processed third derivatives for $\mathbb{P}^2$-polynomial approximation to the linear convection equation with sine initial conditions.

For Thomée's method, we again can pre-compute the derivative post-processing coefficients, similarly using the same post-processing matrix. Due to the nature of the B-spline recurrence relation, this means that we maintain a post-processed polynomial of degree $2k+1$ for all derivatives. However, the support increases in size with the new coefficients being given in formula (3.4b). For example, for the first and second derivatives, the support is $2k'+3$, the third and fourth derivatives have a support of $2k'+5$. In this manner, the support of the derivative post-processed solution widens with successive derivatives and renders the post-processor as being less local.

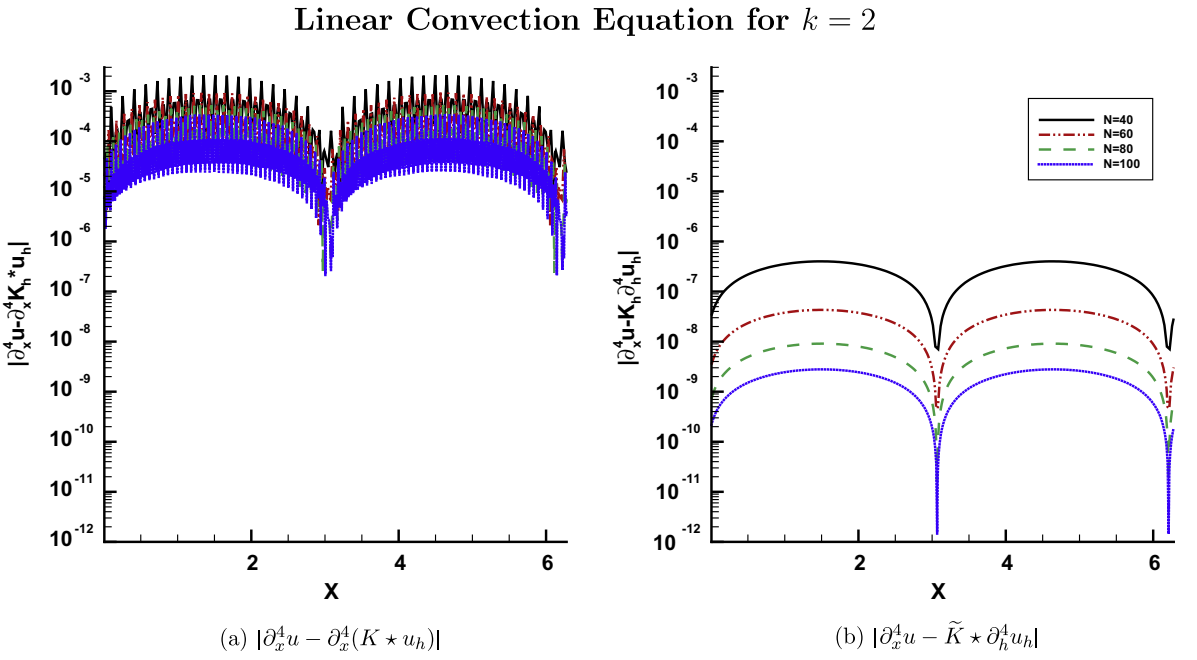


Fig. 4.4. Pointwise errors for the post-processed fourth derivatives for $\mathbb{P}^2$-polynomial approximation to the linear convection equation with sine initial conditions.

Table 4.5
The L^2-errors for the $\mathbb{P}^4$-piecewise polynomial projection of $\sin(x)$, $x \in (0, 2\pi)$, for the first four derivatives. Shown are the errors for the projection onto a piecewise polynomial space as well as the post-processed derivatives. $\partial_x^\alpha (K \star u_h)$ represents the αth-derivative of the post-processed solution, and $\widetilde{K} * \partial_h^\alpha u_h$ represents αth derivative post-processing using higher-order splines.

$\mathbb{P}^4$						
Mesh	$\partial_x^\alpha u_h$		$\partial_x^\alpha (K \star u_h)$		$\widetilde{K} \star \partial_h^\alpha u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Piecewise polynomial projection for $\sin(x), x \in (0, 2\pi)$						
First derivatives						
40	1.1024E−07	–	2.2035E−12	–	1.5131E−10	–
60	2.1779E−08	4.00	3.8429E−14	9.99	3.9630E−12	8.98
80	6.8915E−09	4.00	2.1684E−15	9.99	2.9827E−13	8.99
100	2.8228E−09	4.00	2.3304E−16	10.00	4.0078E−14	9.00
Second derivatives						
40	1.2317E−05	–	2.2035E−12	–	4.9287E−12	–
60	3.6500E−06	3.00	3.8430E−14	9.99	8.6042E−14	9.98
80	1.5400E−06	3.00	2.1685E−15	9.99	4.8566E−15	9.99
100	7.8845E−07	3.00	2.3306E−16	10.00	5.2204E−16	10.00
Third derivatives						
40	8.1074E−04	–	2.2054E−12	–	2.4470E−10	–
60	3.6038E−04	2.00	3.8934E−14	9.96	6.4149E−12	8.98
80	2.0273E−04	2.00	2.3297E−15	9.79	4.8298E−13	8.99
100	1.2975E−04	2.00	2.9424E−16	9.27	6.4906E−14	8.99
Fourth derivatives						
40	3.2055E−02	–	5.3486E−12	–	9.6359E−12	–
60	2.1373E−02	1.00	4.1449E−13	6.31	1.6838E−13	9.98
80	1.6031E−02	1.00	7.3020E−14	6.04	9.5076E−15	9.99
100	1.2825E−02	1.00	1.9101E−14	6.01	1.4455E−15	8.44

3.7. Extensions

In [5], the post-processing of the solutions of finite element methods for symmetric linear hyperbolic problems was described. Those results can be easily extended to approximations of derivatives by proceeding as indicated in [19]. Let us simply point out that the corresponding kernels are a tensor product of the one-dimensional kernels we have described. The results we have so far discussed for the simple model problem (2.1b) can thus be readily extended to the multidimensional case.

In particular, under suitable conditions on the exact solution, if the discontinuous Galerkin method with polynomials of degree k in each component is used on uniform Cartesian grids, the order of convergence of $2k+1$ is obtained independent of the order of the derivative.

4. Numerical examples

In the following examples, we present numerical results demonstrating the effectiveness of our two methods for derivative calculation. We demonstrate that the first method of taking the derivative of the post-processed solution gives order

Table 4.6
The L^2-errors for the first derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the variable coefficient equation with sine initial conditions. $\partial_x(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x u_h$		$\partial_x(K \star u_h)$		$\widetilde{K} * \partial_h u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional variable coefficient equation						
First derivatives						
$\mathbb{P}^1$						
40	3.7026E−02	–	1.7138E−04	–	2.4485E−04	–
60	2.4682E−02	1.00	5.1314E−05	2.97	7.3616E−05	2.96
80	1.8511E−02	1.00	2.1718E−05	3.00	3.1254E−05	2.98
100	1.4809E−02	1.00	1.1136E−05	3.00	1.6059E−05	2.98
$\mathbb{P}^2$						
40	8.7240E−04	–	5.5069E−08	–	2.4411E−06	–
60	3.8775E−04	2.00	6.9067E−08	5.12	3.2245E−06	4.99
80	2.1811E−04	2.00	1.6903E−09	5.03	7.6554E−08	4.99
100	1.3959E−04	2.00	5.8972E−09	4.72	2.5074E−07	5.00
$\mathbb{P}^3$						
40	1.3040E−05	–	2.7510E−10	–	1.8993E−08	–
60	3.8650E−06	3.00	1.0646E−11	8.02	1.1171E−09	6.99
80	1.6308E−06	3.00	1.0519E−12	8.05	1.4928E−10	7.00
100	8.3504E−07	3.00	1.9462E−13	7.56	3.1288E−11	7.00

Table 4.7
The L^2-errors for the second derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the variable coefficient equation with sine initial conditions. $\partial_x^2(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h^2 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^2 u_h$		$\partial_x^2(K \star u_h)$		$\widetilde{K} * \partial_h^2 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional variable coefficient equation						
Second derivatives						
$\mathbb{P}^1$						
40	7.0711E−01	–	1.1045E−03	–	3.1147E−03	–
60	7.0711E−01	–	4.8090E−04	2.05	9.5132E−04	2.93
80	7.0711E−01	–	2.6834E−04	2.03	4.0472E−04	2.97
100	7.0711E−01	–	1.7107E−04	2.02	2.0788E−04	2.99
$\mathbb{P}^2$						
40	3.3923E−02	–	3.2544E−07	–	1.4294E−07	–
60	2.2619E−02	1.00	6.1855E−08	4.10	1.7735E−08	5.15
80	1.6966E−02	1.00	1.9310E−08	4.05	4.2872E−09	4.94
100	1.3573E−02	1.00	7.8612E−09	4.03	1.4798E−09	4.77
$\mathbb{P}^3$						
40	8.5086E−04	–	5.4989E−10	–	6.8195E−10	–
60	3.7826E−04	2.00	1.2133E−11	9.41	2.4965E−11	8.16
80	2.1279E−04	2.00	1.4208E−12	7.46	2.5404E−12	7.94
100	1.3620E−04	2.00	3.8898E−13	5.81	4.9805E−13	7.30

accuracy of $2k+2-\alpha$. Although the support size of the post-processor does not increase with successive derivatives, the amplitude of the oscillations increase. In the method introduced in this paper, we increase the post-processor support size while maintaining the same order of accuracy as the post-processed solution, that is $2k+1$. In our study, we present the results for the first and second derivatives of the given linear hyperbolic equation. For one-dimensional examples, we also present results for the third and fourth derivative. We also provide a discussion of the $\mathbb{P}^4$-polynomial case. Afterwards, we discuss the benefits of using the two methods.

A simple upwind flux for the discontinuous Galerkin solution was used. The choice of flux does not affect the accuracy of the results of the post-processed solution, since the proofs are for linear hyperbolic equations. These results were initially presented in [6]. Additionally, for simplicity, we use periodic boundary conditions in all examples. The plots of the pointwise errors for the DG solution are not presented in order to focus on the comparison of the two derivative methods. However, for non-periodic boundary conditions, we speculate that we would use a one-sided kernel similar to that used in [16]. Improving this one-sided kernel for both the post-processed solution as well as its derivatives is a subject of on going work. We also note that results using $\mathbb{P}^3$- and $\mathbb{P}^4$-polynomials required the use of a multiprecision package [1].

Table 4.8
The L^2-errors for the third derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the variable coefficient equation with sine initial conditions. $\partial_x^3(K\star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K}*\partial_h^3 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^3 u_h$		$\partial_x^3(K\star u_h)$		$\widetilde{K}*\partial_h^3 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional variable coefficient equation						
Third derivatives						
$\mathbb{P}^1$						
40	–	–	3.7043E−02	–	9.4726E−04	–
60	–	–	2.4685E−02	1.00	2.9798E−04	2.85
80	–	–	1.8512E−02	1.00	1.2782E−04	2.94
100	–	–	1.4809E−02	1.00	6.5848E−05	2.97
$\mathbb{P}^2$						
40	–	–	1.0467E−05	–	3.6493E−06	–
60	–	–	3.0913E−06	3.01	4.8281E−07	4.99
80	–	–	1.3028E−06	3.00	1.1479E−07	4.99
100	–	–	6.6672E−07	3.00	3.7663E−08	4.99
$\mathbb{P}^3$						
40	3.3018E−02	–	2.6648E−07	–	5.3993E−08	–
60	2.2017E−02	1.00	2.6427E−08	5.70	8.0352E−10	10.98
80	1.6514E−02	1.00	8.1800E−09	4.08	1.8797E−11	13.05

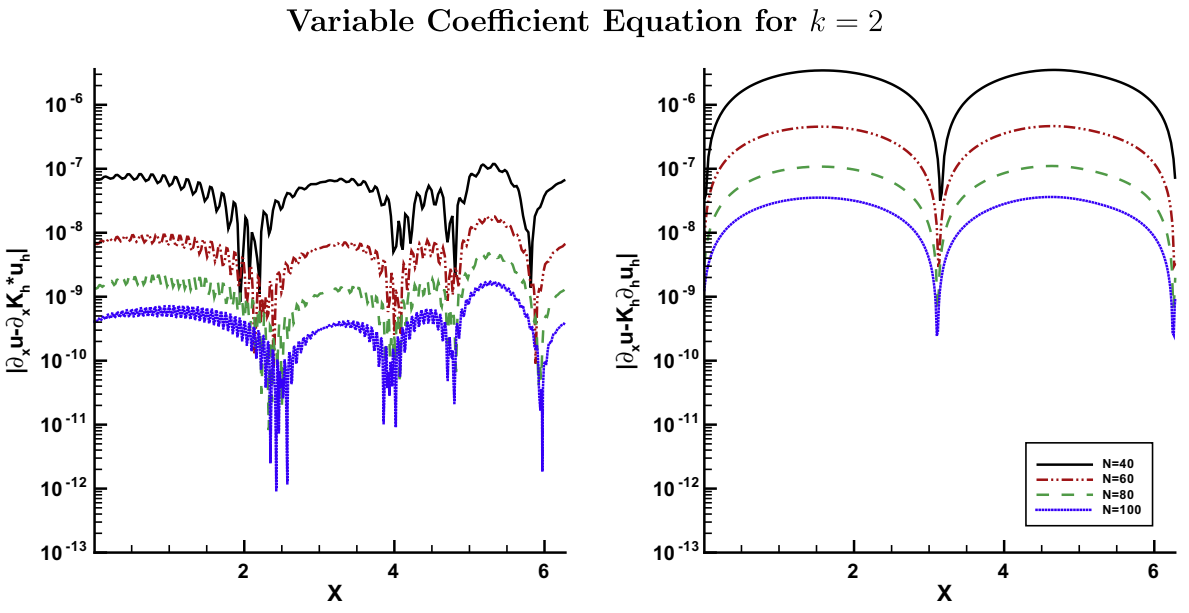
Table 4.9
The L^2-errors for the fourth derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the variable coefficient equation with sine initial conditions. $\partial_x^4(K\star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K}*\partial_h^4 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^4 u_h$		$\partial_x^4(K\star u_h)$		$\widetilde{K}*\partial_h^4 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional variable coefficient equation						
Fourth derivatives						
$\mathbb{P}^1$						
40	–	–	–	–	3.4210E−03	–
60	–	–	–	–	1.1215E−03	2.75
80	–	–	–	–	4.8709E−04	2.90
100	–	–	–	–	2.5210E−04	2.95
$\mathbb{P}^2$						
40	–	–	7.7743E−04	2.00	1.8014E−06	4.76
60	–	–	3.4533E−04	2.00	2.5066E−07	4.86
80	–	–	1.9421E−04	2.00	6.3585E−08	4.77
100	–	–	1.2428E−04	2.00	2.2566E−08	4.64
$\mathbb{P}^3$						
40	–	–	2.6655E−07	–	5.5240E−08	–
60	–	–	2.6427E−08	5.70	8.0352E−10	10.43
80	–	–	8.1800E−09	4.08	1.8797E−11	13.05

4.1. Linear convection equation

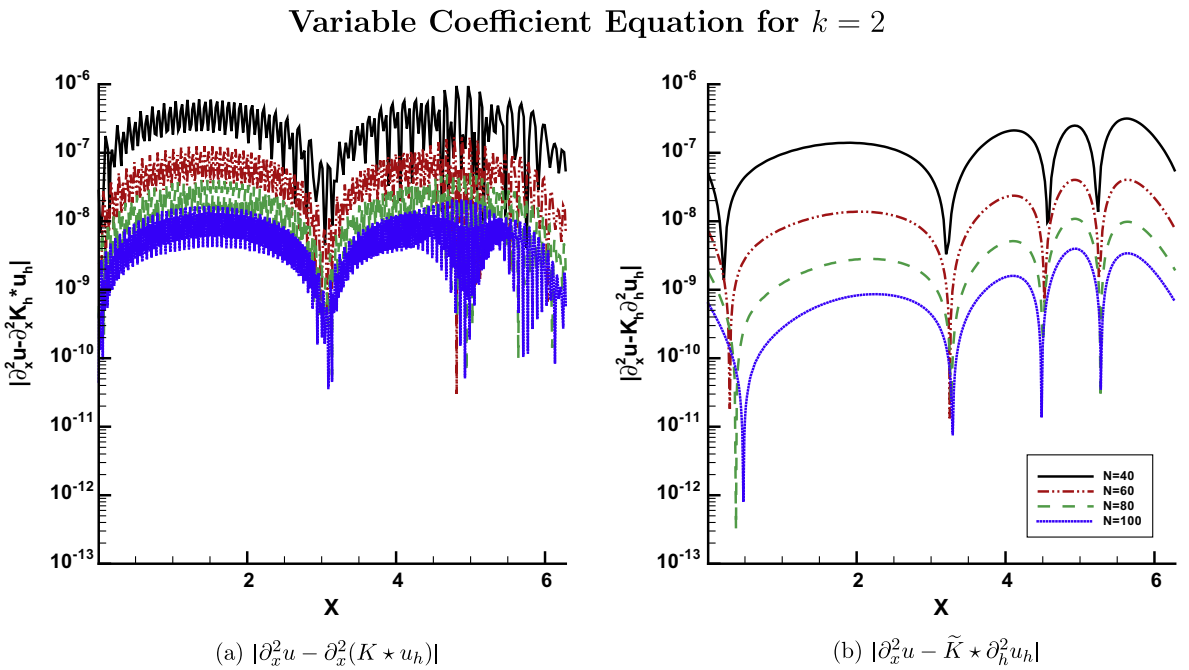
In the first example we investigate the linear convection equation with a periodic initial condition,

$$u_t + u_x = 0, \quad x \in (0, 2\pi) \times (0, T),$$
$$u(x, 0) = \sin(x),$$
$$u(0, t) = u(2\pi, t)$$



(a) $|\partial_x u - \partial_x (K \star u_h)|$ (b) $|\partial_x u - \widetilde{K} \star \partial_h u_h|$

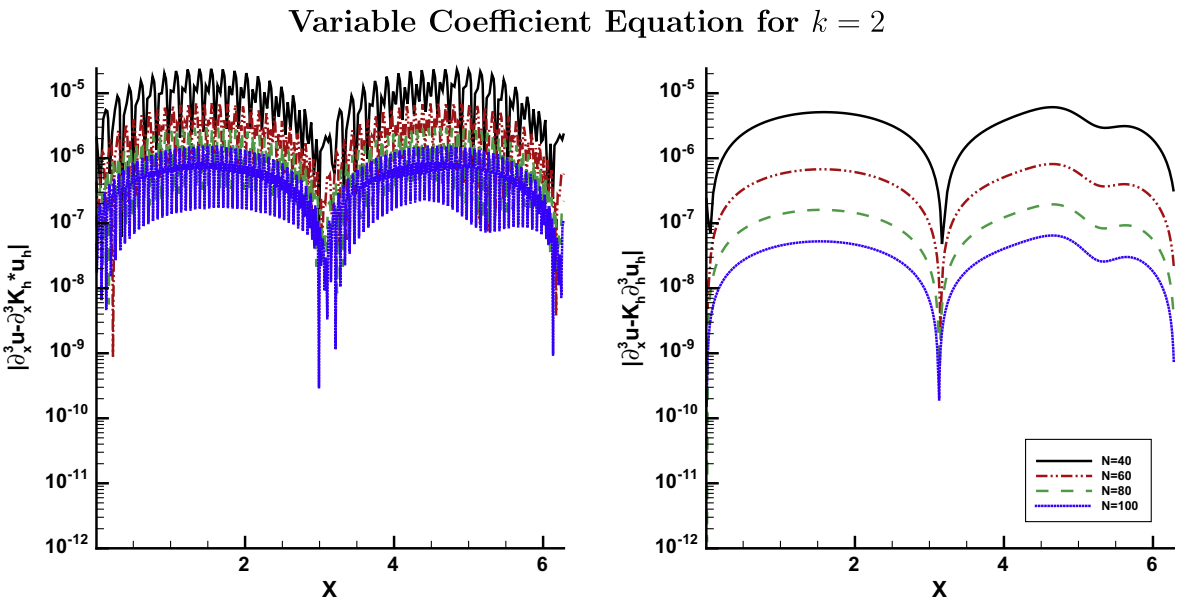
Fig. 4.5. Pointwise errors for the post-processed first derivatives for $\mathbb{P}^2$-polynomial approximation to the variable coefficient equation with sine initial conditions.



(a) $|\partial_x^2 u - \partial_x^2 (K \star u_h)|$ (b) $|\partial_x^2 u - \widetilde{K} \star \partial_h^2 u_h|$

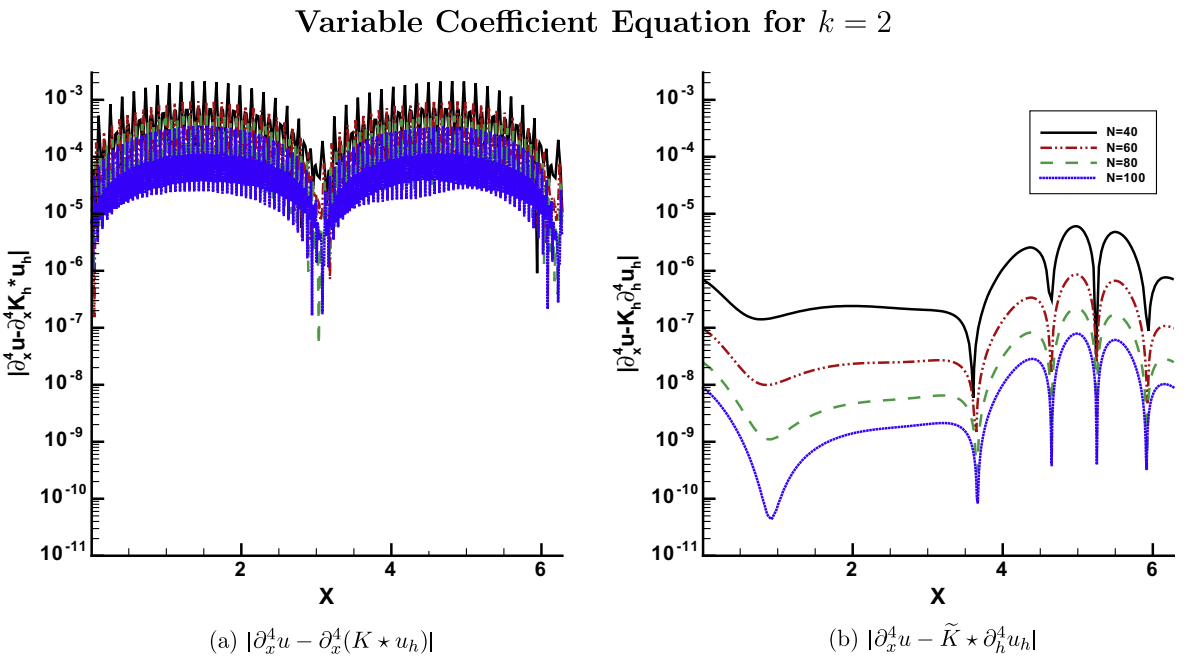
Fig. 4.6. Pointwise errors for the post-processed second derivatives for $\mathbb{P}^2$-polynomial approximation to the variable coefficient equation with sine initial conditions.

calculated at $T = 12.5$. We present the L^2-error results for the first four derivatives in Tables 4.1–4.4 for the discontinuous Galerkin approximation, directly taking the derivative of the post-process solution, and the approach using higher-order B-splines convolved with a finite-difference approximation to the derivative of the discontinuous Galerkin solution. Additionally, to examine the effect of polynomial order we present the results for $\mathbb{P}^1, \mathbb{P}^2$, and $\mathbb{P}^3$-polynomials. Notice, the DG method by itself gives $k+1-\alpha$ order of accuracy for the αth-derivative. But, for both the derivative methods, the order of accuracy improves to $2k+1$ for the first derivative. For the second derivative, the method of Thomée using higher-order B-splines still maintains the $2k+1$ order of accuracy, while the second derivative of the post-processed solution has



(a) $|\partial_x^3 u - \partial_x^3 (K \star u_h)|$ (b) $|\partial_x^3 u - \widetilde{K} \star \partial_h^3 u_h|$

Fig. 4.7. Pointwise errors for the post-processed third derivatives for $\mathbb{P}^2$-polynomial approximation to the variable coefficient equation with sine initial conditions.



(a) $|\partial_x^4 u - \partial_x^4 (K \star u_h)|$ (b) $|\partial_x^4 u - \widetilde{K} \star \partial_h^4 u_h|$

Fig. 4.8. Pointwise errors for the post-processed fourth derivatives for $\mathbb{P}^2$-polynomial approximation to the variable coefficient equation with sine initial conditions.

decreased by one order. This loss in accuracy for the first method also occurs for the third and fourth derivatives, while the method of Thomée continues to maintain the $2k+1$ accuracy. Additionally, there is significant improvement in the magnitude of the errors, with both methods being competitive with each other. In Figs. 4.1–4.4, we can see the corresponding plots of the pointwise errors in log scale for the first four derivatives of the quadratic approximation for the different derivative post-processing techniques. For the first derivative, we can see that directly taking the derivative of the post-processed solution leaves some oscillations, but the magnitude of the errors is improved over filtering using the higher-order B-splines. For the consecutive derivatives, the oscillations remain in the derivative of the post-processed solution, but are filtered out when we use higher-order B-splines. This is because of the smoothness of the kernel used in the first instance, which is $\mathcal{C}^{k-1}$.

4.2. Approximation results using $\mathbb{P}^4$-polynomials

In Table 4.5, we present the results for the projection of

$$u(x) = \sin(x), \quad x \in (0, 2\pi) \tag{4.1}$$

Table 4.10
The L^2-errors for the first derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the wave equation written as a one-dimensional system. $\partial_x(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x u_h$		$\partial_x(K \star u_h)$		$\widetilde{K} * \partial_h u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional system						
First derivatives						
$\mathbb{P}^1$						
40	2.6131E−02	–	3.4020E−04	–	4.2809E−04	–
60	1.7439E−02	1.00	1.0062E−04	3.00	1.2570E−04	3.02
80	1.3084E−02	1.00	4.2403E−05	3.00	5.2768E−05	3.02
100	1.0469E−02	1.00	2.1695E−05	3.00	2.6935E−05	3.01
$\mathbb{P}^2$						
40	6.1658E−04	–	1.0895E−07	–	1.7530E−06	–
60	2.7410E−04	2.00	1.3434E−08	5.16	2.3140E−07	4.99
80	1.5420E−04	2.00	3.1140E−09	5.08	5.4951E−08	5.00
100	9.8690E−05	2.00	1.0182E−09	5.01	1.8012E−08	5.00
$\mathbb{P}^3$						
40	9.2248E−06	–	2.0571E−10	–	1.3433E−08	–
60	2.7338E−06	3.00	8.5214E−12	7.85	7.9027E−10	6.99
80	1.1534E−06	3.00	9.7084E−13	7.55	1.0568E−10	6.99
100	5.9056E−07	3.00	2.0982E−13	6.86	2.2182E−11	7.00

Table 4.11
The L^2-errors for the second derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the wave equation written as a one-dimensional system. $\partial_x^2(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h^2 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^2 u_h$		$\partial_x^2(K \star u_h)$		$\widetilde{K} * \partial_h^2 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional system						
Second derivatives						
$\mathbb{P}^1$						
40	5.0000E−01	–	8.2190E−04	–	3.4527E−04	–
60	5.0000E−01	–	3.4807E−04	2.12	1.0159E−04	3.02
80	5.0000E−01	–	1.9229E−04	2.06	4.2696E−05	3.01
100	5.0000E−01	–	1.2201E−04	2.04	2.1810E−05	3.01
$\mathbb{P}^2$						
40	2.3984E−02	–	2.3400E−07	–	1.4447E−07	–
60	1.5993E−02	1.00	4.3509E−08	4.15	1.6560E−08	5.34
80	1.1996E−02	1.00	1.3523E−08	4.06	3.6702E−09	5.24
100	9.5971E−03	1.00	5.4996E−09	4.03	1.1638E−09	5.15
$\mathbb{P}^3$						
40	6.0175E−04	–	2.0783E−10	–	4.6085E−10	–
60	2.6750E−04	2.00	8.9091E−12	7.77	1.8534E−11	7.93
80	1.5048E−04	2.00	1.0755E−12	7.35	1.9753E−12	7.78
100	9.6312E−05	2.00	2.4237E−13	6.68	3.7849E−13	7.40

onto a piecewise $\mathbb{P}^4$-polynomial basis. This is so that we may simulate the results obtained for a discontinuous Galerkin method. The reasons for this are twofold. First, we would like to study the behavior of a piecewise $\mathbb{P}^4$-polynomial basis, but secondly the *cfl* number required to run numerical simulations for such a higher-order method that will allow us to see the accuracy enhancing capabilities is prohibitively expensive. In Table 4.5, we see that the previous results extend to this polynomial basis. That is, taking the derivative of the post-processed solution, reduces the order of accuracy. However, using higher-order B-splines maintains the order of accuracy to $2k+1$ for any order derivative.

4.3. Variable coefficient equation

Next, we consider the case with a variable coefficient,

$$u_t + (a(x,t)u)_x = f(x,t), \quad x \in (0,2\pi)\times(0,T),$$
$$u(x,0) = \sin(x),$$
$$u(0,t) = u(2\pi,t),$$

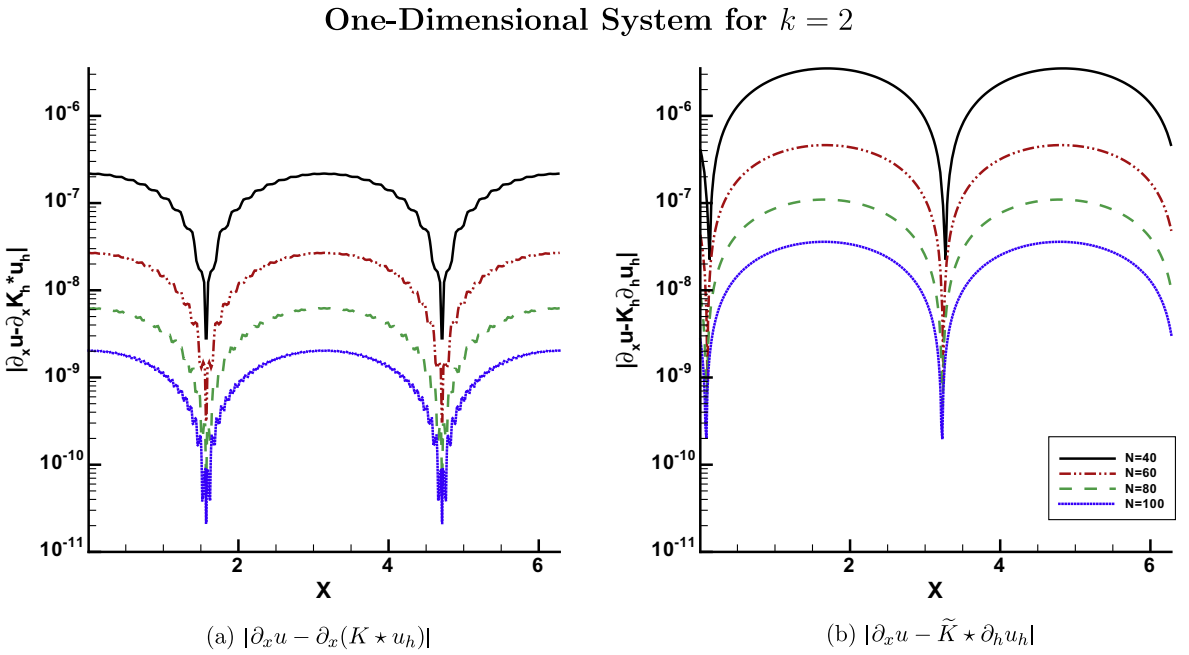
Table 4.12
The L^2-errors for the third derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the wave equation written as a one-dimensional system. $\partial_x^3(K\star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K}*\partial_h^3 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^3 u_h$		$\partial_x^3(K\star u_h)$		$\widetilde{K}*\partial_h^3 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional system						
Third derivatives						
$\mathbb{P}^1$						
40	–	–	2.6131E−02	–	4.8452E−04	3.03
60	–	–	1.7439E−02	1.00	1.4200E−04	3.03
80	–	–	1.3084E−02	1.00	5.9552E−05	3.02
100	–	–	1.0469E−02	1.00	3.0377E−05	3.02
$\mathbb{P}^2$						
40	–	–	7.3632E−06	–	2.5906E−06	–
60	–	–	2.1810E−06	3.00	3.4233E−07	4.99
80	–	–	9.2005E−07	3.00	8.1328E−08	5.00
100	–	–	4.7105E−07	3.00	2.6662E−08	5.00
$\mathbb{P}^3$						
40	2.3348E−02	–	1.3832E−09	–	2.1004E−08	–
60	1.5568E−02	1.00	1.8031E−10	5.03	1.2368E−09	6.99
80	1.1677E−02	1.00	4.2750E−11	5.00	1.6544E−10	6.99
100	9.3420E−03	1.00	1.4006E−11	5.00	3.4731E−11	7.00

Table 4.13
The L^2-errors for the fourth derivative of the discontinuous Galerkin approximation as well as post-processed derivatives for the wave equation written as a one-dimensional system. $\partial_x^4(K\star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K}*\partial_h^4 u_h$ represents derivative post-processing using higher-order splines.

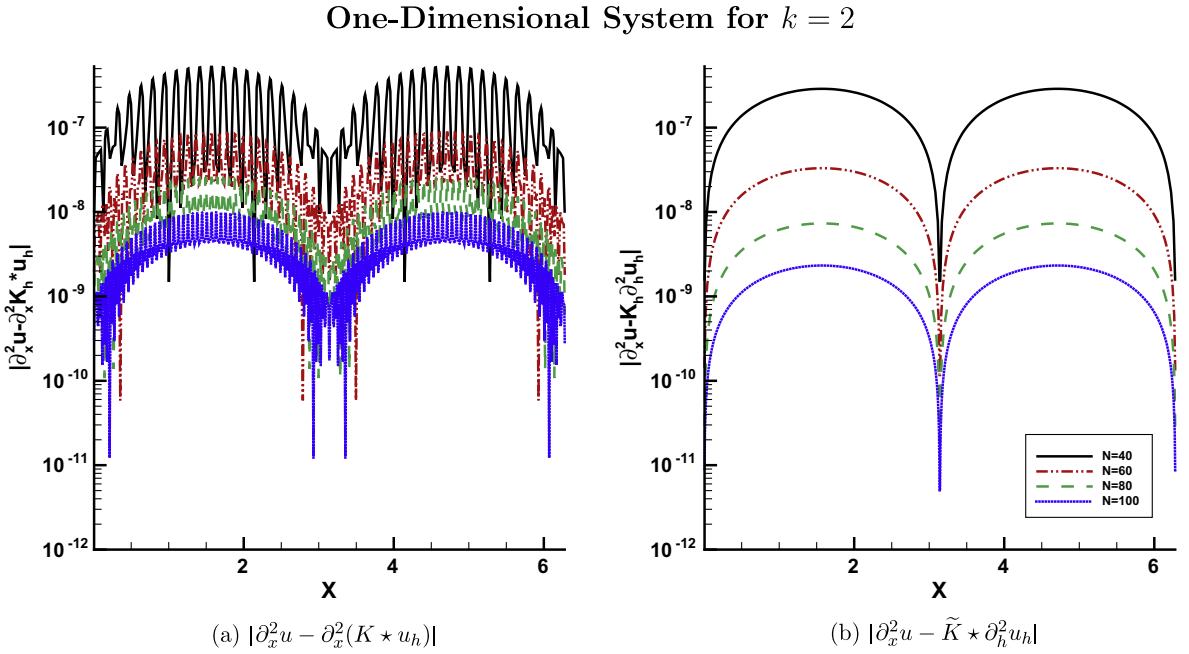
Mesh	$\partial_x^4 u_h$		$\partial_x^4(K\star u_h)$		$\widetilde{K}*\partial_h^4 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
One-dimensional system						
Fourth derivatives						
$\mathbb{P}^1$						
40	–	–	–	–	3.5282E−04	–
60	–	–	–	–	1.0309E−04	3.03
80	–	–	–	–	4.3171E−05	3.03
100	–	–	–	–	2.2004E−05	3.02
$\mathbb{P}^2$						
40	–	–	5.4888E−04	–	2.0026E−07	–
60	–	–	2.4400E−04	2.00	2.1484E−08	5.51
80	–	–	1.3726E−04	2.00	4.5483E−09	5.40
100	–	–	8.7848E−05	2.00	1.3942E−09	5.30
$\mathbb{P}^3$						
40	–	–	9.1297E−08	–	8.8603E−10	–
60	–	–	1.8036E−08	4.00	3.5240E−11	7.95
80	–	–	5.7068E−09	4.00	3.6519E−12	7.88
100	–	–	2.3375E−09	4.00	6.6010E−13	7.66

where $a(x,t) = 2 + \sin(x+t)$ and the forcing function, $f(x,t)$, is taken such that the exact solution is $\sin(x-t)$. We calculate the L^2-errors in the derivative at final time $T = 12.5$ and use periodic boundary conditions. The results that we obtain are much the same as for the linear transport equation and can be seen in Tables 4.6–4.9 and Figs. 4.5–4.8. That is, both methods give $\mathcal{O}(h^{2k+1})$ in the first derivative. Thomée's method of higher-order B-splines is able to maintain this order of accuracy for



(a) $|\partial_x u - \partial_x (K \star u_h)|$ (b) $|\partial_x u - \widetilde{K} \star \partial_h u_h|$

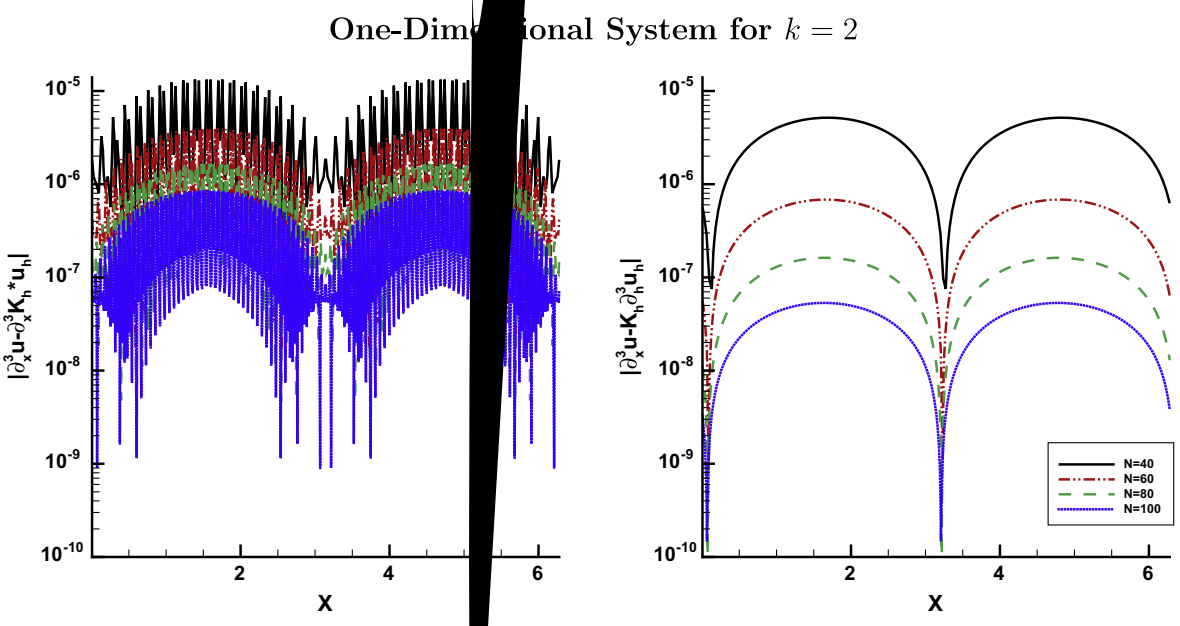
Fig. 4.9. Pointwise errors for the post-processed first derivatives of the u-component for $\mathbb{P}^2$-polynomial approximation to the wave equation written as a one-dimensional system. The errors displayed are similar to those for the v-component.



(a) $|\partial_x^2 u - \partial_x^2 (K \star u_h)|$ (b) $|\partial_x^2 u - \widetilde{K} \star \partial_h^2 u_h|$

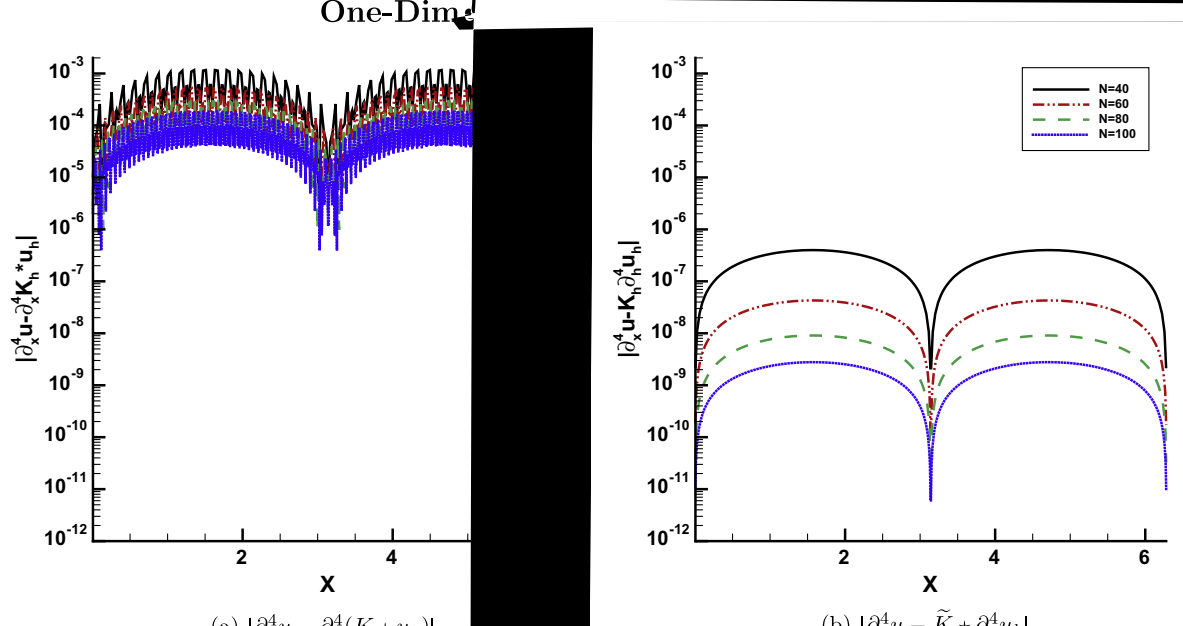
Fig. 4.10. Pointwise errors for the post-processed second derivatives of the u-component for $\mathbb{P}^2$-polynomial approximation to the wave equation written as a one-dimensional system. The errors displayed are similar to those for the v-component.

the second through fourth derivatives while the order of accuracy is decreased by one for directly taking the derivative of the post-processed solution. Again, the magnitude of the errors for both methods are comparable. Examining the quadratic error plots (Figs. 4.5–4.8) shows that the oscillations in the error are more exaggerated for the derivative of the post-processed solution than for using higher-order B-splines.



(a) $|\partial_x^3 u - \partial_x^3 (K \star u_h)|$ (b) $|\partial_x^3 u - \widetilde{K} \star \partial_h^3 u_h|$

Fig. 4.11. Pointwise errors for the post-processed third derivatives of the u-component for $\mathbb{P}^2$-polynomial approximation to the wave equation written as a one-dimensional system. The errors displayed are similar to those for the v-component.



(a) $|\partial_x^4 u - \partial_x^4 (K \star u_h)|$ (b) $|\partial_x^4 u - \widetilde{K} \star \partial_h^4 u_h|$

4.4. One-dimensional system

The last one-dimensional example that we consider is the wave equation written as a system,

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.2}$$

with $u(x,0) = \sin(x)$, $v(x,0) = 0$, and periodic boundary conditions. Again, the derivative errors are calculated at $T = 12.5$ and $x \in (0, 2\pi)$. The error results can be found in Tables 4.10–4.13 and Figs. 4.9–4.12. Results for the L^2-error agree with our previous results obtained for the scalar equations. The plots are shown for the u-component. Similar results are obtained for the v-component.

Table 4.14
The L^2-errors for the first derivative in x of the discontinuous Galerkin approximation as well as post-processed derivatives for the two-dimensional variable coefficient equation with sine initial conditions. $\partial_x(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_h u_h$ represents derivative post-processing with respect to the first partial derivative in x using higher-order splines.

Mesh	$\partial_x u_h$		$\partial_x(K \star u_h)$		$\widetilde{K} * \partial_h u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Two-dimensional variable coefficient equation						
First derivatives in x						
$\mathbb{P}^1$						
40	4.8887E−02	–	2.8224E−04	–	3.6934E−04	–
60	3.2621E−02	1.00	8.7580E−05	2.89	1.1467E−04	2.88
80	2.4474E−02	1.00	3.7717E−05	2.93	4.9289E−05	2.94
100	1.9583E−02	1.00	1.9514E−05	2.95	2.5475E−05	2.96
$\mathbb{P}^2$						
40	2.0024E−03	–	4.8182E−07	–	2.4398E−06	–
60	8.8964E−04	2.00	6.4605E−08	4.96	3.2287E−07	4.99
80	5.0034E−04	2.00	1.5492E−08	4.96	7.6770E−08	4.99
100	3.2019E−04	2.00	5.1204E−09	4.96	2.5183E−08	5.00
$\mathbb{P}^3$						
40	5.6774E−05	–	6.7039E−10	–	1.8988E−08	–
60	1.6833E−05	3.00	3.2102E−11	7.50	1.1163E−09	6.99
80	7.1036E−06	3.00	3.9625E−12	7.27	1.4923E−10	6.99
100	3.6375E−06	3.00	8.2405E−13	7.04	3.1313E−11	7.00

Table 4.15
The L^2-errors for the second derivative in x, ∂_{xx}, of the discontinuous Galerkin approximation as well as post-processed derivatives for the two-dimensional variable coefficient equation with sine initial conditions. $\partial_{xx}(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_{h_{xx}} u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_{xx} u_h$		$\partial_{xx}(K \star u_h)$		$\widetilde{K} * \partial_{h_{xx}} u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Two-dimensional variable coefficient equation						
Second derivative ∂_{xx}						
$\mathbb{P}^1$						
40	7.0711E−01	–	1.2381E−03	–	6.1700E−04	–
60	7.0711E−01	–	5.0685E−04	2.20	1.8005E−04	3.04
80	7.0711E−01	–	2.7615E−04	2.11	7.4309E−05	3.08
100	7.0711E−01	–	1.7417E−04	2.07	3.7588E−05	3.05
$\mathbb{P}^2$						
40	4.6686E−02	–	1.1471E−06	–	1.0818E−06	–
60	3.1126E−02	1.00	1.5734E−07	4.90	1.4200E−07	5.01
80	2.3344E−02	1.00	3.9532E−08	4.80	3.3849E−08	4.98
100	1.8675E−02	1.00	1.3817E−08	4.71	1.1155E−08	4.97
$\mathbb{P}^3$						
40	1.9107E−03	–	1.1233E−09	–	1.3035E−09	–
60	8.4957E−04	2.00	6.2451E−11	7.13	6.6438E−11	7.34
80	4.7797E−04	2.00	8.3344E−12	7.00	8.5007E−12	7.15
100	3.0593E−04	2.00	1.8094E−12	6.85	1.8048E−12	6.95

4.5. Two-dimensional variable coefficient equation

In this example, we consider a two-dimensional variable coefficient equation,

$$u_t + (au)_x + (au)_y = f(x,y,t), \quad [x,y] \in [0,2\pi] \times [0,2\pi], \quad T = 12.5, \tag{4.3}$$

similar to the one-dimensional example. We take the variable coefficient function $a(x,y) = 2 + \sin(x+y)$, and set the forcing function so that our solution is $u(x,y,t) = \sin(x+y-2t)$ and use periodic boundary conditions. In Table 4.14, the results for the first derivative in x is given, the first derivative in y is similar. In the first derivative, we are able to recover $2k+1$ order of accuracy for both methods. It also appears that we are able to recover this $2k+1$ for the second derivatives, which is better than predicted (see Table 4.15 for the second derivative in x, and Table 4.16 for the cross derivative).

Table 4.16
The L^2-errors for the cross derivative, ∂_{xy}, of the discontinuous Galerkin approximation as well as post-processed derivatives for the two-dimensional variable coefficient equation with sine initial conditions. $\partial_{xy}(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_{h_{xy}} u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_{xy} u_h$		$\partial_{xy}(K \star u_h)$		$\widetilde{K} * \partial_{h_{xy}} u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Two-dimensional variable coefficient equation						
Cross derivative ∂_{xy}						
$\mathbb{P}^1$						
40	7.0711E−01	–	6.2079E−04	–	8.5194E−04	–
60	7.0711E−01	–	1.8078E−04	3.04	2.5313E−04	2.99
80	7.0711E−01	–	7.4565E−05	3.08	1.0597E−04	3.03
100	7.0711E−01	–	3.7705E−05	3.06	5.4016E−05	3.02
$\mathbb{P}^2$						
40	4.8942E−02	–	1.0838E−06	–	4.9765E−06	–
60	3.2643E−02	1.00	1.4230E−07	5.01	6.5789E−07	4.99
80	2.4486E−02	1.00	3.3914E−08	4.99	1.5638E−07	4.99
100	1.9590E−02	1.00	1.1174E−08	4.98	5.1290E−08	5.00
$\mathbb{P}^3$						
40	2.0416E−03	–	1.0955E−09	–	3.7968E−08	–
60	9.0805E−04	2.00	6.0582E−11	7.14	2.2337E−09	6.99
80	5.1092E−04	2.00	8.0686E−12	7.01	2.9870E−10	6.99
100	3.2703E−04	2.00	1.7521E−12	6.84	6.2689E−11	7.00

Table 4.17
The first derivative in x for the u-component of the wave equation written as a two-dimensional system.

Mesh	$\partial_x u_h$		$\partial_x(K \star u_h)$		$\widetilde{K} * \partial_h u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Two-dimensional system						
First derivatives in x						
$\mathbb{P}^1$						
40	3.1958E−02	–	2.4550E−03	–	2.4781E−03	–
60	2.1311E−02	1.00	7.3337E−04	2.98	7.3895E−04	2.98
80	1.5996E−02	1.00	3.1028E−04	2.99	3.1236E−04	2.99
100	1.2804E−02	1.00	1.5908E−04	2.99	1.6006E−04	3.00
$\mathbb{P}^2$						
40	1.1895E−03	–	1.6128E−06	–	2.3517E−06	–
60	5.3049E−04	1.99	2.1045E−07	5.02	3.0589E−07	5.03
80	2.9885E−04	1.99	4.9715E−08	5.02	7.2120E−08	5.02
100	1.9142E−04	2.00	1.6251E−08	5.01	2.3541E−08	5.02
$\mathbb{P}^3$						
40	3.8416E−05	–	1.1938E−09	–	1.2561E−08	–
60	1.1279E−05	3.02	6.4091E−11	7.21	7.3711E−10	6.99
80	4.7333E−06	3.02	8.1903E−12	7.15	9.8455E−11	7.00
100	2.4153E−06	3.02	1.6836E−12	7.09	2.0652E−11	7.00

4.6. Two-dimensional system

Lastly, we consider the derivatives of the solution to the second-order wave equation written as a first order system:

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_x + \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.4}$$

with initial conditions

$$u(x,y,0) = \frac{1}{2\sqrt{2}}(\sin(x+y) - \cos(x+y)),$$
$$v(x,y,0) = \frac{1}{2\sqrt{2}}\left((\sqrt{2}-1)\sin(x+y) + (1+\sqrt{2})\cos(x+y)\right),$$

Table 4.18
The L^2-errors for the second derivative in x, ∂_{xx}, of the discontinuous Galerkin approximation as well as post-processed derivatives for the two-dimensional variable coefficient equation with sine initial conditions. $\partial_x^2(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_{h_x}^2 u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_x^2 u_h$		$\partial_x^2(K \star u_h)$		$\widetilde{K} * \partial_{h_x}^2 u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Two-dimensional system						
Second derivatives in x						
$\mathbb{P}^1$						
40	4.6312E−01	–	2.5462E−03	–	2.4600E−03	–
60	4.6312E−01	–	7.9388E−04	2.87	7.3436E−04	2.98
80	4.6312E−01	–	3.5470E−04	2.80	3.1059E−04	2.99
100	4.6312E−01	–	1.9358E−04	2.71	1.5921E−04	2.99
$\mathbb{P}^2$						
40	3.0435E−02	–	1.6150E−06	–	1.6458E−06	–
60	2.0328E−02	1.00	2.1203E−07	5.00	2.1336E−07	5.04
80	1.5259E−02	1.00	5.0586E−08	4.98	5.0233E−08	5.03
100	1.2212E−02	1.00	1.6760E−08	4.95	1.6387E−08	5.02
$\mathbb{P}^3$						
40	1.2904E−03	–	1.1942E−09	–	1.4301E−09	–
60	5.7109E−04	2.01	6.4139E−11	7.21	7.3366E−11	7.33
80	3.2047E−04	2.01	8.2018E−12	7.15	9.1206E−12	7.25
100	2.0479E−04	2.01	1.6874E−12	7.09	1.8398E−12	7.17

Table 4.19
The L^2-errors for the cross derivative, ∂_{xy}, of the discontinuous Galerkin approximation as well as post-processed derivatives for the two-dimensional variable coefficient equation with sine initial conditions. $\partial_{xy}(K \star u_h)$ represents the derivative of the post-processed solution, and $\widetilde{K} * \partial_{h_{xy}} u_h$ represents derivative post-processing using higher-order splines.

Mesh	$\partial_{xy} u_h$		$\partial_{xy}(K \star u_h)$		$\widetilde{K} * \partial_{h_{xy}} u_h$	
	L^2-error	Order	L^2-error	Order	L^2-error	Order
Two-dimensional system						
Cross derivative						
$\mathbb{P}^1$						
40	4.6312E−01	–	2.4551E−03	–	2.5208E−03	–
60	4.6312E−01	–	7.3338E−04	2.98	7.5037E−04	2.99
80	4.6312E−01	–	3.1029E−04	2.99	3.1692E−04	3.00
100	4.6312E−01	–	1.5908E−04	2.99	1.6231E−04	3.00
$\mathbb{P}^2$						
40	2.9822E−02	–	1.6129E−06	–	3.6976E−06	–
60	1.9903E−02	1.00	2.1045E−07	5.02	4.8372E−07	5.02
80	1.4934E−02	1.00	4.9715E−08	5.02	1.1436E−07	5.01
100	1.1950E−02	1.00	1.6251E−08	5.01	3.7384E−08	5.01
$\mathbb{P}^3$						
40	1.3129E−03	–	1.1938E−09	–	2.4962E−08	–
60	5.8379E−04	2.00	6.4091E−11	7.21	1.4673E−09	6.99
80	3.2830E−04	2.00	8.1903E−12	7.15	1.9614E−10	7.00
100	2.1005E−04	2.00	1.6836E−12	7.09	4.1158E−11	7.00

with periodic boundary conditions, $[x, y] \in [0, 2\pi] \times [0, 2\pi]$, and final time $T = 12.5$. The results are presented in Table 4.17 (first derivative in x), Table 4.18 (second derivative in x), and Table 4.19 (cross derivative). For this two-dimensional system we again are able to obtain the $\mathcal{O}(h^{2k+1})$ accuracy using higher-order B-splines and better then predicted accuracy for the method of taking the derivative of the post-processed solution.

4.7. Remarks about derivative post-processing choice

In the previous section, we can see clear distinctions in the numerical results between the two types of derivative post-processing. An obvious result from the numerical examples is that often, especially for the first derivative, the actual L^2-error for the derivative of the post-processed solution is better than that obtained when using higher order B-splines, while the higher-order B-splines consistently maintain their $2k + 1$ convergence rate. For second through fourth derivatives, there is more of a distinction between the results, with the higher order B-spline kernel performing better in terms of convergence rate and L^2-error for $\mathbb{P}^1$- and $\mathbb{P}^2$-polynomials. This is less clear for $\mathbb{P}^3$. Another important aspect to note is that the oscillations in the error are re-introduced with the higher derivatives for the first method. This is because the kernel used is only C^{k-1}. Therefore, when a better L^2-error is desired, it is logical to implement the first method for the first derivative and the method of Thomée for successive derivatives. However, when smoothness of the solution is required, as in filtering for visualization, it is necessary to implement the method using higher order B-splines.

5. Concluding remarks

In this paper, we have introduced two options for post-processing of derivatives for discontinuous Galerkin solutions. Both methods provide higher-order derivative information than the solution itself and can easily be extended to higher dimensions.

In the first method, we obtain the derivative of the post-processed solution itself. In this manner, we can only guarantee $\mathcal{O}(h^{2k+2-\alpha})$ accuracy for the αth-derivative. Additionally, the errors are slightly better than using higher-order splines for the convolution kernel. However, by implementing this derivative post-processing method, we do re-introduce oscillations back into the error.

In the second method, we extend the results of Thomée to discontinuous Galerkin methods. This requires using a higher-order B-spline convolution kernel than the post-processed solution as well as the divided differences of the discontinuous Galerkin solution. This allows us to provably maintain $\mathcal{O}(h^{2k+1})$ accuracy for any derivative. Additionally, oscillations in the derivative errors are filtered out.

Both techniques are effective in improving the errors in the derivatives. The choice of the technique depends on whether higher-order accuracy or smoothness of the solution is required. Further work will be to extend derivative post-processing to non-uniform quadrilateral meshes. Additionally, further investigation of computational efficiency should also be performed.

Acknowledgments

The authors would like to thank the reviewers for their useful suggestions. Portions of this work were performed while the first author was in Department of Mathematics, Virginia Tech, Blacksburg, VA, USA. The second author would like to acknowledge the partial support of the National Science Foundation through Grant DMS-0712955.

References

[1] D.H. Bailey, Y. Hida, K. Jeyabalan, X.S. Li, High Precision Software, Lawrence Berkley National Labratory, 2002. <http://crd.lbl.gov/dhbailey/mpdist/>.
[2] J. Bramble, A. Schatz, Higher order local accuracy by averaging in the finite element method, Math. Comput. 31 (1977) 94–111.
[3] B. Cockburn, S. Hou, C.-W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.
[4] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. III: one dimensional systems, J. Comput. Phys. 84 (1989) 90–113.
[5] B. Cockburn, M. Luskin, C.-W. Shu, E. Süli, Post-processing of Galerkin methods for hyperbolic problems, in: Proceedings of the International Symposium on Discontinuous Galerkin Methods, Springer, 1999, pp. 291–300.
[6] B. Cockburn, M. Luskin, C.-W. Shu, E. Suli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations, Math. Comput. 72 (2003) 577–606.
[7] B. Cockburn, C.-W. Shu, The Runge–Kutta local projection p^1-discontinuous-Galerkin finite element method for scalar conservation laws, Math. Model. Numer. Anal. 25 (1991) 337–361.
[8] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. II: general framework, Math. Comput. 52 (1989) 411–435.
[9] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws. V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[10] B. Cockburn, Discontinuous Galerkin Methods for Convection-Dominated Problems, vol. 9, Springer, 1999.
[11] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.
[12] S. Curtis, R.M. Kirby, J.K. Ryan, C.-W. Shu, Post-processing for the discontinuous Galerkin method over non-uniform meshes, SIAM J. Sci. Comput. 30 (2007) 272–289.
[13] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (1998) 73–85.
[14] T. Moller, R. Machiraju, K. Mueller, R. Yagel, Evaluation and design of filters using a Taylor series expansion, IEEE Trans. Visual. Comput. Graph. 3 (2) (1997) 184–199.

[15] J. Ryan, C.-W. Shu, H. Atkins, Extension of a post-processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem, SIAM J. Sci. Comput. 26 (2005) 821–843.
[16] J. Ryan, C.-W. Shu, On a one-sided post-processing technique for the discontinuous Galerkin methods, Methods Appl. Anal. 10 (2003) 295–307.
[17] L.L. Schumaker, Spline Functions: Basic Theory, John Wiley & Sons, New York, NY, USA, 1981.
[18] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[19] V. Thomée, High order local approximations to derivatives in the finite element method, Math. Comput. 31 (1977) 652–660.