DELFT UNIVERSITY OF TECHNOLOGY

REPORT 10-18

LOCAL DERIVATIVE POST-PROCESSING: CHALLENGES FOR A NON-UNIFORM MESH

JENNIFER K. RYAN

ISSN 1389-6520

Reports of the Delft Institute of Applied Mathematics

Delft 2010

Local Derivative Post-processing: Challenges for a non-uniform mesh

Jennifer K. Ryan[1]

Abstract

Previous investigations into accuracy enhancement for the derivatives of a discontinuous Galerkin solution demonstrated that there are many ways to approach obtaining higher order accuracy in the derivatives, each with different advantageous properties [*J.K. Ryan and B. Cockburn (2009), "Local Derivative Post-Processing for the Discontinuous Galerkin Methods." Journal of Computational Physics, 228:8642-8664.*]. For the discontinuous Galerkin method, the order of accuracy without post-processing for the $d^{th}-$derivative is k+1-d. For the derivative of the post-processed solution it is 2k+1-d. Additionally, it was demonstrated that not only is calculating the derivative of the post-processed solution itself unnecessary, but also that $\mathscr{O}(h^{2k+1})$ can be obtained for the derivative solution for any order derivative, provided the solution is $\mathscr{C}^{2k+1}$. This is done using higher-order B-splines than used for the post-processed solution itself convolved against a finite difference derivative. This introduces higher levels of smoothness into the derivative post-processed approximation. However, this investigation was limited to a uniform mesh consideration, which is highly restrictive for practical applications. In this report, we discuss the advantages and disadvantages of extending accuracy enhancement of derivatives to non-uniform meshes in one-dimension using the ideas of local L^2-projection, characteristic length as well as direct implementation as done for the post-processed solution itself in [*S. Curtis, R. M. Kirby, J. K. Ryan, C.-W. Shu (2007), "Post-processing for the discontinuous Galerkin method over non-uniform meshes." SIAM Journal on Scientific Computing. 30:272-289.*].

Key Words: accuracy enhancement, post-processing, derivatives, discontinuous Galerkin method, hyperbolic equations

AMS(MOS) subject classification: 65M60

[1]j.k.ryan@tudelft.nl. Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands. This work is sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-09-1-3055. This work was originally presented at the International Conference on Spectral and Higher Order Methods (ICOSAHOM) on June 25, 2009 in Trondheim, Norway.

1 Introduction

High order accurate information for derivatives is important in applications such as chemistry and continuum and fluid mechanics. We accomplish this by post-processing a discontinuous Galerkin (DG) solution to a linear hyperbolic equation. However, previous investigations into improving the accuracy of the derivative through post-processing included restrictive assumptions on the mesh [5]. In this paper, we consider how to extend this knowledge to nonuniform meshes, specifically when the mesh is smoothly varying.

The post-processor itself is merely the discontinuous Galerkin solution convolved with some specially designed kernel. That is,

$$u^\star = K_h^{2(k+1),k+1} \star u_h, \tag{1}$$

where u_h is the discontinuous Galerkin solution at the final time, $K_h^{2(k+1),k+1}$ is the convolution kernel, and $u^\star$ is the post-processed solution. The post-processor works by extracting information that is already contained in the discontinuous Galerkin solution. By plotting the pointwise errors at many points within a given element, it is clear that the errors are highly oscillatory. By convolving the DG solution against a specially chosen kernel we are able to extract extra orders of accuracy and smooth the oscillations in the error. Here, we only summarize the post-processor, more details about the post-processor can be found in [2, 6].

The properties of the kernel itself were initial established by Bramble & Schatz [1] and Mock & Lax [4]. Bramble & Schatz noted the increase in accuracy specifically for finite element solutions. This work was extended to DG solutions by Cockburn, Luskin, Shu, and Süli [2].

The idea is the following: The discontinuous Galerkin approximation itself can be shown, in special cases, to be $\mathscr{O}(h^{k+1})$ in the L^2-norm for sufficiently smooth initial data u_o. However, in a negative order norm, the DG solution is $\mathscr{O}(h^{2k+1})$. The convolution kernel allows us to extract this information from the solution so that we can obtain this $\mathscr{O}(h^{2k+1})$ in the easy to compute L^2-norm. This is because of the specially chosen properties of the kernel [1, 2].

The convolution kernel itself is of the form

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

where $K_h^{2(k+1),k+1} \star u = u$ for polynomials u of degree up to and including $2k$. It is supported in at most $2k+2$ elements for a uniform mesh which makes the evaluation of the convolution computationally efficient. We should note that $\psi^{(k+1)}$ is the B-spline obtained by convolving the characteristic function of the interval $(-\frac{1}{2}, \frac{1}{2})$ with itself k times. Alternatively, one could use the recursion formula from Schumaker [8]:

$$\begin{aligned} \psi^{(1)} &= \chi_{[-1/2,1/2]}, \\ \psi^{(k+1)} &= \frac{1}{k} \left[\left(x + \frac{k+1}{2} \right) \psi^{(k)} \left(x + \frac{1}{2} \right) + \left(\frac{k+1}{2} - x \right) \psi^{(k)} \left(x - \frac{1}{2} \right) \right], \\ & \qquad k \geq 1. \end{aligned} \tag{3}$$

2 Post-Processing for Higher Order Derivative Information

There are essentially two methods for derivative Post-Processing, each with their own advantages and disadvantages. The application of these methods is often problem dependent. The first method is post-processing the discontinuous Galerkin solution and then taking the derivative of the post-processed solution. The second method is to use a higher order B-Splines with each successive derivative. Below we will briefly discuss these two methods. Further details can be found in [5].

2.1 Derivative of Post-Processed solution

In this section we briefly outline post-processing the discontinuous Galerkin solution and then taking the α^{th}-derivative, $\frac{d^\alpha}{dx^\alpha}(K_h^{2(k+1),k+1} \star u_h(\cdot,T))(x)$ as initially presented in [6]. We know that the mapping $x \mapsto (K_h^{2(k+1),k+1} \star u_h(\cdot,T))(x)$ is a $\mathscr{C}^{k-1}(\mathbb{R})$-function. If we calculate the derivative of the post-processing polynomial directly, we obtain an $\mathscr{O}(h^{2k+2-\alpha})$ approximation for $\alpha \leq k$. However, the disadvantages of this method are that the order of accuracy decreases with successive derivatives and that oscillations in the error increase. Additionally, in order to implement this method, it can require calculating new post-processing matrix, if we are performing the post-processing using a small matrix-vector format.

2.2 Derivative Post-Processing Using Higher Order B-Splines

The second method that we discuss was presented in [5]. In this method, we again consider the derivative of the post-processed solution,

$$\frac{d^\alpha}{dx^\alpha}(K_h^{2(k+1)+\alpha,k+1} \star u_h(\cdot,T)), \tag{4}$$

which is an approximation to $\frac{d^\alpha u}{dx^\alpha}(x,T)$. However, due to the choice of our kernel, we now have that the order or convergence is independent of α.

For this particular case, our kernel is similar to that of our post-processed solution. That is, it is of the form

$$K_h^{2(k+1)\alpha,k+1}(x) = \frac{1}{h}\sum_{\gamma\in\mathbb{Z}} d_\gamma^{2(k+1)\alpha,k+1}\psi^{(k+1+\alpha)}\left(\frac{x}{h}-\gamma\right), \tag{5}$$

where we point out that we have different coefficients weighting the B-splines, the B-splines are of a higher order and therefore require a larger support. Using this implementation, we are able to maintain the order of convergence independent of the order of the derivative. This essentially is using *smoother* B-splines and was initially presented by Thomée [9].

Furthermore, we note that by the properties of B-splines, if we take the $\alpha^{th}-$derivative of the $(k+1+\alpha)^{th}$ B-spline, this is the same as the α^{th} divided difference of the $(k+1)^{th}$ B-spline. That is,

$$\frac{d^\alpha}{dx^\alpha}\,\psi^{k+1+\alpha} = \partial_h^\alpha\,\psi^{(k+1)},$$

where $\partial_h v(x) := (v(x+h/2) - v(x-h/2))/h$. The consequence is that instead of actually computing the higher-order B-splines, we can use our pre-computed $(k+1)^{th}$ B-spline,

$$\frac{d^\alpha}{dx^\alpha}(K_h^{2(k+1)+\alpha,k+1} \star u_h(\cdot,T)) = \widetilde{K}_h^{\alpha,2(k+1),k+1} \star \partial_h^\alpha u_h. \tag{6}$$

We emphasis again that using smoother B-splines increases the support width of the post-processor. However, as already mentioned, once we compute the convolution of translations of the B-spline $\psi^{(k+1)}$ with u_h, the derivative approximation can be readily computed for any α.

2.3 A Comparison

Here we present a comparison of the derivative post-processing methods for the variable coefficient equation

$$u_t + (a(x,t)u)_x = 0,\ x \in [0,2\pi],\ T \in \mathbb{R}, u(x,0) = \sin(x), u(0,t) = u(2\pi,t). \tag{7}$$

These results were initially presented in [5]. We note that this equation is not supported by the existing theory. However, we are still able to obtain the increase in accuracy. We only display the $\mathbb{P}^2$-polynomial space and refer the reader to the original results. As we can see from Table 1, we obtain the expected results. That is, for the α^{th}-derivative of the DG solution we obtain $\mathscr{O}(h^{k+1-\alpha})$ accuracy and $\mathscr{O}(h^{2k+2-\alpha})$ for the derivative of the post-processed solution. However, if we implement the method using higher order B-splines, then we can improve the order of accuracy for the post-processed derivative to $\mathscr{O}(h^{2k+1})$ for any order derivative, provided the initial condition is smooth enough.

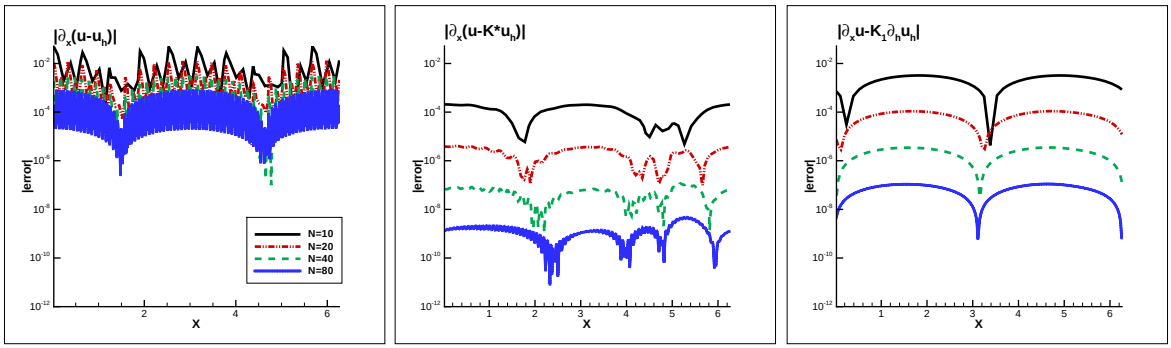

Figure 1: Errors in the first derivatives: DG solution (left), derivative of post-processed solution (center), and using higher-order B-splines (right).

Table 1: L^2-errors for the derivatives of the DG solution (left) as well as the derivatives of the post-processing results (center and right) for the variable coefficient equation with sine initial condition.

$\mathbb{P}^2$						
	$\partial_x^\alpha u_h$		$\partial_x^\alpha (K \star u_h)$		$\widetilde{K} * \partial_h^\alpha u_h$	
mesh	L^2 error	order	L^2 error	order	L^2 error	order
1st Derivatives						
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-09	5.00
2nd Derivatives						
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
3rd Derivatives						
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
4th Derivatives						
40	—	—	7.7743E-04	—	1.8014E-06	—
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

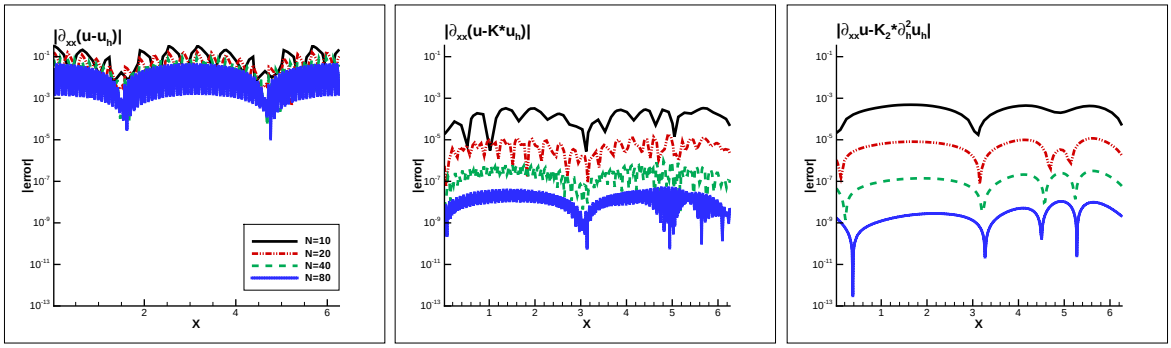

Figure 2: Errors in the second derivatives: DG solution (left), derivative of post-processed solution (center), and using higher-order B-splines (right).

3 Techniques for Post-Processing over a Non-uniform Mesh

As noted above, there are two ways to accomplish higher order accuracy in derivative calculations. The first is by directly calculating the derivative of the post-processed solution.

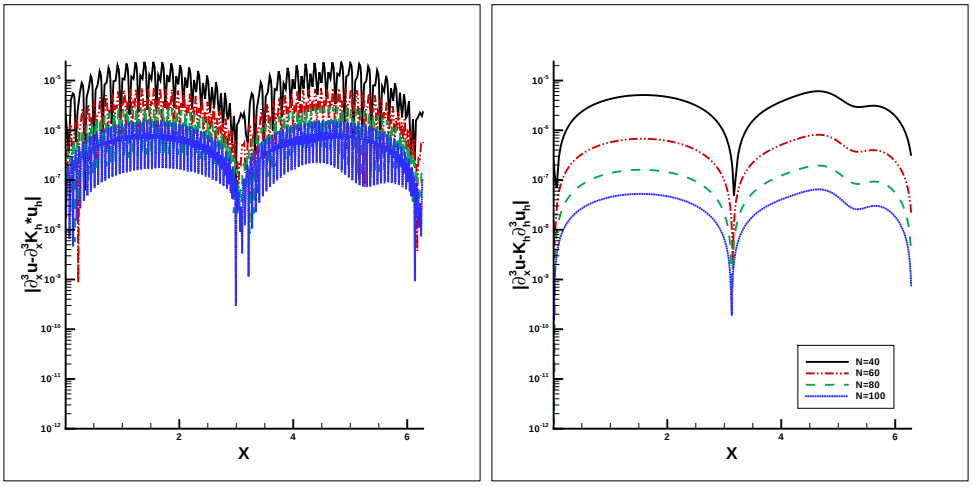

Figure 3: Errors in the third derivatives: derivative of post-processed solution (left), and using higher-order B-splines (right).

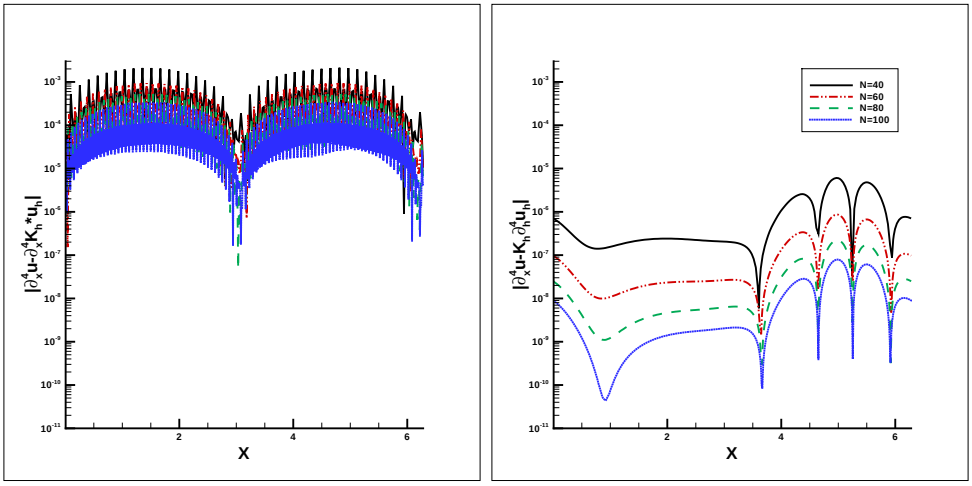

Figure 4: Errors in the fourth derivatives: derivative of post-processed solution (left), and using higher-order B-splines (right).

For this method, we can show that the resulting solution is $\mathcal{O}(h^{2k+2-\alpha})$, but in this case there is no guarantee of increased smoothness. The second method uses a higher order B-spline kernel. For this case, we can provably maintain $\mathcal{O}(h^{2k+1})$ accuracy while increasing smoothness. However, this assumes the ideal case of a linear hyperbolic problem with periodic boundary conditions solved over a uniform mesh. Obviously, we need to extend these results to more challenging problems. Here we address preliminary work on the nonuniform

mesh case. That is, we want to calculate

$$\partial^{\alpha} u^{\star}(x) = \frac{1}{\Delta x_j} \sum_{\gamma=-k}^{k} d_{\gamma}^{2(k+1),k+1} \int_{-\infty}^{\infty} \psi^{(k+1+\alpha)} \left(\frac{y-x}{\Delta x_j} - \gamma \right) \partial_h^{\alpha} u_h(y)\, dy$$

where

$$u_h(y) = \sum_{\ell=0}^{k} u_j^{(\ell)} \phi_j^{(\ell)}(y) \tag{8}$$

is our discontinuous Galerkin solution on element $I_j = (x_{j-1/2}, x_{j+1/2})$.

We first consider the case of a smoothly varying mesh where the mesh is defined by

$$x = \xi + b\sin(\xi) \qquad \text{where} \qquad \xi = \frac{\text{interval length}}{N} i,\ i = 1, \cdots, N.$$

ξ is the uniform mesh variable and $0 \leq b < 1$ is our mesh variation. The difficulty of extending this to the nonuniform mesh case can be more clearly demonstrated if we consider the case where our DG basis consists of monomials:

$$\begin{aligned} \partial_h u^{\star}(x) = & \frac{1}{\Delta x_j \Delta x_{i+j}} \sum_{j} \sum_{\ell=0}^{k} u_{i+j}^{(l)} \sum_{\gamma=-k}^{k} \tilde{d}_{\gamma}^{2(k+1),k+1} \\ & \int_{I_{i+j}} \left[\psi^{(k+1+\alpha)} \left(\frac{y-x}{\Delta x_j} - \frac{1}{2} - \gamma \right) - \psi^{(k+1+\alpha)} \left(\frac{y-x}{\Delta x_j} + \frac{1}{2} - \gamma \right) \right] \left(\frac{y - x_{i+j}}{\Delta x_{i+j}} \right)^{\ell} dy \end{aligned}$$

where element I_{i+j} is in the support of the post-processor and we are post-processing the point $x \in I_j$. We see from this equation that if $j \neq i$, then we have two different mesh sizes: one scaling our B-spline and one scaling our DG basis. This causes a lose the translation invariance of the post-processor so that we can no longer directly implement small matrix-vector multiplications. To overcome this, we have two possibilities, either use a local L^2-projection, or scale our kernel by some characteristic length. These ideas were initially presented in [3] and are briefly summarized below.

3.1 Local L^2-projection

If we choose to implement the local L^2-projection method, then we do not need to modify the post-processing matrix that we obtained for a uniform mesh. We instead are modifying the vector containing the DG coefficients. The algorithm is the following: in order to post-process the derivative on element I_j, we first create a locally uniform mesh of mesh size: $h = \Delta x_j$. We then project the elements from the DG solution, $u_h(x,T)$, onto the locally uniform mesh for all elements in the support of the post-processing region (see Figure 3.1). The projected DG solution over this locally uniform mesh is $u_n(x,T)$. We then use $u_n(x,T)$ to find post-processed derivative on elementI_j. That is, the post-processed derivative is given by

$$\partial u^{\star}(x) = \sum_{i=-2p'}^{2p'} \sum_{l=0}^{k} u_{n_{(i+j)}}^{(l)} D_{i,l,k}(x) \tag{9}$$

where $D_{i,l,k}(x)$ is our post-processing matrix given over a uniform mesh.

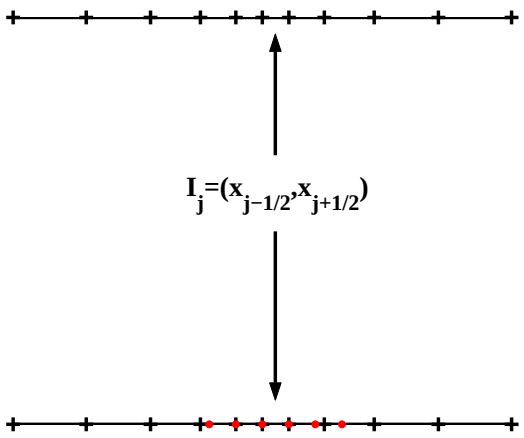

Figure 5: Diagram demonstrating the projection from the nonuniform mesh to the locally uniform mesh. · indicates the created locally uniform mesh for post-processing element I_j onto which we project the approximation.

3.1.1 Characteristic Length

By using a characteristic length for the post-processor, we are modifying the coefficients in the post-processing matrix and these will have to be recalculated for each new mesh type. For this, we denote by L the characteristic length used in the post-processor. In our case we take

$$L = \max_{i=1,\cdots,N} \Delta x_i.$$

Then the modified coefficients used in the post-processing matrix are given by

$$D_L(i,l,k,x) = \frac{1}{L}\int_{I_{i+j}} \left[\psi^{(k+2)}\left(\frac{y-x}{L} - \frac{1}{2} - \gamma\right) - \psi^{(k+2)}\left(\frac{y-x}{L} + \frac{1}{2} - \gamma\right)\right]\left(\frac{y-x_{i+j}}{\Delta x_{i+j}}\right)^l dy.$$

Notice that the B-splines are now scaled by L. Therefore, once we have recalculated the post-processing matrix, we can find the post-processed derivative on I_j via

$$\partial u^\star(x) = \sum_i \sum_{l=0}^{k} u^{(l)}_{(i+j)} D_L(i,l,k,x).$$

4 Combining the Ideas

In this section we present the results for combining the different derivative post-processing techniques with the different post-processing techniques over a uniform mesh.

4.1 Preliminary Results

For our preliminary results, we consider the projection of

$$u(x) = \sin(x)$$

onto a space of piecewise polynomials. We choose this example as the L^2-projection mimics the discontinuous Galerkin solution. Additionally, this is the first term in the errors for calculating the negative-order norm. The results are presented in Table 2 and Figures 6 and 7. In Table 2, we can see that we indeed get $\mathscr{O}(h^{k+1-\alpha}), \quad \alpha = 1,2, \ k = 2$ before post-processing. This improves to $\mathscr{O}(h^{2k+2-\alpha})$ for the derivative of the post-processed solution and $\mathscr{O}(h^{2k+1})$ for the derivative post-processor using higher order B-splines. Additionally, the magnitude of the errors significantly increase for both the derivative post-processors. In Figures 6 and 7, a comparison of the pointwise errors in log-scale using six points per an element are plotted. In the case of the first derivative, we can see the highly oscillatory nature of the DG errors (left), the reintroduction of the oscillations back into the derivative of the post-processed solution (middle), and the smoothness of the errors using higher-order B-splines (right). This is more evident in the plot of the errors for the second derivative (Figure 7).

Table 2: L^2-errors for the first and second derivatives of the discontinuous Galerkin method as well as the derivative of the post-processed solution and the derivative post-processed solution using higher-order B-splines over a smoothly-varying mesh.

$\mathbb{P}^2$						
	DG Errors		Local L^2-projection		Characteristic Length	
	∂u_h		$\tilde{K} * \partial_{h_x} u_h$		$\tilde{K} * \partial_{h_x} u_h$	
mesh	L^2 error	order	L^2 error	order	L^2 error	order
First Derivative						
20	4.4448E-03	—	2.7562E-04	—	8.4862E-04	—
40	1.1153E-03	1.99	8.7798E-06	4.97	3.8223E-05	4.47
60	4.9601E-04	2.00	1.1607E-06	4.99	6.1865E-06	4.49
80	2.7907E-04	2.00	2.7596E-07	4.99	1.6973E-06	4.50
100	1.7863E-04	2.00	9.0571E-08	4.99	6.2217E-07	4.50
Second Derivative						
20	7.3365E-02	—	3.1253E-05	—	5.9007E-05	—
40	3.6774E-02	1.00	2.6804E-06	3.54	1.3287E-06	5.47
60	2.4527E-02	1.00	7.0356E-07	3.30	1.4336E-07	5.49
80	1.8398E-02	1.00	2.5186E-07	3.57	2.9496E-08	5.50
100	1.4720E-02	1.00	1.1063E-07	3.67	8.6490E-09	5.50

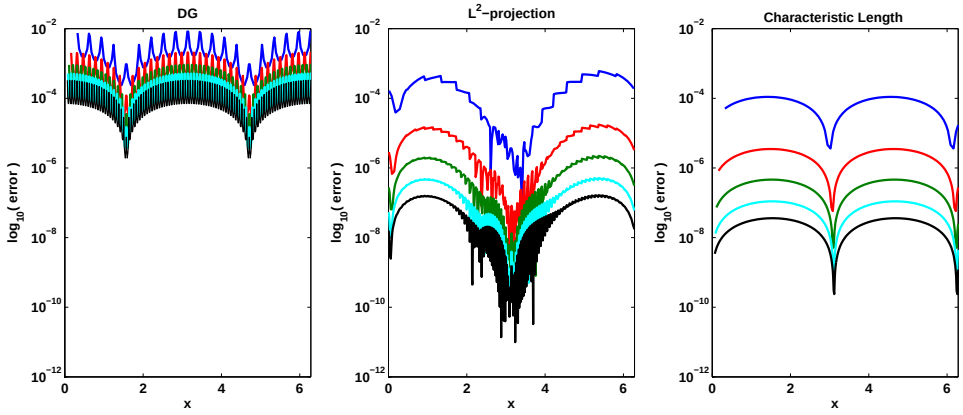

Figure 6: Pointwise errors in the first derivatives over a smoothly-varying mesh using $N = 10, 20, 40, 60, 80,$ and 100 elements: DG solution (left), derivative of post-processed solution (center), and using higher-order B-splines (right).

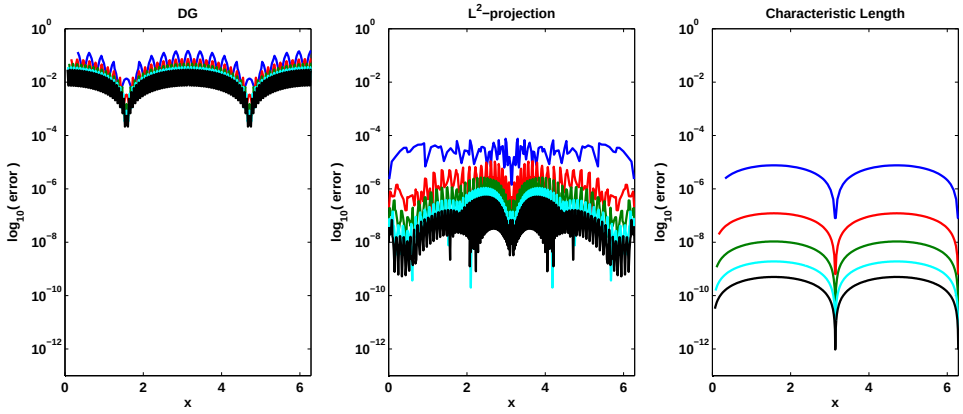

Figure 7: Pointwise errors in the second derivatives over a smoothly-varying mesh using $N = 10, 20, 40, 60, 80,$ and 100 elements: DG solution (left), derivative of post-processed solution (center), and using higher-order B-splines (right).

5 Conclusions

There are two ways to accomplish higher order accuracy in derivative calculations. The first method uses a direct calculation of the derivative of the post-processed solution. This gives us higher order accuracy than for the derivative of the discontinuous Galerkin solution itself, but the accuracy deteriorates with each successive derivative and there is no guarantee of increased smoothness. The second method uses a smoother B-spline kernel and is able to maintain the $2k+1$ order accuracy with each successive derivative while also smoothing the derivative solution. Previous results only include the case of a uniform mesh. In this report, these ideas were paired with using a local L^2-projection as well as characteristic length for scaling the B-splines so that the case of a non-uniform mesh could be addressed. Preliminary

results suggest that we can use this post-processing technique on smoothly-varying meshes using both a local L^2-projection and characteristic length to obtain the increase in accuracy for derivatives of the post-processed solution. However, further studies need to be performed on the mesh assumptions as well as implementation issues associated with the non-uniform mesh.

Acknowledgments: This work is sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-09-1-3055. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon.

References

[1] J. Bramble, A. Schatz (1977), Higher order local accuracy by averaging in the finite element method. Mathematics of Computation, 31:94-111.

[2] B. Cockburn, M. Luskin, C.-W. Shu, E. Süli (2003), Enhanced accuracy by postprocessing for finite element methods for hyperbolic equations. Mathematics of Computation, 72:577-606.

[3] S. Curtis, R. M. Kirby, J. K. Ryan, C.-W. Shu (2007), Post-processing for the discontinuous Galerkin method over non-uniform meshes. SIAM Journal on Scientific Computing. 30:272-289.

[4] M.S. Mock, P.D. Lax (1978), The computation of discontinuous solutions of linear hyperbolic equations. Communications on Pure and Applied Mathematics, 31:423-430.

[5] J.K. Ryan, B. Cockburn (2009), Local Derivative Post-Processing for the Discontinuous Galerkin Methods. Journal of Computational Physics, 228:8642-8664.

[6] J.K. Ryan, C.-W. Shu, H. Atkins (2005), Extension of a post-processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. SIAM Journal of Scientific Computing. 26:821-843.

[7] J.K. Ryan, C.-W. Shu (2003), On a one-sided post-processing technique for the discontinuous Galerkin methods. Methods and Applications of Analysis. 10:295-307.

[8] L. L. Schumaker (1981), Spline Functions: Basic Theory. John Wiley & Sons. New York, NY, USA.

[9] V. Thomée (1977), High order local approximations to derivatives in the finite element method. Mathematics of Computation, 31:652-660.