

**SUPERCONVERGENT ERROR ESTIMATES
FOR POSITION-DEPENDENT SMOOTHNESS-INCREASING
ACCURACY-CONSERVING (SIAC) POST-PROCESSING
OF DISCONTINUOUS GALERKIN SOLUTIONS**

LIANGYUE JI, PAULIEN VAN SLINGERLAND, JENNIFER K. RYAN, AND KEES VUIK

ABSTRACT. Superconvergence of discontinuous Galerkin methods is an area of increasing interest due to the ease with which higher order information can be extracted from the approximation. Cockburn, Luskin, Shu, and Süli showed that by applying a B-spline filter to the approximation at the final time, the order of accuracy can be improved from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^{2k+1})$ in the \mathcal{L}^2 -norm for linear hyperbolic equations with periodic boundary conditions (where k is the polynomial degree and h is the mesh element diameter) [*Math. Comp.* (2003)]. The applicability of this filter for linear hyperbolic problems with non-periodic boundary conditions was computationally extended and renamed a position-dependent smoothness-increasing accuracy-conserving (SIAC) filter by van Slingerland, Ryan, Vuik [*SISC* (2011)]. However, error estimates in the \mathcal{L}^2 -norm for this new position-dependent SIAC filter were never given. Furthermore, error estimates in the \mathcal{L}^∞ -norm have not been established for the original kernel nor the position-dependent kernel. In this paper, for the first time we establish that it is possible to obtain $\mathcal{O}(h^{\min\{2k+1, 2k+2-\frac{d}{2}\}})$ accuracy in the \mathcal{L}^∞ -norm for the position-dependent SIAC filter, where d is the dimension. Furthermore, we extend the error estimates given by Cockburn et al. so that they are applicable to the entire domain when implementing the position-dependent SIAC filter. We also computationally demonstrate the applicability of this filter for visualization of streamlines.

1. INTRODUCTION

Superconvergence of discontinuous Galerkin methods is an area of increasing interest [1, 2, 5, 6, 11, 12, 14]. This is due to the ease with which higher order information can be extracted from the approximation. In most cases one can improve on the solution from $k + \frac{1}{2}$ order accuracy to $2k + 1$. One of these superconvergent techniques, the so-called Smoothness-Increasing Accuracy-Conserving (SIAC) post-processor [6, 12], is based on convolving the numerical solution against a B-spline kernel. It has well-established error estimates in the \mathcal{L}^2 - and negative-order norms for linear hyperbolic equations and convection-diffusion equations [6, 8]. However,

Received by the editor May 25, 2011 and, in revised form, February 1, 2012 and November 8, 2012.

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

Key words and phrases. High-order methods, discontinuous Galerkin, SIAC filtering, accuracy enhancement, B-splines, post-processing, superconvergence, error estimates.

The second author was supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-09-1-3055.

The third author was supported 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.

these \mathcal{L}^2 -error estimates were established for the original symmetric post-processor. This post-processor was a convolution kernel consisting of symmetric B-splines that used a symmetric amount of information from both sides of the point being post-processed. This limitation required that the SIAC filter be applied only to problems with periodic boundary conditions or a distance of $\frac{3k+1}{2}h$ away from the domain boundary, where h is the uniform element size.

Computationally, these limitations were overcome in [13, 14]. The approach in [13] used the original idea of $2k + 1$ central B-splines, but with a shifted support. There were still problems that arose, however. These problems were a stair stepping effect and the magnitude of the errors at the boundaries were larger than in the interior. These issues were again computationally addressed in [14] with the so-called position-dependent SIAC filter. This position-dependent filter made a more intelligent choice for the shifted support and used a kernel that required more central B-splines, which aided in reducing the magnitude of the errors while still achieving the same $2k + 1$ order accuracy (see Figure 1.1(b),(c)). However, \mathcal{L}^2 -error estimates for the one-sided filters in [13] and [14] were never established. Furthermore, \mathcal{L}^∞ -estimates have never been introduced for any type of SIAC filtering, including the original symmetric kernel in [6].

In this paper we establish, for the first time, \mathcal{L}^∞ -error estimates as well as revised \mathcal{L}^2 -error estimates for the position-dependent SIAC filter. These estimates show that the approximation can be improved from $k + \frac{1}{2}$ order accuracy to $\min\{2k + 1, 2k + 2 - \frac{d}{2}\}$ and $2k + 1$, respectively. These estimates are valid for the discontinuous Galerkin (DG) approximation to multi-dimensional linear hyperbolic equations of the form

$$(1.1a) \quad u_t + \sum_{j=1}^d A_j u_{x_j} + A_0 u = 0, \quad (\mathbf{x}, t) \in \Omega \times (0, T],$$

$$(1.1b) \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}),$$

where A_j are real, constant coefficients, $\mathbf{x} = (x_1, x_2, \dots, x_d)$, and $u(\mathbf{x})$ is sufficiently smooth. The domain is taken as $\Omega = [a_1, b_1] \otimes \dots \otimes [a_d, b_d]$ in \mathbb{R}^d . The DG approximation, u_h , is constructed using polynomial basis functions of degree $k \geq 1$ over a uniform mesh with element diameter h , with first order upwind fluxes and exact time integration. These assumptions are quite strong, and usually not valid in practice. Nevertheless, numerical experiments show that the position-dependent post-processor enhances the accuracy in a similar manner for other linear problems as well [8, 9].

It should be emphasized that the estimates for the \mathcal{L}^∞ -error are quite new, as only \mathcal{L}^2 -estimates for discontinuous Galerkin methods were obtained by Cockburn, Luskin, Shu, and Süli in [6]. Bramble and Schatz presented both \mathcal{L}^2 - and \mathcal{L}^∞ -estimates for finite element methods in [4]. However, these results only considered the symmetric post-processor, which cannot be applied near the boundary. In this paper we consider the position-dependent SIAC post-processor which can be applied over the entire domain. Furthermore, we obtain the higher accuracy in the (stronger) \mathcal{L}^∞ -norm, thus extending the estimates in the maximum norm of Bramble and Schatz that were only applicable for *continuous* approximations.

In [14], computational results validating the higher order accuracy in the \mathcal{L}^∞ -norm were presented for examples containing periodic and Dirichlet boundary conditions as well as problems involving stationary shocks. An illustration of this error

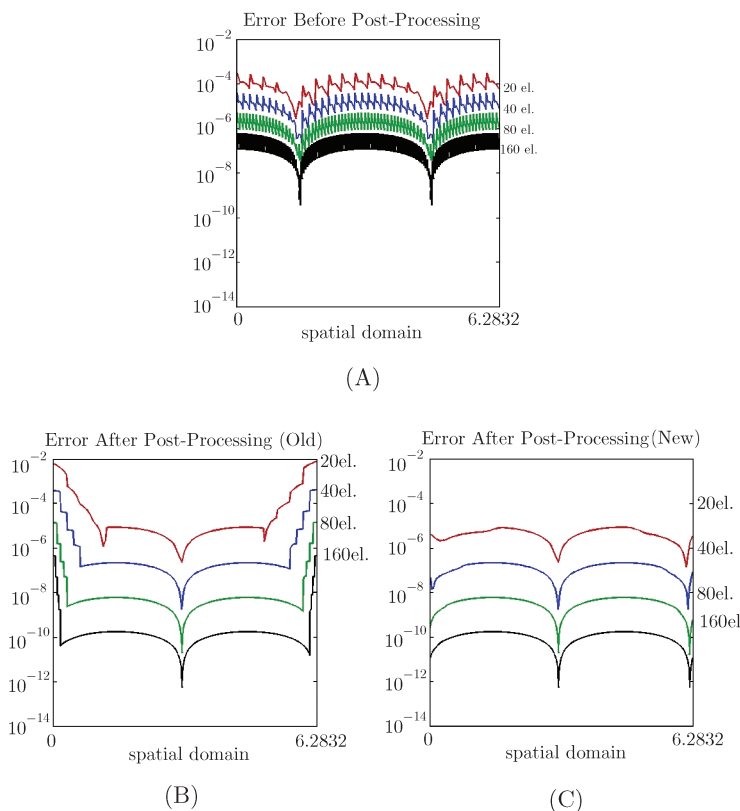


FIGURE 1.1. Comparison of the errors for the SIAC filter for the DG solution to a one-dimensional linear hyperbolic equation with constant coefficients and periodic boundary conditions for $k = 2$. The new filter improves both the smoothness and the accuracy in the entire domain, including the boundary.

improvement for the linear convection equation from [14] is presented in Figure 1.1. In this paper we instead concentrate on demonstrating the applicability of this position-dependent SIAC filter as a filter for visualizing streamlines obtained from discontinuous Galerkin approximations. This further extends the work in [15].

The outline of this paper is as follows. First, we introduce the necessary notation in Section 2. In Section 3, the discontinuous Galerkin method is given along with some important theorems that will be useful in obtaining the required error estimates. Important results from Bramble and Schatz [4] and Cockburn et al. [6] giving prior superconvergent results as well as information regarding the position-dependent SIAC post-processor are discussed in Section 4. The main work of this paper is in Section 5, in which the theoretical error estimates are established. These results are presented in such a way that they can be easily extended for *any* approximation for which superconvergence of the error exists in the negative-order norm. Although this theory was validated in [14], in Section 6 numerical experiments that demonstrate the applicability of the position-dependent SIAC post-processor as a visualization filter are given. Finally, conclusions are given in Section 7.

2. NOTATION & PRELIMINARIES

To develop the necessary theory establishing the \mathcal{L}^∞ - and \mathcal{L}^2 -estimates, we outline the notation that will be used. Throughout this paper, the standard space, derivatives, norms and inner product notation are adopted. In particular, (\cdot, \cdot) and $\langle \cdot, \cdot \rangle$ denote the standard inner product on the computational domain and on the boundary.

The \mathcal{L}^2 -norm on Ω and on the boundary will be given by

$$(2.1) \quad \|\eta\|_{0,\Omega} = \left(\int_{\Omega} \eta^2 d\mathbf{x} \right)^{\frac{1}{2}}, \quad \|\eta\|_{\partial\Omega} = \left(\int_{\partial\Omega} \eta^2 ds \right)^{\frac{1}{2}},$$

the ℓ -norm, maximum ℓ -norm, and semi-norm on Ω are defined as

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

$$(2.3) \quad \|\eta\|_{\infty,\Omega} = \sup_{\mathbf{x} \in \Omega} |\eta(\mathbf{x})|,$$

$$(2.4) \quad |\eta|_{\ell,\Omega} = \sum_{|\alpha| \leq \ell} \|D^\alpha \eta\|_{\infty,\Omega}, \quad \ell > 0.$$

Here, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$ is a multi-index of order $|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_d$, and

$$(2.5) \quad D^\alpha \eta := (D_{x_1}^{\alpha_1} (D_{x_2}^{\alpha_2} \dots (D_{x_d}^{\alpha_d} \eta))),$$

denotes multi-dimensional partial derivatives.

Given $\ell > 0$, the negative-order norm on the domain Ω is defined as

$$(2.6) \quad \|\eta\|_{-\ell,\Omega} = \sup_{\Phi \in C_0^\infty(\Omega)} \frac{(\eta, \Phi)_\Omega}{\|\Phi\|_{\ell,\Omega}}.$$

The difference quotients in the j -th direction are given by the formula

$$(2.7) \quad \partial_{h,j} v(\mathbf{x}) = \frac{1}{h} \left(v \left(\mathbf{x} + \frac{1}{2} h \mathbf{e}_j \right) - v \left(\mathbf{x} - \frac{1}{2} h \mathbf{e}_j \right) \right),$$

here \mathbf{e}_j is a multi-index whose j -th component is 1 and all others 0. The α th-order difference quotient is

$$(2.8) \quad \partial_h^\alpha v(\mathbf{x}) = (\partial_{h,1}^{\alpha_1} (\partial_{h,2}^{\alpha_2} \dots (\partial_{h,d}^{\alpha_d} v(\mathbf{x}))))).$$

Let χ be the characteristic function that is equal to one on the interval $(\frac{1}{2}, \frac{1}{2})$ and let δ denote the Dirac distribution concentrated at $x = 0$. A B-spline of order ℓ is recursively defined as

$$(2.9) \quad \psi^{(0)} = \delta, \quad \psi^{(\ell)} = \psi^{(\ell-1)} \star \chi, \quad \forall \ell \geq 1,$$

where \star denotes the convolution operator given by

$$(2.10) \quad f \star g(\mathbf{x}) = \int_{\mathbb{R}^d} f(\mathbf{x} - \mathbf{z})g(\mathbf{z}) d\mathbf{z}.$$

Multiple dimensional B-splines of order ℓ are defined to be a tensor product of one-dimensional B-splines:

$$(2.11) \quad \psi^{(\ell)}(\mathbf{x}) = \psi^{(\ell)}(x_1)\psi^{(\ell)}(x_2) \dots \psi^{(\ell)}(x_d),$$

for $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$. Denote by $\psi_h^{(\ell)} = \frac{1}{h} \psi^{(\ell)}(\frac{\mathbf{x}}{h})$, the B-spline scaled by h .

The following important properties for establishing the \mathcal{L}^∞ - and \mathcal{L}^2 -error estimates using B-spline properties are given by Lemmas 2.1 and 2.2. The notation $\Omega_0 \subset\subset \Omega_1 \subset\subset \Omega$, will be used.

Lemma 2.1 (Bramble and Schatz [4]). *For any multi-index α and $u \in \mathcal{L}^2(\Omega_1)$,*

$$(2.12) \quad \partial_h^\alpha(\psi_h^{(\ell)} \star u)(\mathbf{x}) = \psi_h^{(\ell)} \star \partial_h^\alpha u(\mathbf{x}), \quad x \in \Omega_0.$$

Lemma 2.2 (Bramble and Schatz [4]). *If s is any fixed integer (positive or negative) and α is any multi-index with $\alpha_j \leq l, j = 1, \dots, d$,*

$$(2.13) \quad \|D^\alpha(\psi_h^{(\ell)} \star u)\|_{s, \Omega_0} \leq C \|\partial_h^\alpha u\|_{s, \Omega_1},$$

for all $u \in H^s(\Omega_1)$ and

$$(2.14) \quad \|D^\alpha(\psi_h^{(\ell)} \star u)\|_{\infty, \Omega_0} \leq C \|\partial_h^\alpha u\|_{\infty, \Omega_1},$$

for all $u \in \mathcal{L}^\infty(\Omega_1)$. Here C is a constant which is independent of h and u .

Remark 2.1. Note that the lemmas given in Bramble and Schatz [4] are for the full B-spline kernel, and not the individual B-splines, however, they are easily extended for individual B-splines.

The following two lemmas will also be useful in proving the error estimates:

Lemma 2.3 ([4]). *If $u \in H^{[d/2]+1}(\Omega_1)$, then (after possible modification on a set of measure zero) $u \in \mathcal{L}^\infty(\Omega_0)$ and*

$$(2.15) \quad \|u\|_{\infty, \Omega_0} \leq C \|u\|_{[d/2]+1, \Omega_0},$$

where C depends on Ω_0, Ω_1 and $[d/2]$ is the integral part of $d/2$.

Lemma 2.4 (Bramble and Schatz [4]). *Let s be an arbitrary but fixed non-negative integer. Then, for $u \in H^s(\Omega_1)$, there is a constant C such that*

$$(2.16) \quad \|u\|_{0, \Omega_0} \leq C \sum_{|\alpha| \leq s} \|D^\alpha u\|_{-s, \Omega_1}.$$

3. DISCONTINUOUS GALERKIN METHOD

In this section a brief outline of the discontinuous Galerkin (DG) method will be given. For more details, consult [7].

Let K define an element in the tessellation \mathcal{T}_h of the domain Ω , where $h > 0$. It is assumed that the elements are invariant under translations by integer multiples of the mesh scaling. Denote by Γ the union of the boundary faces of elements $K \in \mathcal{T}_h$, i.e., $\Gamma = \bigcup_{K \in \mathcal{T}_h} \partial K$.

The DG finite element space used in the approximation is given by

$$(3.1) \quad V_h = \left\{ \varphi \in \mathcal{L}^2(\Omega) : \varphi|_K \in \mathcal{P}^k(K), \quad \forall K \in \mathcal{T}_h \right\},$$

where $\mathcal{P}^k(K)$ is the usual polynomial space. V_h is a piecewise polynomial space that allows for discontinuities across element interfaces.

The approximation u_h can now be determined by the following equations:

$$(3.2a) \quad ((u_h)_t, \zeta) + B(u_h, \zeta) = 0, \quad \zeta \in V_h,$$

$$(3.2b) \quad u_h(\mathbf{x}, 0) = P_h u_0(\mathbf{x}),$$

where P_h is the standard orthogonal projection onto V_h and $B(\cdot, \cdot)$ is a bilinear operator

$$(3.2c) \quad B(u_h, \zeta) = - \sum_{j=1}^d (u_h, A_j \zeta_{x_j}) + \sum_{K \in \mathcal{T}_h} \langle \hat{u}_h, \zeta A \cdot \nu \rangle_{\partial K} + (u_h, A_0 \zeta).$$

Here, A denotes vector (A_1, A_2, \dots, A_d) and $\nu = (\nu_1, \nu_2, \dots, \nu_d)$ is the unit outward normal vector of the integral domain. Without loss of generality we assume $A_j \geq 0, j = 1, \dots, d$, then the numerical flux is taken as an upwind flux, $\hat{u}_h = u_h^-$.

The superconvergent accuracy of the discontinuous Galerkin method in the negative-order norm was previously shown in [6] which follows from results for finite element solutions in [4]. The reader is reminded of the main theorem:

Theorem 3.1 ([6]). *Let u be the exact solution of (1.1), subject to periodic boundary conditions, and u_h the DG approximation defined in (3.2). Then the following error estimates are valid:*

$$(3.3) \quad \|\partial_h^\alpha(u - u_h)\|_{0, \Omega_0} \leq Ch^{k+1} \|\partial_h^\alpha u_0\|_{k+1, \Omega_1},$$

for the approximation itself, and for the difference quotients in the negative-order norm

$$(3.4) \quad \|\partial_h^\alpha(u - u_h)\|_{-(k+1), \Omega_0} \leq Ch^{2k+1} \|\partial_h^\alpha u_0\|_{k+1, \mathcal{D}\Omega_1}$$

with any multi-index α , where $\mathcal{D}\Omega_1$ represents the domain boundary. The constant C is independent of h .

In this paper, the following result is useful for superconvergence of the post-processed solution in the \mathcal{L}^∞ -norm.

Lemma 3.2. *Let u be the exact solution of (1.1), subject to periodic boundary conditions, and u_h the DG approximation defined in (3.2). Then we have the following estimate for the \mathcal{L}^∞ -norm of the divided differences of the error:*

$$(3.5) \quad \|\partial_h^\alpha(u - u_h)\|_{\infty, \Omega_0} \leq Ch^{k+1-\frac{d}{2}}.$$

Proof. To show this inequality, begin by adding and subtracting the standard \mathcal{L}^2 projection $P_h u$ of u . Then use the inverse inequality from [3] in the last step:

$$(3.6) \quad \begin{aligned} \|\partial_h^\alpha(u - u_h)\|_{\infty, \Omega_0} &\leq \|\partial_h^\alpha u - P_h \partial_h^\alpha u\|_{\infty, \Omega_0} + \|P_h \partial_h^\alpha u - \partial_h^\alpha u_h\|_{\infty, \Omega_0} \\ &\leq C_1 h^{k+1} + C_2 h^{-\frac{d}{2}} \|P_h \partial_h^\alpha u - \partial_h^\alpha u_h\|_{0, \Omega_0} \leq Ch^{k+1-\frac{d}{2}}. \end{aligned}$$

□

4. SMOOTHNESS-INCREASING ACCURACY-CONSERVING (SIAC) FILTERING

The main purpose of this section is to show how negative-order norms of difference quotients can be exploited to obtain superconvergence. In [6], the authors found that using finite element methods to solve linear hyperbolic equations (1.1) produces errors with large oscillations, which indicates that there are hidden superconvergent points. Instead of considering the approximation, u_h , a convolution kernel $K_h^{r, \ell}$ was introduced in order to consider certain ‘‘averages’’ of u_h , $K_h^{r, \ell} \star u_h$. The original kernel function $K_h^{r, \ell}$ is a linear combination of central B-splines and is symmetrically and compactly supported.

Superconvergence results for the symmetric kernel in the interior of the domain were established by combining the following two theorems:

Theorem 4.1 (Bramble and Schatz [4]). *For $T > 0$, let $u \in H^{2k+1}(\Omega)$ be the exact solution to the linear hyperbolic equation (1.1). Let $\Omega_0 + 2\text{supp}(K_h^{2k+1,k+1}(x)) \subset \subset \Omega_1 \subset \Omega$, where U is any approximation to u , then*

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

where C_1 and C_2 are independent of h .

Theorem 4.2 ([6]). *Let u be the exact solution of equation (1.1), subject to periodic boundary conditions. Let u_h be the DG approximation defined by (3.2), then*

$$(4.1) \quad \|u(T) - K_h^{2k+1,k+1} \star u_h\|_{0,\Omega_0} \leq C \|u_0\|_{2k+2,\Omega_1} h^{2k+1},$$

where C is a positive constant independent of h .

The main idea of the original kernel is to locally use a symmetric amount of information from both sides of the point being post-processed, which restricts the application of the convolution kernel to the interior of the domain. For periodic boundary conditions, interior results can be periodically extended to the entire domain. Further, in these theorems, the definition of Ω_0 relied on h . To extend superconvergence to the *entire* domain for general boundary value problems, a new position-dependent SIAC filter was constructed [14] that can adaptively adjust the kernel support depending on the position of the evaluation point so that Ω_0 no longer relies on h .

The main goal of this paper is to establish the superconvergence on the entire domain by proving a theorem similar to Theorem 4.1, but applicable to the *entire* domain through the use of the modified kernel in [14]. Furthermore, for the first time, a maximum-norm error estimate will be proven. This will be useful as it expresses the post-processed error using negative-order norms. Technically, the use of the symmetric kernel in the interior of the domain is maintained. The difference is that the number of B-splines used near the boundary is increased. If smoothness is required in the post-processed solution, it is necessary to have transition regions where a convex combination of these kernels is used. The post-processed solution is then given by

$$(4.2) \quad u_h^*(\mathbf{x}) = \theta(\mathbf{x})(K_h^{r_1+1,\ell} \star u_h)(\mathbf{x}) + (1 - \theta(\mathbf{x}))(K_h^{r_2+1,\ell} \star u_h)(\mathbf{x}),$$

where $r_2 > r_1$. Similar to the kernel itself, $\theta(\mathbf{x})$ is a tensor product of one-dimensional functions. In [14], the one-dimensional function, $\theta(x)$ is defined as

$$(4.3) \quad \theta(x) = \begin{cases} 0, & x \in [x_L, x_L + \frac{3k+1}{2}h), \\ p(x), & x \in [x_L + \frac{3k+1}{2}h, x_L + \frac{3k+5}{2}h], \\ 1, & x \in (x_L + \frac{3k+5}{2}h, x_R - \frac{3k+5}{2}h), \\ q(x), & x \in [x_R - \frac{3k+5}{2}h, x_R - \frac{3k+1}{2}h], \\ 0, & x \in (x_R - \frac{3k+1}{2}h, x_R], \end{cases}$$

where $p(x)$, $q(x)$ are polynomials of degree $2k + 3$ and x_L , x_R are the left and right boundaries, respectively. Further, continuity was enforced as follows:

$$p\left(x_L + \frac{3k+1}{2}h\right) = q\left(x_R - \frac{3k+1}{2}h\right) = 0,$$

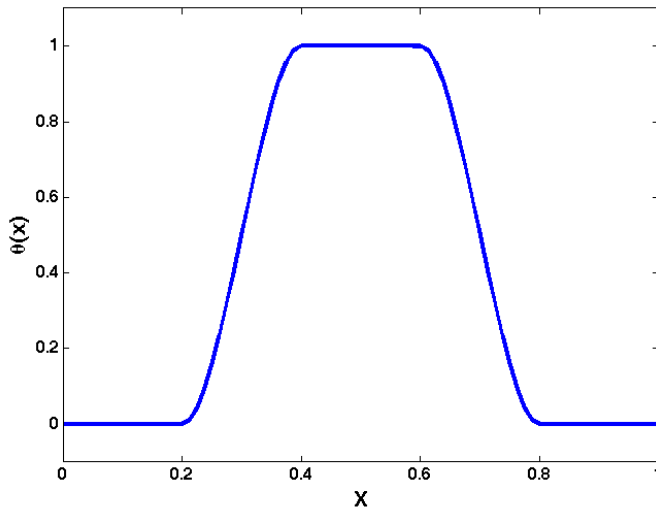


FIGURE 4.1. An example of a coefficient function, $\theta(x)$, that allows a smooth transition between the one-sided filter and the symmetric filter for a linear approximation ($k = 1$) over the unit domain with step-size $h = \frac{1}{10}$.

$$(4.4) \quad \begin{aligned} p\left(x_L + \frac{3k+5}{2}h\right) &= q\left(x_R - \frac{3k+5}{2}h\right) = 1, \\ \frac{d^m}{dx^m}p\left(x_L + \frac{3k+1}{2}h\right) &= \frac{d^m}{dx^m}p\left(x_L + \frac{3k+5}{2}h\right) = 0, \\ \frac{d^m}{dx^m}q\left(x_R - \frac{3k+1}{2}h\right) &= \frac{d^m}{dx^m}q\left(x_R - \frac{3k+5}{2}h\right) = 0, \quad m = 1, \dots, k+1. \end{aligned}$$

We note that these restrictions are actually stronger than necessary. A sufficient requirement is that $\theta(x)$ is a polynomial of degree $2k+1$ with continuity of $k-1$ throughout the domain, with $0 \leq \theta(x) \leq 1$.

An example of such a theta for a linear approximation ($k = 1$) over the domain $[x_L, x_R] = [0, 1]$ is shown in Figure 4.1. In this example,

$$p(x) = 5 - \frac{6}{h}x + \frac{9}{4h^2}x^2 - \frac{1}{4h^3}x^3$$

and

$$q(x) = \frac{1}{4h^3} [-(1-2h)^2(1-5h) + 3(1-4h)(1-2h)x - 3(1-3h)x^2 + x^3]$$

in the definition for $\theta(x)$ given by equation (4.3).

Now, a rigorous general definition for the position-dependent kernel $K_h^{r+1, \ell}$ is given. Then, it will be shown that if u_h^* is used as an approximation to u , super-convergence estimates can be obtained.

4.1. **The position-dependent kernel.** The DG solution is post-processed by convolving u_h against the position-dependent kernel $K_h^{r+1,\ell}$,

$$(4.5) \quad K_h^{r+1,\ell} \star u_h(\mathbf{x}) = \int_{\mathbb{R}^d} K_h^{r+1,\ell}(\mathbf{x} - \mathbf{z})u_h(\mathbf{z})d\mathbf{z}.$$

In this summary, the one-dimensional position-dependent convolution kernel is first defined on a general domain $[a, b]$. The multi-dimensional kernel is then a tensor product of the one-dimensional kernels.

The position-dependent kernel, $K_h^{r+1,\ell}(x)$, is defined as

$$(4.6) \quad K_h^{r+1,\ell}(x) = \frac{1}{h} \sum_{\gamma=0}^r c_\gamma \psi^{(\ell)}\left(\frac{x}{h} - \bar{x}_\gamma\right), \quad \text{for all } x \in R,$$

where $r + 1$ central B-splines of order ℓ are used (see equation (2.9)). Here the kernel coefficients c_0, \dots, c_r are uniquely determined by a small locally linear system

$$(4.7) \quad \sum_{\gamma=0}^r c_\gamma \int_{-\infty}^{\infty} \psi^{(\ell)}(x)(x + \bar{x}_\gamma)^m dx = \begin{cases} 1, & \text{for } m = 0, \\ 0, & \text{for all } m = 1, \dots, r, \end{cases}$$

and the kernel nodes are chosen to be

$$(4.8) \quad \bar{x}_\gamma = -\frac{r}{2} + \gamma + \lambda(\bar{x}), \quad \text{for all } \gamma = 0, \dots, r.$$

$\lambda(\bar{x})$ is a shift function defined to adjust the support of the kernel so that it can be applied to the entire computational domain. The notation \bar{x} is used to signify that it is evaluated at a specific point in the domain, namely the evaluation point used to evaluate the kernel. It has the form

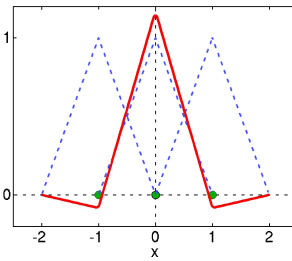
$$(4.9) \quad \lambda(\bar{x}) = \begin{cases} \min\left\{0, -\frac{r+\ell}{2} + \frac{\bar{x}-a-\frac{\epsilon h}{2}}{h}\right\}, & \bar{x} \in [a, \frac{a+b}{2}] \\ \max\left\{0, \frac{r+\ell}{2} + \frac{\bar{x}-b+\frac{\epsilon h}{2}}{h}\right\}, & \bar{x} \in (\frac{a+b}{2}, b] \end{cases}$$

(cf. Figure 4.3). In practice, $\epsilon = 0$. However, to obtain theoretical error estimates, it is necessary to take $\epsilon > 0$ to be arbitrarily small yet fixed. The choices for r and λ affect the shape and support of the kernel (cf. Figure 4.2).

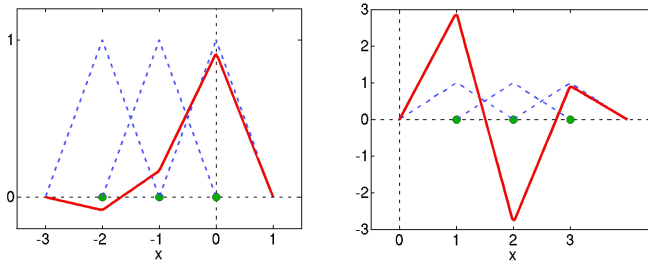
To remind the reader about the differences between the above defined position-dependent kernel and that of [6] some important features of the original kernel are emphasized. Namely,

- **Kernel nodes.** Symmetric nodes are chosen to be evenly distributed around the origin, i.e., the shift function $\lambda(\bar{x})$ is always taken to be 0 for the symmetric kernel.
- **Kernel coefficients.** Since the kernel nodes are independent of the evaluation points for the symmetric kernel, the coefficients are fixed and therefore only need to be calculated once.

The main idea of the modified kernel is to use a shift function $\lambda(\bar{x})$ that changes such that 1) the support of the kernel $K_h^{r+1,\ell}$ lies within the spatial domain $[a, b]$ even when it is applied near a boundary; 2) the post-processor does not (re)introduce unphysical discontinuities in the solution; 3) the post-processor is “as symmetric as possible” (λ is taken to be as close to zero as possible), since a symmetric distribution of the kernel nodes is known to give a more accurate result. This λ can be seen in Figure 4.3. Notice that for points near the boundary a partly one-sided



(A) Symmetric kernel: $\lambda = 0$



(B) Left-sided kernel: $\lambda = -1$ (C) Partly right-sided kernel: $\lambda = 2$

FIGURE 4.2. The SIAC kernel (solid line) is a linear combination of $r + 1$ B-splines of order $k + 1$ (dashed lines) that are centered around kernel nodes (circles). In these examples, $r = 2$ and $k + 1 = 2$. Observe that the location of the kernel nodes affects the shape and support of the kernel.

kernel is used, which requires calculating the kernel coefficients at each evaluation point.

Remark 4.1. Using the definition of the kernel nodes, \bar{x}_γ , it is easy to obtain

$$\begin{aligned}
 (K_h^{r+1,\ell} \star u_h)(\bar{x}) &= \int_{\mathbb{R}} K_h^{r+1,\ell}(\bar{x} - z)u_h(z)dz \\
 &= \frac{1}{h} \sum_{\gamma=0}^r c_\gamma \int_{\mathbb{R}} \psi^{(\ell)}\left(\frac{\bar{x} - z}{h} - \bar{x}_\gamma\right) u_h(z)dz \\
 (4.10) \quad &= \sum_{\gamma=0}^r c_\gamma \int_{\mathbb{R}} \psi^{(\ell)}(z)u_h(\bar{x} - h(z + \bar{x}_\gamma))dz.
 \end{aligned}$$

Using

$$(4.11) \quad \bar{x} \in [a, b], \quad z \in \left[-\frac{\ell}{2}, \frac{\ell}{2}\right], \quad \lambda(\bar{x}) \in \left[-\frac{r + \ell + \epsilon}{2}, \frac{r + \ell + \epsilon}{2}\right],$$

this gives

$$(4.12) \quad \bar{x} - h\bar{x}_\gamma \in \left[a + \frac{\ell + \epsilon}{2}h, b - \frac{\ell + \epsilon}{2}h\right], \quad \bar{x} - h(z + \bar{x}_\gamma) \in \left[a + \frac{\epsilon}{2}h, b - \frac{\epsilon}{2}h\right].$$

From the above it can clearly be seen that the position-dependent kernel can be applied to the entire domain.

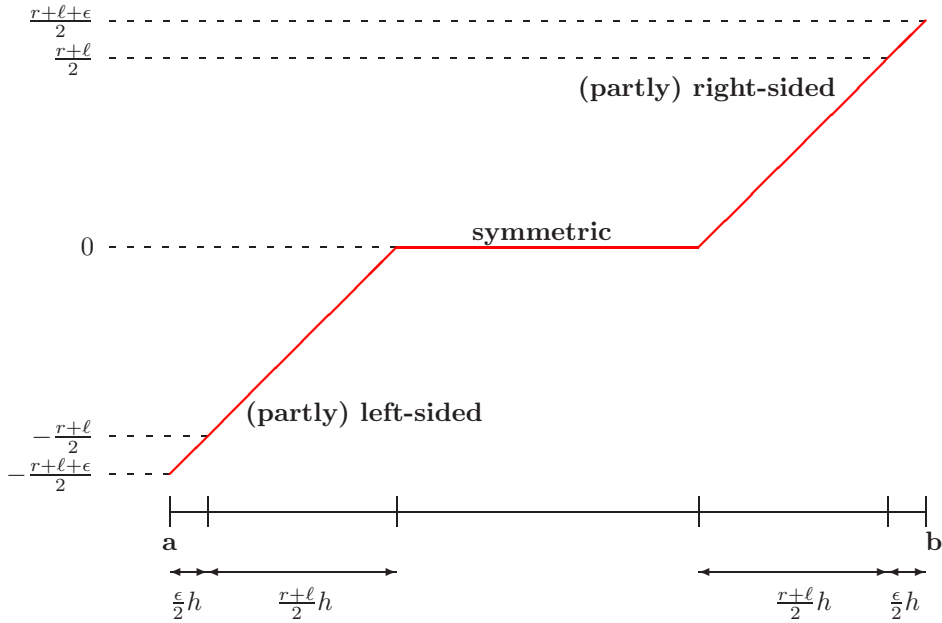


FIGURE 4.3. Shift Function $\lambda(\bar{x})$

To remind the reader of how to apply the kernel to multiple dimensions, take $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$. The position-dependent kernel is then taken to be a tensor product of the one-dimensional kernel

$$(4.13) \quad K_h^{r+1,\ell}(\mathbf{x}) := \left[\frac{1}{h} \sum_{\gamma_1=0}^r c_{\gamma_1}^{(1)} \psi^{(\ell)} \left(\frac{x_1}{h} - \bar{x}_{\gamma_1}^{(1)} \right) \right] \cdots \left[\frac{1}{h} \sum_{\gamma_d=0}^r c_{\gamma_d}^{(d)} \psi^{(\ell)} \left(\frac{x_d}{h} - \bar{x}_{\gamma_d}^{(d)} \right) \right].$$

Denote $\psi_h^{(\ell)}(\mathbf{x}) = \frac{1}{h^d} \psi^{(\ell)}\left(\frac{\mathbf{x}}{h}\right)$, then the above definition is simplified to

$$(4.14) \quad K_h^{r+1,\ell}(\mathbf{x}) = \sum_{\gamma=0}^r k_{\gamma}^{r+1} \psi_h^{(\ell)}(\mathbf{x} - h\bar{\mathbf{x}}_{\gamma}),$$

where $\bar{\mathbf{x}}_{\gamma} = (\bar{x}_{\gamma_1}^{(1)}, \dots, \bar{x}_{\gamma_d}^{(d)})$ and $k_{\gamma}^{r+1} = c_{\gamma_1}^{(1)} \cdots c_{\gamma_d}^{(d)}$.

For a discussion of the computational implementation and efficiency, the reader is referred to [10].

5. THE SUPERCONVERGENT ERROR ESTIMATES

In this section the superconvergent error estimates are given proving that $\mathcal{O}(h^{2k+1})$ accuracy is achieved in the \mathcal{L}^2 -norm and $\mathcal{O}(h^{\min\{2k+1, 2k+2-\frac{d}{2}\}})$ in the \mathcal{L}^∞ -norm throughout the *entire* domain. In order to establish such estimates it is first necessary to recall important properties of the convolution kernel.

5.1. Important kernel properties. In this section, kernel properties that are useful in obtaining the error estimates are outlined. To begin, a reminder that the kernel is constructed in such a way that $K_h^{r+1,\ell}(\mathbf{x})$ reproduces polynomials of degree no more than r in each variable. This is stated by the following theorem:

Lemma 5.1 (Reproduction of polynomials). *Let $\alpha = (\alpha_1, \dots, \alpha_d)$ be a multi-index with $0 \leq |\alpha| \leq r$. Then*

$$(5.1) \quad (K_h^{r+1,\ell} \star p)(\mathbf{x}) = p(\mathbf{x}), \quad \text{with } p(\mathbf{x}) = \mathbf{x}^\alpha,$$

where $\mathbf{x}^\alpha = x_1^{\alpha_1} \dots x_N^{\alpha_N} \in \mathbb{R}^d$.

Proof. The proof of this lemma follows by noting that

$$\begin{aligned} (K_h^{r+1,\ell} \star p)(\mathbf{x}) &= \left(\int_{\mathbb{R}} K_h^{r+1,\ell}(y_1)(x_1 - y_1)^{\alpha_1} dy_1 \right) \\ &\quad \dots \left(\int_{\mathbb{R}} K_h^{r+1,\ell}(y_d)(x_d - y_d)^{\alpha_d} dy_d \right) \\ &= \left(x_1^{\alpha_1} + \sum_{\beta_1=1}^{\alpha_1} \binom{\alpha_1}{\beta_1} x_1^{\alpha_1-\beta_1} (-1)^{\beta_1} \int_{\mathbb{R}} K_h^{r+1,\ell}(y_1)(y_1)^{\beta_1} dy_1 \right) \\ &\quad \dots \left(x_d^{\alpha_d} + \sum_{\beta_d=1}^{\alpha_d} \binom{\alpha_d}{\beta_d} x_d^{\alpha_d-\beta_d} (-1)^{\beta_d} \int_{\mathbb{R}} K_h^{r+1,\ell}(y_d)(y_d)^{\beta_d} dy_d \right). \end{aligned}$$

Then use equation (4.7) to cancel the integrals and obtain

$$(5.2) \quad K_h^{r+1,\ell} \star p(\mathbf{x}) = p(\mathbf{x}), \quad \text{for } p(\mathbf{x}) = \mathbf{x}^\alpha. \quad \square$$

This property means that the kernel $K_h^{r+1,\ell}$ is constructed in such a way that $K_h^{r+1,\ell} \star u$ is an approximation of order $r + 1$ to u . More precisely, the following is true:

Lemma 5.2. *For $u \in H^s(\Omega)$ and $h > 0$,*

$$(5.3) \quad \|u - K_h^{r+1,\ell} \star u\|_{\infty,\Omega} \leq C \left(\sup_{x \in \Omega} \sum_{\gamma} |k_{\gamma}^{r+1}| \right) |u|_{s,\Omega} h^s, \quad 0 \leq s \leq r + 1,$$

where C is a positive constant independent of h .

Proof. This was shown in [4] for the symmetric kernel. Since this proof deals with the kernel applied to the exact solution, it is easily extended to include a larger subset of the domain by implementing the new definition for the kernel nodes, equation (4.8). □

5.2. The main estimates. In the previous section, basic results for a general kernel, $K_h^{r+1,\ell}$, were given. In practice, common choices for the kernel are $r = 2k$ for the interior, and $\ell = k + 1$. Near the boundary, it was computationally demonstrated in [14] that $r = 4k$ is a suitable choice to produce errors at the boundary with the same magnitude as the interior. If smoothness is required, in transition regions it is

necessary to use a convex combination of these kernels so that the post-processed solution is given by

$$(5.4) \quad u_h^*(\mathbf{x}) = \theta(\mathbf{x})(K_h^{2k+1,k+1} \star u_h)(\mathbf{x}) + (1 - \theta(\mathbf{x}))(K_h^{4k+1,k+1} \star u_h)(\mathbf{x}).$$

Recall that θ is defined by equation (4.3), so that the kernel using $4k + 1$ B-splines is only applied near the boundaries and the symmetric post-processor is applied whenever possible.

Remark 5.1. Note that a choice of $r = 2k$ near the boundaries will produce the superconvergent rate of $2k + 1$, but the magnitude of the errors compared to the interior are worse [13]. There is no theoretical benefit *with respect to convergence rate* for using a larger number of B-splines as this is controlled by the negative-order norm of the discontinuous Galerkin solution, which is at best $2k + 1$ order accuracy. It is the entire estimate for the approximation properties of the kernel that reduces with a larger number of B-splines near the boundaries. This will be discussed within the proof of the error estimates.

Now the post-processed error,

$$(5.5) \quad \|u - u_h^*\|_\Omega \leq \|u - u^*\|_\Omega + \|u^* - u_h^*\|_\Omega,$$

can be considered. We note that $\|u - u^*\|_\Omega$ is the error from the kernel construction and $\|u^* - u_h^*\|_\Omega$ is the error from the filtered DG approximation. Here $\|\cdot\|_\Omega$ is used to denote a general norm, which can be the \mathcal{L}^∞ -norm or the \mathcal{L}^2 -norm. The following superconvergence error estimates will be proven:

Theorem 5.3. *Let u be an exact solution and u_h^* be the post-processed DG approximation to the linear hyperbolic equation (1.1). For all h sufficiently small*

$$(5.6) \quad \|u - u_h^*\|_{\infty,\Omega} \leq Ch^s \|u_0\|_{2k+3+[d/2],\Omega}$$

and

$$(5.7) \quad \|u - u_h^*\|_{0,\Omega} \leq Ch^{2k+1} \|u_0\|_{2k+2,\Omega},$$

where $s = \min\{2k + 1, 2k + 2 - \frac{d}{2}\}$ and C is a positive constant independent of h .

Remark 5.2. For the proof of the \mathcal{L}^∞ -error, use of an inverse inequality makes the estimate suboptimal for $d \geq 3$. In practice, the optimal accuracy order of $2k + 1$ is obtained numerically.

In order to obtain the above superconvergent results it is necessary to estimate $\|u - u^*\|_\Omega$ and $\|u^* - u_h^*\|_\Omega$ separately.

5.2.1. *Estimate of $\|u - u^*\|_\Omega$.* The first result gives an estimate of the error between the exact solution u , and the post-processed exact solution. Consider the term

$$\begin{aligned} \|u - u^*\|_\Omega &\leq \|\theta(\mathbf{x})(u(\mathbf{x}) - (K_h^{2k+1,k+1} \star u)(\mathbf{x}))\|_\Omega \\ &\quad + \|(1 - \theta(\mathbf{x}))(u(\mathbf{x}) - (K_h^{4k+1,k+1} \star u)(\mathbf{x}))\|_\Omega \\ &\leq \|u(\mathbf{x}) - (K_h^{2k+1,k+1} \star u)(\mathbf{x})\|_\Omega + \|u(\mathbf{x}) - (K_h^{4k+1,k+1} \star u)(\mathbf{x})\|_\Omega \end{aligned}$$

$$\begin{aligned}
 &\stackrel{\text{Lemma 5.2}}{\leq} \left(C_1 |u|_{2k+1,\Omega} \left(\sup_{\mathbf{x} \in \Omega} \sum_{\gamma} |k_{\gamma}^{2k+1}| \right) h^{2k+1} \right. \\
 &\quad \left. + C_2 |u|_{4k+1,\Omega} \left(\sup_{\mathbf{x} \in \Omega} \sum_{\gamma} |k_{\gamma}^{4k+1}| \right) h^{4k+1} \right) \\
 (5.8) \quad &\leq C h^{2k+1}.
 \end{aligned}$$

For the symmetric kernel, the coefficients k_{γ} are fixed with the number of B-splines used, $2k + 1$. However, for the one-sided kernel these kernel coefficients not only depend on the number of B-splines used, but also the evaluation point. Because of the non-symmetric support of the kernel, this means that $\sum_{\gamma} |k_{\gamma}^{r+1}|$ near the boundary of the domain will tend to be larger than in interior domain. However, this is balanced by h^{r+1} , which for near the boundary gives the following relation:

$$(5.9) \quad \sum_{\gamma} |k_{\gamma}^{r+1}| h^{4k+1} \leq \sum_{\gamma} |k_{\gamma}^{2k+1}| h^{2k+1}$$

for h sufficiently small. This relation is obtained by computing the coefficients for the kernels using $2k + 1$ and $4k + 1$ B-splines through the use of Lemma 5.1. This process is described in detail in [10]. In Figure 5.1, a demonstration of this difference is illustrated for the case $k = 2$.

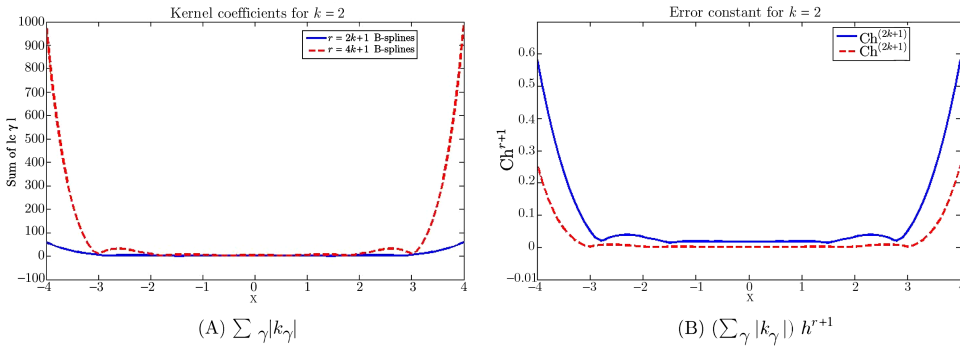


FIGURE 5.1. Plots demonstrating the effect of the coefficients on the error estimate for $r = 2k$ and $r = 4k$ B-splines.

5.2.2. *Estimate of $\|u^* - u_h^*\|_{\Omega}$.* In order to estimate the second term the following lemma will be needed for the general position-dependent kernel.

Lemma 5.4. *Let $\Omega_0 + 2\text{supp}(K_h) \subset\subset \Omega_1 \subset\subset \Omega$, $v \in \mathcal{L}^{\infty}(\Omega)$ and $d_0 = [d/2] + 1$. Then for all h sufficiently small*

$$(5.10) \quad \|\psi_h^{(\ell)} \star v(\mathbf{x})\|_{\infty,\Omega_0} \leq C \left(\sum_{|\alpha| \leq d_0 + \ell} \|\partial_h^{\alpha} v\|_{-\ell,\Omega_1} + h^{\ell} \sum_{|\alpha| \leq \ell} \|\partial_h^{\alpha} v\|_{\infty,\Omega_1} \right)$$

and

$$(5.11) \quad \|\psi_h^{(\ell)} \star v(\mathbf{x})\|_{0,\Omega_0} \leq C \sum_{|\alpha| \leq \ell} \|\partial_h^{\alpha} v\|_{-\ell,\Omega_1},$$

where C is a positive constant independent of h .

Proof. The first step in this estimate is to rewrite the function as

$$(5.12) \quad \psi_h^{(\ell)} \star v = K_h^{r+1, d_0} \star (\psi_h^{(\ell)} \star v) + \psi_h^{(\ell)} \star v - K_h^{r+1, d_0} \star (\psi_h^{(\ell)} \star v).$$

Using Lemmas 2.1, 2.2, 2.3 and 2.4 it is possible to show

$$\begin{aligned}
 \|K_h^{r+1, d_0} \star (\psi_h^{(\ell)} \star v)\|_{\infty, \Omega_0} &\stackrel{\text{Lemma 2.3}}{\leq} C \sum_{|\alpha| \leq d_0} \|D^\alpha (K_h^{r+1, d_0} \star (\psi_h^{(\ell)} \star v))\|_{0, \Omega_0} \\
 &\stackrel{\text{Lemma 2.2}}{\leq} C \sum_{|\alpha| \leq d_0} \|\partial_h^\alpha (\psi_h^{(\ell)} \star v)\|_{0, \Omega'_1} \\
 &\stackrel{\text{Lemma 2.1}}{=} C \sum_{|\alpha| \leq d_0} \|\psi_h^{(\ell)} \star \partial_h^\alpha v\|_{0, \Omega'_1} \\
 &\stackrel{\text{Lemma 2.4, 2.2, 2.1}}{\leq} C \sum_{|\alpha| \leq d_0} \left(\sum_{|\beta| \leq \ell} \|\psi_h^{(\ell-\beta)} \star \partial_h^\beta \partial_h^\alpha v\|_{-\ell, \Omega_1} \right) \\
 (5.13) \quad &\stackrel{\text{def. of negative order norm}}{\leq} C \sum_{|\alpha| \leq d_0 + \ell} \|\partial_h^\alpha v\|_{-\ell, \Omega_1},
 \end{aligned}$$

for some $\Omega_0 + 2\text{supp}(K_h) \subset\subset \Omega'_1 \subset\subset \Omega_1$. Next use Lemmas 5.2 and 2.2 to obtain

$$\begin{aligned}
 (5.14) \quad \|\psi_h^{(\ell)} \star v - K_h^{r+1, d_0} \star \psi_h^{(\ell)} \star v\|_{\infty, \Omega_0} &\leq Ch^\ell \sum_{|\alpha| \leq \ell} \|D^\alpha (\psi_h^{(\ell)} \star v)\|_{\infty, \Omega_0} \\
 &\leq Ch^\ell \sum_{|\alpha| \leq \ell} \|\partial_h^\alpha v\|_{\infty, \Omega_1}.
 \end{aligned}$$

Similarly, it is easy to see that

$$(5.15) \quad \|\psi_h^{(\ell)} \star v\|_{0, \Omega_0} \leq C \sum_{|\alpha| \leq \ell} \|D^\alpha (\psi_h^{(\ell)} \star v)\|_{-\ell, \Omega'_1} \leq C \sum_{|\alpha| \leq \ell} \|\partial_h^\alpha v\|_{-\ell, \Omega_1}.$$

Note again that the same notation, C , is used to denote the different constants. \square

Using the definition of convolution and the multi-dimensional post-processor we have

$$(5.16) \quad K_h^{r+1, \ell} \star (u - u_h)(\mathbf{x}) = \int_{\mathbb{R}^d} K_h^{r+1, \ell}(\mathbf{x} - \mathbf{z})(u - u_h)(\mathbf{z})d\mathbf{z}$$

$$(5.17) \quad = \sum_{\gamma=0}^r k_\gamma \int_{\mathbb{R}^d} \psi_h^{(\ell)}(\mathbf{x} - h\bar{\mathbf{x}}_\gamma - \mathbf{z})(u - u_h)(\mathbf{z})d\mathbf{z}$$

$$(5.18) \quad = \sum_{\gamma=0}^r k_\gamma \psi_h^{(\ell)} \star (u - u_h)(\mathbf{x} - h\bar{\mathbf{x}}_\gamma),$$

which gives

$$(5.19) \quad \|K_h^{r+1, \ell} \star (u - u_h)\|_{\infty, \Omega} \leq \left(\sup_{\mathbf{x} \in \Omega} \sum_{\gamma} |k_\gamma^{r+1}| \right) \|\psi_h^{(\ell)} \star (u - u_h)\|_{\infty, \Omega_0}$$

and

$$(5.20) \quad \|K_h^{r+1,\ell} \star (u - u_h)\|_{0,\Omega} \leq \left(\sup_{x \in \Omega} \sum_{\gamma} |k_{\gamma}^{r+1}| \right) \|\psi_h^{(\ell)} \star (u - u_h)\|_{0,\Omega_0},$$

here $\Omega_0 + 2\text{supp}(K_h) \subset\subset \Omega$. Next it can be shown that, similar to Lemma 5.4, there exists $\Omega_1 \subset\subset \Omega$ such that

$$(5.21) \quad \|K_h^{r+1,\ell} \star (u - u_h)\|_{\infty,\Omega} \leq C \left(\sup_{x \in \Omega} \sum_{\gamma} |k_{\gamma}^{r+1}| \right) \times \left(\sum_{|\alpha| \leq d_0 + \ell} \|\partial_h^{\alpha}(u - u_h)\|_{-\ell,\Omega_1} + h^{\ell} \sum_{|\alpha| \leq \ell} \|\partial_h^{\alpha}(u - u_h)\|_{\infty,\Omega_1} \right)$$

and

$$(5.22) \quad \|K_h^{r+1,\ell} \star (u - u_h)\|_{0,\Omega} \leq C \left(\sup_{x \in \Omega} \sum_{\gamma} |k_{\gamma}^{r+1}| \right) \sum_{|\alpha| \leq \ell} \|\partial_h^{\alpha}(u - u_h)\|_{-\ell,\Omega_1}.$$

The estimates bounding the domain Ω by norms over Ω_0 are obtained through the definition of the kernel nodes, equation (4.8), while the constant still relies on the original domain, Ω . The maximum value that this obtains is when $\lambda(\bar{x}) = \pm \frac{(r+\ell)}{2}$ and is found using Lemma 5.1.

The estimate for the second term is then

$$(5.23) \quad \|u^* - u_h^*\|_{\Omega} \leq \|\theta(\mathbf{x})K_h^{2k+1, k+1} \star (u - u_h) + (1 - \theta(\mathbf{x}))K_h^{4k+1, k+1} \star (u - u_h)\|_{\Omega}$$

$$(5.24) \quad \leq \|K_h^{2k+1, k+1} \star (u - u_h)\|_{\Omega} + \|K_h^{4k+1, k+1} \star (u - u_h)\|_{\Omega}.$$

Taking $\ell = k+1$ and $r = 2k$, $r = 4k$ separately and using the result (5.22) completes the error estimate,

$$(5.25) \quad \|u^* - u_h^*\|_{\infty,\Omega} \leq C \left(\sup_{\mathbf{x} \in \Omega} \sum_{\gamma} |k_{\gamma}^{4k+1}| \right) \times \left(\sum_{|\alpha| \leq d_0 + (k+1)} \|\partial_h^{\alpha}(u - u_h)\|_{-(k+1),\Omega_1} + h^{k+1} \sum_{|\alpha| \leq (k+1)} \|\partial_h^{\alpha}(u - u_h)\|_{\infty,\Omega_1} \right)$$

and

$$(5.26) \quad \|u^* - u_h^*\|_{0,\Omega} \leq C \left(\sup_{\mathbf{x} \in \Omega} \sum_{\gamma} |k_{\gamma}^{4k+1}| \right) \sum_{|\alpha| \leq k+1} \|\partial_h^{\alpha}(u - u_h)\|_{-(k+1),\Omega_1}.$$

Remark 5.3. Notice that the new definition of the kernel nodes, equation (4.8), shifts the support of the kernel depending on the evaluation point. This means that the support of the convolution kernel only excludes an arbitrarily small distance $\epsilon > 0$ from the domain boundary, which allows using the estimates (5.10) and (5.11) in equations (5.19) and (5.20). This is an improvement over the proofs given in [4, 6]. Further, the specific properties of the DG solution have not been used at this point. This guarantees that the proof will be valid for any numerical approximation, similar to Theorem 4.1.

With the help of the above two estimates and Theorem 3.1, the main Theorem 5.3 is finally proven.

6. NUMERICAL VALIDATION

Numerical validation in one and two dimensions was already provided in [14]. The tests in [14] verify the theoretical superconvergence error estimates for the position-dependent SIAC post-processor that are presented in this paper. They included discontinuous Galerkin solutions with Runge-Kutta time-stepping, periodic and Dirichlet boundary conditions, variable coefficients equations, and stationary shocks. Therefore, this section concentrates instead on the applicability of this position-dependent SIAC kernel as a streamline visualization filter. In order to do so, three two-dimensional tests of Steffan et al. [15] are considered.

In each test, a velocity profile (u, v) on the square $[-1, 1]^2$ is given. The exact solution is compared against the \mathcal{L}^2 -projection of that solution onto the space of piecewise polynomials of degree $k + 1$ on a uniform mesh with $N \times N$ elements as given in (3.1). The latter can be interpreted as a DG approximation at the initial time. The exact solution is also compared against the result of post-processing this \mathcal{L}^2 -projection. To obtain the post-processed approximation, the position-dependent post-processor described in Section 4.1 is used.

The implementation is as follows: first, the \mathcal{L}^2 -projection of the field is done onto the space of polynomials described by V_h . The solution is then convolved against the filter functions $K_h^{2k+1, k+1}$ for \mathbf{x} in the interior of the domain (a distance of $\frac{3k+1}{2}h$ away from the domain boundary) and $K_h^{4k+1, k+1}$ for \mathbf{x} near the domain boundary (a distance of $\frac{3k+5}{2}h$ near the domain boundary). These functions can be defined as

$$(6.1) \quad \mathbf{u}_h^{*,1}(\mathbf{x}) = (K_h^{2k+1, k+1} \star \mathbf{u}_h)(\mathbf{x}),$$

$$(6.2) \quad \mathbf{u}_h^{*,2}(\mathbf{x}) = (K_h^{4k+1, k+1} \star \mathbf{u}_h)(\mathbf{x}),$$

where the filtering is done componentwise. The filtered solution is then given by

$$(6.3) \quad \mathbf{u}_h^*(\mathbf{x}) = \theta(\mathbf{x})\mathbf{u}_h^{*,1}(\mathbf{x}) + (1 - \theta(\mathbf{x}))\mathbf{u}_h^{*,2}(\mathbf{x}).$$

as in equation (5.4), where $\mathbf{u}_h^*(\mathbf{x}) = (u_h^*(\mathbf{x}), v_h^*(\mathbf{x}))^T$. Once we have obtained the filtered solution, the streamlines are calculated:

$$(6.4) \quad \frac{d\mathbf{x}}{dt} = \mathbf{u}_h^*(\mathbf{x}).$$

In the following test cases, the velocity (u, v) is obtained as a function of (x, y) from the real and imaginary parts of a complex number ω :

$$u := \Re(\omega), \quad v := \Im(\omega),$$

where, defining the complex number $z := x + iy$, the following three test cases are given:

$$(6.5) \quad \begin{aligned} \omega &= (z - (0.74 + 0.35i))(z - (0.68 - 0.59i)) \\ &\quad (z - (-0.11 - 0.72i))(\bar{z} - (-0.58 + 0.64i)) \\ &\quad (\bar{z} - (0.51 - 0.27i))(\bar{z} - (-0.12 + 0.84i))^2 \end{aligned} \quad (\text{Case 1}),$$

$$(6.6) \quad \begin{aligned} \omega &= (z - (0.94 + 0.15i))(\bar{z} + (-0.38 - 0.39i)) \\ &\quad (z - (0.09 - 0.92i))(\bar{z} - (-0.38 + 0.84i)) \\ &\quad (\bar{z} - (0.71 - 0.07i)) \end{aligned} \quad (\text{Case 2}),$$

$$(6.7) \quad \begin{aligned} \omega &= - (z - (0.74 + 0.35i))(z - (0.11 - 0.11i))^2 \\ &\quad (z - (-0.11 + 0.72i))(z - (-0.58 + 0.64i)) \\ &\quad (\bar{z} - (0.51 - 0.27i)) \end{aligned} \quad (\text{Case 3}).$$

For each test case, the position-dependent post-processor enhances the convergence rate from $\mathcal{O}(h^{k+1})$ to at least $\mathcal{O}(h^{2k+1})$ in both the \mathcal{L}^∞ -norm (as predicted by theory), and the (weaker) \mathcal{L}^2 -norm (where h is the uniform element size). This can be seen from Tables 6.1, 6.2, and 6.3. Figure 6.1 illustrates the *local* accuracy improvement.

An interesting effect can be seen in Tables 6.2 and 6.3: for sufficiently large k , the errors of the post-processed field are of the order of the machine precision, which suggests that the exact solution has been reached. In the second case, for example, the exact solution is a polynomial of degree five. At the same time, the post-processed solution is a piecewise polynomial of degree at most $2k + 1$ ($= 5$ for $k = 2$) in each variable. The latter stems from the fact that the post-processed solution is obtained from the convolution of a piecewise polynomial of degree of at most k (the \mathcal{L}^2 -projection before post-processing) with a piecewise polynomial of degree k (the kernel, a linear combination of B-splines of degree $k + 1$) in each variable. As a consequence, the high accuracy suggests that the post-processed \mathcal{L}^2 -projection onto the space of piecewise polynomials of degree k behaves like the \mathcal{L}^2 -projection onto the space of piecewise polynomials of degree $2k + 1$ in each variable.

A good feature of the post-processor is that it can enhance the accuracy of streamlines, especially near critical points. This was observed by Steffan et al. [15] (Figure 3) for the symmetric post-processor, away from the boundary. Figure 6.2 shows that similar improvements are obtained for the position-dependent post-processor in the entire spatial domain (using a standard RK-4 method with the time step taken to be $\Delta t = 0.01$ to compute the streamlines). The second field of Steffan et al. was translated so that the critical points are located close to the boundary. This helps to emphasize the improved applicability and accuracy of the position-dependent post-processor near the boundary.

TABLE 6.1. The position-dependent post-processor applied to (Case 1) enhances the convergence rate from $\mathcal{O}(h^{k+1})$ to at least $\mathcal{O}(h^{2k+1})$ in both the \mathcal{L}^∞ -norm as predicted by theory, and the weaker \mathcal{L}^2 -norm.

<i>u</i> -component								
Mesh	\mathcal{L}^2 -error				\mathcal{L}^∞ -error			
	Before		After		Before		After	
	Error	Order	Error	Order	Error	Order	Error	Order
\mathcal{P}^1								
20	5.36e-02	-	3.58e-03	-	4.63e-01	-	2.26e-02	-
40	1.35e-02	1.99	1.20e-04	4.90	1.27e-01	1.87	8.48e-04	4.74
80	3.37e-03	2.00	5.98e-06	4.33	3.32e-02	1.93	2.95e-05	4.85
\mathcal{P}^2								
20	1.92e-03	-	6.01e-06	-	1.97e-02	-	7.56e-06	-
40	2.41e-04	2.99	2.00e-07	4.91	2.67e-03	2.89	2.19e-07	5.11
80	3.01e-05	3.00	4.23e-09	5.56	3.47e-04	2.94	4.25e-09	5.69
\mathcal{P}^3								
20	4.96e-05	-	1.08e-23	-	4.22e-04	-	1.84e-22	-
40	3.11e-06	4.00	7.03e-22	-	2.80e-05	3.91	1.40e-20	-
80	1.94e-07	4.00	2.13e-21	-	1.81e-06	3.96	7.16e-20	-
<i>v</i> -component								
Mesh	\mathcal{L}^2 -error				\mathcal{L}^∞ -error			
	Before		After		Before		After	
	Error	Order	Error	Order	Error	Order	Error	Order
\mathcal{P}^1								
20	1.23e-01	-	4.82e-03	-	1.40e+00	-	3.53e-02	-
40	3.10e-02	1.99	1.75e-04	4.78	3.77e-01	1.90	1.31e-03	4.75
80	7.75e-03	2.00	9.84e-06	4.16	9.78e-02	1.95	4.57e-05	4.85
\mathcal{P}^2								
20	4.20e-03	-	1.14e-05	-	4.78e-02	-	1.56e-05	-
40	5.27e-04	3.00	3.07e-07	5.21	6.37e-03	2.91	3.49e-07	5.48
80	6.59e-05	3.00	5.99e-09	5.68	8.21e-04	2.95	6.31e-09	5.79
\mathcal{P}^3								
20	9.09e-05	-	1.75e-23	-	8.18e-04	-	3.47e-22	-
40	5.69e-06	4.00	1.41e-21	-	5.37e-05	3.93	2.87e-20	-
80	3.56e-07	4.00	2.44e-21	-	3.44e-06	3.96	6.81e-20	-

TABLE 6.2. The position-dependent post-processor applied to (Case 2) enhances the convergence rate from $\mathcal{O}(h^{k+1})$ to at least $\mathcal{O}(h^{2k+1})$ in both the \mathcal{L}^∞ -norm as predicted by theory, and the weaker \mathcal{L}^2 -norm. For $k \geq 2$, the errors of post-processed field are of the order of the machine precision.

<i>u</i> -component								
Mesh	\mathcal{L}^2 -error				\mathcal{L}^∞ -error			
	Before		After		Before		After	
	Error	Order	Error	Order	Error	Order	Error	Order
\mathcal{P}^1								
20	2.00e-02	-	2.57e-04	-	1.31e-01	-	7.92e-04	-
40	5.00e-03	2.00	1.25e-05	4.36	3.42e-02	1.93	2.65e-05	4.90
80	1.25e-03	2.00	8.39e-07	3.90	8.76e-03	1.97	9.38e-07	4.82
\mathcal{P}^2								
20	4.65e-04	-	2.68e-26	-	2.80e-03	-	2.95e-25	-
40	5.83e-05	3.00	7.75e-25	-	3.62e-04	2.95	1.08e-23	-
80	7.28e-06	3.00	2.34e-24	-	4.60e-05	2.98	4.55e-23	-
\mathcal{P}^3								
20	6.80e-06	-	4.13e-24	-	2.53e-05	-	5.94e-23	-
40	4.25e-07	4.00	3.91e-22	-	1.61e-06	3.97	4.90e-21	-
80	2.66e-08	4.00	9.63e-22	-	1.02e-07	3.99	2.31e-20	-
<i>v</i> -component								
Mesh	\mathcal{L}^2 -error				\mathcal{L}^∞ -error			
	Before		After		Before		After	
	Error	Order	Error	Order	Error	Order	Error	Order
\mathcal{P}^1								
20	2.69e-02	-	2.72e-04	-	1.84e-01	-	8.04e-04	-
40	6.74e-03	2.00	1.42e-05	4.26	4.80e-02	1.94	2.73e-05	4.88
80	1.69e-03	2.00	9.50e-07	3.90	1.23e-02	1.97	1.05e-06	4.71
\mathcal{P}^2								
20	5.73e-04	-	7.39e-26	-	3.49e-03	-	1.04e-24	-
40	7.17e-05	3.00	1.45e-24	-	4.50e-04	2.96	2.82e-23	-
80	8.97e-06	3.00	3.63e-24	-	5.71e-05	2.98	9.69e-23	-
\mathcal{P}^3								
20	7.45e-06	-	1.01e-23	-	2.81e-05	-	1.77e-22	-
40	4.66e-07	4.00	7.18e-22	-	1.79e-06	3.98	1.22e-20	-
80	2.91e-08	4.00	1.68e-21	-	1.13e-07	3.99	4.11e-20	-

TABLE 6.3. The position-dependent post-processor for the v -component (Case 3) enhances the convergence rate from $\mathcal{O}(h^{k+1})$ to at least $\mathcal{O}(h^{2k+1})$ in both the \mathcal{L}^∞ -norm as predicted by theory, and the weaker \mathcal{L}^2 -norm. For $k \geq 2$, the errors of post-processed field are of the order of the machine precision.

u-component								
Mesh	\mathcal{L}^2 -error				\mathcal{L}^∞ -error			
	Before		After		Before		After	
	Error	Order	Error	Order	Error	Order	Error	Order
\mathcal{P}^1								
20	4.92e-02	-	1.34e-03	-	3.98e-01	-	8.15e-03	-
40	1.23e-02	1.99	3.73e-05	5.16	1.06e-01	1.90	2.82e-04	4.85
80	3.09e-03	2.00	1.56e-06	4.58	2.75e-02	1.95	8.84e-06	5.00
\mathcal{P}^2								
20	1.89e-03	-	4.50e-06	-	1.42e-02	-	4.78e-06	-
40	2.37e-04	3.00	1.11e-07	5.34	1.88e-03	2.92	7.47e-08	6.00
80	2.97e-05	3.00	2.03e-09	5.77	2.41e-04	2.96	1.17e-09	6.00
\mathcal{P}^3								
20	4.27e-05	-	3.98e-24	-	2.40e-04	-	5.29e-23	-
40	2.67e-06	4.00	5.44e-22	-	1.56e-05	3.94	1.45e-20	-
80	1.67e-07	4.00	1.25e-21	-	9.92e-07	3.97	4.04e-20	-
v-component								
Mesh	\mathcal{L}^2 -error				\mathcal{L}^∞ -error			
	Before		After		Before		After	
	Error	Order	Error	Order	Error	Order	Error	Order
\mathcal{P}^1								
20	4.17e-02	-	7.11e-04	-	2.04e-01	-	3.90e-03	-
40	1.05e-02	1.99	2.26e-05	4.98	5.39e-02	1.92	1.28e-04	4.93
80	2.62e-03	2.00	1.10e-06	4.35	1.39e-02	1.96	4.17e-06	4.94
\mathcal{P}^2								
20	1.66e-03	-	3.65e-26	-	6.80e-03	-	5.33e-25	-
40	2.08e-04	3.00	9.01e-25	-	8.95e-04	2.93	2.28e-23	-
80	2.60e-05	3.00	2.58e-24	-	1.15e-04	2.96	9.17e-23	-
\mathcal{P}^3								
20	3.79e-05	-	4.01e-24	-	1.24e-04	-	5.62e-23	-
40	2.37e-06	4.00	4.48e-22	-	8.05e-06	3.95	1.11e-20	-
80	1.48e-07	4.00	9.47e-22	-	5.12e-07	3.97	3.13e-20	-

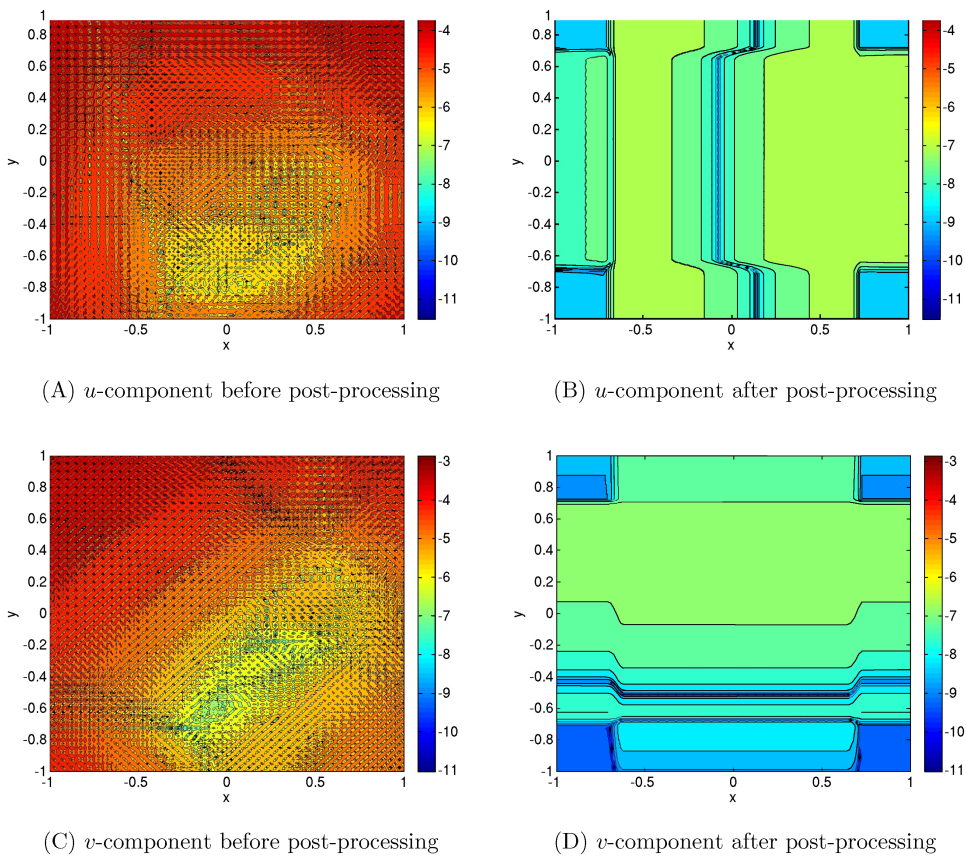


FIGURE 6.1. Contour plots of the absolute value of the error for Case 1. The position-dependent post-processor enhances the local accuracy in the entire domain, including the region near the boundary.

7. CONCLUSION

In this paper multi-dimensional superconvergent error estimates have been established confirming the numerical results for the position-dependent SIAC filter given in [14]. These estimates are valid for both general approximations as well as for discontinuous Galerkin approximations. This position-dependent Smoothness-Increasing Accuracy-Conserving post-processor enhances the accuracy in the entire spatial domain from $\mathcal{O}(h^{k+1})$ in the \mathcal{L}^2 -norm to $\mathcal{O}(h^{2k+1})$ in the \mathcal{L}^2 -norm and $\mathcal{O}(h^{\min\{2k+1, 2k+2-\frac{d}{2}\}})$ in the \mathcal{L}^∞ -norm (where k is the polynomial degree and h is the mesh element diameter). Because many numerical results were given in [14], this paper concentrated on the applicability as a streamline visualization filter.

The advantages of this filter are that it needs to be applied only once, at the final time, and it does not contain any information of the underlying physics or numerics. Unlike the traditional symmetric post-processor, the position-dependent post-processor can be applied in the entire domain, with proven superconvergence obtained in both the \mathcal{L}^2 - and \mathcal{L}^∞ -norms.

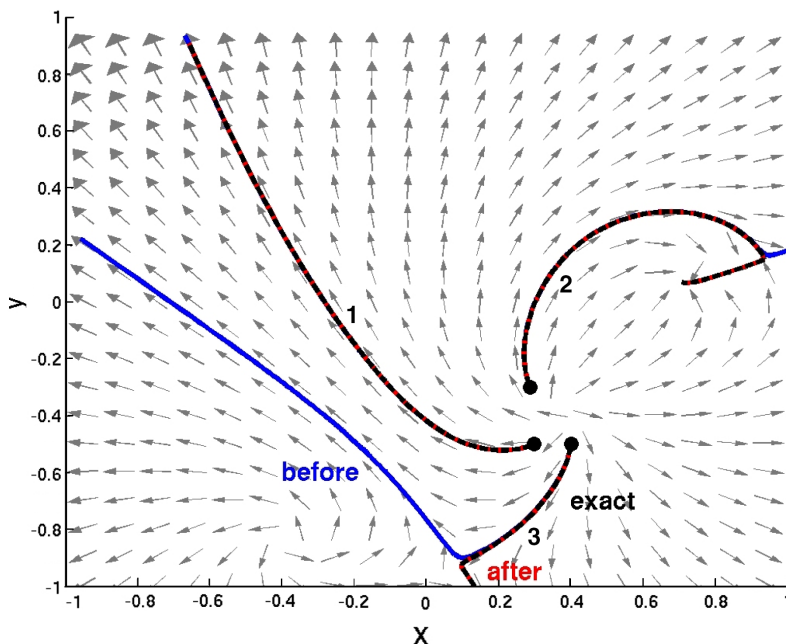


FIGURE 6.2. The Smoothness-Increasing Accuracy-Conserving filter improves the accuracy of the streamlines, started at the seeding points (circles). To see this, consider streamline 1 where the DG, SIAC DG, and exact streamlines agree. In streamlines 2 and 3, the streamlines obtained from the DG approximation diverges from the exact streamline near the boundary. The SIAC filtered DG solution follows the exact streamline.

REFERENCES

- [1] Slimane Adjerid, Karen D. Devine, Joseph E. Flaherty, and Lilia Krivodonova, *A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems*, *Comput. Methods Appl. Mech. Engrg.* **191** (2002), no. 11-12, 1097–1112, DOI 10.1016/S0045-7825(01)00318-8. MR1877682 (2002k:65137)
- [2] Slimane Adjerid and Mahboub Baccouch, *The discontinuous Galerkin method for two-dimensional hyperbolic problems. I. Superconvergence error analysis*, *J. Sci. Comput.* **33** (2007), no. 1, 75–113, DOI 10.1007/s10915-007-9144-x. MR2338333 (2008j:65189)
- [3] James H. Bramble, Joachim A. Nitsche, and Alfred H. Schatz, *Maximum-norm interior estimates for Ritz-Galerkin methods*, *Math. Comput.* **29** (1975), 677–688. MR0398120 (53 #1975)
- [4] J. H. Bramble and A. H. Schatz, *Higher order local accuracy by averaging in the finite element method*, *Math. Comp.* **31** (1977), no. 137, 94–111. MR0431744 (55 #4739)
- [5] Yingda Cheng and Chi-Wang Shu, *Superconvergence of discontinuous Galerkin and local discontinuous Galerkin schemes for linear hyperbolic and convection-diffusion equations in one space dimension*, *SIAM J. Numer. Anal.* **47** (2010), no. 6, 4044–4072, DOI 10.1137/090747701. MR2585178 (2011e:65187)
- [6] Bernardo Cockburn, Mitchell Luskin, Chi-Wang Shu, and Endre Süli, *Enhanced accuracy by post-processing for finite element methods for hyperbolic equations*, *Math. Comp.* **72** (2003), no. 242, 577–606, DOI 10.1090/S0025-5718-02-01464-3. MR1954957 (2004g:65129)

- [7] Bernardo Cockburn and Chi-Wang Shu, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Comput. **16** (2001), no. 3, 173–261, DOI 10.1023/A:1012873910884. MR1873283 (2002i:65099)
- [8] Liangyue Ji, Yan Xu, and Jennifer K. Ryan, *Accuracy-enhancement of discontinuous Galerkin solutions for convection-diffusion equations in multiple-dimensions*, Math. Comp. **81** (2012), no. 280, 1929–1950, DOI 10.1090/S0025-5718-2012-02586-5. MR2945143
- [9] Hanieh Mirzaee, Liangyue Ji, Jennifer K. Ryan, and Robert M. Kirby, *Smoothness-increasing accuracy-conserving (SIAC) postprocessing for discontinuous Galerkin solutions over structured triangular meshes*, SIAM J. Numer. Anal. **49** (2011), no. 5, 1899–1920, DOI 10.1137/110830678. MR2837489 (2012i:65209)
- [10] Hanieh Mirzaee, Jennifer K. Ryan, and Robert M. Kirby, *Efficient implementation of smoothness-increasing accuracy-conserving (SIAC) filters for discontinuous Galerkin solutions*, J. Sci. Comput. **52** (2012), no. 1, 85–112, DOI 10.1007/s10915-011-9535-x. MR2923520
- [11] K. Mustapha, *A superconvergent discontinuous Galerkin method for voltaerra integro-differential equations*, Math. Comp., **82** (2013), 1987–2005. MR3073189
- [12] Jennifer K. Ryan and Bernardo Cockburn, *Local derivative post-processing for the discontinuous Galerkin method*, J. Comput. Phys. **228** (2009), no. 23, 8642–8664, DOI 10.1016/j.jcp.2009.08.017. MR2558770 (2010j:65192)
- [13] Jennifer Ryan and Chi-Wang Shu, *On a one-sided post-processing technique for the discontinuous Galerkin methods*, Methods Appl. Anal. **10** (2003), no. 2, 295–307. MR2074753
- [14] Paulien van Slingerland, Jennifer K. Ryan, and C. Vuik, *Position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering for improving discontinuous Galerkin solutions*, SIAM J. Sci. Comput. **33** (2011), no. 2, 802–825, DOI 10.1137/100782188. MR2801190 (2012g:65211)
- [15] M. Steffan, S. Curtis, R.M. Kirby, and J.K. Ryan, *Investigation of smoothness enhancing accuracy-conserving filters for improving streamline integration through discontinuous fields*, IEEE-TVCG, **14** (2008), pp. 680-692.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA, HEFEI, ANHUI 230026, PEOPLE'S REPUBLIC OF CHINA.

Current address: University of Minnesota, School of Mathematics, 206 Church St. SE Minneapolis, Minnesota 55455

DELFT INSTITUTE OF APPLIED MATHEMATICS, DELFT UNIVERSITY OF TECHNOLOGY, 2628 CD DELFT, THE NETHERLANDS

DELFT INSTITUTE OF APPLIED MATHEMATICS, DELFT UNIVERSITY OF TECHNOLOGY, 2628 CD DELFT, THE NETHERLANDS

Current address: University of East Anglia, School of Mathematics, Norwich Research Park, Norwich NR4 7TJ, United Kingdom

E-mail address: Jennifer.Ryan@uea.ac.uk

DELFT INSTITUTE OF APPLIED MATHEMATICS, DELFT UNIVERSITY OF TECHNOLOGY, 2628 CD DELFT, THE NETHERLANDS