

Exploiting Superconvergence Through Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering

Jennifer K. Ryan

Abstract There has been much work in the area of superconvergent error analysis for finite element and discontinuous Galerkin (DG) methods. The property of superconvergence leads to the question of how to exploit this information in a useful manner, mainly through superconvergence extraction. There are many methods used for superconvergence extraction such as projection, interpolation, patch recovery and B-spline convolution filters. This last method falls under the class of Smoothness-Increasing Accuracy-Conserving (SIAC) filters. It has the advantage of improving both smoothness and accuracy of the approximation. Specifically, for linear hyperbolic equations it can improve the order of accuracy of a DG approximation from $k + 1$ to $2k + 1$, where k is the highest degree polynomial used in the approximation, and can increase the smoothness to $k - 1$. In this article, we discuss the importance of overcoming the mathematical barriers in making superconvergence extraction techniques useful for applications, specifically focusing on SIAC filtering.

1 Introduction

Many numerical methods experience a phenomenon known as superconvergence. **Superconvergence** is higher than theoretical predicted convergence:

$$|(u - u_h)(\xi)| \leq Ch^{r+\sigma},$$

where r is the expected convergence and $\sigma > 0$ [22]. So-called “natural” Superconvergence occurs when the function is evaluated at a point and compared with the exact solution. We can create globally superconvergent solutions through post-processing the approximation. In this article we focus on a specific post-processing technique that uses B-spline convolution to obtain a superconvergent

J.K. Ryan (✉)
University of East Anglia, Norwich, UK
e-mail: Jennifer.Ryan@uea.ac.uk

approximation. Specifically, we concentrate on SIAC filters, which have their roots in work by Bramble and Schatz [2] and Cockburn, Luskin, Shu and Süli [6].

2 Motivation and Background

We frame our discussion in the context of a linear hyperbolic equation with smooth initial data,

$$u_t + \sum_{i=1}^d A_i \frac{\partial}{\partial x_i} u + A_0 u = 0, \quad \mathbf{x} \in \Omega \times [0, T], \quad (1)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (2)$$

We also assume periodic boundary conditions for simplicity. For these types of equations, the superconvergence property is straight-forward to prove in both the pointwise setting and in terms of the negative-order norm.

2.1 Discontinuous Galerkin Methods

The important components that aid in creating a superconvergent approximation from a discontinuous Galerkin solution are that

1. The approximation space consists of piecewise polynomials of degree $\leq k$:

$$V_h^k = \{v \in L^2(\Omega) : v \in \mathbb{P}^k(\tau_e), j = 1, \dots, N\} \quad (3)$$

where τ_e are the elements in the associated mesh and $\Omega = \cup_e \tau_e$.

2. The variational formulation of the discontinuous Galerkin scheme:

$$\begin{aligned} \int_{\tau_e} (u_h)_t v_h(x) dx - \sum_{i=1}^d \int_{\tau_e} A_i u_h(\mathbf{x}, t) (v_h)_{x_i}(x) dx + \int_{\tau_e} A_0 u_h(\mathbf{x}, t) v_h d\mathbf{x} \\ + \sum_{i=1}^d \int_{\partial \tau_e} \widehat{A_i u_h} \hat{n}_i v_h, ds = 0 \end{aligned}$$

3. The weak continuity at the element interfaces that are enforced through the choice of the fluxes in the discontinuous Galerkin scheme.

The reader is advised to consult [5] for a more detailed discussion of the discontinuous Galerkin method.

2.2 Error Estimates: Convergence and Superconvergence

Assuming the initial condition is regular enough, the errors in L^2 for the DG approximation are given by

$$\|u - u_h\|_0 \leq C h^{k+1} |u_0|_{H^{k+2}} \quad (4)$$

[5]. However, Adjerid et al. noted that the approximation has the property of pointwise superconvergence [1]. That is, the *local* error at the “outflow” edge converges at twice the usual convergence rate,

$$(u - u_h)(x_{j+1/2}^-) = \alpha_{k+1} \frac{(-a)^{k+1} k!}{2k+1} h^{2k+2} + \mathcal{O}(h^{2k+3}) \quad (5)$$

for linear equations such as $u' - au = 0$. This occurs at the roots of the right Radau polynomial.

3 Extracting Superconvergence

We would like to turn the local superconvergence property into a globally superconvergent solution. There are many different options for this to be accomplished. A few are to interpolate using superconvergent fluxes [4, 15], elementwise post-processing [3], or convolution kernel post-processing [2, 6]. We focus on the latter, specifically the Smoothness-Increasing Accuracy-Conserving filter [10, 18, 21]. This last technique allows for *global superconvergence* and *smoothness*.

3.1 Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering

The SIAC filter has its roots in an accuracy-enhancing post-processor. Motivated by the work of Mock and Lax [14], Bramble and Schatz introduced a central B-spline kernel to post-process finite element approximations to elliptic equations [2]. This was also explored from a Fourier perspective and for derivative filtering by Thomeé [20]. Cockburn, Luskin, Shu and Süli then extended it to discontinuous Galerkin approximations to linear hyperbolic equations [6]. It was further extended to a broader class of problems in [7, 8, 11].

The basic idea of the original post-processor, $u^*(x)$, is to convolve the numerical approximation with a B-spline kernel,

$$u^*(x) = (K_H^{2(k+1), k+1} * u_h(\cdot, T))(x). \quad (6)$$

This allows us to achieve $u - u_h \sim \mathcal{O}(h^{2k+1})$ in L^2 as shown in [6]. A more general form of the B-spline kernel, $K_H^{2(k+1),k+1}(x)$, will be discussed in Sect. 3.2.

The post-processor is useful for removing the highly oscillatory errors in the discontinuous Galerkin approximation. The result is a solution that has increased smoothness and accuracy.

3.2 The SIAC Kernel

The SIAC kernel is a more general form of the B-spline kernel above. It is a linear combination of suitably scaled B-spline translates,

$$K_H^{(r+1,\ell)}(x) = \frac{1}{H} \sum_{\gamma=0}^r c_\gamma^{(r+1,\ell)} \psi^{(\ell)}\left(\frac{x}{H} - x_\gamma\right), \quad (7)$$

where $r + 1$ is the number of B-splines in the kernel and ℓ is the order of the B-splines. In (7), c_γ are weights of the B-splines, $\psi^{(\ell)}(x)$, and are determined by reproducing polynomials of degree less than or equal to r . For the original kernel $r = 2k$, $\ell = k + 1$ and $x_\gamma = -k + \gamma$ as given in (6). In the more general SIAC filter, x_γ depends on the point being evaluated and we have more flexibility both in the number of B-splines and order of the B-splines.

Central B-splines are defined as $\psi^{(1)} = \chi_{[-\frac{1}{2}, \frac{1}{2}]}$, $\psi^{(\ell)} = \psi^{(\ell-1)} * \chi_{[-\frac{1}{2}, \frac{1}{2}]}$, $\ell \geq 2$. Here, χ is equal to one on $[-\frac{1}{2}, \frac{1}{2}]$ and otherwise is zero. The central B-splines that form the post-processed solution are chosen because of their compact support of $\text{supp}\{\psi^{(\ell)}\} = [-\frac{\ell}{2}, \frac{\ell}{2}]$. Further, they are easy to compute through a recurrence relation. Lastly, there is a natural relation between their derivatives and divided differences: $D^\alpha \psi^{(\ell)} = \partial_H^\alpha \psi^{(\ell-\alpha)}$. In Fig. 1 a plot of the B-splines making up the convolution kernel as well as the convolution kernel is shown for $k = 2$.

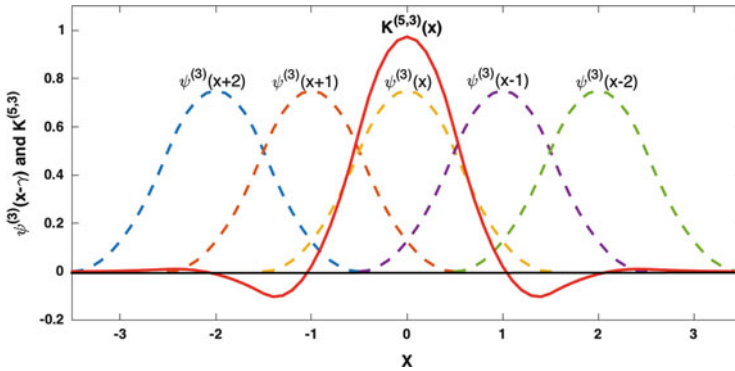


Fig. 1 Dashed lines: the B-splines, $\psi^{(3)}(x + k - \gamma)$, $\gamma = 0, \dots, 4$ used in the $k = 2$ kernel. Solid line: the kernel, $K_H^{5,3}(x)$ for $k = 2$

The convolution coefficients that weigh the B-spline translates are found by using the property of polynomial reproduction.

As an example, we give the original symmetric B-spline kernel for the second order approximation, $k = 1$. The kernel coefficients are found by using $K_h^{4,2} * p = p$ for $p = 1, x, x^2$. This creates the kernel

$$K^{4,2}(x) = \frac{-1}{12} \psi^{(2)}(x-1) + \frac{7}{6} \psi^{(2)}(x) - \frac{1}{12} \psi^{(2)}(x+1). \quad (8)$$

To summarise, the convolution kernel is designed to extract higher order accuracy through polynomial reproduction. It induces smoothness of $\mathcal{C}^{\ell-2}$ through the convolution with the B-splines and uses a local stencil of size $(r + \ell)H$. The kernel is a polynomial of degree $\ell - 1$, making the post-processed solution a polynomial of degree $\ell + k$. It has theoretical and numerical convergence of $\mathcal{O}(h^s)$, $s = \min\{r + 1, 2k + 1\}$ in both L^2 - and L^∞ -norms for linear hyperbolic equations over uniform meshes.

3.3 Implementing the Post-Processor

Assuming the one-dimensional discontinuous Galerkin approximation can be written as

$$u_h(x, t) = \sum_{n=0}^k u_e^{(n)}(t) \phi_e^{(n)}(x), \quad x \in \tau_e, \quad (9)$$

where $\phi_e^{(n)}(x)$ are the basis functions for the DG approximation. Using this modal form of the DG approximation, the post-processed solution can be written as

$$u^*(x) = \sum_{j=-p'}^{p'} \sum_{n=0}^k C(j, n, k, x) u_{e+j}^{(n)} \quad (10)$$

where $p' = \lceil \frac{r+\ell}{2} \rceil$ and

$$C(j, n, k, x) = \frac{1}{h} \sum_{\gamma=0}^r c_{\gamma}^{r+1, \ell} \underbrace{\int_{I_{e+j}} \psi^{(\ell)} \left(\frac{y-x}{h} - \gamma \right) \phi_{e+j}^{(n)}(y) dy}_{\in \mathbb{P}^{\ell+k}}. \quad (11)$$

The multi-dimensional kernel is a tensor product of the one-dimensional kernel. For example, in two-dimensions,

$$K_H(x, y) = \frac{1}{H_x H_y} \sum_{\gamma_x=0}^r \sum_{\gamma_y=0}^r c_{\gamma_x}^{r+1, \ell} c_{\gamma_y}^{r+1, \ell} \psi^{(\ell)} \left(\frac{x}{H_x} - x_{\gamma_x} \right) \psi^{(\ell)} \left(\frac{y}{H_y} - y_{\gamma_y} \right). \quad (12)$$

It is expected that the kernel can be applied to \mathbb{Q}^k -polynomial approximations, but it is also effective for \mathbb{P}^k -polynomial approximations.

3.4 Convergence of the SIAC Filtered Solution

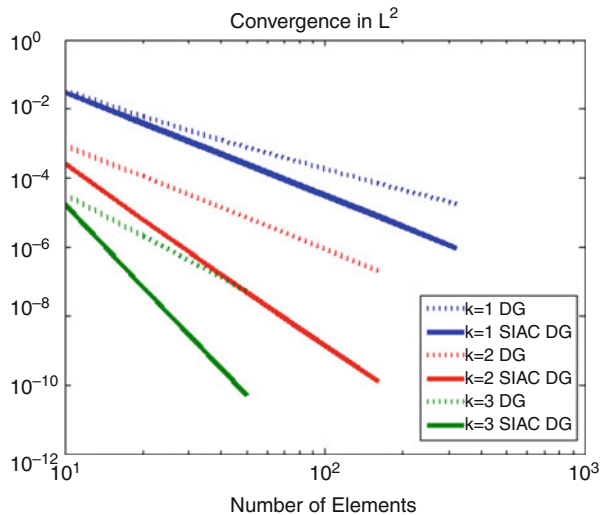
Let $u_h^*(x, T) = K_H * u_h$ be the post-processed DG approximation at the *final time*. Then the errors for the post-processed solution are given by

$$\|u - u_h^*(x, T)\|_{0, \Omega} \leq \underbrace{\|u - u^*\|_{0, \Omega}}_{\text{Exact filtered}} + \underbrace{\|(u - u_h)^*\|_{0, \Omega}}_{\text{DG errors}}. \quad (13)$$

The estimate for the first term comes about from the ability of the kernel to reproduce polynomials of degree r . Then, using a Taylor expansion we obtain $\|u - K_H * u\|_{\Omega} \leq Ch^{r+1}$ [6, 9]. The second term can be bounded by the negative-order norm [2, 6]. If we can show the negative-order norm is of higher order, then we can demonstrate superconvergence of the filtered solution.

In Fig. 2 a comparison of the convergence rates and errors between the discontinuous Galerkin approximation and the SIAC filtered approximation is given. If

Fig. 2 A comparison of the convergence rates and errors between the discontinuous Galerkin approximation and the SIAC filtered DG approximation



we consider the $k = 1$ filtered approximation and compare it with the $k = 2$ DG approximation, we can see that although they have the same convergence rate, the errors for the $k = 2$ approximation are better.

3.5 Applications

Currently, the applications of SIAC filtering include extracting accuracy out of existing code [18] and visualization filtering [19]. However, there is promising relations to image processing [13, 23] as well as potential in LES filtering [7, 8].

3.6 Interesting Challenges

The challenge in making SIAC filtering applicable to broader areas of applications include: A negative-order norm estimate that depends upon the PDE, the ability to extract derivative information, filtering near a boundary, and most importantly mesh geometry. In the following sections we discuss the challenges in extending SIAC filtering to a range of applications.

4 The Error Estimate

Recall that in Eq. (13) the SIAC filtered error estimate is controlled by our ability to prove superconvergence in the negative-order norm, where the negative-order norm is given by

$$\|\partial_H^\alpha(u - u_h)\|_{-(k+1),\Omega} = \sup_{\phi \in \mathcal{C}_0^\infty(\Omega)} \frac{(\partial_H^\alpha(u - u_h), \phi)_\Omega}{\|\phi\|_{k+1,\Omega}} \leq C h^{2k+1} \|u_0\|_{k+1,\mathcal{D}\Omega_1} \quad (14)$$

if $H = h$ [6].

For the negative-order norm, we actually only need to consider the numerator in Eq. (14). In general, the estimate depends on defining a suitable dual equation and we are able to prove $\|u - u_h^*\| \leq C h^{2k+m}$. Details of the existing estimates for various equations are provided in Table 1.

Table 1 SIAC Filter error estimates for various types of equations

m	$s = 2k + m$	Equation
2	$2k + 2$	Elliptic (FEM) [2]
1	$2k + 1$	Linear Hyperbolic (DG) [6]
$1 \leq m \leq 2$	$2k + 1 \leq s \leq 2k + 2$	Convection-diffusion (DG) [7]
1	$2k + 1$	Variable-Coefficient Hyperbolic (DG) [11]
$0, \frac{1}{2}, 2$	$2k \leq s \leq 2k + 2$	Nonlinear hyperbolic (DG) [8]

5 Derivative SIAC Filtering

Another interesting aspect of SIAC filtering is that it allows us to create a superconvergent approximation to derivatives. In general, the approximation obtained via a DG method will give $\|\partial^\alpha(u - u_h)\| \leq Ch^{k+1-\alpha}$ for the derivatives. This makes it impossible to obtain a good second order derivative approximation for $k = 1$. However, using SIAC filtering makes it possible to obtain higher order derivatives even for a piecewise linear approximation. In order to obtain a superconvergent derivative approximation, there are two options: accept a reduction in order of accuracy by taking the derivative of the filtered solution, or forming a kernel that uses higher-order B-splines whose errors do not reduce in order with differentiation. Each method has its advantages and disadvantages and both will give a superconvergent derivative approximation.

In the first method, we compute the derivative of the SIAC filtered solution directly. This gives

$$\frac{d^\alpha}{dx^\alpha} \left(K_h^{r+1,\ell} * u_h(\cdot, T) \right) (x) = \frac{d^\alpha}{dx^\alpha} \left(\frac{1}{H} \int_{\mathbb{R}} K_H^{(r+1,\ell)} \left(\frac{x-y}{h} \right) u_h(y, t) dy \right). \quad (15)$$

Recall that the post-processed approximation induces smoothness of $\mathcal{C}^{\ell-2}$ and is up to $2k + 1$ th-order accurate. If we calculate the derivative of the post-processing polynomial directly we would then have $\sim \mathcal{O}(h^{\min\{2k+2, r+2\}-\alpha})$, for $\alpha \leq \ell - 1$, which would give a reduced order of accuracy with each successive derivative. Further, the oscillations in the error increase [18]. This method may be more advantageous if only a first or second derivative is needed.

There is an alternative that allows us to obtain the same superconvergent approximation to the derivatives. That is, we can obtain a $2k + 1$ order accuracy approximation to the α th-derivative using higher order splines in our kernel [16, 20]. This gives a derivative approximation whose order or convergence is independent of α . The derivative kernel is defined as

$$K_H^{r+1,\alpha,\ell}(x) = \frac{1}{H} \sum_{\gamma=0}^r d_\gamma^{r+1,\alpha,\ell} \psi^{(\ell+\alpha)} \left(\frac{x}{H} - x_\gamma \right). \quad (16)$$

The difference to the kernel in Equation (7) is that it uses *smoother* B-splines. Note that smoother B-splines give an increased support size. Further, computing the α th derivative only requires computing the convolution of translations of the B-spline $\psi^{(\ell)}$ with u_h . This allows us to obtain the error estimate:

Theorem 1 (Ryan and Cockburn [16]) *Let u_h be the approximate solution given by the DG method for the model problem $u_t + (au)_x = 0$, $(x, t) \in \mathbb{R} \times (0, T)$. Assume that the initial data u_o is very smooth. Then*

$$\left\| \frac{d^\alpha}{dx^\alpha} u(x, T) - \frac{d^\alpha}{dx^\alpha} \left(K_h^{r+1, \alpha, \ell} * u_h(\cdot, T) \right) (x) \right\|_{0, \Omega_0} \leq C h^s,$$

where $s = \min\{r + 1, 2k + 1\}$ and C depends upon the smoothness of the solution.

In Fig. 3 and Table 2, we can see how these two methods of obtaining a derivative approximation compare by considering a variable coefficient equation taken from [16]. If we take the derivative of the SIAC filtered approximation, we can still obtain

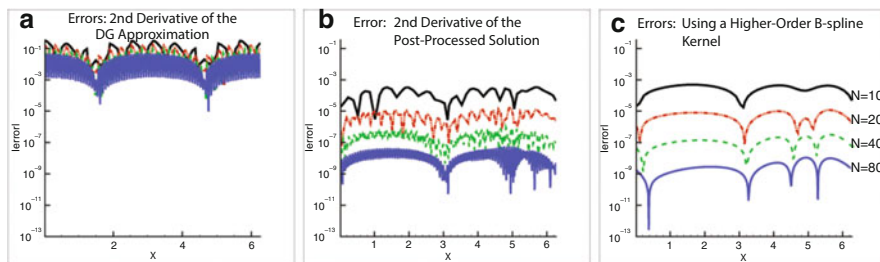


Fig. 3 Pointwise errors in log scale for the second derivatives for the DG approximation together with the SIAC filtered solutions. (a) $\partial^2(u - u_h)$. (b) $\partial^2(u - K_h * u_h)$. (c) $\partial^2(u - \tilde{K}_h * u_h)$

Table 2 L^2 -errors and orders the first and second derivatives for the DG approximation together with the SIAC filtered solutions

\mathbb{P}^2						
N	$\partial_x^\alpha u_h$		$\partial_x^\alpha (K * u_h)$		$\tilde{K} * \partial_h^\alpha u_h$	
	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

$2k + 1$ order accuracy for the first derivative, but each successive derivative loses an order. However, if we use smooth B-splines of higher order, we can maintain $2k + 1$ order accuracy for higher derivatives as well.

6 Filtering Near a Boundary

The next question that would be useful to answer is how to filter near a boundary or discontinuity. This requires modifying the filter [17, 21]. To do so, we first use B-splines that depend continuously on the evaluation point through the shift function $\lambda(\bar{x})$:

$$x_\gamma = -\frac{r}{2} + \gamma + \lambda(x), \quad \lambda(x) = \begin{cases} \min \left\{ 0, -\frac{r+\ell}{2} + \frac{\bar{x}-x_L-\frac{\epsilon h}{2}}{h} \right\}, & x \in [x_L, \frac{x_L+x_R}{2}], \\ \max \left\{ 0, \frac{r+\ell}{2} + \frac{\bar{x}-x_R+\frac{\epsilon h}{2}}{h} \right\}, & x \in (\frac{x_L+x_R}{2}, x_R], \end{cases} \quad (17)$$

where the one-dimensional domain is defined as $\Omega = [x_L, x_R]$.

The accuracy is improved by using extra B-splines near a boundary so that the post-processed solution is

$$u_h^*(\bar{x}) = \underbrace{\theta(\bar{x}) \underbrace{u_{h,2k+1}^*(\bar{x})}_{\text{filtering with } 2k+1 \text{ B-splines}} + (1 - \theta(\bar{x})) \underbrace{u_{h,4k+1}^*(\bar{x})}_{\text{filtering with } 4k+1 \text{ B-splines}}}_{\text{smooth convex combination}}. \quad (18)$$

In this example, θ is chosen such that $\theta(\bar{x}) = 1$ in the interior (giving the symmetric filter); $\theta(\bar{x}) = 0$ near the boundary (to obtain extra accuracy from extra B-splines); θ is smooth in the transition regions between symmetric and boundary filtering.

As an example, we consider the linear equation $u_t + u_x = 0$ with Dirichlet boundary conditions. Plots of the errors are given in Fig. 4 and errors are given in Table 3. We can see from these that we have an improved convergence rate as well as reduction in errors. This occurs even near the boundary and for non-periodic boundary conditions.

Adapting the kernel to handle filtering near boundaries allows us to obtain the following L^∞ -error estimate:

Theorem 2 (Ji et al. [9]) *Let u_h be a DG approximation to an exact solution u for a linear hyperbolic equation. Construct u_h^* by applying the position-dependent SIAC filter to u_h , $k \geq 1$. Then,*

$$\|u - u_h^*\|_{\infty, \Omega} \leq C \|u_0\|_{2k+3+[d/2], \Omega} h^s,$$

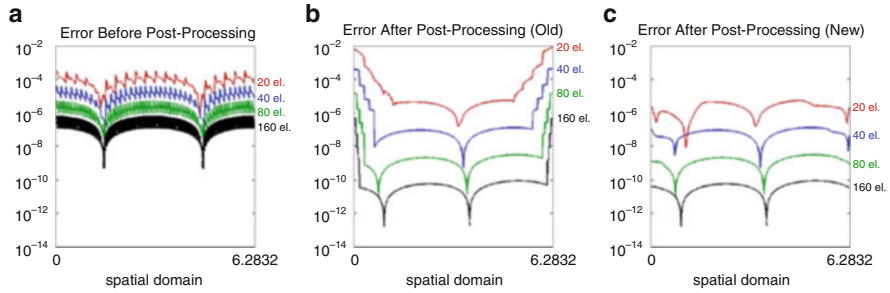


Fig. 4 Pointwise errors in log scale for before and after post-processing for a linear hyperbolic equation with Dirichlet boundary conditions. A comparison of using $2k + 1$ (middle) versus $4k + 1$ (right) central B-splines when near a boundary. (a) DG errors. (b) SIAC DG ($2k + 1$). (c) SIAC DG ($4k + 1$)

Table 3 L^2 -errors and order for before and after post-processing for a linear hyperbolic equation with Dirichlet boundary conditions. A comparison of using $2k + 1$ (middle column) versus $4k + 1$ (right column) central B-splines when near a boundary

Mesh	Before		After ($2k + 1$)		After ($4k + 1$)	
	L^2 -error	Order	L^2 -error	Order	L^2 -error	Order
\mathbb{P}^2						
20	2.681e−04	–	4.003e−03	–	6.984e−06	–
40	3.352e−05	3.00	2.108e−04	4.25	1.850e−07	5.24
80	4.190e−06	3.00	5.464e−06	5.27	4.798e−09	5.27
160	5.238e−07	3.00	1.254e−07	5.45	1.498e−10	5.00
\mathbb{P}^3						
20	5.176e−06	–	1.304e−04	–	3.751e−07	–
40	3.236e−07	4.00	4.712e−06	4.79	6.396e−10	9.20
80	2.023e−08	4.00	3.406e−08	7.11	2.867e−12	7.80
160	1.264e−09	4.00	1.999e−10	7.41	3.079e−14	6.54

and

$$\|u - u_h^*\|_{0,\Omega} \leq C \|u_0\|_{2k+2,\Omega} h^{2k+1},$$

where $s = \min\{2k + 1, 2k + 2 - \frac{d}{2}\}$ and C is a constant, dependent on the L^1 -norm of the kernel coefficients but independent of the mesh.

However, there are still limitations to overcome. For example, using extra B-splines at the boundaries is good for lower-order approximations, but not for higher-order approximations due to the excessive support size and increased condition number of the matrices involved. Further, the added support does not aid in creating a better approximation for non-uniform meshes.

7 Mesh Geometry

Until now, the assumptions on the applicability of the SIAC filter have required a uniform mesh. A logical question to then ask is whether it can work for nonuniform meshes. The challenges that are incurred when attempting to extend the SIAC filter to a nonuniform mesh is that it requires $\mathcal{O}(h^{2k+1})$ convergence in the negative-order norm for both the approximation as well as the divided difference of the approximation. This requires defining a suitable dual equation and a DG scheme for the divided differences. If the mesh is translation invariant, it is easy to show appropriate convergence for the divided differences [10]. However, let us investigate further the actual requirements of the scaling parameter.

Recall that our error estimate is $\|u - K_H * u_h\|_{\Omega} \leq CH^{2k+1}$, where H is the kernel scaling parameter. The translation invariance property requires that $T_H^\ell v(x) = v(x + H\ell)$. Thus the mesh is translation invariant for a scaling of mH , $m \in \mathbb{Z}$ as well. This is illustrated in Fig. 5. In this figure, a kernel scaling of $H = mh$ is used for the convolution kernel in the SIAC filter for a discontinuous Galerkin approximation over a uniform mesh designated by h . We can see that error reduction actually occurs even when $H < h$. Superconvergent order starts to occur around $H = h$ and errors start to increase for $H > h$. The sweetspot of reduced errors and superconvergence seems to occur around $H = h$.

Although the typical meshes tested involve some type of translation invariance, the SIAC filter has also been tested over unstructured triangular meshes with promising results [10, 12]. For example Fig. 6 shows the difference in the pointwise errors for the DG approximation versus the SIAC filtered DG approximation. The

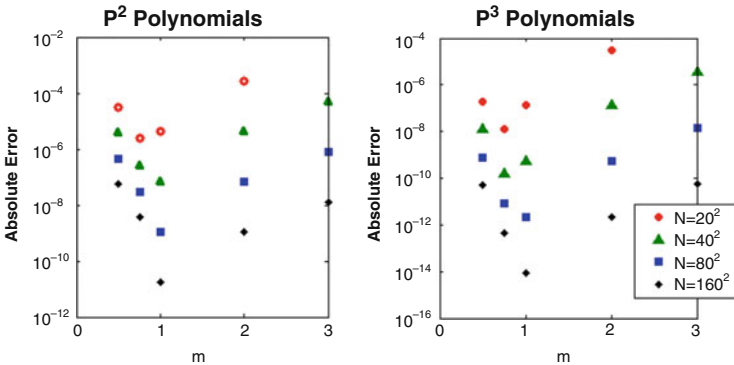


Fig. 5 Effect of H scaling for \mathbb{P}^2 and \mathbb{P}^3 polynomials for a uniform mesh

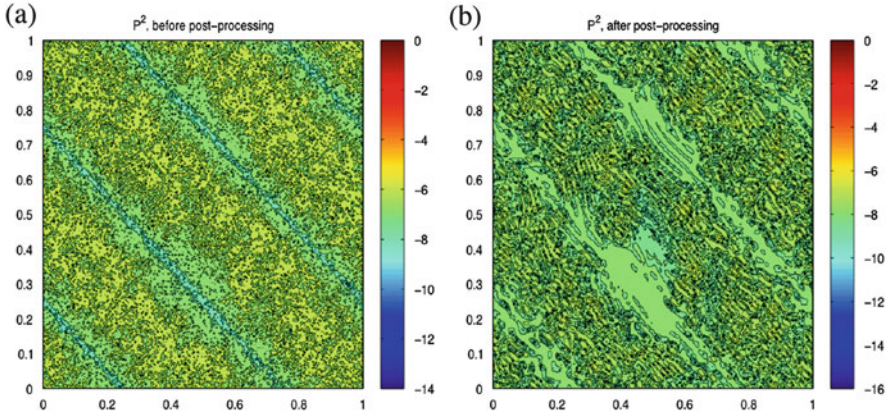


Fig. 6 Typical pointwise error plots for SIAC filtering over a Delaunay Mesh with element splitting. **(a)** DG errors. **(b)** SIAC filtered errors

Table 4 Typical errors for SIAC filtering over a Delaunay Mesh with element splitting

Mesh	$m = 0.5$		$m = 1$		$m = 2$	
	L^2 -error	Order	L^2 -error	Order	L^2 -error	Order
\mathbb{P}^2						
776	7.08E−05	–	1.25E−04	–	x	–
3104	7.84E−06	3.17	6.45E−06	4.27	x	–
12,416	8.24E−07	3.25	5.02E−07	3.68	1.98E−06	–
49,664	1.09E−07	2.20	5.97E−08	3.07	8.11E−08	4.60
\mathbb{P}^3						
776	9.88E−07	–	8.52E−06	–	x	–
3104	2.71E−08	5.18	1.30E−07	6.03	x	–
12,416	3.28E−09	6.02	1.99E−09	6.02	4.58E−08	–
49,664	2.34E−10	3.80	5.85E−11	5.08	6.20E−10	6.20

L^2 -errors are given in Table 4. Figure 7 displays the effect of different scalings, when h is taken to be the longest element edge and the kernel is scaled by $H = mh$. Clearly, one can achieve error reduction.

With SIAC filtering we can usually improve the DG convergence rate from order $k + 1$ to order $2k + 1$ but we have to be careful with kernel scaling [10]. Table 5 gives a list of some of the meshes that SIAC filtering has been tested over and whether reduced errors, improved order or increased smoothness occurs.

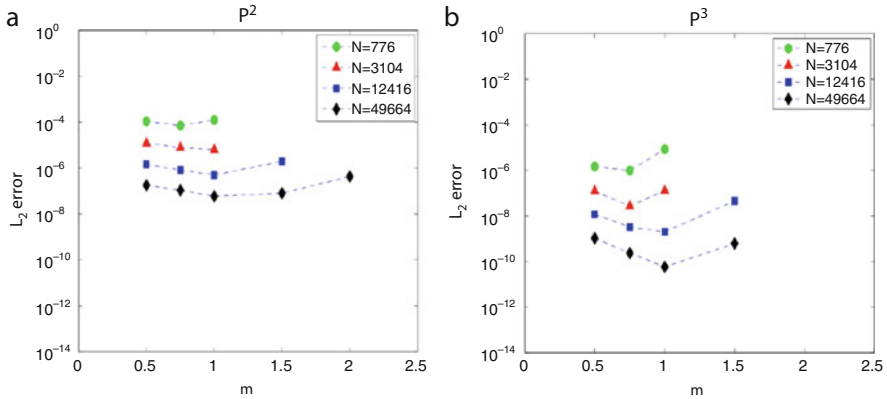


Fig. 7 The effect of the kernel scaling for SIAC filtering over a Delaunay Mesh with element splitting. (a) \mathbb{P}^2 -polynomials. (b) \mathbb{P}^3 -polynomials

Table 5 Some typical meshes over which SIAC filtering has been tested. Listed is the mesh type along with whether SIAC filtering will aid in error reduction, improved convergence order or increasing the smoothness of the solution

Mesh type	Reduced errors	Improved order	Increased smoothness
Uniform quadrilateral	✓	✓	✓
Variable cross quadrilateral	✓	✓	✓
Uniform structured triangle	✓	✓	✓
Structured variable triangle	✓	✓	✓
Delaunay mesh	✓	?	?

8 Summary

We can make superconvergence useful through accuracy extraction techniques. SIAC filtering is one technique that uses a B-spline convolution kernel that induces smoothness on the DG field and enhances accuracy. In general, we can obtain order improvement from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^s)$ where $s = \min\{r + 1, 2k + 1\}$. However, the expected order improvement relies on higher-order estimates in the negative-order norm for the approximation as well as the divided differences. Once we are able to prove these estimates we can concentrate on other issues in SIAC filtering such as modifying the filter for higher-order derivative information or boundary filtering. For the scaling of the kernel, we must exploit information about the mesh geometry in order to have a reduction in the errors.

Acknowledgements Portions of this research are sponsored by the European Office of Aerospace Research and Development (EOARD) under the U.S. Air Force Office of Scientific Research (AFOSR) under grant number FA 8655-13-1-3017.

References

1. S. Adjerid, K.D. Devine, J.E. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Comput. Methods Appl. Mech. Eng.* **191**, 1097–1112 (2002)
2. J.H. Bramble, A.H. Schatz, Higher order local accuracy by averaging in the finite element method. *Math. Comput.* **31**, 94–111 (1977)
3. F. Celiker, B. Cockburn, H.K. Stolarski, Locking-free optimal discontinuous Galerkin methods for Timoshenko Beams. *SIAM J. Numer. Anal.* **44**, 2297–2325 (2006)
4. B. Cockburn, R. Ichikawa, Adjoint recovery of superconvergent linear functionals from Galerkin approximations: the one-dimensional case. *J. Sci. Comput.* **232**, 201–232 (2007)
5. B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, *Advanced Numerical Approximation of Non-linear Hyperbolic Equations*. Lecture Notes in Mathematics, vol. 1697 (Springer, Heidelberg, 1998)
6. B. Cockburn, M. Luskin, C.-W. Shu, E. Süli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Math. Comput.* **72**, 577–606 (2003)
7. L. Ji, Y. Xu, J.K. Ryan, Accuracy enhancement of the linear convection-diffusion equation in multiple dimensions. *Math. Comput.* **81**, 1929–1950 (2012)
8. L. Ji, Y. Xu, J.K. Ryan, Negative-order norm estimates for nonlinear hyperbolic conservation laws. *J. Sci. Comput.* **54**, 269–310 (2013)
9. L. Ji, P. van Slingerland, J.K. Ryan, C.W. Vuik, Superconvergent error estimates for a position-dependent smoothness-increasing accuracy-conserving filter for DG solutions. *Math. Comput.* **83**, 2239–2262 (2014)
10. J. King, H. Mirzaee, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) filtering for discontinuous Galerkin solutions: improved errors versus higher-order accuracy. *J. Sci. Comput.* **53**, 129–149 (2012)
11. H. Mirzaee, L. Ji, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) post-processing for discontinuous Galerkin solutions over structured triangular meshes. *SIAM J. Numer. Anal.* **49**, 1899–1920 (2011)
12. H. Mirzaee, J. King, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) filters for discontinuous Galerkin solutions over unstructured triangular meshes. *SIAM J. Sci. Comput.* **35**, A212–A230 (2013)
13. M. Mirzargar, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) filtering and quasi-interpolation: a unified view. *J. Sci. Comput.* 1–25 (2015). doi:[10.1007/s10915-015-0081-9](https://doi.org/10.1007/s10915-015-0081-9)
14. M.S. Mock, P.D. Lax, The computation of discontinuous solutions of linear hyperbolic equations. *Commun. Pure Appl. Math.* **31**, 423–430 (1978)
15. K. Mustapha, J.K. Ryan, Post-processing discontinuous Galerkin solutions to Volterra integro-differential equations: analysis and simulations. *J. Comput. Appl. Math.* **253**, 89–103 (2013)
16. J.K. Ryan, B. Cockburn, Local derivative post-processing for the discontinuous Galerkin method. *J. Comput. Phys.* **228**, 8642–8664 (2009)
17. J.K. Ryan, C.-W. Shu, One-sided post-processing for the discontinuous Galerkin method. *Methods Appl. Anal.* **10**, 295–307 (2003)
18. J.K. 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**, 821–843 (2005)
19. M. Steffan, S. Curtis, R.M. Kirby, J.K. Ryan, Investigation of smoothness enhancing accuracy-conserving filters for improving streamline integration through discontinuous fields. *IEEE Trans. Vis. Comput. Graph.* **14**, 680–692 (2008)

20. V. Thomée, High order local approximations to derivatives in the finite element method. *Math. Comput.* **31**, 652–660 (1977)
21. P. van Slingerland, J.K. Ryan, C. Vuik, Position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering for accuracy for improving discontinuous Galerkin solutions. *SIAM J. Sci. Comput.* **33**, 802–825 (2011)
22. L.B. Wahlbin, *Superconvergence in Galerkin Finite Element Methods* (Springer, Heidelberg, 1995)
23. G. Wasserman, R. Archibald, A. Gelb, Image reconstruction from fourier data using sparsity of edges polynomial annihilation sparsifying transform. *J. Sci. Comput.* (2014)

Spectral and High Order Methods for Partial Differential
Equations ICOSAHOM 2014

Selected papers from the ICOSAHOM conference, June
23-27, 2014, Salt Lake City, Utah, USA

Kirby, R.M.; Berzins, M.; Hesthaven, J.S. (Eds.)

2015, XI, 530 p., Hardcover

ISBN: 978-3-319-19799-9