J Sci Comput (2016) 67:237–261
DOI 10.1007/s10915-015-0081-9

Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering and Quasi-Interpolation: A Unified View

Mahsa Mirzargar[1] · **Jennifer K. Ryan**[2] · **Robert M. Kirby**[3]

Received: 10 March 2015 / Revised: 3 July 2015 / Accepted: 26 July 2015 / Published online: 9 August 2015
© Springer Science+Business Media New York 2015

Abstract Filtering plays a crucial role in postprocessing and analyzing data in scientific and engineering applications. Various application-specific filtering schemes have been proposed based on particular design criteria. In this paper, we focus on establishing the theoretical connection between quasi-interpolation and a class of kernels (based on B-splines) that are specifically designed for the postprocessing of the discontinuous Galerkin (DG) method called smoothness-increasing accuracy-conserving (SIAC) filtering. SIAC filtering, as the name suggests, aims to increase the smoothness of the DG approximation while conserving the inherent accuracy of the DG solution (superconvergence). Superconvergence properties of SIAC filtering has been studied in the literature. In this paper, we present the theoretical results that establish the connection between SIAC filtering to long-standing concepts in approximation theory such as quasi-interpolation and polynomial reproduction. This connection bridges the gap between the two related disciplines and provides a decisive advancement in designing new filters and mathematical analysis of their properties. In particular, we derive a closed formulation for convolution of SIAC kernels with polynomials. We also compare and contrast cardinal spline functions as an example of filters designed for image processing applications with SIAC filters of the same order, and study their properties.

✉ Mahsa Mirzargar
mirzargar@sci.utah.edu

Jennifer K. Ryan
jennifer.ryan@uea.ac.uk

Robert M. Kirby
kirby@cs.utah.edu

1 Scientific Computing and Imaging (SCI) Institute, University of Utah, Salt Lake City, UT 84112, USA

2 School of Mathematics, University of East Anglia, Norwich, UK

3 School of Computing and the Scientific Computing and Imaging (SCI) Institute, University of Utah, Salt Lake City, UT 84112, USA

Keywords B-splines · Smoothness-increasing accuracy-conserving (SIAC) filtering · Quasi-interpolation · Approximation theory · Fourier analysis

1 Introduction

The SIAC kernel has mostly been developed for discontinuous Galerkin (DG) methods. DG is a widely-used high-order numerical technique to solve (partial) differential equations over complex computational domains using high approximation order [1]. DG methods provide piecewise-continuous approximations to the solution of a (partial) differential equation over a domain (i.e., a collection of elements) and control the fluxes between elements using a weak constraint without enforcing a continuity requirement. The DG solution over each element is represented in terms of (piecewise) polynomials up to degree k where $k+1$ is the order of accuracy of the DG approximation while the weak continuity between elements is controlled through the flux constraints. However, the inter-element discontinuities can be problematic. For instance, the inter-element discontinuity can pose challenges for applications such as feature extraction and visualization. A postprocessing stage to improve the inter-element continuity of the DG solution is therefore desirable. However, special care must be taken not to deteriorate the order of the accuracy of the original DG solution inside the elements. The class of SIAC postprocessing techniques proposed in [2] can be used to raise the continuity degree of the DG approximation while preserving and extracting the superconvergence of the original DG approximation. It uses compactly-supported convolution kernels based on a linear combination of B-splines. While the *smoothness increasing* property of SIAC filtering is a direct result of using B-spline filtering, the *accuracy preserving* property (i.e., superconvergence) of the SIAC filtering is attained through choosing a specific number of B-splines and imposing a polynomial reproduction constraint.

The approximation properties of smoothness-increasing accuracy-conserving (SIAC) kernel as a filtering technique to generate smooth approximations have received attention in simulation science [2–6]. A SIAC filter has the ability to extract a superconvergent solution from a DG approximation for different element types including quadrilateral, structured triangular, tetrahedral and even unstructured triangular meshes [7–9]. One-sided SIAC kernels have been proposed as an extension of this convolution-based postprocessing for simulations involving boundaries or sharp discontinuities such as shocks [5,6,10]. However, the approximation properties of SIAC filtering *in relation to* spline approximation as widely studied in the approximation theory literature have not received much attention. In this paper, we study these classes of kernels in a more general setting from both the approximation theory perspective where spline spaces are heavily studied as well as from an application point of view where kernels are used for generating smooth approximations with some desirable properties. Designing B-spline-based filtering schemes is not a new topic. However, for the first time, we establish the theoretical connection between construction of the SIAC kernel and quasi interpolation. We provide a unified view that enhances the mathematical analysis tools for designing and analyzing general kernels using central B-splines with desirable approximation properties. One of the direct impacts of such analysis is deriving a closed formulation of the convolution of SIAC kernels (and more generic B-spline-based kernels) with polynomials that leads to a *direct* and *exact* computation of the kernel coefficients that was not known previously. Moreover, we introduce a systematic way of constructing variations of the symmetric SIAC kernel that still attain superconvergence properties while having different computational costs and smoothness properties by changing the number and/or the order of B-splines used. In light of this generalization, we also extend the theoretical super-

convergence results concerning the error analysis for SIAC kernels in relation to the number and the order of B-splines used to construct such variations. In addition, we demonstrate that the symmetric SIAC kernel in this context can be considered as a member of a *family* of filtering kernels. Thorough study of the entire family of filtering kernels of which the symmetric SIAC kernels are specific members is beyond the scope of the current work and hence, we leave it as an interesting future work direction. From the application point of view, our results can help practitioners design new kernels of interest with different design criteria. For the purpose of the current work, we only focus on postprocessing of the DG approximation to a linear hyperbolic equation on a uniform quadrilateral mesh using the symmetric SIAC kernel and the one-sided SIAC kernel introduced in [5,10] whose superconvergence properties have been proven in [2–4].

The rest of the paper proceeds as follows. After an introduction to the notation used, we provide a brief review of the spline approximation in Sect. 2.2. The reader conversant with variations of spline approximation such as polynomial spline interpolation and quasi interpolation can move directly to Sect. 3. In Sect. 3 we introduce a generic class of compactly-supported filters based on a finite linear combination of B-splines. Using theoretical results concerning B-spline convolution over polynomial spaces, we demonstrate how filtering kernels can be designed with polynomial reproduction properties from this generic class. Section 4 is devoted to the introduction of SIAC kernels and revisiting its approximation properties in light of analysis techniques discussed in Sect. 3. In Sect. 5, we provide some numerical results and error contour plots to demonstrate and validate the theoretical results presented. Finally, we present our conclusions in Sect. 6.

2 Background

2.1 Notation

We start by introducing the notation used in the remainder of the paper. As the aim of this paper is to study convolution kernels, it is necessary to introduce different convolution operators of interest—continuous convolution, discrete convolution and semi-discrete convolution.

Definition 1 (*Continuous convolution*) The continuous convolution of two continuous functions, f and g is defined as

$$(f * g)(x) := \int_{\Omega} f(\tau) g(x - \tau) d\tau, \tag{1}$$

where Ω represents the domain over which the convolution is computed.

Equivalently, one can write the continuous convolution operator in terms of an inner product as

$$(f * g)(x) := \langle f(\tau), g(x - \tau) \rangle. \tag{2}$$

Definition 2 (*Discrete convolution*) The discrete convolution of two functions, $f[k]$ and $g[k]$ defined over a (sub)set of $\mathbb{Z}$ is defined as

$$(f *' g)[k] := \sum_{m \in \mathbb{Z}} f[m] g[k - m]. \tag{3}$$

Definition 3 (*Semi-discrete convolution*) Considering φ to denote a compactly supported continuous function and f to denote a function (at least) defined on $\mathbb{Z}$, then the 1D semi-discrete convolution can be defined as

$$(\varphi *_{|} f)(\cdot) := \sum_{m\in\mathbb{Z}} f(m)\varphi(\cdot - m). \tag{4}$$

It is worth noting that in comparison to continuous convolution that commutes with translation, semi-discrete convolution only commutes with integer translations. Approximation of polynomials using B-splines plays a major role in derivation of our theoretical results. We use $\mathbb{P}^n$ to denote the polynomial space containing polynomials up to degree n and use normalized monomials as a convenient basis for such polynomial spaces.

Definition 4 (*Nomalized monomial*) The normalized monomial of degree p is defined as

$$[\![x]\!]^p := \frac{x^p}{p!}. \tag{5}$$

We wish to emphasize that in the DG literature the bracket notation is generally reserved for indicating an inter-element jump [1]. Here, unless otherwise noted, we use the bracket notation to indicate a normalized monomial.

Where appropriate, we use ∂^α to denote the central difference operator of order α. To aid the discussion of B-spline convolution in Sect. 3, we use $\hat{f}(\cdot)$ to denote the Fourier transform of an integrable function $f(\cdot)$ defined as: $\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)e^{-ix\omega}dx$. We will use $\delta(\cdot)$ to denote the Dirac delta function(al) and use $\delta^{(p)}(\cdot)$ to denote its pth order functional derivative.

The superconvergence properties of the SIAC kernel is studied in terms of the norm of the postprocessing error. We use $\|\cdot\|_{0,\Omega}$ to represent the usual L^2-norm over Ω and $\|\cdot\|_{-\ell,\Omega}$ to denote the negative-order norm, where Ω represents a bounded open set $\Omega \in \mathbb{R}^d$. The negative-order norm is defined as in [11] in terms of positive norms as

$$\|u\|_{-\ell,\Omega} = \sup_{\phi\in\mathcal{C}_0^\infty(\Omega)} \frac{\int_\Omega u(x)\phi(x)dx}{\|\phi\|_{\ell,\Omega}}, \tag{6}$$

where $\mathcal{C}_0^\infty(\Omega)$ denotes the space of infinitely differentiable functions with compact support on Ω.[1] The negative order norm as Cockburn noted [2] can be used to quantify the oscillatory nature of a function. Negative order norm is often used to prove superconvergence property of SIAC filtering via the following relation it has with L^2 norm [11, Lemma 4.2]

$$\|u\|_{0,\Omega} \leq C \sum_{|\alpha|\leq\ell} \|D^\alpha u\|_{-\ell,\Omega}, \tag{7}$$

where D^α is used to denote the differentiation operator of degree α. Since the focus of this paper is on linear hyperbolic equations, we use $u(\cdot)$ to denote the true solution to a linear hyperbolic equation and $u_h(\cdot)$ to denote its DG approximation of order $k+1$.

2.2 Review of B-Splines and Spline Approximation

The maximal approximation order of B-splines along with their minimal support made B-spline-based approximation techniques popular in a variety of applications, including signal processing, biomedical imaging, finite element methods and superconvergence-extraction

[1] The negative order norm $\|\cdot\|_{-\ell,\Omega}$ is the norm associated with $H^{-\ell}(\Omega)$ (i.e., the dual space of the Sobolev space $H^\ell(\Omega)$).

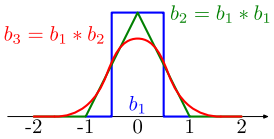

Fig. 1 Self-convolution of B-splines

techniques [2,5,11–18]. In this section, we introduce B-splines in the univariate case, with all the results easily extending to higher-dimensional Cartesian lattices (or uniform quadrilateral meshes) using tensor products.

2.2.1 Introduction to Central B-Splines and Spline Spaces

The first-order univariate central B-spline (Basis splines) is defined as the indicator function over the interval, $T = [\frac{-1}{2}, \frac{1}{2}]$:[2]

$$b_1(x) = \mathcal{X}_T(x) = \begin{cases} 1 & x \in [\frac{-1}{2}, \frac{1}{2}] \\ 0 & \text{otherwise.} \end{cases}. \tag{8}$$

Higher-order central B-splines can be constructed by simply using self-convolution,

$$b_{n+1}(x) = (b_1 * b_n)(x),$$

as shown in Fig. 1. For the rest of the discussion we simply use the term *B-splines* to denote *central B-splines* unless otherwise stated. B-splines define a basis for an approximation space called a spline space. A typical spline space is defined as the spanning space of translations of the basis function (i.e., B-splines), denoted as:

$$\mathbf{S}_n := \text{span}(b_n(\cdot - k))_{k \in \mathbb{Z}}. \tag{9}$$

An arbitrary function can be approximated with an element from the spline space, $s \in \mathbf{S}_n$ by finding the unique set of spline coefficients, c_γ, that best represent that function:

$$s = \sum_{\gamma \in h\mathbb{Z}} c_\gamma b_n(\cdot - \gamma), \tag{10}$$

where c_γ represents the spline coefficient and h denotes the distance between the B-spline centers, $x - \gamma$. The above relation can be translated into a semi-discrete convolution with a B-spline as the kernel and the vector representing the spline coefficients. It has been proven that every piecewise polynomial function of a given degree and smoothness over a domain can be represented by a linear combination (i.e., convolution) of the B-spline of the same degree over the same domain partition [19]. In general, the smoothness and the accuracy of the spline approximation can be controlled by varying the order of the B-spline used.

In addition to having compact support, B-splines provide the maximal approximation order over polynomial spaces. The approximation order of B-splines is defined as the asymptotic behavior of the approximation error as the sampling distance h is refined [20]

$$\text{dist}(f - s_h) = O(h^n), \tag{11}$$

where $s_h \in \mathbf{S}_n$ denotes the spline approximation of any function f in the corresponding Sobolev space and the distance in here is measured in the L_p-norm ($2 \leq p \leq \infty$). Various

[2] The first-order central B-spline is often denoted as $b_0(x)$, but herein the authors chose to follow the notation used in the previously published definition of SIAC kernels throughout the article.

Table 1 Approximation properties of spline space formed by d-dimensional B-spline of order $n+1$

Spline space	Degree	Approx. order	Continuity
$\mathbf{S}_{n+1}$	nd	$n+1$	C^{n-1}

properties of the spline space formed by shifts of a d-dimensional B-spline of order $n+1$ have been summarized in Table 1. Note that a spline space in this context is formed by shifts of a single B-spline. In the next section, we study how kernels can be constructed using a linear combination of multiple B-splines.

The spline space formed by B-splines of order $n+1$ contains polynomials up to degree n [21,22]. Therefore, the approximation order of the spline space can also be studied in terms of its lower-order polynomial space reproduction property [23]. The choice of spline coefficients plays a major role in attaining the best spline approximation when approximating an arbitrary function in the spline space. A *proper* choice of spline coefficients ensures the exact reproduction of polynomials up to the degree that spline space can afford. An improper choice of spline coefficients (such as using the discrete function values as the spline coefficients) can result in degradation of the approximation error from the best that can be offered by spline approximation (often referred to as over-smoothing artifact) [24, Proposition 2.10]. Computation of proper spline coefficients for spline approximation has been studied in the approximation theory literature through a pre-filtering scheme called quasi-interpolation [21, Chapter III].

2.3 Introduction to Quasi-Interpolation

Quasi-interpolation provides an elegant formulation to convert discrete function values of f into spline coefficients in order to provide its best representation (i.e., approximation) in a spline space. Considering a sufficiently smooth function $f(x)$ and its representation in terms of a Taylor series expansion,

$$f(x) = f(0) + Df(0)x + \cdots + D^n f(0)x^n + O(x^{n+1}), \tag{12}$$

a spline-based quasi-interpolation of $f(x)$ using a B-spline of order $(n+1)$ ensures the reproduction of the first $n+1$ terms of its Taylor series expansion which translate into the best approximation of $f(x)$ in $\mathbb{P}^n$. Finding spline coefficients using quasi-interpolation is often carried out by designing a linear functional λ to write the spline approximation of f as [21, III.13]

$$Q_\lambda f(x) = \sum_{\gamma\in\mathbb{Z}} \underbrace{\lambda f(\cdot+\gamma)}_{c_\gamma} b_{n+1}(x-\gamma), \tag{13}$$

where $Q_\lambda f(x)$ represents the quasi-interpolant of $f(x)$ (or spline approximation of $f(x)$ with quasi-interpolation) and c_γ denotes the spline coefficients in Eq. 10. In cases where f is a low-order polynomial, the spline-based quasi-interpolation *exactly* reproduces f, that is $Q_\lambda f(x) = f(x)$ for $f(x) \in \mathbb{P}^n$. Polynomial reproduction guarantees optimal asymptotic behavior of the approximation error using B-splines [21,25] and their higher dimensional counterparts [26,27]. Construction of such a linear functional for quasi-interpolation lends itself to the Fourier analysis of B-splines' (semi-discrete) convolution with polynomials.

The following lemma summarizes the results concerning the B-spline mapping in polynomial spaces that plays the main role in derivation of λ (and quasi-interpolation).

Lemma 1 *Denote the polynomial space consisting of polynomials of degree less than or equal to n as* $\mathbb{P}^n$. *The (semi-discrete) convolution of a polynomial from this space with a B-spline of order* $n+1$ *provides a one-to-one and onto mapping, and is therefore invertible.*

Interested readers should consult [21, Proposition 6] for the proof and the discussion of this result in a general setting. We only provide a summary of this result to study the mapping of polynomials under (univariate) B-spline convolution while the results can easily be extended to higher dimensions using tensor products. The semi-discrete convolution of a normalized monomial of degree $p \leq n$ with B-splines of order $n+1$ results in a *polynomial* of the same order. However, the resultant polynomial is often not equal to $[\![x]\!]^p$ unless $n = 0, 1$. For instance, the following summarizes the results of the convolution of quadratic B-spline with normalized monomials (up to degree 2)

$$\begin{aligned} b_3(x) *_{|} [\![x]\!]^0 &= 1, \\ b_3(x) *_{|} [\![x]\!]^1 &= x, \\ b_3(x) *_{|} [\![x]\!]^2 &= [\![x]\!]^2 + \frac{1}{8}. \end{aligned} \tag{14}$$

While, this example demonstrates that convolution of B-spline of order $n+1$ with $[\![x]\!]^p$ where $p \leq n$ does not result in the reproduction of the original function, from Lemma 1, it is easy to conclude that for $p \leq n$ there exists a polynomial $g_p(x)$ such that

$$b_{n+1}(x) *_{|} g_p(x) = [\![x]\!]^p, \qquad g_p(x) = \sum_{\gamma \leq p} \beta_\gamma [\![x]\!]^{p-\gamma}, \quad p \leq n. \tag{15}$$

Therefore, in order to reproduce $[\![x]\!]^p$ using a B-spline of order $n+1$ where $p \leq n$, one needs to use discrete values of $g_p(x)$ instead of discrete values of $[\![x]\!]^p$. $g_p(x)$ is not an arbitrary polynomial (see Lemma 1) and can be uniquely specified (for any order) in terms of the coefficients β_γ efficiently. The following lemma fully specifies $g_p(x)$ in this context.

Lemma 2 *The polynomial coefficients* β_γ *in Eq.* (15) *are fully specified by the Fourier transform of the B-spline used in* (15) *as* [21, III. 34],

$$\beta_\gamma = D^\gamma \frac{i^\gamma}{\hat{b}_{n+1}(\omega)}\bigg|_{\omega=0}, \tag{16}$$

where i *denotes the unit imaginary number.*

Using Lemma 2, β_γ can be computed analytically. The interested reader can consult [21, Chapter III] for the proof. The polynomial $g_p(x)$ along with Taylor series expansion can then be used to define the linear functional, λ, to construct the spline coefficients from discrete values of an arbitrary function as [21, III.13]

$$\lambda : \ f \mapsto \sum_{\gamma \leq n} g_\gamma(0)(D^\gamma f)(0). \tag{17}$$

The optimal approximation power and smoothness of the spline-approximants are among the main reasons that make B-spline approximation attractive from a practical point of view. As shown in Table 1, using higher-order B-splines as approximation kernels results in a smoother approximation of the underlying function, while the exact interpolation of the sampling points is only satisfied when lower-order B-splines like b_1 or b_2 are used (even with deployment of quasi-interpolation). The exact interpolation of discrete function values using higher-order B-splines (using semi-discrete convolution) has been studied as a cardinal spline

interpolation problem [28,29]. In what follows, we briefly discuss the cardinal spline interpolation problem. Cardinal spline (filters) are examples of *globally-supported* interpolatory kernels constructed based on a linear combination of B-splines to solve spline interpolation problem.

2.4 Cardinal Spline Interpolation

The cardinal spline interpolation problem can be defined as follows: Consider a discrete sequence of function values, $\{f(\gamma)\}$, that are equally spaced, where $f(x)$ is a continuous piecewise polynomial function of degree n with continuous derivatives up to order $n-1$. Find an spline interpolant of this function in the spline space formed by translates of the $(n+1)$th order B-spline. Without any loss of generality, we assume a unit length distance between the discrete function values (and the B-spline translations). This problem is equivalent to finding the spline coefficients c_γ that satisfy

$$f(x) \approx s(x) = \sum_{\gamma \in \mathbb{Z}} c_\gamma b_{n+1}(x-\gamma), \qquad s(\gamma) = f(\gamma). \tag{18}$$

The existence and uniqueness of such an interpolant in the spline space formed by translates of $(n+1)$th order B-splines has been discussed in [28]. The spline coefficients c_γ in this relation are obtained by pre-filtering the discrete function values, $\{f(\gamma)\}$, using the direct B-spline filter $\{q^n_{int}(\gamma)\}$ proposed in [29]:

$$s(x) = \sum_{\gamma \in \mathbb{Z}} \underbrace{(f *' q^n_{int})(\gamma)}_{c_\gamma} b_{n+1}(x-\gamma), \tag{19}$$

where $*'$ denotes discrete convolution introduced in Eq. 3. Equivalently, one can express the spline interpolant $s(x)$ in terms of the discrete function values as

$$s(x) = \sum_{\gamma \in \mathbb{Z}} f(\gamma)\eta^{n+1}(x-\gamma), \qquad \eta^{n+1}(x) = \sum_{\gamma \in \mathbb{Z}} q^n_{int}(\gamma) b_{n+1}(x-\gamma), \tag{20}$$

where $\eta^{n+1}(x)$ denotes the *cardinal spline* of order $n+1$ (with a global support). For instance, the cardinal cubic spline can be written as:

$$\eta^4(x) = \frac{-6\alpha}{(1-\alpha^2)} \sum_{\gamma \in \mathbb{Z}} \alpha^{|\gamma|} b_4(x-\gamma), \tag{21}$$

where $\alpha = \sqrt{3}-2$ [29].

It is important to note that the cubic cardinal spline presented attains a value of 1 at 0 (i.e., for the interpolation point) and value 0 on all other integer points (i.e., for the rest of the discrete function values). This property guarantees the exact interpolation of the discrete function values. From a theoretical point of view, the polynomial spline interpolant is computed as an element of the spline space and hence the approximation error is still bounded by the approximation power of the B-spline used. As demonstrated in Fig. 2, the support of the cardinal spline (filter) tends to vanish rapidly. However, based on Eq. 20 cardinal splines have *global* supports.

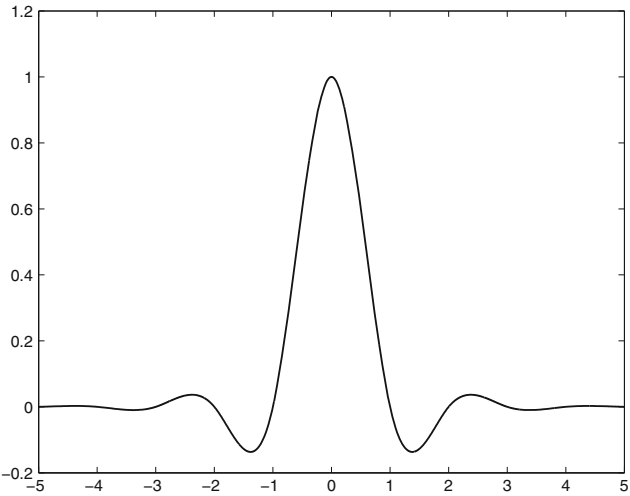

Fig. 2 Cardinal cubic spline function

3 The B-Spline Based Kernels with Compact Support

In the previous section, we briefly reviewed approximation in spline spaces using a semi-discrete convolution framework. In this section, we focus on studying B-spline-based kernels that are often used in practice for *filtering* and generating smooth functions using continuous convolution and study the theoretical connection between the two. An example of applications that widely use B-spline-based filters is postprocessing of the DG approximation, which will be covered in detail in Sect. 4.

Kernels with global support (e.g., cardinal spline) are computationally unattractive. Therefore, in this section, we consider a general approach to define kernels with compact support based on the linear combination of $r+1$ central B-splines of order $n+1$ as

$$K^{r+1,n+1}(x) = \sum_{\gamma=0}^{r} c_\gamma b_{n+1}(x - x_\gamma). \tag{22}$$

For notational convenience, we express the kernel as $K^{r+1,n+1}$ throughout the article. It is hoped that these kernels can balance computational cost with desirable approximation properties. The design choices that should be considered to construct such a kernel are the number of B-splines, the order of the B-splines used, the kernel coefficients, c_γ, and the B-spline centers, x_γ.

Note that the extent of the support of the kernel is specified based on the order, the number of B-splines used and the B-spline centers. Assuming $x_0 < x_1 < \cdots < x_r$ the support of the kernel $K^{r+1,n+1}$ is: $[x_0 - \frac{n+1}{2}, x_r + \frac{n+1}{2}]$ where $n+1$ is the support of b_{n+1}. Kernels constructed through this approach have C^{n-1} continuity. A common design criterion for choosing the kernel coefficients is to impose constraints on the coefficients regarding the behavior of the approximation error. For instance, one can choose to control the behavior of the kernel in a specific approximation space (e.g., spline spaces or polynomial spaces) or to impose exact interpolation of the sampling points to construct a unique kernel.

Since $K^{r+1,n+1}$ is constructed using central B-splines, it is natural to study its properties in the polynomial spaces. As discussed in detail in Sect. 2.3, spline approximation can be better understood as the mapping B-spline (convolution) provides in low-order polynomial spaces in terms of $g_p(x)$ defined in Eq. 15. Consequently, low-order polynomial reproduction is a natural choice for controlling the approximation behavior of $K^{r+1,n+1}$. In order to *fully* benefit from the approximation power of the B-spline used to construct $K^{r+1,n+1}$, the kernel needs to reproduce polynomials up to degree n (i.e., up to the approximation power of the B-spline) . However, special care must be taken to enforce this constraint based on the relation between the number of coefficients (i.e., the degrees of freedom), $r+1$ and the B-spline order, $n+1$. For $r+1 \geq n$, the following relation ensures that $K^{r+1,n+1}$ reproduces the polynomials up to degree n,

$$K^{r+1,n+1}(x) * [\![x]\!]^p = [\![x]\!]^p, \qquad 0 \leq p \leq n, \tag{23}$$

where $*$ in this equation denotes a continuous convolution defined in Eq. 1. Designing kernels with a polynomial reproduction property is not a new topic, and the kernel coefficient can be numerically computed by solving the linear system of equations induced by Eq. 23. However, we aim to demonstrate that well-established results in approximation theory provide theoretical means to write the left hand side of Eq. 23 in closed form, which in turn results in finding the kernel coefficients *exactly*. The following lemma demonstrates some results regarding $(n+1)$th order B-spline mapping over polynomial spaces.

Lemma 3 *Convolution of the $(n+1)$th order B-spline for $n \geq 1$ with normalized monomials of degree higher than or equal to n results in a polynomial of the same order of the form:*

$$[\![x]\!]^p * b_{n+1}(x) = G_p(x) = \sum_{k=0}^{p} a_k [\![x]\!]^k, \quad a_k = \frac{i^{p-k}}{(p-k)!} \cdot \left(D^{p-k} \hat{b}_{n+1}(\omega)\right)\Big|_{\omega=0}, \tag{24}$$

where $i = \sqrt{-1}$.

Remark It is known that B-spline (semi-discrete) convolution over lower order polynomial spaces as discussed in Lemma 1 provides a *one-to-one* and *onto* mapping. Note that the B-spline mapping for polynomial spaces whose degree is higher than the B-spline order is only *into* (yet invertible). Moreover, B-spline mapping using continuous convolution and semi-discrete convolution agrees on polynomial spaces [21, Chapter III].

Proof Convolution in the spatial domain corresponds to multiplication in the frequency domain; hence, we can write the convolution of a generic normalized monomial of order p with B-spline of order $n+1$ in the Fourier domain as:

$$[\![x]\!]^p * b_{n+1}(x) \Leftrightarrow \frac{i^p \delta^{(p)}(\omega)}{p!} \cdot \hat{b}_{n+1}(\omega), \tag{25}$$

Using the distributional definition of the delta function, $\delta(\omega)$, we can write the right-hand-side of the above relation as:

$$\frac{i^p \delta^{(p)}(\omega)}{p!} \cdot \hat{b}_{n+1}(\omega) = \frac{i^p}{p!} \left(\hat{b}_{n+1} \cdot \phi\right)^{(p)}(\omega)\Big|_{\omega=0} = \sum_{k=0}^{p} \left(\left(\frac{i^k \phi^{(k)}}{k!} \cdot \frac{i^{p-k} \hat{b}_{n+1}^{(p-k)}}{(p-k)!} \right)(\omega)\Big|_{\omega=0} \right), \tag{26}$$

where $\phi(\omega)$ in this relation represents a test function. Note that $\hat{b}^{(n+1)}$ can be calculated analytically and evaluated at $\omega = 0$. Therefore, the equation above can be rewritten as a

linear combination of derivatives of the delta function whose coefficients only depends on the evaluation values of the derivatives of Fourier transform of $b_{n+1}(x)$ at 0:

$$\begin{aligned} \frac{i^p}{p!}\big(\hat{b}_{n+1}\cdot\phi\big)^{(p)}(\omega)\Big|_{\omega=0} &= \sum_{k=0}^{p}\Big\langle a_k,\ \frac{i^k\phi^{(k)}(\omega)}{k!}\Big|_{\omega=0}\Big\rangle, \\ a_k &= \frac{i^{p-k}}{(p-k)!}\cdot\big(D^{p-k}\hat{b}_{n+1}(\omega)\big)\Big|_{\omega=0}. \end{aligned} \tag{27}$$

The right-hand-side in the inner product corresponds to the Fourier transform of a kth order normalized monomial in the time domain ($\phi(\omega)$ in this relation represents a test function). Note that the Fourier transform of a B-spline is an even function and therefore, for odd values of $p-k$ we have: $\hat{b}_{n+1}^{(p-k)}(0)=0$. The Fourier analysis above shows that the result of the convolution of a higher-order monomial with B-spline can be interpreted as another polynomial of the form:

$$[\![x]\!]^p * b_{n+1}(x) = \sum_{k=0}^{p} a_k [\![x]\!]^k. \tag{28}$$

□

The following Theorem shows how one can use the B-spline mapping in the polynomial space, using Lemma 3 to enforce polynomial reproduction up to the approximation power of $b_{n+1}(x)$:

Theorem 1 *Let $r+1\geq n$, then the following relations guarantee that $K^{r+1,n+1}$ satisfies the lower-order polynomial reproduction property for $\mathbb{P}^n$ as defined in Eq. (23):*

$$\begin{aligned} &(1)\quad \sum_{\gamma=0}^{r} c_\gamma = 1 \qquad \textit{(partition of unity)} \\ &(2)\quad \sum_{\gamma=0}^{r} c_\gamma \sum_{m=0}^{p} [\![-x_\gamma]\!]^m \sum_{k=0}^{p-m} a_{p-k-m}[\![x]\!]^k = [\![x]\!]^p, \quad p=0,\ldots,n, \end{aligned} \tag{29}$$

where a_k is only dependent on the Fourier transform of the constituent B-spline (see Lemma 2).

Remark: The left-hand side of (2) in Eq. 29 denotes a linear combination of lower order monomials up to degree p. Therefore, a_k can be used to specify the relation between the kernel coefficients analytically and define the kernel coefficients *exactly* in terms of rational numbers.

Proof We start with rewriting (23) as

$$\sum_{\gamma=0}^{r} c_\gamma \langle b_{n+1}(x-y-x_\gamma), y^p\rangle = x^p, \qquad p=0,\ldots,n, \tag{30}$$

We can replace the monomials in the relation above with normalized monomials: $[\![x]\!]^p := \frac{x^p}{p!}$ and use the change of variable $z=x-y-x_\gamma$ to write

$$\sum_{\gamma=0}^{r} c_\gamma \langle b_{n+1}(z), [\![x-z-x_\gamma]\!]^p\rangle = [\![x]\!]^p, \qquad p=0,\ldots,n. \tag{31}$$

Using the binomial expansion we have

$$[\![x - z - x_\gamma]\!]^p = \frac{(x - z - x_\gamma)^p}{p!} = \sum_{k=0}^{p} [\![x - z]\!]^{p-k} [\![-x_\gamma]\!]^k. \tag{32}$$

Consequently, Eq. (30) simplifies to

$$\sum_{\gamma=0}^{r} c_\gamma \sum_{m=0}^{p} [\![-x_\gamma]\!]^m \langle b_{n+1}(z), [\![x - z]\!]^{p-m} \rangle = [\![x]\!]^p, \qquad p = 0, 1, \ldots, n. \tag{33}$$

For $p \leq n$, B-splines of order $n+1$ can reproduce polynomials up to degree n and hence, we can write Eq. (33) in terms of $G_p(x)$ (see Lemma 3) as:

$$\sum_{\gamma=0}^{r} c_\gamma \sum_{m=0}^{p} [\![-x_\gamma]\!]^m G_{p-m}(x) = [\![x]\!]^p, \qquad p = 0, \ldots, n. \tag{34}$$

For $p = 0$, we have $G_0(x) = 1$ and hence, we can conclude that the kernel coefficients satisfy:

$$\sum_{\gamma=0}^{r} c_\gamma = 1. \tag{35}$$

This relation further shows that any order of $K^{r+1,n+1}$ introduced in Eq. (23) satisfies a partition of unity property and completes the proof for (1). By plugging in the relation for $G_p(x)$ we have:

$$\sum_{\gamma=0}^{r} c_\gamma \sum_{m=0}^{p} [\![-x_\gamma]\!]^m \sum_{k=0}^{p-m} a_{p-k-m} [\![x]\!]^k = [\![x]\!]^p, \qquad p = 0, \ldots, n. \tag{36}$$

Note that the values of a_k only depend on the B-spline order and not p. Therefore, it suffices to consider $p = n$. □

Depending on the values chosen for n and r, Theorem 1 may not fully specify the kernel coefficients c_γ. However, Eq. (36) will uniquely specify a *family* of kernels that are C^{n-1} continuous and filtering using $K^{r+1,n+1}$ reproduces any function $f(x) \in \mathbb{P}^n$. Moreover, all the kernels in this family satisfy the partition of unity property. In this situation, additional constraints on the behavior of the kernel are required in order to select a unique kernel from this family. We first focus on symmetric kernels and later demonstrate how the results generalize to a specific class of one-sided kernels as well [5,10].

In simulation science, polynomial reproduction is a desirable property, and correspondingly is the property on which postprocessing kernels such as the symmetric SIAC kernels have been constructed [2,11]. Considering $r+1 \geq n$, let us consider polynomial reproduction up to the number of B-splines (i.e., the number of kernel coefficients) as

$$K^{r+1,n+1}(x) * [\![x]\!]^p = [\![x]\!]^p, \qquad p = 0, \ldots, r. \tag{37}$$

This equation uniquely specifies all $r+1$ kernel coefficients in Eq. (22).

Theorem 1 and Lemma 3 can be used to study the polynomial reproduction property of $K^{r+1,n+1}$ in order to uniquely specify the unknown kernel coefficients.

Without loss of generality, for the rest of the discussion in this section we consider r to represent an even number and $x_\gamma = -\frac{r}{2} + \gamma$ which means $K^{r+1,n+1}$ represents a symmetric kernel where $c_m = c_{r-m}$ and hence, the degrees of freedom (i.e., the number of unknown

kernel coefficients) reduces to $\frac{r}{2}+1$. In Sect. 4, we prove how the results can be extended for a class of filtering kernels with different choice for x_γ and r.

Theorem 2 *Let $\frac{r}{2}+1 \geq \lceil \frac{n}{2} \rceil$ and $x_\gamma = -\frac{r}{2}+\gamma$. The kernel coefficients of the symmetric kernel $K^{r+1,n+1}$ with polynomial reproduction property as introduced in Eq. (37) can then be fully specified using the following relations:*

$$\begin{aligned} &(1)\quad \sum_{\gamma=0}^{r} c_\gamma = 1 \qquad \textit{(partition of unity)} \\ &(2)\quad b_{n+1}(x) * [\![x]\!]^r + 2\sum_{\gamma=0}^{r/2-1} c_\gamma \sum_{m=1}^{\lfloor r/2 \rfloor} [\![x_\gamma]\!]^{2m} \left(b_{n+1}(x) * [\![x]\!]^{r-2m} \right) = [\![x]\!]^r. \end{aligned} \tag{38}$$

Proof Following the proof provided for Theorem 1, we can rewrite the polynomial reproduction property of the kernel presented in Eq. (37) in terms of B-spline convolutions as

$$\sum_{\gamma=0}^{r} c_\gamma \sum_{m=0}^{p} [\![-x_\gamma]\!]^m \left(b_{n+1}(x) * [\![x]\!]^{p-m} \right) = [\![x]\!]^p, \qquad p = n, \ldots, r \tag{39}$$

implying

$$\sum_{\gamma=0}^{r} c_\gamma \left(b_{n+1}(x) * [\![x]\!]^p \right) + \sum_{\gamma=0}^{r} c_\gamma \sum_{m=1}^{p} [\![-x_\gamma]\!]^m \left(b_{n+1}(x) * [\![x]\!]^{p-m} \right) = [\![x]\!]^p. \tag{40}$$

Taking $p=0$, (1) is trivial. Note that equivalently one can use Theorem 1 to prove (1). Due to the symmetry of the kernel, the kernel coefficients c_γ and the B-spline centers, x_γ are symmetric around $\frac{r}{2}$, therefore, we can further simplify the relation

$$\begin{aligned} &\sum_{\gamma=0}^{r} c_\gamma \left(b_{n+1}(x) * [\![x]\!]^p \right) + \sum_{\gamma=0}^{r/2-1} c_\gamma \sum_{m=1}^{p} (-1)^m [\![x_\gamma]\!]^m \left(b_{n+1}(x) * [\![x]\!]^{p-m} \right) \\ &\quad + \sum_{\gamma=r/2+1}^{r} c_\gamma \sum_{m=1}^{p} (-1)^m [\![x_\gamma]\!]^m \left(b_{n+1}(x) * [\![x]\!]^{p-m} \right) \\ &\quad + c_{r/2} \sum_{m=1}^{p} (-1)^m [\![0]\!]^m \left(b_{n+1}(x) * [\![x]\!]^{p-m} \right) = [\![x]\!]^p, \quad p = 0, \ldots, r. \end{aligned} \tag{41}$$

Using (1) and symmetry, we can rewrite the relation above for $p = n, \ldots, r$ as

$$b_{n+1}(x) * [\![x]\!]^p + 2\sum_{\gamma=0}^{r/2-1} c_\gamma \sum_{m=1}^{\lfloor p/2 \rfloor} [\![x_\gamma]\!]^{2m} \left(b_{n+1}(x) * [\![x]\!]^{p-2m} \right) = [\![x]\!]^p. \tag{42}$$

The B-spline convolutions in the relation above can be further simplified using a_k (see Theorem 1 and Lemma 3). Similar to the proof provided for Theorem 1, it suffices to consider $p = r$. The Fourier transform of B-spline is an even function and hence, $a_k = 0$ for $k = 2m+1$. Therefore, the relation above results in exactly $\frac{r}{2}+1$ equations (in terms of a_k) which will uniquely specify the $\frac{r}{2}+1$ kernel coefficients. □

For the rest of the discussion in this section, we will focus on specific *symmetric* kernels that reproduce polynomials up to degree r:

Table 2 Linear system of equations constructed using Theorem 2 in order to compute the kernel coefficients of $K^{2n+1,n+2}$ and $K^{2n+1,n}$

$K^{2n+1,n}$	$K^{2n+1,n+2}$
$n=2$	$n=2$
$\begin{cases} c_2+2c_1+2c_0=1 \\ c_1+4c_0=\frac{-1}{12} \\ c_1+10c_0=\frac{1}{-60} \end{cases}$	$\begin{cases} c_2+2c_1+2c_0=1 \\ c_1+4c_0=\frac{-1}{6} \\ c_1+8c_0=\frac{1}{-20} \end{cases}$
$c_0=c_4, \quad c_1=c_3$	$c_0=c_4, \quad c_1=c_3$
$n=3$	$n=3$
$\begin{cases} c_3+2c_2+2c_1+2c_0=1 \\ c_2+4c_1+9c_0=\frac{-1}{8} \\ 5c_2+44c_1+189c_0=\frac{-13}{80} \\ 23c_2+428c_1+3375c_0=\frac{-41}{168} \end{cases}$	$\begin{cases} c_3+2c_2+2c_1+2c_0=1 \\ c_2+4c_1+9c_0=\frac{-5}{24} \\ 7c_2+52c_1+207c_0=\frac{-23}{48} \\ 77c_2+1028c_1+6933c_0=\frac{-1135}{504} \end{cases}$
$c_2=c_4, \quad c_1=c_5, \quad c_0=c_6$	$c_2=c_4, \quad c_1=c_5, \quad c_0=c_6$

Table 3 Approximation properties of $K^{2n+1,n}$ and $K^{2n+1,n+2}$ for $n=2,3$

	Support	Continuity	Polynom. reprod.
$k^{2n+1,n}(x)$	$3n$	C^{n-2}	$\mathbb{P}^{2n}$
$k^{2n+1,n+2}(x)$	$3n+2$	C^{n}	$\mathbb{P}^{2n}$

1. In simulation sciences, $K^{2n+1,n+1}$ (i.e., $r=2n$) has been studied in order to construct a class of postprocessing kernels which will be discussed in detail in Sect. 4.
2. Motivated by (1) and for illustration purposes, we briefly introduce two new kernels, namely: $K^{2n+1,n+2}$ and $K^{2n+1,n}$ for $n=2,3$. We study their superconvergence properties for postprocessing of the DG approximation of linear hyperbolic equations in the next section.

Table 2 summarizes the linear system of equations used to compute kernel coefficients for $n=2,3$ for both $K^{2n+1,n+2}$ and $K^{2n+1,n}$ using Theorem 2. Furthermore, Table 3 summarizes their approximation properties and Fig. 3 demonstrates these kernels.

To conclude this section, we remark that while we have only demonstrated our results for the univariate case, they easily extend to higher dimensions using tensor products. Moreover, while Eq. (22) defines a *specific* class of kernels with a specific target application in mind, our approach can be generalized to other classes of kernels, based on specific application requirements; for example, one could generate combinations of B-splines that exactly interpolate function samples. Such generalizations are beyond the scope of this article.

4 Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering

We now provide a brief introduction to the symmetric and one-sided SIAC kernels as examples of *compactly-supported* filters designed based on a linear combination of B-splines. The compact support of the SIAC kernel along with its superconvergence properties in approximating the DG solutions is one of the main reasons for its popularity in simulation science [2,13,30].

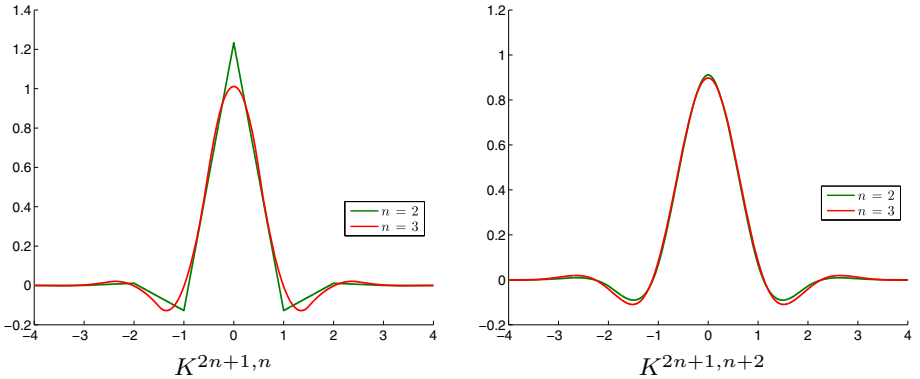

Fig. 3 Kernels introduced in Eq. (37) for specific choices of n and r

We first focus on the *symmetric* SIAC kernel and then show how results from the previous section also generalize to the one-sided SIAC kernel.

4.1 Symmetric SIAC Kernel

The symmetric SIAC kernels form a class of filtering techniques for postprocessing of DG solutions. For a DG approximation of order $k+1$, the symmetric SIAC kernel is constructed using a linear combination of $2k+1$ symmetric B-splines of order $k+1$,

$$K^{(2k+1,k+1)}(x) = \sum_{\gamma=0}^{2k} c_\gamma b_{k+1}(x+k-\gamma), \qquad r = 2k \tag{43}$$

where c_γ denotes the kernel coefficients. The kernel coefficients c_γ are fully specified by enforcing the polynomial reproduction constraint in Theorem 2.

The finite number of B-splines used in the construction of the symmetric SIAC kernel results in the compactness of its support: $[-\frac{3k+1}{2}, \frac{3k+1}{2}]$.

Since a component of the error of the DG method converges with order $2k+1$ in the L^2 norm, the SIAC kernel is constructed with $2k+1$ B-splines and forced to reproduce polynomials up to degree $2k$ [2]. SIAC filtering increases the inter-element continuity up to C^{k-1} and raises the convergence rate of the DG solution from order $k+1$ to order $2k+1$ for linear hyperbolic equations solved over a uniform mesh [2]. Convergence properties of SIAC filtering and its effectiveness have been widely studied in the literature [2–4,11,31,32].

While Theorem 2 uniquely specifies all the kernel coefficients of the SIAC kernel of order k, conventionally the computation of the kernel coefficients is accomplished by inverting a matrix-vector system with a large condition number [13]. By revisiting the symmetric SIAC kernel construction and using the results from Sect. 3, we introduce a new formulation of the symmetric SIAC kernel construction that entails a direct and exact computation of the kernel coefficients in terms of rational numbers. This new formulation also results in the introduction of a *family* of approximation kernels from which the symmetric SIAC kernel is a specific member with desirable and optimal superconvergence properties. In general, we can represent the family of SIAC kernels by $K^{(r+1,n+1)}$ as in Eq. 22. We can better understand the polynomial reproduction of the symmetric SIAC kernel up to the approximation power of b_{n+1}, $(n = k)$ through an analysis similar to the one presented in Theorem 1. Table 4 summarizes such analysis for $k = 1, 2, 3$. Note that these equations are the result of writing the

Table 4 Linear system of equations adopted from Theorem 1 for $p = 0, \ldots, k$

$k = 1$	$k = 2$	$k = 3$
$c_1 + 2c_0 = 1$	$\begin{cases} c_2 + 2c_1 + 2c_0 = 1 \\ c_1 + 4c_0 = \frac{-1}{8} \end{cases}$	$\begin{cases} c_3 + 2c_2 + 2c_1 + 2c_0 = 1 \\ c_2 + 4c_1 + 9c_0 = \frac{-1}{6} \end{cases}$

Table 5 Polynomial reproduction analysis of SIAC filer using Theorem 2 for $k = 1, 2, 3$

$k = 1$	$k = 2$	$k = 3$
$\begin{cases} c_1 + 2c_0 = 1 \\ c_0 = \frac{-1}{12} \end{cases}$	$\begin{cases} c_2 + 2c_1 + 2c_0 = 1 \\ c_1 + 4c_0 = \frac{-1}{8} \\ 5c_1 + 44c_0 = \frac{-312}{1920} \end{cases}$	$\begin{cases} c_3 + 2c_2 + 2c_1 + 2c_0 = 1 \\ c_2 + 4c_1 + 9c_0 = \frac{-1}{6} \\ 6c_2 + 48c_1 + 198c_0 = \frac{-3}{10} \\ 21c_2 + 324c_1 + 2349c_0 = \frac{-12240}{30240} \end{cases}$
$\begin{cases} c_1 = \frac{7}{6} \\ c_0 = c_2 = \frac{-1}{12} \end{cases}$	$\begin{cases} c_2 = \frac{437}{320} \\ c_0 = c_4 = \frac{-97}{480} \\ c_1 = c_3 = \frac{37}{1920} \end{cases}$	$\begin{cases} c_3 = \frac{12223}{7560} \quad c_2 = c_4 = \frac{-919}{2520} \\ c_1 = c_5 = \frac{311}{5040} \quad c_0 = c_6 = \frac{-41}{7560} \end{cases}$

continuous convolution of SIAC kernel with polynomials in closed form rather than numerical evaluation of the convolution. Theorem 1 ensures the full deployment of the approximation power of b_{n+1} ($n = k$) along with enforcing the partition of unity as an essential requirement to construct a valid kernel. As Table 4 shows, the linear system of equations adopted from Theorem 1 is an under-determined system which represents a *family* of symmetric kernels. It is important to note that all the kernels in this family have compact support while providing C^{k-1} continuity. The size of the support varies with B-spline order, $n + 1 = k + 1$, and the number of B-splines, $r + 1 = 2k + 1$.

The symmetric SIAC kernel as one of the members of this family attains superconvergence order of $2k + 1$ through additional polynomial reproduction constraint. Notice that the SIAC kernel is capable of reproducing polynomials up to degree $r = 2k$. Theorem 2 can be directly used to *exactly* specify the kernel coefficients for symmetric SIAC kernel by a simple Gaussian elimination, without any need to numerically solve the system. Table 5 summarizes the system and rational coefficients for the first three orders of the symmetric SIAC kernel.

Before we demonstrate how the new formulation of the symmetric SIAC kernel can be extended for one-sided SIAC kernel, we present a theorem concerning the superconvergence properties of symmetric kernels of type $K^{(r+1,n+1)}$ introduced in Eq. 37 for DG approximation of linear hyperbolic equations. The following theorem specifically states how the error of the filtered DG solution (using $K^{(r+1,n+1)}$) depends upon the number of B-splines, $r + 1$, and the B-spline order, $n + 1$. This provides a wider class of superconvergent filtered solutions for the DG approximation.

Theorem 3 *Let $u_h(x)$ denote the DG approximation of order $k + 1$ to the true solution $u(x) \in H^s$ (Hilbert space), which solves a linear hyperbolic equation (with upwind flux and periodic boundary conditions). Let $K^{(r+1,n+1)}$ denote the class of symmetric kernels of the form introduced in Eq. (37). For $n \geq 1$, and sufficiently smooth $u(x)$ and based on the relation between r and n, we have the following relation for the approximation error bound of the postprocessing of $u_h(x)$ using $K^{(r+1,n+1)}$ at $T > 0$:*

$$\|u(x) - (K_h^{(r+1,n+1)} * u_h)(x)\|_{0,\Omega} \leq Ch^s \tag{44}$$

where $K_h^{(r+1,n+1)}(\cdot) = \frac{1}{h} K_h^{(r+1,n+1)}(\frac{\cdot}{h})$ *and* $s = \min\{r+1, k+n+2, 2k+1\}$ *[2, Theorem 3.3].*

Remark: Here we summarize the important points regarding the relation between the number and order of B-splines to the superconvergence error analysis. In order to demonstrate this relation, we first decompose the left-hand-side of Eq. 44 as

$$\begin{aligned} &\|u(x) - (K_h^{(r+1,n+1)} * u_h)(x)\|_{0,\Omega} \\ &\quad \leq \| \underbrace{u(x) - (K_h^{(r+1,n+1)} * u)(x)}_{\text{filter error}} \|_{0,\Omega} + \| \underbrace{(K_h^{(r+1,n+1)} * (u - u_h))(x)}_{\text{approximation error}} \|_{0,\Omega}. \end{aligned} \tag{45}$$

The first term is entirely dependent on the polynomial reproduction property (not the B-spline order) and hence, this term can be bounded by defining $s = r+1$. The error bound for the second term (i.e., approximation error) in Eq. (45) depends on the order of B-spline used to construct $K^{(r+1,n+1)}$. For $n \geq k$, the negative order norm property (see Eq. 7) can be used to bound the error using $s = 2k+1$. On the other hand, $(u_h * K_h^{(r+1,n+1)})(x) \in \mathbb{P}^{k+n+1}$ and therefore, $s = k+n+2$. Therefore, the error of the postprocessed solution can be bounded by defining $s = \min\{r+1, k+n+2, 2k+1\}$.

4.2 One-Sided SIAC Kernel

Due to the symmetric nature of the previously described SIAC kernel around the evaluation point, its utility for postprocessing near the boundaries or shocks is limited. In order to solve this problem, one-sided SIAC kernels have been introduced [5,10] of the form

$$K^{(r+1,n+1)}(\bar{x}) = \sum_{\gamma=0}^{r} c_\gamma b_{n+1}(\bar{x} - x_\gamma) \tag{46}$$

where c_γ denotes the kernel coefficient, $b_{n+1}(x)$ denotes the symmetric B-spline of order $n+1$, $\bar{x}$ denotes the evaluation point and x_γ represents the B-splines centers defined as

$$x_\gamma = -\frac{r}{2} + \gamma + \zeta(\bar{x}), \qquad \gamma = 0, \ldots, r. \tag{47}$$

$\zeta(\bar{x})$ in this relation represents a shift function as defined in [5,10]. Note that for the one-sided kernel, the kernel coefficients *depend* on the evaluation point (unlike the symmetric SIAC kernel). Therefore, for the rest of the discussion we use $c_\gamma(\bar{x})$ to denote the position-dependent kernel coefficients. Similar to symmetric SIAC kernel, the polynomial reproduction up to degree r uniquely specifies all the kernel coefficients for the one-sided SIAC kernel. The following theorem demonstrates how the results from Sect. 3 can be used in order to compute the kernel coefficients for one-sided SIAC kernel.

Theorem 4 *The kernel coefficients of the one-sided SIAC kernel introduced in Eq. 46 with polynomial reproduction property for polynomials of degree up to r can be uniquely specified*

using the following relations:

$$
\begin{aligned}
&(1)\ \sum_{\gamma=0}^{r} c_\gamma(\bar{x}) = 1 \\
&(2)\ \sum_{\gamma=0}^{r} c_\gamma(\bar{x}) \sum_{m=0}^{p} [\![-x_\gamma]\!]^m G_{p-m}(\bar{x}) = [\![\bar{x}]\!]^p, \quad p = 0, \ldots, n \\
&(3)\ \sum_{\gamma=0}^{r} c_\gamma(\bar{x}) \sum_{m=0}^{p} [\![-x_\gamma]\!]^m \left(b_{n+1}(\bar{x}) * [\![\bar{x}]\!]^{p-m} \right) = [\![\bar{x}]\!]^p, \quad p = 0, \ldots, r.
\end{aligned} \tag{48}
$$

Proof The polynomial reproduction property of the one-sided SIAC kernel can be rewritten in terms of a B-spline convolution (similar to the proof provided for Theorem 1) as

$$
\sum_{\gamma=0}^{r} c_\gamma(\bar{x}) \sum_{m=0}^{p} [\![-x_\gamma]\!]^m \left(b_{n+1}(\bar{x}) * [\![\bar{x}]\!]^{p-m} \right) = [\![\bar{x}]\!]^p, \quad p = 0, \ldots, r. \tag{49}
$$

Note that the kernel coefficients $c_\gamma(\bar{x})$ are dependent on the evaluation points, however, they are not part of the convolution. Therefore, the proof follows from Theorem 1 (for $p \leq n$) and similar to Theorem 2 (for $n < p < r$). □

5 Numerical Results

In this section, we compare and contrast various choices of kernels we discussed in the previous section in terms of their approximation error behavior. The first set of results provide some insight about how function approximation using various choices of kernels would differ from one another for a simple 1D example using semi-discrete convolution. We have generated a set of uniformly spaced sample points and approximated the underlying 1D function using various choices of quadratic kernels, namely: quadratic B-spline using proper quasi-interpolation, symmetric SIAC kernel and cardinal spline function. Figure 4 demonstrates the approximated functions in blue curves. Note that compared to quadratic B-spline approximation, both SIAC filtering and cardinal spline interpolation provide a better approximation for the sampling points, while the cardinal spline provides *exact* interpolation of the sampling points.

Our second set of numerical results aims to study the effect of using various choices of the kernels introduced in postprocessing the DG approximation of a differential equation using quadrature approximation [33] of the continuous convolution in 2D. Note that the kernels introduced in Sect. 3 can all be easily extended to 2D using tensor products. For this example, we used the DG approximation of the linear transport equation at the final time as reported in [31]:

$$
u_t + \nabla \cdot u = 0, \qquad u(\mathbf{x}, 0) = \sin(x + y), \qquad \mathbf{x} \in \Omega = [0, 2\pi]^2. \tag{50}
$$

A DG approximation of the equation above consists of a discontinuous approximate solution u_h over a spatial discretization of the domain, Ω_h. For each element $e_h \in \Omega_h$, we seek $u_h \in V_h$ where V_h denotes the space of piecewise polynomials of degree k. Considering a test function $v_h \in V_h$ and the weak formulation of the transport equation above, for each element e_h we have

$$
\int_{e_h} (u_h)_t v \, dx - \int_{e_h} a u_h \cdot \nabla v \, dx + \int_{\Gamma_{e_h}} \widehat{a u_h} \cdot \mathbf{n} \, ds = 0, \tag{51}
$$

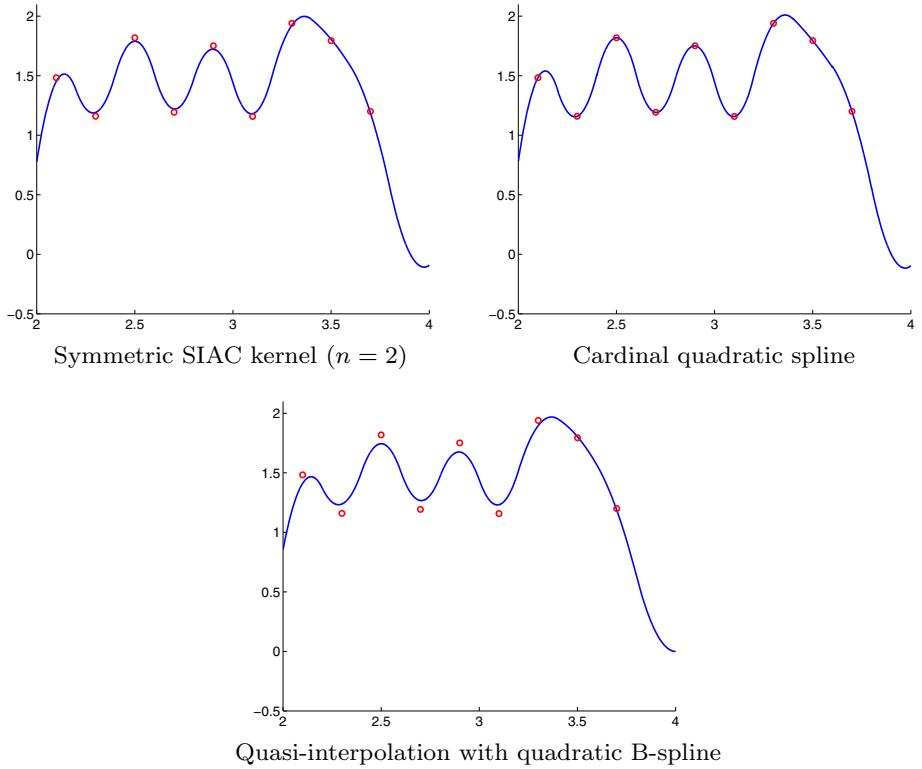

Fig. 4 The effect of using various kernels in the reconstruction of a 1D random function using uniformly spaced sampling points and semi-discrete convolution. The function approximation carried out using various kernels namely: symmetric SIAC kernel for $n = 2$, cardinal quadratic spline and quadratic B-spline along with proper quasi-interpolation. The *blue curve* in the figure represents the approximated function and the function sample points are shown as *red points* (Color figure online)

where **n** denotes the unit outward normal, Γ_{e_h} denotes the boundary of element e_h and $\widehat{ah_h}$ is the numerical flux. One can find $u_h \in V_h$ such that the above equation is satisfied for all test functions in V_h. For the purpose of the current manuscript, we consider an upwind flux to find u_h and used Nektar++ software package [34] in order to find the DG approximation.

The postprocessing of the DG solution $u_h(x)$ at the final time T has been carried out through the quadrature approximation procedure proposed in [33]. The *approximation error* for all the examples is defined as the difference between the postprocessed DG solution and the true solution: $K_h(x) * u_h(x) - u(x)$ where $K_h(x)$ represents a generic scaled kernel, $u_h(x)$ the DG solution and $u(x)$ the true solution of Eq. (50). For all the experiments, we have chosen 80 sampling points over each mesh element to compute the approximation errors.

For cardinal spline interpolation of the DG solution using continuous convolution, special care is required in the interpretation of the results because the infinite support of the cardinal spline filter has been truncated for evaluation. We have used 17 B-splines to form a truncated version of the quadratic cardinal spline filter. As Table 6 demonstrates postprocessing of the DG approximation using a cardinal B-spline filter only increases the order of convergence of the postprocessed solution by one.

Table 6 L_2 and L_∞ norm of the approximation error using cardinal quadratic spline kernel

Quadrilateral meshes								
Mesh	DG				Cardinal B-spline			
	L_2 error	Order	L_∞ error	Order	L_2 error	Order	L_∞ error	Order
$\mathbb{P}^2$								
20^2	9.75e−05	–	5.09e−04	–	5.1e−05	–	7.22e−05	–
40^2	1.22e−05	2.99	6.43e−05	2.98	3.06e−06	4.05	4.33e−06	4.05
80^2	1.52e−06	3.00	8.06e−06	2.99	1.41e−07	4.43	1.99e−07	4.44
160^2	1.90e−07	3.00	1.01e−06	2.99	1.22e−08	3.53	1.73e−08	3.52

Table 7 presents the approximation errors corresponding to various choices of kernels for different resolutions of a quadrilateral mesh. In addition, Fig. 5 shows the contour plot of the approximation error over the whole domain in logarithmic scale. Note that all the kernels reported in this table use $2n + 1$ B-splines while the order of B-spline differs in each case. The numerical values in Table 7 verifies the $2k + 1$ order of convergence of symmetric SIAC filter (as reported in [31]) and kernels constructed using B-spline orders beyond $n + 1$ as proved in Theorem 3. While the numerical results demonstrate better order of convergence for kernels with lower orders of B-splines in some cases, the contour plots in Fig. 5 clearly show the oscillatory nature of the approximation error compared to SIAC kernel and kernels constructed using higher-order B-splines. Notice that the order of convergence has dropped in case of quadratic polynomial for $K^{2n+1,n-1}$ and in case of cubic polynomial at resolution 160^2. It is worth noting that all the experiments have been carried out up to basic floating point precision. For higher resolutions and higher-order polynomial orders (for instance in our case, quartic polynomial at resolution 160^2), extended precision is required in order to achieve the order of accuracy expected.

Using results summarized in Table 3, it is easy to conclude that kernels constructed using B-splines of order less than $n+1$ provides slightly higher computational efficiency compared to symmetric SIAC kernel of the same order. In contrast, kernels constructed using B-splines of order higher than $n + 1$ increases the smoothness of the the postprocessed results with a slightly higher computational cost. These properties can be used to decide which type of kernel to be used in the application based on the tradeoff required between computational efficiency, smoothness and superconvergence considerations.

6 Conclusion

In this paper we presented and established the theoretical results that demonstrate the connection between symmetric SIAC kernel construction and approximation theory. Specifically, we derived a closed formulation for the convolution of SIAC kernels with polynomials that leads to a *direct* and *exact* scheme to solve for the kernel coefficients rather than through numerical computation. Moreover, studying the symmetric SIAC kernel in this framework, introduces a *family* of kernels from which symmetric SIAC kernel satisfies $2n + 1$ order of superconvergence property. The introduction of this family of kernels to the community can be helpful to design application-specific kernels from this family with specific design criteria. For demonstration, we have studied variations of the symmetric SIAC kernel where the order

Table 7 L_2 and L_∞ norm of the approximation error for different kernels discussed in Sects. 3 and 4

Quadrilateral Meshes												
Mesh	DG				$K^{2n+1,n-1}$				$K^{2n+1,n}$			
	L_2 error	Order	L_∞ error	Order	L_2 error	Order	L_∞ error	Order	L_2 error	Order	L_∞ error	Order
$\mathbb{P}^2$												
20^2	9.75e−05	–	5.09e−04	–	3.75e−06	–	1.02e−05	–	2.42e−06	–	3.43e−06	–
40^2	1.22e−05	2.99	6.43e−05	2.98	2.37e−07	3.98	5.82e−07	4.13	3.85e−08	5.97	5.57e−08	5.94
80^2	1.52e−06	3.00	8.06e−06	2.99	1.50e−08	3.98	3.54e−08	4.03	6.21e−10	5.95	9.43e−10	5.88
160^2	1.90e−07	3.00	1.00e−06	3.01	9.40e−10	3.99	2.2e−09	4.00	1.07e−11	5.85	1.85e−11	5.67
$\mathbb{P}^3$												
20^2	1.93e−06	–	1.14e−05	–	4.16e−08	–	5.98e−08	–	8.12e−08	–	1.14e−08	–
40^2	1.2e−07	4.00	7.22e−07	3.98	1.65e−10	7.97	2.47e−10	7.91	3.23e−10	7.97	4.57e−10	4.64
80^2	7.54e−09	3.99	4.52e−08	3.99	6.65e−13	7.95	1.14e−12	7.75	1.26e−12	8.00	1.79e−12	7.99
160^2	4.71e−10	4.00	2.83e−09	3.99	7.01e−15	6.56	3.34e−14	5.09	5.23e−15	7.77	7.90e−15	7.82
$\mathbb{P}^4$												
20^2	3.04e−08	–	1.99e−07	–	1.55e−09	–	2.19e−09	–	2.71e−09	–	3.84e−09	–
40^2	9.53e−10	4.99	6.3e−09	4.98	1.55e−12	9.96	2.20e−12	9.95	2.72e−12	9.96	3.86e−12	9.95
80^2	2.97e−11	5.00	1.97e−10	4.99	2.20e−15	9.46	8.77e−15	7.97	3.48e−15	9.61	1.29e−14	8.22
160^2	9.33e−13	4.99	6.17e−12	4.99	4.47e−16	2.29	2.58e−15	1.76	5.04e−16	2.78	2.52e−15	2.35

Table 7 continued

Mesh	DG				$K^{2n+1,n+1}$ (SIAC)				$K^{2n+1,n+2}$			
	L_2 error	order	L_∞ error	order	L_2 error	order	L_∞ error	order	L_2 error	order	L_∞ error	order
$\mathbb{P}^2$												
20^2	9.75e−05	–	5.09e−04	–	4.47e−06	–	6.34e−06	–	7.23e−06	–	1.02e−06	–
40^2	1.22e−05	2.99	6.43e−05	2.98	7.09e−08	5.97	1.00e−07	5.98	1.14e−07	5.98	1.62e−07	2.65
80^2	1.52e−06	3.00	8.06e−06	2.99	1.11e−09	5.99	1.57e−09	5.99	1.80e−09	5.98	2.55e−09	5.98
160^2	1.90e−07	3.00	1.00e−06	3.01	1.73e−11	6.00	2.46e−11	5.99	2.82e−11	5.99	3.99e−11	5.99
$\mathbb{P}^3$												
20^2	1.93e−06	–	1.14e−05	–	1.37e−07	–	1.94e−07	–	2.14e−07	–	3.03e−07	–
40^2	1.2e−07	4.00	7.22e−07	3.98	5.49e−10	7.96	7.77e−10	7.96	8.59e−10	7.96	1.21e−09	7.96
80^2	7.54e−09	3.99	4.52e−08	3.99	2.15e−12	7.99	3.05e−12	7.99	3.37e−12	7.99	4.78e−12	7.98
160^2	4.71e−10	4.00	2.83e−09	3.99	9.21e−15	7.86	3.04e−14	6.64	1.37e−14	7.94	2.73e−14	7.45
$\mathbb{P}^4$												
20^2	3.04e−08	–	1.99e−07	–	4.37e−09	–	6.18e−09	–	6.66e−09	–	9.42e−09	–
40^2	9.53e−10	4.99	6.3e−09	4.98	4.40e−12	9.95	6.23e−12	9.95	6.73e−12	9.95	9.51e−12	9.95
80^2	2.97e−11	5.00	1.97e−10	4.99	3.16e−15	10.44	1.13e−14	9.10	7.10e−15	9.88	1.03e−14	9.85
160^2	9.33e−13	4.99	6.17e−12	4.99	6.43e−16	2.29	2.44e−15	2.21	6.23e−16	3.51	2.41e−15	2.09

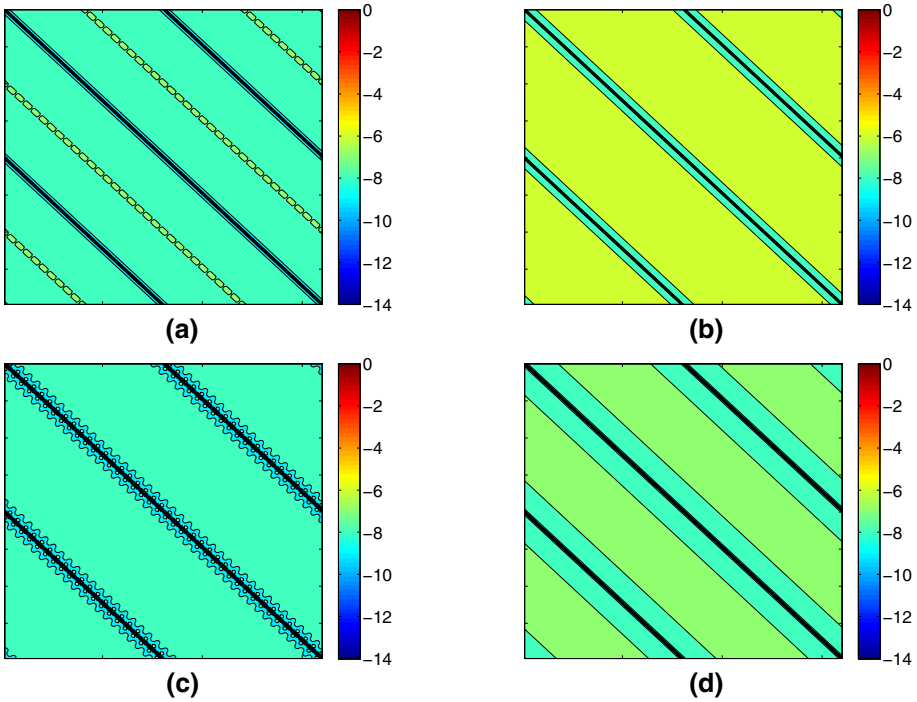

Fig. 5 Contour plot of the approximation error in a logarithmic scale. **a** SIAC kernel ($n = 2$). **b** Cardinal quadratic interpolation. **c** $K^{5,2}$. **d** $K^{5,4}$

of the B-spline used to construct the symmetric SIAC kernel was changed. We provided theoretical results demonstrating that the order $2n + 1$ superconvergence property for linear hyperbolic equations can be preserved when the order of B-splines is higher than the order used in the original symmetric SIAC kernel. Studying other variations of the symmetric SIAC kernel in the family introduced along with their superconvergence properties is an interesting open question which can provide a potential direction for future research. Another interesting line of future research is to investigate the postprocessing of DG solutions with discontinuity and shocks.

Acknowledgments The authors wish to thank Dr. Hanieh Mirzaee and Mr. James King for helpful discussion and sharing the symmetric SIAC postprocessing code, Xiaozhou Li for confirming the convergence results and Varun Shankar for useful suggestions. The authors are sponsored in part by the Air Force Office of Scientific Research (AFOSR), Computational Mathematics Program (Program Manager: Dr. Fariba Fahroo), under grant number FA9550-12-1-0428 (first and third author) and FA8655-13-1-3017 (second author).

Compliance with Ethical Standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Cockburn, B., Karniadakis, G.E., Shu, C.W. (eds.): Discontinuous Galerkin Methods. Springer, Berlin (2000)

2. Cockburn, B., Luskin, M., Shu, C.W., Süli, E.: Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. Mathematics of Computation **72**(242), 577–606 (2003)
3. Ji, L., van Slingerland, P., Ryan, J.K., Vuik, K.: Superconvergent error estimates for a position-dependent smoothness-increasing accuracy-conserving filter for DG solutions. Math. Comput. **83**, 2239–2262 (2014)
4. Ji, L., Xu, Y., Ryan, J.K.: Accuracy enhancement of the linear convection–diffusion equation in multiple dimensions. Math. Comput. **81**, 1929–1950 (2012)
5. Ryan, J.K., Shu, C.W.: One-sided post-processing technique for the discontinuous Galerkin methods. Methods Appl. Anal. **10**(2), 295–308 (2003)
6. van Slingerland, P., Ryan, J.K., Vuik, K.: Position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering for improving discontinuous Galerkin solutions. SIAM J. Sci. Comp. **33**(2), 802–825 (2011)
7. Mirzaee, H., Ji, L., Ryan, J.K., Kirby, R.M.: Smoothness-increasing accuracy-conserving (SIAC) post-processing for discontinuous Galerkin solutions over structured triangular meshes. SIAM J. Numer. Anal. **49**(5), 1899–1920 (2011)
8. Mirzaee, H., King, J., Ryan, J.K., Kirby, R.M.: Smoothness-increasing accuracy-conserving filters for discontinuous Galerkin solutions over unstructured triangular meshes. SIAM J. Sci. Comput. **35**(1), A212–A230 (2013)
9. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Smoothness-increasing accuracy-conserving (SIAC) filters for discontinuous Galerkin solutions: Application to structured tetrahedral meshes. J. Sci. Comp. **58**(3), 690–704 (2014)
10. Walfisch, D., Ryan, J.K., Kirby, R.M., Haimes, R.: One-sided smoothness-increasing accuracy-conserving filtering for enhanced streamline integration through a discontinuous fields. J. Sci. Comp. **38**(2), 164–184 (2009)
11. Bramble, J.H., Schatz, A.H.: Higher order local accuracy by averaging in the finite element method. Math. Comput. **31**(137), 94–111 (1977)
12. Höllig, K.: Finite element methods with B-splines. Soc. Ind. Appl. Math. (2003)
13. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Efficient implementation of smoothness-increasing accuracy-conserving (SIAC) filters for discontinuous Galerkin solutions. J. Sci. Comp. **52**(1), 85–112 (2012)
14. Thomée, V.: High order local approximations to derivatives in the finite element method. Math. Comput. **31**(139), 652–660 (1977)
15. Unser, M.: Splines: a perfect fit for signal and image processing. IEEE Signal Process. Mag. **16**(6), 22–38 (1999)
16. Unser, M.: Sampling: 50 years after Shannon. Proc. IEEE **88**(4), 569–587 (2000)
17. Unser, M., Aldroubi, A., Eden, M.: B-Spline signal processing: part I-theory. IEEE Trans. Signal Process. **41**(2), 821–833 (1993)
18. Vuçini, E., Möller, T., Gröller, M.E.: On visualization and reconstruction from non-uniform point sets using b-splines. Comput. Graph. Forum **28**(3), 1007–1014 (2009)
19. DeBoor, C.: A Practical Guide to Splines. Springer, Berlin (1978)
20. de Boor, C.: Approximation order without quasi-interpolants. In: Cheney, E., Chui, C., Schumaker, L. (eds.) Approximation Theory VII, pp. 1–18 (1993)
21. de Boor, C., Höllig, K., Riemenschneider, S.: Box Splines. Springer New York Inc., New York, NY (1993)
22. Cohen, E., Riesenfeld, R.F., Elber, G.: Geometric Modeling with Splines: An Introduction. A K Peters, Natick, MA (2001)
23. Strang, G., Fix, G.: A fourier analysis of the finite element variational method. In: Constructive Aspects of Functional Analysis, pp. 796–830 (1971)
24. De Boor, C., Daniel, J.: Splines with Non-Negative B-Spline Coefficients: CNA. Defense Technical Information Center (1973)
25. Marsden, M.: An identity for spline functions with applications to variation diminishing spline approximation. MRC technical summary report. University of Wisconsin–Madison (1968)
26. Mirzargar, M.: A Reconstruction Framework for Common Sampling Lattices. Ph.D. thesis, University of Florida, Gainesville, FL (2012)
27. Mirzargar, M., Entezari, A.: Quasi interpolation with voronoi splines. IEEE Trans. Vis. Comput. Graph. **17**(12), 1832–1841 (2011)
28. Schoenberg, I.J.: Cardinal Spline Interpolation. Society for Industrial and Applied Mathematics, Philadelphia (1973)
29. Unser, M., Aldroubi, A., Eden, M.: Fast B-Spline transforms for continuous image representation and interpolation. IEEE Trans. Pattern Anal. Mach. Intell. **13**(3), 277–285 (1991)
30. Steffen, M., Curtis, S., Kirby, R.M., Ryan, J.K.: Investigation of smoothness-increasing accuracy-conserving filters for improving streamline integration through discontinuous fields. IEEE Trans. Vis. Comput. Graph. **14**(3), 680–692 (2008)

31. King, J., Mirzaee, H., Ryan, J.K., Kirby, R.M.: Smoothness-increasing accuracy-conserving (SIAC) filtering for discontinuous Galerkin solutions: Improved errors versus higher-order accuracy. J. Sci. Comp. **53**(1), 129–149 (2012)
32. Mirzaee, H.: Smoothness-Increasing Accuracy-conserving Filters (SIAC) for Discontinuous Galerkin Solutions. Ph.D. thesis, University of Utah, Salt Lake City, UT (2012)
33. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Quantification of errors introduced in the numerical approximation and implementation of smoothness-increasing accuracy conserving (SIAC) filtering of discontinuous Galerkin (DG) fields. J. Sci. Comp. **45**(1–3), 447–470 (2010)
34. Nektar++ (2014). http://www.nektar.info