Hexagonal Smoothness-Increasing Accuracy-Conserving Filtering

Mahsa Mirzargar · Ashok Jallepalli · Jennifer K. Ryan · Robert M. Kirby

In honor of Prof. Chi-Wang Shu's 60th Birthday.

Abstract Discontinuous Galerkin (DG) methods are a popular class of numerical techniques to solve partial differential equations due to their higher order of accuracy. However, the inter-element discontinuity of a DG solution hinders its utility in various applications, including visualization and feature extraction. This shortcoming can be alleviated by postprocessing of DG solutions to increase the inter-element smoothness. A class of postprocessing techniques proposed to increase the inter-element smoothness is SIAC filtering. In addition to increasing the inter-element continuity, SIAC filtering also raises the convergence rate from order $k+1$ to order $2k+1$. Since the introduction of SIAC filtering for univariate hyperbolic equations by Cockburn et al. [9], many generalizations of SIAC filtering have been proposed. Recently, the idea of dimensionality reduction through rotation has been the focus of studies in which a univariate SIAC kernel has been used to postprocess a two-dimensional DG solution [31]. However, the scope of theoretical development of multidimensional SIAC filters has never gone beyond the usage of tensor product multidimensional B-splines or the reduction of the filter dimension. In this paper, we define a new SIAC filter called hexagonal SIAC (HSIAC) that uses a nonseparable class of two-dimensional spline functions called hex splines. In

Mahsa Mirzargar
Department of Computer Science, University of Miami, Coral Gables, FL, 33146, USA
E-mail: mirzargar@cs.miami.edu

Jennifer K. Ryan
School of Mathematics, University of East Anglia, Norwich E-mail: jennifer.ryan@uea.ac.uk

Ashok Jallepalli
Scientific Computing and Imaging (SCI) Institute, University of Utah, Salt Lake City, UT, 84112 USA E-mail: ashokj@sci.utah.edu

Robert M. Kirby
School of Computing and the Scientific Computing and Imaging (SCI) Institute, University of Utah, Salt Lake City, UT, 84112 USA E-mail: kirby@cs.utah.edu

addition to relaxing the separability assumption, the proposed HSIAC filter provides more symmetry to its tensor-product counterpart. We prove that the superconvergence property holds for a specific class of structured triangular meshes using HSIAC filtering and provide numerical results to demonstrate and validate our theoretical results.

Keywords B-splines · hex splines · box splines · Smoothness-Increasing Accuracy-Conserving (SIAC) filtering · quasi-interpolation · approximation theory · Discontinuous Galerkin.

1 Introduction

Discontinuous Galerkin (DG) methods are heavily used for numerical solutions of partial differential equations (PDE). DG methods solve a variational form of a partial differential equation without a global continuity requirement over the domain (i.e., a collection of elements). A DG solution is continuous only inside an element whereas discontinuities are allowed across the element boundaries. The inter-element discontinuity of a DG solution presents challenges when using DG in applications. Hence, postprocessing of the DG solution is often required. B-spline-based postprocessing techniques are among the most well-accepted filtering techniques to enhance the smoothness and accuracy of a DG solution.

A class of postprocessing techniques proposed to increase the inter-element smoothness is Smoothness-Increasing Accuracy-Conserving (SIAC) filtering. SIAC filtering of a DG solution of order $k+1$ increases the inter-element continuity up to C^{k-1} and also raises the convergence rate from order $k+1$ to order $2k+1$. Since the introduction of SIAC filtering for univariate hyperbolic equations by Cockburn et al. [9], many generalizations of SIAC filtering have been proposed in both one-dimensional and multidimensional cases [9, 15, 19, 6, 12, 14, 16]. However, the scope of theoretical development of *multidimensional* SIAC filters has never gone beyond the usage of tensor product multidimensional B-splines. In this paper, we use a geometric approach and define a new SIAC filter called hexagonal SIAC (HSIAC) that uses a nonseparable class of two-dimensional spline functions called hex splines.

The main motivation for using hex splines for postprocessing of DG solutions is that they provide a well-defined spline space, containing a polynomial space exactly matching the spline space a tensor-product B-spline counterpart defines, but in a nonseparable fashion. That is, hex splines relax the tensor product assumption. This property provides a natural extension of the superconvergence results of two-dimensional SIAC filters on quadrilateral meshes to structured triangle meshes generated from the subdivision of a hexagonal lattice. We provide the proof of superconvergence properties of HSIAC on these specific structured triangular meshes (i.e., from subdivision of a hexagonal lattice).

In addition to going beyond a tensor-product construction of SIAC filtering, the HSIAC filter will be more (radially) symmetric compared to its B-spline

counterpart. This property is especifically important when SIAC filtering is used for feature extraction [33,31].

The paper proceeds as follows. After an introduction to the notation used in Section 2.1, we provide a brief introduction to B-splines and their generalization, namely the hex splines in Section 2.2. We introduce the HSIAC filter and its superconvergence property in Section 3. Section 4 discusses some numerical results and error contour plots to demonstrate and validate the theoretical results presented and to compare HSIAC with its B-spline counter-part. Finally, we present our conclusions in Section 5.

2 Background

2.1 Notation

We start by introducing the notation used in the remainder of the paper. The superconvergence properties of the SIAC kernel are studied in terms of various error norms. Hence, we provide a brief introduction to the appropriate norms. The interested reader can refer to [9,6,12,14,24] for more details.

Let $H^\ell(\Omega)$ denote a Sobolev space over a bounded open set $\Omega \in \mathbb{R}^d$ as the domain. We use $\|\cdot\|_{0,\Omega}$ to represent the usual L^2-norm over Ω and $\|\cdot\|_{-\ell,\Omega}$ to denote the negative-order norm over the dual space of $H^\ell(\Omega)$ or $H^{-\ell}(\Omega)$ [6] as

$$\|u\|_{-\ell,\Omega} = \sup_{\phi \in \mathcal{C}_0^\infty(\Omega)} \frac{\int_\Omega u(x)\phi(x)dx}{\|\phi\|_{\ell,\Omega}}, \tag{1}$$

where $\mathcal{C}_0^\infty(\Omega)$ denotes the space of infinitely differentiable functions with compact support on Ω. The negative order norm as Cockburn et al. noted in [9] can be used to quantify the oscillatory nature of a function and is often used to prove the superconvergence property of SIAC filtering via the following relation [6, Lemma 4.2]

$$\|u\|_{0,\Omega} \leq C \sum_{|\alpha|\leq \ell} \|D^\alpha u\|_{-\ell,\Omega}, \tag{2}$$

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 $u_h(\cdot)$ its DG solution of order $k+1$.

Where appropriate, we use ∂^α to denote the partial derivative, and ∂_h^α to denote the central difference operator of order α with spacing h. Similarly, we use $\partial^{\boldsymbol{\alpha}}$ and $\partial_h^{\boldsymbol{\alpha}}$ to denote the partial derivative in the multidimensional case where $\boldsymbol{\alpha}$ is a multi-index (e.g., a two-dimensional $\partial^{\boldsymbol{\alpha}}$ is defined as $\partial_x^{\alpha_1}\partial_y^{\alpha_2}$ where $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2$).

2.2 B-spline and Its Generalization to Hexagonal Lattice

The first-order univariate central B-spline (Basis splines) b_1 is defined as the indicator function over the interval, $T = [\frac{-1}{2}, \frac{1}{2}]$[1]:

$$b_1(x) = \mathcal{X}_T(x) = \begin{cases} 1 \; x \in [\frac{-1}{2}, \frac{1}{2}] \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

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

$$b_{n+1}(x) = (b_1 * b_n)(x), \quad n \geq 0. \tag{4}$$

For the rest of the discussion, we simply use the term B-splines to denote a 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

$$\mathbf{S}_n := \text{span}(b_n(\cdot - k))_{k \in \mathbb{Z}}. \tag{5}$$

$\mathbf{S}_{n+1}$ is C^{n-1} continuous and contains polynomials up to degree n [5,10,27].

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 represents that function:

$$s = \sum_{\gamma \in h\mathbb{Z}} c_\gamma b_n(\cdot - \gamma), \tag{6}$$

where c_γ represents the spline coefficient and h denotes the distance between the B-spline centers. In addition to their approximation power, B-splines are attractive from a computational point of view and provide an efficient approximation technique due to their compact support.

To use B-splines for an approximation in higher dimensions, we need to define higher dimensional B-spline functions. A two-dimensional B-spline can be defined using the concept of tensor product:

$$b_n(x, y) = b_n(x) b_n(y). \tag{7}$$

The two-dimensional spline space can now be defined as

$$\mathbf{S}_n := \text{span}(b_n(\cdot - \mathbf{k}))_{\mathbf{k} \in \mathbb{Z}^2}. \tag{8}$$

The spline space defined using two-dimensional B-splines of order $n + 1$ as the basis function is C^{n-1} continuous and contains polynomials up to (total) degree $2n$ [5,10,27]. Note that the definition of a spline space as presented in Eq. 8 has the implicit assumption that $\mathbf{k} \in \mathbb{Z}^2$ is a tensor product of $k \in \mathbb{Z}$ in the univariate case and formally represents a two-dimensional *Cartesian lattice*.

A lattice is a subset of Euclidean space that is formed by periodic arrangement of discrete points and is required to include origin. A lattice is fully

[1] In spline theory, the first-order central B-spline is often denoted as $b_0(x)$.

characterized using the lattice directions, usually denoted in terms of a matrix. For example, the lattice directions for a two-dimensional Cartesian lattice are

$$\mathbf{C} = [e_1, e_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{9}$$

For any lattice there exists a region in which origin is the only lattice point. This region is called a Voronoi region or a Voronoi cell [32]. A lattice is closed under addition and negation, and hence, this region can be considered for any other lattice point via a translation. Therefore, the Voronoi cell of a lattice is unique.

We can use the geometry of the Cartesian lattice to define the first order of a two-dimensional B-spline *directly* as follows

$$b_1(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \text{Voronoi cell} \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where $\mathbf{x}$ denotes a 2D point, and the Voronoi cell in this case corresponds

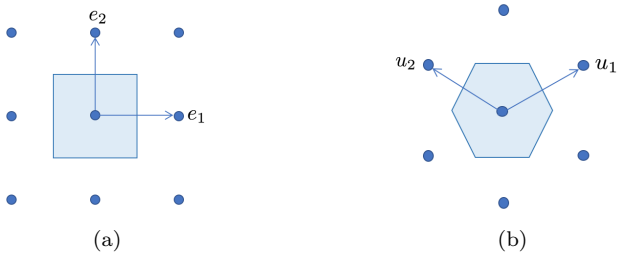
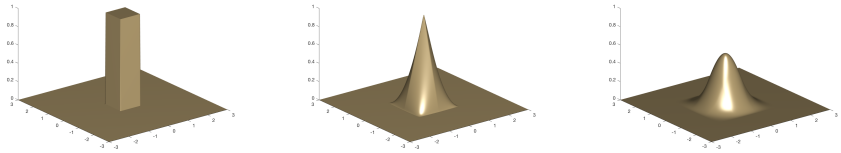


Fig. 1 a) A two-dimensional Cartesian lattice along with its Voronoi cell (the shaded region). The first order of a two-dimensional B-spline is defined as the indicator function of the Voronoi cell. b) A hexagonal lattice along with its Voronoi cell (the shaded region). The first order of hex spline is defined as the indicator function of the Voronoi cell.

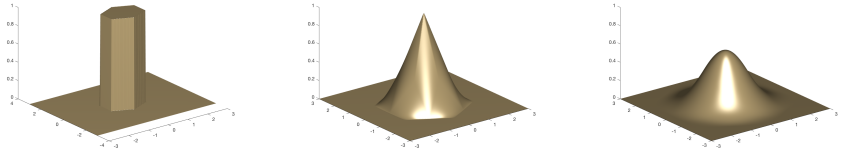
to a two-dimensional Cartesian lattice. A two-dimensional Cartesian lattice along with its Voronoi cell (i.e., the support of the first order B-spline in 2D) is depicted in Figure 1 (a). It is easy to show that the higher order B-splines can still be defined using self-convolution analogous to Eq. 4. The first three orders of two-dimensional B-splines are demonstrated in Figure 2.

In comparison to the formulation of B-splines provided in Eq. 7 that hinders the capability of defining spline-type functions for nonseparable structures, the formulation of B-splines provided in Eq. 10 allows us to define spline spaces for *any* lattice in *any* dimensions [25]. In this paper, we focus on only the two-dimensional case, i.e., the definition of spline functions for a hexagonal lattice. Equivalent spline families for (arbitrary) higher dimensions can be defined using the idea behind Eq. 10 that are referred to as Voronoi splines [26, 25].



(a) (b) (c)

Fig. 2 First (a), second (b), and third (c) order two-dimensional tensor product B-splines.



(a) (b) (c)

Fig. 3 First (a), second (b), and third (c) order hex splines.

2.3 An Introduction to Hexagonal Splines.

The only other lattice in two dimensions that tessellates a two-dimensional Euclidean space is a hexagonal lattice [32], whose lattice directions can be defined as

$$\mathbf{H} = \begin{bmatrix} u_1, u_2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{11}$$

We can now define a two-dimensional hex spline as follows

$$\eta_1(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \text{Voronoi cell} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

where $\mathbf{x}$ denotes a 2D point, and the Voronoi cell in this case represents the Voronoi cell of a hexagonal lattice. A hexagonal lattice along with its Voronoi cell (i.e., the support of the first-order hex spline) is depicted in Figure 1 (b). The higher order hex splines can still be defined using self-convolution analogous to Eq. 4. The first three orders of hex splines are demonstrated in Figure 3.

A spline space can also be defined as the spanning space of the translations of hex splines

$$\mathbf{S}_n^{\eta} := \text{span}(\eta_n(\cdot - \mathbf{k}))_{\mathbf{k} \in \mathbf{H}\mathbb{Z}^2}. \tag{13}$$

Note that in this case the translations of the basis function belong to a hexagonal lattice. The spline space defined using the hex spline of order $n+1$ as the basis function has the *exact* same properties as the spline space defined using

the two-dimensional B-spline of order $n+1$ [34,27]. That is, $\mathbf{S}^{\eta}_{n+1}$ is C^{n-1} continuous and contains polynomials up to (total) degree $2n$ [34,27].

The definition of a hex spline is intuitive from a geometric point of view, yet evaluation of higher order hex splines based on the self-convolution of the first order are hard to evaluate due to the complexity of the boundaries of the hex spline. Therefore, for evaluation, we use the definition of hex splines in terms of a more generic class of splines called box-splines [5]. Box-spline representation of hex splines provides a compact representation and an efficient and tractable computational procedure. For brevity of discussion, we have summarized the box-spline formulation of hex splines in the appendix.

Similar to a two-dimensional B-spline, a hex spline of order $n+1$ can be represented as a C^{n-1} piecewise continuous function where each piece is a polynomial of (total) degree $2n$ [34,26,27]. As discussed in the appendix, one can use box-spline representation of a hex spline in order to show that the partial derivatives of a hex spline can be *exactly* computed using finite differences of the same order [34,5]. This property will be used in Section 3 to prove superconvergence properties of hexagonal SIAC.

From an approximation theory point of view, a hexagonal lattice has multiple advantages over a Cartesian lattice, and hence, the use of hex splines has been studied in the context of signal processing, image processing, and visualization [34,27,26]. A thorough discussion of the advantages of hexagonal lattice and hex splines is beyond the scope of the current manuscript; interested readers can consult [34,11] for further discussion. However, we would like to emphasize that the main motivation for using hex splines for filtering is that hex splines provide a well-defined spline space, containing a polynomial space, on a nonseparable structure (i.e., hexagonal lattice). As we will discuss in the sequel, this property provides a natural extension of the superconvergence results of tensor-product SIAC filters on quadrilateral meshes to structured triangle meshes generated by subdividing a hexagonal lattice. In addition, hex splines provide a 12-fold symmetry in comparison to the 8-fold symmetry of two-dimensional B-splines. Hence, the HSIAC filter introduced in Section 3 will be more (radially) symmetric compared to its Cartesian counterpart. This property is especifically important when the SIAC kernel is used for feature extraction [33,31]. In the next section, we introduce the hexagonal SIAC filter (HSIAC) that uses hex splines as the underlying spline kernel.

3 HSIAC: Hexagonal Smoothness-Increasing Accuracy-Conversing Filter

The construction of hexagonal SIAC (HSIAC) mimics the construction of the (original) SIAC filter using a geometric approach. For brevity of discussion, we first introduce the SIAC filter, and then a geometric approach for construction a SIAC filter construction. We then use this geometric construction to introduce the hexagonal SIAC filter (HSIAC).

3.1 SIAC Filter Review

The SIAC filter is a B-spline-based kernel designed for postprocessing a DG solution. The main motivation for introducing the SIAC filter is defining a compactly supported filter that would preserve the superconvergence properties of the underlying DG solution while enhancing the inter-element smoothness. The SIAC kernel is defined as

$$K^{(2k+1,k+1)}(x) = \sum_{\gamma=0}^{2k} c_\gamma b_{k+1}(x + k - \gamma), \tag{14}$$

where c_γ denotes the kernel coefficients. Using $2k+1$ B-splines to construct the SIAC kernel is motivated by the fact that a component of the error of a DG solution converges with order $2k+1$ in the L^2 norm [9]. The kernel coefficients c_γ are fully specified by enforcing that the kernel reproduces polynomials up to degree $2k$

$$K^{(2k+1,k+1)}(x) * x^p = x^p, \qquad 0 \le p \le 2k. \tag{15}$$

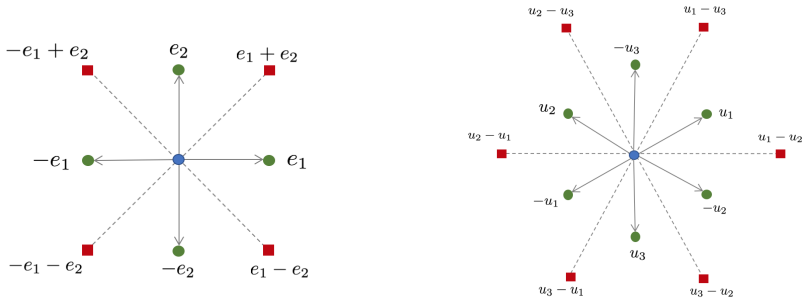
In the relation above, the kernel coefficients can be computed numerically [23] or theoretically [28]. The finite number of B-splines used in the construction of the symmetric SIAC kernel results in the compactness of its support, which provides a balance between the computational cost of the filtering and the approximation properties.

The 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 $2k+1$ (i.e., superconvergence) for linear [9] and nonlinear [15,18,17] hyperbolic equations solved over a uniform mesh, and variable coefficient euqations [17, 24]. The proof has also been extended for local derivative computation [29] and boundary filtering [30,4,13]. Convergence properties of SIAC filtering and its effectiveness have been widely studied in the literature [9,19,6,12,14,16]. Numerical results have shown promising results for the application of a two-dimensional SIAC filter on structured triangular meshes [24], unstructured [21] triangular meshes, tetrahedral meshes [24], and streamline extraction [33,35].

Conventionally, higher dimensional SIAC filters have been defined using the concept of tensor product. For instance, the two-dimensional SIAC kernel can be defined using the one-dimensional version as

$$K^{(2k+1,k+1)}(x) = \sum_{\gamma_1=0}^{2k} \sum_{\gamma_2=0}^{2k} c_{\gamma_1} c_{\gamma_2} b_{k+1}(x + k - \gamma_1)(y + k - \gamma_2). \tag{16}$$

The proof of the superconvergence for multidimensional SIAC on quadrilateral meshes follows the univariate case trivially using the concept of tensor product [9].



(a) (b)

Fig. 4 The direct and indirect neighbors of a two-dimensional evaluation point on (a) Cartesian lattice and (b) a hexagonal lattice. The evaluation point, at the center, is colored in blue. The direct neighbors are the green circles in both cases and the indirect neighbors are the red squares.

3.2 Geometric Construction

Similar to B-splines, the tensor-product formulation of higher dimensional SIAC filters restricts the definition of this class of B-spline-based filters to separable structures. Hence, we resort to a geometrical approach that enables us to define SIAC for nonseparable structures, in particular on a hexagonal lattice. A similar idea can be used for higher dimensional cases and other lattice structures. Without loss of generality, we demonstrate the geometric approach for the first order SIAC. Higher order SIAC filters can be constructed similarly.

Considering a 2D evaluation point $\mathbf{x} = (x, y)$, the two-dimensional SIAC kernel defined in Eq. 16 can be considered as centering nine B-splines kernels as follows: one kernel on the evaluation point and eight more B-splines on direct and indirect neighbors of the evaluation point on a Cartesian lattice. Figure 4 (a) demonstrates the B-spline centers constituting the first order of a two-dimensional SIAC kernel. Using this geometrical approach, we can now rewrite the two-dimensional SIAC kernel as

$$K^{(r+1,k+1)}(x) = \sum_{\substack{\boldsymbol{\gamma} \\ -k \le \gamma_1, \gamma_2 \le k}} c_{\boldsymbol{\gamma}} b_{k+1}(\mathbf{x} - \boldsymbol{\gamma}\mathbf{C}^T), \tag{17}$$

where $\mathbf{C}$ denotes the matrix including the lattice direction of a Cartesian lattice presented in Eq. 9, and $r = 8 \times (2k-1)$. It is easy to show the equivalence between Eq. 17 and Eq. 16. One can also show that the kernel coefficients, $c_{\boldsymbol{\gamma}}$, can still be computed as $c_{\gamma_1} c_{\gamma_2}$ which, coincide with the kernel coefficients in Eq. 16. The kernel coefficients can also be computed directly (i.e., without using the tensor product concept) by considering the multidimensional polynomial reproduction property

$$K^{(r+1,k+1)}(\mathbf{x}) * \mathbf{x}^{\mathbf{p}} = \mathbf{x}^{\mathbf{p}}, \qquad 0 \le |\mathbf{p}| \le 4k, \tag{18}$$

where $|\mathbf{p}| = p_1 + p_2$ is a multi-index and represents the total degree of a two-dimensional polynomial. Note that unlike the univariate case, the total number of kernel coefficients in the two-dimensional case does not match the number of (total) degree polynomials that the kernel requires to reproduce. For instance, the first-order two-dimensional SIAC kernel has nine kernel coefficients (see Figure 4 (a)) whereas the kernel is required to reproduce polynomials up to (total) degree four. There exist 15 such polynomials: $1, x, y, x^2, y^2, xy, \cdots, x^4, y^4$. However, considering the symmetry, we need to consider only nine polynomials: 1, x, x^2, xy, x^3, x^2y, x^3y, x^2y^2, x^4. The system can be further reduced to three equations by considering the symmetry of the kernel. Therefore, we obtain a square system of size 3×3 to determine the three distinct coefficients.

3.3 HSIAC Filter Construction

We can now use the same geometrical approach to define the hexagonal SIAC kernel (HSIAC) on a hexagonal lattice; that is, we can use the direct and indirect neighbors of an evaluation point on a hexagonal lattice to position translations of hex spline kernels. Figure 4 (b) demonstrates the position of centers of hex splines for the first-order HSIAC. Note that even though the hexagonal lattice requires only two lattice directions for its definition, navigation of the points on this lattice can be better represented using three directions (demonstrated in Figure 4 (b))

$$\mathbf{H}_3 = \begin{bmatrix} u_1, & u_2, & u_3 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & -1 \end{bmatrix}. \tag{19}$$

The first two columns in the relation above represent the lattice directions of the hexagonal lattice and the third column is $u_3 = -u_1 - u_2$. We can now formally define HSIAC as

$$K_{\eta}^{(r+1,k+1)}(x) = \sum_{\substack{\boldsymbol{\gamma}=(\gamma_1,\gamma_2,\gamma_3) \\ -k \leq \gamma_1,\gamma_2,\gamma_3 \leq k}} c_{\boldsymbol{\gamma}} \eta_{k+1}(\mathbf{x} - \boldsymbol{\gamma}\mathbf{H}_3^T), \tag{20}$$

where η_{k+1} denotes the $(k+1)^{th}$ order hex spline, and $r = 12 \times (2k-1)$. Similar to the previous case, the polynomial reproduction property can be used to find the kernel coefficients

$$K_{\nu}^{(r+1,k+1)}(\mathbf{x}) * \mathbf{x}^{\mathbf{p}} = \mathbf{x}^{\mathbf{p}}, \qquad 0 \leq |\mathbf{p}| \leq 4k. \tag{21}$$

Note that in comparison to Eq. 16, HSIAC requires a larger number of hex splines (see Figure 4 (b)). Similar to the Cartesian case, let us consider the first-order two-dimensional HSIAC kernel. This kernel has 13 kernel coefficients (see Figure 4 (a)) and is required to reproduce polynomials up to (total) degree four. Similar to the Cartesian case, using the symmetry of the polynomials and the fact that the number of *unique* coefficients stays the same, one ends up with a square system of size 3×3 to determine the three distinct coefficients.

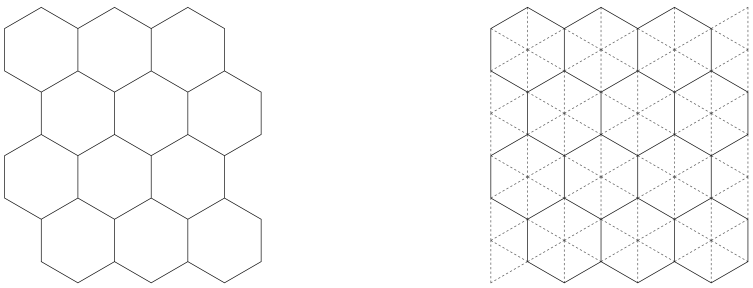
The squares and circles in Figure 4 (b) demonstrate the equivalent kernel coefficients for the first order of both kernels as well as the distinction between direct and indirect neighbors of an evaluation point on each lattice, respectively. The geometric construction based on direct and indirect neighbors as proposed in this section can be used to define higher dimensional nonseparable SIAC filters using Voronoi splines [26,25].

In the next section, we provide our main theoretical result that proves the superconvergence of HSIAC on structured triangular meshes constructed using a hexagonal lattice.

3.4 HSIAC filter Superconvergence

In this section, we prove that the HSIAC kernel has superconvergence properties for postprocessing of a DG solution for a linear hyperbolic equation using an upwind flux and periodic boundary conditions on structured triangular meshes based on a hexagonal lattice.

There are two main components to the proof: the approximation power of the kernel in terms of polynomial reproduction and superconvergence of the DG solution. The mesh structure has a significant impact on both factors. Note that the spline space formed by the translations of the constituent spline kernel is the basis of the SIAC kernel polynomial reproduction properties. Therefore, we can prove the superconvergence of hexagonal SIAC kernel only on a mesh structure that is consistent with a hexagonal lattice.



(a) (b)

Fig. 5 (a) A hexagonal mesh structure, (b) a triangular mesh constructed by subdividing each hexagonal in (a) into six triangles.

A natural choice for the mesh structure when using the HSIAC kernel for postprocessing of a DG solution is a uniform hexagonal mesh where each element is constructed using a linear combination of column vectors of $\mathbf{H}$. A hexagonal mesh of this form is demonstrated in Figure 5 (a). A hexagonal

mesh requires the DG solution to be computed on six-sided elements (i.e., the hexagons), which is unconventional. Even though the DG solution on triangular and quadrilateral meshes has been well studied, generalization of DG solutions on generic polygons has only recently received some attention [7,8], and the proof of superconvergence of the DG solution for generic polygons is yet to be developed. Therefore, we consider the subdivisions of the hexagonal mesh into structured triangles. An example of subdivision of a hexagonal mesh into a triangular mesh is shown in Figure 5 (b). This type of structured triangular mesh has consistent geometry with a hexagonal lattice. In addition, the superconvergence of the DG solution has been well studied for triangular meshes [2,3,19]. Other choices of subdivision of a hexagonal mesh will be discussed further in Section 4.

The following theorem provides the proof for superconvergence of hexagonal SIAC on a structured triangular mesh that is constructed using a hexagonal lattice:

Theorem 1 *Let $u_h(x)$ denote the DG solution of order $k+1$ to the true solution $u(x) \in H^{2k+1}$ (Sobolev space), which solves a linear hyperbolic equation (using an upwind flux and periodic boundary conditions) on a structured triangular mesh constructed using a hexagonal lattice. Let $K_\eta^{(r+1,k+1)}$ denote the HSIAC kernel of the form introduced in Eq. 20. For $n \geq 1$, and sufficiently smooth $u(x)$, we have the following relation for the approximation error bound of the postprocessing of $u_h(x)$ using $K_\eta^{(r+1,k+1)}$ at $T > 0$:*

$$\|u(x) - (K_{\eta,h}^{(r+1,k+1)} * u_h)(x)\|_{0,\Omega} \leq Ch^{2k+1}, \tag{22}$$

where $r = 12 \times (2k-1)$.

Proof : The proof of superconvergence for HSIAC follows the proof by Cockburn et al. [9]. That is, we first decompose the left-hand side of Eq. 22:

$$\begin{aligned}\|u(x) - (K_{\eta,h}^{(r+1,k+1)} * u_h)(x)\|_{0,\Omega} \\ \leq \| \underbrace{u(x) - (K_{\eta,h}^{(r+1,k+1)} * u)(x)}_{\text{filter error}} \|_{0,\Omega} + \| \underbrace{(K_{\eta,h}^{(r+1,k+1)} * (u - u_h))(x)}_{\text{approximation error}} \|_{0,\Omega}.\end{aligned} \tag{23}$$

As shown before in [9], the first term is entirely dependent on the polynomial reproduction property. Hence, this term can be bounded by $C_1 h^{4k+1}$ (see Eq. 21) where C_1 is a constant value independent of h [28]. However, we are limited by the error bound of the second term.

The error bound for the second term (i.e., approximation error) in Eq. (23) depends on the order of the constituting hex spline and its derivatives. We use the negative-order norm analysis proposed by Cockburn et al. [9] and shown in [6,14,24] to compute the error bound for the second term

$$\begin{aligned}\|(K_{\eta,h}^{(r+1,k+1)} * (u - u_h))(x)\|_{0,\Omega} \\ \leq C_2 \sum_{|\boldsymbol{\alpha}| \leq k+1} \|D^{\boldsymbol{\alpha}}\big(K_{\eta,h}^{r+1,k+1} * (u - u_h)\big)\|_{-(k+1),\Omega},\end{aligned} \tag{24}$$

where $D^{\boldsymbol{\alpha}}$ is a multidimensional derivative where

$$D^{\boldsymbol{\alpha}} = D_x^{\alpha_1} D_y^{\alpha_2} \text{ and } |\boldsymbol{\alpha}| = \alpha_1 + \alpha_2. \tag{25}$$

As discussed in Section 2 and the appendix, unlike B-splines, the derivatives of a hex spline cannot be written in terms of lower order hex splines. However, given the equivalent properties of both kernels [34,27], the derivatives of a hex spline up to degree $k+1$ can still be computed exactly using a finite difference operator of the same degree. Hence, we have

$$\begin{aligned}\sum_{|\boldsymbol{\alpha}|\leq k+1} \|D^{\boldsymbol{\alpha}}\big(K_{\eta,h}^{r+1,k+1} * (u-u_h)\big)\|_{-(k+1),\Omega} \\ = \sum_{|\boldsymbol{\alpha}|\leq k+1} \|{}^{\boldsymbol{\alpha}}K_{\eta,h}^{r+1,k+1} * \partial_h^{\boldsymbol{\alpha}}\big(u-u_h\big)\|_{-(k+1),\Omega},\end{aligned} \tag{26}$$

where ${}^{\boldsymbol{\alpha}}K_{\eta,h}^{r+1,k+1}$ refers to the $\boldsymbol{\alpha}^{th}$ derivative of the hexagonal SIAC kernel. Furthermore, we can write

$$\begin{aligned}\sum_{|\boldsymbol{\alpha}|\leq k+1} \|{}^{\boldsymbol{\alpha}}K_{\eta,h}^{r+1,k+1} * \partial_h^{\boldsymbol{\alpha}}\big(u-u_h\big)\|_{-(k+1),\Omega} \\ \leq \sum_{|\boldsymbol{\alpha}|\leq k+1} \big(\|{}^{\boldsymbol{\alpha}}K_{\eta,h}^{r+1,k+1}\|_{L^1(\mathbb{R}^d)} \|\partial_h^{\boldsymbol{\alpha}}\big(u-u_h\big)\|_{-(k+1),\Omega}\big).\end{aligned} \tag{27}$$

The first term on the right-hand side is a direct result of hex splines being piecewise polynomials of (total) degree $2k$. Therefore, we can replace the first term by a constant C_3, and write

$$\|(K_{\eta,h}^{(r+1,k+1)} * (u-u_h))(x)\|_{0,\Omega} \leq C_2 C_3 \sum_{|\boldsymbol{\alpha}|\leq k+1} \|\partial_h^{\boldsymbol{\alpha}}\big(u-u_h\big)\|_{-(k+1),\Omega}, \tag{28}$$

which is the same as in [9]. This completes the proof.

4 Numerical Results

In this section, we provide numerical results to show the superconvergence of HSIAC filter in practice and compare it with (Cartesian) SIAC filtering. For our first set of experiments, we use the DG projection of the initial condition for the following linear transport equation

$$\frac{\partial u}{\partial t} + \mathbf{a}\nabla\cdot u = 0, \quad u(x,0) = \sin(2\pi(x + \frac{y}{0.989})), \quad x \in [0,1.0]\times[0,0.989], \tag{29}$$

where $\mathbf{a} = 1.0\mathbf{i} + 0.989\mathbf{j}$. In the second set of experiments, we perform postprocessing of the DG solution of the same equation at the final time $T = 1.0$ (one periodic cycle). We used Nektar++ [1] in order to compute the DG projection and the DG solution on the structured triangular mesh introduced in the previous section (see Figure 5 (b)). Note that in order to have periodic boundary

conditions, we had to consider a jagged boundary as depicted in Figure 5 (b). The postprocessing using SIAC (or HSIAC) kernel has been carried out using the quadrature approximation proposed in [22]. The approximation error for all the examples has been computed as the difference between the postprocessing of the DG projection (or the DG solution) and the true solution using Ł$_2$ and L_∞ norms. The kernel scaling for (Cartesian) SIAC considered to be the largest element size as proposed in [20,21] for structured triangular meshes and the kernel scaling for the HSIAC filter is based on the edge length of the underlying hexagonal lattice.

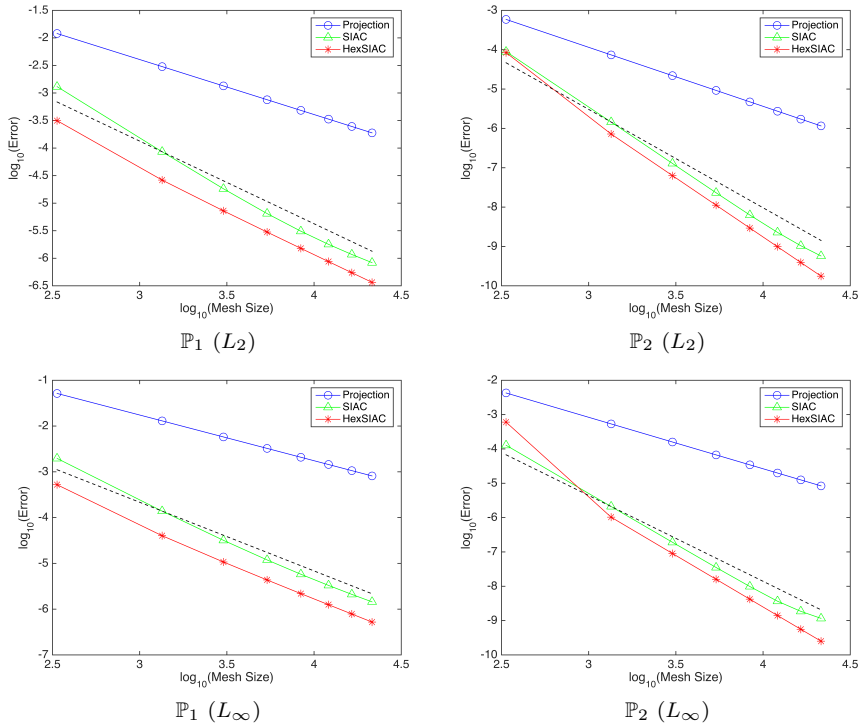


Fig. 6 A base−10 logarithmic-scale plot of the approximation error versus the number of elements to demonstrate the order of convergence of the postprocessor compared to the order of convergence of the DG projection. The dashed line represents $2k+1$ (i.e., the expected order of convergence).

Table 1 presents the L_2 and L_∞ norms of the approximation error of the postprocessing of the DG projection using both SIAC and HSIAC filters. In this case, the HSIAC postprocessing has lower error values compared to the SIAC filter. Figure 6 also presents the order of convergence of both filters for DG projection. The dashed line represents the expected convergence rate (i.e., $2k+1$). Both SIAC and HSIAC postprocessing follows the dashed line as the

Table 1 The approximation error for postprocessing of the DG projection. For the SIAC kernel, the largest element size was used for kernel scaling.

	num. of elements	DG projection		SIAC		HSIAC	
		L_2	L_∞	L_2	L_∞	L_2	L_∞
$\mathbb{P}_1$	336	$1.20e-02$	$5.16e-02$	$1.30e-03$	$1.96e-03$	$3.15e-04$	$5.22e-04$
	3024	$3.02e-03$	$1.30e-02$	$8.59e-05$	$1.39e-04$	$2.63e-05$	$4.03e-05$
	3024	$1.34e-03$	$5.78e-03$	$1.83e-05$	$3.19e-05$	$7.26e-06$	$1.07e-05$
	5376	$7.55e-04$	$3.25e-03$	$6.48e-06$	$1.19e-05$	$2.99e-06$	$4.34e-06$
	8400	$4.83e-04$	$2.08e-03$	$3.10e-06$	$5.79e-06$	$1.51e-06$	$2.18e-06$
	12096	$3.36e-04$	$1.45e-03$	$1.79e-06$	$3.31e-06$	$8.72e-07$	$1.25e-06$
	16464	$2.47e-04$	$1.06e-03$	$1.17e-06$	$2.10e-06$	$5.47e-07$	$7.83e-07$
	21504	$1.89e-04$	$8.13e-04$	$8.33e-07$	$1.44e-06$	$3.66e-07$	$5.23e-07$
$\mathbb{P}_2$	336	$5.88e-04$	$4.27e-03$	$8.95e-05$	$1.28e-04$	$8.43e-05$	$6.02e-04$
	3024	$7.38e-05$	$5.36e-04$	$1.45e-06$	$2.11e-06$	$7.22e-07$	$1.03e-06$
	3024	$2.19e-05$	$1.59e-04$	$1.29e-07$	$1.91e-07$	$6.31e-08$	$9.01e-08$
	5376	$9.24e-06$	$6.71e-05$	$2.32e-08$	$3.53e-08$	$1.12e-08$	$1.60e-08$
	8400	$4.73e-06$	$3.44e-05$	$6.29e-09$	$9.84e-09$	$2.94e-09$	$4.20e-09$
	12096	$2.74e-06$	$1.99e-05$	$2.28e-09$	$3.68e-09$	$9.86e-10$	$1.41e-09$
	16464	$1.72e-06$	$1.25e-05$	$1.04e-09$	$1.86e-09$	$3.92e-10$	$5.59e-10$
	21504	$1.16e-06$	$8.40e-06$	$5.73e-10$	$1.16e-09$	$1.76e-10$	$2.52e-10$

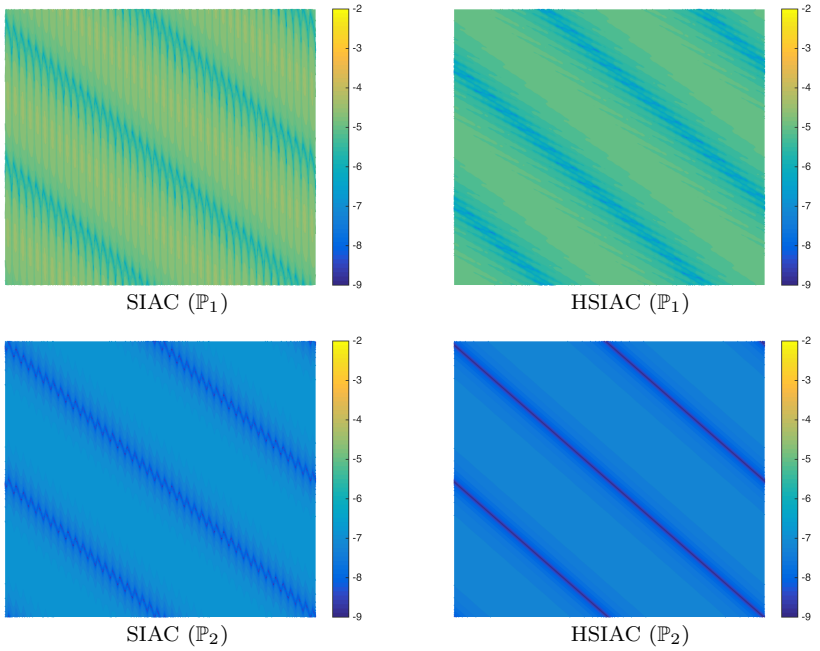


Fig. 7 The contour plot of base−10 logarithmic-scale L_2 approximation error of SIAC and HSIAC for postprocessing of the DG projection of Eq. 29 using the mesh structure presented in Figure 5 (b) (number of mesh elements: 3024). The jagged part of the meshes is trimmed and the error is computed at 90k uniformly spaced points.

mesh size increases. In some cases, the low resolution meshes are not resolving the supercovergence property numerically. Figure 7 shows the contour plot

of the L_2 approximation error over the whole domain in base−10 logarithmic scale where 90k uniform points have been sampled to produce the error contour plots and the jagged part of the mesh has been trimmed before sampling.

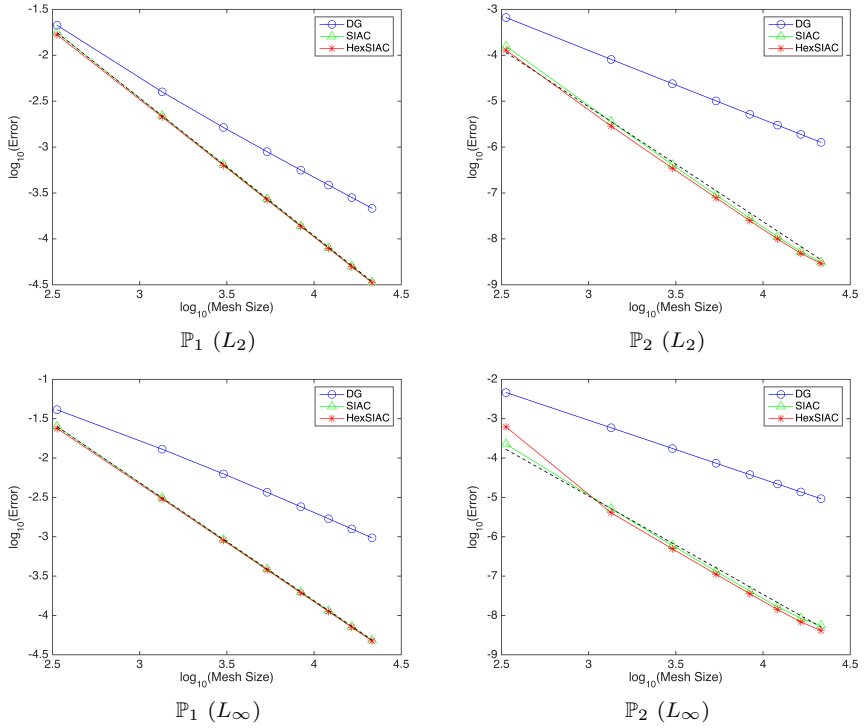


Fig. 8 A base−10 logarithmic-scale plot of the approximation error versus the number of elements to demonstrate the order of convergence of the postprocessor compared to the order of convergence of the DG solution. The dashed line represents $2k+1$ (i.e., the expected order of convergence).

Similarly, Table 2 and Figure 8 demonstrate the approximation error of the postprocessing of the DG solution of the advection equation presented in Eq. 29 at the final time $T = 1.0$. The approximation error in this case is higher compared to the DG projection counterparts. This behavior is expected and is the result of the addition of the advection errors such as the unwinding error and stepping error. Table 2 and Figure 9 also demonstrate that HSIAC in this case provides only marginal improvement compared to the SIAC kernel. This behavior suggests that even though HSIAC performs better than SIAC in improving the projection error, it was not able to resolve the unwinding error or stepping error better that the SIAC kernel. One potential cause might be the fact that HSIAC has not been applied on a true hexagonal mesh. However,

Table 2 The approximation error for postprocessing of the DG solution. For the SIAC kernel, the largest element size was used for kernel scaling.

	num. of elements	DG solution		SIAC		HSIAC	
		L_2	L_∞	L_2	L_∞	L_2	L_∞
$\mathbb{P}_1$	336	$2.12e-02$	$4.10e-02$	$1.76e-02$	$2.51e-02$	$1.66e-02$	$2.37e-02$
	3024	$3.99e-03$	$1.29e-02$	$2.19e-03$	$3.13e-03$	$2.12e-03$	$3.02e-03$
	3024	$1.63e-03$	$6.28e-03$	$6.46e-04$	$9.21e-04$	$6.31e-04$	$8.98e-04$
	5376	$8.90e-04$	$3.67e-03$	$2.71e-04$	$3.87e-04$	$2.66e-04$	$3.79e-04$
	8400	$5.61e-04$	$2.40e-03$	$1.38e-04$	$1.97e-04$	$1.36e-04$	$1.94e-04$
	12096	$3.86e-04$	$1.69e-03$	$8.00e-05$	$1.14e-04$	$7.89e-05$	$1.12e-04$
	16464	$2.82e-04$	$1.25e-03$	$5.03e-05$	$7.20e-05$	$4.97e-05$	$7.07e-05$
	21504	$2.15e-04$	$9.70e-04$	$3.36e-05$	$4.82e-05$	$3.33e-05$	$4.73e-05$
$\mathbb{P}_2$	336	$6.62e-04$	$4.62e-03$	$1.57e-04$	$2.24e-04$	$1.28e-04$	$6.16e-04$
	3024	$8.13e-05$	$5.89e-04$	$3.59e-06$	$5.18e-06$	$2.86e-06$	$4.07e-06$
	3024	$2.40e-05$	$1.75e-04$	$4.11e-07$	$5.98e-07$	$3.46e-07$	$4.92e-07$
	5376	$1.01e-05$	$7.38e-05$	$9.02e-08$	$1.32e-07$	$7.84e-08$	$1.11e-07$
	8400	$5.18e-06$	$3.78e-05$	$2.82e-08$	$4.16e-08$	$2.50e-08$	$3.55e-08$
	12096	$3.00e-06$	$2.19e-05$	$1.11e-08$	$1.66e-08$	$9.98e-09$	$1.42e-08$
	16464	$1.89e-06$	$1.38e-05$	$5.29e-09$	$8.51e-09$	$4.83e-09$	$6.87e-09$
	21504	$1.26e-06$	$9.24e-06$	$3.11e-09$	$5.56e-09$	$2.91e-09$	$4.14e-09$

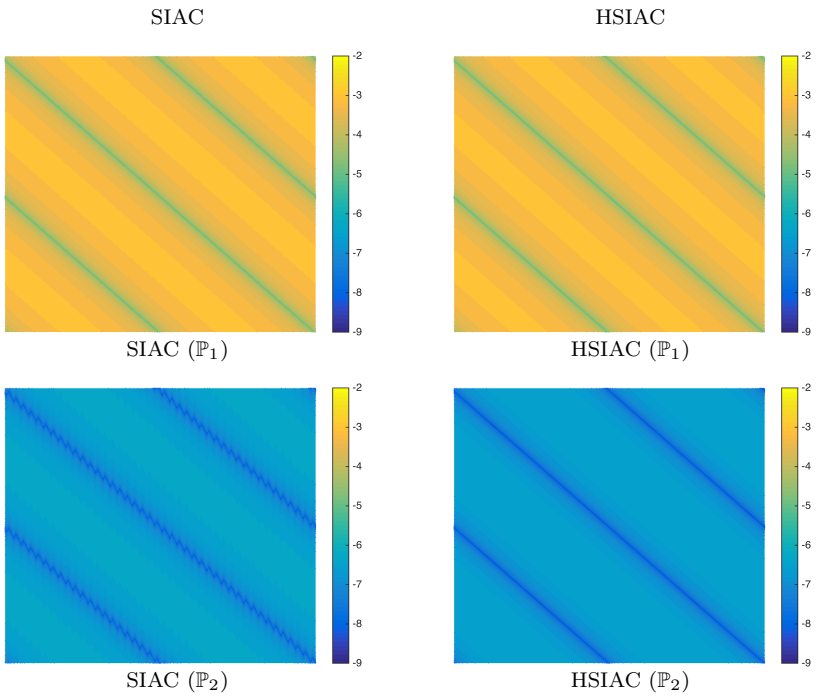


Fig. 9 The contour plot of base−10 logarithmic-scale L_2 approximation error of SIAC and HSIAC for postprocessing of the DG solution of Eq. 29 at the final time using the mesh structure presented in Figure 5 (b) (number of mesh elements: 3024). The jagged part of the meshes is trimmed and the error is computed at 90k uniformly spaced points.

further investigation of this behavior is beyond the scope of this manuscript and is considered as a future research question to be investigated.

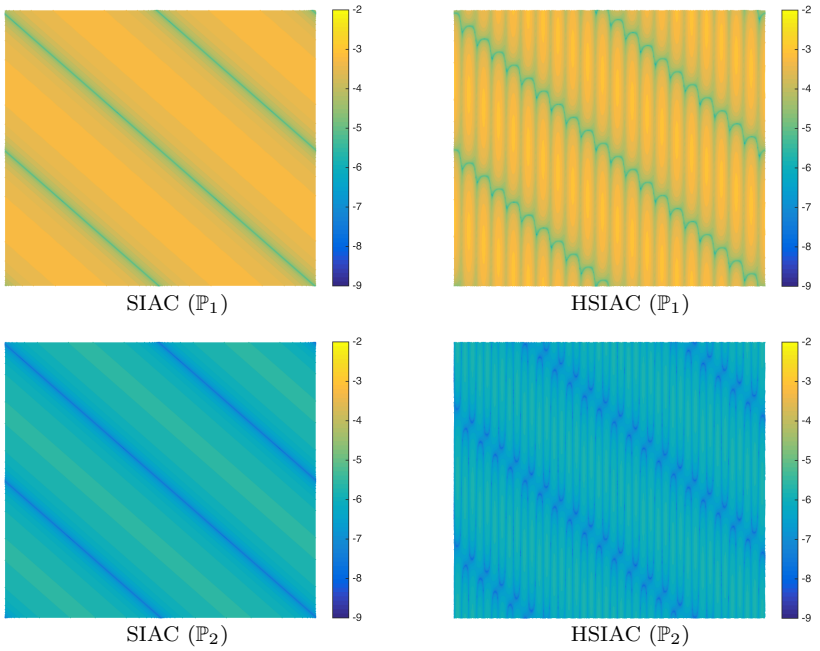


Fig. 10 The contour plot of base−10 logarithmic-scale L_2 approximation error of SIAC and HSIAC for postprocessing of the DG projection of Eq. 29 using the mesh structure presented in Figure 11 (number of mesh elements: 6048). The jagged part of the meshes is trimmed and the error is computed at 90k uniformly spaced points.

In addition to understanding the error reduction and superconvergence properties, it is interesting to study the effect of subdivision of a hexagonal mesh into a triangular mesh on the profile of the error (i.e., the patterns of jaggedness in contour plots). The mesh presented in Figure 5 (b) is the simplest way one can subdivide a hexagonal mesh into a triangular mesh. However, there are other options that can be considered. Figure 11 shows another example where the mesh presented in Figure 5 (b) has been subdivided further. Notice that compared to Figure 5 (b), Figure 11 allows one to define an axis-aligned rectangular replicating pattern. We used the new mesh to postprocess the DG projection and solution using similar setup as presented at the beginning of this section.

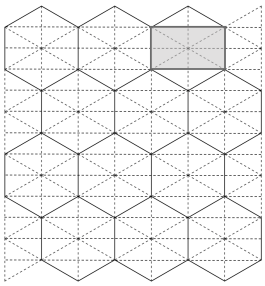


Fig. 11 A triangular mesh constructed by subdividing the mesh in Figure 5 (b).

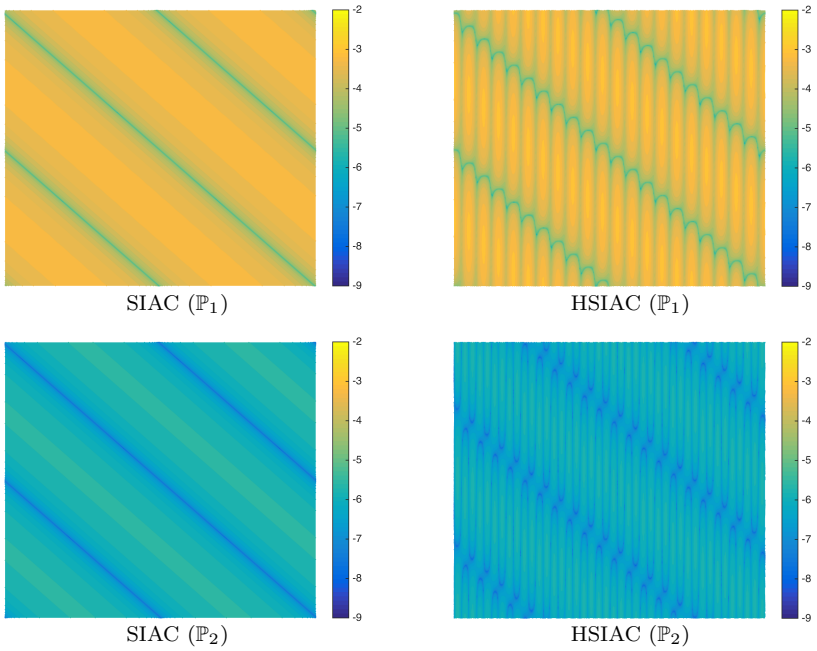


Fig. 12 The contour plot of base−10 logarithmic-scale L_2 approximation error of SIAC and HSIAC for postprocessing of the DG solution of Eq. 29 using the mesh structure presented in Figure 11 (number of mesh elements: 6048). The jagged part of the meshes is trimmed and the error is computed at 90k uniformly spaced points.

As shown in [20,16], when a structured triangular mesh admits an axis-aligned replicating pattern, the characteristic length of the replicating pattern should be used for scaling of the SIAC kernel (instead of the largest element size)[2].

The contour plot of the base−10 logarithmic-scale L_2 approximation error of both kernels are presented in Figure 10 and 12. Notice that compared to Figure 7 and 9 where the SIAC kernel shows patterns of jaggedness in the vertical direction, the new contour plots are smooth. This behavior can be associated with the fact that the new mesh is better suited for SIAC kernel (i.e., the presence of a rectangular replicating pattern). On the other hand, the HSIAC filtering contour plots show more jaggedness compared to Figure 7 and 9. It is important to note that the new mesh in less suitable for the HSIAC filter, as it provides (horizontal) element boundaries that do not fit the kernel structure. In addition to studying the profile of the error in the contour plots, we also performed numerical experiments confirming that the order of accuracy of SIAC and HSIAC stays similar for the new meshes.

[2] The scaling of the HSIAC filter in this case does not change since the underlying hexagonal mesh has not changed.

5 Conclusion

In this paper, we used a geometric approach to design a new multidimensional SIAC kernel called hexagonal SIAC (HSIAC) that goes beyond the conventional tensor-product assumption. HSIAC uses a nonseparable class of two-dimensional spline functions called hex splines as the underlying spline kernel. Hex splines have approximation and continuity properties equivalent to those of their tensor-product B-spline counterparts but they are nonseparable and more radially symmetric. The nonseparability of hex splines can be employed to prove the superconvergence property of HSIAC filtering for a specific class of structured triangular meshes (i.e., from subdivision of a hexagonal lattice). The introduction of this family of kernels and the geometric construction idea behind it will pave the way for the design of new SIAC kernels with proven superconvergence properties for more complicated geometries and DG solutions with more complicated features (e.g., DG solutions with shocks or discontinuities). An interesting line of future research is to investigate the utility of HSIAC filtering for postprocessing of DG solutions on generic polygons from the numerical and theoretical points of view. The more radially symmetric nature of HSIAC kernel should prove to be more appropriate for streamlining applications.

6 Appendix

In this appendix, we provide some details for the box-spline-based formulation of hex splines. This formulation provides a compact and efficient mechanism to evaluate and study the properties of higher order hex splines. We start the discussion with a brief introduction to box splines as one of the most generic spline functions.

A box spline $\mathbb{R}^d$ is defined by a set of n vectors $\xi_1, \cdots, \xi_n \in \mathbb{R}^d$. Geometrically, a box spline is the shadow of a hypercube in $\mathbb{R}^n$ that has been projected onto $\mathbb{R}^d$ where $n > d$. Each ξ denotes the shadow of an edge of the hypercube in $\mathbb{R}^n$. The simplest box spline in $\mathbb{R}^d$ corresponds to the case where $d = n$. In this case, a box spline is defined as the (normalized) indicator function of the parallelepiped formed by the d vectors in $\mathbb{R}^d$.

$$M_{\Xi}(\mathbf{x}) = \begin{cases} \frac{1}{|\det(\Xi)|} & \mathbf{x} = \sum_{i=1}^{d} t_i \xi_i \text{ for } 0 \leq t_i \leq 1 \\ 0 & \text{otherwise} \end{cases}, \tag{30}$$

where $\Xi := [\xi_1, \cdots, \xi_n]$ denotes the matrix of directions. For $n > d$, a box spline can be defined recursively as

$$M_{\Xi \cup \xi_k}(\mathbf{x}) = \int_0^1 M_{\Xi}(\mathbf{x} - t\boldsymbol{\xi}) dt, \tag{31}$$

where $\Xi \cup \xi_k$ denotes the addition of ξ_k to the matrix of direction $\Xi \cup \xi_k := [\xi_1, \cdots, \xi_n, \xi_k]$. The convolution in the relation above can be considered as

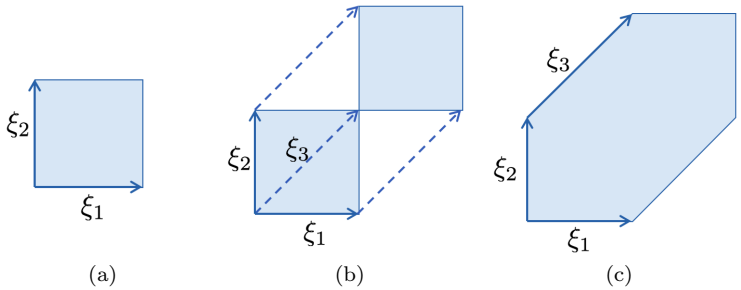


Fig. 13 The support of $M_{[\xi_1,\xi_2]}$ (a). The support of $M_{[\xi_1,\xi_2,\xi_3]}$ can be considered as smearing the support of $M_{[\xi_1,\xi_2]}$ along ξ_3 (b). The support of $M_{[\xi_1,\xi_2,\xi_3]}$ can be considered as the Minkowski sum of ξ_1, ξ_2, and ξ_3 (c).

smearing the original box spline M_Ξ along the new direction ξ_k. Consequently, the support of the new box spline $M_{\Xi\cup\xi_k}$ can be considered as the Minkowski sum of the vectors in $\Xi \cup \xi_k$. Figure 13 demonstrates the idea behind the convolution in Eq. 31 graphically. Similarly, the convolution of two box splines together results in a new box spline

$$M_{\Xi_1} * M_{\Xi_2} = M_{[\Xi_1\cup\Xi_2]}. \tag{32}$$

Note that the vectors can appear with some multiplicity in the matrix of directions of a box spline.

The matrix of directions fully specifies all the properties of M_Ξ but the order of the vectors in Ξ does not have any effect on its properties. For instance, let κ denote the minimum number of directions whose removal from Ξ makes the remaining directions not span $\mathbb{R}^d$. The value of κ specifies the order of continuity of a box spline. That is, a box spline formed by Ξ is $C^{\kappa-2}$ continuous [5]. A box spline is a piecewise polynomial function and can be efficiently evaluated using the recursive definition in Eq. 30 or using a Fourier transform analysis [5]. It is worth mentioning that the derivatives of M_Ξ in any direction $Z \in \Xi$ can be exactly evaluated using a differencing operator

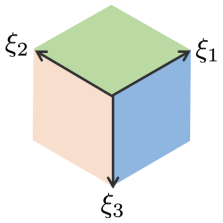


Fig. 14 First-order hex spline can be decomposed into three box splines.

$$D_Z M_\Xi = \partial_Z M_{\Xi\setminus Z} \text{ for } Z \subseteq \Xi, \tag{33}$$

where $\Xi\setminus Z$ denotes removal of direction Z from the matrix of direction and ∂_Z denotes a backward difference operator in the direction of Z. In general, the derivative of M_Ξ in any arbitrary direction can still be rewritten in terms of (a linear combination of) the derivatives in the direction Z where $Z \in \Xi$ [5,

Lemma 34]. The interested reader can consult [5] for a thorough discussion of box splines and it properties.

Both B-splines and hex splines can be written in terms of box splines. For example, the first-order hex spline can be written and evaluated as a summation of the indicator function of three parallelepipeds that constitute the Voronoi cell of the hexagonal lattice (i.e., the hexagon presented in Figure 1 (b)). Each parallelepiped can be represented using a box spline

$$\eta_1 = \frac{1}{3}\big(M_{[\xi_1,\xi_2]} + M_{[\xi_1,\xi_3]} + M_{[\xi_2,\xi_3]}\big), \tag{34}$$

where ξ_i represents the i^{th} column of matrix $\mathbf{H}_3$. Figure 14 demonstrates the decomposition of the first-order hex spline into three box splines. We can now use the box spline convolution rule in order to write the higher order hex splines as

$$\eta_n = \big(\eta_1\big)^{*n} = \frac{1}{3^n}\big(M_{[\xi_1,\xi_2]} + M_{[\xi_1,\xi_3]} + M_{[\xi_2,\xi_3]}\big)^{*n}, \tag{35}$$

where $(\eta_1)^{*n}$ is a short-hand notation for n-times self-convolution.

For example, the second-order hex spline can be written in terms of six box splines

$$\begin{aligned}\eta_2 &= \frac{1}{3^2}\big(M_{[\xi_1,\xi_2]} + M_{[\xi_1,\xi_3]} + M_{[\xi_2,\xi_3]}\big)^{*2} \\ &= \frac{1}{9}\big(M_{[\xi_1,\xi_1,\xi_2,\xi_2]} + M_{[\xi_1,\xi_1,\xi_3,\xi_3]} + M_{[\xi_2,\xi_2,\xi_3,\xi_3]}\big) \\ &+ \frac{2}{9}\big(M_{[\xi_1,\xi_1,\xi_2,\xi_3]} + M_{[\xi_1,\xi_2,\xi_2,\xi_3]} + M_{[\xi_1,\xi_2,\xi_3,\xi_3]}\big).\end{aligned} \tag{36}$$

Not unlike B-splines, the derivatives of a hex spline cannot be written in terms of lower order hex splines. However, the derivatives of hex splines can be written compactly in terms of the derivative of its constituent box splines (see Eq. 33), and be exactly evaluated using difference operators [5, Lemma 34].

Acknowledgements This work was sponsored in part by the Air Force Office of Scientific Research (AFOSR), Computational Mathematics Program (program Manager: Dr. Jean-Luc Cambier), under grant number FA9550-12-1-0428 (first and fourth author), FA8655-13-1-3017 (third author). The second and fourth authors are sponsored in part by the Army Research Office (program manager: Dr. Mike Coyle) under grant number W911NF-15-1-0222. In addition, the authors would like to acknowledge the anonymous reviewers for providing comments that improved the manuscript.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Nektar++ (2016). http://www.nektar.info

2. Adjerid, S., Baccouch, M.: The discontinuous Galerkin method for two-dimensional hyperbolic problems. part I: superconvergence error analysis. Journal of Scientific Computing **33**(1), 75–113 (2007)
3. Adjerid, S., Baccouch, M.: The discontinuous Galerkin method for two-dimensional hyperbolic problems part II: A posteriori error estimation. Journal of Scientific Computing **38**(1), 15–49 (2009)
4. Archibald, R., Gelb, A., Gottlieb, S., Ryan, J.: One-sided post-processing for the discontinuous Galerkin method using ENO type stencil choosing and the local edge detection method. Journal of Scientific Computing **28**(2-3), 167–190 (2006)
5. de Boor, C., Höllig, K., Riemenschneider, S.: Box Splines. Springer-Verlag New York, Inc., New York, NY, USA (1993)
6. Bramble, J.H., Schatz, A.H.: Higher order local accuracy by averaging in the finite element method. Mathematics of Computation **31**(137), 94–111 (1977)
7. Cangiani, A., Dong, Z., Georgoulis, E.H., Houston, P.: hp-version discontinuous Galerkin methods for advection-diffusion-reaction problems on polytopic meshes. ESAIM: Mathematical Modelling and Numerical Analysis **50**(3), 699–725 (2016)
8. Cockburn, B., Fu, G., Sayas, F.: Superconvergence by M-decompositions. part I: General theory for HDG methods for diffusion. Mathematics of Computation (2016)
9. 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)
10. Cohen, E., Riesenfeld, R.F., Elber, G.: Geometric modeling with splines - an introduction. A K Peters (2001)
11. Entezari, A.: Optimal sampling lattices and trivariate box splines. Ph.D. thesis, Simon Fraser University, Burnaby, BC, Canada, Canada (2007)
12. Ji, L., van Slingerland, Ryan, J.K., Vuik, K.: Superconvergent error estimates for a position-dependent Smoothness-Increasing Accuracy-Conserving filter for DG solutions. Mathematics of computation **83**, 2239–2262 (2014)
13. Ji, L., Van Slingerland, P., Ryan, J.K., Vuik, K.: Superconvergent error estimates for position-dependent Smoothness-Increasing Accuracy-Conserving (SIAC) post-processing of discontinuous Galerkin solutions. Mathematics of Computation **83**(289), 2239–2262 (2014)
14. Ji, L., Xu, Y., Ryan, J.K.: Accuracy enhancement of the linear convection-diffusion equation in multiple dimensions. Mathematics of computation **81**, 1929–1950 (2012)
15. Ji, L., Xu, Y., Ryan, J.K.: Negative-order norm estimates for nonlinear hyperbolic conservation laws. Journal of Scientific Computing **54**(2-3), 531–548 (2013)
16. 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. Journal of Scientific Computing **53**(1), 129–149 (2012)
17. Meng, X., Ryan, J.K.: Discontinuous Galerkin methods for nonlinear scalar hyperbolic conservation laws: divided difference estimates and accuracy enhancement. Numerische Mathematik pp. 1–47 (2016)
18. Meng, X., Ryan, J.K.: Divided difference estimates and accuracy enhancement of discontinuous Galerkin methods for nonlinear symmetric systems of hyperbolic conservation laws. IMA Journal of Numerical Analysis (2016)
19. Mirzaee, H.: Smoothness-Increasing Accuracy-Conserving filters (SIAC) for discontinuous Galerkin solutions. Ph.D. thesis, University of Utah, Salt Lake City, Utah, USA (2012)
20. Mirzaee, H., Ji, L., Ryan, J.K., Kirby, R.M.: Smoothness-Increasing Accuracy-Conserving (SIAC) postprocessing for discontinuous Galerkin solutions over structured triangular meshes. SIAM Journal of Numerical Analysis **49**(5), 1899–1920 (2011)
21. Mirzaee, H., King, J., Ryan, J.K., Kirby, R.M.: Smoothness-Increasing Accuracy-Conserving filters for discontinuous Galerkin solutions over unstructured triangular meshes. SIAM Journal on Scientific Computing **35**(1), A212–A230 (2013)
22. 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. Journal of Scientific Computing **45**(1-3), 447–470 (2010)

23. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Efficient implementation of Smoothness-Increasing Accuracy-Conserving (SIAC) filters for discontinuous Galerkin solutions. Journal of Scientific Computing **52**(1), 85–112 (2012)
24. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Smoothness-Increasing Accuracy-Conserving (SIAC) filters for discontinuous Galerkin solutions: Application to structured tetrahedral meshes. Journal of Scientific Computing **58**(3), 690–704 (2014)
25. Mirzargar, M.: A reconstruction framework for common sampling lattices. Ph.D. thesis, University of Florida, Gainesville, Florida, USA (2012)
26. Mirzargar, M., Entezari, A.: Voronoi splines. Signal Processing, IEEE Trans. on **58**(9), 4572–4582 (2010)
27. Mirzargar, M., Entezari, A.: Quasi interpolation with voronoi splines. IEEE Trans. on Visualization and Computer Graphics **17**(12), 1832–1841 (2011)
28. Mirzargar, M., Ryan, J.K., Kirby, R.M.: Smoothness-Increasing Accuracy-Conserving (SIAC) filtering and quasi-interpolation: a unified view. Journal of Scientific Computing **67**(1), 237–261 (2016)
29. Ryan, J.K., Cockburn, B.: Local derivative post-processing for the discontinuous Galerkin method. Journal of Computational Physics **228**(23), 8642–8664 (2009)
30. Ryan, J.K., Shu, C.W.: One-sided post-processing technique for the discontinuous Galerkin methods. Methods and Applications of Analysis **10**(2), 295–308 (2003)
31. Sánchez, J.D., Ryan, J.K., Mirzargar, M., Kirby, R.M.: Multi-dimensional filtering: Reducing the dimension through rotation. arXiv preprint arXiv:1610.02317 (2016)
32. Senechal, M.: Quasicrystals and geometry. CUP Archive (1996)
33. 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. on Visualization and Computer Graphics **14**(3), 680–692 (2008)
34. Van De Ville, D., Blu, T., Unser, M., Philips, W., Lemahieu, I., Van de Walle, R.: Hex-splines: a novel spline family for hexagonal lattices. IEEE Transactions on Image Processing **13**(6), 758–772 (2004)
35. Walfisch, D., Ryan, J.K., Kirby, R.M., Haimes, R.: One-sided Smoothness-Increasing Accuracy-Conserving filtering for enhanced streamline integration through discontinuous fields. Journal of Scientific Computing **38**(2), 164–184 (2009)