Capitalizing on Superconvergence for More Accurate Multi-Resolution Discontinuous Galerkin Methods

Jennifer K. Ryan¹

Received: 26 October 2020 / Revised: 8 January 2021 / Accepted: 16 January 2021 /
Published online: 9 July 2021
© The Author(s) 2021

Abstract

This article focuses on exploiting superconvergence to obtain more accurate multi-resolution analysis. Specifically, we concentrate on enhancing the quality of passing of information between scales by implementing the Smoothness-Increasing Accuracy-Conserving (SIAC) filtering combined with multi-wavelets. This allows for a more accurate approximation when passing information between meshes of different resolutions. Although this article presents the details of the SIAC filter using the standard discontinuous Galerkin method, these techniques are easily extendable to other types of data.

Keywords Multi-resolution analysis · Multi-wavelets · Discontinuous Galerkin · Smoothness-Increasing Accuracy-Conserving (SIAC) · Post-processing · Superconvergence · Accuracy enhancement

Mathematics Subject Classification 65M60

1 Introduction

Application-driven developments of complex fluid flows at multiple scales can benefit from deep understanding of the mathematical aspects of multi-resolution analysis (MRA) combined with existing knowledge of accuracy enhancement. This allows for more accurate approximations when passing information between meshes of different resolutions. In this article, we combine a natural technique for simulations at multiple scales, that of multi-wavelet MRA, and that of accuracy enhancing post-processing. Specifically, the techniques introduced in this article reduce the error in the approximation when moving information between coarse and fine grids (and vice versa). This is accomplished by exploiting the underlying properties of the numerical scheme such as the superconvergence paired with the multi-wavelet/MRA. This ability to reduce the errors when moving information from coarse data to finer data is demonstrated in Fig. 1. In this figure, the L^2 and L^∞ errors are presented in log-log scale for six different meshes. Specifically, the initial information is

✉ Jennifer K. Ryan
jkryan@mines.edu

¹ Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO 80401, USA

Fig. 1 L^2 and L^∞ errors on a series of six meshes. The initial information is given on a mesh consisting of twenty elements, and the mesh is refined five times so that the final mesh consists of 640 elements. Errors are given for a simple L^2 projection onto a finer grid as well as constructing an enhance reconstruction before mesh refinement. Left: $p = 3$ approximation; right: $p = 4$ approximation

given on a mesh consisting of twenty elements, and the mesh is refined five times so that the final mesh consists of 640 elements. Errors are given for a simple L^2 projection onto a finer grid as well as constructing an enhanced reconstruction before mesh refinement.

The MRA is a useful technique for expressing the discontinuous Galerkin (DG) approximation. It is based on a series of papers by Alpert, Belykin, and collaborators [1, 9], and was introduced in the DG framework in [1]. It has since been used by Cheng, Gao, and collaborators for sparse representations [7, 15, 25, 26, 31], by Müller and collaborators for adaptivity [3, 6, 13], and by Vuik and Ryan for discontinuity detection [29, 30]. It is based on the idea that a DG approximation over a mesh consisting of $2N$ elements, denoted $u_h^{2N}(x, t)$, and be written as a DG approximation over a mesh consisting of N elements, denoted $u_{2h}^N(x, t)$, together with the multi-wavelet information. Mathematically, the expression is the following:

$$u_h^{2N}(x, t) = u_{2h}^N(x, t) + \sum_{j=1}^N \sum_{k=0}^p d_{k,j}^N(t) \psi_{k,j}^N(x). \quad (1)$$

Here, $\psi_{k,j}^N(x)$ are called multi-wavelets and are generally taken to be those defined by Alpert [1]. They are piecewise polynomials of degree p , defined on an element. $d_{k,j}^N$ are the multi-wavelet coefficients that can be thought of as the detail coefficients between a mesh consisting of N elements and a mesh consisting of $2N$ elements, whereas the approximation on a mesh consisting of N elements is the average.

We can further improve the reconstruction by using the information from neighboring cells combined with the idea of superconvergence. This produces an alternative set of multi-wavelets that creates an improved reconstruction and reduces the error when passing the approximation from a coarse grid to a finer grid. It combines accuracy extraction techniques with the MRA. This combination creates an alternative set of multi-wavelets that allows for error reduction while allowing information to be passed from a fine scale to a coarse scale and vice versa.

The property of the superconvergence for Galerkin methods means that the solution is converging faster than the “optimal” convergence rate, which is tied to more accurate propagation of waves and allows for the physically relevant eigenvalue to dominate in long time

simulations [8]. For the DG, the optimal rate is $p + 1$. In [4], it was shown that a specially designed post-processor can extract the higher order information of the dispersion and dissipation error and create an approximation that converges with $2p + 1$ order accuracy. This post-processing filtering has been further developed as the Smoothness-Increasing Accuracy-Conserving (SIAC) filtering [10–12, 14, 18–20, 22, 24]. The ability to extract the higher order information will be exploited to allow for accurate transfer of information between scales.

The SIAC kernel has mostly been developed for DG methods. In [20], it was demonstrated that the approximation properties of the SIAC procedure actually lend themselves to describing a family of filtering kernels. Specifically, the smoothness increasing the property of the SIAC filtering is a direct result of using the 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 authors in [20] also studied variations of the symmetric SIAC kernel that still attain superconvergence properties while having different computational costs and smoothness properties. From the application point of view, the results can help practitioners design new kernels of interest with different design criteria.

It should be noted that combining these ideas leads to a variant of the Nyström reconstruction [21]. The Nyström reconstruction provides for more accurate reconstruction using convolution. Here, we present these ideas for DG approximations, but expect that they can be translated to other areas as well. The outline of this paper is the following. In Sect. 2, we present the DG approximation as well as how it can be written in terms of a multi-wavelet MRA. We also discuss the SIAC kernel that will be used to generate an alternative set of wavelets. In Sect. 3, we show how to combine these ideas to generate an alternative set of multi-wavelets and multi-wavelet coefficients. Finally, numerical examples are presented in Sect. 4.

2 Background

In this section, the discussion will be limited to the important components that serve as the mathematical formulation of the DG method, the MRA, and a specific filtering that both extracts accuracy and maintains accurate wave propagation properties. That is, a discussion of the SIAC filtering.

2.1 MRA

To illustrate the usefulness of the DG-MRA, an example of translating from a coarse scale to a finer scale is given in Fig. 2. In this figure, the approximation space for the coarsest mesh (level 0) consisting of one element is given in the top left. The approximation space for a mesh consisting of two elements (level 1) is given in the top right. We can relate these two approximations spaces by adding the multi-wavelet information where the multi-wavelets are given on the bottom row. In other words, the DG approximation space on level 1 can be rewritten as a direct sum of the scaling function basis of the coarse level 0 together with multi-wavelets. The coefficients multiplying the multi-wavelets contain the details needed to go from level 0 to level 1. Hence,

Fig. 2 The scaling functions on level 0 and level 1 for a $p = 4$ approximation. To produce the scaling functions on level 1, the $\psi_{k,1}^{(4)}(x)$, $k = 0, \dots, 4$ multi-wavelets are added to the scaling functions on level 0

$$u_h(x, t) = \underbrace{2^{-\frac{1}{2}} \sum_{j=0}^1 \sum_{\ell=0}^p u_j^{(\ell)}(t) \phi_{\ell j}^1(x)}_{\text{DG approximation on level 1}} = \underbrace{\sum_{\ell=0}^p s_{\ell 0}^0 \phi_{\ell j}^0(x)}_{\text{DG approximation on level 0}} + \underbrace{\sum_{\ell=0}^p d_{\ell j}^1(t) \psi_{\ell j}^1(x)}_{\text{finer details required for level 1}}, \quad (2)$$

where $d_{\ell j}^0$ are the detail coefficients.

The DG-MRA can be generalized for multiple levels. Here, $\psi_{\ell j}^m(x)$ are multi-wavelets that allow for translation from one level to another. This means that by subtracting details, the approximation on a coarser level can be obtained. This knowledge has already been established for the DG approximation basis itself [29, 30].

To provide more details of how the MRA works in the general case, consider a DG approximation for a given equation. To construct such an approximation, it is necessary to first break the domain up into N elements. In one dimension, denote elements as I_j , $j = 1, \dots, N$ such that the domain can be written as $\Omega = \bigcup_{j=1}^N I_j$. On each element, the DG approximation is formed of piecewise polynomials of degree less than or equal to p , denoted $\mathbb{P}^p(I_j)$. The entire DG approximation space is then given by

$$V_h^p = \{f : f \in \mathbb{P}^p(I_j), j = 1, \dots, N\}, \quad (3)$$

where h denotes the element size. However, this is directly related to a multi-resolution approximation. In Fig. 3, the DG space is given at level n with the element size $h = \frac{|\Omega|}{N}$ and $N = 2^n$. The approximation itself is represented as

$$u_h^N(x, t) = 2^{-\frac{n}{2}} \sum_{j=0}^{2^n-1} \sum_{\ell=0}^p u_j^{(\ell)}(t) \phi_{\ell j}^n(x), \quad (4)$$

where $\phi_{\ell j}^n(x) \in V_h^p$ are the basis functions in the approximation. The DG basis functions are also the scaling functions of the MRA on level n . The finer details are given by the multi-wavelet space, which is defined as the orthogonal complement of the scaling function space on a given level. For example, the DG approximation space on level n , denoted V_h^p , can be

Fig. 3 Diagram of the approximation levels in the MRA. Level n represents the level at which a DG solution is given. Using the relation from the MRA, there exists an automatic and precise relation to coarsened levels

rewritten as a direct sum of the scaling function basis on level $n - 1$ (with the element size $2h$) and its orthogonal complement:

$$V_h^p = V_{2h}^p \oplus W_{2h}^p. \quad (5)$$

The coefficients multiplying the wavelets in the space W_{2h}^p contain the details needed to go from level $n - 1$ to level n . Hence,

$$\begin{aligned} u_h^N(x, t) &= 2^{-\frac{n}{2}} \sum_{j=0}^{2^n-1} \sum_{\ell=0}^p u_j^{(\ell)}(t) \phi_{\ell j}^n(x) \\ &= \underbrace{\sum_{j=0}^{2^{n-1}-1} \sum_{\ell=0}^p s_{\ell n-1}^{n-1} \phi_{\ell j}^{n-1}(x)}_{\text{DG approximation on level } n-1} + \underbrace{\sum_{j=0}^{2^{n-1}-1} \sum_{\ell=0}^p d_{\ell j}^m(t) \psi_{\ell j}^n(x)}_{\text{finer details required for level } n}. \end{aligned} \quad (6)$$

This means that by subtracting details, the approximation on a coarser level can be obtained. For example, to get from level n to level 0, the coarsest level, the DG approximation at level 0 can be written as

$$\sum_{\ell=0}^p s_{\ell 0}^0(t) \phi_{\ell}^0(x) = u_h^N(x, t) - \sum_{m=0}^{n-1} \sum_{j=0}^{2^m-1} \sum_{\ell=0}^p d_{\ell j}^m(t) \psi_{\ell j}^m(x), \quad (7)$$

where $d_{\ell j}^m$ are the detail coefficients and $\psi_{\ell j}^m(x)$ are multi-wavelets that allow for translation from one level to another. A diagram of the different multi-resolution levels and their scaling function bases is given in Fig. 3. The extra information provided by the multi-wavelets allows for translation from one level to another.

2.2 SIAC Filtering

The symmetric SIAC kernel was originally introduced in [2, 4]. It was extended to include the superconvergent derivative information in [22, 27]. The utility for post-processing near the boundaries or shocks was addressed through one-sided SIAC kernels in [23, 28]. Recent

investigations into nonlinear equations show that it may be possible to improve on the filtered solution by modifying the kernel to account for the reduced regularity assumptions [12, 16, 17]. Traditionally, introducing these filters in engineering applications can be challenging since a tensor product filter grows in support size as the field dimension increases, becoming computationally expensive. Recently, this was overcome with the introduction of the line SIAC filter [5]. This one-dimensional filter for multi-dimensional data preserves the properties of traditional tensor product filtering, including smoothness and improvement in the convergence rate, at a greatly reduced computational cost.

The SIAC filter uses the continuous convolution

$$u_h^*(x, T) = K_H \star u_h = \int_{-\infty}^{\infty} K_H^{(r+1, \ell)}(x-y) u_h(y, T) dy, \quad x \in \Omega, \quad (8)$$

where u_h denotes the approximation at final time and the kernel (symmetric) is a linear combination of central B-splines,

$$K^{(r+1, \ell)}(\eta) = \sum_{\gamma=0}^r c_{\gamma} \psi^{(\ell)}\left(\eta - \left(\frac{r}{2} - \gamma\right)\right). \quad (9)$$

Here, γ denotes the B-splines centers and c_{γ} their weights. The kernel subindex H in (8) acts as a scaling factor, i.e., $K_H(x-y) = \frac{1}{H} K\left(\frac{x-y}{H}\right) = \frac{1}{H} K(\eta)$. To give an idea of the filter size, for uniform meshes, the usual scaling choice is $H = h$, where h denotes the element size. The superindexes $(r+1, \ell)$ indicate the number of B-splines employed to build the kernel $(r+1)$ and the spline order (ℓ) . The convolution kernel then has a support of size $(2r + \ell)H$ around the point being post-processed. For typical DG solutions, the superconvergent extracting SIAC kernel employs $r+1 = 2p+1$ B-Splines of order $\ell = p+1$ (p denotes the highest polynomial degree employed for the numerical approximation). In multiple dimensions, computational costs can be saved by applying this filter as a line integral [5].

In this article, we take $2p+1$ B-splines of order $\ell = 1$. Using higher order splines will lead to over-smoothing, a decrease in accuracy, and a decrease in accuracy with refinement. In this instance, the first three kernels are

$$\begin{aligned} K^{(3,1)}(x) &= -\frac{1}{24}(\mathbb{1}_{[-3/2, -1/2)} + \mathbb{1}_{[1/2, 3/2)}) + \frac{13}{12} \mathbb{1}_{[-1/2, 1/2)}, \\ K^{(5,1)}(x) &= \frac{3}{640}(\mathbb{1}_{[-5/2, -3/2)} + \mathbb{1}_{[3/2, 5/2)}) - \frac{29}{480}(\mathbb{1}_{[-3/2, -1/2)} + \mathbb{1}_{[1/2, 3/2)}) + \frac{1067}{960} \mathbb{1}_{[-1/2, 1/2)}, \\ K^{(7,1)}(x) &= -\frac{5}{7168}(\mathbb{1}_{[-7/2, -5/2)} + \mathbb{1}_{[5/2, 7/2)}) + \frac{159}{17920}(\mathbb{1}_{[-5/2, -3/2)} + \mathbb{1}_{[3/2, 5/2)}) \\ &\quad - \frac{7621}{107520}(\mathbb{1}_{[-3/2, -1/2)} + \mathbb{1}_{[1/2, 3/2)}) + \frac{30251}{26880} \mathbb{1}_{[-1/2, 1/2)}. \end{aligned}$$

These kernels will be utilized to obtain more accurate approximations when refining coarse mesh information.

3 SIAC-MRA

In this section, we introduce an enhanced reconstruction using the SIAC-MRA. This provides a basis of alternative multi-wavelets as well as alternative multi-wavelet coefficients. To do so, we introduce the following notation:

$$V_h^{p*} = \left\{ \sqrt{\frac{1}{2}} \cup \{ \chi_L^{(k)}(\xi_j) \}_{k=0}^p, j = 1, \dots, N \right\} \quad (10)$$

with $\xi_j = \frac{2}{\Delta x_j}(x - x_j)$ and

$$\chi_L^{(k)}(\eta) = \frac{1}{2} \int_{-1}^{\eta} \phi^{(k)}(\xi) d\xi.$$

This is the approximation space of the post-processed solution. There is a right counterpart to these functions defined as

$$\chi_R^{(k)}(\eta) = \frac{1}{2} \int_{\eta}^1 \phi^{(k)}(\xi) d\xi.$$

However, this expression can be further simplified if the basis functions are taken to be the orthonormal Legendre polynomials, $\phi^{(k)}(\xi) = \sqrt{k + \frac{1}{2}} P^k(\xi)$ in $[-1, 1]$. In this case, we can write $\chi_R^{(k)}(\eta) = -\chi_L^{(k)}(\eta) = -\frac{\sqrt{k+1/2}}{2(2k+1)} (P^{(k+1)}(\eta) - P^{(k-1)}(\eta))$ when $k > 0$. For $k = 0$, we have $\chi_L^{(0)}(\eta) = \frac{1}{2\sqrt{2}}(\eta + 1)$ and $\chi_R^{(0)}(\eta) = \frac{1}{2\sqrt{2}}(1 - \eta)$ which means that $\chi_R^{(0)}(\eta) = -\chi_L^{(0)}(\eta) + \sqrt{\frac{1}{2}}$. Plots of these functions are given in Fig. 4 (left).

To describe the procedure of the enhanced reconstruction, we begin with the DG approximation represented by orthonormal Legendre polynomials:

$$u_h^N(x, t) \Big|_{I_j} = \sum_{k=0}^p u_j^{(N,k)} \phi^{(k)}(\xi_j)$$

on a mesh consisting of N elements. Next, using a post-processor of $2p + 1$ B-Splines of order one we obtain the post-processed solution:

Fig. 4 Graphs of the left and right enhanced polynomials

$$\begin{aligned}
u_h^*(\bar{x}) &= c_{-p} \sum_{k=0}^p u_{j-(p+1)}^{(N,k)} \chi_R^{(k)}(\zeta + 1) \\
&\quad + \sum_{\gamma=-p}^{p-1} \sum_{k=0}^p u_{j+\gamma}^{(N,k)} \left(c_\gamma \chi_L^{(k)}(\zeta + 1) + c_{\gamma+1} \chi_R^{(k)}(\zeta + 1) \right) \\
&\quad + c_p \sum_{k=0}^p u_{j+p}^{(N,k)} \chi_L^{(k)}(\zeta + 1), \\
&= \sqrt{\frac{1}{2}} b_{j,R}^{(0)} + \sum_{k=0}^p b_j^{(k)} \chi_L^{(k)}(\zeta + 1),
\end{aligned} \tag{11}$$

where $\bar{x} = x_j + \frac{\Delta x}{2} \zeta$. The above is valid when $\zeta \in (-1, 0)$. Here,

$$b_{j,R}^{(0)} = \sum_{\gamma=-(p+1)}^{p-1} c_{\gamma+1} u_{j+\gamma}^{(N,k)}, \tag{12}$$

$$b_j^{(k)} = -c_{-p} u_{j-(p+1)}^{(N,k)} + \sum_{\gamma=-p}^{p-1} (c_\gamma - c_{\gamma+1}) u_{j+\gamma}^{(N,k)} + c_p u_{j+p}^{(N,k)}, \quad k = 0, \dots, p. \tag{13}$$

Similarly, if $\zeta \in (0, 1)$, the post-processed solution can be written as

$$u_h^*(\bar{x}) = \sqrt{\frac{1}{2}} \tilde{b}_{j,R}^{(0)} + \sum_{k=0}^p \tilde{b}_j^{(k)} \chi_L^{(k)}(\zeta - 1), \tag{14}$$

where

$$\begin{aligned}
\tilde{b}_{j,R}^{(0)} &= \sum_{\gamma=-p}^p c_\gamma u_{j+\gamma}^{(N,0)}, \\
\tilde{b}_j^{(k)} &= -c_p u_{j-p}^{(N,k)} + \sum_{\gamma=-(p-1)}^p (c_{\gamma-1} - c_\gamma) u_{j+\gamma}^{(N,k)} + c_p u_{j+p}^{(N,k)}.
\end{aligned}$$

We then need to obtain the approximation on the finer mesh, which can be expressed in terms of the following approximation space:

$$V_{h/2}^p = \{ \phi^{(k)}(\xi_j) \}_{k=0}^p, \quad j = 1, \dots, 2N \}. \tag{15}$$

This gives the coefficients for the refined approximation as

$$u_{2j-1}^{(2N,m)} = b_{j,R}^{(0)} \left(\sqrt{\frac{1}{2}}, \phi^{(m)} \right)_{(-1,1)} + \sum_{k=0}^p b_j^{(k)} \left(\chi_L^{(k)} \left(\frac{1}{2}(\xi + 1) \right), \phi^{(m)}(\xi) \right)_{(-1,1)}, \tag{16}$$

$$\begin{aligned}
u_{2j}^{(2N,m)} &= \tilde{b}_{j,R}^{(0)} \left(\sqrt{\frac{1}{2}}, \phi^{(m)} \right)_{(-1,1)} + \sum_{k=0}^p \tilde{b}_j^{(k)} \left(\chi_L^{(k)} \left(\frac{1}{2}(\xi - 1) \right), \phi^{(m)}(\xi) \right)_{(-1,1)} \\
j &= 1, \dots, N, \quad m = 0, \dots, p.
\end{aligned} \tag{17}$$

The inner products used above are defined in the following way:

$$\begin{aligned} \left(\chi_L^{(k)} \left(\frac{1}{2}(\xi + 1) \right), \phi^{(m)}(\xi) \right)_{(-1,1)} &= \int_{-1}^1 \chi_L^{(k)} \left(\frac{1}{2}(\xi + 1) \right) \phi^{(m)}(\xi) d\xi, \quad m = 0, \dots, p, \\ \left(\chi_L^{(k)} \left(\frac{1}{2}(\xi - 1) \right), \phi^{(m)}(\xi) \right)_{(-1,1)} &= \int_{-1}^1 \chi_L^{(k)} \left(\frac{1}{2}(\xi - 1) \right) \phi^{(m)}(\xi) d\xi, \quad m = 0, \dots, p. \end{aligned}$$

Hence, an alternative multi-wavelet coefficient expression of

$$\sum_{j=1}^{2N} \left(\sum_{k=0}^p \tilde{d}_j^{(k)} \psi^{(k)}(\xi_j) \right) = \mathcal{P}_{h/2} u_h^* - u_h^N \quad (18)$$

can be obtained, where $\mathcal{P}_{h/2}$ denotes the projection onto the piecewise polynomial approximation space (15) consisting of $2N$ elements, and $\tilde{d}_j^{(k)}$ are new multi-wavelets coefficients that may be inserted into (6).

Remark 1 (Errors) The enhanced reconstruction using the SIAC is of order $p + 2$. This allows for a reduced error constant when refining the mesh.

4 Numerical Results

In this section, we present the effectiveness of the enhanced reconstruction for a series of test problems involving both projections and time-evolution equations. We begin with an application to an analytical function and move on to a discontinuous function. Finally, we compare the time evolution of the usual MRA solution against the enhanced reconstruction.

4.1 Analytic Function

In this section, we consider

$$f(x) = \sin(2\pi x), \quad x \in [-1, 1].$$

The function is first projected onto a mesh consisting of twenty elements using a piecewise orthonormal Legendre basis. This approximation is then further projected onto a series of five finer meshes, where the number of elements are doubled for every refinement. The projection using the standard orthonormal Legendre polynomials is compared with enhanced reconstructions that exploit the SIAC information. We compare using the enhanced reconstruction for every refinement versus the enhanced reconstruction only for the first refinement. The log-log plots of the errors for $\mathbb{P}^1 - \mathbb{P}^4$ are given in Fig. 5 and pointwise errors in Fig. 6. The corresponding error tables are given in Table 1. Figure 5 and Table 1 show that even if the enhanced reconstruction is used just for the first refinement, it is beneficial. The errors can be further reduced using the enhanced reconstruction for each refinement thereafter. In other words, the magnitude of the error decreases with the first refinement, and the errors continue to decrease if enhancement is done with each refinement. This is because the affect of the SIAC is the greatest on the original approximation. It should be noted that using the SIAC and then refining always produces better errors than refinement by itself.

Fig. 5 The log-log plots of N vs. error for $f(x) = \sin(2\pi x)$ on a series of six meshes. Using an enhanced reconstruction reduces the errors, even if it is done only for the first refinement

4.2 Two-Dimensional Analytic Function

We repeat the experiment in Sect. 4.1 for two dimensions, that is, we consider the two-dimensional function

$$f(x, y) = \sin(2\pi(x + y)), \quad (x, y) \in [-1, 1]^2.$$

The log-log plots of the errors are given in Fig. 7 and the corresponding error table is given in Table 2. Similar to the one-dimensional case, we find that using the enhanced reconstruction is beneficial, even if it is only performed for the first refinement.

4.3 Discontinuous Function

In this section, we consider a simple box function,

Fig. 6 Pointwise error plots $f(x) = \sin(2\pi x)$ on the first three meshes. Using an enhanced reconstruction for every refinement reduces the errors and produces fewer oscillations in the errors

$$f(x) = \mathbb{1}_{(-0.45, 0.45)}, \quad x \in [-1, 1].$$

Note that the projections can achieve machine zero provided the discontinuities are at domain boundaries. For illustrative purposes, we only consider $p = 1, 2$. Plots of the function using a simple projection as well as enhanced reconstructions are given in Fig. 8. The log-log plots of N vs. error are given in Fig. 9. We see that it is still beneficial to use the enhanced reconstruction for the first refinement. However, continually using the enhanced reconstruction, while still better, may not be beneficial. The results can be seen in Table 3 and Fig. 10. The errors are calculated away from the discontinuities.

4.4 Time Evolution

In this example, we consider the linear transport equation

Table 1 L^2 and L^∞ errors for $f(x) = \sin(2\pi x)$ for a series of projections onto finer meshes. Left: projection only. Middle: enhanced only for the first projection. Right: enhanced each refinement. Using an enhanced reconstruction reduces the errors, even if it is done only for the first refinement

N	Projection error		Enhanced once		Enhanced each refinement	
	L^2 error	L^∞ error	L^2 error	L^∞ error	L^2 error	L^∞ error
$\mathbb{P}^1$						
20	1.465E-02	2.620E-02	1.465E-02	2.620E-02	1.465E-02	2.620E-02
40	1.465E-02	2.931E-02	3.981E-03	5.774E-03	3.981E-03	5.774E-03
80	1.465E-02	3.091E-02	3.981E-03	6.459E-03	1.563E-03	1.982E-03
160	1.465E-02	3.172E-02	3.981E-03	6.813E-03	1.265E-03	1.615E-03
320	1.465E-02	3.213E-02	3.981E-03	7.001E-03	1.244E-03	1.529E-03
$\mathbb{P}^2$						
20	7.782E-04	1.255E-03	7.782E-04	1.255E-03	7.782E-04	1.255E-03
40	7.782E-04	1.611E-03	1.168E-04	2.024E-04	1.168E-04	2.024E-04
80	7.782E-04	1.802E-03	1.168E-04	2.394E-04	5.673E-05	9.807E-05
160	7.782E-04	1.900E-03	1.168E-04	2.591E-04	5.518E-05	9.157E-05
320	7.782E-04	1.950E-03	1.168E-04	2.693E-04	5.516E-05	9.087E-05
640	7.782E-04	1.975E-03	1.168E-04	2.745E-04	5.516E-05	9.078E-05
$\mathbb{P}^3$						
20	3.082E-05	3.956E-05	3.082E-05	3.956E-05	3.082E-05	3.956E-05
40	3.082E-05	6.339E-05	3.636E-06	4.945E-06	3.636E-06	4.945E-06
80	3.082E-05	7.723E-05	3.636E-06	6.342E-06	2.772E-06	3.021E-06
160	3.082E-05	8.457E-05	3.636E-06	7.214E-06	2.768E-06	2.880E-06
320	3.082E-05	8.835E-05	3.636E-06	7.680E-06	2.768E-06	2.877E-06
640	3.082E-05	9.027E-05	3.636E-06	7.920E-06	2.768E-06	2.877E-06
$\mathbb{P}^4$						
20	9.736E-07	1.032E-06	9.736E-07	1.032E-06	9.736E-07	1.032E-06
40	9.734E-07	1.716E-06	9.736E-07	1.032E-06	1.953E-07	1.032E-06
80	9.734E-07	2.373E-06	1.953E-07	2.564E-07	1.891E-07	1.915E-07
160	9.734E-07	2.733E-06	1.953E-07	2.633E-07	1.891E-07	1.913E-07
320	9.734E-07	2.922E-06	1.953E-07	2.670E-07	1.891E-07	1.913E-07
640	9.734E-07	3.019E-06	1.953E-07	2.690E-07	1.891E-07	1.913E-07

Fig. 7 The log-log plots of N vs. error for $f(x) = \sin(2\pi(x + y))$ on a series of six meshes. Using an enhanced reconstruction reduces the errors, even if it is done only for the first refinement

$$\begin{cases} u_t + u_x = 0, & x \in [-1, 1], t > 0, \\ u(x, 0) = \sin(2\pi x). \end{cases} \quad (19)$$

The initial approximation is done for twenty elements and the errors are calculated after four periods in time ($T = 8$). The same calculation is performed where the initialization uses the L^2 projection onto twenty elements and then is refined either with or without the enhanced reconstruction. The log-log plots of the errors for $p = 1, 2, 3, 4$ is given in

Table 2 L^2 and L^∞ errors for $f(x, y) = \sin(2\pi(x + y))$ for a series of projections onto finer meshes. Left: projection only. Middle: enhanced only for the first projection. Right: enhanced each refinement. Using an enhanced reconstruction reduces the errors, even if it is done only for the first refinement

N	Projection error		Enhanced once		Enhanced each refinement	
	L^2 error	L^∞ error	L^2 error	L^∞ error	L^2 error	L^∞ error
$\mathbb{P}^1$						
20	2.930E-02	5.318E-02	2.930E-02	5.318E-02	2.930E-02	5.318E-02
40	2.930E-02	5.955E-02	8.128E-03	1.180E-02	8.128E-03	1.180E-02
80	2.930E-02	6.282E-02	8.128E-03	1.336E-02	3.585E-03	4.063E-03
160	2.930E-02	6.447E-02	8.128E-03	1.416E-02	3.083E-03	3.180E-03
320	2.930E-02	6.531E-02	8.128E-03	1.457E-02	3.049E-03	2.959E-03
640	2.930E-02	6.572E-02	8.128E-03	1.477E-02	3.047E-03	2.905E-03
$\mathbb{P}^2$						
20	1.556E-03	2.498E-03	1.556E-03	2.498E-03	1.556E-03	2.498E-03
40	1.556E-03	3.215E-03	2.477E-04	4.048E-04	2.477E-04	4.048E-04
80	1.556E-03	3.600E-03	2.477E-04	4.785E-04	1.410E-04	1.962E-04
160	1.556E-03	3.799E-03	2.477E-04	5.180E-04	1.386E-04	1.833E-04
320	1.556E-03	3.900E-03	2.477E-04	5.384E-04	1.385E-04	1.818E-04
640	1.556E-03	3.951E-03	2.477E-04	5.488E-04	1.385E-04	1.818E-04
$\mathbb{P}^3$						
20	6.163E-05	7.902E-05	6.163E-05	7.902E-05	6.163E-05	7.902E-05
40	6.163E-05	1.268E-04	9.028E-06	1.008E-05	9.028E-06	1.008E-05
80	6.163E-05	1.545E-04	9.028E-06	1.309E-05	7.719E-06	5.870E-06
160	6.163E-05	1.692E-04	9.028E-06	1.479E-05	7.712E-06	5.639E-06
320	6.163E-05	1.767E-04	9.028E-06	1.570E-05	7.712E-06	5.630E-06
640	6.163E-05	1.805E-04	9.028E-06	1.616E-05	7.712E-06	5.630E-06
$\mathbb{P}^4$						
20	1.947E-06	2.153E-06	1.947E-06	2.153E-06	1.947E-06	2.153E-06
40	1.947E-06	3.422E-06	5.438E-07	4.879E-07	5.432E-07	4.879E-07
80	1.947E-06	4.738E-06	5.432E-07	5.125E-07	5.347E-07	3.873E-07
160	1.947E-06	5.463E-06	5.432E-07	5.267E-07	5.347E-07	3.882E-07
320	1.947E-06	5.842E-06	5.432E-07	5.342E-07	5.347E-07	3.881E-07
640	1.947E-06	6.036E-06	5.432E-07	5.380E-07	5.347E-07	3.881E-07

Fig. 8 One-dimensional box function plots for $p = 1$. Left: approximation using simple projections. Middle: approximation using the enhanced reconstruction for each refinement. Right: approximation using the enhanced reconstruction only for the first refinement. Notice that the enhanced reconstruction reduces the oscillations at discontinuities

Fig. 9 The log-log plots of N vs. error for the one-dimensional box function using $p = 1$ (left) and $p = 2$ (right) polynomials. It is still beneficial to use the enhanced reconstruction for the first refinement. However, continually using the enhanced reconstruction, while still better, may not be beneficial

Fig. 10 Pointwise error plots for the one-dimensional box function for $p = 1, 2$. In this case, we see that the enhanced reconstruction may create a pollution region around the discontinuity, while reducing the oscillations near a discontinuity

Table 3 L^2 and L^∞ errors for the one-dimensional box function using $p = 1$ and $p = 2$ polynomials. It is still beneficial to use the enhanced reconstruction for the first refinement. However, continually using the enhanced reconstruction, while still better, may not be beneficial

N	Projection error		Enhanced once		Enhanced each refinement	
	L^2 error	L^∞ error	L^2 error	L^∞ error	L^2 error	L^∞ error
$\mathbb{P}^1$						
30	1.607 625E-01	7.066 503E-01	1.607 625E-01	7.066 503E-01	1.607 625E-01	7.066 503E-01
60	1.698 464E-01	8.296 876E-01	1.576 846E-01	7.324 577E-01	1.576 846E-01	7.324 577E-01
120	1.708 998E-01	9.183 919E-01	1.583 456E-01	7.840 795E-01	1.574 693E-01	7.957 835E-01
240	1.708 998E-01	9.257 023E-01	1.583 456E-01	7.883 338E-01	1.576 523E-01	7.993 378E-01
$\mathbb{P}^2$						
20	2.066 583E-01	7.985 227E-01	2.066 583E-01	7.985 227E-01	2.066 583E-01	7.985 227E-01
40	2.105 062E-01	9.772 546E-01	1.812 104E-01	7.367 205E-01	1.812 104E-01	7.367 205E-01
80	2.105 062E-01	9.888 648E-01	1.812 109E-01	7.459 403E-01	1.818 846E-01	7.645 790E-01
160	2.105 062E-01	9.944 918E-01	1.812 112E-01	7.505 385E-01	1.818 858E-01	7.690 328E-01

Fig. 11. After the initial refinement, a better approximation is obtained using an enhanced reconstruction. This matches the corresponding L^2 and L^∞ errors given in Table 4. Pointwise errors for $p = 3, 4$ are given in Fig. 12. The errors for the enhanced reconstruction have fewer oscillations.

Fig. 11 Log-log plots of N vs. Error for the linear transport equation at time $T = 8$ for the initial condition $u(x, 0) = \sin(2\pi x)$ using $p = 1 - 4$ polynomials. it is beneficial to use the enhanced reconstruction after the first refinement

5 Conclusions

In this article, we demonstrated that it is possible to construct a more accurate MRA that exploits the multi-scale structure of the underlying numerical approximation while taking advantage of the hidden accuracy of the approximation. This allows for reducing errors when passing numerical information from coarse scales to fine scales (and vice versa). We specifically utilized the SIAC filtering combined with multi-wavelets. Although this article presents the details of the SIAC filtering using the standard DG method, these techniques are easily extendable to other types of data.

Fig. 12 Errors for the linear transport equation at time $T = 8$ for the initial condition $u(x, 0) = \sin(2\pi x)$ using approximations of polynomial degrees $p = 3$ (top), and $p = 4$ (bottom). The enhanced reconstruction (right column) reduces oscillations in the error over a simple L^2 projection

Table 4 Errors for the linear transport equation at time $T = 8$ for the initial condition $u(x, 0) = \sin(2\pi x)$ using approximations of polynomial degrees $p = 1, 2, 3, 4$. Left: errors using an L^2 projection. Right: enhanced reconstruction for each refinement. The enhanced reconstruction reduces the L^2 and L^∞ errors after the initial refinement

N	Projection error		Enhanced each refinement	
	L^2 error	L^∞ error	L^2 error	L^∞ error
$p=1$				
20	9.385 927E-02	9.710 125E-02	9.385 927E-02	9.710 125E-02
40	1.362 784E-02	1.602 357E-02	1.432 201E-02	1.673 339E-02
80	2.405 020E-03	2.788 872E-03	2.993 704E-03	3.562 015E-03
160	4.228 499E-03	4.940 496E-03	1.283 499E-03	1.584 684E-03
$p=2$				
20	1.711 748E-03	2.773 912E-03	1.711 748E-03	2.773 912E-03
40	2.106 244E-04	3.727 888E-04	2.170 764E-04	3.650 103E-04
80	4.097 155E-04	5.380 845E-04	6.487 097E-05	8.658 932E-05
160	6.083 433E-04	7.706 624E-04	6.509 706E-05	9.096 444E-05
$p=3$				
20	5.455 002E-05	9.708 707E-05	5.455 002E-05	9.708 707E-05
40	7.048 234E-06	1.354 767E-05	4.642 709E-06	8.743 112E-06
80	1.674 569E-05	2.897 475E-05	2.907 231E-06	3.222 856E-06
160	2.873 016E-05	4.396 982E-05	2.880 625E-06	2.890 150E-06
$p=4$				
20	1.065 960E-06	2.237 968E-06	1.065 960E-06	2.237 968E-06
40	7.295 335E-08	1.721 074E-07	1.398 750E-07	2.210 778E-07
80	4.666 768E-07	1.048 080E-06	1.355 866E-07	1.912 797E-07
160	5.772 129E-07	1.218 533E-06	1.355 753E-07	1.913 621E-07

Acknowledgements This work was motivated by discussions with Dr. Venke Sankaran (Edwards Air Force Research Lab, USA) and was performed while visiting the Applied Mathematics group at Heinrich-Heine University, Düsseldorf, Germany. Research supported by the Air Force Office of Scientific Research (AFOSR) Computational Mathematics Program (Program Manager: Dr. Fariba Fahroo) under Grant numbers FA9550-18-1-0486 and FA9550-19-S-0003.

Compliance with Ethical Standards

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alpert, B.K.: A class of bases in L^2 for the sparse representation of integral operators. *SIAM J. Math. Anal.* **24**, 246–262 (1993)
2. Bramble, J.H., Schatz, A.H.: Higher order local accuracy by averaging in the finite element method. *Math. Comput.* **31**, 94–111 (1977)
3. Caviedes-Voullième, D., Gerhard, N., Sikstel, A., Müller, S.: Multiwavelet-based mesh adaptivity with discontinuous Galerkin schemes: exploring 2d shallow water problems. *Adv. Water Resour.* **138**, 103559 (2020)
4. Cockburn, B., Luskin, M., Shu, C.-W., Suli, E.: Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Math. Comput.* **72**, 577–606 (2003)
5. Docampo-Sánchez, J., Ryan, J.K., Mirzargar, M., Kirby, R.M.: Multi-dimensional filtering: reducing the dimension through rotation. *SIAM J. Sci. Comput.* **39**, A2179–A2200 (2017)
6. Gerhard, N., Iacono, F., May, G., Müller, R.S.S.: A high-order discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows. *J. Sci. Comput.* **62**, 25–52 (2015)
7. Guo, W., Cheng, Y.: A sparse grid discontinuous Galerkin method for high-dimensional transport equations and its application to kinetic simulations. *SIAM J. Sci. Comput.* **38**, A3381–A3409 (2016)
8. Guo, W., Zhong, X., Qiu, J.-M.: Superconvergence of discontinuous Galerkin and local discontinuous Galerkin methods: eigen-structure analysis based on Fourier approach. *J. Comput. Phys.* **235**, 458–485 (2013)
9. Hovhannisyann, N., Müller, S., Schäfer, R.: Adaptive Multiresolution Discontinuous Galerkin Schemes for Conservation Laws. Report 311, Institut für Geometrie und Praktische Mathematik, Aachen (2010). <http://www.igpm.rwth-aachen.de/en/reports2010>
10. Ji, L., van Slingerland, P., Ryan, J.K., Vuik, K.: Superconvergent error estimates for position-dependent smoothness-increasing accuracy-conserving post-processing of discontinuous Galerkin solutions. *Math. Comput.* **83**, 2239–2262 (2014)
11. Ji, L., Yan, X., Ryan, J.K.: Accuracy enhancement for the linear convection-diffusion equation in multiple dimensions. *Math. Comput.* **81**, 1929–1950 (2012)
12. Ji, L., Yan, X., Ryan, J.K.: Negative-order norm estimates for nonlinear hyperbolic conservation laws. *J. Sci. Comput.* **54**, 269–310 (2013)
13. Kesserwani, G., Caviedes-Voullième, D., Gerhard, N., Müller, S.: Multiwavelet discontinuous Galerkin h -adaptive shallow water model. *Comput. Methods Appl. Mech. Eng.* **294**, 56–71 (2015)
14. 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. Comput.* **53**, 129–149 (2012)

15. Liu, Y., Cheng, Y., Chen, S., Zhang, Y.-T.: Krylov implicit integration factor discontinuous Galerkin methods on sparse grids for high dimensional reaction-diffusion equations. *J. Comput. Phys.* **388**, 90–102 (2019)
16. Meng, X., Ryan, J.K.: Discontinuous Galerkin methods for nonlinear scalar hyperbolic conservation laws: divided difference estimates and accuracy enhancement. *Numer. Math.* **136**, 27–73 (2017)
17. Meng, X., Ryan, J.K.: Divided difference estimates and accuracy enhancement of discontinuous Galerkin methods for nonlinear symmetric systems of hyperbolic conservation laws. *IMA J. Numer. Anal.* **38**, 125–155 (2018)
18. 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**, 1899–1920 (2011)
19. Mirzaee, H., King, J., Ryan, J.K., Kirby, R.M.: Smoothness-Increasing Accuracy-Conserving (SIAC) filters for discontinuous Galerkin solutions over unstructured triangular meshes. *SIAM J. Sci. Comput.* **35**, A212–A230 (2013)
20. Mirzargar, M., Ryan, J.K., Kirby, R.M.: Smoothness-Increasing Accuracy-Conserving (SIAC) filtering and quasi-interpolation: a unified view. *J. Sci. Comput.* **67**, 237–261 (2016)
21. Nyström, E.J.: Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Math.* **54**, 185–204 (1930)
22. Ryan, J.K., Cockburn, B.: Local derivative post-processing for the discontinuous Galerkin method. *J. Comput. Phys.* **228**, 8642–8664 (2009)
23. Ryan, J.K., Shu, C.-W.: One-sided post-processing for the discontinuous Galerkin method. *Methods Appl. Anal.* **10**, 295–307 (2003)
24. Ryan, J.K.: Exploiting superconvergence through Smoothness-Increasing Accuracy-Conserving (SIAC) filtering. In: Kirby, R., Berzins, M., Hesthaven, J. (eds.) *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014. Lecture Notes in Computational Science and Engineering*, vol. 106, pp. 87–102. Springer, Cham (2015)
25. Tao, Z., Chenand, A., Zhang, M., Cheng, Y.: Sparse grid central discontinuous Galerkin method for linear hyperbolic systems in high dimensions. *SIAM J. Sci. Comput.* **41**, A1626–A1651 (2019)
26. Tao, Z., Guo, W., Cheng, Y.: Sparse grid discontinuous Galerkin methods for the Vlasov-Maxwell system. *J. Comput. Phys.* **3**, 100022 (2019)
27. Thomée, V.: High order local approximations to derivatives in the finite element method. *Math. Comput.* **31**, 652–660 (1977)
28. Van Slingerland, P., Ryan, J.K., Vuik, K.: Position-dependent Smoothness-Increasing Accuracy-Conserving (SIAC) filtering for accuracy for improving discontinuous Galerkin solutions. *SIAM J. Sci. Comput.* **33**, 802–825 (2011)
29. Vuik, M.J., Ryan, J.K.: Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes. *J. Comput. Phys.* **270**, 138–160 (2014)
30. Vuik, M.J., Ryan, J.K.: Multiwavelets and jumps in DG approximations. In: Kirby, R., Berzins, M., Hesthaven, J. (eds.) *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014. Lecture Notes in Computational Science and Engineering*, vol. 106, pp. 503–511. Springer, Cham (2015)
31. Wang, Z., Tang, Q., Guo, W., Cheng, Y.: Sparse grid discontinuous Galerkin methods for high-dimensional elliptic equations. *J. Comput. Phys.* **314**, 244–263 (2016)