Check for
updates

Residual Estimates for Post-processors in Elliptic Problems

Andreas Dedner[1] · Jan Giesselmann[2] · Tristan Pryer[3] · Jennifer K Ryan[4]

Abstract

In this work we examine a posteriori error control for post-processed approximations to elliptic boundary value problems. We introduce a class of post-processing operator that "tweaks" a wide variety of existing post-processing techniques to enable efficient and reliable a posteriori bounds to be proven. This ultimately results in optimal error control for all manner of reconstruction operators, including those that superconverge. We showcase our results by applying them to two classes of very popular reconstruction operators, the Smoothness-Increasing Accuracy-Conserving filter and superconvergent patch recovery. Extensive numerical tests are conducted that confirm our analytic findings.

Keywords Post-processing · SIAC filter · Superconvergent patch recovery · Finite element method · Discontinuous Galerkin · A posteriori error bound · Adaptivity

1 Introduction

Post-Processing techniques are often used in numerical simulations for a variety of reasons from visualisation purposes [4] to designing superconvergent approximations [5] through to becoming fundamental building blocks in constructing numerical schemes [6,12,13]. Another application of these operators is that they are a very useful component in the a posteriori analysis for approximations of partial differential equations (PDEs) [2,33]. The goal of an *a posteriori* error bound is to computationally control the error committed in approximating the

✉ Tristan Pryer
  tmp38@bath.ac.uk

  Andreas Dedner
  A.S.Dedner@warwick.ac.uk

  Jan Giesselmann
  giesselmann@mathematik.tu-darmstadt.de

  Jennifer K Ryan
  jkryan@mines.edu

[1] Mathematics Institute, University of Warwick, Coventry CV4 7AL, UK

[2] Department of Mathematics, TU Darmstadt, 64293 Darmstadt, Germany

[3] Mathematical Sciences, University of Bath, Bath BS2 7AY, UK

[4] Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO 80401, USA

solution to a PDE. In order to illustrate the ideas, let u denote the solution to some PDE and let u_h denote a numerical approximation. Then, the simplest possible use of post-processing in a posteriori estimates is to compute some u^* from u_h and to use

$$\| u - u_h \| \approx \| u^* - u_h \|$$

as an error estimator.

However, a key observation here (and in several more sophisticated approaches) is that u^* must be at least as good of an approximation of the solution u as u_h is. In fact, in many cases, u^* is actually expected to be a better approximation. This raises a natural question: If an adaptive algorithm computes (on any given mesh) not only u_h but also u^* and u^* is a better approximation of u than u_h is, why is u_h and not u^* considered as the "primary" approximation of u? Indeed, the focus of this paper is to consider u^* as the primary approximation of u. We are therefore interested in control of the error $\| u - u^* \|$ and in adaptivity based on an a posteriori estimator for $\| u - u^* \|$. Specifically, *we aim to provide reliable and efficient error control for $\| u - u^* \|$.*

Note that our goal is not to try to construct "optimal" superconvergent post-processors. Rather we try to determine, from the a posteriori viewpoint, the accuracy of some given post-processed solution and to determine how this is useful for the construction of adaptive numerical schemes based on an error tolerance for $u - u^*$.

There are several examples of superconvergent post-processors, including SIAC and superconvergent patch recovery. More details on the history, properties and implementation of these methods will be provided in Sects. 4.1 and 4.2 . However, our a posteriori analysis, aims at being applicable for a wide variety of post-processors and, therefore, we avoid special assumptions that are only valid for specific post-processors. Indeed, our analysis makes only very mild assumptions on the post-processing operator. Specifically, we only require that:

1. The post-processed solution u^* belongs to a finite dimensional space that contains piecewise polynomials, although it does not necessarily need to be piecewise polynomial itself.
2. The post-processed solution should be piecewise smooth over the same triangulation, or a sub-triangulation, of the finite element approximation.

Given a post-processor, u^* that satisfies these rather mild assumptions, we perturb it slightly and call the result u^{**}. This is to ensure an orthogonality condition holds which then allows us to show various desirable properties including:

1. The orthogonal post-processor provides a better approximation than the original post-processor, i.e. $\| u - u^{**} \|_{\mathscr{A}_h} \leq \| u - u^* \|_{\mathscr{A}_h}$ in the energy norm, see Lemma 3.4.
2. The orthogonal post-processor has an increased convergence order in the L^2 norm. Practically, this is not always the case for the original post-processor, see Lemma 3.6.
3. Efficient and reliable a posteriori bounds are available for the error committed by the orthogonal post-processor.

This, motivates us to consider u^{**} (and not u^* or u_h) as the primary approximation. Since the improved accuracy of u^{**}, compared to u_h, stems from superconvergence it is much more sensitive with respect to smoothness of the exact solution, i.e. in regions where the exact solution is C^∞ we expect u^{**} to be much more accurate than u_h whereas in places where the exact solution is less regular, e.g. has kinks, u_h and u^{**} are expected to have similar accuracy. Therefore, meshes constructed based on error estimators for $u - u_h$ will usually not be optimal when used for computing u^{**} in the sense that the ratio of degrees of freedom to error $\|u - u_h\|$ would be much better for other meshes, this is elaborated upon in Sect. 5.

We will demonstrate the good approximation properties of our modification strategy for post-processors and the benefits of basing mesh adaptation on an estimator for $\|u - u_h\|$ in a series of numerical experiments. In order to highlight the versatility of our approach, we conduct experiments based on two popular post-processing techniques: The Smoothness Increasing Accuracy Conserving (SIAC) filter and superconvergent patch recovery (SPR). Background on these methods is provided in Sects. 4.1 and 4.2 respectively.

The rest of the paper is structured as follows: In Sect. 2 we introduce the model elliptic problem and its dG approximation. We also recall some standard results for this method. In Sect. 3, for a given reconstruction, we perturb it so it satisfies Galerkin orthogonality and show some *a priori* type results. We then study a posteriori results and give upper and lower bounds for a residual type estimator. In Sect. 4 we describe the two families of post-processor that we consider in this work. Finally, in Sect. 5 we perform extensive numerical tests on the SIAC and SPR post-processors to show the performance of the *a posteriori* bounds, the effect of smoothness of the solution on the post-processors and to study adaptive methods driven by these estimators.

2 Problem Setup and Notation

Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ be bounded with Lipschitz boundary $\partial\Omega$. We denote by $L^p(\Omega)$, $p \in [1, \infty]$, the standard Lebesgue spaces and $H^s(\Omega)$, the Sobolev spaces of real-valued functions defined over Ω. Further we denote $H_0^1(\Omega)$ the space of functions in $H^1(\Omega)$ with vanishing trace on $\partial\Omega$.

For $f \in L^2(\Omega)$ we consider the problem

$$
\begin{aligned}
- \operatorname{div}(\mathfrak{D}\nabla u) &= f \text{ in } \Omega \\
u &= 0 \text{ on } \partial\Omega,
\end{aligned}
\tag{2.1}
$$

where $\mathfrak{D} : \Omega \to \mathbb{R}^{d \times d}$ is a uniformly positive definite diffusion tensor and $\mathfrak{D} \in \left[H^1(\Omega) \cap L^\infty(\Omega)\right]^{d \times d}$. Weakly, the problem reads: find $u \in H_0^1(\Omega)$ such that

$$
\mathscr{A}(u, v) := \int_\Omega \mathfrak{D}\nabla u \cdot \nabla v \, d\boldsymbol{x} = \int_\Omega f v \, d\boldsymbol{x} \quad \forall\, v \in H_0^1(\Omega).
\tag{2.2}
$$

Let $\mathscr{T}$ be a triangulation of Ω into disjoint simplicial or box-type (quadrilateral/hexahedral) elements $K \in \mathscr{T}$ such that $\overline{\Omega} = \bigcup_{K \in \mathscr{T}} \overline{K}$. Let $\mathscr{E}$ be the set of edges which split into the set of interior edges $\mathscr{E}_i$ and the set of boundary edges $\mathscr{E}_b$.

We introduce the standard broken Sobolev spaces. For $s \in \mathbb{N}_0$ we define

$$
H^s(\mathscr{T}) := \{v \in L^2(\Omega) : v|_K \in H^s(K) \quad \forall\, K \in \mathscr{T}\},
\tag{2.3}
$$

and we will use the notation

$$
\| v \|_{H^s(\mathscr{T})} = \left(\sum_{K \in \mathscr{T}} \| v \|_{H^s(K)}^2 \right)^{1/2}
\tag{2.4}
$$

as an elementwise norm for the broken space.

For $p \in \mathbb{N}$ we denote the set of all polynomials over K of total degree at most p by $\mathbb{P}^p(K)$. For $p \geq 1$, we consider the finite element space

$$
\mathbb{V}_h^p := \{\phi \in L^2(\Omega) : v|_K \in \mathbb{P}^p(K), K \in \mathscr{T}\}.
\tag{2.5}
$$

Let $v \in \mathrm{H}^1(\mathscr{T})$ be an arbitrary scalar function. For any interior edge $e \in \mathscr{E}_i$ there are two adjacent triangles K^-, K^+ and we can consider the traces $v^\pm$ of v from $K^\pm$ respectively. We denote the outward normal of $K^\pm$ by $\boldsymbol{n}^\pm$ and define average and jump operators for one $\mathscr{E}_i$ by

$$
\begin{aligned}
\{\!\{v\}\!\} &:= \frac{1}{2}\left(v^+ + v^-\right) := \frac{1}{2}\left(\lim_{s \searrow 0} v(\cdot + s\boldsymbol{n}^+) + \lim_{s \searrow 0} v(\cdot + s\boldsymbol{n}^-)\right), \\
[\![v]\!] &:= \left(v^-\boldsymbol{n}^- + v^+\boldsymbol{n}^+\right) := \lim_{s \searrow 0} v(\cdot + s\boldsymbol{n}^-)\boldsymbol{n}^- + \lim_{s \searrow 0} v(\cdot + s\boldsymbol{n}^+)\boldsymbol{n}^+.
\end{aligned}
\tag{2.6}
$$

For boundary edges there is only one trace of v and one outward pointing normal vector $\boldsymbol{n}$ and we define

$$
\{\!\{v\}\!\} := v \qquad [\![v]\!] := v\boldsymbol{n}.
\tag{2.7}
$$

For vector valued functions $\boldsymbol{v} \in [\mathrm{H}^1(\mathscr{T})]^d$ we define jumps and averages on interior edges by

$$
\{\!\{\boldsymbol{v}\}\!\} := \frac{1}{2}\boldsymbol{v}^+ + \frac{1}{2}\boldsymbol{v}^-, \quad [\![\boldsymbol{v}]\!] := \left(\boldsymbol{v}^- \cdot \boldsymbol{n}^- + \boldsymbol{v}^+ \cdot \boldsymbol{n}^+\right).
\tag{2.8}
$$

As before, for boundary edges, we define jumps and averages using traces from the interior only. Note that $[\![\boldsymbol{v}]\!]$, $\{\!\{v\}\!\} \in \mathrm{L}^2(\mathscr{E})$ and $[\![v]\!]$, $\{\!\{\boldsymbol{v}\}\!\} \in [\mathrm{L}^2(\mathscr{E})]^d$.

For any triangle $K \in \mathscr{T}$ we define $h_K := \operatorname{diam} K$ and collect these values into an element-wise constant function $h : \Omega \to \mathbb{R}$ with $h|_K = h_K$. We denote the radius of the largest ball inscribed in K by ρ_K. For every edge e we denote by $h_e = \{\!\{h\}\!\}$, i.e., the mean of diameters of adjacent triangles. For our analysis we will assume that $\mathscr{T}$ belongs to a family of triangulations which is quasi-uniform and shape-regular. Let us briefly recall the definitions of these two notions: The triangulation $\mathscr{T}$ is called

- shape-regular if there exists $C > 0$ so that

$$
h_K < C\rho_K \quad \forall K \in \mathscr{T}
\tag{2.9}
$$

- quasi-uniform if there exists $C > 0$ so that

$$
\max_{K \in \mathscr{T}} h_K < Ch_K \quad \forall K \in \mathscr{T}.
\tag{2.10}
$$

Note that for shape-regular triangulations we have inverse and trace inequalities [10, Lemmas 1.44, 1.46]. Note that the quasi-uniformity assumption is only required for the first part of our analysis, in Sect. 3.1 and can be relaxed in Sect. 3.2.

In this work we will consider a standard interior penalty method to approximate solutions of (2.2). We consider the Galerkin method to seek $u_h \in \mathbb{V}_h^p$ such that

$$
\mathscr{A}_h(u_h, v_h) = \int_\Omega f v_h \, d\boldsymbol{x} \quad \forall v_h \in \mathbb{V}_h^p,
\tag{2.11}
$$

where $\mathscr{A}_h : \mathrm{H}^2(\mathscr{T}) \times \mathrm{H}^2(\mathscr{T}) \to \mathbb{R}$ is given by

$$
\mathscr{A}_h(u, v) = \int_\mathscr{T} \mathfrak{D}\nabla u \cdot \nabla v - \int_\mathscr{E} [\![v]\!] \cdot \{\!\{\mathfrak{D}\nabla u\}\!\} - \int_\mathscr{E} [\![u]\!] \cdot \{\!\{\mathfrak{D}\nabla v\}\!\} + \int_\mathscr{E} \sigma h_e^{-1} [\![u]\!] \cdot [\![v]\!]
\tag{2.12}
$$

Note that the bilinear form (2.12) is stable provided $\sigma = \sigma(\mathfrak{D})$ is large enough, see [1].

Remark 2.1 (Continuous Galerkin methods). Note that if we restrict test and trial functions to $\mathbb{V}_h^p \cap \mathrm{H}^1(\Omega)$ then all jumps on interior edges vanish and (2.11), (2.12) reduces to a (continuous) finite element method with weakly enforced boundary data. Our analysis is equally valid in this case.

We introduce two dG norms

$$
\begin{aligned}
\| v \|_{dG}^2 &:= \| \nabla v \|_{\mathrm{L}^2(\mathscr{T})}^2 + \| h_e^{-\frac{1}{2}} [\![v]\!] \|_{\mathrm{L}^2(\mathscr{E})}^2 \\
\| v \|_{\mathscr{A}_h}^2 &:= \mathscr{A}_h(v, v),
\end{aligned}
\tag{2.13}
$$

which are equivalent provided $\sigma > 0$ is sufficiently large and conclude this section by stating a-priori estimates for the Galerkin method as is standard in the literature [1,17].

Theorem 2.2 (*Error bounds for the dG approximation*). *Let $u \in \mathrm{H}^s(\Omega)$ for $s \geq 2$ be the solution of (2.1) and $u_h \in \mathbb{V}_h^p$ be the unique solution to the problem (2.11). Then,*

$$
\| u - u_h \|_{\mathrm{L}^2(\Omega)} + h \| u - u_h \|_{dG} \leq C_1 h^{\min(p+1,s)} |u|_{\mathrm{H}^s(\Omega)}.
\tag{2.14}
$$

Further, for $u \in \mathrm{H}^1(\Omega)$, we have the a posteriori error bound

$$
\| u - u_h \|_{dG} \leq C_2 R_h := C_2 \left(\sum_{K \in \mathscr{T}} \left(\eta_K^2 + \frac{1}{2} \sum_{e \in \partial K} \eta_e^2 \right) \right)^{\frac{1}{2}},
\tag{2.15}
$$

where

$$
\begin{aligned}
\eta_K^2 &:= \| h_K (f + \mathrm{div}(\mathfrak{D} \nabla u_h)) \|_{\mathrm{L}^2(K)}^2 \\
\eta_e^2 &:= \| h_e^{\frac{1}{2}} [\![\mathfrak{D} \nabla u_h]\!] \|_{\mathrm{L}^2(e)}^2 + \| h_e^{-\frac{1}{2}} [\![u_h]\!] \|_{\mathrm{L}^2(e)}^2,
\end{aligned}
\tag{2.16}
$$

and C_1 is a constant depending on the shape-regularity and quasi-uniformity constants of $\mathscr{T}$ and C_2 depends only upon the shape-regularity. Here R_h is a computable residual that we refer to during our numerical simulations.

3 The Orthogonal Reconstruction, a priori and a posteriori Error Estimates

In this section, we derive robust and efficient error estimates. We make the assumption that we have access to a computable reconstruction, $u^* \in \mathbb{V}_h^* \subset \mathrm{H}^2(\mathscr{T})$ generated from our numerical solution u_h, where $\mathbb{V}_h^*$ is required to contain the original finite element space, that is $\mathbb{V}_h^p \subset \mathbb{V}_h^*$. We are unable to provide reliable a posteriori error estimates for u^* directly, but we can modify, and, as we shall demonstrate, improve any such reconstruction such that a robust and efficient error estimate can be obtained for the modified reconstruction.

We split this section into two parts, the first subsection contains the definition of the improved reconstruction and some of its properties. In particular, we study this from an *a priori* viewpoint, show that it satisfies Galerkin orthogonality as well as some desirable *a priori* bounds. Throughout this subsection we assume that $u \in \mathrm{H}^2(\Omega)$ and the underlying mesh is quasi-uniform. In the second part we derive reliable and efficient a posteriori estimates under the weaker assumption that $u \in \mathrm{H}^1(\Omega)$ and the mesh is shape regular.

3.1 Improved Reconstruction

In the following assume that $u \in H^2(\Omega)$ solves (2.2) and let $u^* \in \mathbb{V}_h^* \subset H^2(\mathcal{T})$ be a reconstruction of the discrete solution u_h, e.g. a SIAC reconstruction as described in Sect. 4.1 or obtained by some patch recovery operator as described in Sect. 4.2.

Definition 3.1 Let $R : H^2(\mathcal{T}) \rightarrow \mathbb{V}_h^p$ denote the Ritz projection with respect to $\mathscr{A}_h(\cdot, \cdot)$, i.e.,

$$\mathscr{A}_h(Rv, \phi_h) = \mathscr{A}_h(v, \phi_h) \quad \forall \phi_h \in \mathbb{V}_h^p. \tag{3.1}$$

We define the improved reconstruction as

$$u^{**} := u^* - Ru^* + u_h \in \mathbb{V}_h^* \tag{3.2}$$

Remark 3.2 We make the following remarks:

1. The finite element approximation from (2.11) satisfies $u_h = Ru$.
2. We work under the assumption that a post-processor u^* is already being computed. To realise u^{**} we are required to solve the original elliptic problem a second time with a different forcing term. This means the improved reconstruction u^{**} is computable at a small additional cost to u^*. Once u^* has been computed, u^{**} can be computed by solving a discrete elliptic problem over $\mathbb{V}_h^p$. A typical scenario is that the user already has a good scheme for computing u_h, and that the cost of computing u^{**} (after the post-processing to obtain u^*) is just that of solving the same system as that for u_h with a different right hand side. This means the assembly and preconditioning, perhaps ILU or AMG, can be reused without change.
   Estimating the cost of computing u^* is more complicated and will depend on the method used and the implementation. While our implementation for solving u_h and the correction are optimized (and implemented in C++) the computation of u^{**} is a proof of concept implementation in Python and is therefore not competitive.
3. Note that

$$u - u^{**} = u - u_h - u^* + Ru^* = (id - R)(u - u^*), \tag{3.3}$$

   where id is the identity mapping, i.e., the error of u^{**} is the Ritz-projection of the error of u^* onto the orthogonal complement of $\mathbb{V}_h^p$.
4. Even if u^* is continuous, this does not necessarily hold for u^{**} as $\mathbb{V}_h^p$ may contain discontinuous functions.

One of the key properties of the improved reconstruction is that it satisfies a Galerkin orthogonality result.

Lemma 3.3 (Galerkin orthogonality). *The reconstruction u^{**} from (3.2) satisfies Galerkin orthogonality, i.e.,*

$$\mathscr{A}_h\big(u - u^{**}, v_h\big) = 0 \quad \forall v_h \in \mathbb{V}_h^p. \tag{3.4}$$

Proof For any $v_h \in \mathbb{V}_h^p$, we have using (3.3)

$$\mathscr{A}_h\big(u - u^{**}, v_h\big) = \mathscr{A}_h\big((id - R)(u - u^*), v_h\big) = 0 \tag{3.5}$$

by definition of the Ritz projection, as required.   □

Now, we show that with respect to $\| \cdot \|_{\mathscr{A}_h}$ the new reconstruction u^{**} indeed improves upon u^*:

Lemma 3.4 (Better approximation of the improved reconstruction). *Let u^{**} be defined by* (3.2), *then the following holds:*

$$\| u - u^{**} \|_{\mathscr{A}_h} \leq \| u - u^* \|_{\mathscr{A}_h} . \tag{3.6}$$

In (3.6) *the inequality is an equality if and only if $u^{**} = u^*$, i.e., if the original reconstruction u^* itself satisfies Galerkin orthogonality.*

Proof Since the images of R and $(id - R)$ are orthogonal with respect to $\mathscr{A}_h(\cdot, \cdot)$, Pythagoras' theorem implies

$$
\begin{aligned}
\| u - u^* \|_{\mathscr{A}_h}^2 &= \| (id - R)(u - u^*) + R(u - u^*) \|_{\mathscr{A}_h}^2 \\
&\geq \| (id - R)(u - u^*) \|_{\mathscr{A}_h}^2 = \| u - u^{**} \|_{\mathscr{A}_h}^2
\end{aligned}
\tag{3.7}
$$

We have used (3.3) in the third step. Note that if u^* is not Galerkin orthogonal then

$$\| R(u - u^*) \|_{\mathscr{A}_h} > 0$$

leading to a strict inequality in the first step. This completes the proof. $\qquad\square$

Remark 3.5 One appealing feature of the new reconstruction that results from Galerkin orthogonality is that if the reconstruction u^* has some superconvergence properties in the energy norm this is inherited by u^{**} and also immediately implies an additional order of accuracy in L^2. This results from an Aubin-Nitsche trick being available.

Lemma 3.6 (Dual bounds). *Let Ω be a convex polygonal domain and let u^{**} be defined by* (3.2), *then there exists a constant $C > 0$ (only depending on the shape regularity of the mesh) such that*

$$\| u - u^{**} \|_{L^2(\Omega)} \leq Ch \| u - u^{**} \|_{\mathscr{A}_h} . \tag{3.8}$$

Proof Let $\psi \in H^2(\Omega) \cap H_0^1(\Omega)$ solve

$$- \operatorname{div}(\mathfrak{D}\nabla\psi) = u - u^{**}$$

which implies

$$\int_{\mathscr{T}} \mathfrak{D}\nabla\psi \cdot \nabla v - \int_{\mathscr{E}} [\![v]\!] \cdot \{\!\!\{\mathfrak{D}\nabla\psi\}\!\!\} = \int_{\Omega} (u - u^{**})v \quad \forall v \in H^1(\mathscr{T}). \tag{3.9}$$

Thus, by choosing $v = u - u^{**}$ in (3.9) we obtain

$$
\begin{aligned}
\| u - u^{**} \|_{L^2(\Omega)}^2 &= \int_{\mathscr{T}} \mathfrak{D}\nabla\psi \cdot \nabla(u - u^{**}) - \int_{\mathscr{E}} [\![u - u^{**}]\!] \cdot \{\!\!\{\mathfrak{D}\nabla\psi\}\!\!\} \\
&= \mathscr{A}_h(\psi, u - u^{**}) \\
&= \mathscr{A}_h(\psi - \psi_h, u - u^{**})
\end{aligned}
\tag{3.10}
$$

for any $\psi_h \in \mathbb{V}_h^p$ where the last equality follows from Galerkin orthogonality (3.4). Thus, choosing ψ_h as the best approximation of ψ in the piecewise linear subspace of $\mathbb{V}_h^p$, we obtain

$$
\begin{aligned}
\| u - u^{**} \|_{L^2(\Omega)}^2 &\leq \| \psi - \psi_h \|_{\mathscr{A}_h} \| u - u^{**} \|_{\mathscr{A}_h} \\
&\leq Ch \| u - u^{**} \|_{\mathscr{A}_h} \| \nabla^2\psi \|_{L^2(\Omega)} \\
&\leq Ch \| u - u^{**} \|_{L^2(\Omega)} \| u - u^{**} \|_{\mathscr{A}_h},
\end{aligned}
\tag{3.11}
$$

by elliptic regularity of the dual problem, concluding the proof.   □

3.2 A Posteriori Error Estimates

Now that we have shown some fundamental results on the improved reconstruction, we relax the regularity requirements on u in this subsection allowing for weak solutions to (2.1), that is, $u \in \mathrm{H}^1(\Omega)$. With that in mind we modify the definition of $\mathscr{A}_h(\cdot, \cdot)$ such that it is a suitable extension over $\mathrm{H}^1(\mathscr{T}) \times \mathrm{H}^1(\mathscr{T})$ to

$$\mathscr{A}_h(u, v) := \int_{\mathscr{T}} \mathfrak{D}\nabla u \cdot \nabla v - r_h^*(\llbracket v \rrbracket) \cdot \mathfrak{D}\nabla u - r_h^*(\llbracket u \rrbracket) \cdot \mathfrak{D}\nabla v + \int_{\mathscr{E}} \sigma h_e^{-1} \llbracket u \rrbracket \cdot \llbracket v \rrbracket, \tag{3.12}$$

for $u, v \in \mathrm{H}^1(\mathscr{T})$ and where $r_h^* : [\mathrm{L}^2(\mathscr{E})]^d \to \left[\mathbb{V}_h^*\right]^d$ is the lifting operator that we recall from [10, Section 4.3.1]

$$\int_{\Omega} r_h^*(\varphi) \cdot \mathfrak{D}\psi_h = \int_{\mathscr{E}} \varphi \cdot \{\!\!\{\mathfrak{D}\psi_h\}\!\!\} \qquad \forall \, \psi_h \in \left[\mathbb{V}_h^*\right]^d. \tag{3.13}$$

The lifting operators satisfy the stability estimate, [10, Lemma 4.34],

$$\| r_h^*(\varphi) \|_{\mathrm{L}^2(\Omega)} \le C \| h_e^{-\frac{1}{2}} \varphi \|_{\mathrm{L}^2(\mathscr{E})}, \tag{3.14}$$

For test and trial functions in $\mathbb{V}_h^*$ (which contains $\mathbb{V}_h^p$ by assumption) the new definition of $\mathscr{A}_h(\cdot, \cdot)$ is equivalent to the one given in (2.12). Therefore for any function $v^* \in \mathbb{V}_h^*$ the Ritz projection given in Definition 3.1 remains the same still satisfying $\mathscr{A}_h(Rv^*, \phi_h) = \mathscr{A}_h(v^*, \phi_h)$ for all $\phi_h \in \mathbb{V}_h^p$. But note that we no longer have $u_h = Ru$ and Galerkin orthogonality for u^{**} no longer holds in general, it only holds for a $\mathrm{H}^1(\Omega)$ conforming subspace of $\mathbb{V}_h^p$:

Lemma 3.7 *For $u \in \mathrm{H}^1(\Omega)$ and $z_h \in \mathbb{V}_h^p \cap \mathrm{H}_0^1(\Omega)$ it holds that*

$$\mathscr{A}_h\left(u^{**} - u, z_h\right) = 0. \tag{3.15}$$

Proof By definition of u^{**} we have that

$$\begin{aligned}
\mathscr{A}_h\left(u^{**} - u, z_h\right) &= \mathscr{A}_h\left(u^* - Ru^* + u_h - u, z_h\right) \\
&= \mathscr{A}_h\left(u^* - Ru^*, z_h\right) + \mathscr{A}_h(u_h - u, z_h) \\
&= \mathscr{A}_h(u_h - u, z_h),
\end{aligned} \tag{3.16}$$

since $u^* \in \mathbb{V}_h^*$. Now, notice that by definition

$$\mathscr{A}_h(u_h - u, z_h) = \langle f, z_h \rangle - \int_{\mathscr{T}} \mathfrak{D}\nabla u \cdot \nabla z_h - r_h^*(\llbracket z_h \rrbracket) \cdot \mathfrak{D}\nabla u = 0 \tag{3.17}$$

as z_h is an element of $\mathbb{V}_h^p \cap \mathrm{H}_0^1(\Omega)$ and, hence, continuous, as required.   □

Let a quantity of interest be given by the linear functional $\mathscr{J} \in \mathrm{H}^{-1}(\mathscr{T})$, the dual space of $\mathrm{H}_0^1(\mathscr{T})$. Note that $\mathrm{H}^{-1}(\mathscr{T}) \subset \mathrm{H}^{-1}(\Omega)$ where the latter is the dual space of $\mathrm{H}_0^1(\Omega)$. We begin by deriving an error representation formula. Following [15], we split u^{**} into a continuous part $u_C^{**} \in \mathbb{V}_h^* \cap \mathrm{H}_0^1(\Omega)$ and a discontinuous part $u_{\perp}^{**} \in \mathbb{V}_h^*$ so that

$$u^{**} = u_C^{**} + u_{\perp}^{**} \quad \text{and} \quad \mathscr{A}_h\left(u_{\perp}^{**}, \psi_h\right) = 0 \quad \forall \, \psi \in \mathbb{V}_h^* \cap \mathrm{H}_0^1(\Omega). \tag{3.18}$$

Theorem 3.8 *(Dual error representation).* *Let* $u \in H_0^1(\Omega)$ *be the solution of (2.2) and let* u^{**} *be given by (3.2), then*

$$\mathscr{J}(u - u^{**}) = \langle f, z - z_h \rangle - \mathscr{A}_h(u^{**}, z - z_h) + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}) \qquad (3.19)$$

where $\langle \cdot, \cdot \rangle$ *denotes the* L^2 *scalar product and* $z \in H_0^1(\Omega)$ *is the solution of the dual problem*

$$\mathscr{A}(v, z) = \mathscr{J}(v) \quad \forall \, v \in H_0^1(\Omega) \qquad (3.20)$$

and z_h *is an arbitrary function in* $\mathbb{V}_h^p \cap H_0^1(\Omega)$.

Proof By definition of z, we have, for any $z_h \in \mathbb{V}_h^p \cap H_0^1(\Omega)$,

$$
\begin{aligned}
\mathscr{J}(u - u^{**}) &= \mathscr{J}(u - u_C^{**} - u_\perp^{**}) \\
&= \mathscr{J}(u - u_C^{**}) - \mathscr{J}(u_\perp^{**}) \\
&= \mathscr{A}(u - u_C^{**}, z) - \mathscr{J}(u_\perp^{**}) \\
&= \langle f, z \rangle - \mathscr{A}_h(u_C^{**}, z) - \mathscr{J}(u_\perp^{**}) \\
&= \langle f, z \rangle - \mathscr{A}_h(u^{**}, z) + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}) \\
&= \langle f, z \rangle - \mathscr{A}_h(u^{**}, z - z_h) - \mathscr{A}_h(u^{**}, z_h) + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}) \\
&= \langle f, z - z_h \rangle - \mathscr{A}_h(u^{**}, z - z_h) + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}),
\end{aligned} \qquad (3.21)
$$

where we made use of Galerkin orthogonality, Lemma 3.7, in the last step. $\square$

Theorem 3.9 *(Primal error estimate).* *There exists some constant* $C_A > 0$ *depending on mesh geometry and polynomial degree such that*

$$\| u - u^{**} \|_{dG} \leq C_A R^{**} := C_A \left(\sum_{K \in \mathscr{T}} \left((\eta_K^{**})^2 + \frac{1}{2} \sum_{e \in \partial K} (\eta_e^{**})^2 \right) \right)^{\frac{1}{2}}, \qquad (3.22)$$

where

$$
\begin{aligned}
(\eta_K^{**})^2 &:= \| h_K (f + \mathrm{div}(\mathfrak{D}\nabla u^{**})) \|_{L^2(K)}^2 \\
(\eta_e^{**})^2 &:= \| h_e^{\frac{1}{2}} [\![\mathfrak{D}\nabla u^{**}]\!] \|_{L^2(e)}^2 + \| h_e^{-\frac{1}{2}} [\![u^{**}]\!] \|_{L^2(e)}^2
\end{aligned} \qquad (3.23)
$$

Proof Since

$$\| u - u^{**} \|_{dG} \leq C \left(\| \mathfrak{D}^{\frac{1}{2}} (\nabla u - \nabla u^{**}) \|_{L^2(\mathscr{T})}^2 + \sum_{e \in \mathscr{E}} \| h_e^{-\frac{1}{2}} [\![u^{**}]\!] \|_{L^2(e)}^2 \right)^{\frac{1}{2}}, (3.24)$$

where C is some constant depending on $\mathfrak{D}$ only, it is sufficient to show that

$$\| \mathfrak{D}^{\frac{1}{2}} (\nabla u - \nabla u^{**}) \|_{L^2(\mathscr{T})}$$

is bounded by the right hand side of (3.22).

In Theorem 3.8 we may choose

$$\mathscr{J}(v) := \int_{\mathscr{T}} \mathfrak{D}(\nabla u - \nabla u^{**}) \nabla v. \qquad (3.25)$$

Note that, by definition, $z \in H_0^1(\Omega)$ so that, if $\mathbb{V}_h^p$ contains discontinuous functions, $z \neq u - u^{**}$. Nevertheless, z satisfies the stability estimate

$$\| \mathfrak{D}^{\frac{1}{2}} \nabla z \|_{L^2(\Omega)} \leq \| \mathfrak{D}^{\frac{1}{2}} (\nabla u - \nabla u^{**}) \|_{L^2(\mathscr{T})}. \qquad (3.26)$$

Then, for any $z_h \in \mathbb{V}_h^p \cap H_0^1(\Omega)$, Theorem 3.8 implies

$$
\begin{aligned}
&\| \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \|_{L^2(\mathscr{T})}^2 \\
&= \mathscr{J}(u - u^{**}) \\
&= \langle f, z - z_h \rangle - \mathscr{A}_h(u^{**}, z - z_h) + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}) \\
&= \langle f, z - z_h \rangle - \int_{\mathscr{T}} (\nabla u^{**} - r_h^*([\![u^{**}]\!])) \mathfrak{D} \nabla (z - z_h) \\
&\quad + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}).
\end{aligned}
\tag{3.27}
$$

Integrating by parts in (3.27) and using (3.14) we obtain

$$
\begin{aligned}
&\| \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \|_{L^2(\mathscr{T})}^2 \\
&= \int_{\mathscr{T}} (f + \mathrm{div}(\mathfrak{D} \nabla u^{**}))(z - z_h) - \int_{\mathscr{E}} [\![\mathfrak{D} \nabla u^{**}]\!](z - z_h) \\
&\quad + \int_{\mathscr{T}} r_h^*([\![u^{**}]\!]) \mathfrak{D} \nabla (z - z_h) + \mathscr{A}_h(u_\perp^{**}, z) - \mathscr{J}(u_\perp^{**}) \\
&\le \sum_K h_K \| f + \mathrm{div}(\mathfrak{D} \nabla u^{**}) \|_{L^2(K)} \, h_K^{-1} \| z - z_h \|_{L^2(K)} \\
&\quad + \frac{1}{2} \sum_{e \in \partial K} h_e^{\frac{1}{2}} \| [\![\mathfrak{D} \nabla u^{**}]\!] \|_{L^2(e)} \, h_e^{-\frac{1}{2}} \| z - z_h \|_{L^2(e)} \\
&\quad + C \| h_e^{-\frac{1}{2}} [\![u^{**}]\!] \|_{L^2(\mathscr{E})} \| \mathfrak{D}^{\frac{1}{2}} \nabla (z - z_h) \|_{L^2(\Omega)} \\
&\quad + C \| u_\perp^{**} \|_{dG} \| \mathfrak{D}^{\frac{1}{2}} \nabla (z - z_h) \|_{L^2(\Omega)} + \| \mathscr{J} \|_{H^{-1}(\Omega)} \| u_\perp^{**} \|_{dG}
\end{aligned}
\tag{3.28}
$$

From [14, Theorem 5.3] we obtain

$$
\| u_\perp^{**} \|_{dG} \le C_P \| h_e^{-\frac{1}{2}} [\![u^{**}]\!] \|_{L^2(\mathscr{E})}
\tag{3.29}
$$

with a constant $C_P > 0$ which is independent of h but depends on the shape regularity of the mesh and the polynomial degree and we also note that

$$
\| \mathscr{J} \|_{H^{-1}(\Omega)} \le C \| \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \|_{L^2(\mathscr{T})} .
\tag{3.30}
$$

We insert (3.29) and (3.30) into (3.28) and apply trace inequality and Cauchy-Schwarz inequality and obtain

$$
\begin{aligned}
\| \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \|_{L^2(\mathscr{T})}^2 &\le \left(\sum_K \left((\eta_K^{**})^2 + C \sum_{e \in \partial K} (\eta_e^{**})^2 \right) \right)^{\frac{1}{2}} \| h^{-1}(z - z_h) \|_{L^2(\Omega)} \\
&\quad + C \left(\sum_K \sum_{e \in \partial K} (\eta_e^{**})^2 \right)^{\frac{1}{2}} \| \mathfrak{D}^{\frac{1}{2}} \nabla (z - z_h) \|_{L^2(\Omega)} .
\end{aligned}
\tag{3.31}
$$

Now, we choose $z_h \in \mathbb{V}_h^p \cap H_0^1(\Omega)$ as the Clément interpolant of z so that

$$
\begin{aligned}
\| h^{-1}(z - z_h) \|_{L^2(\Omega)} + \| \mathfrak{D}^{\frac{1}{2}} \nabla (z - z_h) \|_{L^2(\Omega)} &\le C \| \nabla z \|_{L^2(\Omega)} \\
&\le C \| \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \|_{L^2(\mathscr{T})},
\end{aligned}
\tag{3.32}
$$

and insert (3.32) into (3.31) to obtain the assertion of the theorem.   □

The error estimator, derived in Theorem 3.9, is locally efficient in the following sense:

Theorem 3.10 (*Local efficiency*). *Assume f and $\mathfrak{D}$ are piecewise polynomial on $\mathscr{T}$. Then, there exists a constant $C > 0$ independent of h such that for any $K \in \mathscr{T}$ and any $e \in \mathscr{E}$ the following estimates hold:*

$$\eta_K^{**} \leq C \parallel \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \parallel_{\mathrm{L}^2(K)} \tag{3.33}$$

and

$$\eta_e^{**} \leq C \parallel \mathfrak{D}^{\frac{1}{2}} \nabla (u - u^{**}) \parallel_{\mathrm{L}^2(K_e)} \tag{3.34}$$

where K_e denotes the union of cells sharing common edge e.

Proof Both proofs are standard and follow [33].   □

Remark 3.11 (*Data oscillation*). In case f or $\mathfrak{D}$ are not polynomial the right hand side of (3.33) contains additional data oscillation terms.

4 Post-processors

In order to show the versatility of our results, we consider two families of reconstruction operators. Namely, the Smoothness-Increasing Accuracy-Conserving (SIAC) post-processing [5,30,32] as well as patch reconstruction via the Zienkiewicz and Zhu [37,39] Superconvergent Patch Recovery (SPR) technique. Below we outline the procedure for performing these reconstructions as well as error estimates for the ideal case.

4.1 SIAC Post-processors

One example of a superconvergent post-processor that we examine is the Smoothness Increasing Accuracy Conserving (SIAC) filter. The SIAC filter has its roots in an accuracy-enhancing post-processor developed by Bramble and Schatz [5]. The original analysis was done for finite element approximations for elliptic equations. This technique has desirable qualities including its locality, allowing for efficient parallel implementations, and its effectiveness in almost doubling the order of accuracy rather than increasing the order of accuracy by one or two orders. This post-processor was also explored from a Fourier perspective and for derivative filtering by Thomeé [32] and Ryan and Cockburn [29].

SIAC filters are an extension of the above ideas and have traditionally been used to reduce the error oscillations and recover smoothness in the solution and its derivatives for visualization purposes [24,31,34] or to extract accuracy out of existing code [28]. It has been extended to a variety of PDEs as well as meshes [16]. A quasi-interpolant perspective on SIAC can be found in [26]. The important property of these filters is that, in addition to increasing the smoothness, for smooth initial data and linear problems, the filtered solution is more accurate than the DG solution. To combat the high computational cost of the tensor-product nature of the multi-dimensional kernel, a line filter was introduced in [11].

For ease of presentation the following discussion only details the design of the filter and presents a-priori error estimates for the case of a smooth solution. Although the discussion is limited to one-dimension, it can be extended to Cartesian meshes in more than

one space dimension using a tensor product approach. More advanced applications of the multi-dimensional SIAC post-processor are the Hexagonal SIAC [22] or Line SIAC [11].

The basic idea is that the reconstruction is done via convolution post-processing:

$$u^*(\bar{x}) = K_H^{2r+1,m+1} * u_h = \frac{1}{H} \int_{-\infty}^{\infty} K\left(\frac{\bar{x}-y}{H}\right) u_h(y)\, dy,$$

where h is the mesh size of the numerical scheme and H is the scaling of the post-processor. The convolution kernel, $K^{2r+1,m+1}(\cdot)$, is defined as

$$K^{2r+1,m+1}(x) = \sum_{\gamma=-r}^{r} c_\gamma^{2r+1,m+1} \psi^{(m+1)}(x-\gamma).$$

This is a linear combination of $2r+1$ shifted copies of some function, $\psi^{(m+1)}(x)$. The function weights are real scalars, $c_\gamma^{2r+1,m+1} \in \mathbb{R}$. For the kernel, r is chosen to satisfy consistency as well as $2r$ moment requirements, i.e., polynomial reproduction conditions, which are necessary for preserving the accuracy of the Galerkin scheme and m is chosen for smoothness requirements. We focus on kernels built from B-splines which are defined via the B-Spline recurrence relation:

$$\psi^{(1)} = \chi_{[-1/2,1/2]}$$
$$\psi^{(m+1)} = \frac{1}{m}\left[\left(x+\frac{m+1}{2}\right)\psi^{(m)}\left(x+\frac{1}{2}\right) + \left(\frac{m+1}{2}-x\right)\psi^{(m)}\left(x-\frac{1}{2}\right)\right], \quad (4.1)$$

for $m \geq 1$.

Remark 4.1 (Kernel scaling.). For cartesian grids, the kernel scaling is typically chosen to be the element size, $H = h$. In adaptive meshes or structured triangular meshes and tetrahedral meshes, H is typically chosen to be the length of the mesh pattern [21]. For Line SIAC, the kernel scaling is taken to be the element diagonal [11], for unstructured meshes, the kernel scaling is taken to be largest element side [23,25].

It can be shown that when the solution is sufficiently smooth the post-processed numerical solution u^* is a superconvergent approximation.

In particular, if $u \in C^\infty(\Omega)$, then the Galerkin solution converges in Theorem 2.2 as

$$\| u - u_h \|_{L^2(\Omega)} = O(h^{p+1}). \qquad (4.2)$$

If we choose $r = p$ and $m = p - 2$ then

$$\| u - u^* \|_{L^2(\Omega)} = O(h^{2p}), \qquad (4.3)$$

see Theorem 1 in [5,32], which for $p \geq 2$ constitutes an improvement. It is possible to obtain the same estimates in H^1 by taking higher order B-Splines.

In this paper, in order to apply the post-processor globally, we mirror the underlying approximation as an odd function at the boundary as discussed in [5].

Remark 4.2 (Impact of in-cell regularity of u^{**}). If $u^*, u^{**} \notin H^2(\mathcal{T})$ Theorem 3.9 does not hold. Still, as long as $u^* \in H^1(\mathcal{T})$ similar results can be obtained by slightly modifying the proof of Theorem 3.9. One interesting example is SIAC reconstruction with $m = 0$. In this case, for any $K \in \mathcal{T}$ the restriction $u^{**}|_K$ contains several kinks, i.e. there are hypersurfaces (points for $d = 1$, lines for $d = 2$) across which u is continuous but not differentiable. However, for any K there exists a triangulation $\mathcal{T}_K$ of K, such that $u^{**}|_K \in H^2(\mathcal{T}_K)$. If we

follow the steps of the proof of Theorem 3.9 we realise that integration by parts can only be carried out on elements of $\cup_{K \in \mathscr{T}} \mathscr{T}_K$ and each term $\parallel h_K(f + \Delta u^{**}) \parallel^2_{\text{L}^2(K)}$ in the error bound needs to be replaced by

$$\sum_{T \in \mathscr{T}_K} \parallel h_K(f + \Delta u^{**}) \parallel^2_{\text{L}^2(T)} + \frac{1}{2} \sum_{e \in \mathscr{E}_K} \parallel h_K^{\frac{1}{2}} [\![\nabla u^{**}]\!] \parallel^2_{\text{L}^2(e)}, \tag{4.4}$$

where $\mathscr{E}_K$ denotes the set of interior edges of $\mathscr{T}_K$. Efficiency of this modified estimator can be shown along the same lines as in Theorem 3.10 but bubble functions with respect to the elements and edges in the sub-triangulation $\mathscr{T}_K$ need to be used.

4.2 Superconvergent Patch Recovery

The second post-processing operator we study is based on the superconvergent patch recovery (SPR) technique. This was originally studied numerically and showed a type of superconvergence for elliptic equations using finite element approximations [36]. The mathematical theory behind this recovery technique was addressed by Zhang and Zhu [38] for the two-point boundary value problems and for two-dimensional problems and extended to parabolic problems in [18,19]. The superconvergent patch recovery method works by recovering the derivative approximation values for one element from patches surrounding the nodes of that element using a least squares fitting of the superconvergent values at the nodes and edges. In typical derivative recovery, the derivative approximation is a continuous piecewise polynomial of some given degree. For overlapping patches, the recovered derivative is just an average of the approximations obtained on the surrounding patches. Unlike SIAC post-processing, this recovery technique does not rely on translation invariance for the high-order recovery. The superconvergent patch recovery technique has been shown to work well for elliptic equations that have a smooth solution, and for less smooth solutions with a suitably refined mesh.

The usual application of this technique is for gradient recovery. However, in this article we apply this technique to recover function values.

As mentioned, we suitably modify the algorithm to construct a function $u^* \in \mathbb{V}_h^*$ with $\mathbb{V}_h^* = \mathbb{V}_h^{2p+1} \cap C^0(\Omega)$. The construction of u^*, given some finite element function u_h, is carried out in two steps:

1. Construct a polynomial q_i of order $2p$ at each node v_i of the mesh using a least squares fitting of function values of u_h evaluated at suitable points in elements surrounding v_i.
2. Given an element K we use linear interpolation of the values of q_i for the three nodes of K to compute $u^* \in \mathbb{V}_h^*$.

There are many approaches for constructing the polynomials q_i in the first step at a given node v_i with surrounding triangles K'. For our experiments, we use the following approach. For $p = 1$, we construct a quadratic polynomial q_i by fitting the values of u_h at the nodes of all K'. For a piecewise quadratic u_h ($p = 2$) we also use the midpoints of all edges of the K'. Finally for our tests with $p = 3$ we evaluate u_h at two points on each edge chosen symmetrically around the midpoint of the edge (we use the Lobatto points with local coordinates $\frac{1}{2} \pm \frac{\sqrt{5}}{10}$) and also add the value of u_h at the barycentre of the K'. This is depicted in Fig. 1. To guarantee that we have enough function values to compute the least squares fits, we add a second layer of triangles around v_i if necessary, e.g., at boundary nodes.

Note that this procedure is similar, although not the same, as the approach investigated in [35]. Another related procedure was proposed in [27].
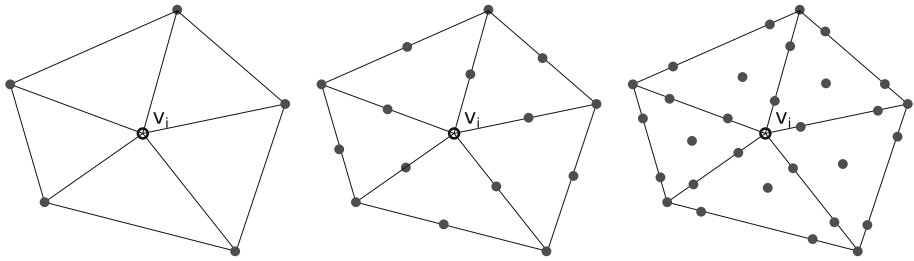
Fig. 1 Evaluation points of u_h used for the least squares fit of a polynomial at node v_i for $p = 1, 2, 3$ (from left to right)

5 Numerical Results

In this section we study the numerical behaviour of the error indicators proposed for the SIAC and SPR post-processing operators. We compare this behaviour with the true error on some typical model problems. The computational work was done in the DUNE package [3] based on the new Python frontend for the DUNE-FEM module [8,9].

5.1 Smoothness-Increasing Accuracy-Conserving Post-processors

The implementation of the post-processor is done through simple matrix-vector multiplication and is discussed in [20].

We first investigate the behaviour of the error and the residual estimator for the problem (2.1) in one space dimension with $\mathfrak{D} = 1$, i.e. the Laplace problem

$$-u'' = f \text{ in } \Omega$$
$$u = 0 \text{ on } \partial\Omega, \tag{5.1}$$

where the forcing function f is chosen so that the exact solution is

$$u(x) = \sin(6\pi x)^2 \cos\left(\frac{9}{2}\pi x\right) \tag{5.2}$$

on the interval $(0, 1)$. We show both the L^2 and H^1 errors for the Galerkin approximation u_h, the SIAC postprocessed approximation, u^* and the orthogonal postprocessor, u^{**}. We also show the two residual indicators R_h (from Theorem 2.2) and R^{**} from (Theorem 3.9). For the basis we consider the continuous Lagrange polynomials for u_h and impose the boundary conditions weakly with a penalty parameter $\frac{10p^2}{h}$, where p is the polynomial degree and h is the grid spacing. Additional experiments were conducted using a discontinuous Galerkin approximation, but no significant differences in the outcome where found and therefore do not include the results. We solve the resulting linear system using an exact solver [7] to avoid issues with stopping tolerances.

We will mainly focus on $p = 2$ but also show results for $p = 1$ and $p = 3$. The SIAC postprocessing is constructed using a continuous B-spline, $m = 1$, as well as setting $r = \lceil \frac{p+1}{2} \rceil$. This leads to an inner stencil of $2\lceil r + \frac{1}{2} - 1\rceil + 1 = 2\lceil \frac{p+1}{2} + 3\rceil$ elements. We also tested other choices of r, m for $p = 2$ but the above choice provided the best results and these are the results shown.
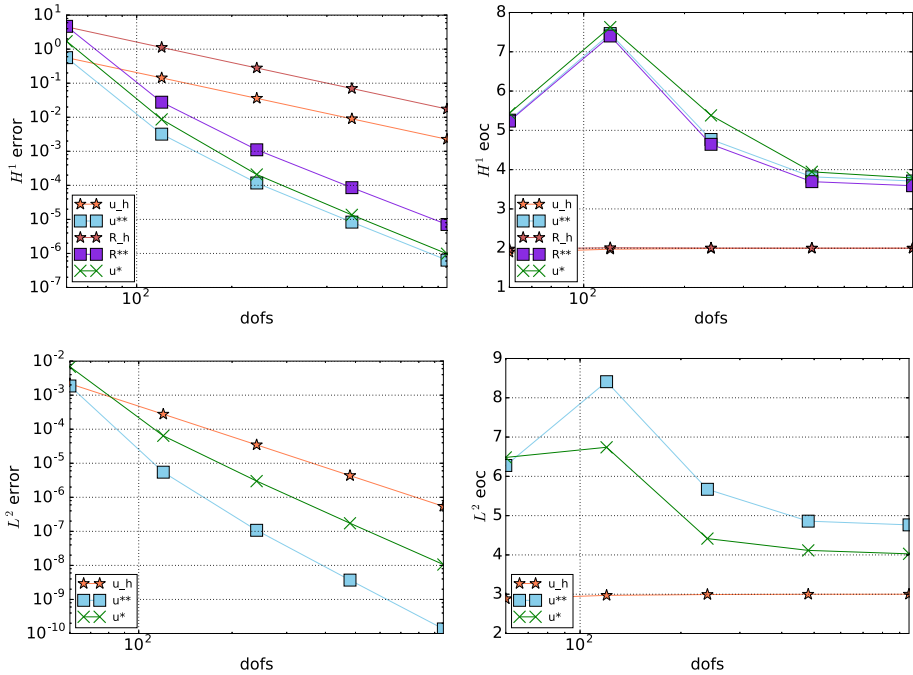
Fig. 2 Errors and convergence rates for H^1 (left two) and L^2 (right two) for polynomial degree $p = 2$ using a SIAC reconstruction

In Fig. 2 we show the errors for $p = 2$ for a series of grid refinement levels starting with 20 intervals and doubling that number on each level. In Fig. 2 we plot the corresponding Experimental Orders of Convergence (EOCs). As can clearly be seen, SIAC postprocessing (u^*) improves the convergence rate in H^1 from 2 to 3 and in L^2 from 3 to 4. While in H^1 the Galerkin orthogonality trick only leads to a small improvement in the error, in L^2 we see an improvement of a full order leading to a convergence rate of 5. As expected from the theory the residual indicators follow the H^1 errors of u_h and u^{**} closely. The efficiency index is comparable between R_h and R^{**}.

For a better understanding of how the error is reduced by utilizing SIAC postprocessing and the Galerkin orthogonality treatment, we show the pointwise errors of the approximations in Fig. 3. It is evident that the function values are much smoother when applying SIAC. The move from u^* and u^{**} does reintroduce small scale errors, but at a far lower level compared to the original approximation, u_h. As expected from the errors, the differences in H^1 are less pronounced.

In Fig. 4 we show errors and EOCs for $p = 3$. Due to the very low errors on the final grid the actual convergence rates for u^* and u^{**} is not clear. However, the improvement due to the Galerkin orthogonality trick, especially in L^2 is quite noticeable as it reduced the error by two orders of magnitude.

We next show results for $p = 1$ in Fig. 5. Again, there is a clear reduction in the values of the errors from u_h to u^* to u^{**} in H^1 together with an improvement in the convergence rate due to the SIAC postprocessing. This improvement in the convergence rate is about 1 order. While the convergence rate going from u_h to u^{**} seems to only be half an order on the higher grid resolution, it is important to note that the error using u^{**} is still significantly smaller
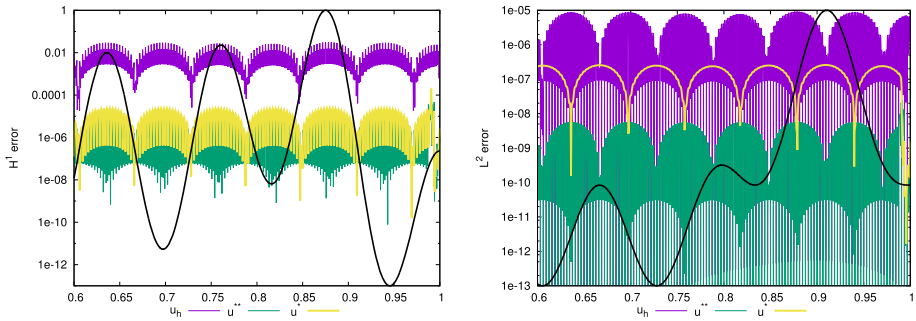
Fig. 3 Pointwise errors for the three different solutions on the right half of the interval, evaluating the function in a number of points per interval. These are results with $p = 2$ and $h = 1/320$ showing the errors in the gradient (left) and in the function values (right)
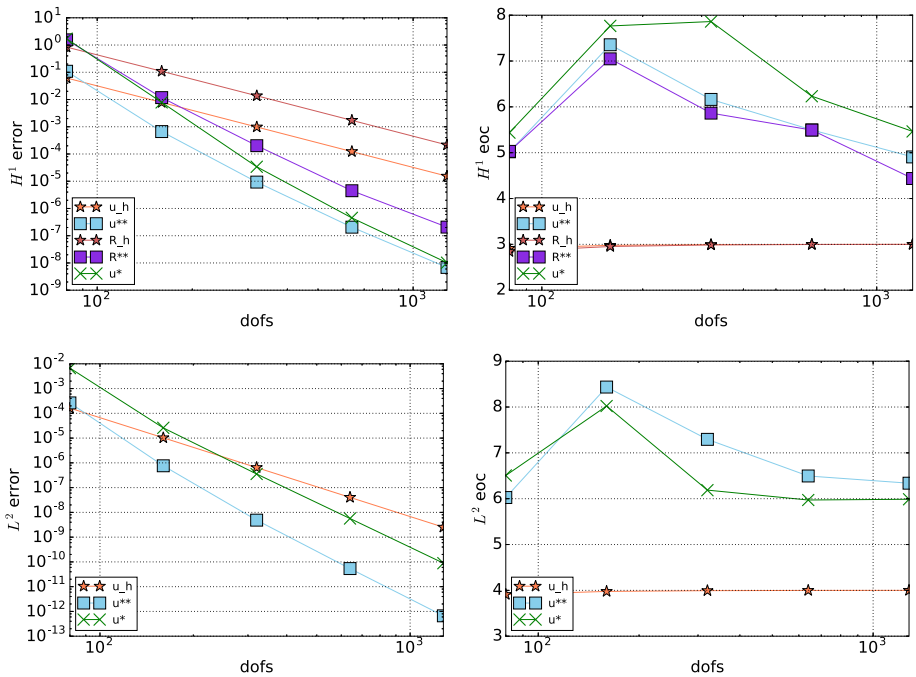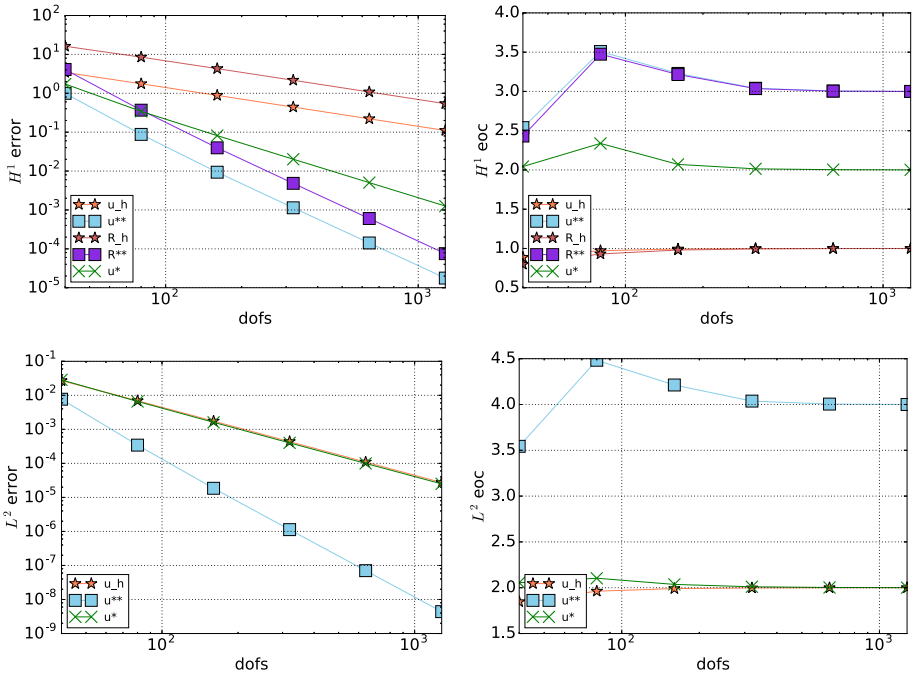

Fig. 4 Errors and convergence rates for H^1 (left two) and L^2 (right two) for polynomial degree $p = 3$ using SIAC reconstruction

than the error between the exact solution and u^* by at least a factor of 2. Hence the results do not contradict the theory. In L^2, SIAC leads to no improvement while the convergence rate of the error using u^{**} is at least half an order higher. Overall the improvement in the convergence rate is not quite as good as for the higher polynomial degrees. The following tests summarized in Fig. 6 show that the weak form of the boundary conditions is responsible for the reduced order improvement. The figure shows results using a hyperpenalty of the
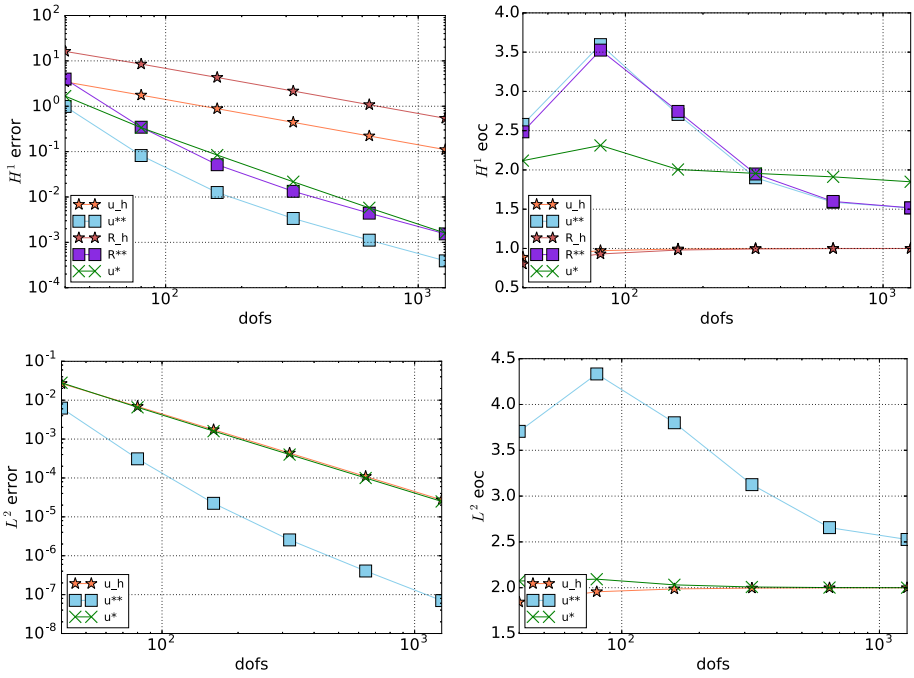
Fig. 5 Errors and convergence rates for H^1 (left two) and L^2 (right two) for polynomial degree $p = 1$ using SIAC reconstruction

Fig. 6 Errors and convergence rates for H^1 (left two) and L^2 (right two) for polynomial degree $p = 2$ with hyperpenalty at the boundary

Table 1 Experimental rates of convergence for the smooth problem using different values for the polynomial degree p. The convergence rates are shown for the three approximations, i.e., u_h, u^*, u^{**}

	$p = 1$			$p = 2$			$p = 3$		
	EOC(u_h)	EOC(u^*)	EOC(u^{**})	EOC(u_h)	EOC(u^*)	EOC(u^{**})	EOC(u_h)	EOC(u^*)	EOC(u^{**})
L^2-error	2	2	4	3	4	5	4	6	6
H^1-error	1	2	3	2	4	4	3	5	5

form $\frac{10p^2}{h^2}$. Applying this hyperpenalty term leads to improvements that are again more in line with our observations for higher order polynomials. We note that strong enforcement of the boundary conditions also lead to similar results.

We summarize our results for the smooth problem in Table 1. It can be clearly seen that the step from u^* to u^{**}, which requires solving one additional low order problem, is quite advantageous and increases the convergence rate in the L^2 norm by at least one. In the linear case this improves by two and by one in the H^1 norm. This makes it highly efficient in this case, at least when implementing the hyperpenalization or strong constraints to enforce the Dirichlet boundary conditions. The reason for this restriction will be investigated further in future work. For $p = 3$ the actual EOCs of the postprocessed solutions are difficult to determine and therefore we provide approximate numbers. In this case, SIAC shows a higher order in L^2 compared to the H^1 norm. The Galerkin orthogonalty trick does not improve the rate further, but note that the overall error is still a factor of 100 smaller. Additionally note that in the other cases where there is no improvement in the rate, the error is reduced by enforcing Galerkin orthogonality, e.g., in the H^1 norm with $p = 2$ the error is still reduced by about a factor of two. In addition, the orthogonality of u^{**} allows us to compute a reliable and efficient error estimator with a comparable efficiency index to the error estimator for u_h using R_h.

We conclude our investigations of the SIAC reconstruction and the residual estimates by studying problems with less smooth solutions. We change the forcing function so that the exact solution is of the form

$$u(x) = \begin{cases} w\left(\frac{x-0.3}{0.4}\right) & x \in (0.3, 0.7), \\ 0 & \text{otherwise} \end{cases}$$

where

$$w(s) = \sin\left(6\pi s\right)^2 \cos\left(\frac{9}{2}\pi s\right) \tag{5.3}$$

is the smooth function from previous studies. We show results for polynomial degree $p = 2$. We again implement a simple $O(h^{-1})$ penalty term at the boundary. Note that the solution is in $C^2 \setminus C^3$ at $x = 0.7$ and only in $C^1 \setminus C^2$ for $x = 0.3$. Overall the solution is an element of $H^2(0, 1)$ but not of $H^3(0, 1)$, i.e., it is not smooth enough to achieve optimal convergence rates for $p = 2$ when the mesh is not aligned. Even when the mesh is aligned, as in our experiments, we do not expect an increase of the convergence rate using the SIAC reconstruction as can be seen in Fig. 7. The local loss of regularity at $x = 0.3$ and $x = 0.7$ is clearly visible for the pointwise errors of the two reconstructions as shown in Fig. 8. Examining the errors in the original approximation, u_h, the reduced smoothness is hardly visible. However, in both of the reconstructions a jump in the error is clearly visible. At $x = 0.7$, where the solution is still in C^2 the error in u^{**} increases approximately by two orders while at $x = 0.3$ it is
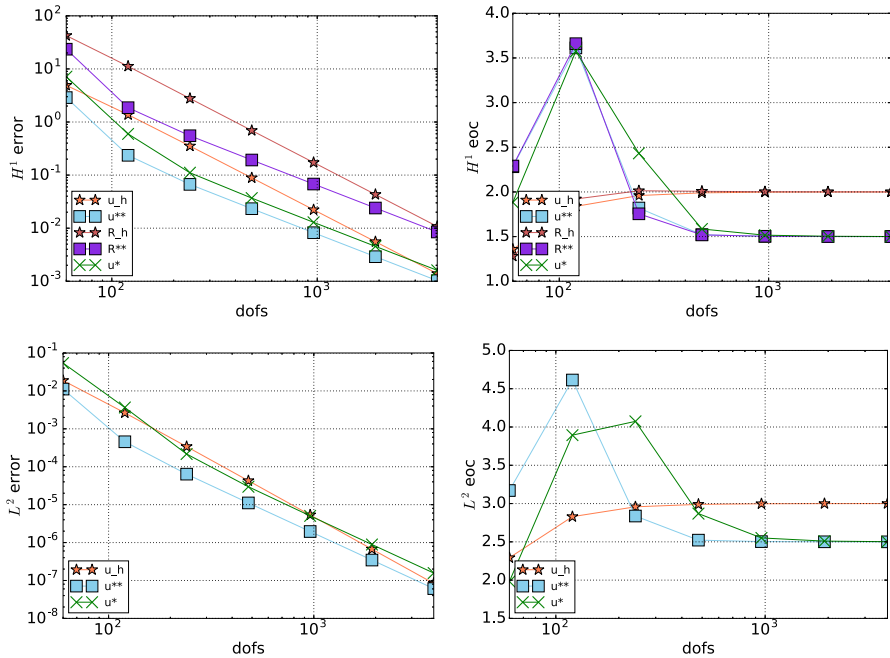
Fig. 7 Errors and convergence rates for H^1 (left two) and L^2 (right two) for polynomial degree $p = 2$ using SIAC for solution with reduced smoothness

close to four orders of magnitude larger since the solution is only C^1 at this point. The lack of smoothness is also identified by the residual indicator R^{**}, the spatial distribution of which is shown in Fig. 9 together with the distribution of R_h. It is worthwhile to note that the region of the 'reduced smoothness' is better isolated by R^{**} than by R_h. Hence it would be easier for an adaptive algorithm to separate these different smoothness regions which would lead to more optimal meshes. The picture clearly shows that R_h does not 'see' the kink so that an adaptive algorithm would either refine the whole non-constant region or nothing at all depending on the tolerance. In contrast, with R^{**} (and the right algorithm) refinement could be isolated to the kinks.

5.2 Superconvergent Patch Recovery

In the following we solve

$$- \operatorname{div}(\mathfrak{D} \nabla u) = f \tag{5.4}$$

in a two dimensional domain Ω where the forcing function f is chosen by prescribing an exact solution u. This function is also used to prescribe Dirichlet boundary conditions on all of $\partial\Omega$. In the first example we chose a smooth exact solution u with a scalar diffusion coefficient $\mathfrak{D} = I_2(|x|^2 + \frac{1}{2})$, while for the second test we use a solution with a corner singularity and $\mathfrak{D} = I_2$.

In the following we show results using a DG scheme on a triangular grid. The grid is refined by splitting each element into four elements. In the final examples with local adaptivity, this
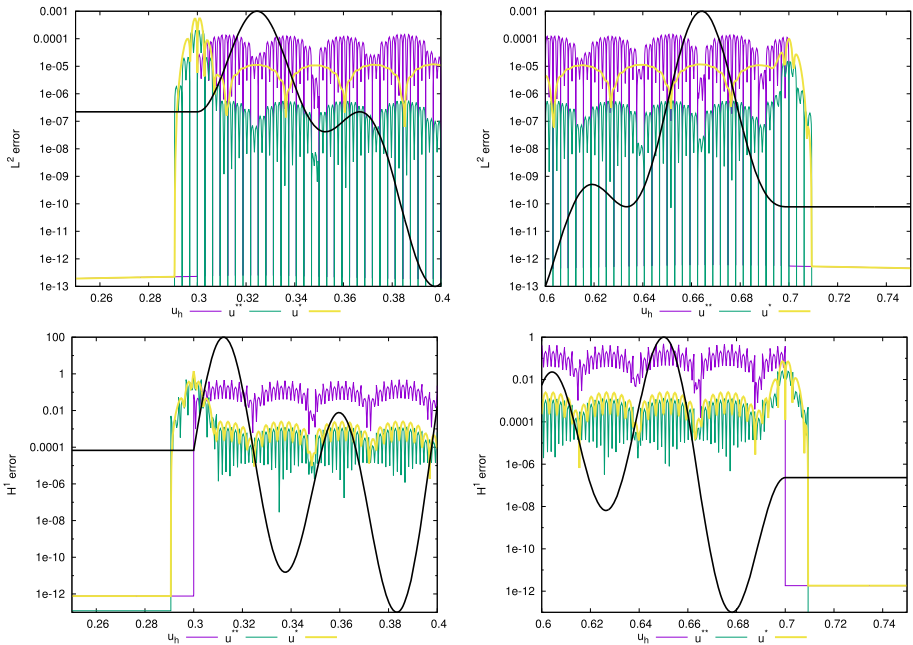
Fig. 8 Pointwise errors of the three different approximations for the piecewise smooth problem. Top row shows the difference in the solution values around $x = 0.3$ (left) and around $x = 0.7$ (right). The bottom row shows gradient errors in the same two regions. These are results with $p = 2$ and $h = 1/320$
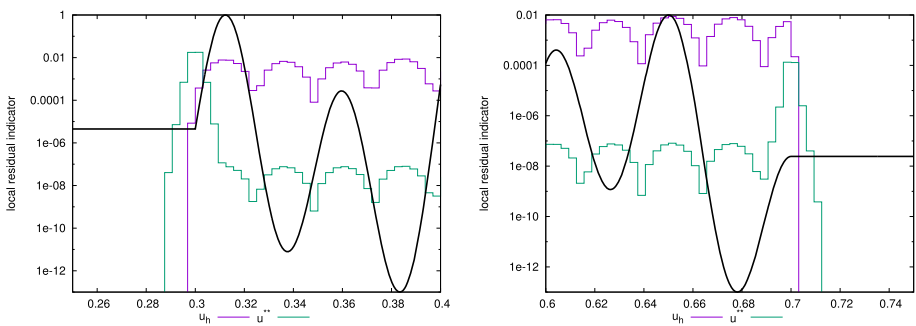

Fig. 9 Elementwise residual indicators using u_h and u^{**} for the piecewise smooth problem around $x = 0.3$ (left) and around $x = 0.7$ (right). bottom row shows gradient errors in the same two regions. These are results with $p = 2$ and $h = 1/320$

leads to a grid with hanging nodes. We also carried out experiments using a continuous ansatz space with very simular results.

Note that in all figures depicting errors and EOCs, the x-axis shows the number of degrees of freedom for u_h. While the other approximations have a larger number of degrees of freedom, the global problem that has to be solved, i.e. solving the linear system for u_h and for Ru^*, scales with the number of degrees of freedom for u_h and thus this seems a reasonable indication of the computational complexity.
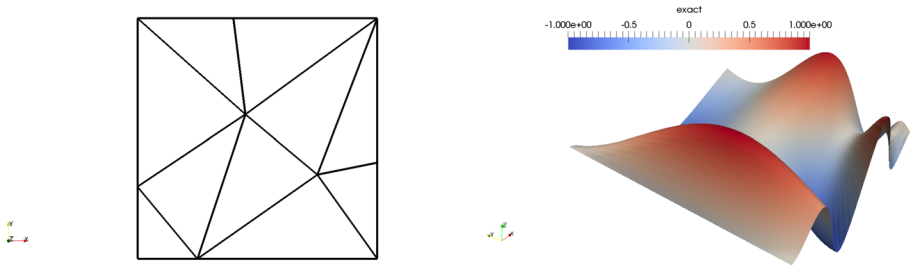
Fig. 10 Macro grid and exact solution u for smooth problem. Note that the solution has been scaled down by a factor of 4
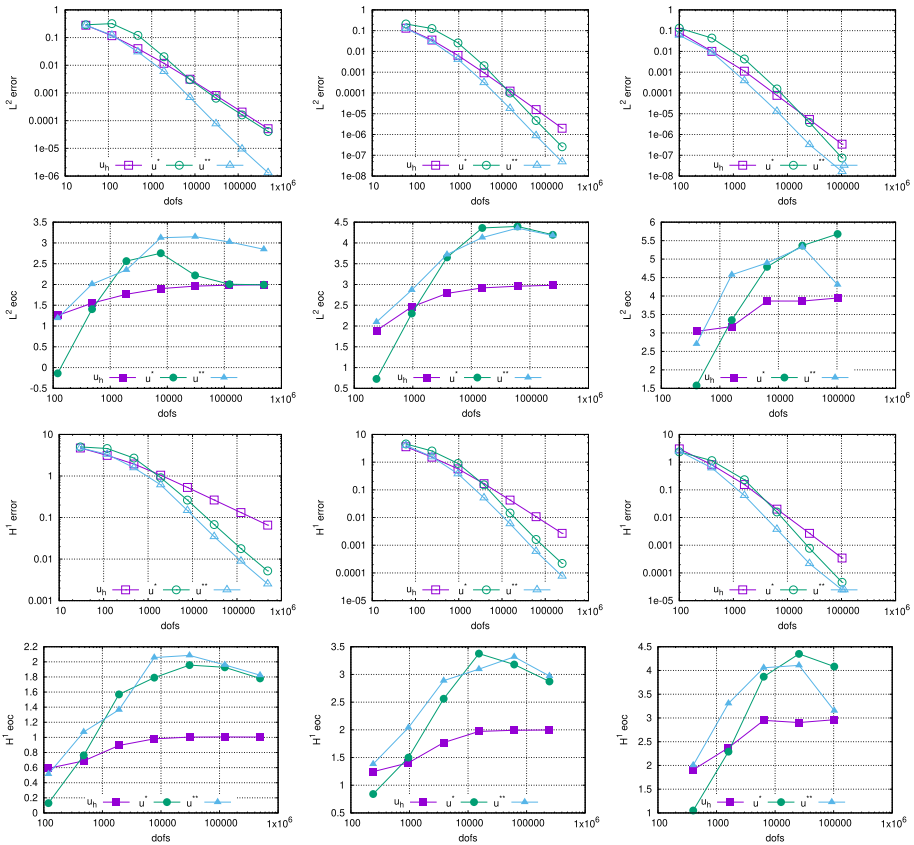
Fig. 11 L^2 errors and EOC (top two rows) and H^1 errors and EOC (bottom two rows) for the smooth problem with $p = 1, 2, 3$ (left to right)

For our first test we choose $u(x, y) = \sin(\pi x/(0.25 + xy))\sin(\pi(x + y))$, and $\Omega = (0, 1)^2$. We start with an initial grid which is slightly irregular as shown in Fig. 10. This is to avoid any superconvergence effects due to a structured layout of the triangles.

Figure 11 shows L^2 and H^1 errors and EOCs for the three approximations u_h, u^*, u^{**} with polynomial degrees $p = 1, 2, 3$. It can be seen that, in general, the postprocessor u^* improves the EOC by an order of 1 in the H^1 norm and that the EOC of the improved postprocessor u^{**}
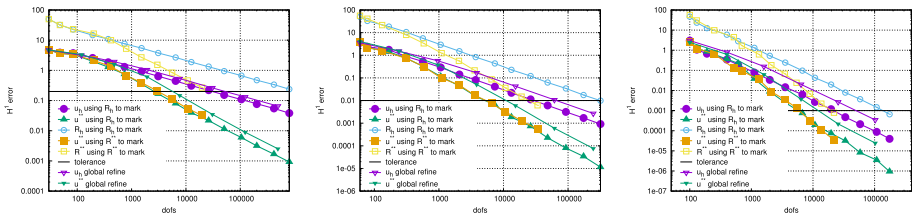
Fig. 12 H^1 errors and residuals for smooth problem with $p = 1, 2, 3$ (left to right). Results are shown for an adaptive mesh using a tolerance of 0.2, 0.01, 0.001 for $p = 1, 2, 3$, respectively

is at least as good. While the actual error of u^* can be larger on coarser grids than the error computed with u_h, the error using u^{**} is significantly better in all cases. Focusing now on the L^2 norm, we see that when computing the error using u^{**}, the EOC is one order better then the convergence rate in the H^1 norm, as expected. For $p = 2, 3$, this is also true when using u^*, while for $p = 1$ the EOC is only 2 in this case, and an increase to 3 is only achieved with the improved postprocessor u^{**}. The same observation can be made when using the SIAC postprocessor in the previous section.

Using the same problem setting, we investigate the performance of an adaptive algorithm in Fig. 12. We use a *modified equal distribution* strategy where elements are marked for refinement when the local indicator η_K exceeds $\frac{\sum \eta_K}{\#elements}$. We compute the local indicator on either u_h or on the improved reconstruction u^{**}. The advantage of basing the marking strategy on u^{**} is clearly demonstrated. While marking with respect to u_h and then using the postprocessor only on the final solution (filled upward triangles) leads to a significant reduction of the final error, the difference in the convergence rate between R_h and R^{**} results in a finer grid than necessary for a given tolerance. A reduction in the number of degrees of freedom by a factor of 10 to 100 can be easily achieved by using R^{**}.

For our final test we study a reentrant corner type problem, i.e., $\Omega = (-1, 1)^2 \backslash ([0, 1] \times [-1, 0])$ using a regular triangulation. First we choose the well known exact solution $u \in H^{\frac{3}{2}}$ leading to $f = 0$. Since the solution is not even H^2 we can not expect the postprocessed solution to have an increased convergence rate. This is confirmed by our numerical tests summarized in Fig. 13. Due to the reduced smoothness and the simplicity of the solution away from the corner, the postprocessing does not only not improve the EOC but can even lead to a slight increase in the overall error clearly noticeable in the H^1 error for the $p = 2$ case. This is even more obvious when the postprocessor, u^*, is used directly. Alternatively, going from u^* to u^{**} leads to an approximation which is very close to the original u_h in all cases. Although the results for the globally refined grid are not that promising, the postprocessing nevertheless has considerable benefits when adapting the grid using the residual indicator based on u^{**}. Indeed Fig. 14 shows that, for a given number of dofs, mesh adaptation based on R^{**} produces an approximation u^{**} which has a much smaller error than u_h (on a mesh constructed using R_h).

For a more challenging test, especially for $p = 3$, we construct the forcing function so that the exact solution is $u(x, y) = \omega(x, y)u_{corner}(x, y)$, where u_{corner} is the solution to the above corner problem and $\omega(x, y) = -\sin\left(\frac{3}{2}\pi(1 - x^2)(1 - y^2)\right)$. The function u still has the same corner singularity but is also smooth. However, the challenging nature of this solution is that it has large gradients towards the outer boundaries. Results for $p = 2, 3$ are summarized in Figs. 15 and 17 . The final grids for $p = 3$ are shown in Fig. 16 using R_h and R^{**} to mark cells for refinement. In both cases 22 steps were needed and the resulting
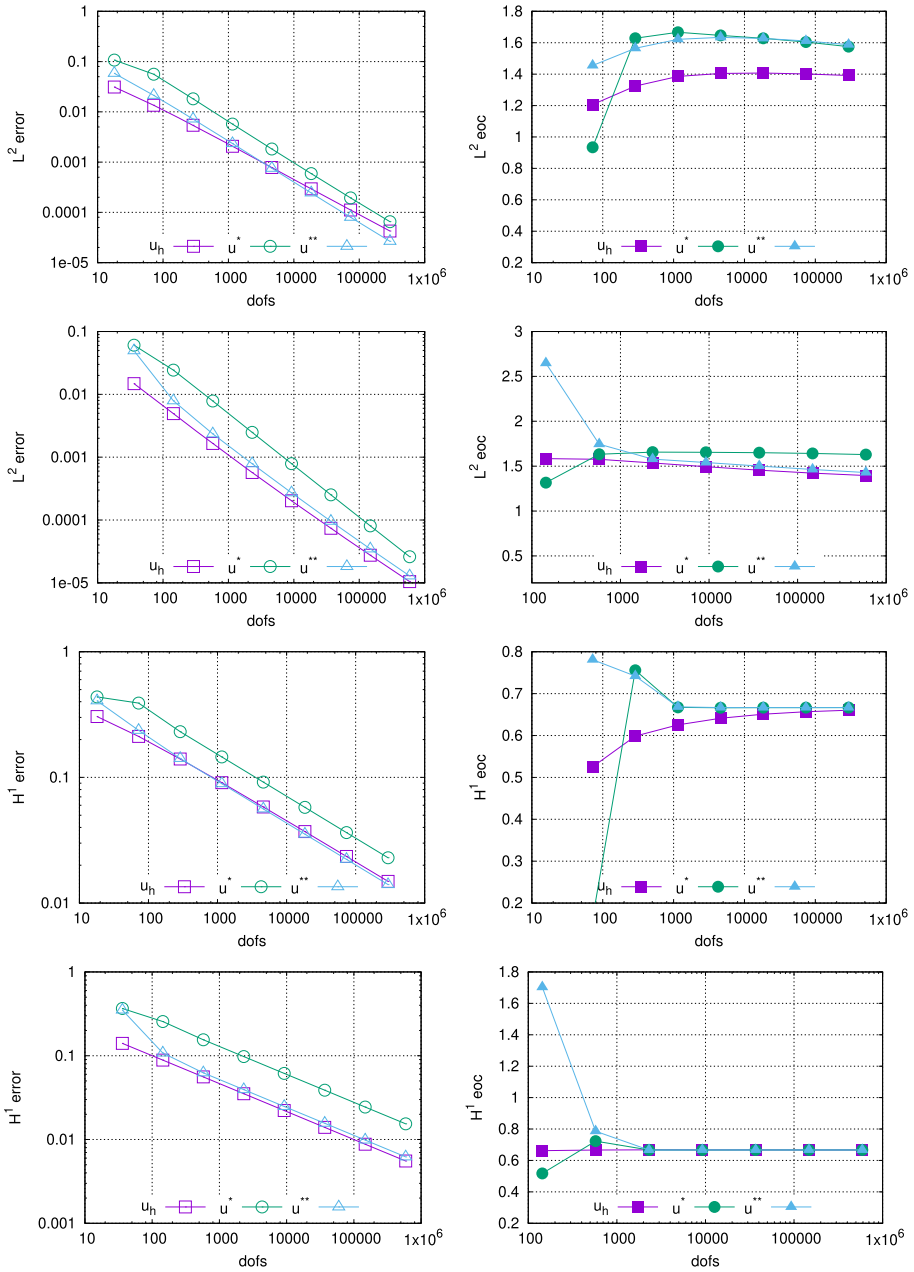
Fig. 13 Errors (right) and EOCs (left) for a simple corner problem. From top to bottom: L^2 with $p = 1, 2$ and H^1 with $p = 1, 2$
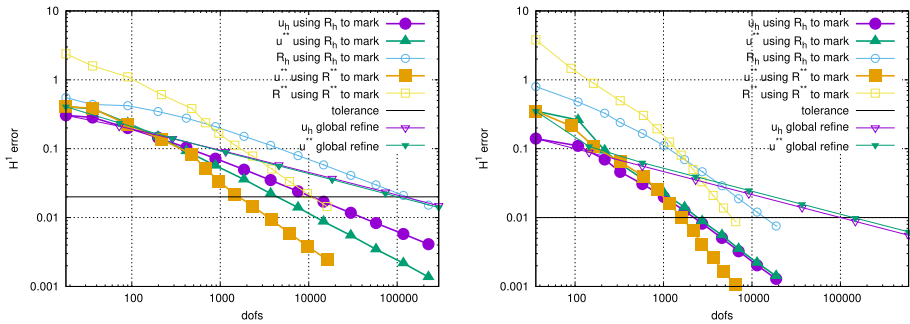
Fig. 14 H^1 errors and residuals for simple corner problem with $p = 1$ (left) and $p = 2$ (right). Results are shown for an adaptive mesh using a tolerance of 0.01

grids have 1597 and 4540 cells (20725 and 7381 degrees of freedom), respectively. While the corner is highly refined in both cases, the regions that are smooth but strongly varying in their solution are far less refined when using R^{**}. When using R_h the final errors are $\|u - u_h\|_{dG} \approx 7.4 \cdot 10^{-4}$ and $\|u - u^{**}\|_{dG} \approx 6.8 \cdot 10^{-4}$ while adaptivity based on R^{**} results in errors of the size $\|u - u_h\|_{dG} \approx 3.9 \cdot 10^{-3}$ and $\|u - u^{**}\|_{dG} \approx 7.0 \cdot 10^{-4}$. Because of the corner singularity, using the postprocessor after finishing the refinement (based on R_h) does not lead to a significant improvement while basing the adaptive process on R^{**} leads to an almost identical error while requiring only 35% of the cells.

Figure 18 shows the efficiency index for all three test cases on globally refined grids. The results seem to indicate that there is only a slight increase in the efficiency index $\frac{R^{**}}{\|\nabla(u^{**}-u)\|}$ compared to $\frac{R_h}{\|\nabla(u_h-u)\|}$.

6 Summary Discussion

In this article, we provide a strategy for improving existing post-processing strategies for numerical solutions of a model elliptic problem. The main idea is to modify the post-processed solution so that it satisfies Galerkin orthogonality. We prove various *a priori* type results showing desirable convergence properties of the orthogonal post-processor including an increased order of accuracy in the L^2 norm. We supported the analysis with numerical examples using two types of post-processors – that of SIAC and SPR – approximating smooth and non-smooth solutions.

In addition to the a priori results, we provide a reliable and efficient a posteriori error estimator for the orthogonal post-processed solution, it should be noted that no such estimator is available for $u - u^*$. We demonstrate in several examples that much more efficient meshes are obtained when adaptation is based on R^{**} than when refinement is based on R_h and the post-processor is only applied to the numerical solution on the final mesh.
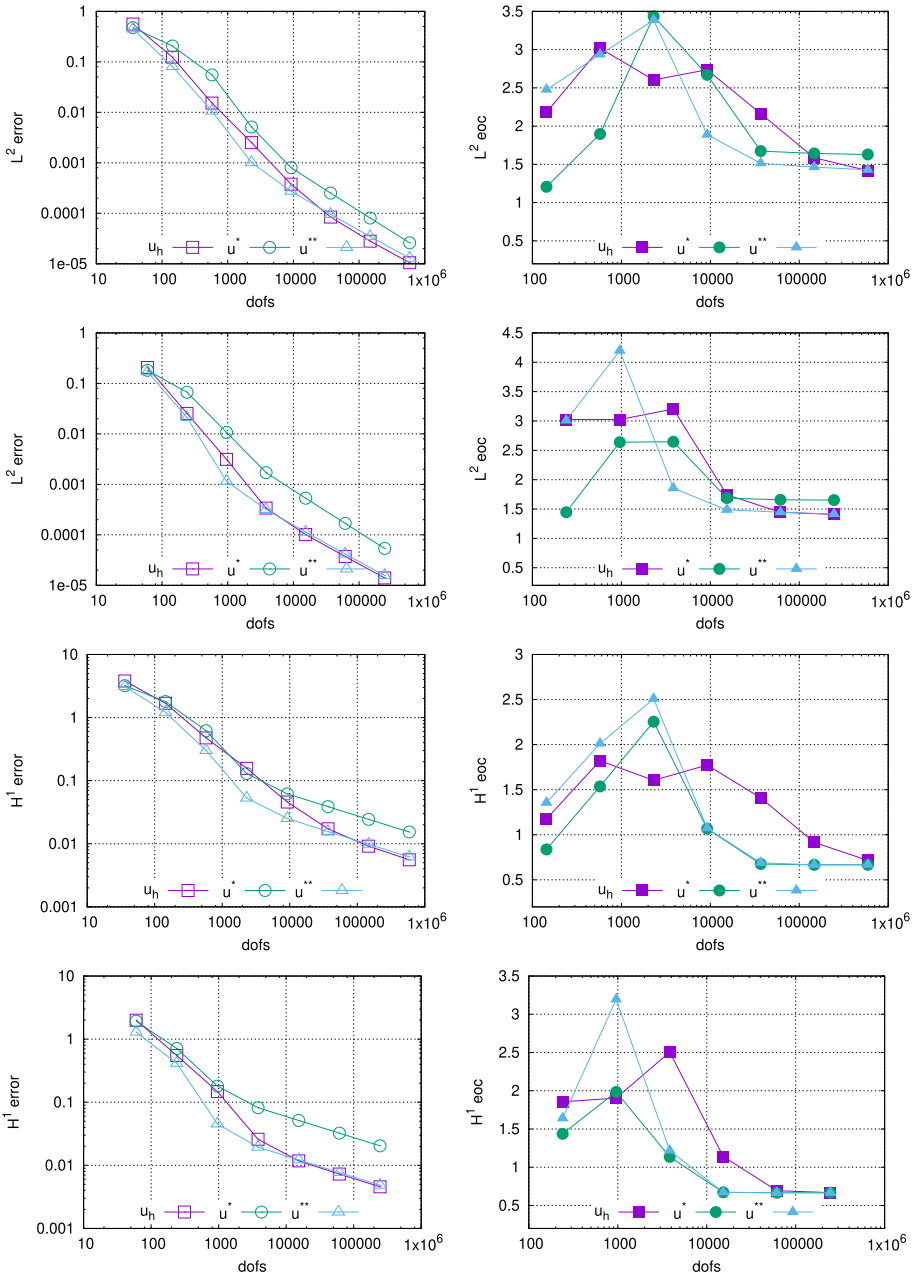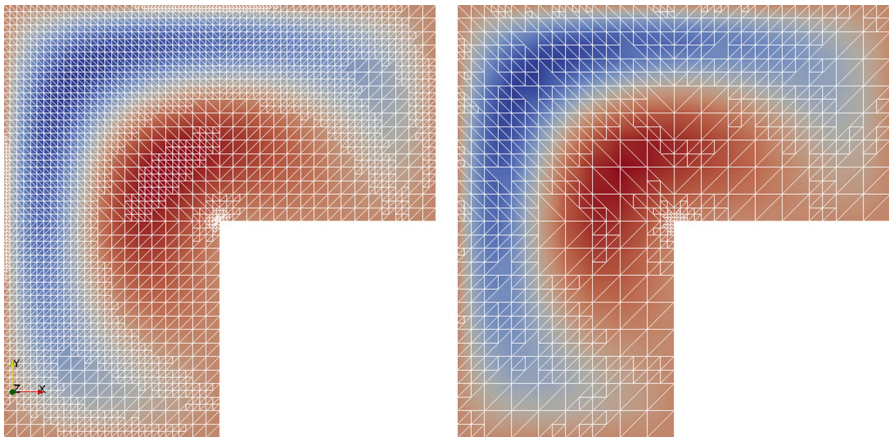
Fig. 15 Errors (right) and EOCs (left) for the extended corner problem. From top to bottom: L^2 with $p = 2, 3$ and H^1 with $p = 2, 3$
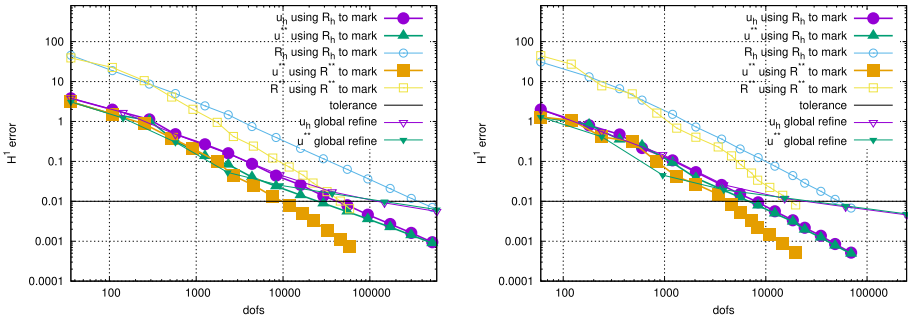
Fig. 16 H^1 errors and residuals for extended corner problem with $p = 2$ (left) and $p = 3$ (right). Results are shown for an adaptive mesh using a tolerance of 0.01

Fig. 17 Discrete solution and adapted grid for extended corner problem with $p = 2$ using R_h (left) and R^{**} (right) for marking. The iterate is chosen so that the resulting errors satisfty $e_h \approx e^{**} \approx 0.013$
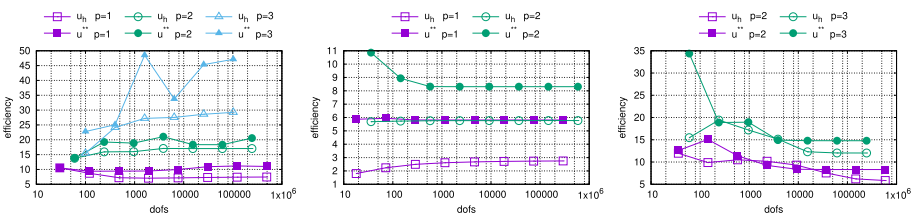
Fig. 18 Efficiency index on globally refined grids for smooth problem with $p = 1, 2, 3$ (left), simple corner problem with $p = 1, 2$ (middle), and extended corner problem with $p = 2, 3$ (right)

References

1. Arnold, D. N., Brezzi, F., Cockburn, B., Donatella Marini, L.: Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal., 39(5):1749–1779, (2001/02)
2. Ainsworth, M., Oden, J.T.: A Posteriori Error Estimation in Finite Element Analysis. Pure and Applied Mathematics. Wiley, New York (2000)
3. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. II. Implementation and tests in DUNE. Computing, 82(2-3):121–138 (2008)
4. Benzley, S.E., Merkley, K., Blacker, T.D., Schoof, L.: Pre-and post-processing for the finite element method. Finite Elements Anal. Des. **19**(4), 243–260 (1995)
5. Bramble, J.H., Schatz, A.H.: Higher order local accuracy by averaging in the finite element method. Math. Comp. **31**(137), 94–111 (1977)
6. Chen, H., Guo, H., Zhang, Z., Zou, Q.: A C^0 linear finite element method for two fourth-order eigenvalue problems. IMA J. Numer. Anal. **37**(4), 2120–2138 (2017)
7. Davis, T.A.: Algorithm 832: Umfpack v4. 3–an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. **30**(2), 196–199 (2004)
8. Dedner, A., Klöfkorn, R., Nolte, M., Ohlberger, M.: A generic interface for parallel and adaptive discretization schemes: abstraction principles and the DUNE-FEM module. Computing **90**, 165–196 (2010)
9. Dedner, A., Nolte, M.: The Dune Python Module. *ArXiv preprint*, arXiv:1807.05252 (2018)
10. Pietro, D. A. D., Ern, A.: Mathematical aspects of discontinuous Galerkin methods. Mathématiques & Applications (Berlin) [Mathematics & Applications], vol. 69. Springer, Heidelberg (2012)
11. Docampo-Sánchez, J., Ryan, J.K., Mirzargar, M., Kirby, R.M.: Multi-dimensional filtering: reducing the dimension through rotation. SIAM J. Sci. Comput. **39**(5), A2179–A2200 (2017)
12. Georgoulis, E.H., Pryer, T.: Recovered finite element methods. Comput. Methods Appl. Mech. Eng. **332**, 303–324 (2018)
13. Guo, H., Zhang, Z., Zou, Q.: A C^0 linear finite element method for biharmonic problems. J. Sci. Comput. **74**(3), 1397–1422 (2018)
14. Houston, P., Perugia, I., Schötzau, D.: Mixed discontinuous Galerkin approximation of the Maxwell operator. SIAM J. Numer. Anal. **42**(1), 434–459 (2004)
15. Houston, Paul, Schötzau, Dominik, Wihler, Thomas P.: Energy norm a posteriori error estimation for mixed discontinuous Galerkin approximations of the Stokes problem. J. Sci. Comput. **22**(23), 347–370 (2005)
16. Ji, L., Yan, X., Ryan, J.K.: Accuracy enhancement of the linear convection-diffusion equation in multiple dimensions. Math. Comput. **81**, 1929–1950 (2012)
17. Karakashian, O.A., Pascal, F.: A posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems. SIAM J. Numer. Anal. **41**(6), 2374–2399 (2003)
18. Lakkis, O., Pryer, T.: Gradient recovery in adaptive methods for parabolic problems. IMA J. Numer. Anal. **32**(1), 246–278 (2012)
19. Leykekhman, D., Wahlbin, L.: A posteriori error estimates by recovered gradients in parabolic finite element equations. BIT Numer. Math. **48**, 585–605 (2008)
20. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Efficient implementation of smoothness-increasing accuracy-conserving (SIAC) filters for discontinuous Galerkin solutions. J. Sci. Comput. **52**(1), 85–112 (2012)
21. 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)
22. Mirzargar, Mahsa, Jallepalli, Ashok, Ryan, Jennifer K., Kirby, Robert M.: Hexagonal smoothness-increasing accuracy-conserving filtering. J. Sci. Comput. **73**(2), 1072–1093 (2017)
23. 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)

24. Mirzaee, H., Ryan, J.K., Kirby, R.M.: Quantification of errors introduced in the numerical approximation and implementation of smoothness-increasing accuracy conserving (siac) filtering of discontinuous galerkin (dg) fields. J. Sci. Comput. **45**(1–3), 447–470 (2010)
25. Mirzaee, Hanieh, Ryan, Jennifer K., Kirby, Robert M.: Smoothness-increasing accuracy-conserving (siac) filters for discontinuous galerkin solutions: Application to structured tetrahedral meshes. J. Sci. Comput. **58**(3), 690–704 (2014)
26. Mirzargar, M., Ryan, J.K., Kirby, R.M.: Smoothness-increasing accuracy-conserving (siac) filtering and quasi-interpolation: a unified view. J. Sci. Comput. **67**(1), 237–261 (2016)
27. Ovall, Jeffrey: Function, gradient and hessian recovery using quadratic edge-bump functions. J. Sci. Comput. **45**(3), 1064–1080 (2007)
28. Ryan, J.K., Shu, C.-W., Atkins, H.: Extension of a postprocessing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. SIAM J. Sci. Comput. **26**(3), 821–843 (2005)
29. Ryan, Jennifer K., Cockburn, B.: Local derivative post-processing for the discontinuous Galerkin method. J. Comput. Phys. **228**(23), 8642–8664 (2009)
30. Ryan, J. K.: *Exploiting Superconvergence through Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering*, pp. 87–102. Springer International Publishing, Cham (2015)
31. Steffan, Michael, Curtis, Sean, Kirby, Robert M., Ryan, Jennifer K.: Investigation of smoothness enhancing accuracy-conserving filters for improving streamline integration through discontinuous fields. IEEE-TVCG **14**, 680–692 (2008)
32. Thomée, Vidar: High order local approximations to derivatives in the finite element method. Math. Comput. **31**(139), 652–660 (1977)
33. Verfürth, R.: A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. Wiley-Teubner, Chichester-Stuttgart (1996)
34. Walfisch, D., Ryan, J.K., Kirby, R.M., Haimes, R.: One-sided smoothness-increasing accuracy-conserving filtering for enhanced streamline integration through discontinuous fields. J. Sci. Comput. **38**(2), 164–184 (2009)
35. Zhang, Z., Naga, A.: A new finite element gradient recovery method: superconvergence property. SIAM J. Sci. Comput **26**, 1192–1213 (2005)
36. Zienkiewicz, O.C., Zhu, J.Z.: A simple error estimator and adaptive procedure for practical engineering analysis. Internat. J. Numer. Methods Engrg. **24**(2), 337–357 (1987)
37. Zienkiewicz, O.C., Zhu, J.Z.: The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique. Int. J. Numer. Methods Eng. **33**(7), 1331–1364 (1992)
38. Zhang, Z.,Zhu, J. Z.: Superconvergence of the derivative patch recovery technique and a posteriori error estimation. In *Modeling, mesh generation, and adaptive numerical methods for partial differential equations (Minneapolis, MN, 1993)*, volume 75 of *IMA Vol. Math. Appl.*, pp. 431–450. Springer, New York (1995)
39. Zhang, Z., Zhu, J.Z.: Analysis of the superconvergent patch recovery technique and a posteriori error estimator in the finite element method (ii). Computer Methods Appl. Mech. Eng. **163**(1), 159–170 (1998)