J Sci Comput (2017) 73:644–666
DOI 10.1007/s10915-017-0507-7

Adjoint-Based, Superconvergent Galerkin Approximations of Linear Functionals

Bernardo Cockburn[1] · Zhu Wang[2]

Received: 27 February 2017 / Revised: 16 May 2017 / Accepted: 14 July 2017 /
Published online: 1 August 2017

Abstract We propose a new technique for computing highly accurate approximations to linear functionals in terms of Galerkin approximations. We illustrate the technique on a simple model problem, namely, that of the approximation of $J(u)$, where $J(\cdot)$ is a very smooth functional and u is the solution of a Poisson problem; we assume that the solution u and the solution of the adjoint problem are both very smooth. It is known that, if u_h is the approximation given by the continuous Galerkin method with piecewise polynomials of degree $k > 0$, then, as a direct consequence of its property of Galerkin orthogonality, the functional $J(u_h)$ converges to $J(u)$ with a rate of order h^{2k}. We show how to define approximations to $J(u)$, with a computational effort about twice of that of computing $J(u_h)$, which converge with a rate of order h^{4k}. The new technique combines the adjoint-recovery method for providing precise approximate functionals by Pierce and Giles (SIAM Rev 42(2):247–264, 2000), which was devised specifically for numerical approximations without a Galerkin orthogonality property, and the accuracy-enhancing convolution technique of Bramble and Schatz (Math Comput 31(137):94–111, 1977), which was devised specifically for numerical methods satisfying a Galerkin orthogonality property, that is, for finite element methods like, for example, continuous Galerkin, mixed, discontinuous Galerkin and the so-called hybridizable discontinuous Galerkin methods. For the latter methods, we present numerical experiments, for $k = 1, 2, 3$ in one-space dimension and for $k = 1, 2$ in two-space dimensions, which show that $J(u_h)$ converges to $J(u)$ with order h^{2k+1} and that the new approximations converges

Dedicated to Chi-Wang Shu on the occasion of his 60-th birthday.

Research supported by the U.S. National Science Foundation Grants DMS-1522657 and DMS-1522672.

✉ Bernardo Cockburn
cockburn@math.umn.edu

Zhu Wang
wangzhu@math.sc.edu

[1] School of Mathematics, University of Minnesota, 206 Church St SE, Minneapolis, MN 55455, USA

[2] Department of Mathematics, University of South Carolina, 1523 Greene Street, Columbia, SC 29208, USA

with order h^{4k}. The numerical experiments also indicate, for the p-version of the method, that the rate of exponential convergence of the new approximations is about twice that of $J(u_h)$.

Keywords Approximation of linear functionals · Adjoint-based error correction · Galerkin methods · Filtering · Convolution

Mathematics Subject Classification 35J47 · 65N12 · 65N30

1 Introduction

This is the first of a series of papers devoted to devising techniques for using Galerkin approximations to define superconvergent approximations to functionals. In many engineering applications such as flow control and optimization, it is more important to obtain accurate approximations of certain functionals $J(\cdot)$ of the state variables u than to get accurate approximations of the variables themselves. These functionals are useful in describing quantities of interest like, for example, significant physical parameters of a dynamical system, the mean value in a domain, or the flux crossing certain boundary. See the 2002 paper by Giles and Süli [11] which contains a thorough overview of these *adjoint* methods. Here, we consider the problem of approximating the model functional

$$J(u) := (g, u)_\Omega := \int_\Omega g(x)\, u(x)\, dx,$$

where u is the solution of a second-order elliptic problem

$$-\Delta u = f \quad \text{in } \Omega, \tag{1.1a}$$

$$u = u_D \quad \text{on } \partial\Omega, \tag{1.1b}$$

in terms of a Galerkin approximation u_h to the state variable u and show that, by *only* doubling the computational effort needed for computing $J(u_h)$, a new approximation can be obtained which is *significantly* closer to $J(u)$ than $J(u_h)$.

This new technique is based on a combination of the adjoint-recovery method for approximating functionals obtained by Pierce and Giles in 2000 [14] and the accuracy-enhancing convolution method proposed by Bramble and Schatz in 1977 [1]. Next, we describe these methods for the functional $J(u)$ just introduced. However, note that the two above-mentioned methods are general enough as to be applicable to very general functionals, and not only to a wide variety of partial differential equations but to a wide class of Galerkin numerical approximations including those provided by the mixed methods, by all *adjoint-consistent* DG methods and by the classic continuous Galerkin methods.

1.1 The Adjoint Error Correction Method

Let us begin by describing the adjoint error correction method by Pierce and Giles [14]. If u_h is any $H^1(\Omega)$ approximation to u such that $u_h = u_D$ on $\partial\Omega$, we can write that

$$\begin{aligned} J(u) &= (u, g)_\Omega \\ &= (u_h, g)_\Omega + (u - u_h, g)_\Omega \\ &= (u_h, g)_\Omega + (u - u_h, -\Delta v)_\Omega \end{aligned}$$

$$\begin{aligned}
&= (u_h, g)_\Omega + (\nabla(u - u_h), \nabla v)_\Omega \\
&= (u_h, g)_\Omega + (\nabla(u - u_h), \nabla v_h)_\Omega + (\nabla(u - u_h), \nabla(v - v_h))_\Omega,
\end{aligned}$$

where v is the solution of the adjoint problem

$$-\Delta v = g \quad \text{in } \Omega, \tag{1.2a}$$

$$v = 0 \quad \text{on } \partial\Omega, \tag{1.2b}$$

and v_h is any $H^1(\Omega)$ approximation to v.

Thus, if we take, as approximation to $J(u)$, not $J(u_h)$ but

$$J_h(u_h, v_h) := J(u_h) + (\nabla(u - u_h), \nabla v_h)_\Omega,$$

we obtain

$$|J(u) - J_h(u_h, v_h)| \le \|\nabla(u - u_h)\|_{L^2(\Omega)} \|\nabla(v - v_h)\|_{L^2(\Omega)}.$$

Therefore, if we are willing to pay the price of computing an approximation v_h to v (which essentially *doubles* the computational effort) in order to incorporate the *adjoint-correction* term

$$AC_h := (\nabla(u - u_h), \nabla v_h)_\Omega,$$

into the approximation of the functional, we can, remarkably enough, *double* the order of convergence of the approximation since the *approximation error* is

$$J(u) - J_h(u_h, v_h) = E_h := (\nabla(u - u_h), \nabla(v - v_h))_\Omega,$$

because

$$|J(u) - J(u_h)| \le \|\nabla(u - u_h)\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)}.$$

This is the adjoint-correction technique proposed by Pierce and Giles [14].

1.2 Extension to Galerkin Methods

Note that the adjoint-correction term is zero if the method defining u_h has the well-known Galerkin orthogonality property. Indeed, in this case

$$AC_h = (\nabla(u - u_h), \nabla v_h)_\Omega = 0.$$

In general, the adjoint-correction terms are different from zero for numerical methods without a Galerkin orthogonality property. Because of this, one might conclude that it is not possible to obtain better approximations of the functional under consideration if u_h satisfies the Galerkin orthogonality property. However, if we could use the finite element approximations u_h and v_h to *efficiently* compute approximations u_h^* and v_h^* which converge faster to u and v that u_h and v_h, respectively, the new approximation, $J_h(u_h^*, v_h^*)$, would converge faster than $J_h(u_h, v_h)$ since

$$|J(u) - J_h(u_h^*, v_h^*)| \le \|\nabla(u - u_h^*)\|_{L^2(\Omega)} \|\nabla(v - v_h^*)\|_{L^2(\Omega)}.$$

The first result of this type was obtained by Pierce and Giles in their original 2000 work [14]. They considered the model problem (1.1) with Ω a square, defined u_h as the continuous finite element solution using piecewise bilinear elements, and took the postprocessing u_h^* as the bicubic spline interpolation through the computed nodal values. They did obtain that $J_h(u_h^*, v_h^*)$ converges with a rate of order $\mathcal{O}(h^4)$. Later in 2004, Giles et al. [10], remarkably enough, extended this result to unstructured meshes made of triangles. On the other hand, Pierce and Giles ended their 2000 paper [14] wondering if there was a systematic way of doing this for $k > 1$ and for other partial differential equations.

An effort to answer such question was carried out by Cockburn and Ichikawa in 2007 [7] in the framework of ordinary differential equations and one-dimensional convection-diffusion equations; discontinuous Galerkin (DG) approximations u_h and v_h were used. For these two problems, the approximation $J_h(u_h^*, v_h^*)$ was proven to converge with a rate of order $\mathcal{O}(h^{4k})$ when polynomials of degree k were used to define u_h and v_h. The convergence properties of the locally computed functions u_h^* and v_h^* are based on the superconvergence properties of the numerical traces of the DG methods obtained, for ODEs, back in 1981 [9] and, for the above-mentioned one-dimensional problem, in 2007 [2]. Unfortunately, these properties do not hold in the multidimensional case and so this approach cannot be used in that case.

1.3 Filtering by Convolution

Fortunately, in the multidimensional case, there is a technique that allows us to locally post-process an approximation u_h defined by Galerkin methods to get a better approximation u_h^*. It is well known that functions satisfying a Galerkin orthogonality property must oscillate around the exact solutions in a fixed pattern. If the mesh is *translation invariant*, it is possible to filter out these oscillations and achieve a much better approximation in a very efficient manner. Indeed, in 1977, Bramble and Schatz [1] showed that if u_h is given by a Galerkin method converging with order h^{k+1}, a simple, local convolution converges with order h^{2k} in the interior of the domain. Applications of this technique to discontinuous Galerkin methods for linear hyperbolic problems was carried out in 2003 by Cockburn et al. [8]. In 1977, Thomée [18] showed how to use the technique to approximate derivatives of any order and in 2009, Ryan and Cockburn [17] applied Thomée's approach to discontinuous Galerkin methods for hyperbolic problems.

In 2010, Ichikawa [12] considered our model elliptic problem and proposed for the first time to use the 1977 accuracy-enhancing local filter (by convolution) of Bramble and Schatz [1] to compute the functions u_h^* and v_h^* from the approximations u_h and v_h, respectively. He used the hybridizable discontinuous Galerkin (HDG) [5,6] with piecewise polynomial approximations of degree k to define u_h and v_h, and then applied a convolution, using both the symmetric convolution kernel advocated by Bramble and Schatz [1] and the non-symmetric ones proposed by Ryan and Shu [15], to define u_h^* and v_h^* in the whole two-dimensional square domain Ω. Ichikawa's [12] numerical results, on uniform meshes made of squares, show that $J_h(u_h^*, v_h^*)$ actually converges with a rate of order $\mathcal{O}(h^{4k})$ for $k = 1, 2, 3$.

1.4 The New Technique

Here, we propose a variation of Ichikawa's way of computing the functions u_h^* and v_h^*. This variation is motivated by the fact that, we can use the symmetric kernels advocated by Bramble and Schatz [1] and Thomée [18] to compute these functions only in a *strict* subdomain Ω_0 of Ω. To compute them in the whole domain Ω, we could use one-sided convolution kernels, see Ryan and Shu [15], to be able to go up to the boundary, as shown by Ichikawa [12] in 2010. However, we prefer not to do that for three reasons. The first is that there is still no theoretical justification that going up to the boundary with one-sided filters results in a high-order approximation. The second is that filters able to deal with curved or very complicated boundaries remain to be developed. The last is that, as Ryan et al. [16] put it: "…*near the*

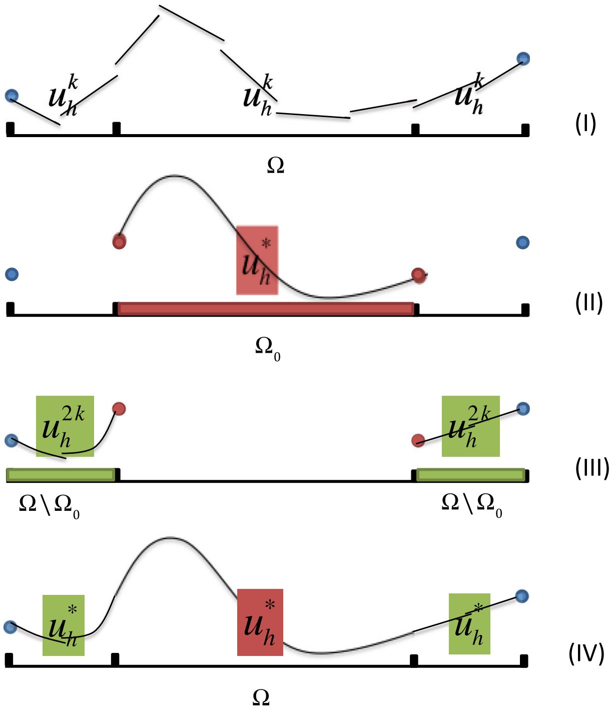

Fig. 1 A schematic of the definition of the function u_h^*

boundaries, where the one-sided post-processor is applied, the error is larger than in the interior, where the symmetric post processor is applied".

Next, we describe how do we compute u_h^*. A schematic illustration of such definition is shown in Fig. 1. To fix ideas, let u_h^ℓ denote the HDG approximation which uses polynomial approximations of degree ℓ on each element; a similar definition holds for any other finite element method. We proceed in four steps as follows:

(I) Using a translation-invariant mesh in most of the domain, we set u_h^k to be the HDG solution of the model problem.
(II) Using the accuracy-enhancing local convolution of Bramble and Schatz [1], we compute u_h^* on the subdomain Ω_0 with a symmetric convolution kernel.
(III) We compute the approximation u_h^{2k} given by the HDG method on the domain $\Omega_1 := \Omega \setminus \overline{\Omega_0}$ where the boundary condition on the boundary of Ω_0 is the trace of u_h^*.
(IV) On Ω_1, we set u_h^* to be the approximation u_h^{2k}.

Note that, for some special domains and boundary data, it is possible to take the set Ω_0 to be Ω, as Bramble and Schatz themselves showed in their original work [1]. In this case, we do not need to carry out the last two steps of the construction of u_h^*. Let us also note that we have tacitly assumed that the set Ω_0 is independent of the mesh and of k. This need not to be the case, as our numerical results show.

The computation of the function v_h^* is obtained in the same manner except that there is *no* need to compute v_h^* outside Ω_0. As we are going to see, the reason is that u_h^* outside Ω_0 is defined by a Galerkin method and so the corresponding contribution to the adjoint-correction term is zero.

To end, let us give the heuristics of why we believe this new algorithm provides what we want. First, note that it can be proven that $J(u_h^k)$ converges with a rate of order h^{2k+1} to $J(u)$ whenever u and v are both very smooth. For the same reasons, u_h^* and v_h^* can also be proven to converge to u and v, respectively, with a rate of order h^{2k+1} in $L^2(\Omega_0)$ and even in $L^\infty(\Omega_0)$, if u and v are sufficiently smooth. This motivates the definition of u_h^* in Ω_1 as u_h^{2k} since we would have that it converges to u with a rate of order h^{2k+1} in $L^2(\Omega_1)$ and even in $L^\infty(\overline{\Omega_1})$, if u and v are sufficiently smooth. In this way, we ensure that u_h^{2k} and v_h^* converge to u and v, respectively, with a rate of order h^{2k+1} in $L^2(\Omega)$ and even in $L^\infty(\Omega)$, and that, as a consequence, $J_h(u_h^*, v_h^*)$ converges to $J(u)$ with a rate of order $\mathcal{O}(h^{4k})$. Our numerical experiments show that this is indeed the case. This is the main contribution of this paper. A rigorous *a priori error* analysis putting in firm mathematical ground of this approach will be carried out elsewhere.

1.5 The Organization of the Paper

The remainder of the paper is organized as follows. In Sect. 2, we present our main result, Theorem 2.1, which provides a systematic way to defining the adjoint-correction approximations $J_h(u_h^*, v_h^*)$ while providing the corresponding approximation errors. It is a simple variation of the approach proposed by Pierce and Giles [14]. We then present two particular cases which provide the new approximations we seek. They differ in how the approximation in Ω_1 is used to define them. One approximation uses the *piecewise gradients* while the other, the approximate *fluxes*. Expressions for their approximation errors are then given in Theorems 2.2 and 2.3, respectively. In Sect. 3, we give detailed proofs of our theorems. In Sect. 4, we provide numerical experiments showing the performance of the approximations. Finally, in Sect. 5, we end with a short discussion on possible variations, extensions, and with a short description of our forthcoming work.

2 Adjoint-Based Super-Convergent Approximations

In this section, we show how to define the adjoint-based approximations $J_h(u_h^*, v_h^*)$. We begin by introducing the Galerkin methods we are going to use, namely, the HDG methods. We then introduce the convolution kernel and recall how to actually compute it. We finally present a general approach for defining the adjoint-based approximations and then give two particular cases.

2.1 The HDG Methods

The HDG method was introduced in the framework of steady-state diffusion by Cockburn et al. in 2009 [6]; see also the recent review in [5]. We use it here because it has a framework general enough which allows us to consider other Galerkin methods.

For any given triangulation of the domain Ω, $\mathcal{T}_h$, made of elements K, we set $\partial\mathcal{T}_h$ to be the set of all boundaries ∂K of the elements K. We say that F is an interior face of the triangulation if there are two elements K^+ and K^- in $\mathcal{T}_h$ such that $F = \partial K^+ \cap \partial K^-$ and its $(d-1)-$Lebesgue measure is not zero; we denote by $\mathcal{F}_h^i$ the set of interior faces associated with the triangulation. We say that F is a boundary face of the triangulation if there is an element $K \in \mathcal{T}_h$ such that $F = \partial K \cap \partial\Omega$ and its $(d-1)-$Lebesgue measure is not zero;

we denote by $\mathcal{F}_h^b = \partial\Omega \cap \partial\mathcal{T}_h$ the set of boundary faces. We denote by $\mathcal{F}_h$ the set of all faces.

On any face $F \in \mathcal{F}_h^i$ such that $F = \partial K^+ \cap \partial K^-$, we denote by $\mathbf{v}^+$ and $\mathbf{v}^-$ the traces of a function $\mathbf{v}$ defined inside K^+ and K^-, respectively. We denote by $\mathbf{n}^\pm$ the unit outward normal to $K^\pm$ and set

$$[\![\mathbf{w}]\!]|_F := \mathbf{w}^- \cdot \mathbf{n}^- + \mathbf{w}^+ \cdot \mathbf{n}^+,$$

for any vector-valued functions $\mathbf{w} \in \boldsymbol{H}^1(\mathcal{T}_h)$. Similarly, for $\widehat{\mathbf{w}} \in \boldsymbol{L}^2(\partial\mathcal{T}_h)$ we set

$$[\![\widehat{\mathbf{w}}]\!]|_F := \widehat{\mathbf{w}}^- \cdot \mathbf{n}^- + \widehat{\mathbf{w}}^+ \cdot \mathbf{n}^+,$$

where $\widehat{\mathbf{w}}^+$ and $\widehat{\mathbf{w}}^-$ are the values of the function $\widehat{\mathbf{w}}$ on ∂K^+ and ∂K^-, respectively. Note that for a function $\widehat{\mathbf{w}} \in \boldsymbol{L}^2(\partial\mathcal{T}_h)$ these two values are not necessarily the same. In contrast, any function $\widehat{\mathbf{w}} \in \boldsymbol{L}^2(\mathcal{F}_h)$ is single valued on any face of the triangulation.

Finally, we set

$$(\cdot,\cdot)_{\mathcal{T}_h} := \sum_{K\in\mathcal{T}_h} (\cdot,\cdot)_K, \quad \langle\cdot,\cdot\rangle_{\partial\mathcal{T}_h} := \sum_{K\in\mathcal{T}_h} \langle\cdot,\cdot\rangle_{\partial K}, \quad \text{and} \quad \langle\cdot,\cdot\rangle_{\mathcal{F}_h} := \sum_{F\in\mathcal{F}_h} \langle\cdot,\cdot\rangle_F,$$

where $(\eta,\zeta)_D$ and $\langle\eta,\zeta\rangle_G$ denote the integral of the product η,ζ on $D \subset \mathbb{R}^d$ and $G \subset \mathbb{R}^{d-1}$, respectively. Note that, for $\zeta \in L^2(\mathcal{F}_h)$, $\mathbf{w} \in \boldsymbol{H}^1(\mathcal{T}_h)$ and $\widehat{\mathbf{w}} \in \boldsymbol{L}^2(\partial\mathcal{T}_h)$, we have

$$\begin{aligned}
\langle \mathbf{w}\cdot\mathbf{n}, \zeta\rangle_{\partial\mathcal{T}_h} &= \langle \mathbf{w}\cdot\mathbf{n}, \zeta\rangle_{\mathcal{F}_h^b} + \langle [\![\mathbf{w}]\!], \zeta\rangle_{\mathcal{F}_h^i},\\
\langle \widehat{\mathbf{w}}\cdot\mathbf{n}, \zeta\rangle_{\partial\mathcal{T}_h} &= \langle \widehat{\mathbf{w}}\cdot\mathbf{n}, \zeta\rangle_{\mathcal{F}_h^b} + \langle [\![\widehat{\mathbf{w}}]\!], \zeta\rangle_{\mathcal{F}_h^i}.
\end{aligned}$$

We are now ready to define the HDG method. It seeks an approximation $(\mathbf{q}_h, u_h, \widehat{u}_h) \in \mathbf{V}_h \times W_h \times M_h(u_D)$ where

$$\begin{aligned}
\mathbf{V}_h &:= \left\{ \mathbf{v} \in \boldsymbol{L}^2(\Omega) : \quad \mathbf{v}|_K \in \mathcal{P}_\ell(K)^d \,\forall K \in \mathcal{T}_h \right\},\\
W_h &:= \left\{ w \in L^2(\Omega) : \quad w|_K \in \mathcal{P}_\ell(K) \,\forall K \in \mathcal{T}_h \right\},\\
M_h(\eta) &:= \left\{ m \in L^2(\mathcal{F}_h) : \, m|_F \in \mathcal{P}_\ell(F) \,\forall F \in \mathcal{F}_h^i, m|_{\partial\Omega} = \eta \right\},
\end{aligned}$$

to the exact solution $(\mathbf{q}|_\Omega, u|_\Omega, u|_{\mathcal{F}_h})$ by considering a global problem for the trace $\widehat{u}_h$ and local problems for $(\mathbf{q}_h, u_h)$. Wherein, the approximation $(\mathbf{q}_h, u_h)$ is expressed in terms of f and $\widehat{u}_h$ by local solvers:

$$\begin{aligned}
(\mathbf{q}_h, \mathbf{r})_{\mathcal{T}_h} - (u_h, \nabla\cdot\mathbf{r})_{\mathcal{T}_h} &= -\langle \widehat{u}_h, \mathbf{r}\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_h}, && \forall\, \mathbf{r} \in \mathbf{V}_h,\\
-(\mathbf{q}_h, \nabla w)_{\mathcal{T}_h} + \langle \widehat{\mathbf{q}}_h\cdot\mathbf{n}, w\rangle_{\partial\mathcal{T}_h} &= (f, w)_{\mathcal{T}_h}, && \forall\, w \in W_h,\\
\widehat{\mathbf{q}}_h\cdot\mathbf{n} &= \mathbf{q}_h\cdot\mathbf{n} + \tau(u_h - \widehat{u}_h) && \text{on } \partial\mathcal{T}_h,
\end{aligned}$$

and $\widehat{u}_h \in M_h(u_D)$ is determined from the global solver by imposing the transmission condition

$$\langle \widehat{\mathbf{q}}_h\cdot\mathbf{n}, \mu\rangle_{\partial\mathcal{T}_h} = \langle [\![\widehat{\mathbf{q}}_h]\!], \mu\rangle_{\mathcal{F}_h^i} = 0 \quad \forall\, \mu \in M_h(0).$$

2.2 Filtering the Errors of a Galerkin Approximation

Here, we recall the definition of the convolution used by Bramble and Schatz [1] and show that to compute it on any element, we only have to carry out a matrix multiplication with a single matrix which can be obtained offline. We illustrate this in the one-dimensional case.

Let us first describe the kernel of the convolution. The convolution uses a symmetric kernel which is a linear combination of B-splines and is defined as follows. Denote by χ a function which is one on the interval $(-\frac{1}{2}, \frac{1}{2})$ and zero outside of it, and set $\psi^{(0)} = \delta$, where δ is the Dirac delta function. The B-spline of n-th order is $\psi^{(n)} = \psi^{(n-1)} * \chi$. Then, given u_h, the convolution is $u_h^* := K_h * u_h(x)$, where $K_h := \frac{1}{h} K\left(\frac{x}{h}\right)$ and

$$K(x) = \sum_{\gamma=\gamma_1}^{\gamma_2} C_\gamma \psi^{(k+1)}(x-\gamma),$$

where h is the mesh size. For a symmetric kernel, $-\gamma_1 = \gamma_2 = k$. The coefficients of the kernel C_γ are determined by requiring that $(x^l * K)(x) = x^l$ for $l = 0, \ldots, 2k$. A simple calculation gives the system of equations of C_γ

$$\sum_{\gamma=\gamma_1}^{\gamma_2} C_\gamma \int_{-\infty}^{\infty} y^l \psi^{(k+1)}(y-\gamma)\,\mathrm{d}y = \begin{cases} 1 & \text{if } l = 0, \\ 0 & \text{if } l = 1, \ldots, 2k. \end{cases} \tag{2.1}$$

Note that the support of $\psi^{(k+1)}(\cdot)$ is $[-\frac{k+1}{2}, \frac{k+1}{2}]$. As a consequence, the support of the kernel K_h involves only a fixed number of elements.

Finally, let us point out that another way of computing the kernels is provided by Thomée in 1977 [18] by using Fourier techniques. It can be easily implemented in any symbolic manipulator.

Now, suppose u_h is an approximate solution to a one-dimensional problem on a uniform mesh. Let us show how to compute $u_h^*(x)$ for any x in the interval $I_i = (x_{i-1}, x_i)$. If there are N intervals, then u_h can be written in the following expansion form:

$$u_h(x) = \sum_{\ell=1}^{N} \sum_{m=1}^{N_p} u_m^\ell l_m \left(\frac{x - x_{\ell-1/2}}{h/2}\right) \chi_{I_\ell}(x),$$

where l_m is the Legendre polynomial of degree $m-1$ and $x_{\ell-1/2}$ is the midpoint of the interval I_ℓ, and $N_p = k+1$.

For $x \in I_i$, we have

$$\begin{aligned} K_h * u_h(x) &= \int_{-\infty}^{\infty} \frac{1}{h} \sum_{\gamma=\gamma_1}^{\gamma_2} C_\gamma \psi^{(k+1)}\left(\frac{x-y}{h} - \gamma\right) u_h(y)\,\mathrm{d}y \\ &= \sum_{\gamma=\gamma_1}^{\gamma_2} \frac{C_\gamma}{h} \int_{x_{\ell-1}}^{x_\ell} \psi^{(k+1)}\left(\frac{x-y}{h} - \gamma\right) \sum_{\ell=1}^{N} \sum_{m=1}^{N_p} u_m^\ell l_m \left(\frac{y - x_{\ell-1/2}}{h/2}\right) \mathrm{d}y \\ &= \sum_{\gamma=\gamma_1}^{\gamma_2} \sum_{m=1}^{N_p} \sum_{\ell=1}^{N} \frac{C_\gamma}{2} u_m^\ell \int_{-1}^{1} \psi^{(k+1)}\left(\frac{x - x_{\ell-1/2} - \frac{h}{2}r}{h} - \gamma\right) l_m(r)\,\mathrm{d}r, \end{aligned}$$

where $r \in (-1, 1)$ is defined by $y = x_{\ell-1/2} + \frac{h}{2}r$.

Now, for $x \in I_i$, we have that $\xi \in (-1, 1)$ when we define it by $x = \frac{h}{2}\xi + x_{i-1/2}$. Then

$$\frac{x - x_{\ell-1/2}}{h} = \frac{x - x_{i-1/2}}{h} + \frac{x_{i-1/2} - x_{\ell-1/2}}{h} = \frac{\xi}{2} + i - \ell,$$

and, setting $j := \ell - i$, we get

$$K_h * u_h(x) = \sum_{\gamma=\gamma_1}^{\gamma_2} \sum_{m=1}^{N_p} \sum_{j=-2k}^{2k} \frac{C_\gamma}{2} u_m^\ell \int_{-1}^{1} \psi^{(k+1)}\left(\frac{\xi}{2} - j - \frac{r}{2} - \gamma\right) l_m(r)\mathrm{d}r.$$

Note that we are taking $-2k \leq j \leq 2k$ because, for the values of j outside this range, the function $\psi^{(k+1)}$ is zero at $\left(\frac{\xi}{2} - j - \frac{r}{2} - \gamma\right)$.

In our implementation, we need to evaluate $K_h * u_h$ at N_q quadrature points x_n on the Interval I_i. This means that we *only* have to precompute a *single* array, independent of the elements, namely,

$$A_{n,j,s,m} = \int_{-1}^{1} \psi^{(k+1)} \left(\frac{\xi_n}{2} - j - \frac{r}{2} - \gamma_s\right) l_m(r) \mathrm{d}r, \tag{2.2}$$

where $j = -2k, \ldots, 2k$, $s = -k, \ldots, k$, and $m, n = 1, \ldots, N_p$. More details on the efficient implementation of this convolution can be found in the 2012 paper by Mirzaee et al. [13].

Algorithm 1: Calculation of the coefficient matrix for $K_h * u_h$

```
for $\xi_n$, $n = 1, \ldots, N_q$ ;          /* # of quadrature points on the element */
do
    for $j = -2k, \ldots, 2k$ ;                                          /* Neighbors */
    do
        Calculate $\begin{bmatrix} A_{n,j,-k,1} & A_{n,j,-k+1,1} & \cdots & A_{n,j,k,1} \\ A_{n,j,-k,2} & A_{n,j,-k+1,2} & \cdots & A_{n,j,k,2} \\ \vdots & \vdots & \ldots & \vdots \\ A_{n,j,-k,N_p} & A_{n,j,-k+1,N_p} & \cdots & A_{n,j,k,N_p} \end{bmatrix}$ ;
    end
end
```

2.3 A General Approach to Getting Adjoint-Based Approximations

The choices for the new approximations we seek are in fact particular cases of a general, but simple result which provides both the definition of the approximation and its corresponding error. As we pointed out in the Introduction, it is a simple variation of the approach by Pierce and Giles [14].

Theorem 2.1 *For any functions*

$$(\mathbf{q}_h, \mathsf{u}_h, \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \widehat{\mathsf{u}}_h), (\mathbf{p}_h, \mathsf{v}_h, \widehat{\mathbf{p}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h) \in \boldsymbol{L}^2(\mathcal{T}_h) \times H^1(\mathcal{T}_h) \times L^2(\partial \mathcal{T}_h) \times L^2(\mathcal{F}_h),$$

such that $\widehat{\mathsf{u}}_h = u$ *and* $\widehat{\mathsf{v}}_h = 0$ *on* $\partial\Omega$*, we have that*

$$J(u) := (u, g)_\Omega = J(\mathsf{u}_h) + AC_h + E_h,$$

where

$$\begin{aligned} AC_h := {} & (f, \mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h, \nabla \mathsf{v}_h)_{\mathcal{T}_h} - \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \mathsf{v}_h \rangle_{\partial \mathcal{T}_h} \\ & + (\mathbf{q}_h + \nabla \mathsf{u}_h, \mathbf{p}_h)_{\mathcal{T}_h} - \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \mathbf{p}_h \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} \\ & + \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h \rangle_{\partial \mathcal{T}_h \setminus \partial\Omega} \\ & + \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, (\mathbf{p}_h - \widehat{\mathbf{p}}_h) \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h}, \\ E_h := {} & (\boldsymbol{q} - \mathbf{q}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} \\ & + (\boldsymbol{q} - \mathbf{q}_h, \mathbf{p}_h + \nabla \mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h + \nabla \mathsf{u}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} \\ & + \langle (\widehat{\mathbf{q}}_h - \boldsymbol{q}) \cdot \mathbf{n}, \mathsf{v}_h - \widehat{\mathsf{v}}_h \rangle_{\partial \mathcal{T}_h} + \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, (\widehat{\mathbf{p}}_h - \boldsymbol{p}) \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h}. \end{aligned}$$

Let us briefly discuss this identity. Note that this result holds for arbitrary functions

$$(\boldsymbol{\mathsf{q}}_h, \mathsf{u}_h, \widehat{\boldsymbol{\mathsf{q}}}_h \cdot \mathbf{n}, \widehat{\mathsf{u}}_h) \quad \text{and} \quad (\boldsymbol{\mathsf{p}}_h, \mathsf{v}_h, \widehat{\boldsymbol{\mathsf{p}}}_h \cdot \mathbf{n}.\widehat{\mathsf{v}}_h).$$

This means that we can use *any* numerical approximation to define these functions; the use of HDG methods is not required. Of course, these functions should be approximations of

$$(\boldsymbol{q}|_\Omega, u|_\Omega, \boldsymbol{q} \cdot \mathbf{n}|_{\partial\mathcal{T}_h}, u|_{\mathcal{F}_h}) \quad \text{and} \quad (\boldsymbol{p}|_\Omega, v|_\Omega, \boldsymbol{p} \cdot \mathbf{n}|_{\partial\mathcal{T}_h}, v|_{\mathcal{F}_h})$$

respectively, if we expect the *error* term E_h to be *small*. Because of this, it is reasonable to take $\widehat{\mathsf{u}}_h = u$ and $\widehat{\mathsf{v}}_h = v = 0$ on $\partial\Omega$, as well as to expect that the functions $\boldsymbol{\mathsf{q}}_h + \nabla \mathsf{u}_h$ and $\boldsymbol{\mathsf{p}}_h + \nabla \mathsf{v}_h$ to be small because $\boldsymbol{q} + \nabla u = \mathbf{0}$ and $\boldsymbol{p} + \nabla v = \mathbf{0}$. These considerations are behind the choices we have taken in order to obtain the approximations presented in the previous subsection.

Note also that we are assuming that the numerical traces $\widehat{\mathsf{u}}_h$ and $\widehat{\mathsf{v}}_h$ lie in $L^2(\mathcal{F}_h)$, which means that they are *single valued* on each face lying on $\mathcal{F}_h$. In contrast, we are *not* making this assumption on the normal components of the numerical traces $\widehat{\boldsymbol{\mathsf{q}}}_h$ and $\widehat{\boldsymbol{\mathsf{p}}}_h$.

Finally, note that the definition of the *adjoint-correction* term AC_h is suggested by the very definition of the HDG methods. Thus, the first two terms correspond to the definition of the so-called local problems and the last term to the transmission condition. The fourth term can be interpreted as the contribution of the lack of conformity of the spaces of the HDG method.

The motivation for defining the adjoint-correction term AC_h in this way is that we want to *extend* the idea that the adjoint-correction term must be different from zero whenever the numerical scheme does not satisfy a "Galerkin orthogonality" property. In our extension, instead of the "Galerkin orthogonality" property, we use the weak formulations defining the scheme both inside the elements and across their boundaries. Moreover, we also consider a term (the fourth term) which is generated by the *lack of conformity* of the numerical methods.

Let us illustrate this result on some simple cases. To consider the case treated in the Introduction, we set

$$(\boldsymbol{\mathsf{q}}_h, \mathsf{u}_h, \widehat{\boldsymbol{\mathsf{q}}}_h \cdot \mathbf{n}, \widehat{\mathsf{u}}_h) := (-\nabla u_h, u_h, -\nabla u_h \cdot \mathbf{n}, u_h),$$
$$(\boldsymbol{\mathsf{p}}_h, \mathsf{v}_h, \widehat{\boldsymbol{\mathsf{p}}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h) := (-\nabla v_h, v_h, -\nabla v_h \cdot \mathbf{n}, v_h),$$

and, since u_h and v_h lie in $\mathcal{C}^1(\Omega)$, we get that

$$AC_h = (f - f_h, v_h)_{\mathcal{T}_h}, \quad E_h = (\nabla(u - u_h), \nabla(v - v_h))_{\mathcal{T}_h}.$$

We thus recover the identity

$$J(u) = J(u_h) + (f - f_h, v_h)_{\mathcal{T}_h} + (\nabla(u - u_h), \nabla(v - v_h))_{\mathcal{T}_h},$$

obtained in the Introduction.

Let us now consider the case in which

$$(\boldsymbol{\mathsf{q}}_h, \mathsf{u}_h, \widehat{\boldsymbol{\mathsf{q}}}_h \cdot \mathbf{n}, \widehat{\mathsf{u}}_h) := (\boldsymbol{q}_h, u_h, \widehat{\boldsymbol{q}}_h \cdot \mathbf{n}, \widehat{u}_h),$$
$$(\boldsymbol{\mathsf{p}}_h, \mathsf{v}_h, \widehat{\boldsymbol{\mathsf{p}}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h) := (\boldsymbol{p}_h, v_h, \widehat{\boldsymbol{p}}_h \cdot \mathbf{n}, \widehat{v}_h),$$

where the above functions are provided by an HDG method for the model and adjoint problems, respectively. In this case, a simple calculation gives that

$$AC_h = \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, (\boldsymbol{\mathsf{p}}_h - \widehat{\boldsymbol{\mathsf{p}}}_h) \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h}.$$

Note that the first three terms of the adjoint correction terms AC_h are zero by the very definition of the HDG methods. This is certainly not an accident, as the adjoint correction term was devised with this in mind, as we pointed out above. The fourth term measures the lack of conformity of the methods. It is zero for any mixed method and for the staggered DG method [3,4].

2.4 Two Adjoint-Based Approximations

Here, we define two adjoint-based approximations to the functional $J(u)$.

2.4.1 Notation

To do that, we begin by introducing some notation. We assume that any given element $K \in \mathcal{T}_h$ is fully included in either Ω_0 or Ω_1 and set, or $j = 0, 1$,

$$\begin{aligned}\mathcal{T}_{jh} &:= \left\{K \in \mathcal{T}_h : K \subset \Omega_j\right\},\\ \partial\mathcal{T}_{jh} &:= \left\{\partial K \in \mathcal{T}_h : K \subset \Omega_j\right\},\\ \mathcal{F}_{jh} &:= \left\{F \in \mathcal{F}(K) : K \in \mathcal{T}_{jh}\right\}.\end{aligned}$$

We denote by

$$\left(\boldsymbol{q}_h^k, u_h^k, \widehat{\boldsymbol{q}}_h^k \cdot \mathbf{n}, \widehat{u}_h^k\right) \quad \text{and} \quad \left(\boldsymbol{p}_h^k, v_h^k, \widehat{\boldsymbol{p}}_h^k \cdot \mathbf{n}, \widehat{v}_h^k\right),$$

the HDG approximations, using polynomials of degree $\ell := k$, of the model and adjoint problems, respectively, on the whole domain Ω. We also denote by

$$\left(\boldsymbol{q}_h^{2k}, u_h^{2k}, \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, \widehat{u}_h^{2k}\right) \quad \text{and} \quad \left(\boldsymbol{p}_h^{2k}, v_h^{2k}, \widehat{\boldsymbol{p}}_h^{2k} \cdot \mathbf{n}, \widehat{v}_h^{2k}\right),$$

the HDG approximations, using polynomials of degree $\ell := 2k$, of the model and adjoint problems, respectively, on the domain Ω_1. The Dirichlet boundary conditions for the model and adjoint problems on $\partial\Omega_1 \setminus \partial\Omega$ are

$$\widehat{u}_h^{2k} = K_h * u_h^k \quad \text{and} \quad \widehat{v}_h^{2k} = K_h * v_h^k.$$

Finally, we denote our approximations by $J_h(u_h^*, v_h^*)$, where

$$u_h^* := \begin{cases} K_h * u_h^k & \text{in } K \in \mathcal{T}_{0h}, \\ u_h^{2k} & \text{in } K \in \mathcal{T}_{1h}, \end{cases} \quad \text{and} \quad v_h^* := \begin{cases} K_h * v_h^k & \text{in } K \in \mathcal{T}_{0h}, \\ v_h^{2k} & \text{in } K \in \mathcal{T}_{1h}, \end{cases}$$

even though, by Theorem 2.1, the approximations do depend on the numerical traces and the approximation of the gradients. The two approximations we present next have the same way of using the information on Ω_0; they use both u_h^* and v_h^* as well as their gradients ∇u_h^* and ∇v_h^* therein. However, the approximations differ only in the way the information on the solution in Ω_1 is used.

2.4.2 An Approximation Using the Piecewise Gradients in Ω_1

Our first approximation uses both u_h^{2k} and v_h^{2k} as well as their piecewise gradients ∇u_h^{2k} and ∇v_h^{2k} on $\mathcal{T}_{1h}$. (This is why we use the superscript $^{\mathrm{G}}$ to denote it.) It is given by

$$J_h^{\mathrm{G}}(u_h^*, v_h^*) := J(u_h^*) + AC_h^{\mathrm{G}},$$

where the adjoint-correction term is

$$\begin{aligned} AC_h^{\mathrm{G}} := & (f, v_h^*)_{\mathcal{T}_h} - (\nabla u_h^*, \nabla v_h^*)_{\mathcal{T}_h} \\ & + \langle \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, \widehat{v}_h^{2k} - v_h^{2k} \rangle_{\partial \mathcal{T}_{1h}} \\ & + \langle \widehat{u}_h^{2k} - u_h^{2k}, \widehat{\boldsymbol{p}}_h^{2k} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_{1h}}, \end{aligned}$$

The expression of the error of this approximation is provided in the following result.

Theorem 2.2 *We have that* $J(u) = J_h^{\mathrm{G}}(u_h^*, v_h^*) + E_h^{\mathrm{G}}$*, where the error of the approximation is*

$$\begin{aligned} E_h^{\mathrm{G}} := & (\boldsymbol{q} + \nabla u_h^*, \boldsymbol{p} + \nabla v_h^*)_{\mathcal{T}_h} \\ & + \langle (\widehat{\boldsymbol{q}}_h^{2k} - \boldsymbol{q}) \cdot \mathbf{n}, v_h^{2k} - \widehat{v}_h^{2k} \rangle_{\partial \mathcal{T}_{1h}} \\ & + \langle u_h^{2k} - \widehat{u}_h^{2k}, (\widehat{\boldsymbol{p}}_h^{2k} - \boldsymbol{p}) \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_{1h}}. \end{aligned}$$

2.4.3 An Approximation Using the Approximate Fluxes in Ω_1

Our second approximation uses the approximate fluxes $\boldsymbol{q}_h^{2k}$ and $\boldsymbol{p}_h^{2k}$ instead piecewise gradients ∇u_h^{2k} and ∇v_h^{2k} in $\mathcal{T}_{1h}$. (This is why we use the superscript $^{\mathrm{F}}$ to denote it.) It is given by

$$J_h^{\mathrm{F}}(u_h^*, v_h^*) := J(u_h^*) + AC_h^{\mathrm{F}},$$

where the adjoint-correction term is

$$\begin{aligned} AC_h^{\mathrm{F}} := & (f, v_h^*)_{\mathcal{T}_{0h}} - (\nabla u_h^*, \nabla v_h^*)_{\mathcal{T}_{0h}} \\ & + \langle \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, \widehat{v}_h^{2k} \rangle_{\partial \Omega_1 \setminus \partial \Omega} \\ & + \langle u_h^{2k} - \widehat{u}_h^{2k}, (\boldsymbol{p}_h^{2k} - \widehat{\boldsymbol{p}}_h^{2k}) \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_{1h}}. \end{aligned}$$

Note that the contribution to the adjoint-correction term of the approximation in Ω_1 can be expressed only in terms of boundary integrals which are easier to compute that volume integrals.

The error of this approximation is characterized in the following result.

Theorem 2.3 *We have* $J(u) = J_h^{\mathrm{F}}(u_h^*, v_h^*) + E_h^{\mathrm{F}}$*, where the error of the approximation is*

$$\begin{aligned} E_h^{\mathrm{F}} := & (\boldsymbol{q} + \nabla u_h^*, \boldsymbol{p} + \nabla v_h^*)_{\mathcal{T}_{0h}} \\ & + (\boldsymbol{q} - \boldsymbol{q}_h^{2k}, \boldsymbol{p} - \boldsymbol{p}_h^{2k})_{\mathcal{T}_{1h}} \\ & + (\boldsymbol{q} - \boldsymbol{q}_h^{2k}, \boldsymbol{p}_h^{2k} + \nabla v_h^{2k})_{\mathcal{T}_{1h}} + (\boldsymbol{q}_h^{2k} + \nabla u_h^{2k}, \boldsymbol{p} - \boldsymbol{p}_h^{2k})_{\mathcal{T}_{1h}} \\ & + \langle (\widehat{\boldsymbol{q}}_h^{2k} - \boldsymbol{q}) \cdot \mathbf{n}, v_h^{2k} - \widehat{v}_h^{2k} \rangle_{\partial \mathcal{T}_{1h}} + \langle u_h^{2k} - \widehat{u}_h^{2k}, (\widehat{\boldsymbol{p}}_h^{2k} - \boldsymbol{p}) \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_{1h}}. \end{aligned}$$

3 Proofs

This Section is devoted to providing detailed proofs of the expressions and corresponding approximation errors for our two approximations, Theorems 2.2 and 2.3. Since these theorems are *particular* cases of Theorem 2.1, we begin by proving it.

3.1 Proof of the Identity for $J(u)$ of Theorem 2.1

To prove Theorem 2.1, we begin by noting that, if we use the fact that $-\Delta v = g$ and set $\boldsymbol{p} = -\nabla v$, we easily get that

$$\begin{aligned}
J(u) &= J(\mathsf{u}_h) + (u - \mathsf{u}_h, \nabla \cdot \boldsymbol{p})_{\mathcal{T}_h}\\
&= J(\mathsf{u}_h) - (\nabla(u - \mathsf{u}_h), \boldsymbol{p})_{\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}\\
&= J(\mathsf{u}_h) + (\boldsymbol{q} + \nabla\mathsf{u}_h, \boldsymbol{p})_{\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h},
\end{aligned}$$

since $\boldsymbol{q} = -\nabla u$. Now, we perform very simple algebraic manipulations in order to exploit the fact that we expect the functions $\mathbf{q}_h + \nabla\mathsf{u}_h$ and $\mathbf{p}_h + \nabla\mathsf{v}_h$ to be *small*. We have

$$\begin{aligned}
J(u) - J(\mathsf{u}_h) &= (\boldsymbol{q} - \mathbf{q}_h, \boldsymbol{p})_{\mathcal{T}_h} + (\mathbf{q}_h + \nabla\mathsf{u}_h, \boldsymbol{p})_{\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}\\
&= (\boldsymbol{q} - \mathbf{q}_h, \mathbf{p}_h)_{\mathcal{T}_h} + (\mathbf{q}_h + \nabla\mathsf{u}_h, \mathbf{p}_h)_{\mathcal{T}_h}\\
&\quad + (\boldsymbol{q} - \mathbf{q}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} + (\mathbf{q}_h + \nabla\mathsf{u}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}\\
&= -(\boldsymbol{q} - \mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h + \nabla\mathsf{u}_h, \mathbf{p}_h)_{\mathcal{T}_h} + (\boldsymbol{q} - \mathbf{q}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h}\\
&\quad + (\mathbf{q}_h + \nabla\mathsf{u}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} + (\boldsymbol{q} - \mathbf{q}_h, \mathbf{p}_h + \nabla\mathsf{v}_h)_{\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}.
\end{aligned}$$

Next, we suitably rewrite the second term of the above right-hand side. Integrating by parts and using the fact that $\nabla \cdot \boldsymbol{q} = f$, we get

$$\begin{aligned}
-(\boldsymbol{q} - \mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h} &= -(\boldsymbol{q}, \nabla\mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h}\\
&= (f, \mathsf{v}_h)_{\mathcal{T}_h} - \langle \boldsymbol{q} \cdot \mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_h} + (\mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h}\\
&= (f, \mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h} - \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_h} + \langle (\widehat{\mathbf{q}}_h - \boldsymbol{q}) \cdot \mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_h}\\
&= (f, \mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h} - \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_h}\\
&\quad + \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} + \langle (\widehat{\mathbf{q}}_h - \boldsymbol{q}) \cdot \mathbf{n}, \mathsf{v}_h - \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} - \langle \boldsymbol{q} \cdot \mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h}.
\end{aligned}$$

Using this expression and rearranging terms, we get

$$\begin{aligned}
J(u) - J(\mathsf{u}_h) &= (f, \mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h} - \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_h}\\
&\quad + (\mathbf{q}_h + \nabla\mathsf{u}_h, \mathbf{p}_h)_{\mathcal{T}_h}\\
&\quad + \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h}\\
&\quad + (\boldsymbol{q} - \mathbf{q}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h}\\
&\quad + (\mathbf{q}_h + \nabla\mathsf{u}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} + (\boldsymbol{q} - \mathbf{q}_h, \mathbf{p}_h + \nabla\mathsf{v}_h)_{\mathcal{T}_h}\\
&\quad + \langle (\widehat{\mathbf{q}}_h - \boldsymbol{q}) \cdot \mathbf{n}, \mathsf{v}_h - \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h}\\
&\quad - \langle \boldsymbol{q} \cdot \mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}.
\end{aligned}$$

Next, we add and subtract several terms to obtain

$$\begin{aligned}
J(u) - J(\mathsf{u}_h) &= (f, \mathsf{v}_h)_{\mathcal{T}_h} + (\mathbf{q}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_h} - \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_h}\\
&\quad + (\mathbf{q}_h + \nabla\mathsf{u}_h, \mathbf{p}_h)_{\mathcal{T}_h} - \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \mathbf{p}_h \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}\\
&\quad + \langle \widehat{\mathbf{q}}_h \cdot \mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h}\\
&\quad + \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \mathbf{p}_h \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h} - \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \widehat{\mathbf{p}}_h \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}\\
&\quad + (\boldsymbol{q} - \mathbf{q}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h}\\
&\quad + (\mathbf{q}_h + \nabla\mathsf{u}_h, \boldsymbol{p} - \mathbf{p}_h)_{\mathcal{T}_h} + (\boldsymbol{q} - \mathbf{q}_h, \mathbf{p}_h + \nabla\mathsf{v}_h)_{\mathcal{T}_h}\\
&\quad + \langle (\widehat{\mathbf{q}}_h - \boldsymbol{q}) \cdot \mathbf{n}, \mathsf{v}_h - \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} + \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \widehat{\mathbf{p}}_h \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h} - \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h}\\
&\quad - \langle \boldsymbol{q} \cdot \mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} + \langle u - \mathsf{u}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h} - \langle \mathsf{u}_h - \widehat{\mathsf{u}}_h, \boldsymbol{p} \cdot \mathbf{n}\rangle_{\partial\mathcal{T}_h},
\end{aligned}$$

and so,

$$J(u) = J(\mathsf{u}_h) + AC_h + E_h - \langle \boldsymbol{q}\cdot\mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} + \langle u - \widehat{\mathsf{u}}_h, \boldsymbol{p}\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_h}$$

Note that so far, we have not used any property of continuity across inter-element boundaries or any property of single-valuedness of the numerical traces. However, now we are going to use the fact that $\widehat{\mathsf{u}}_h$ and $\widehat{\mathsf{v}}_h$ are single valued, since they belong to $L^2(\mathcal{F}_h)$, to conclude that

$$\begin{aligned}\langle u - \widehat{\mathsf{u}}_h, \boldsymbol{p}\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_h} &= \langle u - \widehat{\mathsf{u}}_h, \boldsymbol{p}\cdot\mathbf{n}\rangle_{\partial\Omega} = 0,\\ \langle \boldsymbol{q}\cdot\mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_h} &= \langle \boldsymbol{q}\cdot\mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\Omega} = 0,\end{aligned}$$

because $\widehat{\mathsf{u}}_h = u$ and $\widehat{\mathsf{v}}_h = 0$ on $\partial\Omega$.

This completes the proof of Theorem 2.1.

3.2 Proof of the Identities Defining the Two Approximations

We are now ready to prove Theorems 2.2 and 2.3. We proceed in several steps.

Step 1

We apply Theorem 2.1 with $\mathsf{u}_h := u_h^*$ and $\mathsf{v}_h := v_h^*$, so that we do have that $J(u) = J_h(u_h^*, v_h^*) + E_h$ where $J_h(u_h^*, v_h^*) := J(u_h^*) + AC_h$ with $AC_h = AC_{0h} + AC_{1h}$, where

$$\begin{aligned}AC_{0h} :=\ & (f, \mathsf{v}_h)_{\mathcal{T}_{0h}} + (\boldsymbol{\mathsf{q}}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_{0h}} - \langle\widehat{\boldsymbol{\mathsf{q}}}_h\cdot\mathbf{n}, (\mathsf{v}_h - \widehat{\mathsf{v}}_h)\rangle_{\partial\mathcal{T}_{0h}}\\ & + (\boldsymbol{\mathsf{q}}_h + \nabla\mathsf{u}_h, \boldsymbol{\mathsf{p}}_h)_{\mathcal{T}_{0h}} - \langle\mathsf{u}_h - \widehat{\mathsf{u}}_h, \widehat{\boldsymbol{\mathsf{p}}}_h\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_{0h}},\\ AC_{1h} :=\ & (f, \mathsf{v}_h)_{\mathcal{T}_{1h}} + (\boldsymbol{\mathsf{q}}_h, \nabla\mathsf{v}_h)_{\mathcal{T}_{1h}} - \langle\widehat{\boldsymbol{\mathsf{q}}}_h\cdot\mathbf{n}, \mathsf{v}_h\rangle_{\partial\mathcal{T}_{1h}}\\ & + (\boldsymbol{\mathsf{q}}_h + \nabla\mathsf{u}_h, \boldsymbol{\mathsf{p}}_h)_{\mathcal{T}_{1h}} - \langle\mathsf{u}_h - \widehat{\mathsf{u}}_h, \boldsymbol{\mathsf{p}}_h\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_{1h}}\\ & + \langle\widehat{\boldsymbol{\mathsf{q}}}_h\cdot\mathbf{n}, \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_{1h}\setminus\partial\Omega}\\ & + \langle\mathsf{u}_h - \widehat{\mathsf{u}}_h, (\boldsymbol{\mathsf{p}}_h - \widehat{\boldsymbol{\mathsf{p}}}_h)\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_{1h}},\end{aligned}$$

and $E_h = E_{0h} + E_{1h}$, where

$$\begin{aligned}E_{0h} :=\ & (\boldsymbol{q} - \boldsymbol{\mathsf{q}}_h, \boldsymbol{p} - \boldsymbol{\mathsf{p}}_h)_{\mathcal{T}_{0h}}\\ & + (\boldsymbol{q} - \boldsymbol{\mathsf{q}}_h, \boldsymbol{\mathsf{p}}_h + \nabla\mathsf{v}_h)_{\mathcal{T}_{0h}} + (\boldsymbol{\mathsf{q}}_h + \nabla\mathsf{u}_h, \boldsymbol{p} - \boldsymbol{\mathsf{p}}_h)_{\mathcal{T}_{0h}}\\ & + \langle(\widehat{\boldsymbol{\mathsf{q}}}_h - \boldsymbol{q})\cdot\mathbf{n}, \mathsf{v}_h - \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_{0h}} + \langle\mathsf{u}_h - \widehat{\mathsf{u}}_h, (\widehat{\boldsymbol{\mathsf{p}}}_h - \boldsymbol{p})\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_{0h}},\\ E_{1h} :=\ & (\boldsymbol{q} - \boldsymbol{\mathsf{q}}_h, \boldsymbol{p} - \boldsymbol{\mathsf{p}}_h)_{\mathcal{T}_{1h}}\\ & + (\boldsymbol{q} - \boldsymbol{\mathsf{q}}_h, \boldsymbol{\mathsf{p}}_h + \nabla\mathsf{v}_h)_{\mathcal{T}_{1h}} + (\boldsymbol{\mathsf{q}}_h + \nabla\mathsf{u}_h, \boldsymbol{p} - \boldsymbol{\mathsf{p}}_h)_{\mathcal{T}_{1h}}\\ & + \langle(\widehat{\boldsymbol{\mathsf{q}}}_h - \boldsymbol{q})\cdot\mathbf{n}, \mathsf{v}_h - \widehat{\mathsf{v}}_h\rangle_{\partial\mathcal{T}_{1h}} + \langle\mathsf{u}_h - \widehat{\mathsf{u}}_h, (\widehat{\boldsymbol{\mathsf{p}}}_h - \boldsymbol{p})\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_{1h}}.\end{aligned}$$

Step 2

If u_h and v_h lie on $\mathcal{C}^1(\overline{\Omega}_0)$, we can take

$$\begin{aligned}\boldsymbol{\mathsf{q}}_h &:= -\nabla\mathsf{u}_h, & \boldsymbol{\mathsf{p}}_h &:= -\nabla\mathsf{v}_h & &\text{in } \mathcal{T}_{0h},\\ \widehat{\mathsf{u}}_h &:= \mathsf{u}_h, & \widehat{\mathsf{v}}_h &:= \mathsf{v}_h & &\text{on } \mathcal{F}_{0h},\end{aligned}$$

and obtain that

$$\begin{aligned} AC_{0h} &= (f, \mathsf{v}_h)_{\mathcal{T}_{0h}} - (\nabla \mathsf{u}_h, \nabla \mathsf{v}_h)_{\mathcal{T}_{0h}} = AC^{\mathrm{G}}_{0h} = AC^{\mathrm{F}}_{0h},\\ E_{0h} &= (\boldsymbol{q} + \nabla \mathsf{u}_h, \boldsymbol{p} + \nabla \mathsf{v}_h)_{\mathcal{T}_{0h}} \qquad = E^{\mathrm{G}}_{0h} = E^{\mathrm{F}}_{0h}, \end{aligned}$$

with the obvious notation.

Step 3

Now, if we take

$$\begin{array}{rlrll} \mathsf{u}_h &:= u_h^* := u_h^{2k}, & \mathsf{v}_h &:= v_h^* := v_h^{2k} & \text{in } \mathcal{T}_{1h},\\ \boldsymbol{\mathsf{q}}_h &:= -\nabla u_h^{2k}, & \boldsymbol{\mathsf{p}}_h &:= -\nabla v_h^{2k} & \text{in } \mathcal{T}_{1h},\\ \widehat{\boldsymbol{\mathsf{q}}}_h \cdot \mathbf{n} &:= \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, & \widehat{\boldsymbol{\mathsf{p}}}_h \cdot \mathbf{n} &:= \widehat{\boldsymbol{p}}_h^{2k} \cdot \mathbf{n} & \text{on } \partial\mathcal{T}_{1h},\\ \widehat{\mathsf{u}}_h &:= \widehat{u}_h^{2k}, & \widehat{\mathsf{v}}_h &:= \widehat{v}_h^{2k} & \text{on } \mathcal{F}_{1h}, \end{array}$$

we get, taking into account that $\widehat{v}_h^{2k} = 0$ on $\partial\Omega$,

$$\begin{aligned} AC_{1h} &= (f, v_h^*)_{\mathcal{T}_h} - (\nabla u_h^*, \nabla v_h^*)_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, v_h^{2k} \rangle_{\partial\mathcal{T}_{1h}}\\ &\quad + \langle \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, \widehat{v}_h^{2k} \rangle_{\partial\mathcal{T}_{1h}}\\ &\quad - \langle u_h^{2k} - \widehat{u}_h^{2k}, \widehat{\boldsymbol{p}}_h^{2k} \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_{1h}}\\ &= AC^{\mathrm{G}}_{1h},\\ E_{1h} &= (\boldsymbol{q} + \nabla u_h^{2k}, \boldsymbol{p} + \nabla v_h^{2k})_{\mathcal{T}_{1h}}\\ &\quad + \langle (\widehat{\boldsymbol{q}}\, 2k_h - \boldsymbol{q}) \cdot \mathbf{n}, v_h^{2k} - \widehat{v}_h^{2k} \rangle_{\partial\mathcal{T}_{1h}} + \langle u_h^{2k} - \widehat{u}_h^{2k}, (\widehat{\boldsymbol{p}}_h^{2k} - \boldsymbol{p}) \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_{1h}}\\ &= E^{\mathrm{G}}_{1h}, \end{aligned}$$

and the identity of Theorem 2.2 follows.

Step 4

If we now take

$$\begin{array}{rlrll} u_h &:= u_h^* := u_h^{2k}, & v_h &:= v_h^* := v_h^{2k} & \text{in } \mathcal{T}_{1h},\\ \boldsymbol{\mathsf{q}}_h &:= \boldsymbol{q}_h^{2k}, & \boldsymbol{\mathsf{p}}_h &:= \boldsymbol{p}_h^{2k} & \text{in } \mathcal{T}_{1h},\\ \widehat{\boldsymbol{\mathsf{q}}}_h \cdot \mathbf{n} &:= \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, & \widehat{\boldsymbol{\mathsf{p}}}_h \cdot \mathbf{n} &:= \widehat{\boldsymbol{p}}_h^{2k} \cdot \mathbf{n} & \text{on } \partial\mathcal{T}_{1h},\\ \widehat{\mathsf{u}}_h &:= \widehat{u}_h^{2k}, & \widehat{\mathsf{v}}_h &:= \widehat{v}_h^{2k} & \text{on } \mathcal{F}_{1h}, \end{array}$$

we get, by using the definition of the HDG method on Ω_1,

$$\begin{aligned} AC_{1h} &= \langle \widehat{\boldsymbol{q}}_h^{2k} \cdot \mathbf{n}, \widehat{v}_h^{2k} \rangle_{\partial\Omega_1 \setminus \partial\Omega} + \langle u_h^{2k} - \widehat{u}_h^{2k}, (\boldsymbol{p}_h^{2k} - \widehat{\boldsymbol{p}}_h^{2k}) \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_{1h}} = AC^{\mathrm{F}}_{1h},\\ E_{1h} &= (\boldsymbol{q} - \boldsymbol{q}_h^{2k}, \boldsymbol{p} - \boldsymbol{p}_h^{2k})_{\mathcal{T}_{1h}}\\ &\quad + (\boldsymbol{q} - \boldsymbol{q}_h^{2k}, \boldsymbol{p}_h^{2k} + \nabla v_h^{2k})_{\mathcal{T}_{1h}} + (\boldsymbol{q}_h^{2k} + \nabla u_h^{2k}, \boldsymbol{p} - \boldsymbol{p}_h^{2k})_{\mathcal{T}_{1h}}\\ &\quad + \langle (\widehat{\boldsymbol{q}}_h^{2k} - \boldsymbol{q}) \cdot \mathbf{n}, v_h^{2k} - \widehat{v}_h^{2k} \rangle_{\partial\mathcal{T}_{1h}} + \langle u_h^{2k} - \widehat{u}_h^{2k}, (\widehat{\boldsymbol{p}}_h^{2k} - \boldsymbol{p}) \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_{1h}}\\ &= E^{\mathrm{F}}_{1h}, \end{aligned}$$

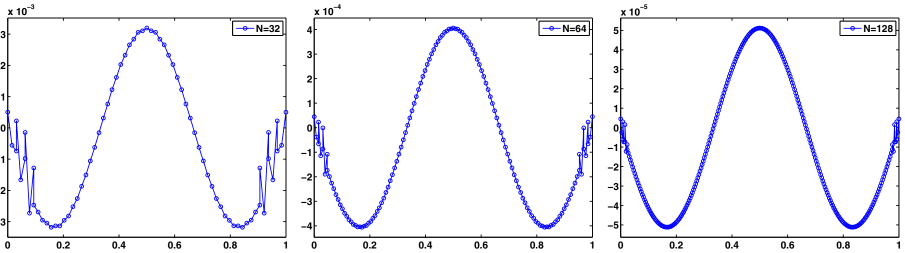

Fig. 2 Approximation errors of the postprocessed solution u_h^* when $k = 1$ and $N = 32, 64$ and 128. Note that the error in the set $\Omega \setminus \Omega_0$, which consists of the three leftmost and three rightmost elements of the mesh, is not bigger than the error in the rest of the domain. Note also that the maximum errors are about $3 \times 10^{-3}, 4 \times 10^{-4}, 5 \times 10^{-5}$, for $N = 32, 64, 128$ which means that the error converges with order 3, as it diminishes by a factor 8 each time N is doubled

and the identity of Theorem 2.3 follows.

This completes the proofs of Theorems 2.2 and 2.3.

4 Numerical Experiments

This section is devoted to testing the convergence properties of the approximations $J_h^{\mathrm{G}}(u_h^*, v_h^*)$ and $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ to $J(u) := (g, u)_\Omega$ whenever the solutions u and v of the model problem (1.1) and its adjoint (1.2), respectively, are *very smooth*. As argued in the introduction, we expect that, when k-th degree of polynomials are used for the HDG approximation, the above-mentioned adjoint-corrected approximations will converge with a rate of order $\mathcal{O}(h^{4k})$. The numerical results we present confirm this expectation.

4.1 The One-Dimensional Case

We consider the model problem (1.1) with $\Omega := (0, 1)$ and take f such that the exact solution of the problem is $u := \sin(3\pi x)$. Moreover, we take $g := 9\pi^2 \sin(3\pi x)$ so that the exact solution to the adjoint problem is $v = \sin(3\pi x)$.

In our implementation, to be able to clearly see the orders of convergence for highly refined meshes, we use MATLAB symbolic toolbox and select 60 digits in the variable precision arithmetic (vpa). We use the HDG method with polynomials of degree k on a uniform mesh of N elements to define u_h and v_h. We define Ω_0 by removing from Ω the $2k + 1$ leftmost and $2k + 1$ rightmost intervals of its mesh.

Let us begin by showing how the function u_h^* converges as we refine the mesh. In Fig. 2, for $k = 1$ and $N = 32, 64$, and 128, we see that the magnitude of the approximation error $u - u_h^*$ decays at the order of $\mathcal{O}(h^{3=2k+1})$, as expected. We also see that the error $u - u_h^*$ behaves differently on Ω_0 and on the region Ω_1. The latter consists of the $3 = 2k+1$ leftmost intervals and the $3 = 2k + 1$ rightmost intervals of the mesh. This reflects the fact that u_h^* is computed differently in each of those sets. Indeed, recall that in Ω_0, u_h^* is $K_h * u_h^k$ whereas in Ω_1, it is u_h^{2k}. Since the effect of the convolution is to filter out the oscillations of the error $u - u_h^k$, we expect the error $u - u_h^*$ not to oscillate much around zero within each element on Ω_0. In contrast, in Ω_1, we see that, although the error is oscillatory, it does not really oscillate around zero. This is consistent with the fact that the boundary data on the points of Ω_1 lying on the boundary of Ω are *exact* whereas the boundary data at the points lying in the interior of Ω are *not*.

Table 1 History of convergence of $J_h^{\mathrm{G}}(u_h^*, v_h^*)$ (h-version) for $g(x) = \pi^2 \sin(3\pi x)$

N	$\|u - u_h\|_{L^2(\Omega)}$	Order	$\|J(u) - J(u_h)\|$	Order	$\|J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)\|$	Order
$k = 1$						
16	1.10e−01	–	1.13e+00	–	1.96e−01	–
32	2.78e−02	1.99	1.45e−01	2.97	2.96e−03	6.05
64	6.96e−03	2.00	1.83e−02	2.99	5.50e−05	5.75
128	1.74e−03	2.00	2.29e−03	3.00	2.11e−06	4.71
256	4.36e−04	2.00	2.86e−04	3.00	1.17e−07	4.17
512	1.09e−04	2.00	3.58e−05	3.00	7.15e−09	4.04
1024	2.73e−05	2.00	4.48e−06	3.00	4.44e−10	4.01
2048	6.81e−06	2.00	5.60e−07	3.00	2.77e−11	4.00
$k = 2$						
16	5.04e−03	–	3.97e−03	–	1.03e−07	–
32	6.35e−04	2.99	1.26e−04	4.98	1.79e−10	9.17
64	7.95e−05	3.00	3.96e−06	4.99	5.21e−13	8.43
128	9.95e−06	3.00	1.24e−07	5.00	1.62e−15	8.33
256	1.24e−06	3.00	3.88e−09	5.00	5.84e−18	8.12
512	1.56e−07	3.00	1.21e−10	5.00	2.26e−20	8.02
1024	1.94e−08	3.00	3.80e−12	5.00	8.87e−23	7.99
2048	2.43e−09	3.00	1.19e−13	5.00	3.49e−25	7.99
$k = 3$						
16	1.80e−04	–	7.07e−06	–	8.12e−10	–
32	1.13e−05	3.99	5.60e−08	6.98	1.17e−13	12.76
64	7.07e−07	4.00	4.39e−10	6.99	2.17e−18	15.72
128	4.42e−08	4.00	3.44e−12	7.00	3.11e−23	16.09
256	2.77e−09	4.00	2.69e−14	7.00	4.41e−28	16.11
512	1.73e−10	4.00	2.10e−16	7.00	1.54e−32	14.80
1024	1.08e−11	4.00	1.64e−18	7.00	2.83e−36	12.41
2048	6.75e−13	4.00	1.28e−20	7.00	6.89e−40	12.00

Next, we show the history of convergence of the h-version of the method. The numerical results related to the approximation using the piecewise gradients of u_h^* and v_h^* in Ω_1 are listed in Table 1; those related to the approximation using the approximate fluxes in Ω_1 are listed in Table 3. As expected, we observe that u_h converges at the rate of $\mathcal{O}(h^{k+1})$, that the linear functional approximation $J(u_h)$ converges to $J(u)$ at the rate of $\mathcal{O}(h^{2k+1})$, and that $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ converges to $J(u)$ at the rate of $\mathcal{O}(h^{4k})$ in both approaches. In Table 3, we also see that the approximation using the approximate fluxes seems to be superior to that using the piecewise gradients, especially in coarser meshes and $k = 1$. The difference, however, becomes smaller and smaller as we refine the mesh. Moreover, for $k = 2$ and $k = 3$ there seems to be no difference between the two approximations at all.

Table 2 History of convergence of $J_h^{\mathrm{G}}(u_h^*, v_h^*)$ (p-version) for $g(x) = \pi^2 \sin(3\pi x)$

k	$\|u - u_h\|_{L^2(\Omega)}$	Rate	$\|J(u) - J(u_h)\|$	Rate	$\|J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)\|$	Rate
$N = 1024$						
1	2.73e−05	–	4.48e−06	–	4.44e−10	–
2	1.94e−08	6.9	3.79e−12	13.9	8.87e−23	29.2
3	1.08e−11	7.8	1.64e−18	14.7	2.83e−36	31.1
$N = 2048$						
1	6.81e−06	–	5.60e−07	–	2.77e−11	–
2	2.43e−09	7.9	1.19e−13	15.4	3.49e−25	32.0
3	6.75e−13	8.2	1.28e−20	16.0	6.89e−40	33.9

Table 3 History of convergence of $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ (h-version) and $g(x) = \pi^2 \sin(3\pi x)$, and comparison with the approximation $J_h^{\mathrm{G}}(u_h^*, v_h^*)$

Here R_h^* denotes the ratio of the approximation error $|J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)|$ to the approximation error $|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)|$. For $k = 2$ and $k = 3$ this ratio is 1.00 in all the meshes considered below

N	$\|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)\|$	Order	R_h^*
$k = 1$			
16	2.93e−02	–	6.69
32	7.81e−04	5.23	3.79
64	3.13e−05	4.64	1.76
128	1.76e−06	4.15	1.20
256	1.10e−07	4.00	1.06
512	6.95e−09	3.98	1.03
1024	4.38e−10	3.99	1.01
2048	2.75e−11	3.99	1.01

Finally, let us show the history of convergence of the p-version of the methods. In Table 2, we compare the rates of exponential convergence with respect to the polynomial degree k. (We say that $e(k)$ converges exponentially to zero with rate α if we can write that $e(k) = c\,\exp(-\alpha\,k)$.) We see that the rate of exponential convergence of $|J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)|$ (and that of $|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)|$ by the results in Table 3) is *twice* that of $|J(u) - J(u_h)|$ which, in turn, is *twice* that of $\|u - u_h\|_{L^2(\Omega)}$.

4.2 The Two-Dimensional Case

For our model problem (1.1) in two-space dimensions, we now select the exact solution to be $u(x, y) := \sin(3\pi x)\sin(3\pi y)$ on $\Omega = (0, 1) \times (0, 1)$; the boundary conditions and the source term f is determined accordingly. We also take $g(x, y) := 18\pi^2 \sin(3\pi x)\sin(3\pi y)$ so that $v(x, y) := \sin(3\pi x)\sin(3\pi y)$.

Unlike what was done in the one-dimension case, we use the MATLAB default double precision arithmetic in this implementation. We use the HDG method with polynomials of degree k on a uniform mesh of $2N$ triangular elements to define u_h and v_h. The mesh is obtained by first dividing Ω into N^2 identical squares and then dividing each of the squares

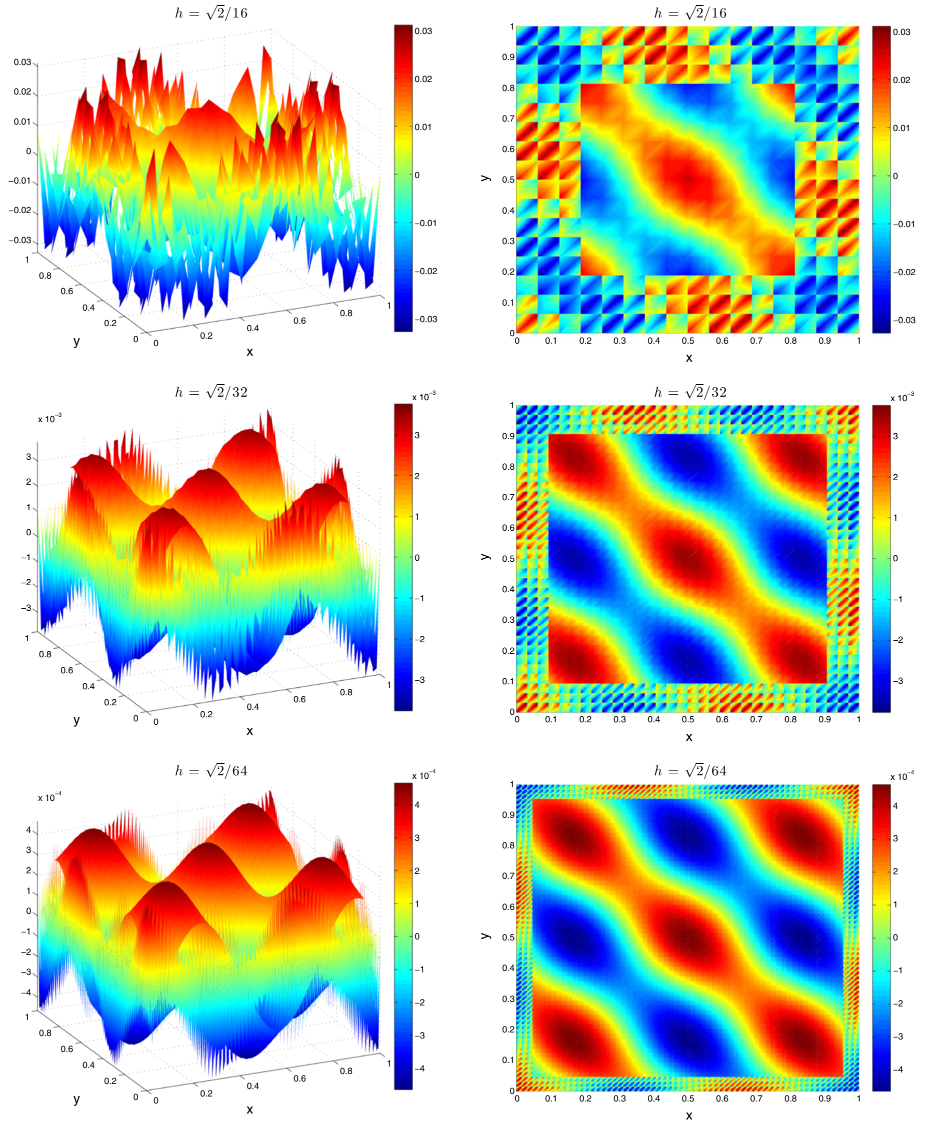

Fig. 3 Approximation errors of u_h^* when $k = 1$ and $N = 16$ (*top*), $N = 32$ (*middle*) and $N = 64$ (*bottom*). Note that the maximum error size in the set $\Omega \setminus \Omega_0$, which consists of a strip of elements of the mesh, is essentially the same as the maximum of the error size in the rest of the domain. Note also that the maximum errors are about 3.2×10^{-3}, 3.8×10^{-4}, 4.6×10^{-5}, for $N = 16, 32, 64$ which means that the error converges with order 3, as it diminishes by a factor 8 each time N is doubled

by joining the upper-right and lower-left vertices. We define Ω_0 by removing from Ω a boundary layer with at thickness of $2k + 1$ squares.

Let us start by showing that the function u_h^* seems to converge just as in the one-dimensional case. Indeed, in Fig. 3, for $k = 1$ and $N = 16, 32$, and 64, we see that the approximation errors $u - u_h^*$ are *smaller* in the region Ω_1 than in the region Ω_0. We also see

Table 4 History of convergence of $J_h^{\mathrm{G}}(u_h^*, v_h^*)$ (h-version) for $g(x, y) = 18\pi^2 \sin(3\pi x) \sin(3\pi y)$

h	$\|u - u_h\|_{L^2(\Omega)}$	Order	$\lvert J(u) - J(u_h)\rvert$	Order	$\lvert J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)\rvert$	Order
$k = 1$						
$\sqrt{2}/8$	2.76e−01	–	7.97e+00	–	8.73e+00	–
$\sqrt{2}/16$	7.73e−02	1.84	1.23e+00	2.69	4.22e−01	4.37
$\sqrt{2}/32$	1.99e−02	1.96	1.64e−01	2.91	1.67e−02	4.66
$\sqrt{2}/64$	5.02e−03	1.99	2.10e−02	2.97	6.49e−04	4.68
$k = 2$						
$\sqrt{2}/8$	4.44e−02	–	3.32e−01	–	8.51e−03	–
$\sqrt{2}/16$	5.97e−03	2.90	1.33e−02	4.64	3.15e−05	8.08
$\sqrt{2}/32$	7.61e−04	2.97	5.12e−04	4.70	8.18e−08	8.59
$\sqrt{2}/64$	9.57e−05	2.99	2.11e−05	4.60	1.49e−10	9.10

Table 5 History of convergence of $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ (h-version) for $g(x, y) = 18\pi^2 \sin(3\pi x) \sin(3\pi y)$, and comparison with the approximation $J_h^{\mathrm{G}}(u_h^*, v_h^*)$

h	$\lvert J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)\rvert$	Order	R_h^*
$k = 1$			
$\sqrt{2}/8$	1.30e−01	–	67.15
$\sqrt{2}/16$	4.09e−02	1.67	10.31
$\sqrt{2}/32$	2.75e−03	3.90	6.07
$\sqrt{2}/64$	1.78e−04	3.95	3.64
$k = 2$			
$\sqrt{2}/8$	1.10e−04	–	77.36
$\sqrt{2}/16$	3.34e−06	5.04	9.43
$\sqrt{2}/32$	2.87e−08	6.86	2.85
$\sqrt{2}/64$	6.19e−11	8.86	2.41

Here R_h^* denotes the ratio of the approximation error $|J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)|$ to the approximation error $|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)|$

that the errors have the same shape as N increases and that their magnitude decays at the order of $\mathcal{O}(h^{3=2k+1})$, as expected. Note that Ω_0 is obtained by removing from Ω a boundary layer thick of $3 = 2k + 1$ squares.

Let us now show the history of convergence of the h-version of the methods. In Tables 4 and 5, we see that the HDG solution u_h obtained by using k-th order polynomials converges at the rate of $\mathcal{O}(h^{k+1})$, the approximation $J(u_h)$ has the accuracy of $\mathcal{O}(h^{2k+1})$ and the adjoint-correction approximations $J_h^{\mathrm{G}}(u_h^*, v_h^*)$ and $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ converges not slower than $\mathcal{O}(h^{4k})$, as anticipated. We also see that the approximation by $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ seems to be better.

Finally, let us show the history of convergence of the p-version of the method. In Tables 6 and 7, we compare the rates of exponential convergence with respect to the polynomial degree k. We see a behavior similar to that of the one-dimensional case. Indeed, the rate of

Table 6 History of convergence of $J_h^{\mathrm{G}}(u_h^*, v_h^*)$ (p-version) for $g(x, y) = 18\pi^2 \sin(3\pi x) \sin(3\pi y)$

k	$\|u - u_h\|_{L^2(\Omega)}$	Rate	$\lvert J(u) - J(u_h)\rvert$	Rate	$\lvert J(u) - J_h^{\mathrm{G}}(u_h^*, v_h^*)\rvert$	Rate
$h = \sqrt{2}/32$						
1	1.99e−02	–	1.64e−01	–	1.67e−02	–
2	7.61e−04	3.3	5.12e−04	5.8	8.18e−08	12.2
$h = \sqrt{2}/64$						
1	5.02e−03	–	2.10e−02	–	6.49e−04	–
2	9.57e−05	4.0	2.11e−05	6.9	1.49e−10	15.3

Table 7 History of convergence of $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ (p-version) for $g(x, y) = 18\pi^2 \sin(3\pi x) \sin(3\pi y)$

k	$\|u - u_h\|_{L^2(\Omega)}$	Rate	$\lvert J(u) - J(u_h)\rvert$	Rate	$\lvert J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)\rvert$	Rate
$h = \sqrt{2}/32$						
1	1.99e−02	–	1.64e−01	–	2.75e−03	–
2	7.61e−04	3.3	5.12e−04	5.8	2.87e−08	11.5
$h = \sqrt{2}/64$						
1	5.02e−03	–	2.10e−02	–	1.78e−04	–
2	9.57e−05	4.0	2.11e−05	6.9	6.19e−11	14.9

exponential convergence of $|J(u) - J(u_h)|$ is about $\frac{7}{4}$ times that of $\|u - u_h\|_{L^2(\Omega)}$ whereas that of $|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)|$ is about*four* times that of $\|u - u_h\|_{L^2(\Omega)}$.

4.3 Computational Complexity of the Method

From the results of the previous two subsections, it is reasonable to conclude that, in the case in which both u and v are very smooth functions, our proposed method allows us to obtain an approximation converging with a rate of order $\mathcal{O}(h^{4k})$ for the h-version of the method. Not only that, the actually errors are actually significantly smaller, as we see in Table 8 for the 2D example. Therein, we see that the error $|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)|$ is always smaller than the error $|J(u) - J(u_h)|$, except for the first mesh of the case $k = 1$. When $h = \sqrt{2}/32$, for example, the error is about 10 times smaller for $k = 1$ and 6000 times for $k = 2$. And, when $h = \sqrt{2}/64$, the error is about 30 times smaller for $k = 1$ and 120,000 times for $k = 2$! Let us now argue that the computational effort needed to achieve such a remarkable result is, essentially, only *twice* the effort needed to compute u_h.

First, it is clear that to compute $J_h^{\mathrm{F}}(u_h^*, v_h^*)$, we need to compute v_h. However, for the particular problem we are dealing with, the problem solved by v_h differs from that solved by u_h only in the data, and this implies that computing both u_h and v_h involves numerically inverting the same exact matrix. The additional computational effort to get v_h is thus negligible. Of course, in more general situations, this is certainly not the case. Assuming that numerical

Table 8 The ratio $R_h := |J(u) - J(u_h)|/|J(u) - J_h^{\mathrm{F}}(u_h^*, v_h^*)|$ for the 2D example

h	R_h	h	R_h
$k = 1$		$k = 2$	
$\sqrt{2}/8$	9.13e−01	$\sqrt{2}/8$	3.90e+01
$\sqrt{2}/16$	2.91e+00	$\sqrt{2}/16$	4.22e+02
$\sqrt{2}/32$	9.82e+00	$\sqrt{2}/32$	6.26e+03
$\sqrt{2}/64$	3.24e+01	$\sqrt{2}/64$	1.23e+05

solving for v_h takes as much effort as computing u_h, we could then conclude that the new method requires, at least and roughly speaking, the doubling of the computational effort.

The computation of u_h^* in Ω_0 entails the multiplication of the degrees of freedom of u_h by a matrix whose size is only depends on the dimension of the local spaces used by the discretization. If we can manage to carry out those multiplications in parallel, the computational cost is then negligible with that of solving for u_h. Moreover, although to solve for u_h^* in Ω_1 implies the use of higher-degree polynomials, the number of degrees of freedom is reduced by a factor $1/h$, by the very construction of Ω_0. As a consequence the computational effort to get u_h^* in Ω_1 is also negligible in comparison with the computational effort needed to get u_h and v_h.

For these reasons, we say that to compute $J_h^{\mathrm{F}}(u_h^*, v_h^*)$ takes essentially twice the computational effort needed to compute $J(u_h)$.

5 Concluding Remarks

Although we considered a simple, second-order elliptic problem and a very simple functional, a wide variety of boundary-value problems and functionals can be treated with the very same technique as already indicated by Pierce and Giles [14]; see also the overview by Giles and Süli [11].

The technique proposed here has been tested for a particular HDG method, but it certainly works for *any* other numerical method for which the filtering proposed by Bramble and Schatz [1] also works. The distinctive feature of those methods is that their approximate solution must oscillate in a fixed pattern when defined in translation-invariant meshes. This is why we mentioned in the Introduction that the technique can be used with Galerkin methods like the mixed methods, the *adjoint-consistent* discontinuous Galerkin methods (including the LDG and IP methods), and the continuous Galerkin methods.

Finally, let us point out that we only considered problems with very smooth exact solutions and very smooth functionals in order to stress the power of the technique. How to handle their lack of smoothness and more involved boundary-valued problems constitutes the subject of ongoing work.

References

1. Bramble, J.H., Schatz, A.H.: Higher order local accuracy by averaging in the finite element method. Math. Comput. **31**(137), 94–111 (1977)
2. Celiker, F., Cockburn, B.: Superconvergence of the numerical traces of discontinuous Galerkin and hybridized mixed methods for convection-diffusion problems in one space dimension. Math. Comput. **76**, 67–96 (2007)

3. Chung, E.T., Cockburn, B., Fu, G.: The staggered DG method is the limit of a hybridizable DG method. SIAM J. Numer. Anal. **52**, 915–932 (2014)
4. Chung, E.T., Engquist, B.: Optimal discontinuous Galerkin methods for the acoustic wave equation in higher dimensions. SIAM J. Numer. Anal. **47**(5), 3820–3848 (2009)
5. Cockburn, B.: Static condensation, hybridization, and the devising of the HDG methods. In: Barrenechea, G.R., Brezzi, F., Cagniani, A., Georgoulis, E.H. (eds.) Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations, vol 114 of Lect. Notes Comput. Sci. Engrg., pp. 129–177. Springer, Berlin, 2016. LMS Durham Symposia funded by the London Mathematical Society. Durham, U.K., on July 8–16 (2014)
6. Cockburn, B., Gopalakrishnan, J., Lazarov, R.: Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. SIAM J. Numer. Anal. **47**(2), 1319–1365 (2009)
7. Cockburn, B., Ichikawa, R.: Adjoint recovery of superconvergent linear functionals from Galerkin approximations. The one-dimensional case. J. Sci. Comput. **32**(2), 201–232 (2007)
8. Cockburn, B., Luskin, M., Shu, C.-W., Süli, E.: Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. Math. Comput. **72**(242), 577–606 (2003)
9. Delfour, M., Hager, W., Trochu, F.: Discontinuous Galerkin methods for ordinary differential equations. Math. Comput. **36**, 455–473 (1981)
10. Giles, M.B., Pierce, N.A., Süli, E.: Progress in adjoint error correction for integral functionals. Comput. Visual. Sci. **6**, 113–121 (2004)
11. Giles, M.B., Süli, E.: Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. Acta Numer. **11**, 145–236 (2002)
12. Ichikawa, R.: Adjoint recovery of superconvergent linear functionals from Galerkin approximations. Ph.D. thesis, School of Mathematics, University of Minnesota, Minneapolis (2012)
13. 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)
14. Pierce, N.A., Giles, M.B.: Adjoint recovery of superconvergent functionals from PDE approximations. SIAM Rev **42**(2), 247–264 (2000)
15. Ryan, J.K., Shu, C.-W.: On a one-sided post-processing technique for the discontinuous Galerkin methods. Methods Appl. Anal. **10**(2), 295–307 (2003)
16. Ryan, J.K., Shu, C.-W., Atkins, H.: Extension of a post processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. SIAM J. Sci. Comput. **26**(3), 821–843 (2005)
17. Ryan, J.K., Cockburn, B.: Local derivative post-processing for the discontinuous Galerkin method. J. Comput. Phys. **228**(23), 8642–8664 (2009)
18. Thomée, V.: High order local approximations to derivatives in the finite element method. Math. Comput. **31**, 652–660 (1977)