Journal of Computational Physics 298 (2015) 360–386

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Optimal test functions for boundary accuracy in discontinuous finite element methods

Steven M. Kast *, Johann P.S. Dahm, Krzysztof J. Fidkowski

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, United States

ARTICLE INFO

Article history:
Received 2 October 2014
Received in revised form 31 March 2015
Accepted 27 May 2015
Available online 16 June 2015

Keywords:
Optimal test functions
Discontinuous Galerkin
Boundary discontinuous Petrov–Galerkin
Adjoint equations
Computational fluid dynamics

ABSTRACT

Obtaining accuracy along domain boundaries is often the primary goal of a numerical simulation. In this work, we show how the test space of discontinuous Galerkin (DG) methods can be optimized to achieve enhanced boundary accuracy for linear problems. Given some norm in which accuracy is desired, the optimal test functions render the numerical solution the best approximation to the true solution in that norm. For most norms, these test functions would be global in nature. However, for norms emphasizing boundary accuracy, they can be computed in an element-local manner and represent adjoint solutions for the interface fluxes. The resulting accuracy in these interface fluxes propagates globally, leading to convergence rates on the domain boundaries greater than $2p+1$. Indeed, if the test functions and interface fluxes are well-resolved, exact boundary fluxes are obtained. Here, we demonstrate these ideas for several Computational Fluid Dynamics (CFD) simulations, including advection–diffusion and linearized Euler problems in both one and two dimensions. An extension of the theory to nonlinear problems will be presented in a future work.

1. Introduction

Over the years, finite element methods have become the primary computational tool in many branches of engineering. Within the field of structural mechanics, the principal components of these methods – the test and trial functions – were originally chosen to be identical, resulting in a so-called Galerkin formulation. Based on this choice, the continuous Galerkin (CG) method has since seen wide application, owing both to its simplicity and provable optimality for many structural problems of interest.

When CG was applied to fluid dynamics problems, however, the results were poor. For convection–diffusion equations, spurious oscillations arose in the presence of gradients and boundary layers, corrupting the numerical solution. It soon became apparent that this lack of stability could be blamed on a suboptimal test space, and many so-called Petrov–Galerkin (or "stabilized") schemes have since arisen to address this issue [1–4]. These schemes, the most popular of which is the Streamline Upwind Petrov–Galerkin (SUPG) method [1], improve the stability of CG by modifying its test space in an upwind-biased manner.

With the emergence of discontinuous finite element methods, however, the impetus for optimizing the test space largely disappeared, since the use of a Riemann flux provides an inherent measure of stability [5]. Thus, methods of a discontinuous Galerkin (DG) type, which take the simplest possible test space, have come to dominate the literature [6–13]. But the

* Corresponding author.
E-mail address: kastsm@umich.edu (S.M. Kast).

http://dx.doi.org/10.1016/j.jcp.2015.05.048
0021-9991/

question naturally arises: is this the best option? Or, for discontinuous methods, is there a better – optimal – choice of test space?

This is the question we address here. Before doing so, however, we must say what we mean by "optimal." For discontinuous methods, this requires a slight shift in perspective. Rather than viewing the test space as a means to "fix" stability issues, we can instead view it as a means to obtain a specific, goal-oriented approximation of the true solution – one that provides accuracy in the regions we care about. This is, after all, the role of the test space in *any* finite element method: to define its goal. The appearance of oscillations or "instabilities" is just evidence that, in some sense, that goal has not been well defined.

Typically, the goal of a simulation is to achieve accuracy in a certain norm of interest. The "optimal" test functions can then be defined as those that render the numerical solution the best approximation to the true solution in the desired norm. This idea has been pursued by several authors in a continuous context [14–19], dating back to the work of Barrett and Morton in 1984 [14]. In addition, the test functions of stabilized schemes such as SUPG (which achieves H^1 optimality for certain problems [20]) can be viewed as "optimal" in a similar sense. More recently, Demkowicz and Gopalakrishnan have introduced discontinuous Petrov–Galerkin (DPG) methods, which employ optimal test functions within a more general, discontinuous framework [21–24]. These methods, developed initially within an "ultra-weak" [22] context and adapted to hybrid methods in [25], have L^2 optimality as their primary goal.

In the present work, we pursue a different goal. We note that while domain-interior accuracy is important, from an engineering standpoint the regions of greatest interest are often the domain *boundaries*. Indeed, obtaining the forces, fluxes, and distributions of quantities along the boundaries is often the principal goal of a simulation. This is the aim of the present work. Furthermore, for purposes of both familiarity and computational efficiency, we pursue this aim within the context of standard DG and hybrid DG (HDG) methods.

To that end, we present a simple framework for deriving and computing optimal test functions. These test functions render the solution optimal in a desired error norm, which in this case we choose to emphasize boundary accuracy. While in general the optimal test functions would satisfy global differential equations, when boundary accuracy is desired they can be computed in a purely element-local manner. When used within a standard DG or HDG framework, they result in a scheme we call the "Boundary Discontinuous Petrov–Galerkin" (BDPG) method.

Here, we list some relevant properties of the optimal test functions and the corresponding BDPG method:

1. The optimal test functions are elementwise adjoint solutions (i.e. generalized Greens functions [26]), similar to the fine-scale Greens functions used in several multiscale methods [3,27,20]
2. As adjoints, they ensure that information is properly "upwinded" within each element
3. They lead to accuracy in the element-interface fluxes, which propagates globally to the domain boundaries
4. They have close ties to *a posteriori* error estimation [28,29], which explains their ability to eliminate the flux errors
5. For 1D linear problems, if the test functions are well-represented, exact boundary fluxes are obtained
6. In higher dimensions, if the test functions *and* interface fluxes are well-represented, exact boundary fluxes are obtained

While the theory applies to general linear PDEs, here we focus on fluid applications, showing results for steady advection–diffusion and linearized Euler in both one and two dimensions. We begin with a simple approximation problem and a one-dimensional example that emphasizes the main ideas of the method. We then move on to multiple dimensions and systems of equations. The benefits and limitations of optimal test functions are discussed, and remaining challenges are identified. Application of the theory to nonlinear problems will be presented in a future work.

2. An approximation problem

In this section, we introduce some ideas that will be relevant to the derivation of optimal test functions later on.

Imagine we have some one-dimensional function, $u(x)$, the shape of which is known to us. Now assume that we would like to approximate this $u(x)$ with a polynomial.[1] If we call our polynomial approximation $u_h(x)$, then we can expand this $u_h(x)$ as

$$u_h(x) = \sum_{i=1}^{N} U_i \phi_i(x) , \tag{1}$$

where the ϕ_i are basis functions for our chosen polynomial space, and the U_i are discrete coefficients. If our polynomial space is of order p, then the dimension N is $N = p + 1$.

[1] In other words, we would like to perform a type of continuous "curve fit."

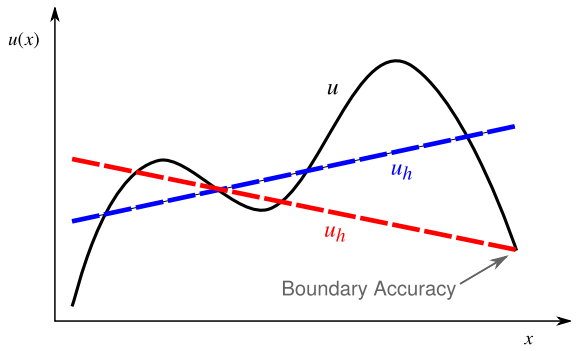


Fig. 1. Two $p = 1$ approximations of a one-dimensional function $u(x)$. The blue curve provides interior accuracy, while the red curve provides right-boundary accuracy. The amount of boundary accuracy provided depends solely on the type of error norm minimized. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Now, we would like to find the coefficients U_i that make u_h the "best" approximation to the original curve u. However, before we can do this, we must specify which norm we would like the best approximation *in*. For example, if we desire a least-squares approximation of u, then we need u_h to minimize the following norm:

$$\|e\|^2 = \int_\Omega (u_h - u)^2 \, dx \,. \tag{2}$$

Here, $e \equiv u_h - u$ is the approximation error, while $\Omega = [x_L, x_R]$ is the extent of the domain over which u is defined.

While the above norm would ensure that u_h provides an accurate approximation on the domain interior, we may also wish to obtain accuracy on the domain *boundaries*. If we are interested in accuracy near the right boundary, for example, then we could modify our error norm to read:

$$\|e\|^2 = \int_\Omega (u_h - u)^2 \, dx + w_R \, (u_h - u)^2 \Big|_{x_R} \,. \tag{3}$$

Here, w_R is a weight that determines how much emphasis is placed on the boundary; if it is taken large, more accuracy is obtained there.

Assume that Eq. (3) is our error norm of interest. Then in order to find the coefficients U_i that minimize this norm, we take the partial derivative of $\|e\|^2$ with respect to each U_i, and set this equal to zero. By basic calculus, the derivative of a function is zero at a critical point, and in the case of $\|e\|^2$ this critical point is in fact a minimum.

Since $u_h = \sum_{i=1}^{N} U_i \, \phi_i$, differentiating Eq. (3) with respect to U_i gives:

$$\frac{\partial \|e\|^2}{\partial U_i} = 0 = \int_\Omega 2 \, (u_h - u) \, \phi_i \, dx + 2 \, w_R \, (u_h - u) \, \phi_i \Big|_{x_R} \,. \tag{4}$$

Thus, if u_h is to provide the minimum error in $\|e\|^2$, it must satisfy the following N equations:

$$\int_\Omega \phi_i \, (u_h - u) \, dx + w_R \, \phi_i \, (u_h - u) \Big|_{x_R} = 0 \qquad i = 1..N \tag{5}$$

Fig. 1 shows two potential u_h curves that satisfy the above equations – one that emphasizes boundary accuracy (corresponding to large w_R) and another that emphasizes interior accuracy (corresponding to small w_R). From the figure, a few relevant conclusions can be drawn.

First, we see that the amount of boundary accuracy obtained depends solely on the choice of error norm minimized. Thus, if the goal is to achieve boundary accuracy, the type of polynomial used in the approximation (i.e. the "trial" space) is irrelevant. This idea holds in general: the lower the dimension of a given region of interest (in this case a zero-dimensional boundary), the more important the choice of error norm becomes, and the less important the choice of trial space becomes.

As we will show, in the context of a finite element method, the function u plays the role of the exact solution to a PDE, u_h is the numerical solution, and the norm is controlled by the test space. It is clear then that if boundary accuracy is desired, it is the test space rather than the trial space that should be optimized. This idea, along with Eq. (5), will be critical in deriving an optimal finite element scheme.

3. Optimal test functions (one-dimensional example)

Let us shift focus now and discuss our actual goal: solving PDEs. For the sake of clarity, consider a PDE of the following form:

$$\begin{aligned} \underbrace{a\frac{\partial u}{\partial x} + cu}_{Lu} &= f \quad x \in \Omega \\ u &= u_L \quad x = x_L \end{aligned} \tag{6}$$

This is a linear advection–reaction problem with a source term $f(x)$, where we assume $a > 0$ so the Dirichlet condition u_L is well-posed. The differential operator L is given by $L \equiv a\frac{\partial()}{\partial x} + c()$, and the residual is defined as $r(u) \equiv Lu - f$. The domain $\Omega = [x_L, x_R]$ is the same as in Section 2 and will be assumed here to consist of a single element.[2]

To obtain the weak form of the above problem, we multiply the residual by a test function and integrate, giving

$$\int_\Omega v\,(Lu - f)\,dx = 0 \qquad \forall v \in \mathcal{V}\,, \tag{7}$$

where v is any test function in some continuous space $\mathcal{V}$. For sufficiently smooth[3] u and v, integrating the vLu term by parts then gives

$$\int_\Omega \underbrace{\left[-a\frac{\partial v}{\partial x} + cv\right]}_{L^* v} u\,dx + vau\Big|_{x_L}^{x_R} - \int_\Omega vf\,dx = 0 \qquad \forall v \in \mathcal{V}\,. \tag{8}$$

Here, we have defined the operator that emerges after integration by parts as $L^* \equiv -a\frac{\partial()}{\partial x} + c()$. Finally, after inserting the boundary condition and moving the "known" terms to the right-hand side, we obtain

$$\underbrace{\int_\Omega L^* v\,u\,dx + vau\Big|_{x_R}}_{b(u,v)} = \underbrace{\int_\Omega vf\,dx + vau_L\Big|_{x_L}}_{l(v)} \qquad \forall v \in \mathcal{V}\,. \tag{9}$$

This equation, which relates the bilinear form $b(u, v)$ to the load $l(v)$, is satisfied by the exact solution u for any smooth test function v.

Now, to compute an approximate solution to the PDE in Eq. (6), a standard upwind DG method attempts to mimic Eq. (9). In other words, it seeks an approximate solution $u_h \in \mathcal{U}_h$ that satisfies

$$\underbrace{\int_\Omega L^* v_h\,u_h\,dx + v_h a u_h\Big|_{x_R}}_{b(u_h, v_h)} = \underbrace{\int_\Omega v_h f\,dx + v_h a u_L\Big|_{x_L}}_{l(v_h)} \qquad \forall v_h \in \mathcal{V}_h \tag{10}$$

where v_h is any test function in some discrete space $\mathcal{V}_h$. Once a basis $\{\phi_i\}$ is chosen for the approximation space $\mathcal{U}_h$ (typically assumed to be a polynomial space of order p), the discrete u_h can be represented as

$$u_h = \sum_{i=1}^{N} U_i\,\phi_i(x)\,, \tag{11}$$

where the U_i are the unknown solution coefficients. Then the remaining question is: what should our test space $\mathcal{V}_h$ be? A standard Galerkin method would choose $\mathcal{V}_h = \mathcal{U}_h$, so that the test space is identical to the trial space. But is this the best choice?

We have arrived at the critical point: we have used the word "best." *Best in what way?* Recall that in Section 2 we encountered the same issue. To get a "best" approximation, we had to first define the norm we wanted the best approximation *in*. In the same way, when solving a PDE, we must say *how* we would like the numerical solution u_h to approximate the exact u.

[2] The single-element setting allows us to assume a smooth numerical solution. While in the end we are interested in discontinuous finite element methods, the ideas in this section will ultimately be applied only *within* each element, where the smoothness assumption is justified.

[3] We assume that it is valid to evaluate u and v on the domain boundaries.

Typically, we would like u_h to obtain some amount of interior and (particularly in this work) boundary accuracy. For the current problem, the relevant boundary to request accuracy in is the right boundary, since the solution on the left is already known from the Dirichlet condition. Thus, the norm we desire the best approximation in may look something like:

$$\|e\|^2 = \int_\Omega (u_h - u)^2 \, dx + w_R \, (u_h - u)^2 \Big|_{x_R} .$$

This is the same norm used in Section 2. Recall from that section that if u_h is to minimize this norm, it must satisfy the zero-derivative condition given by Eq. (5).

Now, we claim that with a certain ("optimal") choice of test functions, we can ensure that Eq. (5) is satisfied by our finite element solution. First, note that since $\mathcal{V}_h \subset \mathcal{V}$, we can choose a common test function $v_h \in \mathcal{V}_h \subset \mathcal{V}$ for Eqs. (9) and (10). Doing so and subtracting the two equations then results in:

$$\underbrace{\int_\Omega L^* v_h \, (u_h - u) \, dx + v_h a \, (u_h - u) \Big|_{x_R}}_{b(e, v_h)} = 0 \qquad \forall v_h \in \mathcal{V}_h \, . \tag{12}$$

This equation is satisfied by the finite element error regardless of how the test space is chosen. However, to see how an optimal scheme can be created, note that the above expression involves some quantity that is equal to zero. Next, note that the error minimization statement (Eq. (5)) *also* involves some quantity that is equal to zero. Then the idea is this: if we can make the above bilinear form *look* like the error minimization statement, our finite element solution u_h will necessarily minimize that error.

To make this clearer, we first make a simple notational change. Regardless of how the test space is chosen, it must have the same dimension (N) as the trial space, in order for the number of equations to equal the number of unknowns. Therefore, we can replace the general test function v_h above with a specific test function v_i, where i ranges from 1 to N. Doing so gives:

$$\int_\Omega L^* v_i \, (u_h - u) \, dx + a \, v_i \, (u_h - u) \Big|_{x_R} = 0 \qquad i = 1..N \tag{13}$$

Now, our goal is to make this equation look like Eq. (5), which is:

$$\int_\Omega \phi_i \, (u_h - u) \, dx + \; w_R \, \phi_i \, (u_h - u) \Big|_{x_R} = 0 \qquad i = 1..N$$

By simply comparing these equations, we see that the way to make them identical is to set

$$\left.\begin{array}{lll} L^* v_i & = & \phi_i \qquad x \in \Omega \\ a \, v_i & = & w_R \, \phi_i \quad x = x_R \end{array}\right\} \qquad i = 1..N \tag{14}$$

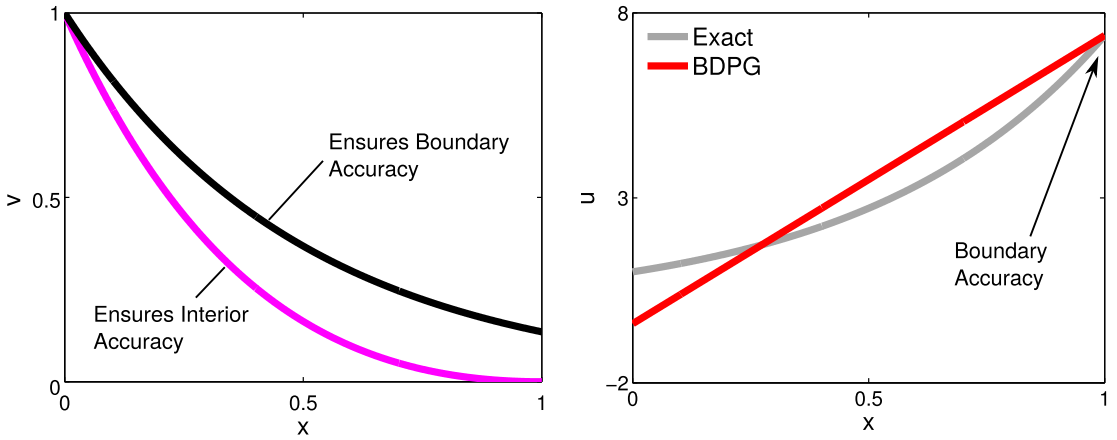
Thus, if the discrete solution u_h is to minimize the error given by Eq. (3), the test functions v_i must satisfy the above differential equation(s) and boundary condition(s). The test functions that satisfy these equations are the *optimal test functions*, in the sense that they make u_h the best approximation to u in the desired error norm.

At this point, it is worth making a few remarks.

Remark 1. While for simple problems the optimal test functions can be found analytically, in general we must compute them numerically. This can be done by (e.g.) approximating them in a high-order space. A similar strategy has been pursued in much of the DPG literature [22].

Remark 2. The idea of making $b(e, v_h)$ reduce to the derivative of a desired error norm has arisen previously in the context of continuous finite element methods [14,19,16,15]. More recently, the DPG methods of Demkowicz, Gopalakrishnan, et al. have employed similar ideas, primarily within the context of a discontinuous "ultra-weak" formulation [21–24]. In contrast, here we consider standard DG and HDG formulations,[4] with the specific goal of achieving boundary accuracy.

[4] Note that for certain problems (such as pure advection), the ultra-weak DPG formulation may be preferable to DG or HDG formulations, since its independent treatment of interior and flux unknowns enables an independent optimization of each [21]. However, for problems such as advection–diffusion, the HDG formulation is preferred since its number of globally coupled unknowns is just half that of the corresponding ultra-weak formulation.



(a) Optimal test functions (b) Single-element BDPG solution

Fig. 2. One-dimensional advection–reaction: (a) Normalized optimal test functions corresponding to a $p = 1$ trial space and large w_R. The black test function (v_2) provides right-boundary accuracy, while the remaining test function (v_1) provides interior L^2 accuracy. Note the upwind/leftward bias of both functions. (b) The solution obtained using the optimal test functions. Right-boundary accuracy is achieved.

Remark 3. Recall that to achieve boundary accuracy we simply choose w_R to be large. This emphasizes the boundary term in the error norm (Eq. (3)), which in the limit of large w_R ensures that u_h provides the best approximation to u on the boundary. When the corresponding optimal test functions are used within a DG or HDG framework, we refer to the resulting scheme as a "Boundary Discontinuous Petrov–Galerkin" (BDPG) method.

Remark 4. Note that since information propagates to the right in this problem, requesting accuracy in the right-boundary state $u_h|_{x_R}$ may also be viewed as requesting accuracy in the *outgoing flux*. This is the view we adopt when generalizing to multidimensional systems.

Remark 5. We have posed the above derivations in a single-element setting, which means the equations satisfied by the optimal test functions are in fact *global* differential equations. In practice, solving these would be prohibitive, so we will need to localize them in some way.

Remark 6. Finally, use of the optimal test functions will not, in general, result in a conservative scheme. This is because the solutions to the test function equations (Eq. (14)) will not always contain the constant mode. However, from a purely geometrical standpoint, when boundary accuracy is desired we do not necessarily *want* the scheme to be conservative. For example, for a situation like the one depicted in Fig. 1, requesting that u_h interpolate u on both boundaries would preclude it from satisfying a relation such as $\int_\Omega u_h\,dx = \int_\Omega u\,dx$.

We will elaborate on these remarks later on. For now, let us return to our one-dimensional advection–reaction problem. For this problem, we can solve Eq. (14) analytically to find the optimal test functions. If we assume a $p = 1$ trial space with Lagrange basis functions $\phi_1 = 1 - x$ and $\phi_2 = x$, we can compute two corresponding test functions. For a parameter choice of $a = 1$, $c = -2$, and $f(x) = 0$, these test functions are

$$v_1 = \frac{1}{4}e^{2(1-x)} + \frac{x}{2} - \frac{3}{4} \qquad \text{and} \qquad v_2 = \left(w_R - \frac{1}{4}\right)e^{2(1-x)} - \frac{x}{2} + \frac{1}{4}. \tag{15}$$

These functions are plotted in Fig. 2, where w_R has been taken large and the maximum values have been normalized to unity. The solution obtained by using these test functions in Eq. (10) is also shown. Note that the solution achieves accuracy on the right boundary, as desired.

From the figure, we see that in contrast to the symmetric nature of the standard Galerkin test functions, the optimal test functions display a clear upwind bias. Thus, from a physical perspective, we can think of the optimal test functions as performing a proper upwinding of information within the domain. This idea is related to another important property of the optimal test functions – namely, that they are *adjoint* solutions.

4. Optimal test functions as adjoints

To see that the optimal test functions satisfy adjoint equations, recall that for a given differential operator L, the definition of its adjoint operator L^* is:

$$(Lu, v) = (u, L^* v) \qquad \forall u \in \mathcal{U}, \ \forall v \in \mathcal{V}, \tag{16}$$

where $\mathcal{U}$ and $\mathcal{V}$ are function spaces over which the above inner product is defined. In practical cases, L^* is simply the operator that emerges after integrating by parts, and hence is exactly the operator in the test function equations (Eq. (14)).

From optimization [30] and *a posteriori* error estimation [28,29], it is known that adjoint equations provide the sensitivity of a certain output to perturbations in the residual of a PDE. Therefore, if the optimal test functions themselves satisfy an adjoint equation, this begs the question: *for what output*? By examining the right-hand side of Eq. (14) (which contains the output linearization), we find that the associated output is given by:

$$J_i = \int_\Omega \phi_i u \, dx + w_R \phi_i u \Big|_{x_R}. \tag{17}$$

This output represents the *projection* of the exact solution u against the i-th trial basis function.

Now, from *a posteriori* error estimation, it is known that the adjoint-weighted residual represents the error in a given output. Thus, since the optimal test functions v_i are adjoints for the outputs J_i, when we use them in the finite element weighted residual,

$$\int_\Omega v_i \underbrace{(Lu_h - f)}_{r(u_h)} dx = 0 = \delta J_i \qquad i = 1..N, \tag{18}$$

we are directly enforcing that the error in each projection output J_i, i.e. $\delta J_i \equiv J_i(u_h) - J_i(u)$, is zero. This implies that the discrete solution u_h is the direct projection of the exact solution u into the trial space, with respect to the desired error norm. This is the same conclusion arrived at in the previous section, but viewed from a different perspective.

Finally, to further clarify the relationship between the outputs J_i and the minimization of the error $\|e\|^2$, consider the following. We have said that using the optimal v_i gives zero error in the J_i. But from the above definition of J_i, zero error in J_i implies

$$\delta J_i = J_i(u_h) - J_i(u) = \int_\Omega \phi_i (u_h - u) \, dx + w_R \phi_i (u_h - u) \Big|_{x_R} = 0. \tag{19}$$

This is identical to the statement that $\frac{\partial \|e\|^2}{\partial U_i} = 0$, i.e. it is identical to Eq. (5), and thus implies that $\|e\|^2$ is minimized.

Remark 7. Note that if the test functions are approximated numerically (e.g. at some order p_{test}), the error in the J_i will not be identically zero. However, from Eq. (18), the convergence rate of this error will at a minimum correspond to the sum of the test function and residual convergence rates (i.e. $p_{\text{test}} + p + 1$), and at a maximum to a value of $2p_{\text{test}} + 1$ (via a Galerkin orthogonality argument). Thus, taking $p_{\text{test}} > p$ will yield higher output convergence rates than the standard $2p + 1$ rates of DG.

Remark 8. Since adjoint solutions are a type of "generalized" Green's function [26], the optimal test functions are closely related to the fine-scale Green's functions used in many multiscale methods, such as the Variational Multiscale Method [27]. Approximating the optimal test functions with $p_{\text{test}} > p$ brings in the "fine-scale" information that ultimately leads to improved solution accuracy.

5. Localization of test functions

With the nature of the optimal test functions discussed, we now turn to the issue of localization. In the above sections, we derived the optimal test functions while assuming a single-element mesh. This means that the adjoint equations satisfied by the test functions are in fact *global* differential equations. Since in practice these would be prohibitive to solve, we need to find a way to localize the test functions without giving up their accuracy.

Fortunately, for norms emphasizing boundary accuracy, localization is straightforward. We simply loop over each element in the mesh and apply the theory from Section 3 *inside* each element. We thus solve purely local adjoint problems (with support over a single element) to compute the test functions, with the local outputs defined by

$$J_i = \int_K \phi_i u \, dx + w_R \phi_i u \Big|_{\partial K_R}. \tag{20}$$

Here, K represents the domain of a given element, while ∂K_R represents its right (downwind) boundary. Note that this output has the same form as in Eq. (17), but is defined over element K rather than the entire domain.

In this localized context, the optimal test functions then minimize the desired error norm within each element. Thus, taking w_R large in Eq. (20) will provide accuracy in the outgoing flux on each *element* boundary, rather than on the domain boundary. However, the critical idea is that, if flux accuracy is obtained on each element boundary, then this accuracy will propagate downstream, ultimately yielding accuracy on the domain boundary. A similar idea holds for more general problems, including those with diffusion terms. Since the fluxes represent the only means by which elements in the mesh communicate, if these local fluxes can be made accurate, global accuracy follows naturally.

This idea is supported analytically as well. Appendix A shows that for the current problem, as w_R is taken large, the test space formed by the local adjoints "contains" the global adjoints corresponding to the domain-boundary fluxes. This implies that (if well-represented) the local test functions are in fact globally optimal, in the sense that they deliver zero error in the boundary fluxes.

6. Implementation (one-dimensional example)

With the localization defined, we are ready to use the optimal test functions in a practical setting. First, we note that the above ideas can be applied to several discontinuous formulations. For the current problem, we will apply them to a standard DG formulation, while later on we will use a hybrid method. Regardless of the method used, there are two main steps to be performed: first, the computation of optimal test functions on each element; and second, the use of these test functions in the bilinear form.

6.1. Computation of test functions

If the mesh is uniform or the topology self-similar, the localized test functions can be computed just once on a reference element and can then be "copied" onto each element in turn. Otherwise, independent test functions are computed on each element. The simplest way to compute the test functions is to solve the local adjoint problems using a DG method. For the current advection–reaction problem, we thus solve the following equation for each v_i on a given element:

$$\text{Find } v_i \in \mathcal{U}_{\text{test}} \;\text{ s.t. }\; \underbrace{\int_K L^* v_i \, \delta u \, dx + a v_i \, \delta u \Big|_{\partial K_R}}_{b_K(\delta u, v_i)} = \underbrace{\int_K \phi_i \, \delta u \, dx + w_R \phi_i \delta u \Big|_{\partial K_R}}_{J_i(\delta u)} \qquad \forall \delta u \in \mathcal{U}_{\text{test}} \,. \tag{21}$$

Note that this is just the weak form of the original adjoint equations derived in Eq. (14). These equations are solved in an enriched space $\mathcal{U}_{\text{test}}$ with corresponding order p_{test}, which must be *higher* than the original order p of the space $\mathcal{U}_h$. The higher the p_{test}, the more accurate the test functions, and the more accurate the final solution on the boundaries. Indeed, in one dimension, if the local test functions are represented exactly, exact element and boundary fluxes are obtained.

For DG codes that construct a primal Jacobian matrix, the above adjoint equations do not need to be explicitly discretized. Instead, a discrete adjoint approach [29] can be taken in which the equations

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \mathbf{V}_i = \frac{\partial J_i}{\partial \mathbf{U}}^T \qquad i = 1..N \,, \tag{22}$$

are solved for the test function coefficients $\mathbf{V}_i$ on each element, where $\partial \mathbf{R}/\partial \mathbf{U}$ and $\partial J_i/\partial \mathbf{U}$ are the elementwise order-p_{test} Jacobian and output linearization, respectively. Finally, since taking w_R large leads to large test function magnitudes, the test functions can be orthonormalized after computation with respect to a discrete or continuous norm of choice. This ensures that the primal system (discussed below) remains well-conditioned.

6.2. Construction of primal system

Once computed, the optimal test functions can be used in place of the standard DG test functions, and the primal problem can be solved as usual. Upon doing so, our first inclination would be to use high-order quadrature rules to perform all integrations, due to the high-order nature of the test functions. However, it turns out that high-order quadrature is, for the most part, unnecessary.

For example, for our current problem, the primal equation on a given element K is (from Eq. (10)):

$$\underbrace{\int_K L^* v_i \, u_h \, dx + v_i a u_h \Big|_{\partial K_R}}_{b_K(u_h, v_i)} = \int_K v_i \, f \, dx + v_i a u_{K-1} \Big|_{\partial K_L} \qquad \forall v_i \in \mathcal{V}_{\text{test}} \tag{23}$$

where we assume the v_i have been computed from Eq. (21), and u_{K-1} is the neighbor-element state. But note that, since u_h is contained in the space $\mathcal{U}_{\text{test}}$, Eq. (21) implies that the left-hand side of Eq. (23) can be rewritten as:

$$\underbrace{\int_K \phi_i u_h \, dx + w_R \phi_i u_h \Big|_{\partial K_R}}_{J_i(u_h)} = \int_K v_i f \, dx + v_i a u_{K-1} \Big|_{\partial K_L} \qquad \forall v_i \in \mathcal{V}_{\text{test}} \tag{24}$$

Thus, for implementation purposes, we can use the above formulation of the primal problem rather than Eq. (23).[5,6] Note that all terms involving the optimal test functions have vanished from the left-hand side, and have been replaced with low-order integrals. Furthermore, in practical cases the source term f on the right-hand side is often zero, so that *all* high-order interior integrals disappear. If this is the case, the optimal test functions appear only in the right-most term of Eq. (24) – i.e. on the upwind boundary. An interesting conclusion is that, for most problems, the accuracy of BDPG rests solely on obtaining accurate test function values on the element boundaries.

6.3. Summary of method

The BDPG method can be summarized as follows:

1. Loop over each element.
2. Compute the local optimal test functions (adjoints) v_i by solving Eq. (22) at order p_{test}, where $p_{\text{test}} \geq p$.
3. Normalize the v_i with respect to (e.g.) a discrete or continuous L^2 norm.
4. Use the v_i in place of the standard DG test functions. To avoid high-order quadrature, the alternate form of the primal problem (Eq. (24)) can be used.
5. Obtain accelerated convergence of the boundary fluxes. In 1D, a minimum rate of $p_{\text{test}} + p + 1$ will be obtained, with rates of up to $2p_{\text{test}} + 1$ possible. By choosing $p_{\text{test}} > p$, these rates are necessarily higher than the maximum DG rate of $2p + 1$.

7. Results (one-dimensional example)

With the above procedures established, we are ready to solve a multi-element problem. As an example, we solve the advection–reaction equations with $a = 1$, $c = -8.5$, and $f = 0$. To see if BDPG obtains boundary accuracy, we choose the boundary weight in the error norm to be high – in this case, taking it to be $w_R = 10^{12}$ – and choose the test space order to be $p_{\text{test}} = 10$.

From Fig. 3, we see that BDPG achieves boundary errors approximately 10 orders of magnitude lower than standard DG. This is encouraging, and confirms that our purely local test space (shown in Fig. 4) is capable of achieving global optimality. Note that the initial convergence rate of the BDPG fluxes (which is observed to be $2p_{\text{test}} + 1$) is due solely to the inexact representation of the test functions, and that if analytical test functions were used, exact boundary fluxes would be obtained. Finally, while the primary goal of BDPG is to achieve boundary accuracy, from Fig. 4 we see that it also performs well in an L^2 sense, exhibiting none of the preasymptotic behavior seen with standard DG.

8. General theory for multi-dimensional systems

So far, we have derived the optimal test functions in the setting of a one-dimensional advection–reaction problem. However, the relevant concepts extend naturally to systems of equations, as well as to multiple dimensions. We will briefly describe those extensions here. To simplify the presentation, we will again assume that the domain Ω consists of a single element.

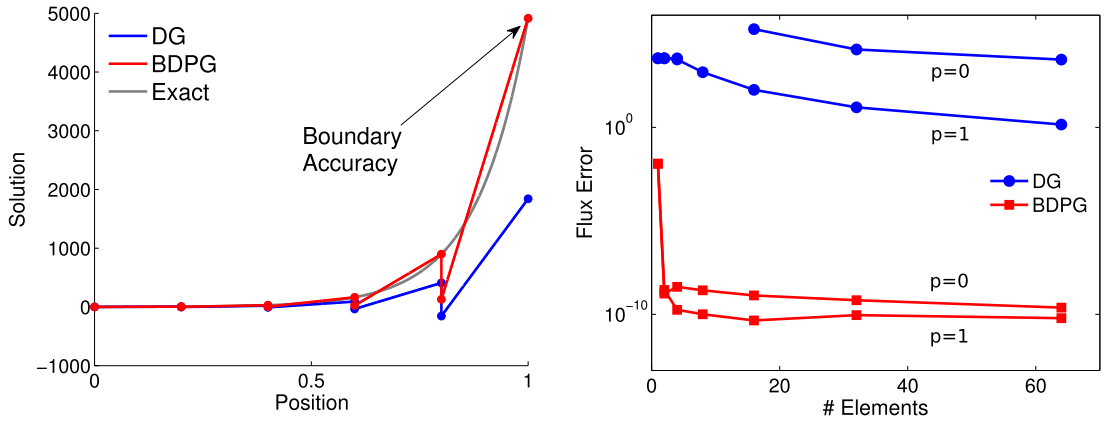
A general steady-state conservation law in multiple dimensions can be written as

$$\nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \vec{\mathbf{q}}) = \mathbf{0}, \tag{25}$$

$$\vec{\mathbf{q}} - \nabla \mathbf{u} = \vec{\mathbf{0}}, \tag{26}$$

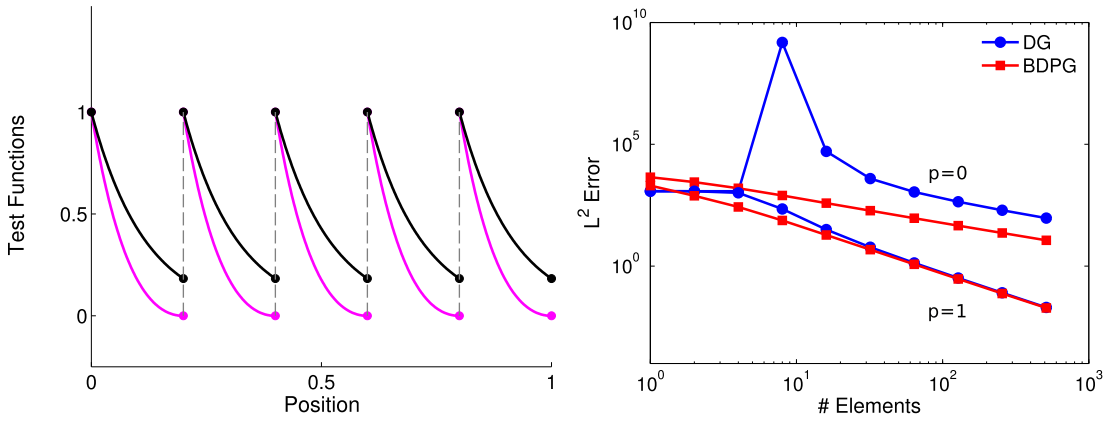
[5] A similar idea was proposed in the context of continuous finite elements by Givoli in 1988 [16].

[6] Note that the left-hand side of this equation (i.e. the element self-block of the Jacobian) is symmetric and positive definite. However, the coupling to u_{K-1} via the upwind flux means that (unlike other DPG methods [22]) the global Jacobian is nonsymmetric.



(a) Numerical solutions (b) Boundary flux error convergence

Fig. 3. One-dimensional advection–reaction: (a) $p = 1$ solutions for DG and BDPG on a 5-element mesh. (b) The error in the right-boundary flux for $p = 0$ and $p = 1$ runs. The BDPG fluxes converge at a rate of $2p_{\text{test}} + 1$ and quickly attain machine precision accuracy.



(a) Localized optimal test functions (b) L^2 error convergence

Fig. 4. One-dimensional advection–reaction: (a) Localized optimal test functions for the 5-element BDPG solution shown in Fig. 3(a). The two test functions on each element correspond to the two $p = 1$ trial bases. (b) L^2 error convergence for $p = 0$ and $p = 1$ DG and BDPG solutions. The L^2 performance of the methods is similar, with BDPG showing greater stability on coarse meshes.

where $\mathbf{u}$ is the state vector and $\vec{\mathbf{q}}$ represents the gradient of the state. $\vec{\mathbf{F}}$ is a flux vector, which may contain both advective and diffusive components, and consists of r state components in *dim* dimensions. (Note that boldface indicates a state vector, while an arrow indicates a spatial vector.) In general, a source term $\mathbf{S}(\mathbf{u})$ could be added to Eq. (25), though we omit it for brevity here. Furthermore, we assume that $\vec{\mathbf{q}}$ is an independent unknown, since this is the case for the hybridized method presented later. However, this is not critical to the theory.

To obtain the weak form of the above problem, we weight Eqs. (25) and (26) by test functions $\mathbf{v}$ and $\vec{\boldsymbol{\tau}}$, respectively, giving a total weighted residual (upon summation) of

$$R \equiv \int_{\Omega} \vec{\boldsymbol{\tau}}^{\,T} \cdot \left(\vec{\mathbf{q}} - \nabla \mathbf{u}\right) d\Omega + \int_{\Omega} \mathbf{v}^T \left(\nabla \cdot \vec{\mathbf{F}}\right) d\Omega \; = 0\,. \tag{27}$$

Note that this residual is just a scalar value. We next integrate both terms in Eq. (27) by parts, giving

$$R = \int_{\Omega} \vec{\boldsymbol{\tau}}^{\,T} \cdot \vec{\mathbf{q}}\, d\Omega + \int_{\Omega} \nabla \cdot \vec{\boldsymbol{\tau}}^{\,T}\, \mathbf{u}\, d\Omega - \int_{\Omega} (\nabla \mathbf{v})^T \cdot \vec{\mathbf{F}}\, d\Omega + \int_{\partial\Omega} \mathbf{v}^T (\vec{\mathbf{F}} \cdot \vec{n})\, ds - \int_{\partial\Omega} \left(\vec{\boldsymbol{\tau}} \cdot \vec{n}\right)^T \mathbf{u}\, ds = 0. \tag{28}$$

If we now assume Dirichlet boundary conditions (denoted by $\mathbf{u}_B$), the right-most term above becomes a "known" value and can be moved to the right-hand side. After making this change and for convenience defining $\hat{\mathbf{F}} = \vec{\mathbf{F}} \cdot \vec{n}$, we obtain:

$$\underbrace{\int_\Omega \vec{\boldsymbol{\tau}}^{\,T} \cdot \vec{\mathbf{q}}\, d\Omega + \int_\Omega \nabla \cdot \vec{\boldsymbol{\tau}}^{\,T}\, \mathbf{u}\, d\Omega - \int_\Omega (\nabla \mathbf{v})^T \cdot \vec{\mathbf{F}}\, d\Omega + \int_{\partial\Omega} \mathbf{v}^T \hat{\mathbf{F}}\, ds}_{b(\mathbf{u},\vec{\mathbf{q}},\mathbf{v},\vec{\boldsymbol{\tau}})} = \underbrace{\int_{\partial\Omega} \left(\vec{\boldsymbol{\tau}} \cdot \vec{n}\right)^T \mathbf{u}_B\, ds}_{l(\mathbf{v},\vec{\boldsymbol{\tau}})}. \tag{29}$$

From this equation, we are able to define the bilinear form $b\left(\mathbf{u}, \vec{\mathbf{q}}, \mathbf{v}, \vec{\boldsymbol{\tau}}\right)$.

Next, as in one dimension (i.e. Eq. (9)), we would like to write this bilinear form as a product of the state variables and the adjoint operator applied to the test functions. In order to do this, we must first write all domain integrals explicitly in terms of $\mathbf{u}$ and $\vec{\mathbf{q}}$. To start, we rewrite the flux $\vec{\mathbf{F}}$, assumed linear, as

$$\vec{\mathbf{F}} = \frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\, \mathbf{u} + \frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\, \mathbf{q}_j, \tag{30}$$

where summation over the spatial dimension j is implied.[7] We now substitute this expression (Eq. (30)) into Eq. (29) and transpose the first three terms, giving

$$b = \int_\Omega \vec{\mathbf{q}}^T \cdot \vec{\boldsymbol{\tau}}\, d\Omega + \int_\Omega \mathbf{u}^T \nabla \cdot \vec{\boldsymbol{\tau}}\, d\Omega - \int_\Omega \left(\mathbf{u}^T \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\right]^T + \mathbf{q}_j^T \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\right]^T \right) \cdot (\nabla \mathbf{v})\, d\Omega + \int_{\partial\Omega} \mathbf{v}^T \hat{\mathbf{F}}\, ds. \tag{31}$$

Grouping the $\mathbf{u}$ and $\vec{\mathbf{q}}$ terms then results in

$$b = \int_\Omega \mathbf{q}_j^T \underbrace{\left(\boldsymbol{\tau}_j - \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\right]^T \cdot \nabla \mathbf{v} \right)}_{L^*_{q,j}(\vec{\boldsymbol{\tau}},\mathbf{v})} d\Omega + \int_\Omega \mathbf{u}^T \underbrace{\left(\nabla \cdot \vec{\boldsymbol{\tau}} - \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\right]^T \cdot \nabla \mathbf{v} \right)}_{L^*_u(\vec{\boldsymbol{\tau}},\mathbf{v})} d\Omega + \int_{\partial\Omega} \mathbf{v}^T \hat{\mathbf{F}}\, ds. \tag{32}$$

Next, if we define group variables (denoted by a tilde) for the states and test functions as

$$\tilde{\mathbf{u}} \equiv \begin{bmatrix} \mathbf{q}_j \\ \mathbf{u} \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{v}} \equiv \begin{bmatrix} \boldsymbol{\tau}_j \\ \mathbf{v} \end{bmatrix}, \tag{33}$$

and we define the *adjoint* operator L^* (based on the operators $L^*_{q,j}$ and L^*_u in Eq. (32)) as

$$L^* \equiv \begin{bmatrix} L^*_{q,j} \\ L^*_u \end{bmatrix} = \begin{bmatrix} I(\,) & -\left[\dfrac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\right]^T \cdot \nabla(\,) \\ \partial_j(\,) & -\left[\dfrac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\right]^T \cdot \nabla(\,) \end{bmatrix}, \tag{34}$$

then Eq. (32) can be rewritten in the following form:

$$b(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \int_\Omega \tilde{\mathbf{u}}^T \left(L^* \tilde{\mathbf{v}} \right) d\Omega + \int_{\partial\Omega} \mathbf{v}^T \hat{\mathbf{F}}(\tilde{\mathbf{u}})\, ds = l\left(\tilde{\mathbf{v}}\right) \qquad \forall \tilde{\mathbf{v}} \in \tilde{\mathcal{V}}, \tag{35}$$

where the operator L^* acts as a matrix on the group variable $\tilde{\mathbf{v}}$. Note that the above $b(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ is essentially the same as the one-dimensional bilinear form (i.e. Eq. (9)), except that the states and test functions are now vectors, and we have a general flux $\hat{\mathbf{F}}$ on the boundary rather than the one-dimensional flux au. Thus, from this point on, the development will parallel the one-dimensional theory.

To approximate the above equation, a DG method chooses a set of discrete states and test functions, $\tilde{\mathbf{u}}_h$ and $\tilde{\mathbf{v}}_h$, as well as a numerical flux $\hat{\mathbf{F}}(\tilde{\mathbf{u}}_h)$, resulting in the discrete bilinear form:

$$b(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = \int_\Omega \tilde{\mathbf{u}}_h^T \left(L^* \tilde{\mathbf{v}}_h \right) d\Omega + \int_{\partial\Omega} \mathbf{v}_h^T \hat{\mathbf{F}}(\tilde{\mathbf{u}}_h)\, ds = l\left(\tilde{\mathbf{v}}_h\right) \qquad \forall \tilde{\mathbf{v}}_h \in \tilde{\mathcal{V}}_h. \tag{36}$$

[7] Note that for nonlinear problems, a similar expression would hold for the Fréchet linearization of the flux.

Once a basis is chosen for the trial space representations of $\mathbf{u}_h$ and $\vec{\mathbf{q}}_h$, these states can be expanded as

$$u_{s,h} = \sum_{m=1}^{n_U} U_{s,m}\,\phi_{s,m}(\vec{x}) \quad \text{and} \quad q_{s,d,h} = \sum_{m=1}^{n_Q} Q_{s,d,m}\,\phi_{s,d,m}(\vec{x})\,. \tag{37}$$

Here, s indexes the state component (ranging from 1 to the state rank, r), m indexes the basis function (ranging from 1 to the number of nodes, n_U or n_Q), and d indexes the dimension (ranging from 1 to dim). Finally, $U_{s,m}$ and $Q_{s,d,m}$ represent the unknown solution coefficients, the total number of which is given by $N \equiv N_U + N_Q = r\,n_U + r\,n_Q \cdot dim$.

The remaining task is to define the test space. In order to derive the optimal test space, we follow a similar strategy as before: we first define an error norm we wish to minimize, then choose the test functions such that the bilinear form reduces to the *derivative* of that norm. To help determine an appropriate error norm to minimize, we first choose a common test function $\tilde{\mathbf{v}}_h$ for Eqs. (35) and (36). This allows us to equate (and subtract) the exact and discrete bilinear forms, resulting in the equation

$$b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_h) = 0\,, \tag{38}$$

where $\tilde{\mathbf{e}}$ is a group variable representing the errors in the states. If we now set $\tilde{\mathbf{v}}_h = \tilde{\mathbf{v}}_i$, where i ranges from 1 to N, then writing out Eq. (38) in a form similar to Eq. (32) gives:

$$b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i) = 0 = \int_\Omega \left(\mathbf{q}_{j,h} - \mathbf{q}_j\right)^T L^*_{q,j}(\tilde{\mathbf{v}}_i)\,d\Omega + \int_\Omega (\mathbf{u}_h - \mathbf{u})^T\, L^*_u(\tilde{\mathbf{v}}_i)\,d\Omega + \int_{\partial\Omega} \mathbf{v}_i^T \left[\hat{\mathbf{F}}(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{\mathbf{F}}(\mathbf{u}, \vec{\mathbf{q}})\right] ds\,. \tag{39}$$

This equation is satisfied regardless of how the test space is chosen. However, when using optimal test functions, we would like this expression to represent the minimization of a certain error norm. It is important to note that, due to the form of $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$, only certain types of error norm are valid to minimize. In other words, we must be careful to select an error norm whose derivative it is *possible* to represent by $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$. To that end, we propose minimizing the following norm:

$$\|\,\tilde{\mathbf{e}}\,\|^2 = \underbrace{\sum_{s=1}^{r}\sum_{d=1}^{dim}\int_\Omega (q_{s,d,h} - q_{s,d})^2\,d\Omega}_{\text{interior } q \text{ accuracy}} + \underbrace{\sum_{s=1}^{r}\int_\Omega (u_{s,h} - u_s)^2\,d\Omega}_{\text{interior } u \text{ accuracy}} + \underbrace{\sum_{s=1}^{r} w_s \int_{\partial\Omega}\left[\hat{F}_s(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\right]^2 ds}_{\text{flux accuracy}} \tag{40}$$

This norm contains errors in the state, its gradients, and the boundary flux – all of which are present in the above $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$. Furthermore, with this norm, we see that choosing the weights w_s to be large emphasizes accuracy in the boundary fluxes, which, as in one dimension, is our ultimate goal.

If we are to minimize this norm, we need its derivatives with respect to both the $U_{k,m}$ and $Q_{k,d,m}$ coefficients to be zero. Thus, we need

$$\frac{1}{2}\frac{\partial\|\,\tilde{\mathbf{e}}\,\|^2}{\partial U_{k,m}} = 0 = \int_\Omega (u_{k,h} - u_k)\,\phi_{k,m}\,d\Omega + \sum_{s=1}^{r} w_s \int_{\partial\Omega}\left[\hat{F}_s(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\right]\frac{\partial \hat{F}_s}{\partial u_{k,h}}\,\phi_{k,m}\,ds \tag{41}$$

and

$$\frac{1}{2}\frac{\partial\|\,\tilde{\mathbf{e}}\,\|^2}{\partial Q_{k,d,m}} = 0 = \int_\Omega (q_{k,d,h} - q_{k,d})\,\phi_{k,d,m}\,d\Omega + \sum_{s=1}^{r} w_s \int_{\partial\Omega}\left[\hat{F}_s(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\right]\frac{\partial \hat{F}_s}{\partial q_{k,d,h}}\,\phi_{k,d,m}\,ds\,. \tag{42}$$

For these equations to be satisfied by our finite element method, we must choose the test functions $\tilde{\mathbf{v}}_i$ such that the bilinear form $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$ reduces to them. A given $\tilde{\mathbf{v}}_i$ will then ensure that *one* of the above equations is satisfied. Since Eqs. (41) and (42) represent N derivative equations altogether, with N test functions (i.e. a square system) we can ensure that each of them is satisfied in turn.

By comparing $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$ (Eq. (39)) to Eq. (41), we see that to make these expressions identical the test functions must satisfy:

$$i = 1 \ldots N_U \quad \begin{cases} L^*_{q,j}(\tilde{\mathbf{v}}_i) = \mathbf{0} & j = 1 \ldots dim \quad x \in \Omega \\ L^*_{u,s}(\tilde{\mathbf{v}}_i) = \phi_{k,m}\,\delta_{s,k} & s = 1 \ldots r \quad x \in \Omega \\ v_{i,s} = w_s \dfrac{\partial \hat{F}_s}{\partial u_{k,h}}\,\phi_{k,m} & s = 1 \ldots r \quad x \in \partial\Omega \end{cases} \tag{43}$$

Here, $\delta_{s,k}$ denotes the Kronecker delta function, $L^*_{u,s}$ denotes the sth component (i.e. equation) associated with the operator L^*_u, and repeated indices do not imply summation. As before, we see that the optimal test functions satisfy adjoint equations

in which the trial bases appear as source terms on the right-hand side. The above equations are solved for each u basis function $\phi_{k,m}$, with the test function index i enumerating all combinations of (k,m). Since $1 \le k \le r$ and $1 \le m \le n_U$, there are a total of $N_U = r\,n_U$ basis functions altogether, which provides, in the end, a corresponding N_U test functions.

Next, to make $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$ reduce to Eq. (42), we see that the remaining test functions should satisfy:

$$i = 1 \ldots N_Q \quad \begin{cases} L^*_{q,j,s}(\tilde{\mathbf{v}}_i) = \phi_{k,d,m}\,\delta_{j,d}\,\delta_{s,k} & j,s = 1 \ldots dim, r \quad x \in \Omega \\ L^*_u(\tilde{\mathbf{v}}_i) = \mathbf{0} & x \in \Omega \\ v_{i,s} = w_s \dfrac{\partial \hat{F}_s}{\partial q_{k,d,h}} \phi_{k,d,m} & s = 1 \ldots r \quad x \in \partial\Omega \end{cases} \tag{44}$$

This set of equations is solved for each q trial basis $\phi_{k,d,m}$, with the test function index i enumerating all combinations of (k,d,m), where $1 \le k \le r$, $1 \le d \le dim$, $1 \le m \le n_Q$. The result is an additional $N_Q = r\,n_Q \cdot dim$ test functions, for a total of $N_U + N_Q = N$. When used in place of the standard Galerkin test functions, these optimal test functions ensure that Eqs. (41) and (42) are satisfied, and hence that the error in Eq. (40) is minimized.

Finally, as in one dimension, the optimal test functions can be interpreted as adjoint solutions for certain "projection" outputs. These outputs are closely related to the error norm derivatives. By inspection of Eqs. (41) and (42), we can write the effective outputs as

$$J^u_{k,m} = \int_\Omega u_k\,\phi_{k,m}\,d\Omega + \sum_{s=1}^{r} w_s \int_{\partial\Omega} \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}}) \frac{\partial \hat{F}_s}{\partial u_{k,h}} \phi_{k,m}\,ds \tag{45}$$

and

$$J^q_{k,d,m} = \int_\Omega q_{k,d}\,\phi_{k,d,m}\,d\Omega + \sum_{s=1}^{r} w_s \int_{\partial\Omega} \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}}) \frac{\partial \hat{F}_s}{\partial q_{k,d,h}} \phi_{k,d,m}\,ds\,. \tag{46}$$

It is easy to verify that enforcing zero error in these outputs is equivalent to enforcing zero derivative of $\|\tilde{\mathbf{e}}\|^2$ – which of course is the actual goal.

8.1. Localization and adjoint consistency

As in 1D, the above test functions satisfy global adjoint equations. However, if we choose the flux weights w_s to be large, we can again localize the adjoint problems to individual elements, since accuracy in the local fluxes will propagate globally. Thus, for multi-element meshes, the outputs J^u and J^q are defined over each element in turn, and the $\hat{F}_s$ terms are chosen to reflect the fluxes on a given element's boundaries.

Specifically, to define the adjoint problems for a domain-interior element, the neighbor-element states are treated as local Dirichlet conditions, and the fluxes $\hat{F}_s$ are defined as if a single-element DG or HDG problem were being solved. Since there is no physical boundary condition on interior faces, the convective part of $\hat{F}_s$ is just taken to be a Roe flux [31] between the element and the neighbor states. This Roe flux ensures that information is properly upwinded and that the local adjoint problems remain well-posed.[8]

Likewise, for an element with a domain-boundary face, the flux $\hat{F}_s$ is defined as usual for a DG or HDG method – for example, the convective flux is just the analytical flux function evaluated with a corresponding boundary state. Furthermore, if the boundary condition specifies a certain flux component directly (e.g. if it is a wall boundary, where zero mass flux is specified), then there is no need to request accuracy in this flux component, and it should be removed from the outputs J^u and J^q. This removal occurs automatically when using a discrete adjoint approach, since the discrete $\partial \hat{F}_s / \partial u_h$ and $\partial \hat{F}_s / \partial q_h$ will be zero and will hence vanish from both the output definitions and the residual Jacobian.

Fig. 5 shows a set of localized optimal test functions corresponding to a low Reynolds number advection–diffusion problem. The "upwinding" nature of the optimal test functions is apparent, and we see that their role is to ensure that the fluxes on a given element face have the proper domain of dependence *within* that element.

8.2. Boundary enrichment of trial space

Localization of the test functions has obvious advantages: it makes their computation relatively inexpensive and their application within an existing method straightforward. However, in multiple dimensions, localization is – in some ways – a double-edged sword. By using localized test functions, we are giving up certain global properties of the test space, and focusing all of our attention on achieving accuracy in the interface fluxes. And while it is true that *if* these local fluxes can be made accurate, then this accuracy will propagate globally, it is also true that if these fluxes *cannot* be made accurate, then this *inaccuracy* will propagate globally.

[8] For these local problems, use of a non-upwinding flux such as Lax–Friedrichs would result in an adjoint inconsistency [32,33], and would lead to oscillations in (and suboptimal performance of) the test functions.

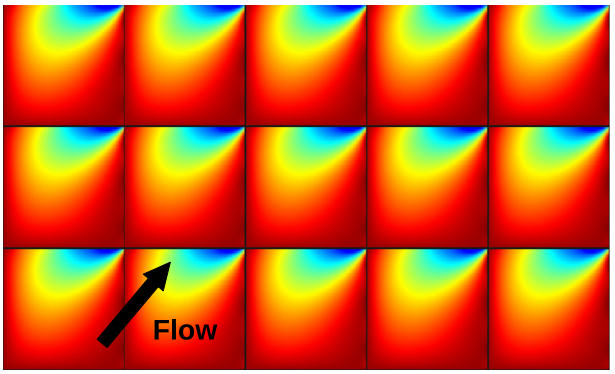


Fig. 5. Optimal test functions corresponding to the q_y (gradient in the vertical direction) trial basis in the upper-right corner of each element. Blue corresponds to a large value, while red is near zero. These test functions ensure that accuracy in the top flux of each element is obtained. Similar test functions ensure accuracy in the remaining fluxes. Note the upwinding nature of the test functions. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

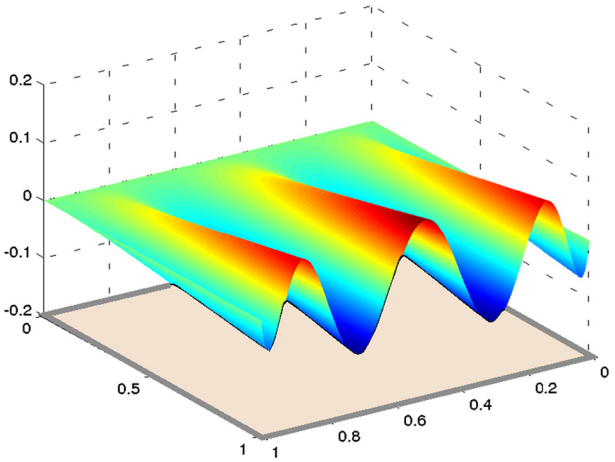


Fig. 6. An eighth-order Lobatto function defined along an edge of a quadrilateral reference element. These functions are added to the trial space to improve flux resolution.

In one dimension, the fluxes are just scalar values, since the boundaries of a given element are zero-dimensional. Thus, if the optimal test functions are well-represented, the flux errors can be driven to zero regardless of the trial space. However, in two dimensions, the fluxes themselves are one-dimensional profiles along the element boundaries. In this case, the pointwise errors in the fluxes cannot in general be driven to zero: their magnitude depends inevitably on the *trial* space resolution near element boundaries. For example, if the exact fluxes happen to be quadratic along a given face, but the trial space is linear, then $O(h^2)$ errors in the fluxes will necessarily remain. These flux errors will then propagate globally, and, in many cases, would render methods using localized optimal test functions no better than standard DG methods.

Thus, if optimal test functions are to provide a benefit in multiple dimensions, not only must the test functions *request* accuracy in the fluxes, but the trial space must be capable of *providing* that accuracy. To ensure that the latter requirement is satisfied, in multiple dimensions the trial space can be enriched near element boundaries. In the current work, this is done by adding high-order one-dimensional Lobatto functions [34] along the element faces, which are then blended linearly into the element interior. Fig. 6 shows an example of a blended eighth-order Lobatto function defined on a single edge of a reference quadrilateral.

In the results section to follow, this is the enrichment strategy used. Note that we keep the interior interpolation order, p_I, at a (low) value of 1, and then enrich this $p_I = 1$ space with Lobatto functions of some higher order, p_B. The combination of linear interior basis with order-p_B Lobatto functions ensures that the trial space on element boundaries spans a full order-p_B space. In the end, this is equivalent to using a standard order-p_B hierarchical basis, but with all interior modes removed.

8.3. Benefit in multiple dimensions

Before moving on, it is worth pausing to reflect on the potential benefit of optimal test functions in multiple dimensions. The primary advantage is that, by requiring trial space resolution only near element boundaries, a BDPG scheme requires fewer degrees of freedom than a DG method. For example, on a two-dimensional quadrilateral mesh, BDPG requires $4p_B$

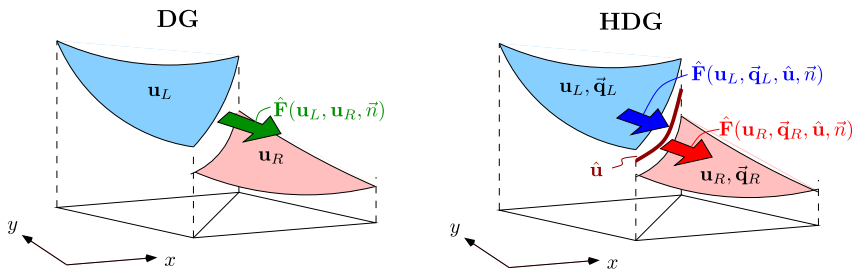


Fig. 7. In the HDG method, additional unknowns on element interfaces allow for elimination of the element-interior degrees of freedom. This results in a global system size in which the number of unknowns scales as p^{dim-1} instead of p^{dim} for DG.

degrees of freedom, whereas for a similar level of accuracy DG would require $(p_B+1)^2$. Thus, the number of degrees of freedom scales as p^{dim} for DG methods, but as p^{dim-1} for BDPG. In a sense then, BDPG may be viewed as a form of "hybridization," since hybridization of a standard DG scheme results in a similar reduction in the number of globally coupled unknowns.

However, even within an existing hybrid framework (such as HDG), optimal test functions can still provide a benefit. Since BDPG requires trial space resolution only near element boundaries, it opens up the possibility of performing a targeted trial space optimization in those regions. For example, if the trial space were tuned to include the primary "modes" of the true interface fluxes, then hybridized BDPG schemes could significantly outperform standard HDG schemes. As a step in this direction – and to show that optimal test functions can be used within a hybrid framework – in the following sections we present a hybridized BDPG method.

9. A hybridized BDPG method

In this section, we first give a brief overview of hybridized discontinuous Galerkin (HDG) methods [35–37]. We then describe how these methods can be modified to incorporate optimal test functions, resulting in a hybridized BDPG scheme. The primary advantage of hybridized methods is that, by introducing new unknowns on element interfaces, they decouple elements during the linear solve and (for sufficiently high order p [35]) result in a smaller global system than DG. An illustration of the primary differences between DG and HDG is provided in Fig. 7.

9.1. HDG discretization

While the HDG method can be applied to general nonlinear systems, here we consider a linear steady-state system in conservation form:

$$\vec{\mathbf{q}} - \nabla \mathbf{u} = \vec{\mathbf{0}}, \tag{47}$$

$$\nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \vec{\mathbf{q}}) = \mathbf{0}, \tag{48}$$

where $\mathbf{u}$ is the state, $\vec{\mathbf{q}}$ is the state gradient, and $\vec{\mathbf{F}}$ is the conservative flux. Weighting the above equations with test functions, discretizing, and integrating by parts yields, for an element K:

$$R^Q \equiv \int_K \vec{\boldsymbol{\tau}}_h^T \cdot \vec{\mathbf{q}}_h \, d\Omega + \int_K \nabla \cdot \vec{\boldsymbol{\tau}}_h^T \mathbf{u}_h \, d\Omega - \int_{\partial K} \left(\vec{\boldsymbol{\tau}}_h^T \cdot \vec{n} \right) \hat{\mathbf{u}}_h \, ds = 0 \qquad \forall \vec{\boldsymbol{\tau}}_h \in [\mathcal{V}_h]^{dim}, \tag{49}$$

$$R^U \equiv -\int_K \nabla \mathbf{v}_h^T \cdot \vec{\mathbf{F}} \, d\Omega + \int_{\partial K} \mathbf{v}_h^T \hat{\mathbf{F}}_L \, ds = 0 \qquad \forall \mathbf{v}_h \in \mathcal{V}_h, \tag{50}$$

$$R^\Lambda \equiv \int_f \boldsymbol{\mu}_h^T \left\{ \hat{\mathbf{F}}_L + \hat{\mathbf{F}}_R \right\} ds = 0 \qquad \forall \boldsymbol{\mu}_h \in \mathcal{M}_h. \tag{51}$$

Here, an additional variable $\hat{\mathbf{u}}_h$ has been introduced on element interfaces. The test spaces $\mathcal{V}_h$ and $\mathcal{M}_h$ are polynomials on an element and its adjacent interfaces, respectively, while the third equation is a weak flux continuity statement required to close the system, since the fluxes on either side of an interface f need not match pointwise. These "one-sided" fluxes are defined as

$$\hat{\mathbf{F}}_L = \vec{\mathbf{F}}\left(\hat{\mathbf{u}}_h, \vec{\mathbf{q}}_{h,L}\right) \cdot \vec{n}_L + \underline{\underline{\mathbf{S}}}\left(\hat{\mathbf{u}}_h\right)\left(\mathbf{u}_{h,L} - \hat{\mathbf{u}}_h\right) \tag{52}$$

and likewise for $\hat{\mathbf{F}}_R$, where the L and R indicate the side of a given face with which the flux is associated. (In Eq. (50) we arbitrarily designate the "left" side as that lying within element K, and define $\vec{n}_L$ to be the outward-pointing normal to K.) Finally, the $\underline{\underline{\mathbf{S}}}$ term in the above expression is a stabilization tensor, which to obtain a Roe-like flux can be chosen as

$$\underline{\underline{\mathbf{S}}} = \underline{\underline{\mathbf{R}}} \, |\underline{\underline{\boldsymbol{\Lambda}}}| \, \underline{\underline{\mathbf{L}}} + \tau_{\text{visc}} \underline{\underline{\mathbf{I}}}. \tag{53}$$

Here, $\tau_{\text{visc}} = \nu/\ell_{\text{visc}}$ includes the viscosity ν and a user-specified viscous length scale ℓ_{visc}, while the matrices $\underline{\underline{\mathbf{R}}}$, $\underline{\underline{\mathbf{\Lambda}}}$, and $\underline{\underline{\mathbf{L}}}$ come from an eigen-decomposition of the convective flux Jacobian evaluated about $\hat{\mathbf{u}}_h$.

The above $\hat{\mathbf{F}}_L$ and $\hat{\mathbf{u}}_h$ are defined on all interior faces. For faces on domain boundaries, no $\hat{\mathbf{u}}_h$ is employed, and the one-sided fluxes are instead replaced by a standard boundary flux. As with DG, this boundary flux (which we will call simply $\hat{\mathbf{F}}$) consists of the analytical flux function evaluated with an appropriate boundary state $\mathbf{u}_B$.[9] Finally, a stabilization tensor $\underline{\underline{\mathbf{S}}}^B = \tau_{\text{visc}}\underline{\underline{\mathbf{I}}}$ is also included, so that the total boundary flux is given by

$$\hat{\mathbf{F}} = \vec{\mathbf{F}}\left(\mathbf{u}_B, \vec{\mathbf{q}}_{h,L}\right)\cdot\vec{n}_L + \underline{\underline{\mathbf{S}}}^B\left(\mathbf{u}_{h,L} - \mathbf{u}_B\right). \tag{54}$$

In the end, when the HDG system described above is assembled, the interior degrees of freedom on each element can be *statically condensed*, resulting in a smaller global system involving the $\hat{\mathbf{u}}_h$ unknowns alone. After solving this system for $\hat{\mathbf{u}}_h$, the interior states $\mathbf{u}_h$ and $\vec{\mathbf{q}}_h$ can be computed through a series of element-local solves.

9.2. Optimal test function (BDPG) implementation

Next, we give a brief overview of using optimal test functions within the above HDG framework. First, we note that for single-element problems, Eq. (51) vanishes and the sum of R^Q and R^U (Eqs. (49) and (50)) reduces to the bilinear form in Eq. (29). The test function theory derived in Section 8 therefore carries over directly, and the optimal test functions make Eqs. (49) and (50) reduce to the error norm derivatives in Eqs. (41) and (42), thus minimizing the desired error.

For multi-element problems, localized optimal test functions can be computed as described in Section 8.1. This amounts to solving the following $N_U + N_Q$ adjoint problems for the test functions $\tilde{\mathbf{v}}_i = [\boldsymbol{\tau}_j\ \mathbf{v}]^T$ on each element K:

$$b_K\left(\delta\tilde{\mathbf{u}}, \tilde{\mathbf{v}}_i\right) = J_i^u\left(\delta\tilde{\mathbf{u}}\right) \qquad \forall\,\delta\tilde{\mathbf{u}} \in \tilde{\mathcal{U}}_{\text{test}} \qquad i = 1..N_U \tag{55}$$

$$b_K\left(\delta\tilde{\mathbf{u}}, \tilde{\mathbf{v}}_i\right) = J_i^q\left(\delta\tilde{\mathbf{u}}\right) \qquad \forall\,\delta\tilde{\mathbf{u}} \in \tilde{\mathcal{U}}_{\text{test}} \qquad i = 1..N_Q \tag{56}$$

Here, $b_K(\cdot,\cdot)$ is the bilinear form given by Eq. (36) (corresponding to a single-element problem on element K), J_i^u and J_i^q are the outputs defined in Eqs. (45) and (46), and $\tilde{\mathcal{U}}_{\text{test}}$ is an enriched space of order p_{test}. These equations can be written in discrete form as

$$\frac{\partial\tilde{\mathbf{R}}}{\partial\tilde{\mathbf{U}}}^T\tilde{\mathbf{V}}_i = \frac{\partial J_i^u}{\partial\tilde{\mathbf{U}}}^T \qquad i = 1..N_U \tag{57}$$

$$\frac{\partial\tilde{\mathbf{R}}}{\partial\tilde{\mathbf{U}}}^T\tilde{\mathbf{V}}_i = \frac{\partial J_i^q}{\partial\tilde{\mathbf{U}}}^T \qquad i = 1..N_Q \tag{58}$$

and solved to find the test function coefficients $\tilde{\mathbf{V}}_i$. Here, the single-element Jacobian matrix $\partial\tilde{\mathbf{R}}/\partial\tilde{\mathbf{U}}$ contains contributions from both the R^Q and R^U residuals, while $\tilde{\mathbf{V}}_i$ and $\tilde{\mathbf{U}}$ contain the coefficients associated with $\vec{\boldsymbol{\tau}}_i, \mathbf{v}_i$ and $\vec{\mathbf{q}}, \mathbf{u}$, respectively.

When used to weight the R^Q and R^U residuals, the optimal test functions result in the following set of equations on each interior element and its faces:

$$J_i^u(\mathbf{u}_h, \vec{\mathbf{q}}_h) + \int_{\partial K}\mathbf{v}_i^T\left(\hat{\mathbf{F}}_L - \hat{\mathbf{F}}\right)ds = \int_{\partial K}\left(\vec{\boldsymbol{\tau}}_i\cdot\vec{n}\right)^T\hat{\mathbf{u}}_h\,ds \qquad i = 1..N_U \tag{59}$$

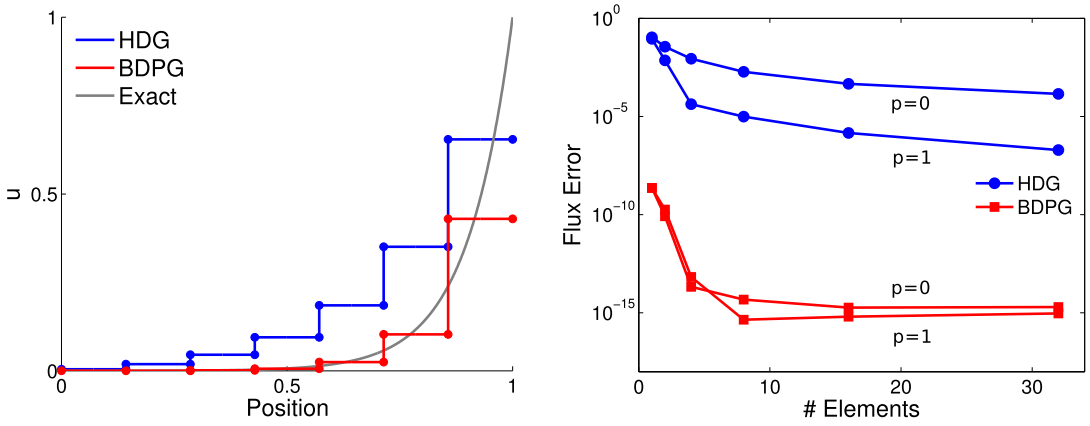
$$J_i^q(\mathbf{u}_h, \vec{\mathbf{q}}_h) + \int_{\partial K}\mathbf{v}_i^T\left(\hat{\mathbf{F}}_L - \hat{\mathbf{F}}\right)ds = \int_{\partial K}\left(\vec{\boldsymbol{\tau}}_i\cdot\vec{n}\right)^T\hat{\mathbf{u}}_h\,ds \qquad i = 1..N_Q \tag{60}$$

$$\int_f \boldsymbol{\mu}_h^T\left\{\hat{\mathbf{F}}_L + \hat{\mathbf{F}}_R\right\}ds = 0 \qquad \forall\boldsymbol{\mu}_h \in \mathcal{M}_h. \tag{61}$$

Here, the $\hat{\mathbf{F}}_L - \hat{\mathbf{F}}$ terms arise due to the fact that the flux $\hat{\mathbf{F}}$ used in the adjoint problems is not necessarily identical to the one-sided flux $\hat{\mathbf{F}}_L$. This is because, as discussed in Section 8.1, $\hat{\mathbf{F}}$ represents the flux of a single-element Dirichlet problem (and contains a full Roe flux), whereas $\hat{\mathbf{F}}_L$ depends solely on "one-sided" information and in most cases is only approximately equal to $\hat{\mathbf{F}}$.

Finally, note that we leave the face residual (Eq. (61)) the same as in standard HDG – i.e. we do not compute optimal test functions for the interface states $\hat{\mathbf{u}}_h$. Instead, we view the interface states as a passive "glue" that transmits the boundary accuracy on a given element to its neighbors across each face. Thus, if the interior trial space is of order p_B on element boundaries, we choose $\mathcal{M}_h$ to be a standard order-p_B polynomial space, and take $\hat{\mathbf{u}}_h \in \mathcal{M}_h$. Lastly, we note that in the final formulation (Eqs. (59)–(61)) the optimal test functions appear only on element boundaries, so that no high-order interior integration is required.

[9] In certain cases, e.g. at farfield boundaries, a Roe flux is used rather than the analytical convective flux.



(a) Sample $p = 0$ solutions (b) Right-flux error

Fig. 8. One-dimensional advection–diffusion: (a) Sample $p = 0$ solutions for both HDG and BDPG. (b) Convergence of the right-boundary flux, where $w = 10^{15}$ was used for BDPG. BDPG provides interior accuracy while achieving significantly greater flux accuracy than HDG.

9.3. Summary of hybrid BDPG method

The multi-dimensional BDPG method can be summarized as follows:

1. Use an order-p_B hierarchical basis for the trial space of $\mathbf{u}$ and $\vec{\mathbf{q}}$, but with all interior modes removed.
2. Use an order-p_B space for the interface states $\hat{\mathbf{u}}$.
3. Loop over each element.
4. Compute the local optimal test functions $\tilde{\mathbf{v}}_i$ by solving Eqs. (57) and (58) at order p_{test}, where $p_{\text{test}} \geq p_B$. Choosing $p_{\text{test}} = p_B$ often suffices.
5. Normalize the $\tilde{\mathbf{v}}_i$ with respect to a discrete or continuous L^2 norm.
6. Use the $\tilde{\mathbf{v}}_i$ to weight R^Q and R^U, while using a standard order-p_B test space for R^Λ. This amounts to solving Eqs. (59)–(61).

10. Results

In this section, we present results for the hybrid BDPG method and compare its performance to standard HDG. We begin with a one-dimensional advection–diffusion problem before progressing to two-dimensional boundary layer and airfoil cases.

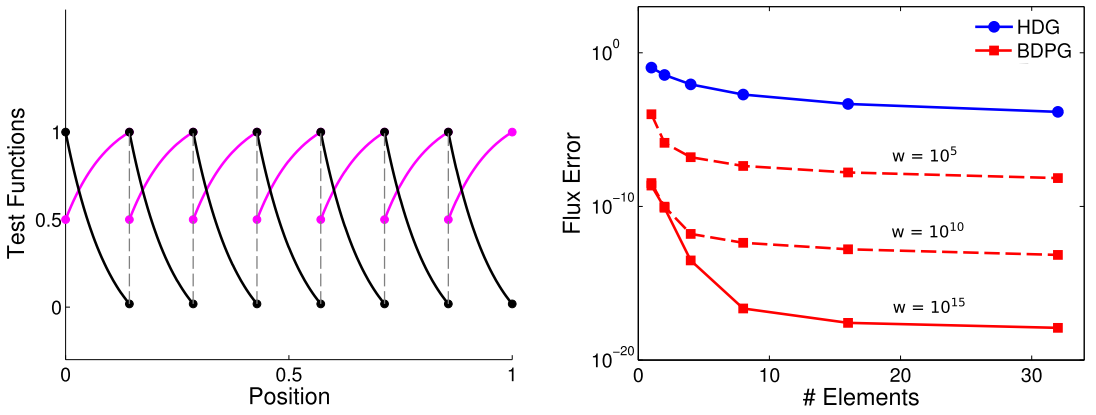
10.1. One-dimensional advection–diffusion

We have shown that BDPG performs well for one-dimensional problems with advection. To demonstrate its performance for viscous problems, we solve the following advection–diffusion equation:

$$\begin{aligned} a\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} &= 0 \qquad x \in \Omega \\ u &= 0 \qquad x = x_L \\ u &= 1 \qquad x = x_R\,. \end{aligned} \tag{62}$$

As an example, we choose the Reynolds number to be $aL/\nu = 10$ (where L is the domain width), the trial space orders to be $p = 0$ and $p = 1$, and the test space order to be $p_{\text{test}} = 10$. The viscous length ℓ_{visc} is kept fixed at $O(1)$. Sample $p = 0$ solutions for HDG and BDPG are shown in Fig. 8(a), while Fig. 8(b) gives the convergence of the right-boundary flux for $p = 0$ and $p = 1$ runs.

We see that, with a boundary weight of $w = 10^{15}$, BDPG provides nearly 10 orders of magnitude lower flux errors than HDG, while maintaining interior accuracy in u. Furthermore, Fig. 9(b) shows that (as expected) the choice of w determines the amount of flux accuracy obtained, with higher w leading to proportionally greater accuracy. Finally, the optimal test functions for the $p = 0$ case are shown in Fig. 9(a). These test functions provide accuracy in both the left and right fluxes



(a) Optimal test functions for $p = 0$ (b) Left-flux error

Fig. 9. One-dimensional advection–diffusion: (a) Normalized v-component of the optimal test functions for the $p = 0$ solution in Fig. 8(a). The black test functions are associated with the q trial bases, while the remaining test functions are associated with the u bases. (b) Convergence of the left-boundary flux for $p = 0$ and various choices of boundary weight, w. The higher the boundary weight, the more accurate the flux.

leaving each element, which leads ultimately to accuracy in the domain-boundary fluxes. As in the advection–reaction example, the initial convergence rate of the BDPG fluxes (which is observed to be $p_{\text{test}} + p + 1$) is due solely to the inexact representation of the test functions. Thus, if analytical rather than numerical optimal test functions were used, machine-precision flux accuracy would be obtained on any mesh.

Remark 9. Note that for this problem, if we were to consider pure advection or pure diffusion (by setting $\nu = 0$ or $a = 0$, respectively), HDG would achieve exact fluxes without the need for optimal test functions. This is because, in general, the adjoints for the fluxes satisfy homogeneous equations of the form $L^* v = 0$. This equation reduces to $L^* v = -a\frac{\partial v}{\partial x} = 0$ for advection and $L^* v = -\nu\frac{\partial^2 v}{\partial x^2} = 0$ for diffusion. The solutions to these equations are just constant and linear functions, respectively. For $p \geq 1$, HDG already contains these adjoint solutions in its test space, so in that sense it is already optimal.

10.2. Two-dimensional advection–diffusion: manufactured solution

Next, we move on to two dimensions. Before solving practical problems, we investigate two ideas related to the multi-dimensional test function theory: first, whether adequate trial space representation of the fluxes *is* actually important (as claimed in Section 8.2); and second, whether, given adequate flux representation, the localized optimal test functions *can* actually provide boundary accuracy.

As mentioned, in two dimensions we expect BDPG to perform well only if the trial space is capable of adequately representing the true fluxes. In order to confirm this theory, we construct a manufactured solution whose true fluxes lie exactly in a $p = 1$ space. This solution is given by

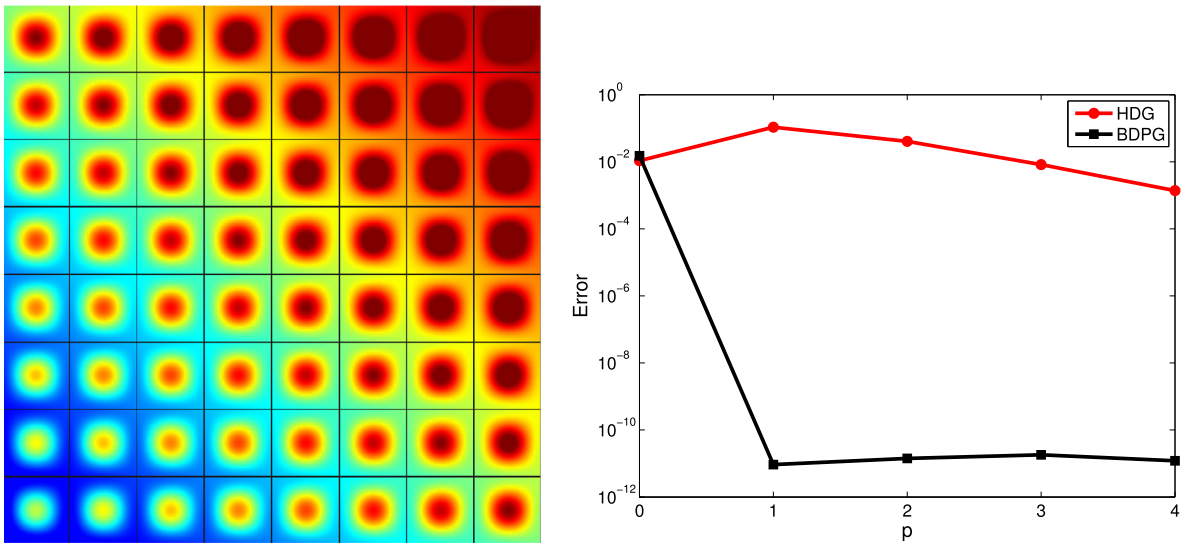
$$u(x, y) = \sin(8\pi x)^2 \sin(8\pi y)^2 + x + y\,, \tag{63}$$

with contours shown in Fig. 10. Note that the sinusoidal terms vanish on all element boundaries, thus leaving a linear $(x + y)$ variation there. To define the problem, the advective velocity is chosen to be $\vec{a} = [0.4, 0.8]$, the viscosity is taken to be $\nu = 0.01$, and the domain has length $L = 1$.

We then solve the problem using BDPG with a standard trial space – i.e. with no Lobatto enrichment on element boundaries. When using a $p = 0$ trial space, we expect the performance of BDPG to suffer, since the true (linear) fluxes cannot be adequately represented. However, as soon as the trial space order is increased to $p = 1$, the true fluxes become representable, and we expect BDPG to be capable of delivering nearly exact boundary values.

To perform the test, we sweep through trial space orders from $p = 0$ to $p = 4$ and record the error in the boundary fluxes for both BDPG and HDG. For BDPG, we keep the test space order fixed at a high value of $p_{\text{test}} = 10$ to ensure that the test function representation has a minimal influence on the results.

Fig. 10 shows the error in the top-boundary flux for both methods, which is representative of the fluxes on all boundaries. The results are as we expect: for $p = 0$, the BDPG errors are large since the flux is not representable, but as soon as $p = 1$ is used, the error drops to machine-precision levels. This highlights the importance of flux resolution for multidimensional problems, and justifies the idea of enriching the trial space near element boundaries. Furthermore, the performance of BDPG for $p \geq 1$ confirms that the optimal test space is functioning well, since if it were not, boundary accuracy – even *with* adequate flux resolution – would not be achieved. This point is demonstrated clearly by the performance of standard HDG, which due to its suboptimal test space has nearly 10 orders of magnitude larger errors than BDPG.



(a) Manufactured solution (b) Top flux error convergence

Fig. 10. Two-dimensional advection–diffusion: (a) Manufactured solution with fluxes that are exactly representable in a $p = 1$ space. (b) Convergence of the top-boundary flux as a function of p. As the order is increased above $p = 0$, the fluxes become representable and BDPG attains machine-precision accuracy. This verifies the performance of BDPG and highlights the importance of flux resolution in multiple dimensions.

Finally, we note that in order to achieve the machine-precision accuracy shown in Fig. 10, the viscous length scale for BDPG had to be taken small; specifically, a value of $\ell_{\text{visc}} = 10^{-7}$ was used for the $p \geq 1$ runs. (A more standard value of $\ell_{\text{visc}} = 10^{-1}$ was used for all other runs.) This suggests that for multidimensional viscous problems, the test function localization becomes more effective as the elements become more tightly coupled, since the effect of a small viscous length is to penalize the inter-element jumps in u. While this issue warrants further analysis, we find that for practical problems (where the fluxes are not exactly representable) the performance of BDPG is relatively insensitive to the choice of viscous length, and more modest values of ℓ_{visc} can be used. Finally, note that this issue does not arise for inviscid problems, since in that case no ℓ_{visc} is defined. Indeed, when a similar manufactured solution is solved with the linearized Euler equations, BDPG attains machine-precision boundary fluxes with no "free" parameters involved.

10.3. Two-dimensional advection–diffusion: boundary layer

Next, we try a more practical advection–diffusion problem. With the same domain as above, we take $\vec{a} = [0.8, 0.4]$, $\nu = 0.01$ (so that $Re \approx 100$), and specify a Dirichlet boundary condition on all sides of the domain given by

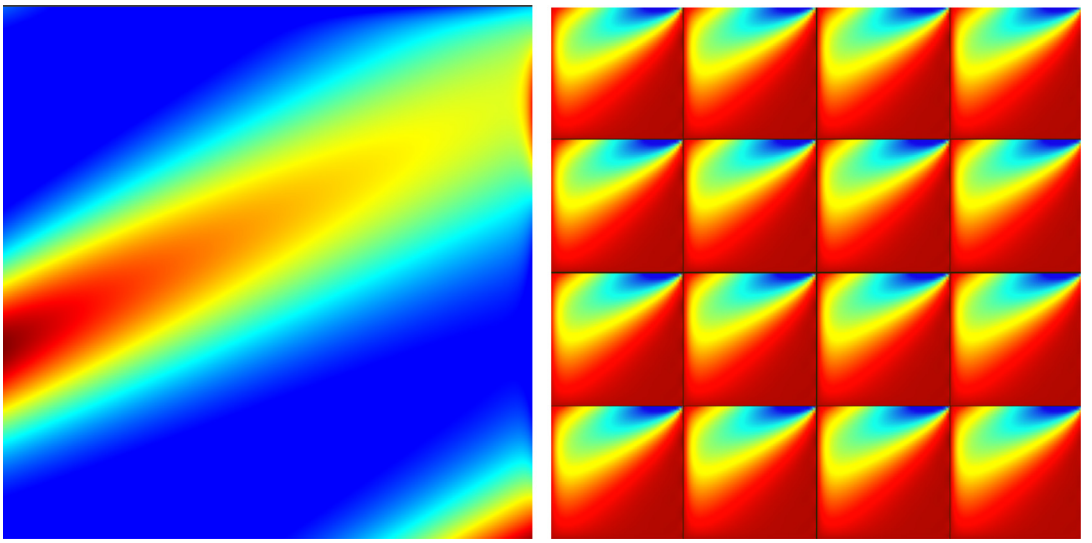
$$u(x, y) = \exp\left[\frac{1}{2}\sin(-4x + 6y) - \frac{4}{5}\cos(3x - 8y)\right] \qquad \vec{x} \in \partial\Omega\,. \tag{64}$$

This condition generates boundary layers on the two outflow boundaries (the top and right), and provides a test as to whether BDPG can accurately predict these features. Fig. 11 shows contours of both the solution and the optimal test functions (which are again computed with $p_{\text{test}} = 10$) for a $p = 1$ trial space.

Since we verified above that the resolution of interface fluxes is critical for BDPG, we enrich the trial space with Lobatto functions near element boundaries. For the results shown in Fig. 12, we consider enrichment orders of $p_B = 6, 7, 8$, while keeping the interior basis at a low order of $p_I = 1$. We compare the BDPG results to a standard HDG method with the same interior trial space order of $p_I = 1$. Viscous lengths of $\ell_{\text{visc}} = 10^{-4}$ and $\ell_{\text{visc}} = 1$ are used for BDPG and HDG, respectively, with the results being relatively insensitive to these values.

From Fig. 12, we see that BDPG with Lobatto enrichment can provide a nearly 6-orders-of-magnitude reduction in the flux errors, while maintaining accuracy in interior outputs. Furthermore, in addition to providing accuracy in the total flux through each boundary, BDPG also achieves accuracy in the solution *profiles* along the boundaries. The solution and gradient profiles along the right boundary of the domain are shown in Fig. 13, from which the enhanced accuracy of BDPG is apparent.

These results, of course, should be kept in perspective: the boundary enrichment of BDPG represents an additional expense (and additional degrees of freedom) compared to $p = 1$ HDG, so the comparison is in that sense unfair. However,



(a) Solution, u (b) Optimal test function component

Fig. 11. Two-dimensional advection–diffusion: (a) The solution to a $Re = 100$ problem on a fine mesh. (b) The optimal test functions associated with the upper-right q_y trial basis on each element. Note the upwinding nature of the test functions.

the results demonstrate clearly that, with BDPG, the attainment of global boundary accuracy depends solely on the ability to resolve the interface fluxes – a fact that is not true of standard HDG,[10] and one that may be capitalized on in the future.

10.4. Two-dimensional linearized Euler: manufactured solution

With the performance of BDPG verified for scalar problems in one and two dimensions, we next move on to two-dimensional *systems*. In particular, we solve the homentropic linearized Euler equations, with state variables and fluxes given by

$$\mathbf{u} = \begin{bmatrix} p \\ u_i \end{bmatrix}, \quad \mathbf{F}_j = \begin{bmatrix} u_{0_j} p + \rho_0 a_0^2 u_j \\ \frac{p}{\rho_0}\delta_{ij} + u_{0_j} u_i \end{bmatrix}, \tag{65}$$

where $1 < i, j < \text{dim}$. The state variables $\mathbf{u}$ represent velocity and pressure perturbations about the background state, which is described by the parameters a_0, u_{0_j}, and ρ_0 (speed of sound, velocity, and density, respectively).

As an initial test, we construct a manufactured solution on a square domain ($L = 1$) given by

$$\begin{aligned} p(x, y) &= \sin(8.5x)\sin(8.5y) \\ u_i(x, y) &= 0\,. \end{aligned} \tag{66}$$

A plot of the pressure contours is provided in Fig. 14, along with a set of optimal test functions. The background state is chosen as $\rho_0 = 1$, $a_0 = 3$, $u_{0_1} = 0.8$, and $u_{0_2} = 0.2$, so that the Mach number is approximately 0.3. We again choose the test space order and boundary weights to be high (10 and 10^{10}, respectively), and consider the same boundary enrichment orders as in the previous section.

The results shown in Fig. 15 are encouraging, and mirror those obtained in the two-dimensional advection–diffusion case. We see that BDPG achieves output error reductions of over 10 orders of magnitude compared to HDG at the same interior trial space order. Furthermore, BDPG also obtains accurate boundary profiles, as shown in Fig. 16. These results verify the effectiveness of optimal test functions for systems of equations.

10.5. Two-dimensional linearized Euler: cylinder and airfoil

Lastly, we consider linearized Euler cases of engineering interest: subsonic flow over a cylinder and an airfoil. For these cases, the background state is chosen as $\rho_0 = 1$, $a_0 = 3$, $u_{0_1} = 1$, $u_{0_2} = 0$, so that the flow is horizontal and the Mach number is approximately 0.3. In addition, the farfield boundary conditions on the state variables are $p = 1$, $u_1 = 1$, and $u_2 = 1$, so that a uniform perturbation travels upward and to the right. Finally, the mesh elements themselves are curved and are represented with $Q = 4$ polynomials, providing a first test of BDPG with curved geometry.

[10] Fig. 18(b) provides an explicit demonstration of the fact that, with standard HDG, flux resolution does not guarantee accuracy.

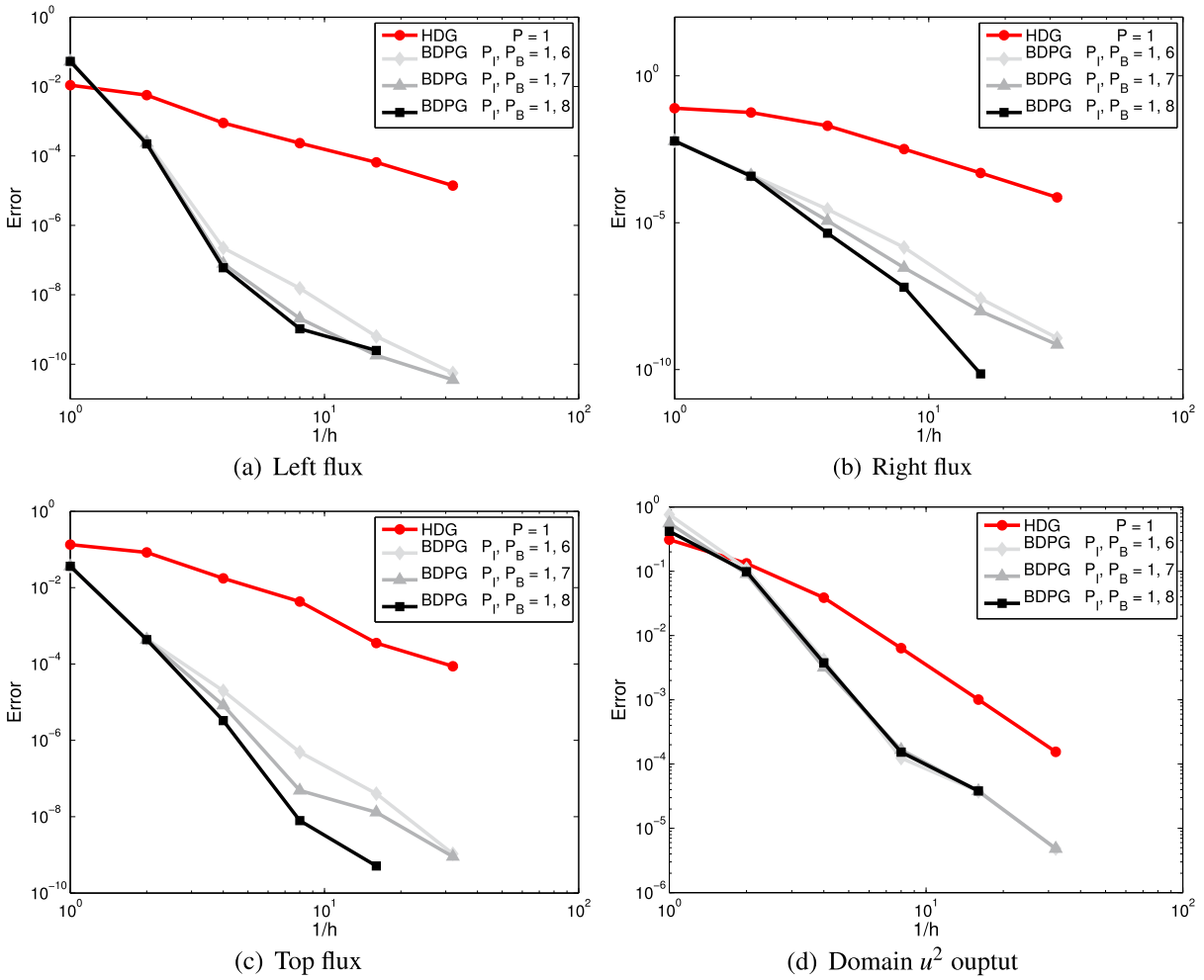


Fig. 12. Two-dimensional advection–diffusion: Convergence rates for various outputs. Note that p_I and p_B denote the interior and boundary interpolation orders, respectively. Higher accuracy is obtained with BDPG as the amount of boundary enrichment increases. Note that BDPG also achieves accuracy in the interior u^2 output.

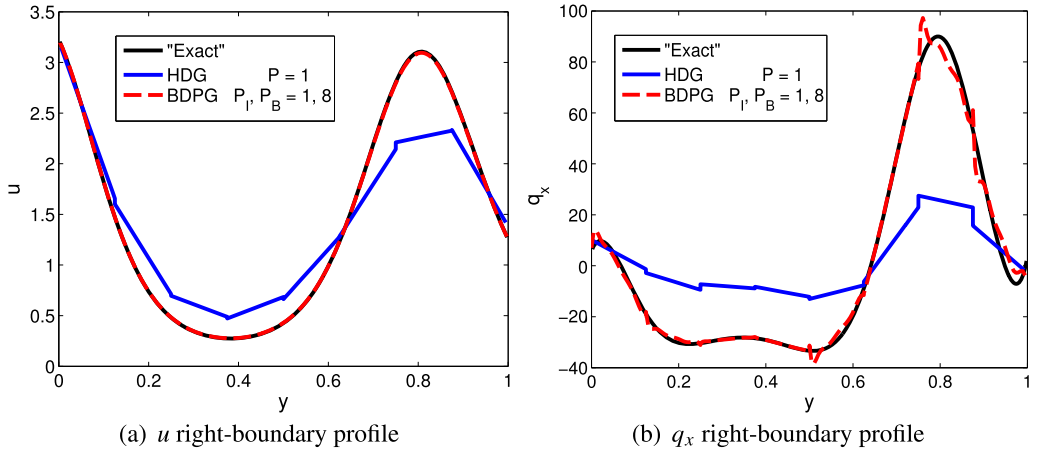
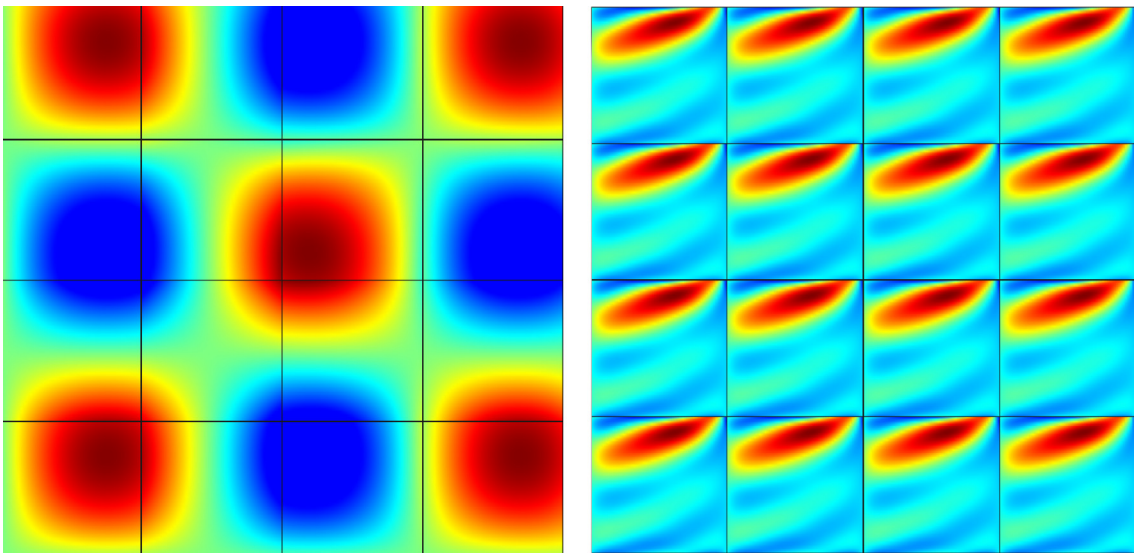


Fig. 13. Two-dimensional advection–diffusion: Solution profiles along the right boundary of the domain. BDPG with enrichment achieves greater accuracy than standard HDG.



(a) Manufactured solution (pressure contours) (b) Optimal test function component

Fig. 14. Two-dimensional linearized Euler: (a) Manufactured solution pressure contours. (b) Component of the optimal test functions corresponding to the trial basis in the upper-right corner of each element.

First, we simulate the flow around a cylinder of radius unity. Solution contours are shown in Fig. 17, while the convergence of the pressure flux along the cylinder wall is shown in Fig. 18(a). If the trial space is enriched appropriately, we see that BDPG can achieve nearly 6 orders of magnitude lower flux errors than HDG.

To demonstrate that these accuracy gains are not *just* due to the trial space enrichment, we perform another HDG simulation in which the same boundary-enriched ($p_B = 8$) trial space is used as for BDPG. In this case, the only difference between BDPG and HDG is the test space. Fig. 18(b) shows the results of this comparison. We see that BDPG still achieves flux errors that are nearly 6 orders of magnitude lower than HDG. This again demonstrates that, in multiple dimensions, it is the combination of both optimal test functions *and* trial space resolution that is critical to achieving boundary accuracy.

Finally, to conclude our tests, we simulate the flow around a NACA 0012 airfoil. The airfoil has a unit chord and the background state is the same as above. Pressure contours for this case are provided in Fig. 19, which also gives the convergence of the x-velocity flux through the airfoil. Although the trailing-edge singularity limits the uniform-refinement rates for this problem, we see that, once again, BDPG achieves superior boundary accuracy.

11. Remaining challenges

Overall, the results shown in the above sections are encouraging, and verify the concept of using local optimal test functions to achieve global boundary accuracy. That said, before BDPG sees more widespread application, a few challenges remain.

The first of these, as mentioned, is the issue of trial space resolution near element boundaries. In the present work, we added order-p_B Lobatto functions to the trial space to ensure that the fluxes are well-represented. While this is an effective strategy for primal DG formulations, for an already-hybridized method it represents a relatively large computational expense, since these additional degrees-of-freedom are globally coupled. Thus, dealing with this issue in a more efficient way is an important next step. One promising option – as mentioned in Section 8.3 – is to perform a local optimization of the trial space in order to reduce the number of degrees of freedom on element boundaries.

Another important issue, which has not yet been emphasized, is the representation of the test functions themselves. For certain problems, the optimal test functions can exhibit nonsmooth behavior that makes their approximation difficult. Nonsmoothness of the test functions arises, for instance, for pure advection problems in two dimensions. In this case, the exact local adjoints (test functions) contain discontinuities within each element. Since polynomials cannot adequately resolve these discontinuities, the error between the discrete and exact optimal test functions can be large. This leads to errors in the elementwise fluxes, which propagate globally. A similar issue arises for high Reynolds number advection–diffusion cases, where – rather than discontinuities – steep boundary layers appear in the test functions.

These issues exist for many multiscale methods (including other DPG schemes, as discussed in e.g. [21,38]), and there are various means of addressing them. One option is the use of a “subgrid” within each element to resolve the relevant fine-scale features. However, for BDPG methods, an alternative option presents itself. For most cases – as mentioned in Section 6.2 – it is only necessary to resolve the test functions on the *boundaries* of each element. Thus, if test function discontinuities or boundary layers exist *inside* a given element, these features do not actually need to be resolved. Therefore,

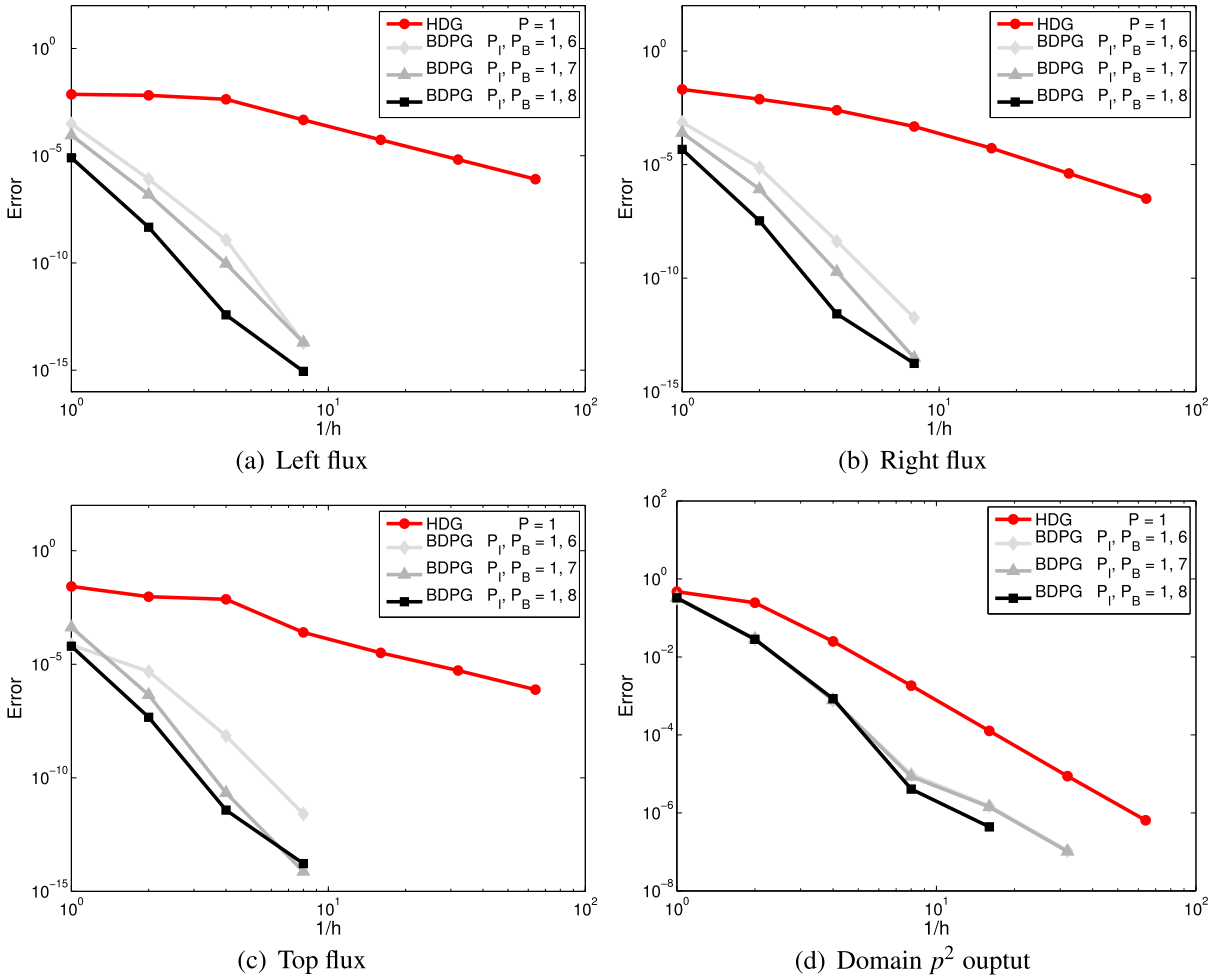


Fig. 15. Two-dimensional linearized Euler: Output convergence for HDG and BDPG runs. The flux outputs represent the sum of all state components of the flux vector. Higher accuracy is obtained as the amount of BDPG boundary enrichment increases. BDPG also achieves accuracy in the interior p^2 output.

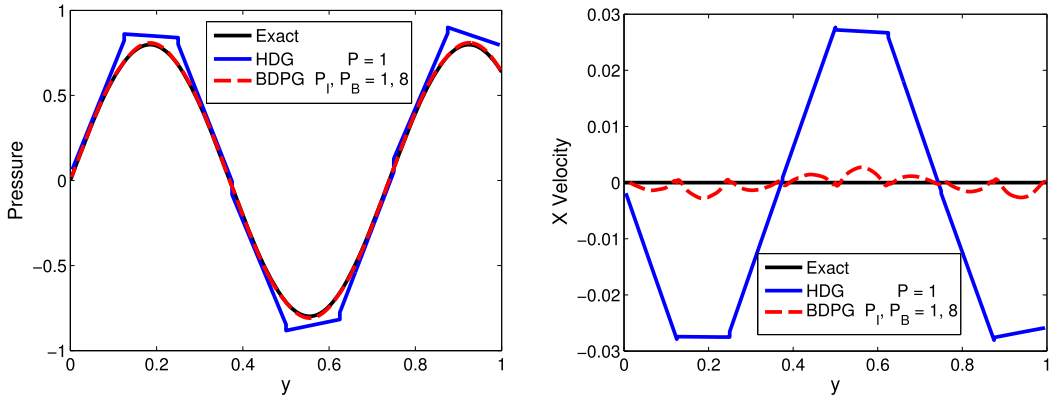
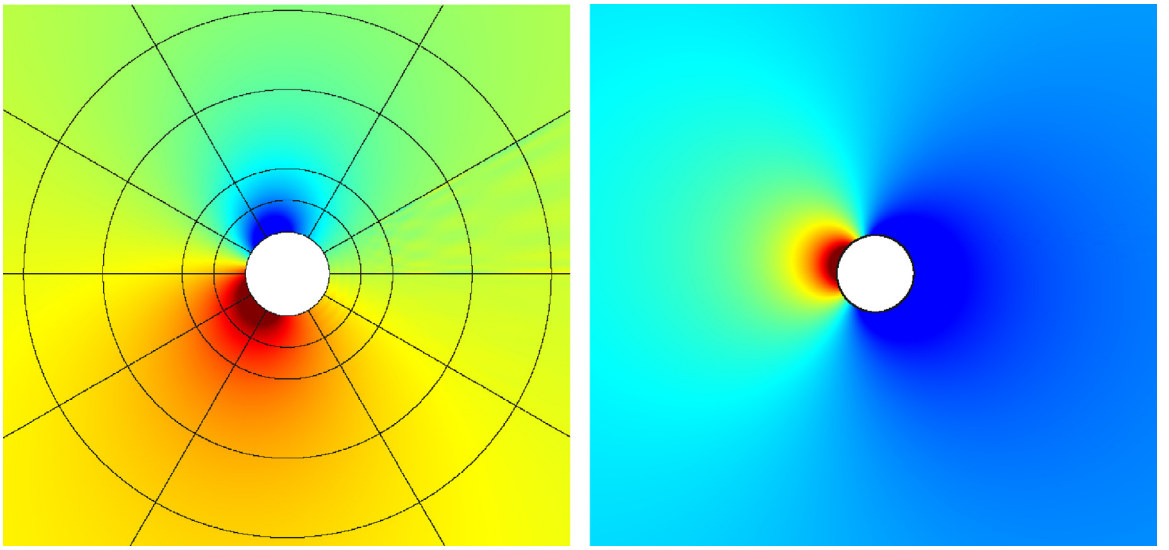


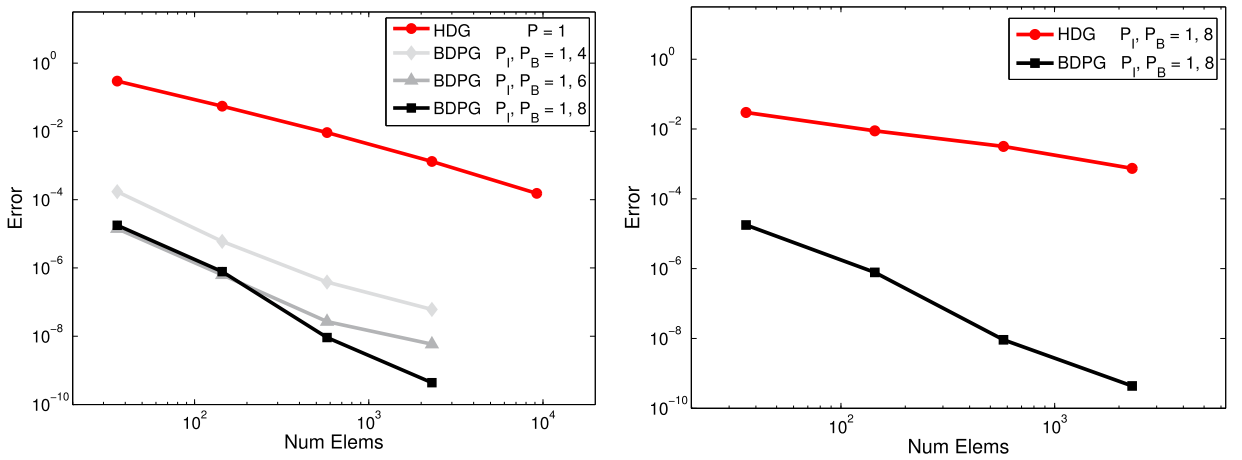
Fig. 16. Two-dimensional linearized Euler: Solution profiles along the right boundary of the domain. BDPG with boundary enrichment is again more accurate than HDG with the same interior basis.

when computing the optimal test functions, rather than using a standard DG or HDG method, we could instead attempt to tailor the discretization to focus solely on obtaining boundary accuracy in the test functions. Indeed, since achieving boundary accuracy has been the primary goal of this work, it may be possible to apply some of the present ideas to the test function problem itself.



(a) Pressure contours (b) y-velocity contours

Fig. 17. Two-dimensional cylinder: (a) Pressure and (b) y-velocity contours from a high-order HDG solution.

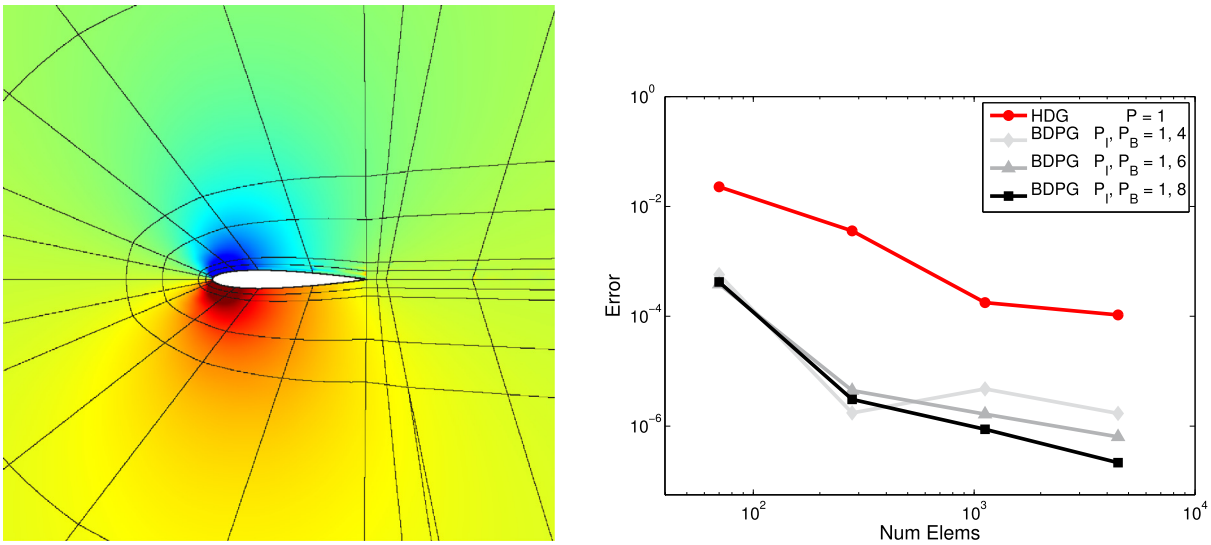


(a) Pressure flux, BDPG vs. $p = 1$ HDG (b) Pressure flux, BDPG vs. boundary-enriched HDG

Fig. 18. Two-dimensional cylinder: (a) Pressure flux convergence for BDPG and $p = 1$ HDG. (b) Pressure flux convergence where the same $p_B = 8$ trial space is used for both BDPG and HDG. Since the only difference is the test space, the results show that the optimal test functions of BDPG are effective in reducing boundary errors.

12. Conclusion

In this work, we presented a strategy for optimizing the test space of both primal and hybrid DG methods. The theory applies to linear PDEs and can be extended to nonlinear equations. We have shown that if the primary goal is to achieve boundary accuracy, the optimal test functions can be localized and computed independently on each element in the mesh. These test functions satisfy local adjoint equations and ensure that a proper upwinding of information occurs within each element. As shown, if the test functions and fluxes are well-represented, exact boundary fluxes are obtained. The resolution of both test functions and fluxes are critical issues, and while we have addressed certain aspects of these issues, additional challenges remain. Extension of the theory to nonlinear problems is another important step, and one that we explore in a future work.



(a) Pressure contours (b) Airfoil x-velocity flux

Fig. 19. Two-dimensional airfoil: (a) Pressure contours from a high-order BDPG solution. (b) x-velocity flux convergence for both BDPG and HDG. While the convergence rates are limited by the trailing-edge singularity, BDPG still provides a benefit over HDG.

Acknowledgements

The authors acknowledge support given by the Department of Energy under grant DE-FG02-13ER26146/DE-SC0010341, the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program, and the Air Force Office of Scientific Research under grant FA9550-10-C-0040.

Appendix A. Localization of the test space (one-dimensional advection–reaction)

Our primary goal in this work is to achieve accuracy in the fluxes through the domain boundaries. For the advection–reaction problem in Section 5, this means that we would like the flux on the right boundary,

$$J = a u_h(x_R)\,, \tag{A.1}$$

to be accurate. Ideally we would like it to have zero error.

From *a posteriori* error estimation, the error in a certain output (including the above J) can be represented as the inner product of a corresponding adjoint solution and the residual of the governing PDE [28,39,29]. For our one-dimensional advection–reaction problem, it is straightforward to show that the adjoint solution v corresponding to J satisfies the following equation:

$$L^* v = -a\frac{\partial v}{\partial x} + cv = 0 \qquad v(x_R) = 1\,. \tag{A.2}$$

This is a global differential equation, which can be solved analytically to obtain

$$v(x) = \underbrace{e^{-cx_R/a}}_{\text{const.}}\, e^{cx/a}\,. \tag{A.3}$$

The output error $\delta J = J(u_h) - J(u)$ can then be written as a product of this v and the residual of the PDE:

$$\delta J = \int_\Omega v\, r(u_h)\, dx = b(u_h, v) - l(v)\,. \tag{A.4}$$

From this expression, it is clear that if the adjoint v happens to lie in the test space of our finite element method, then the error in the flux J will be zero. This is because if v were in the test space, one of our finite element equations would be

$$b(u_h, v) = l(v) \quad \Longrightarrow \quad b(u_h, v) - l(v) = 0 \quad \Longrightarrow \quad \delta J = 0\,. \tag{A.5}$$

Our goal in this appendix is to show that, when using the local optimal test functions defined in Section 5, this v *is* in fact contained in the test space, and therefore the local test space *can* in fact deliver zero error in the domain-boundary flux.

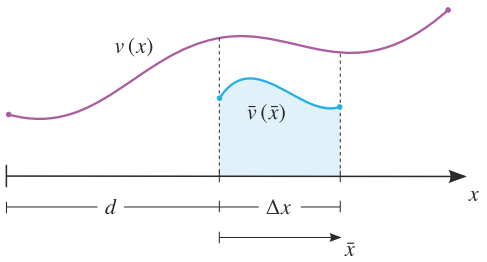


Fig. 20. Local and global coordinate systems and test functions. Note that a bar designates local quantities.

As described in Section 5, to compute the local optimal test functions, we solve elementwise adjoint problems for the following outputs on each element K:

$$J_i = \int_K \phi_i u \, dx + w_R \phi_i u \Big|_{\partial K_R} . \tag{A.6}$$

Now, for a trial basis ϕ_i that is nonzero on the right boundary, taking w_R large will make the boundary term in J_i dominate the interior term. Therefore, in the limit of large w_R, the interior term can be neglected and the output effectively becomes

$$J_i = w_R \phi_i u \Big|_{\partial K_R} . \tag{A.7}$$

This is just a constant multiple of the flux through the downwind boundary of the element. Since the constant makes no difference to the final test space, for purposes of analysis we can treat the above J_i as a pure (local) flux output. This means that, on any given element, one of the local optimal test functions (denoted by $\bar{v}(\bar{x})$) satisfies the following adjoint equation:

$$-a\frac{\partial \bar{v}}{\partial \bar{x}} + c\bar{v} = 0 \qquad \bar{v}(\Delta x) = 1 . \tag{A.8}$$

This equation is analogous to that for a global flux output (i.e. Eq. (A.2)), but is defined over an individual element rather than the entire domain. Here, $\bar{x}$ is a local coordinate associated with the element in question (see Fig. 20 for a definition of the relevant quantities). Solving this equation for the test function $\bar{v}$ gives

$$\bar{v}(\bar{x}) = \underbrace{e^{-c\Delta x/a}}_{\text{const.}} e^{c\bar{x}/a} . \tag{A.9}$$

Finally, writing $\bar{x}$ in terms of x using the transformation $\bar{x} = x - d$ results in

$$\bar{v}(x) = \underbrace{e^{-c\Delta x/a} e^{-cd/a}}_{\text{const.}} e^{cx/a} . \tag{A.10}$$

Comparing this local test function $\bar{v}$ to the global adjoint v in Eq. (A.3), we see these functions are indeed just constant multiples of each other. Thus, regardless of where a given element is located (i.e. regardless of d) and regardless of the size of the element (i.e. regardless of Δx), one of the local test functions on each element satisfies

$$\bar{v}(x) = C\, v(x) \tag{A.11}$$

for some constant C. Since by definition the test space includes all constant multiples of $\bar{v}$, this means that the global adjoint v is in fact contained in the test space. From our earlier discussion, this then implies that the error in the domain-boundary flux J is zero. Thus, the local optimal test space is in fact globally optimal with respect to the domain-boundary flux. While we have focused on an advection–reaction problem here, similar logic holds for more general problems (such as advection–diffusion).

References

[1] A. Brooks, T. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1982) 199–259.
[2] L.P. Franca, S.L. Frey, T.J. Hughes, Stabilized finite element methods: I. Application to the advective–diffusive model, Comput. Methods Appl. Mech. Eng. 95 (2) (1992) 253–276.
[3] T. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Eng. 127 (1995) 387–401.
[4] F. Brezzi, L. Franca, A. Russo, Further considerations on residual-free bubbles for advective–diffusive equations, Comput. Methods Appl. Mech. Eng. 166 (1998) 25–33.
[5] F. Brezzi, B. Cockburn, L.D. Marini, E. Süli, Stabilization mechanisms in discontinuous Galerkin finite element methods, Comput. Methods Appl. Mech. Eng. 195 (25) (2006) 3293–3310.

[6] F. Bassi, S. Rebay, A high-order discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.
[7] B. Cockburn, G.E. Karniadakis, C.-W. Shu, The Development of Discontinuous Galerkin Methods, Springer, 2000.
[8] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2002) 1749–1779.
[9] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations, J. Comput. Phys. 227 (2008) 9670–9685.
[10] H. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA Paper 2007-4079, 2007.
[11] P.E. Vincent, P. Castonguay, A. Jameson, A new class of high-order energy stable flux reconstruction schemes, J. Sci. Comput. 47 (1) (2011) 50–72.
[12] H. Gao, Z. Wang, A conservative correction procedure via reconstruction formulation with the chain-rule divergence evaluation, J. Comput. Phys. 232 (1) (2013) 7–13.
[13] J. Schütz, G. May, A hybrid mixed method for the compressible Navier–Stokes equations, J. Comput. Phys. 240 (2013) 58–75.
[14] J. Barrett, K.W. Morton, Approximate symmetrization and Petrov–Galerkin methods for diffusion–convection problems, Comput. Methods Appl. Mech. Eng. 45 (1) (1984) 97–122.
[15] L. Demkowicz, J. Oden, An adaptive characteristic Petrov–Galerkin finite element method for convection-dominated linear and nonlinear parabolic problems in one space variable, J. Comput. Phys. 67 (1986) 188–213.
[16] D. Givoli, Non-local and semi-local optimal weighting functions for symmetric problems involving a small parameter, Int. J. Numer. Methods Eng. 26 (1988) 1281–1298.
[17] M. Celia, T. Russell, I. Herrera, R. Ewing, An Eulerian–Lagrangian localized adjoint method for the advection–diffusion equation, Adv. Water Resour. 13 (1990) 187–206.
[18] I. Herrera, Trefftz method: a general theory, Numer. Methods Partial Differ. Equ. 16 (2000) 561–580.
[19] P. Barbone, I. Harari, Nearly H^1-optimal finite element methods, Comput. Methods Appl. Mech. Eng. 190 (2000) 5679–5690.
[20] T. Hughes, G. Sangalli, Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods, SIAM J. Numer. Anal. 45 (2) (2007) 539–557.
[21] L. Demkowicz, J. Gopalakrishnan, A class of discontinuous Petrov–Galerkin methods. Part I: the transport equation, Comput. Methods Appl. Mech. Eng. 199 (2010) 1558–1572.
[22] L. Demkowicz, J. Gopalakrishnan, A class of discontinuous Petrov–Galerkin methods. II. Optimal test functions, Numer. Methods Partial Differ. Equ. 27 (2011) 70–105.
[23] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, V.M. Calo, A class of discontinuous Petrov–Galerkin methods. Part IV: the optimal test norm and time-harmonic wave propagation in 1D, J. Comput. Phys. 230 (2011) 2406–2432.
[24] J. Chan, L. Demkowicz, R. Moser, N. Roberts, A new discontinuous Petrov–Galerkin method with optimal test functions. Part V: solution of 1d Burgers and Navier–Stokes equations, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, 2010.
[25] D. Moro, N. Nguyen, J. Peraire, A hybridized discontinuous Petrov–Galerkin scheme for scalar conservation laws, Int. J. Numer. Methods Eng. 91 (9) (2012) 950–970.
[26] D. Estep, M. Holst, M. Larson, Generalized Green's functions and the effective domain of influence, SIAM J. Sci. Comput. 26 (4) (2005) 1314–1339.
[27] T.J. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method – a paradigm for computational mechanics, Comput. Methods Appl. Mech. Eng. 166 (1) (1998) 3–24.
[28] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, in: A. Iserles (Ed.), Acta Numerica, Cambridge University Press, 2001, pp. 1–102.
[29] K.J. Fidkowski, D.L. Darmofal, Review of output-based error estimation and mesh adaptation in computational fluid dynamics, AIAA J. 49 (4) (2011) 673–694.
[30] A. Jameson, Aerodynamic design via control theory, J. Sci. Comput. 3 (1988) 233–260.
[31] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[32] J. Lu, An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.
[33] R. Hartmann, Adjoint consistency analysis of discontinuous Galerkin discretizations, SIAM J. Numer. Anal. 45 (6) (2007) 2671–2696.
[34] P. Solín, K. Segeth, I.D. Zel, Higher-Order Finite Element Methods, Chapman and Hall, 2003.
[35] J. Peraire, N.C. Nguyen, B. Cockburn, An embedded discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations, AIAA Paper 2011-3228, 2011.
[36] N. Nguyen, J. Peraire, B. Cockburn, Hybridizable discontinuous Galerkin methods, in: Spectral and High Order Methods for Partial Differential Equations, Springer, 2011, pp. 63–84.
[37] K. Fidkowski, High-order output-based adaptive methods for steady and unsteady aerodynamics, in: 37th Advanced VKI CFD Lecture Series, von Karman Institute, 2013.
[38] J. Chan, N. Heuer, T. Bui-Thanh, L. Demkowicz, A robust DPG method for convection-dominated diffusion problems II: adjoint boundary conditions and mesh-dependent test norms, Comput. Math. Appl. 67 (4) (2014) 771–795.
[39] R. Hartmann, P. Houston, Error estimation and adaptive mesh refinement for aerodynamic flows, in: H. Deconinck (Ed.), 36th CFD/ADIGMA Course on hp-Adaptive and hp-Multigrid Methods: VKI Lecture Series 2010-01, Oct. 26–30, 2009, von Karman Institute for Fluid Dynamics, 2010.