AIAA SciTech
5-9 January 2015, Kissimmee, Florida
53rd AIAA Aerospace Sciences Meeting

AIAA 2015-1956

A Hybrid Petrov-Galerkin Method for Optimal Output Prediction

Steven M. Kast, [*] Johann P.S. Dahm, [†] and Krzysztof J. Fidkowski [‡]
University of Michigan, Ann Arbor, MI, 48109, United States

We present a new Boundary Discontinuous Petrov-Galerkin (BDPG) method for Computational Fluid Dynamics (CFD) simulations. The method represents a modification of the standard Hybrid Discontinuous Galerkin (HDG) scheme, and uses locally-computed optimal test functions to achieve enhanced accuracy along the domain boundaries. This leads to improved accuracy in relevant boundary outputs such as lift and drag. Results demonstrate that, for linear problems in both one and two dimensions, exact boundary outputs are obtained if the test functions and fluxes are well-represented. Furthermore, for nonlinear problems such as the Navier-Stokes equations, the method can achieve $2p + 2$ output convergence rates, which represents an improvement over the $2p + 1$ rates of standard HDG.

I. Introduction

Recently, finite element methods have been gaining popularity in the aerospace community as an alternative to finite volume methods. In addition to high-order accuracy, finite element methods provide a rigorous setting for output-based error estimation and mesh adaptation. The ability to compute accurate outputs is an attractive feature, since predicting quantities such as lift or drag is often the primary goal of a CFD simulation.

The typical strategy for achieving output accuracy is to use a standard (e.g. discontinuous) Galerkin method in combination with output-based mesh adaptation.[1–3] In this case, the numerical method itself is a "general-purpose" scheme, while the mesh bears the burden of providing accuracy in outputs of interest. However, in the current work, rather than optimizing the mesh, we present an alternative – and largely overlooked – strategy: that of optimizing the scheme itself.

In particular, we show that the test space of a finite element method can be optimized to obtain accuracy in certain outputs of interest. This results in a Petrov-Galerkin method whose test functions differ from the trial functions. In this work, we provide a general framework for deriving and computing the optimal test space, and present a "boundary" discontinuous Petrov-Galerkin (BDPG) method that achieves accuracy specifically in *boundary* outputs. These outputs are often the most relevant from an engineering standpoint.

The concept behind optimal test functions is relatively straightforward. The idea is to choose the test functions such that the finite element weighted residual becomes the derivative of a certain error norm of interest. Since, by definition, the method forces the residual to be zero, it therefore forces the derivative of the error to be zero as well. This then implies that the error is minimized, and that the method is optimal in the desired norm.

[*] Ph.D. Candidate, Department of Aerospace Engineering, Student Member AIAA

[†] Ph.D. Candidate, Department of Aerospace Engineering, Student Member AIAA

[‡] Associate Professor, Department of Aerospace Engineering, Senior Member AIAA

As we show, the optimal test functions turn out to be adjoint solutions for certain "projection" outputs related to the chosen error norm. For general error norms, computing the optimal test functions on a given element requires solving a global differential equation. From a practical standpoint, this is infeasible, and appears to limit the method to one of purely theoretical interest.

However, our desire for accurate boundary outputs actually saves us, since the error norms that emphasize boundary accuracy turn out to be exactly those that are easily *localizable.* As we show, a key idea is to choose the norm such that errors in the inter-element fluxes are minimized, as this reduces global error propagation and (among other things) leads to accuracy in boundary outputs. The test functions themselves can be computed locally on each element, and represent adjoint solutions for the local fluxes. For linear problems, if both test functions and inter-element fluxes are well-represented, exact boundary outputs are obtained.

The work is inspired by the discontinuous Petrov-Galerkin (DPG) methods introduced in,[4–7] but differs in several ways. Our emphasis on boundary outputs is unique, and unlike previous DPG methods, we do not require the use of an "optimal test norm."[6] Furthermore, the ideas in this work apply to both primal and hybrid formulations, as opposed to the more expensive "ultra-weak" formulation espoused in.[5] Finally, we note that the optimal test function theory has close ties to *a posteriori* error estimation,[1,2,8] multiscale methods,[9–12] and other stabilized schemes.[13–19]

In this work, we implement the optimal test functions within a hybrid framework (similar to[20]), and demonstrate the resulting hybridized BDPG method on a series of problems in both one and two dimensions. The results show that BDPG can achieve significant reductions in boundary error compared to standard Galerkin methods.

II. Optimal Test Function Theory

In this section, we present the theory of optimal test functions. We begin with a 1D example before progressing to multidimensional and nonlinear systems of equations. While the theory itself is only optimal for linear problems, it can be extended readily to nonlinear problems once they have been locally linearized.

II.A. A 1D Example

To make things concrete, assume we have a linear partial differential equation (PDE) of the following form:

$$\begin{aligned} a\frac{\partial u}{\partial x} &= f(x) \qquad && x \in \Omega \\ u\big|_{x_L} &= u_L \qquad && x \in \partial\Omega \end{aligned} \tag{1}$$

This is just linear advection with a source term, $f(x)$. Here, we assume $a > 0$, so that the Dirichlet boundary condition u_L on the left is well-posed.

To generalize the notation, we see that the above PDE can be written as

$$\begin{aligned} Lu &= f \qquad && x \in \Omega \\ u\big|_{x_L} &= u_L \qquad && x \in \partial\Omega \end{aligned} \tag{2}$$

where the differential operator L is given by $L \equiv a\frac{\partial()}{\partial x}$. Furthermore, we define the residual as $r(u) \equiv Lu - f$. Since this residual is zero for the exact solution u, it is clear that we can write:

$$\int_\Omega v\,(Lu - f)\,dx = 0 \qquad \forall v \in V\,, \tag{3}$$

where v is any test function in some continuous test space V.

Now, to compute an approximate solution to the above PDE, a finite element method attempts to mimic the weighted-residual statement in Eqn. 3. In other words, it seeks an approximate solution $u_h \in U_h$ that satisfies

$$\int_{\Omega} v_h\,(Lu_h - f)\,dx = 0 \qquad \forall v_h \in V_h\,, \tag{4}$$

where v_h is any test function in some discrete test space V_h. Once a basis is chosen for the approximation space U_h, which is typically taken to be a polynomial space of order p, our discrete u_h can be represented as

$$u_h = \sum_{i=1}^{N} U_i\,\phi_i(x)\,, \tag{5}$$

where the ϕ_i form a basis of U_h, and the U_i are the unknown solution coefficients.

Then the remaining question is: what should our test space V_h be? A standard Galerkin method would choose $V_h = U_h$, 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?* In general, we can think of our discrete solution u_h as a type of "curve fit" to the exact solution u. And when performing a curve fit, we know that in order to get a best-fit approximation, we must first define the error norm that we want the best fit *in.* In the same way, when approximating the solution to a PDE, we need to say how we would like our discrete u_h to "fit" the exact u.

In practice, we would often like the discrete u_h to give a good L^2 approximation to the true solution on both the interior and boundaries of the domain. Thus, for our current 1D problem, the error norm that we desire the best approximation in might look something like:

$$||e||^2 = \int_{\Omega} (u_h - u)^2\,dx \,+\, w_R\,(u_h - u)^2\Big|_{x_R}\,. \tag{6}$$

Here, w_R is a weight that determines how much emphasis to place on the solution near the right boundary. If we want u_h to match u closely at the right boundary, we would take w_R to be large, whereas if we are interested in only interior accuracy, we would choose $w_R = 0$. Note that there is no need to ask for accuracy on the left boundary, since the solution there is already known from the Dirichlet condition.

In order to find the coefficients U_i that minimize this norm, we can 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 this case, we know that our critical point is in fact a minimum, since $||e||^2$ is a positive (concave up) quadratic.

Given that $u_h = \sum_{i=1}^{N} U_i\,\phi_i$, we see that differentiating Eqn. 6 with respect to U_i gives:

$$\frac{\partial ||e||^2}{\partial U_i} = 0 = \int_{\Omega} 2\,(u_h - u)\,\phi_i\,dx \,+\, 2\,w_R\,(u_h - u)\,\phi_i\Big|_{x_R}\,. \tag{7}$$

Thus, if the approximation u_h is to provide the minimum error in our chosen norm ($||e||^2$), it must satisfy the following equation(s)

$$\int_{\Omega} \phi_i\,(u_h - u)\,dx \,+\, w_R\,\phi_i\,(u_h - u)\Big|_{x_R} = 0 \tag{8}$$

for $i = 1$ to N.

Now, we claim that with a certain ("optimal") choice of test functions, we can ensure that the above equation is satisfied by our finite element solution. To see how this is done, note that Eqn. 8 involves some quantity that is equal to zero. Next, note that our weighted residual statement (Eqn. 4) also involves a quantity that is equal to zero. Then the idea is this: if we can make our weighted residual statement *look* like the error minimization statement, our finite element solution u_h will necessarily minimize that error. In other words, our weighted residual statement will simply *become* a direct statement of error minimization.

We now show how this is done. From Eqn 4, we have that

$$\int_\Omega v_h(Lu_h - f)\,dx = 0 \qquad \forall v_h \in V_h\,, \tag{9}$$

and since the exact residual is zero pointwise, we also have

$$\int_\Omega v_h(Lu - f)\,dx = 0 \qquad \forall v_h \in V_h\,. \tag{10}$$

Now, since both of these quantities (Eqns. 9 and 10) are zero, we can equate them to get

$$\int_\Omega v_h(Lu_h - f)\,dx = \int_\Omega v_h(Lu - f)\,dx \qquad \forall v_h \in V_h\,. \tag{11}$$

Bringing everything to the left-hand side (and canceling the f terms) then gives:

$$\int_\Omega v_h(Lu_h - Lu)\,dx = 0 \qquad \forall v_h \in V_h\,. \tag{12}$$

Since L is a linear operator, we can rewrite this as:

$$\int_\Omega v_h\,L(u_h - u)\,dx = 0 \qquad \forall v_h \in V_h\,. \tag{13}$$

Notice that we now have an error term $(u_h - u)$ in the equation. This is desirable, since Eqn. 8 also contains error terms, and we are attempting to make the weighted residual look like this equation. To introduce a boundary term, we can integrate the above equation by parts. To make this more concrete, let us first substitute $L = a\frac{\partial()}{\partial x}$ back in for the differential operator. Then Eqn. 13 rewritten is just

$$\int_\Omega v_h\,a\frac{\partial(u_h - u)}{\partial x}\,dx = 0 \qquad \forall v_h \in V_h\,. \tag{14}$$

Now integrating by parts gives

$$\int_\Omega \underbrace{\left[-a\frac{\partial v_h}{\partial x}\right]}_{L^* v_h}(u_h - u)\,dx \;+\; v_h a(u_h - u)\Big|_{x_L}^{x_R} = 0 \qquad \forall v_h \in V_h\,. \tag{15}$$

Since we are specifying a Dirichlet condition on the left boundary, the error $(u_h - u)$ is zero there, so the left boundary term above vanishes. Then if we define the operator that emerges after integration by parts as $L^* \equiv -a\frac{\partial()}{\partial x}$, we can rewrite the above equation as just:

$$\int_\Omega L^* v_h\,(u_h - u)\,dx \;+\; v_h a\,(u_h - u)\Big|_{x_R} = 0 \qquad \forall v_h \in V_h\,. \tag{16}$$

Finally, regardless of what the test space is, 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:

$$\int_\Omega L^* v_i \left(u_h - u\right) dx \,+\, v_i a \left(u_h - u\right)\Big|_{x_R} = 0 \qquad \forall v_i \in V_h \,. \tag{17}$$

Recall that we are trying to make this look like Eqn. 8, which is:

$$\int_\Omega \phi_i \left(u_h - u\right) dx \,+\; w_R\, \phi_i \left(u_h - u\right)\Big|_{x_R} = 0 \qquad \forall \phi_i \in U_h$$

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

$$\begin{aligned} L^* v_i &= \phi_i & \qquad & x \in \Omega \\ a\, v_i\Big|_{x_R} &= w_R\, \phi_i\Big|_{x_R} & & x \in \partial\Omega \,. \end{aligned} \tag{18}$$

for $i = 1$ to N.

So we see that if we want the discrete solution u_h to minimize the error given by Eqn. 6, we need the test functions v_i to satisfy the above differential equation(s) and boundary condition(s). The test functions that satisfy the above equations are the *optimal test functions*, in the sense that they give the best possible solution u_h in the desired error norm.

Furthermore, recall that through our choice of the weight w_R, we have direct control over *where* we obtain solution accuracy. Choosing w_R large will give an accurate solution on the right boundary, while taking $w_R = 0$ will provide a least-squares fit of u over the domain interior. For the BDPG method presented in the current work, we are interested in obtaining accurate boundary outputs, which in this case corresponds to choosing w_R large.

Note that we have posed the above derivations in a continuous setting, which means the differential equation satisfied by the v_i is a global equation over the whole domain. In practice, this would be too expensive to solve, so we will need to develop a method of localizing it. In addition, the above equations assume that the v_i are represented in an infinite-dimensional space. This is also impossible to do in practice; instead, we can approximate the v_i with (e.g.) high-order polynomials. Note that if we were to compute the v_i in the same space as the trial functions (i.e. in an order-p space) then the method would reduce exactly to a standard Galerkin method. So in the end, if we wish to do better than a Galerkin method, we must use test functions that are of higher order than the trial space.

In this section, we have derived the optimal test functions. If we wanted, we could stop here. However, it is worthwhile at this point to mention an interesting fact about the optimal test functions – namely, that they are *adjoint* solutions.

II.B. Optimal Test Functions as Adjoints

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

$$(Lu, v) = (u, L^* v) \qquad \forall u \in U,\;\; \forall v \in V, \tag{19}$$

where U and V are function spaces over which the above inner product is defined. In practical cases, the left-hand side of this relationship is just a differential equation (Lu) weighted by a test function (v). Thus, in order to find what L^* is, we can simply integrate the left-hand side by parts. This will remove all derivatives from u and place them on v. The resulting operator acting on v is then defined to be L^*. So this adjoint operator L^* could just as well be called the "integration by parts" operator.

The important point is that the adjoint operator is exactly the operator in the optimal test function equations (Eqn. 18). If we look back at the derivation (specifically, Eqn. 15), we see that the L^* there was indeed defined to be the operator obtained after integrating by parts.

Now, from optimization[21] and *a posteriori* error estimation,[1,2] we know that adjoint equations relate 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*?

It turns out that the output associated with the optimal test functions is:

$$J_i = \int_\Omega \phi_i u \, dx \, + \, w_R \phi_i u \Big|_{x_R} \tag{20}$$

This is straightforward to verify, though we omit the proof for brevity. From the above equation, we see that a given output J_i represents the *projection* of the exact solution u against the i-th trial basis function.

From *a posteriori* error estimation, it is known that the adjoint-weighted residual for a certain output gives the error in that 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,

$$\int_\Omega v_i \underbrace{(Lu_h - f)}_{r(u_h)} \, dx = 0 = \delta J_i \qquad \forall v_i \in V_h \, , \tag{21}$$

we are directly enforcing that the error in each projection output, $\delta J_i \equiv J_i(u_h) - J_i(u)$, is zero. This means that our discrete solution u_h is a direct projection of the true solution u into the trial space, with respect to the desired ($||e||^2$) norm. This is the same conclusion we arrived at in the previous section, but we now understand it from a different angle.

Finally, to drive home 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 test functions v_i, which are adjoints for the J_i, gives us zero error in the J_i. But from the above definition of J_i, zero error in J_i implies

$$\delta J_i = J_i(u_h) - J_i(u) = 0 = \int_\Omega \phi_i (u_h - u) \, dx \, + \, w_R \phi_i (u_h - u) \Big|_{x_R} . \tag{22}$$

We see that this is identical to the statement that $\frac{\partial ||e||^2}{\partial U_i} = 0$, i.e. it is identical to Eqn. 8, and thus implies that $||e||^2$ is minimized.

II.C. Extension to Multi-Dimensional Systems

So far, we have derived the optimal test functions only in the context of a one-dimensional scalar 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 assume that the domain Ω consists of a single element.

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

$$\nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \vec{\mathbf{q}}) = \mathbf{0}, \tag{23}$$

$$\vec{\mathbf{q}} - \nabla \mathbf{u} = \vec{\mathbf{0}} \, , \tag{24}$$

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.)

To obtain the weak form of the above problem, we weight Eqns. 23 and 24 by test functions $\mathbf{v}$ and $\vec{\boldsymbol{\tau}}$, respectively, giving a total weighted residual (upon summation) of

$$R \equiv \int_\Omega \vec{\boldsymbol{\tau}}^T \cdot (\vec{\mathbf{q}} - \nabla \mathbf{u})\; d\Omega + \int_\Omega \mathbf{v}^T (\nabla \cdot \vec{\mathbf{F}})\, d\Omega \;=\; 0\,. \tag{25}$$

Note that this residual is just a scalar value. We next integrate both terms in Eqn. 25 by parts, giving

$$\int_\Omega \vec{\boldsymbol{\tau}}^T \cdot \vec{\mathbf{q}}\, d\Omega + \int_\Omega \nabla \cdot \vec{\boldsymbol{\tau}}^T\, \mathbf{u}\, d\Omega - \int_\Omega (\nabla \mathbf{v})^T \cdot \vec{\mathbf{F}}\, d\Omega \;+ \int_{\partial\Omega} \mathbf{v}^T (\vec{\mathbf{F}} \cdot \vec{n})\, ds - \int_{\partial\Omega} (\vec{\boldsymbol{\tau}} \cdot \vec{n})^T\, \mathbf{u}\, ds \;= 0. \tag{26}$$

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:

$$\underbrace{\int_\Omega \vec{\boldsymbol{\tau}}^T \cdot \vec{\mathbf{q}}\, d\Omega + \int_\Omega \nabla \cdot \vec{\boldsymbol{\tau}}^T\, \mathbf{u}\, d\Omega - \int_\Omega (\nabla \mathbf{v})^T \cdot \vec{\mathbf{F}}\, d\Omega \;+ \int_{\partial\Omega} \mathbf{v}^T\, \hat{\mathbf{F}}\; ds}_{b\,(\mathbf{u}, \vec{\mathbf{q}}, \mathbf{v}, \vec{\boldsymbol{\tau}})} = \underbrace{\int_{\partial\Omega} (\vec{\boldsymbol{\tau}} \cdot \vec{n})^T\, \mathbf{u}_B\, ds}_{l\,(\mathbf{v}, \vec{\boldsymbol{\tau}})}\,. \tag{27}$$

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

Next, as in one dimension, 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

$$\vec{\mathbf{F}} = \frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\, \mathbf{u} + \frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\, \mathbf{q}_j\,, \tag{28}$$

where summation over the spatial dimension j is implied. Note that for nonlinear problems a similar expression would hold for the Fréchet linearization of the flux. We now substitute this expression (Eqn. 28) into Eqn. 27 and transpose the first three terms, giving

$$b \;=\; \int_\Omega \vec{\mathbf{q}}^T \cdot \vec{\boldsymbol{\tau}}\, d\Omega + \int_\Omega \mathbf{u}^T \nabla \cdot \vec{\boldsymbol{\tau}}\, d\Omega - \int_\Omega \left(\mathbf{u}^T \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\right]^T + \mathbf{q}_j^T \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\right]^T \right) \cdot (\nabla \mathbf{v})\; d\Omega \;+ \int_{\partial\Omega} \mathbf{v}^T\, \hat{\mathbf{F}}\; ds\,. \tag{29}$$

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

$$b \;=\; \int_\Omega \mathbf{q}_j^T \underbrace{\left(\boldsymbol{\tau}_j - \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{q}_j}\right]^T \cdot \nabla \mathbf{v} \right)}_{L^*_{q,j}(\vec{\boldsymbol{\tau}}, \mathbf{v})} d\Omega + \int_\Omega \mathbf{u}^T \underbrace{\left(\nabla \cdot \vec{\boldsymbol{\tau}} - \left[\frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{u}}\right]^T \cdot \nabla \mathbf{v} \right)}_{L^*_u(\vec{\boldsymbol{\tau}}, \mathbf{v})} d\Omega + \int_{\partial\Omega} \mathbf{v}^T\, \hat{\mathbf{F}}\; ds\,. \tag{30}$$

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

$$\tilde{\mathbf{u}} \equiv \begin{bmatrix} \mathbf{q}_j \\ \mathbf{u} \end{bmatrix} \qquad \text{and} \qquad \tilde{\mathbf{v}} \equiv \begin{bmatrix} \boldsymbol{\tau}_j \\ \mathbf{v} \end{bmatrix}\,, \tag{31}$$

then the weak form of the problem reduces to

$$\int_\Omega \mathbf{q}_j^T\, L^*_{q,j}(\tilde{\mathbf{v}})\, d\Omega \,+ \int_\Omega \mathbf{u}^T\, L^*_u(\tilde{\mathbf{v}})\, d\Omega \,+ \int_{\partial\Omega} \mathbf{v}^T\, \hat{\mathbf{F}}\; ds \;=\; l\,(\tilde{\mathbf{v}})\;. \tag{32}$$

To approximate the above equation, a finite element method chooses a set of discrete states and test functions (denoted by a subscript h), resulting in

$$\int_\Omega \mathbf{q}_{j,h}^T\, L^*_{q,j}(\tilde{\mathbf{v}}_h)\, d\Omega \,+ \int_\Omega \mathbf{u}_h^T\, L^*_u(\tilde{\mathbf{v}}_h)\, d\Omega \,+ \int_{\partial\Omega} \mathbf{v}_h^T\, \hat{\mathbf{F}}\; ds \;=\; l\,(\tilde{\mathbf{v}_h})\;. \tag{33}$$

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

$$u_{s,h} = \sum_{m=1}^{n_U} U_{s,m}\,\phi_{s,m}(\vec{x}) \quad \text{and} \quad q_{s,d,h} = \sum_{m=1}^{n_Q} Q_{s,d,m}\,\phi_{s,d,m}(\vec{x})\,. \tag{34}$$

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. Choosing a common test function $\tilde{\mathbf{v}}_i$ for Eqns. 32 and 33 and subtracting yields

$$\underbrace{\int_\Omega (\mathbf{q}_{j,h} - \mathbf{q}_j)^T\, L^*_{q,j}(\tilde{\mathbf{v}}_i)d\Omega + \int_\Omega (\mathbf{u}_h - \mathbf{u})^T\, L^*_u(\tilde{\mathbf{v}}_i)d\Omega + \int_{\partial\Omega} \mathbf{v}_i^T\left[\hat{\mathbf{F}}(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{\mathbf{F}}(\mathbf{u}, \vec{\mathbf{q}})\right] ds}_{b(\tilde{\mathbf{e}},\tilde{\mathbf{v}}_i)} = 0\,. \tag{35}$$

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. To that end, we propose minimizing the following norm:

$$||\tilde{\mathbf{e}}||^2 = \sum_{s=1}^{r}\left\{ \sum_{d=1}^{dim} \underbrace{\int_\Omega (q_{s,d,h} - q_{s,d})^2 d\Omega}_{\text{interior } q \text{ accuracy}} + \underbrace{\int_\Omega (u_{s,h} - u_s)^2 d\Omega}_{\text{interior } u \text{ accuracy}} + \underbrace{w_s \int_{\partial\Omega}\left[\hat{F}_s(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\right]^2 ds}_{\text{flux accuracy}} \right\} \tag{36}$$

where $\tilde{\mathbf{e}} \equiv \tilde{\mathbf{u}}_h - \tilde{\mathbf{u}}$ is a group variable representing the error in the state and gradients. The above norm emphasizes accuracy in the state, its gradients, and the boundary fluxes. Furthermore, we see that choosing the weights w_s to be large places particular emphasis on the flux accuracy, which, as in one dimension, is our primary aim.

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

$$\frac{1}{2}\,\frac{\partial ||\tilde{\mathbf{e}}||^2}{\partial U_{k,m}} = 0 = \int_\Omega (u_{k,h} - u_k)\,\phi_{k,m} d\Omega + \sum_{s=1}^{r} w_s \int_{\partial\Omega}\left[\hat{F}_s(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\right] \frac{\partial \hat{F}_s}{\partial u_{k,h}} \phi_{k,m} ds \tag{37}$$

and

$$\frac{1}{2}\,\frac{\partial ||\tilde{\mathbf{e}}||^2}{\partial Q_{k,d,m}} = 0 = \int_\Omega (q_{k,d,h} - q_{k,d})\phi_{k,d,m} d\Omega + \sum_{s=1}^{r} w_s \int_{\partial\Omega}\left[\hat{F}_s(\mathbf{u}_h, \vec{\mathbf{q}}_h) - \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\right] \frac{\partial \hat{F}_s}{\partial q_{k,d,h}} \phi_{k,d,m} ds. \tag{38}$$

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 Eqns. 37 and 38 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 $b(\tilde{\mathbf{e}}, \tilde{\mathbf{v}}_i)$ (Eqn. 35) to Eqn. 37, we see that to make these expressions identical the test functions must satisfy:

$$i = 1 \ldots N_U \quad \left\{ \begin{array}{llll} L^*_{q,j}(\tilde{\mathbf{v}}_i) \;=\; \mathbf{0} & & j = 1 \ldots dim & x \in \Omega \\[2ex] L^*_{u,s}(\tilde{\mathbf{v}}_i) \;=\; \phi_{k,m}\, \delta_{s,k} & & s = 1 \ldots r & x \in \Omega \\[2ex] v_{i,s} \;=\; w_s \dfrac{\partial \hat{F}_s}{\partial u_{k,h}}\, \phi_{k,m} & & s = 1 \ldots r & x \in \partial\Omega \end{array} \right. \tag{39}$$

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 $1 \leq k \leq r$ and $1 \leq m \leq n_U$, 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 Eqn. 38, we see that the remaining test functions should satisfy:

$$i = 1 \ldots N_Q \quad \left\{ \begin{array}{lll} L^*_{q,j,s}(\tilde{\mathbf{v}}_i) \;=\; \phi_{k,d,m}\, \delta_{j,d}\, \delta_{s,k} & j, s = 1 \ldots dim, r & x \in \Omega \\[2ex] L^*_u(\tilde{\mathbf{v}}_i) \;=\; \mathbf{0} & & x \in \Omega \\[2ex] v_{i,s} \;=\; w_s \dfrac{\partial \hat{F}_s}{\partial q_{k,d,h}}\, \phi_{k,d,m} & s = 1 \ldots r & x \in \partial\Omega \end{array} \right. \tag{40}$$

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 $1 \leq k \leq r$, $1 \leq d \leq dim$, $1 \leq m \leq n_Q$. 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 Eqns. 37 and 38 are satisfied, and hence that the error in Eqn. 36 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 Eqns. 37 and 38, we can write the effective outputs as

$$J^u_{k,m} \;=\; \int_\Omega u_k\, \phi_{k,m}\, d\Omega \;+\; \sum_{s=1}^{r} w_s \int_{\partial\Omega} \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\, \frac{\partial \hat{F}_s}{\partial u_{k,h}}\, \phi_{k,m}\, ds \tag{41}$$

and

$$J^q_{k,d,m} \;=\; \int_\Omega q_{k,d}\, \phi_{k,d,m}\, d\Omega \;+\; \sum_{s=1}^{r} w_s \int_{\partial\Omega} \hat{F}_s(\mathbf{u}, \vec{\mathbf{q}})\, \frac{\partial \hat{F}_s}{\partial q_{k,d,h}}\, \phi_{k,d,m}\, ds\,. \tag{42}$$

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.

II.D. Localization of Test Functions and Minimization of Flux Errors

II.D.1. Computation of Test Functions

Above, we derived the optimal test functions while assuming a single-element mesh. This means that the adjoint equations satisfied by the optimal test functions are *global* differential equations, and that, if we have a multi-element mesh, we would need to solve a global equation for the test functions on each element. This is not feasible in practice. Thus, we need to find a way to localize the computation of the test functions without giving up their accuracy.

Fortunately, if all we desire is accuracy in the global boundary fluxes, then localizing the test function computation is straightforward. We simply loop over each element in the mesh and "pretend" that it is the only element in the domain. For example, to compute the optimal test functions on an interior element, we just replace the $\hat{F}$ flux terms in Eqns. 41 and 42 with the local numerical flux. If necessary (i.e. for nonlinear problems), the states on neighboring elements act as local Dirichlet boundary conditions. On the other hand, if we are on an element with a boundary face, then we replace the $\hat{F}$ terms with the relevant analytical boundary flux. In this way, we solve local adjoint problems (defined over only a single element) to compute the optimal test functions, with the local outputs involving whichever flux is naturally defined on the element's boundaries. These local adjoint problems are well-posed[22, 23] so long as the numerical flux performs a proper upwinding of the data (as does, e.g., the Roe flux[24]).

II.D.2. Justification for Localization

We have claimed that it is valid to localize the test functions in this way, but we have not explained why. The first argument we can give is mathematical. It can be proven analytically that for 1D advection and advection-diffusion problems, the test space formed by the local adjoints as the weights $w_1, ..., w_{sr}$ are taken large *contains* the global adjoints corresponding to the domain-boundary fluxes. This implies that the local test spaces are in fact optimal for achieving accuracy in the boundary fluxes.

A more intuitive argument is as follows. By using local optimal test functions with large flux weights, we obtain accuracy in the outgoing flux on each *element*, rather than on the global boundary. However, the critical idea is that, if local flux accuracy is obtained on each element boundary, then this accuracy will propagate downstream, ultimately yielding global accuracy on the domain boundary. This idea holds for 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.

II.D.3. Flux Accuracy Issues

The question then becomes, *can* the local fluxes always be made accurate? There are two potential impediments to obtaining accuracy in the local fluxes: (i) nonsmoothness of the local adjoint (i.e. test function) problems, and (ii) the inability of the trial space to accurately represent the exact flux.

Nonsmoothness of the local adjoints arises, for instance, in 2D for pure advection problems. In this case, the local adjoints contain discontinuities within each element, and high-order polynomials cannot do an adequate job of representing them. Thus, the error between the discrete test functions and optimal test functions is large, leading to errors in the elementwise fluxes. To a lesser extent, advection-diffusion problems at high Reynolds number encounter a similar issue – steep boundary layers develop in the test functions, which again make them difficult to approximate discretely. While these boundary layers reduce in severity upon mesh refinement, finding an efficient way to represent them is an important topic for future investigation.

The second issue – that of inadequate flux representation – is also critical, though this can be dealt with more directly. The idea here is that even if our discrete test functions are perfectly accurate, if the *trial* space cannot represent the true flux on element boundaries, then relatively large pointwise errors in the local fluxes will remain, despite the best efforts of the test functions. These errors will then compound globally and hinder the output accuracy.

To address this issue, in multiple dimensions we supplement the trial space with Lobatto functions,[25] which are high-order polynomials associated specifically with the *boundaries* of each element. For example, we may have a $p = 1$ space defined over the interior of a given element, but have $p = 8$ Lobatto functions defined on its boundaries. In the results section, we show that this is an effective way to achieve accurate global fluxes. In the future, targeted trial space adaptation near element boundaries may be a more efficient strategy.

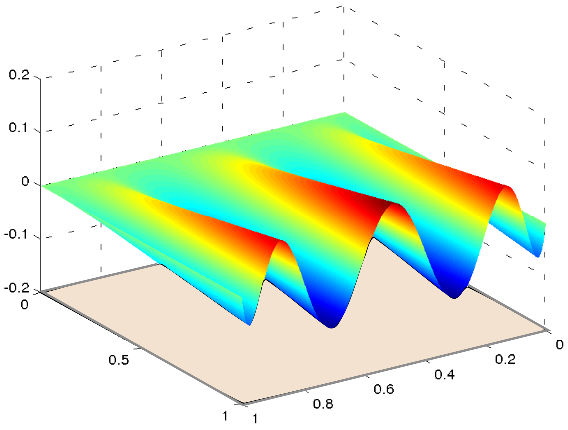


Figure 1. An eighth-order Lobatto function defined along an edge of a quadrilateral reference element. These functions are added to the trial space to improve flux resolution. Note that they are blended into the element interior in a linear manner.

II.E. Extension to Nonlinear Problems

The discussion so far has focused primarily on linear problems. However, the above ideas can be extended to nonlinear equations by simply linearizing and applying the same theory. After all, the above test function theory can be described solely in terms of adjoint equations, and it is widespread practice in optimization, error estimation, and other fields to compute adjoints based on a local linearization of a given nonlinear problem.

There is, however, a price to be paid for this linearization. If we use the above test function theory for a nonlinear problem, then the best output convergence rates we can expect to obtain will correspond to the convergence rates of so-called "corrected" outputs – i.e. standard Galerkin outputs that have been corrected by an adjoint-weighted-residual error estimate. This is because if the optimal test space already contains the global adjoints, then the BDPG outputs will converge as if they have already been "corrected" by an error estimate.

However, as is well known in error estimation, the linearization error[2] associated with a standard adjoint-based error estimate is $O(\delta u^2)$. Since δu typically converges at a rate of h^{p+1}, this gives an order of accuracy for a given adjoint-corrected output of $\mathcal{O}(h^{2p+2})$. Thus, for nonlinear problems, the best output convergence rate we expect BDPG to achieve is $\mathcal{O}(h^{2p+2})$. This is one order higher than the rate of a standard (but superconverging) DG method, which is $\mathcal{O}(h^{2p+1})$. So in the end, for nonlinear problems, the benefit of optimal test functions is to increase the order of accuracy by one.

This rate limit does have one beneficial implication. Since the best we can do is to increase the order of accuracy by one, this means that using a test space order *higher* than $p+1$ will provide no further benefit in terms of convergence rates. Thus, although we can only gain one order of accuracy with nonlinear BDPG, we can do this while using test functions that are of relatively low order – specifically, of order $p_{\text{test}} = p+1$.

III. Discretization: A Hybrid BDPG Method

The theory derived above applies to both primal and hybridized discontinuous finite element methods. However, in this work, we implement the above ideas within the framework of a hybridized discontinuous Galerkin (HDG) method. A standard HDG method treats both $\mathbf{u}$ and $\vec{\mathbf{q}}$ as independent unknowns, and introduces a separate "trace" variable, $\hat{\mathbf{u}}$, which is defined on element boundaries. For details on the HDG discretization, see e.g.[26–28] The primary benefit of HDG over DG is that it performs a static condensation of the element-interior degrees-of-freedom in terms of the face degrees-of-freedom, resulting in a smaller global system size. Figure 2 shows a pictorial comparison of DG and HDG.

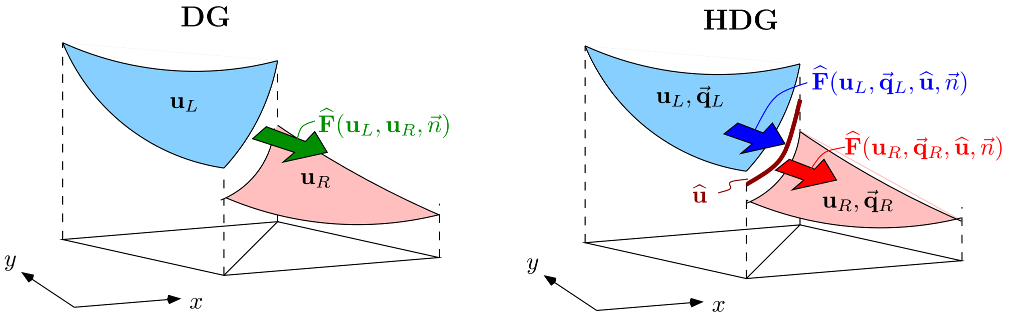

DG
$\mathbf{u}_L$
$\widehat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_R, \vec{n})$
$\mathbf{u}_R$
y
x
HDG
$\mathbf{u}_L, \vec{\mathbf{q}}_L$
$\widehat{\mathbf{F}}(\mathbf{u}_L, \vec{\mathbf{q}}_L, \hat{\mathbf{u}}, \vec{n})$
$\widehat{\mathbf{F}}(\mathbf{u}_R, \vec{\mathbf{q}}_R, \hat{\mathbf{u}}, \vec{n})$
$\hat{\mathbf{u}}$
$\mathbf{u}_R, \vec{\mathbf{q}}_R$
y
x

Figure 2. 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.

We will avoid a detailed discussion of the hybridized implementation and instead just mention a few important points. The first is that, for the hybrid BDPG method, we only compute optimal test functions associated with the $\mathbf{u}$ and $\vec{\mathbf{q}}$ trial basis functions. We do not compute optimal test functions associated with $\hat{\mathbf{u}}$. The reason for this is that accuracy in $\hat{\mathbf{u}}$ follows naturally from accuracy in $\mathbf{u}$ and $\vec{\mathbf{q}}$, so it is not necessary to attempt to optimize $\hat{\mathbf{u}}$ directly.

Secondly, we compute the discrete optimal test functions by injecting a given element HDG Jacobian (i.e. "A" matrix) to the desired order, p_{test}, and multiplying its inverse transpose by the local output linearization vector. This is a standard method of solving discrete adjoint equations.

Finally, for nonlinear problems, the optimal test functions depend on the state, so must be updated within each Newton iteration. However, when computing the primal Jacobian for the Newton solve, we assume that the test functions are "frozen;" that is, we ignore their derivatives with respect to the state. This is justified so long as the nonlinearity is not too strong.

IV. Results

In this section, we present results for BDPG in one and two dimensions. These results demonstrate that for linear problems BDPG obtains machine-precision output errors, provided the fluxes and test functions are well represented, while for nonlinear problems it has the capability to achieve

$2p + 2$ convergence rates.

IV.A. 1D Advection-Diffusion

To start, we consider a simple advection-diffusion problem in one dimension. We solve the following equation with Dirichlet boundary conditions and $a > 0$:

$$\begin{aligned} a\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} &= 0 \qquad x \in \Omega \\ u|_{x_L} &= 0 \\ u|_{x_R} &= 1\,. \end{aligned} \tag{43}$$

We choose the Reynolds number to be $aL/\nu = 10$ (where L is the domain width), the trial space order to be $p = 0$ and $p = 1$, the test space order (for BDPG) to be $p_{\text{test}} = 10$, and the boundary weights to be $w_L, w_R = 10^{15}$. The high test space order and boundary weights are chosen to demonstrate the potential boundary accuracy that can be achieved with BDPG.

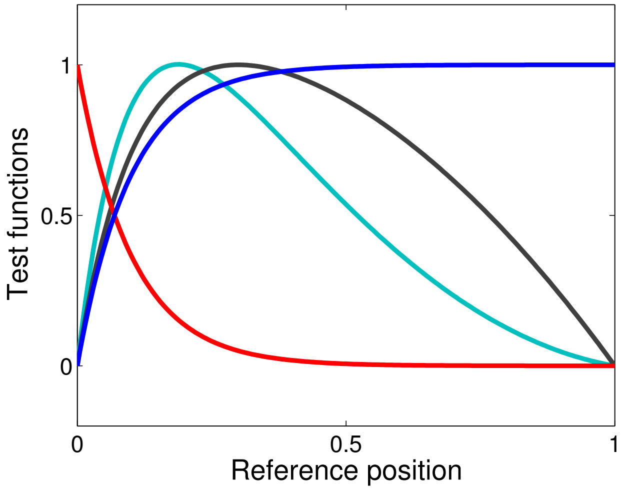


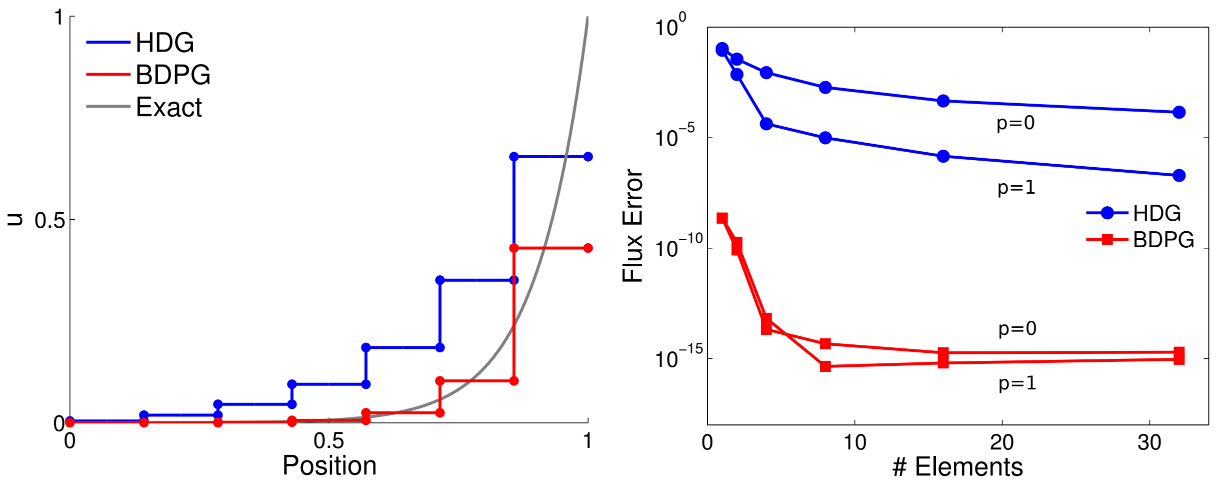
Figure 3. 1D Advection-Diffusion: The optimal test functions, in reference space, for a $p = 1$ Lagrange trial basis. Note the upwind (leftward) bias.

The optimal test finctions for a $p = 1$ trial basis are shown in Figure 3. The test functions display a clear upwind bias, which is expected, since their adjoint nature implies that they should provide a proper upwinding of the data within each element.

A comparison of the HDG and BDPG solutions in terms of both convergence rates and solution profiles is shown in Figure 4. We see that even while using completely local test functions, BDPG is able to obtain machine-precision boundary flux outputs. This verifies our earlier idea that achieving accuracy in the local fluxes is enough to guarantee accuracy in the global fluxes. Finally, note that the initial convergence rate of BDPG is due solely to the fact that the test functions are not represented exactly, but are instead approximated in a $p_{\text{test}} = 10$ space. If the test functions were exactly represented, the boundary flux error would be zero to machine precision on any size mesh.

IV.B. 1D Nonlinear Problems

Next, we move on to nonlinear problems. As mentioned, the test function theory is no longer optimal for these cases (even if the test functions are computed gobally), and the best flux convergence rate we can hope for is $2p + 2$. On the other hand, we only need $p_{\text{test}} = p + 1$ to achieve this rate. This is the test space order used for the following runs.



(a) Numerical solutions

(b) Output convergence rates

Figure 4. 1D Advection-Diffusion: Numerical solutions and output convergence rates for standard HDG and the (optimal) BDPG method. BDPG is significantly more accurate near the boundaries. The error in the right-boundary output for $p = 0$ and $p = 1$ runs is shown in (b).

IV.B.1. Euler

We first try a subsonic Euler case, which consists of an inflow at the left boundary, an inviscid wall at the right boundary, and a source term $\mathbf{S}^T = [\,0.4\rho^2 \;\; 0.7(\rho u)^2 \;\; 0.1(\rho H)^2\,]$ added to the left-hand side of the equations. The flow enters from the left and collides with the wall on the right, while the source term acts as a sink that relieves the buildup of mass that would otherwise occur within the domain. Figure 5 shows the steady-state values of the solution states (density, momentum, and energy) throughout the domain.

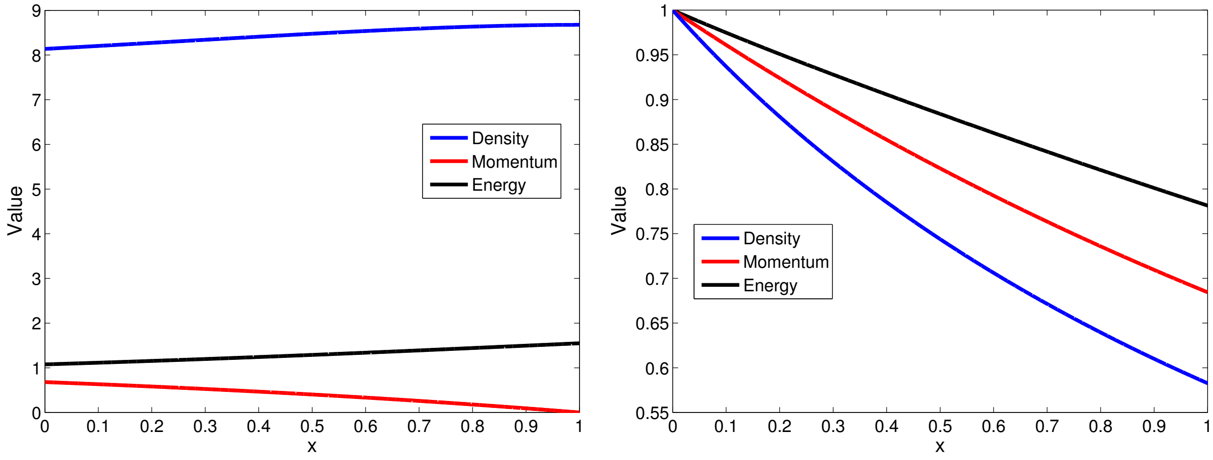
We solve this problem numerically with both HDG and BDPG for a trial space order of $p = 1$. For BDPG, we choose the test space order to be $p_{\text{test}} = p + 1 = 2$ and the boundary weights to be large; specifically, $w_L, w_R = 10^8$. Note that for nonlinear problems, since the accuracy of the fluxes depends to a certain extent on the accuracy of the element-interior solution, we do not take the boundary weights to be as large as for linear problems.

Figure 6a shows the convergence of the energy flux on the left boundary for both HDG and BDPG as the mesh is refined. For the BDPG runs, we observe a rate of 3.72, which is close to (but slightly below) the optimal rate of $2p + 2 = 4$. Similar rates are obtained for the remaining flux components on both left and right boundaries.

Next, we try a supersonic case, where the Mach number is approximately 2 and Dirichlet values of 1 (in all state components) are set on the left. The same quadratic source term is used as in the above case, and steady-state solution profiles are shown in Figure 5b. To solve the problem, we again use HDG and BDPG, with trial space orders ranging from $p = 0$ to $p = 2$. The test space order and boundary weights are again taken to be $p_{\text{test}} = p + 1$ and $w_L, w_R = 10^8$, respectively. Figure 6 (parts *b*-*d*) shows convergence rates for various flux outputs and trial space orders p. We see that for this problem BDPG achieves the optimal rate of $2p + 2$ for all fluxes and trial space orders.

IV.B.2. Navier-Stokes

Next, to confirm that BDPG performs well for nonlinear problems with viscosity, we try a Navier-Stokes run similar to the above Euler cases. The flow starts out supersonic at the inflow (with Mach



(a) Subsonic case

(b) Supersonic case

Figure 5. 1D Euler: (a) Subsonic case with inviscid wall on right boundary. (b) Supersonic case with Dirichlet conditions on left and outflow on right.

number of 1.5) and becomes subsonic as it collides with a wall on the right boundary. The inflow Reynolds number is 10 (the same as our earlier advection-diffusion case), and a quadratic source term of $\mathbf{S}^T = [\,0.3\rho^2 \;\; 0.1(\rho u)^2 \;\; 0.1(\rho H)^2\,]$ is used to allow for a steady-state solution. Figure 7 shows the corresponding solution profiles throughout the domain.

The boundary flux convergence for both HDG and BDPG is shown in Figure 8, where the same test space properties are used for BDPG as in the Euler cases. We see that BDPG again outperforms HDG, and attains the optimal rate of $2p+2$ in the fluxes on both the left and right boundaries.

IV.C. 2D Linear Problems

With the performance of BDPG confirmed for both linear and nonlinear problems in one dimension, we now turn to linear problems in two dimensions.

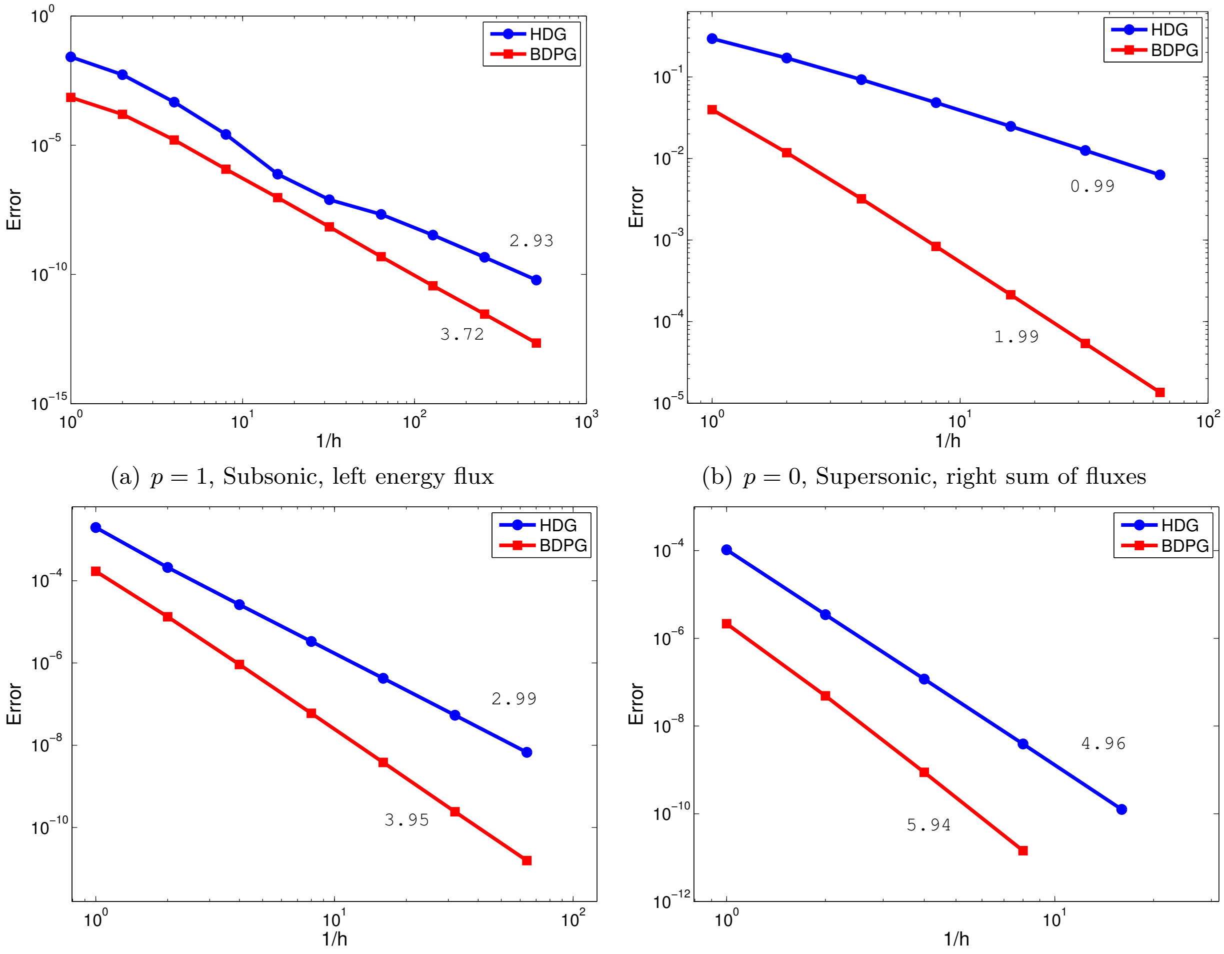
IV.C.1. 2D Advection-Diffusion

We first try an advection-diffusion case with $Re = 100$. An analytic Dirichlet boundary condition of

$$u(x,y) = \exp\left[\frac{1}{2}\sin\left(-4x+6y\right) - \frac{4}{5}\cos\left(3x-8y\right)\right] \qquad \vec{x} \in \partial\Omega \tag{44}$$

is specified on all sides of the domain, which generates boundary layers on the two outflow boundaries (the top and right). Figure 9 shows contours of both the solution and a sample optimal test function.

As mentioned, to ensure adequate flux resolution for multidimensional problems, we enrich the BDPG trial space with Lobatto functions on element boundaries. For the results shown in Figure 10, we consider boundary enrichment orders of $p_B = 6, 7, 8$, while keeping the interior basis at a low order of $p_I = 1$. The test space order is set to $p_{\text{test}} = p_B$. We compare the BDPG results to a standard HDG method with the same (interior) order of $p_I = 1$. From Figure 10, we see that BDPG delivers a reduction in boundary-flux errors of nearly 6 orders of magnitude. As an additional benefit, it also achieves greater accuracy in interior outputs, as evidenced by the domain integral output shown in Figure 10d.



(a) $p = 1$, Subsonic, left energy flux

(b) $p = 0$, Supersonic, right sum of fluxes

(c) $p = 1$, Supersonic, right sum of fluxes

(d) $p = 2$, Supersonic, right sum of fluxes

Figure 6. 1D Euler: Various flux outputs for subsonic and supersonic cases. The supersonic BDPG runs achieve the optimal $2p + 2$ rate, while the subsonic run nearly attains this rate.

IV.C.2. 2D Linearized Euler

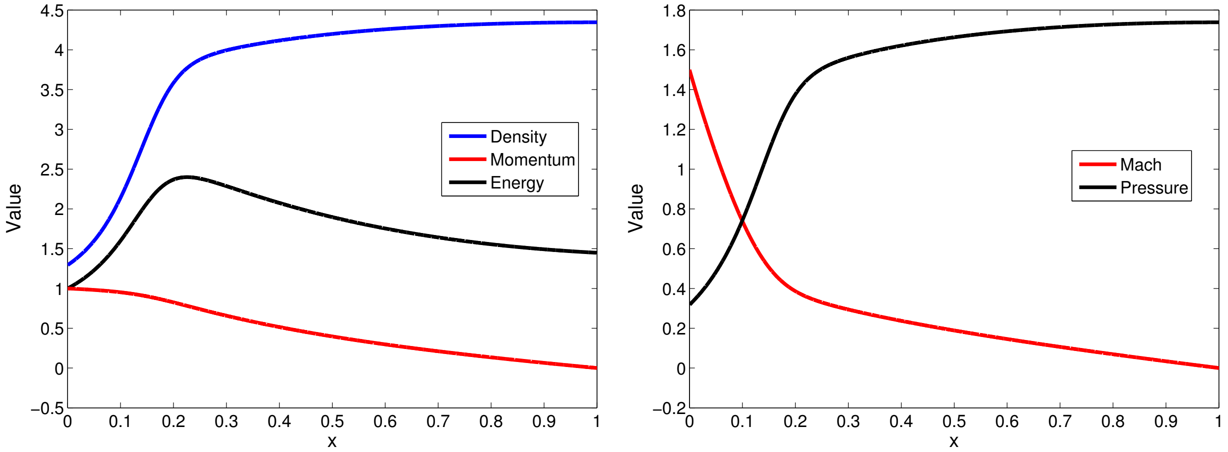
Finally, we move on to two-dimensional systems of equations. In particular, we solve the homentropic linearized Euler equations, with state variables and fluxes given by

$$\mathbf{u} = \begin{bmatrix} p \\ u_i \end{bmatrix}, \quad \mathbf{F}_j = \begin{bmatrix} u_{0_j} p + \rho_0 a_0^2 u_j \\ \frac{p}{\rho_0}\delta_{ij} + u_{0_j} u_i \end{bmatrix}, \tag{45}$$

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 a first test, we try a NACA 0012 airfoil with horizontal background flow at a Mach number of 0.3. The pressure and velocity perturbations are set to unity on the farfield boundaries, so that the net velocity perturbation is upward and to the right. The mesh and solution contours are shown in Figure 11. Note that the mesh is curved (with $Q = 4$ geometry representation) and has hanging-node refinement near the airfoil, providing a first test of BDPG on a general mesh topology.

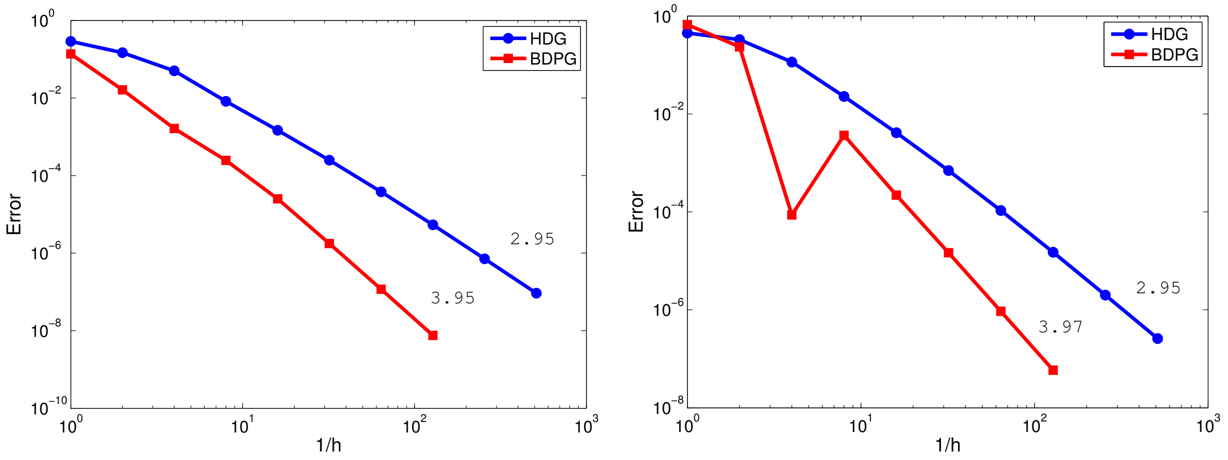
For the BDPG runs, we choose the boundary weights to be 10^{10} and the test space order to



(a) Conservative states

(b) Pressure and Mach number

Figure 7. 1D Navier-Stokes: (a) State variables within the domain. (b) Mach number and pressure variation within the domain. Note that the flow transitions from supersonic to subsonic near the inflow.



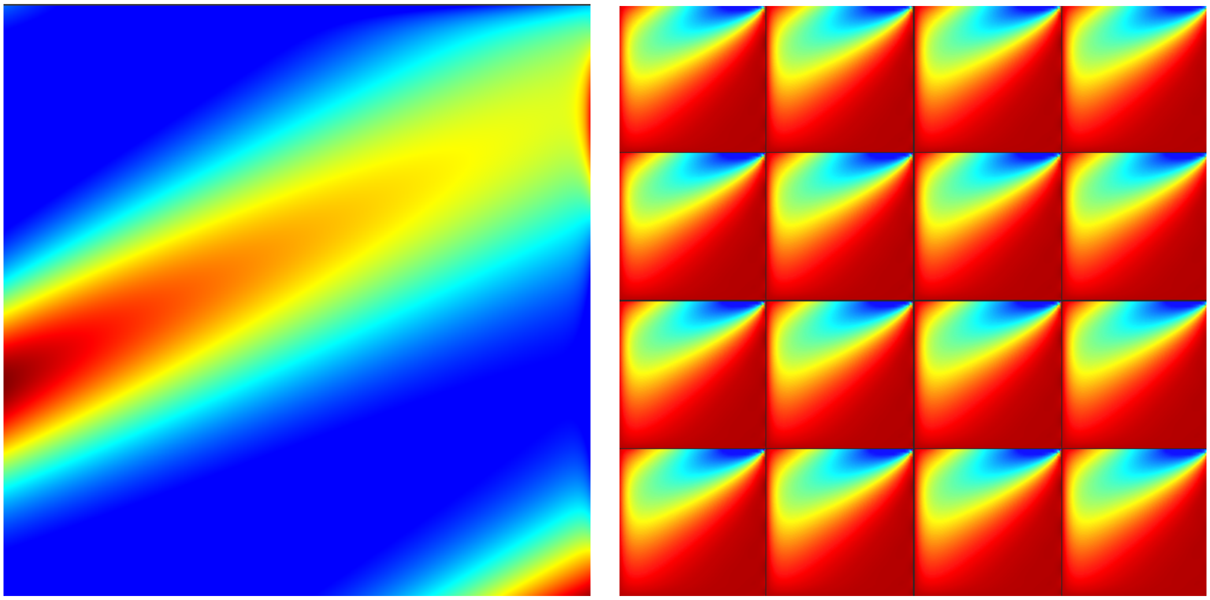
(a) Right x-momentum flux

(b) Left energy flux

Figure 8. 1D Navier-Stokes: Convergence rates for a mixed supersonic/subsonic flow with $p = 1$. BDPG obtains optimal $2p + 2$ rates.

be equal to the enrichment order, i.e. $p_{\text{test}} = p_B$. (There is not much benefit in taking $p_{\text{test}} > p_B$, since the pointwise flux errors cannot be reduced beyond the interpolation error dictated by p_B.) Convergence rates for $p = 1$ HDG and BDPG with various edge enrichments are provided in Figure 11. From the figure, we see that BDPG achieves orders of magnitude lower errors in the boundary fluxes compared to HDG. As expected, use of a higher enrichment order p_B results in improved performance for BDPG.

While BDPG performs well for the above airfoil, the trailing edge singularity limits the potential convergence rates. To eliminate the effect of this singularity, we round off the trailing edge and instead simulate the flow around the flattened ellipse shown in Figure 13. Figure 14 shows the convergence rates for the fluxes along the ellipse boundary. We see that with the singularity removed, BDPG shows even greater gains relative to HDG, achieving boundary flux errors roughly 6 orders of magnitude lower than HDG by the final refinement.



(a) Solution, u

(b) Optimal test functions

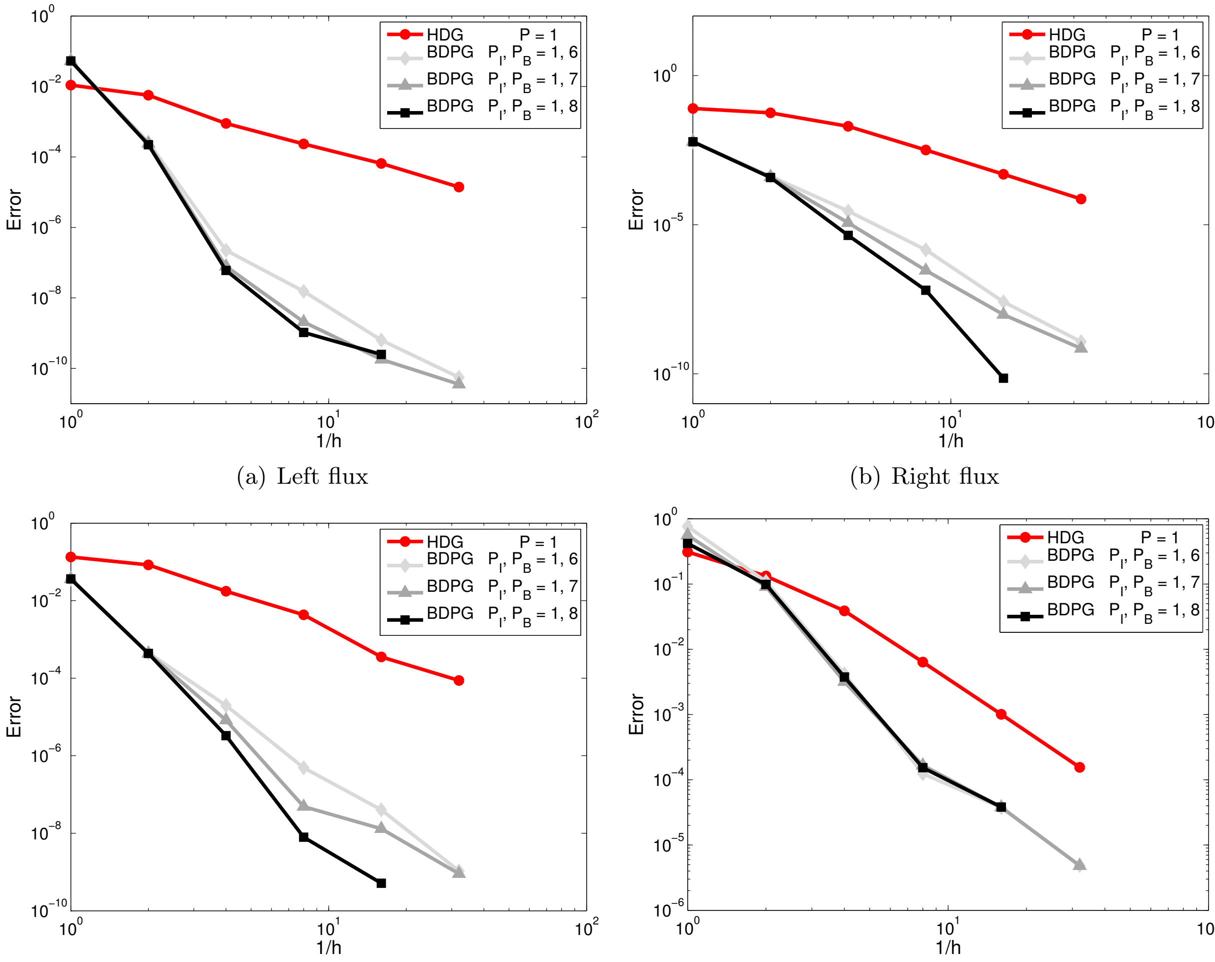
Figure 9. 2D Advection-Diffusion: (a) The solution to a $Re = 100$ problem on a fine mesh. (b) A particular optimal test function shown on each element in the domain. This specific test function ensures accuracy in the flux through the top face of each element. Additional test functions (not shown) ensure accuracy in the fluxes through the remaining faces. The advective velocity is upward and to the right, and the upwinding nature of the optimal test functions is apparent.

IV.C.3. 2D Ellipse: Additional Comparisons

At this point, we may wonder to what extent BDPG's error reduction is due specifically to the use of optimal test functions, since the fact that the BDPG trial space is enriched may provide improved accuracy on its own. To demonstrate that BDPG's accuracy gains are not *just* due to the trial space enrichment, we perform another ellipse simulation in which the same boundary-enriched trial space is used for both BDPG and HDG. In this case, the only difference between BDPG and HDG is the test space. Figure 15a illustrates the results of this test, showing the pressure flux convergence for both methods. (The other fluxes show similar behavior.) We see that even when HDG uses an enriched trial space, BDPG achieves boundary-flux errors that are roughly 6 orders of magnitude lower for $p_B = 8$. 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.

Lastly, as a final demonstration of the 2D test function performance, we compare BDPG runs in which $p_{\text{test}} = p_B$ to standard HDG runs of order $p = p_B$. In other words, we do the following: We start with a standard HDG method at order p. For the BDPG scheme, we then delete all interior trial space modes, leaving only the edge representation at order p, so that $p_B = p$ while $p_I = 1$. Finally, we stipulate that the test functions be computed in an order $p_{\text{test}} = p$ space. This creates a situation in which BDPG cannot possibly do better than HDG, because HDG contains all order-p functions in its test space, and will necessarily include any of the test functions used by BDPG.

The goal then is to see how BDPG's performance compares to HDG in this situation. The idea is that, if BDPG's test functions are truly optimal, then we should theoretically be able to recover the same boundary accuracy as HDG, despite the fact that we have deleted *all* of the interior trial functions (and their associated test functions). On the other hand, if there were some error in the optimal test function derivation or computation, the performance of BDPG would suffer relative



(a) Left flux

(b) Right flux

(c) Top flux

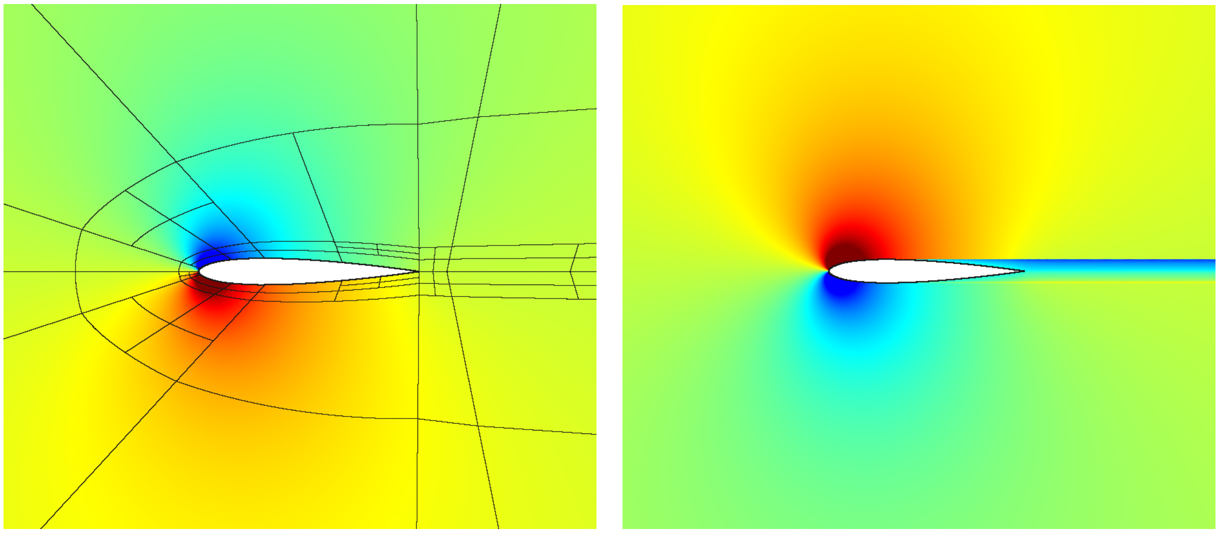
(d) Domain u^2 ouptut

Figure 10. 2D 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 as the amount of boundary enrichment increases. Note that the interior u^2 output is also accurate, despite the fact that it is nonlinear and that the method is not designed for interior ouptuts.

to HDG.

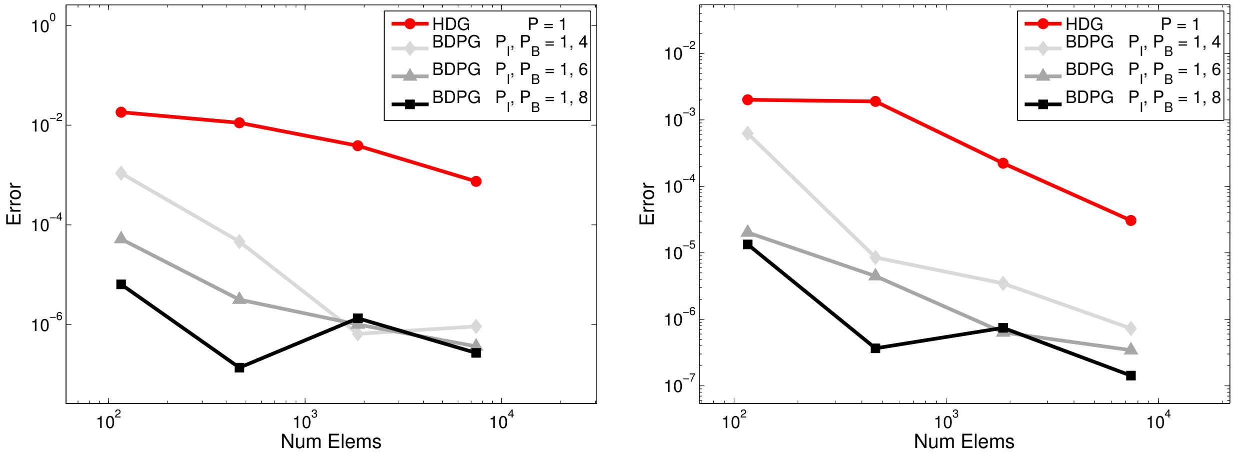
Figure 15 shows these comparisons for $p = 4$, 6, and 8 for the ellipse case. From the figure, we see that the BDPG results lie exactly on top of the HDG values – meaning that the optimal test functions are in fact performing well, and are enabling BDPG to achieve the same accuracy as HDG with fewer total degrees of freedom. For example, if we consider the case where $p = 6$, then standard HDG has $(p+1)^2 = 49$ basis coefficients (i.e. unknowns) per element. On the other hand, after deleting the interior trial space modes, BDPG has only $4p = 24$ unknowns per element. So for this case, BDPG attains the same accuracy as HDG with just *half* the total number of degrees of freedom.

That said, it must be emphasized that fewer total degrees of freedom does not necessarily translate into lower computational time. In particular, for hyridized methods, the computational time scales primarily with the number of degrees of freedom on element boundaries (i.e. the trace degrees of freedom). In this case, the advantage of BDPG over HDG in terms of CPU time is not clear, since the edge enrichment leaves a relatively large number of degrees of freedom on element boundaries. However, the edge enrichment used in the current work is a “brute-force” approach,



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

Figure 11. 2D airfoil: Pressure and x-velocity contours.



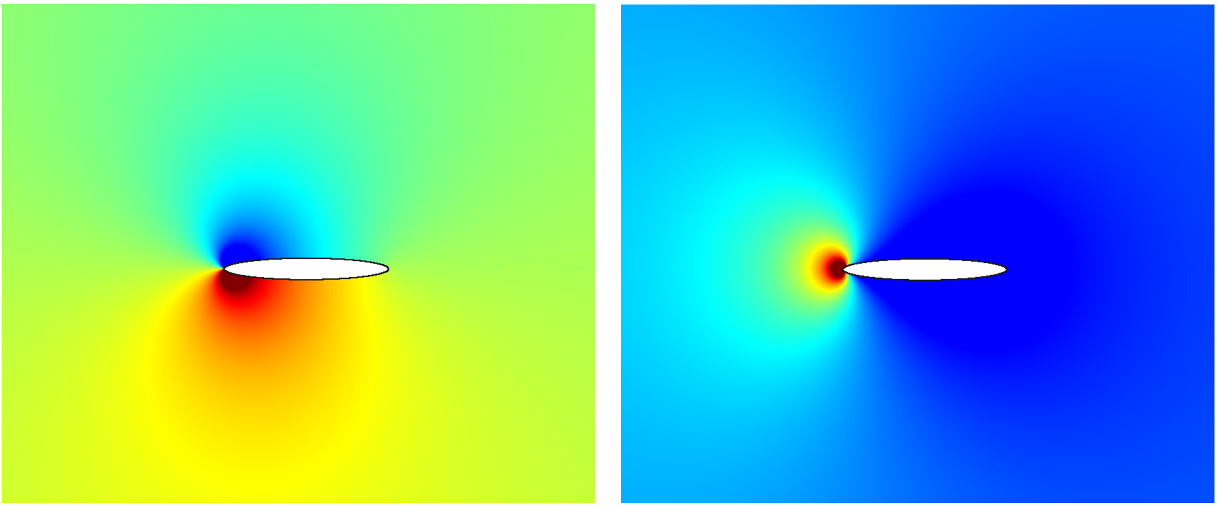
(a) Pressure flux (b) x-velocity flux

Figure 12. 2D airfoil: Pressure and x-velocity flux convergence for both BDPG and HDG. While the convergence rates are limited by the trailing-edge singularity, BDPG still outperforms standard HDG.

and can be performed in a more intelligent manner that would provide further CPU time reductions. In addition, 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 in terms of CPU time.

V. Conclusions and Future Work

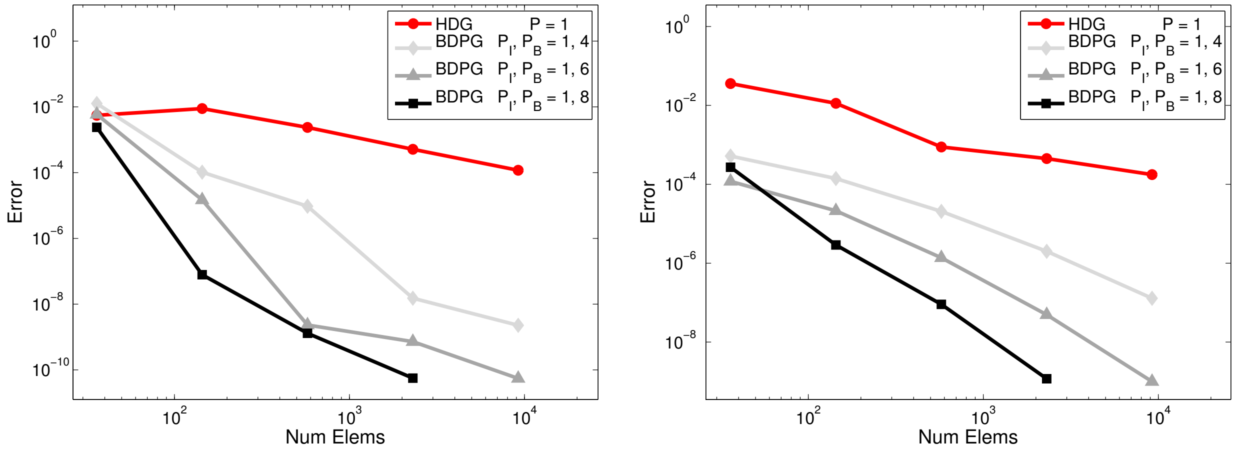
In this work, we have shown how to derive and compute optimal test functions for discontinuous finite element methods. These test functions render a finite element method optimal in a chosen error norm. The theory applies to arbitrary linear PDEs in multiple dimensions and can be extended to nonlinear equations. We have further shown that, if the primary goal is to achieve



(a) Pressure contours

(b) y-velocity contours

Figure 13. 2D ellipse: Pressure and y-velocity contours.



(a) Pressure flux

(b) x-velocity flux

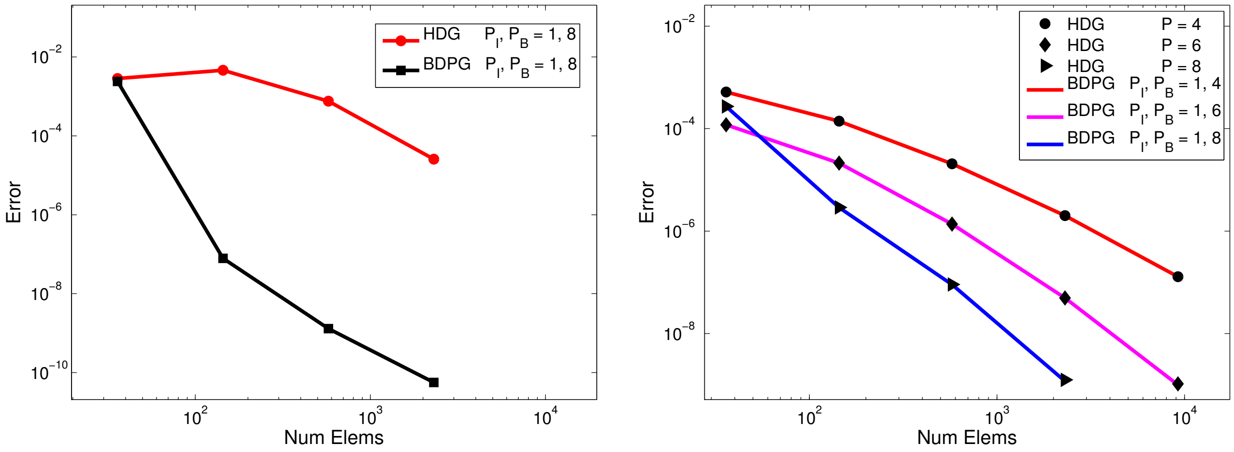
Figure 14. 2D ellipse: Pressure and x-velocity flux convergence for both BDPG and HDG.

accuracy in boundary outputs, the theory can be localized and the test functions can be computed independently on each element. These test functions satisfy local adjoint equations and ensure that a proper upwinding of information occurs within each element. When used in a discrete setting, a "boundary" discontinuous Petrov-Galerkin (BDPG) method results.

For linear problems, if the test functions and fluxes are well-represented, BDPG can obtain exact boundary fluxes, while for nonlinear problems rates of $O(h^{2p+2})$ are achieved. On the other hand, if the test functions are nonsmooth or the fluxes not well-represented, the performance of the method can suffer. Addressing these issues while providing further reductions in CPU time is the subject of future research.

References

[1] Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011,



(a) Pressure flux

(b) x-velocity flux

Figure 15. 2D ellipse: (a) BDPG vs. HDG using the same edge-enriched trial basis for both methods. The only difference between the methods is the test space. (b) BDPG with $p_{\text{test}} = p_B$ compared to HDG with a full order-p basis. The fact that the BDPG results lie exactly on top of the HDG values demonstrates that BDPG can achieve the same accuracy as HDG while using fewer total degrees of freedom (specifically, $4p_B$ vs. $(p+1)^2$).

pp. 673–694.

[2] Becker, R. and Rannacher, R., “An optimal control approach to a posteriori error estimation in finite element methods,” *Acta Numerica*, edited by A. Iserles, Cambridge University Press, 2001, pp. 1–102.

[3] Giles, M. B. and Süli, E., “Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality,” *Acta Numerica*, Vol. 11, 2002, pp. 145–236.

[4] Demkowicz, L. and Gopalakrishnan, J., “A class of discontinuous Petrov-Galerkin methods. Part I: The transport equation,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, April 2010, pp. 1558–1572.

[5] Demkowicz, L. and Gopalakrishnan, J., “A class of discontinuous Petrov-Galerkin methods. II. Optimal test functions,” *Numerical Methods for Partial Differential Equations*, Vol. 27, 2011, pp. 70–105.

[6] Zitelli, J., Muga, I., Demkowicz, L., Gopalakrishnan, J., Pardo, D., and Calo, V. M., “A class of discontinuous Petrov-Galerkin methods. Part IV: The optimal test norm and time-harmonic wave propagation in 1D,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 2406–2432.

[7] Chan, J., Demkowicz, L., Moser, R., and Roberts, N., “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*, Vol. 78712, 2010.

[8] Venditti, D. A. and Darmofal, D. L., “Grid adaptation for functional outputs: application to two-dimensional inviscid flows,” *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–69.

[9] Hughes, T., “Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 127, March 1995, pp. 387–401.

[10] Hughes, T. J., Feijóo, G. R., Mazzei, L., and Quincy, J.-B., “The variational multiscale method – a paradigm for computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 166, No. 1, 1998, pp. 3–24.

[11] Farhat, C., Harari, I., and Hetmaniuk, U., “The discontinuous enrichment method for multiscale analysis,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, May 2003, pp. 3195–3209.

[12] Hughes, T., Scovazzi, G., Bochev, P., and Buffa, A., “A multiscale discontinuous Galerkin method with the computational structure of a continuous Galerkin method,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, June 2006, pp. 2761–2787.

[13] Barrett, J. and Morton, K. W., “Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems,” *Computer methods in applied mechanics and engineering*, Vol. 45, No. 1, 1984, pp. 97–122.

[14] Celia, M., Russell, T., Herrera, I., and Ewing, R., “An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation,” *Advances in Water Resources*, Vol. 13, December 1990, pp. 187–206.

[15] Herrera, I., "Trefftz Method: A General Theory," *Numerical Methods for Partial Differential Equations*, Vol. 16, November 2000, pp. 561–580.

[16] Brooks, A. and Hughes, T., "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, September 1982, pp. 199–259.

[17] Barbone, P. and Harari, I., "Nearly H^1-optimal finite element methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, November 2000, pp. 5679–5690.

[18] Givoli, D., "Non-local and Semi-local Optimal Weighting Functions for Symmetric Problems Involving a Small Parameter," *International Journal for Numerical Methods in Engineering*, Vol. 26, 1988, pp. 1281–1298.

[19] Demkowicz, L. and Oden, J., "An Adaptive Characteristic Petrov-Galerkin Finite Element Method for Convection-Dominated Linear and Nonlinear Parabolic Problems in One Space Variable," *Journal of Computational Physics*, Vol. 67, 1986, pp. 188–213.

[20] Moro, D., Nguyen, N., Peraire, J., and Gopalakrishnan, J., "A hybridized discontinuous Petrov-Galerkin method for compressible flows," AIAA-2011-197, Orlando, FL, 2011.

[21] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260.

[22] Lu, J., *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.

[23] Hartmann, R., "Adjoint consistency analysis of Discontinuous Galerkin discretizations," *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.

[24] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[25] Solín, P., Segeth, K., and Zel, I. D., *Higher–Order Finite Element Methods*, Chapman and Hall, 2003.

[26] Peraire, J., Nguyen, N. C., and Cockburn, B., "An Embedded Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations," AIAA Paper 2011-3228, 2011.

[27] Nguyen, N., Peraire, J., and Cockburn, B., "Hybridizable discontinuous Galerkin methods," *Spectral and High Order Methods for Partial Differential Equations*, Springer, 2011, pp. 63–84.

[28] Fidkowski, K., "High-Order Output-Based Adaptive Methods for Steady and Unsteady Aerodynamics," *37th Advanced VKI CFD Lecture Series*, von Karman Institute, 2013.