INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2012; **91**:950–970
Published online 13 June 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4300

A hybridized discontinuous Petrov–Galerkin scheme for scalar conservation laws

D. Moro*,†, N. C. Nguyen and J. Peraire

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

SUMMARY

We present a hybridized discontinuous Petrov–Galerkin (HDPG) method for the numerical solution of steady and time-dependent scalar conservation laws. The method combines a hybridization technique with a local Petrov–Galerkin approach in which the test functions are computed to maximize the inf-sup condition. Since the Petrov–Galerkin approach does not guarantee a conservative solution, we propose to enforce this explicitly by introducing a constraint into the local Petrov–Galerkin problem. When the resulting nonlinear system is solved using the Newton–Raphson procedure, the solution inside each element can be locally condensed to yield a global linear system involving only the degrees of freedom of the numerical trace. This results in a significant reduction in memory storage and computation time for the solution of the matrix system, albeit at the cost of solving the local Petrov–Galerkin problems. However, these local problems are independent of each other and thus perfectly scalable. We present several numerical examples to assess the performance of the proposed method. The results show that the HDPG method outperforms the hybridizable discontinuous Galerkin method for problems involving discontinuities. Moreover, for the test case proposed by Peterson, the HDPG method provides optimal convergence of order $k+1$. Copyright © 2012 John Wiley & Sons, Ltd.

Received 5 October 2011; Revised 17 January 2012; Accepted 27 January 2012

KEY WORDS: hybridized discontinuous Galerkin; Petrov–Galerkin; nonlinear conservation laws

1. INTRODUCTION

The development of robust, accurate, and efficient methods for the numerical solution of conservation laws in complex geometries is a topic of considerable importance. Indeed, hyperbolic systems of conservation laws govern a wide range of physical phenomena and arise in several areas of applied mathematics and mechanics such as fluid dynamics, thermodynamics, population dynamics, magnetohydrodynamics, multiphase flow in nonlinear material, and traffic flow. The most fundamental phenomenon of hyperbolic systems is the formation and propagation of discontinuities and shock waves even if initial and boundary data are smooth. The presence of shock waves is a serious challenge for any numerical methods to provide a physical and stable solution. Although significant progress has been made over the years in both the theoretical and numerical investigation, the numerical solution of hyperbolic conservation laws remains an active research area with many challenging problems to be addressed.

In recent years, considerable attention has been turned to discontinuous Galerkin (DG) methods [1–13] for the numerical solution of hyperbolic conservation laws. DG methods possess several attractive properties for solving hyperbolic problems. In particular, they are flexible for complicated geometry, locally conservative, high-order accurate, highly parallelizable, and have low dissipation and dispersion. However, most existing DG methods suffer from two major drawbacks. The first

*Correspondence to: D. Moro, 77 Massachusetts Ave. 37-422 Cambridge MA 02139, USA.
†E-mail: dmoro@mit.edu

drawback is that they are computationally expensive due to the large number of degrees of freedom (DOFs) caused by nodal duplication at the element boundary interfaces. The memory storage and computation cost of DG methods are typically several times greater than that of CG methods. The second drawback is that higher-order DG methods are generally less robust than low-order methods when solution features are under-resolved (in particular at shock waves).

More recently, a new class of implicit DG methods—the so-called hybridizable discontinuous Galerkin (HDG) method—was first introduced for elliptic problems [14]. The HDG methods have already been extended to convection–diffusion systems [15, 16], linear and nonlinear elastodynamics [17, 18], incompressible and compressible flows [19–26], and electromagnetics [27]. The main idea of HDG methods is a hybridization of DG methods, which aims to solve for the numerical trace of the approximate solution instead of the approximate solution itself. Because the numerical trace is defined over inter-element boundaries and is single-valued over the element faces, HDG methods have significantly less DOFs than standard DG methods. In fact, a variant of the HDG method—the so-called embedded DG method [28, 29]—has the same global DOFs as CG methods and has the stability properties of a DG method. This large reduction in the number of DOFs can lead to significant savings for both computational time and memory storage. Another advantage of HDG methods is that their post-processed solution and approximate gradient converge with one order higher than those of other DG and CG methods for diffusion-dominated problems. These advantages render HDG methods competitive with CG methods even for diffusion problems and elasticity problems [15, 17, 18, 30].

Another interesting DG approach is the discontinuous Petrov–Galerkin method (DPG) first introduced for convection problems [7] and later extended to linear convection–diffusion problems [31]. The main idea of the DPG method is an automatic construction of optimal test functions to maximize the stability constant. The performance of the DPG method is shown to be superior to the standard DG method. In particular, the DPG method delivers optimal convergence rate $k + 1$ for the Peterson example, where it has been known that other DG methods yield a convergence rate of only $k + 1/2$. The stability of the DPG scheme is excellent. However, the DPG method is more expensive than other DG methods because it contains more globally coupled unknowns. Another drawback of the DPG method is that the method is *not* conservative because its test space does not contain a constant function.

In this paper, we introduce a hybridized discontinuous Petrov–Galerkin (HDPG) method that combines the efficiency of the HDG method with the excellent stability of the DPG method. The main idea here is to use the DPG method for the local problem and the HDG method for the global problem. The global unknown and in fact the matrix structure of the HDPG method is thus the same as that of the HDG method. Moreover, in order to render the HDPG method *conservative*, we propose to enrich the test space with a constant function by introducing a constraint into the local problem. We present several numerical examples to demonstrate the performance of the HDPG method. Numerical results show that the HDPG method is more robust and stable than the HDG method for a number of test cases. Moreover, for the test case proposed by Peterson [32], the HDPG method provides optimal convergence of order $k + 1$.

The paper is organized as follows. In Section 2, we introduce some notation used throughout the paper and present a brief overview of both the HDG method and the DPG method. We then describe the HDPG method in Section 3 and present numerical results in Section 4. Finally, in Section 5, we draw some concluding remarks.

2. OVERVIEW

2.1. Problem statement and notation

We first introduce the problem of interest, which is a scalar conservation law of the following form:

$$\begin{aligned} \nabla \cdot \mathbf{F}(u) - \nabla \cdot (\epsilon \nabla u) &= f, \quad && \text{in } \Omega, \\ u &= g_D, && \text{on } \partial\Omega. \end{aligned} \tag{1}$$

DOI: 10.1002/nme

As usual in the DG context, the problem is written as a first-order system:

$$\begin{aligned} \nabla \cdot \mathbf{F}(u) - \nabla \cdot (\epsilon \mathbf{q}) &= f, && \text{in } \Omega, \\ \mathbf{q} - \nabla u &= 0, && \text{in } \Omega, \\ u &= g_D, && \text{on } \partial\Omega. \end{aligned} \tag{2}$$

Here, $\Omega \in \mathbb{R}^d$ is the physical domain in d spatial dimensions with Lipschitz boundary $\partial\Omega$, $f \in L^2(\Omega)$ is a square integrable source term, $\epsilon \in L^\infty(\Omega)$ represents the isotropic diffusion coefficient, and $g_D \in L^2(\partial\Omega)$ represents the boundary data. Moreover, $u \in L^2(\Omega)$ represents the scalar field, and $\mathbf{F}(u) \in (L^\infty(\Omega))^d$ is a vector-valued function of the solution u.

Let $\mathcal{T}_h$ denote a collection of disjoint elements K that partition the domain Ω. Let $\partial\mathcal{T}_h$ denote the set of faces of the triangulation $\mathcal{T}_h$, formed by collecting the faces of each element K, this is, $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$. For a given element of the triangulation K, $e = \partial K \cap \partial\Omega$ is a boundary face if its $d-1$ Lebesgue measure is non-zero. Similarly, for two elements K^+ and K^- of the triangulation, $e = \partial K^+ \cap \partial K^-$ is an interior face if its $d-1$ Lebesgue measure is non-zero. Let $\mathcal{E}_h^i$ denote the set of interior faces and $\mathcal{E}_h^\partial$ denote the set of boundary faces. We denote by $\mathcal{E}_h := \mathcal{E}_h^i \cup \mathcal{E}_h^\partial$ the union of both sets. Notice that each interior face in $\mathcal{E}_h^i$ is represented twice in $\partial\mathcal{T}_h$, whereas each boundary face in $\mathcal{E}_h^\partial$ is represented only once. Finally, let $\mathbf{n}^+$ and $\mathbf{n}^-$ denote the outward unit normal for element K^+ and K^-, respectively. Notice, by definition, if K^+ and K^- share a face e of $\mathcal{E}_h$, then $\mathbf{n}^+ = -\mathbf{n}^-$.

Let $\mathcal{P}^p(D)$ denote the set of polynomials of order at most p in a domain D. We define the finite element spaces as follows:

$$\begin{aligned} V_h^p &= \left\{ v \in L^2(\mathcal{T}_h) : v|_K \in \mathcal{P}^p(K) \quad \forall K \in \mathcal{T}_h \right\}, \\ \mathbf{W}_h^p &= \left\{ \mathbf{w} \in (L^2(\mathcal{T}_h))^d : \mathbf{w}|_K \in (\mathcal{P}^p(K))^d \quad \forall K \in \mathcal{T}_h \right\}, \\ M_h^p &= \left\{ \mu \in L^2(\mathcal{E}_h) : \mu|_e \in \mathcal{P}^p(e) \quad \forall e \in \mathcal{E}_h \right\}, \end{aligned}$$

that are polynomials inside each element (in the case of V_h^p and $\mathbf{W}_h^p$) or face (in the case of M_h^p), but discontinuous across them. We also define $M_h^p(g_D) = \{\mu \in M_h^p : \mu = P(g_D)\}$, where $P(\cdot)$ denotes the L^2 projection of the given data $(\cdot)$ on M_h^p.

Finally, we define the following inner products as follows:

$$\begin{aligned} (a,b)_{\mathcal{T}_h} &= \sum_{K \in \mathcal{T}_h} (a,b)_K, & (\mathbf{a},\mathbf{b})_{\mathcal{T}_h} &= \sum_{K \in \mathcal{T}_h} \sum_{i=1}^{d} (a_i, b_i)_K, \\ \langle a,b \rangle_{\mathcal{E}_h} &= \sum_{e \in \mathcal{E}_h} \langle a,b \rangle_e, & \langle a,b \rangle_{\partial\mathcal{T}_h} &= \sum_{K \in \mathcal{T}_h} \langle a,b \rangle_{\partial K}, \end{aligned}$$

where, for any functions $a, b \in L^2(D)$, we define $(a,b)_D = \int_D ab$ if $D \in \mathbb{R}^d$ and $\langle a,b \rangle_D = \int_D ab$ if $D \in \mathbb{R}^{d-1}$.

2.2. HDG Scheme

Given an element K of the triangulation $\mathcal{T}_h$, let $\hat{u}$ be a function supported on ∂K. We introduce the so-called local problem as follows:

$$\begin{aligned} \nabla \cdot \mathbf{F}\left(u^{\hat{u}}\right) - \nabla \cdot \left(\epsilon \mathbf{q}^{\hat{u}}\right) &= f, && \text{in } K, \\ \mathbf{q}^{\hat{u}} - \nabla u^{\hat{u}} &= 0, && \text{in } K, \\ u^{\hat{u}} &= \hat{u}, && \text{on } \partial K. \end{aligned} \tag{3}$$

The local problem defines a Dirichlet-to-Neumann mapping $T : \hat{u} \mapsto \left(\mathbf{F}\left(u^{\hat{u}}\right) - \epsilon \mathbf{q}^{\hat{u}}\right) \cdot \mathbf{n}$ that maps boundary data $\hat{u}$ to the fluxes on ∂K. It is clear that if $\hat{u} = u|_{\partial K}$, then, we have $\mathbf{q}^{\hat{u}} = \mathbf{q}, u^{\hat{u}} = u$.

DOI: 10.1002/nme

However, $\hat{u}$ is unknown unless an extra condition is prescribed. Thanks to the conservative character of the equation, we can impose that jumps in the fluxes across faces must be zero in the interior, together with the appropriate boundary condition:

$$\begin{aligned} \left(\mathbf{F}\left(u^{\hat{u}}\right)-\epsilon\mathbf{q}^{\hat{u}}\right)_e^{+}\cdot\mathbf{n}^{+}+\left(\mathbf{F}\left(u^{\hat{u}}\right)-\epsilon\mathbf{q}^{\hat{u}}\right)_e^{-}\cdot\mathbf{n}^{-}=0, \qquad & \forall e\in\mathcal{E}_h^i, \\ \hat{u}=g_D, \quad & \forall e\in\mathcal{E}_h^{\partial}. \end{aligned} \tag{4}$$

This equation gives rise to the global system in terms of $\hat{u}$ because $(u^{\hat{u}},\mathbf{q}^{\hat{u}})$ is a function of $\hat{u}$ by (3). The discrete version of the system (3)–(4) defines the HDG method.

As we discuss later, depending on the choice of the approximation space, several HDG schemes can be devised [14]. In this particular case, we take the usual DG spaces consisting of piecewise polynomials of the same order p for all the unknowns. Multiplying (3)–(4) by test functions, and integrating by parts over each element or face, we arrive at the following weak formulation for the approximate solution; find $(u_h,\mathbf{q}_h,\hat{u}_h)\in V_h^p\times\mathbf{W}_h^p\times M_h^p(g_D)$ such that

$$\begin{aligned} \langle\left(\hat{\mathbf{F}}_h-\epsilon\hat{\mathbf{q}}_h\right)\cdot\mathbf{n},v\rangle_{\partial K}-(\mathbf{F}(u_h)-\epsilon\mathbf{q}_h,\nabla v)_K&=(f,v)_K,\\ (\mathbf{q}_h,\mathbf{w})_K+(u_h,\nabla\cdot\mathbf{w})_K-\langle\hat{u}_h,\mathbf{w}\cdot\mathbf{n}\rangle_{\partial K}&=0,\\ \langle\left(\hat{\mathbf{F}}_h-\epsilon\hat{\mathbf{q}}_h\right)^{+}\cdot\mathbf{n}^{+}+(\hat{\mathbf{F}}_h-\epsilon\hat{\mathbf{q}}_h)^{-}\cdot\mathbf{n}^{-},\mu\rangle_e&=0, \end{aligned} \tag{5}$$

for all $K\in\mathcal{T}_h$, all $e\in\mathcal{E}_h^i$ and all $(v,\mathbf{w},\mu)\in V_h^p\times\mathbf{W}_h^p\times M_h^p(0)$. Owing to the discontinuous nature of the approximation spaces, the integration by parts introduces the so-called numerical fluxes $\hat{\mathbf{F}}_h$ and $\hat{\mathbf{q}}_h$ that represent an approximation to the fluxes at the interfaces and have to satisfy certain properties to render the system well posed.

By summing (5) over all the elements, the problem reads as follows: find $(u_h,\mathbf{q}_h,\hat{u}_h)\in V_h^p\times\mathbf{W}_h^p\times M_h^p(g_D)$ such that

$$\begin{aligned} \langle\left(\hat{\mathbf{F}}_h-\epsilon\hat{\mathbf{q}}_h\right)\cdot\mathbf{n},v\rangle_{\partial\mathcal{T}_h}-(\mathbf{F}(u_h)-\epsilon\mathbf{q}_h,\nabla v)_{\mathcal{T}_h}&=(f,v)_{\mathcal{T}_h}, && \forall v\in V_h^p,\\ (\mathbf{q}_h,\mathbf{w})_{\mathcal{T}_h}+(u_h,\nabla\cdot\mathbf{w})_{\mathcal{T}_h}-\langle\hat{u}_h,\mathbf{w}\cdot\mathbf{n}\rangle_{\partial\mathcal{T}_h}&=0, && \forall\mathbf{w}\in\mathbf{W}_h^p,\\ \langle\left(\hat{\mathbf{F}}_h-\epsilon\hat{\mathbf{q}}_h\right)\cdot\mathbf{n},\mu\rangle_{\partial\mathcal{T}_h}&=0, && \forall\mu\in M_h^p(0), \end{aligned} \tag{6}$$

where the numerical fluxes are given by

$$\hat{\mathbf{F}}_h-\epsilon\hat{\mathbf{q}}_h=\mathbf{F}\left(\hat{u}_h\right)-\epsilon\mathbf{q}_h+\tau\left(u_h,\hat{u}_h\right)\left(u_h-\hat{u}_h\right)\cdot\mathbf{n}. \tag{7}$$

Here, $\tau(u_h,\hat{u}_h)$ is the stabilization parameter. The choice of τ is vital for the system to be well posed. A detailed analysis can be found in [16] and yields the following condition, that requires $\mathbf{F}(u)$ to be a differentiable function of u:

$$\tau>\frac{\epsilon}{l}+\frac{1}{2}\sup\left|\mathbf{F}'(s)\cdot\mathbf{n}\right|, \quad s\in\left[\min\{u_h,\hat{u}_h\},\max\left\{u_h,\hat{u}_h\right\}\right], \tag{8}$$

where l is a characteristic length of the problem and the second term is related to the maximum wave speed across interfaces. For the cases of interest in this paper, a constant value of τ will be chosen big enough to satisfy (8).

2.3. DPG Scheme

2.3.1. Optimal test functions. We will now describe the theory of the optimal test functions first introduced by Demkowicz and Gopalakrishnan in the series of papers [7, 31, 33] for linear convection–diffusion equations. Towards this end, we consider an abstract weak formulation: find $u\in U$ such that

$$b(u,v)=l(v), \quad \forall v\in V, \tag{9}$$

DOI: 10.1002/nme

where U and V are Hilbert spaces, $b(\cdot,\cdot) : U \times V \mapsto \mathbb{R}$ is a continuous bilinear form, and $l(\cdot) \in V'$ is an element of the dual space of V. It is well known that the existence and uniqueness of the solution of this problem is associated with the so-called inf-sup constant γ (Babuska, [34]):

$$\gamma := \inf_{u \in U} \sup_{v \in V} \frac{b(u,v)}{\|u\|_U \|v\|_V}, \tag{10}$$

in particular, if $\gamma > 0$, then problem 9 has a unique solution.

In the finite element context, we replace U with U_h and V with V_h, where U_h and V_h are suitable finite element spaces. As a result, we arrive at the following problem: find $u_h \in U_h$ such that

$$b(u_h, v_h) = l(v_h), \quad \forall v_h \in V_h. \tag{11}$$

The discrete inf-sup constant is then defined as

$$\gamma_h := \inf_{u_h \in U_h} \sup_{v_h \in V_h} \frac{b(u_h,v_h)}{\|u_h\|_U \|v_h\|_V}, \tag{12}$$

and plays exactly the same role as the continuous one (10) in the sense that $\gamma_h > 0$ is required for existence and uniqueness. Our interest resides in using trial spaces U_h with good approximation properties (e.g., polynomials of order p), and computing the tests space V_h so that the stability constant γ_h is maximized. As described in [31], this can be achieved when the test functions are computed using the composition of the operator that defines the problem and the inverse Riesz mapping to yield $T : U_h \mapsto V$. The test space is obtained using the auxiliary problem: find $Te_i \in V$ such that:

$$(Te_i, v)_V = b(e_i, v), \quad \forall v \in V, \tag{13}$$

for each basis function of the trial space e_i ($U_h = \text{span}\{e_i\}$). The discrete test space is then taken as $V_h = \text{span}\{Te_i\}$.

2.3.2. DPG formulation. In practice, the previous step involves the non-trivial task of inverting a continuous operator; hence, some degree of discretization has to be introduced. The approach proposed in [7, 31, 33] relies on a space of candidate test functions $\tilde{V}_h$ based on polynomials of order $p + \Delta p$. This way, the approximate trial-to-test map reads: find $T_h e_i \in \tilde{V}_h$ such that

$$(T_h e_i, v)_V = b(e_i, v), \quad \forall v \in \tilde{V}_h. \tag{14}$$

The biggest obstacle becomes the inversion of the metric of the inner product $(\cdot,\cdot)_V$ for the basis of $\tilde{V}_h$. It is clear that in the case where continuous finite element spaces are used, the continuity of the basis functions across elements generates a metric with a sparsity pattern similar to the original bilinear form; hence, as expensive to solve as the weak formulation itself. However, when no such continuity conditions appear between elements, as in the case of the usual DG spaces, the metric can be efficiently inverted element-wise.

This is the basic approach proposed in [7, 31, 33], where the discontinuous Petrov–Galerkin scheme of Bottasso *et al.* [35] is used in combination with the approximate optimal test functions described here. Finally, note the choice of the space $\tilde{V}_h$ is not unique and several other approaches could be used, e.g. bubbles, provided $\dim(\tilde{V}_h) > \dim(U_h)$. In any case, the use of higher order polynomials seems to be the most cost-effective solution as we would expect the approximate test space to converge towards the optimal one (in the sense of (13)) as Δp is increased.

3. HYBRIDIZED DISCONTINUOUS PETROV–GALERKIN SCHEME

3.1. Nonlinear optimal test functions

We extend the previous theory to nonlinear conservation laws. For that, the theory has to be extended to the case of a nonlinear weak formulation: find $u \in U$ such that

$$a(u, v) = l(v), \quad \forall v \in V, \tag{15}$$

DOI: 10.1002/nme

where the operator $a(\cdot,\cdot) : U \times V \mapsto \mathbb{R}$ is *linear* in v (indeed $a(u,\cdot) \in V'$), but *nonlinear* in u. In this case, the problem can be usually written in residual form as follows: find $u \in U$ such that

$$r(u,v) := a(u,v) - l(v) = 0, \quad \forall v \in V. \tag{16}$$

Note that the operator r is linear in v. The discrete approximation then reads as follows: find $u_h \in U_h$ such that

$$r(u_h,v) = 0, \quad \forall v \in V_h. \tag{17}$$

The test space V_h is constructed as follows.

Following the spirit of the previous section, we will assume that the test functions are not known a priori and will be selected from a certain space $\tilde{V}_h$ that satisfies $\dim\left(\tilde{V}_h\right) > \dim(U_h)$. In particular, one can use a similar approach to compute the test functions by means of the following trial-to-test map:

$$(T_h e_i, v)_V = r'_{u_h}(e_i, v), \quad \forall v \in \tilde{V}_h, \tag{18}$$

where $r'_{u_h}(w,v)$ is the bilinear form induced by the Frechet derivative of $r(u,v)$ with respect to u evaluated at u_h. The discrete test space is then taken as $V_h = \text{span}\{Te_i\}$. It is important to point out that in the nonlinear case, the test space V_h *depends* on u_h because $r'_{u_h}(w,v)$ depends on u_h. Therefore, the test functions can not be computed in advance as in the linear case.

We would like to point out a main difference between the proposed approach and the original DPG scheme introduced in [36] for nonlinear problems. Our approach applies the concept of optimal test functions to the nonlinear residual $r(u,v)$, whereas the DPG scheme applies this concept to the linearized problem arising from the Newton iteration on the nonlinear system. In other words, our approach is *Petrov–Galerkin projection followed by linearization*, whereas the original DPG scheme is *linearization followed by Petrov–Galerkin projection*. Hence, the test functions generated by the original DPG approach are optimal with respect to the linearized problem, but *not* to the original nonlinear problem. Owing to this lack of consistency in the Jacobian, the DPG scheme suffers from slow convergence (as reported in [36]) in some cases.

Proposition 1

The nonlinear weak formulation (17) solved using the approximate optimal test space from (18) is equivalent to the solution of the following problem:

$$u_h = \underset{w_h \in U_h}{\arg\inf} \sup_{v \in \tilde{V}_h} \frac{r(w_h,v)}{\|v\|_V}. \tag{19}$$

Proof

Let $\mathbf{v} \in \mathbb{R}^n$ denote the vector of coefficients of an element $v_h \in \tilde{V}_h$. Similarly, let $\mathbf{u} \in \mathbb{R}^m$ ($\mathbf{w} \in \mathbb{R}^m$) denote the vector of coefficients of an element $u_h \in U_h$ ($w_h \in U_h$, respectively). We can rewrite the min–max statement as follows:

$$\mathbf{u} = \underset{\mathbf{w} \in \mathbb{R}^m}{\arg\min} \max_{\mathbf{v} \in \mathbb{R}^n} \frac{\mathbf{v}^T \mathbf{r}(\mathbf{w})}{\sqrt{\mathbf{v}^T X_V \mathbf{v}}}, \tag{20}$$

where the vector $\mathbf{r}(\mathbf{w})$ represents the usual duality pairing $r = r(w_h,v)$ against each element of the basis for $\tilde{V}_h$, and X_V represents the metric of the space $\tilde{V}_h$. The inner maximization statement can be solved exactly using the first order optimality condition:

$$\frac{\mathbf{r}(\mathbf{w})}{\sqrt{\mathbf{v}^T X_V \mathbf{v}}} - \frac{\mathbf{v}^T \mathbf{r}(\mathbf{w})}{\left(\mathbf{v}^T X_V \mathbf{v}\right)^{3/2}} X_V \mathbf{v} = 0 \Rightarrow \mathbf{v} = \left(\mathbf{v}^T X_V \mathbf{v}\right) \frac{X_V^{-1} \mathbf{r}(\mathbf{w})}{\mathbf{v}^T \mathbf{r}(\mathbf{w})}. \tag{21}$$

The solution to the problem is unique (up to a scaling of $\mathbf{v}$) as one can prove by rewriting it with the additional constraint $\|\mathbf{v}\|_V = 1$, in which case, the objective is linear with convex constraints. Inserting (21) into (20), we obtain the following:

$$\mathbf{u} = \underset{\mathbf{w} \in \mathbb{R}^m}{\arg\min} \sqrt{\mathbf{r}(\mathbf{w})^T X_V^{-1} \mathbf{r}(\mathbf{w})}. \tag{22}$$

DOI: 10.1002/nme

The first order optimality condition for (22) yields

$$\frac{1}{2\sqrt{\mathbf{r}(\mathbf{u})^T X_V^{-1}\mathbf{r}(\mathbf{u})}}\frac{\partial \mathbf{r}(\mathbf{u})}{\partial \mathbf{u}}^T X_V^{-1}\mathbf{r}(\mathbf{u}) = 0 \Rightarrow \frac{\partial \mathbf{r}(\mathbf{u})}{\partial \mathbf{u}}^T X_V^{-1}\mathbf{r}(\mathbf{u}) = 0, \tag{23}$$

to form a system of nonlinear equations for $\mathbf{u}$.

To show the equivalence, we take (17) and write it in discrete form as follows: find $\mathbf{u} \in \mathbb{R}^m$ such that

$$\mathbf{v}^T\mathbf{r}(\mathbf{u}) = 0, \quad \forall \mathbf{v} \in \tilde{V}_h. \tag{24}$$

Here, $\tilde{V}_h = \text{span}\{\mathbf{t}_i, 1 \leqslant i \leqslant m\}$ is obtained using the mapping defined in (18): find $\mathbf{t}_i \in \mathbb{R}^n$ such that

$$\mathbf{t}_i = X_V^{-1}\frac{\partial \mathbf{r}(\mathbf{u})}{\partial \mathbf{u}}\mathbf{e}_i, \tag{25}$$

where $\mathbf{e}_i$ denotes the vector of coefficients of $e_i \in U_h$. Combining (24) and (25), we obtain

$$\mathbf{w}^T\frac{\partial \mathbf{r}(\mathbf{u})}{\partial \mathbf{u}}^T X_V^{-1}\mathbf{r}(\mathbf{u}) = 0, \quad \forall \mathbf{w} \in \mathbb{R}^m. \tag{26}$$

The desired result follows directly from (22) and (26). □

Remark 1

The minimization statement (22) can also be written as follows:

$$\mathbf{u} = \underset{\mathbf{w}\in\mathbb{R}^m}{\arg\min}\ \frac{1}{2}\mathbf{r}(\mathbf{w})^T X_V^{-1}\mathbf{r}(\mathbf{w}), \tag{27}$$

which represents a very general and flexible point of departure. In particular, it can be easily constrained to guarantee certain properties such as local conservation.

3.2. HDPG formulation

A limitation of the DPG scheme presented in [31,33,36] is the fact that the DPG system is formed by multiplying the transposed Jacobian by itself, hence, yielding a denser system to solve. Furthermore, the scheme is not conservative in the sense that the constant mode *is not* guaranteed to belong to the optimal test space. Despite these limitations, the scheme has certain good properties. For example, it is clear that in the case of linear equations, the final system to solve for is symmetric positive-definite and hence can be efficiently solved using well-developed iterative algorithms such as the conjugate gradients method. Moreover, the DPG scheme gives optimal $p + 1$ error convergence for the well-known case proposed by Peterson [32], whereas other DG methods can only achieve $p + 1/2$ error convergence.

Our goal in this paper is to propose a new scheme that

- preserves the efficiency of the HDG scheme;
- incorporates the excellent stability of the DPG scheme; and
- enforces conservative solutions.

To define the HDPG scheme, we first introduce the residuals associated with the HDG formulation (6)–(7) as

$$\begin{aligned} r_u^K(a_h, \mathbf{b}_h, v; \hat{a}_h) &= \langle \mathbf{F}(\hat{a}_h)\cdot\mathbf{n} - \epsilon\mathbf{b}_h\cdot\mathbf{n} + \tau(a_h, \hat{a}_h)(a_h - \hat{a}_h), v\rangle_{\partial K} \\ &\quad - (\mathbf{F}(a_h) - \epsilon\mathbf{b}_h, \nabla v)_K - (f, v)_K, \\ r_{\mathbf{q}}^K(a_h, \mathbf{b}_h, \mathbf{w}; \hat{a}_h) &= (\mathbf{b}_h, \mathbf{w})_K + (a_h, \nabla\cdot\mathbf{w})_K - \langle \hat{a}_h, \mathbf{w}\cdot\mathbf{n}\rangle_{\partial K}, \\ r_{\hat{u}}(a_h, \mathbf{b}_h, \hat{a}_h, \mu) &= \langle \mathbf{F}(\hat{a}_h)\cdot\mathbf{n} - \epsilon\mathbf{b}_h\cdot\mathbf{n} + \tau(a_h, \hat{a}_h)(a_h - \hat{a}_h), \mu\rangle_{\partial\mathcal{T}_h}, \end{aligned} \tag{28}$$

where we have inserted the definition of the numerical fluxes (7) into (6).

DOI: 10.1002/nme

The HDPG formulation then reads as follows: find $\hat{u}_h \in M_h^p(g_D)$ such that

$$r_{\hat{u}}(u_h, \mathbf{q}_h, \hat{u}_h, \mu) = 0, \quad \forall \mu \in M_h^p(0), \tag{29}$$

where $(u_h(\hat{u}_h), \mathbf{q}_h(\hat{u}_h))|_K \in \mathcal{P}^p(K) \times (\mathcal{P}^p(K))^d$ satisfies the following:

$$\begin{aligned}(u_h(\hat{u}_h), \mathbf{q}_h(\hat{u}_h))|_K = & \underset{(a_h,\mathbf{b}_h)\in\mathcal{P}^p(K)\times(\mathcal{P}^p(K))^d}{\arg\inf} \left(\max_{v\in\mathcal{P}^{p+\Delta p}(K)} \frac{r_u^K(a_h, \mathbf{b}_h, v; \hat{u}_h)}{\|v\|_V} \right), \\ & \text{s.t.} \quad r_{\mathbf{q}}^K(a_h, \mathbf{b}_h, \mathbf{w}; \hat{u}_h) = 0, \qquad \forall \mathbf{w} \in (\mathcal{P}^p(K))^d \\ & \text{s.t.} \quad r_u^K(a_h, \mathbf{b}_h, 1; \hat{u}_h) = 0,\end{aligned} \tag{30}$$

for all $K \in \mathcal{T}_h$. The system (29)-(30) completes the definition of the HDPG scheme.

Some remarks about the HDPG scheme are in order. The first equation (29) weakly enforces the continuity of the normal component of the numerical fluxes across elemental interfaces whereas the second equation (30) defines $(u_h, \mathbf{q}_h)$ as a function of $\hat{u}_h$ locally on every element. Therefore, (29) is called the global problem, which gives rise to a nonlinear algebraic system for the DOFs of $\hat{u}_h$ only. As for the local problem (30), we apply the optimal test function approach to the conservation law only and strongly impose an equality constraint on the kinematic relationship by requiring that $r_{\mathbf{q}}^K(a_h, \mathbf{b}_h, \mathbf{w}; \hat{u}_h) = 0$, $\forall \mathbf{w} \in (\mathcal{P}^p(K))^d$. In addition, we explicitly enforce the conservation of the HDPG scheme at the local level by requiring the integration against a constant test function $v \in \mathcal{P}^0(K)$ to be strongly satisfied.

3.3. Solution procedure

Our focus now is the solution of system (29)–(30). We shall seek an iterative algorithm that takes advantage of the definition of the local problem to solve for the globally coupled DOFs of $\hat{u}_h$ only. In particular, we apply the Newton–Raphson method to the global problem (29), thereby obtaining a linear system at each iteration. This, in turn, requires us to solve the local problem (30) for $(u_h(\hat{u}_h), \mathbf{q}_h(\hat{u}_h))$ and the associated sensitivities.

3.3.1. Algebraic system. Some further notation is required before introducing the iteration. We denote by $(\mathbf{u}, \mathbf{Q}, \hat{\mathbf{u}})$ the vectors of coefficients associated with the functions $(u_h, \mathbf{q}_h, \hat{u}_h) \in V_h^p \times \mathbf{W}_h^p \times M_h^p(g_D)$. We also denote by $\mathbf{r}_u^K$, $\mathbf{r}_{\mathbf{q}}^K$ and $\mathbf{r}_{\hat{u}}$ the residual vectors associated with r_u^K, $r_{\mathbf{q}}^K$ and $r_{\hat{u}}$, respectively (see Appendix for description). This allows us to rewrite the HDPG formulation (29)–(30) as an algebraic system: find $\hat{\mathbf{u}} \in \mathbb{R}^n$ such that

$$\mathbf{r}_{\hat{u}}(\mathbf{u}, \mathbf{Q}, \hat{\mathbf{u}}) = 0 \tag{31}$$

where $(\mathbf{u}(\hat{\mathbf{u}}), \mathbf{Q}(\hat{\mathbf{u}}))|_K \in \mathbb{R}^m \times \mathbb{R}^{m\times d}$ satisfies

$$\begin{aligned}(\mathbf{u}, \mathbf{Q})|_K = & \underset{(\mathbf{a},\mathbf{B})\in\mathbb{R}^m\times\mathbb{R}^{m\times d}}{\arg\min} \frac{1}{2} \mathbf{r}_u^K(\mathbf{a}, \mathbf{B}; \hat{\mathbf{u}})^T X_V^{-1} \mathbf{r}_u^K(\mathbf{a}, \mathbf{B}; \hat{\mathbf{u}}), \\ & \text{s.t.} \quad \mathbf{r}_{\mathbf{q}}^K(\mathbf{a}, \mathbf{B}; \hat{\mathbf{u}}) = 0, \\ & \text{s.t.} \quad \mathbf{c}^T \mathbf{r}_u^K(\mathbf{a}, \mathbf{B}; \hat{\mathbf{u}}) = 0,\end{aligned} \tag{32}$$

for all $K \in \mathcal{T}_h$, where $\mathbf{c}$ is the vector of coefficients for the constant mode represented in the local basis for $\mathcal{P}^{p+\Delta p}(K)$.

3.3.2. Global solver. As indicated earlier, in order to solve the system (31), we will use a Newton–Raphson iteration. For this, given a current iterate $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}, \bar{\hat{\mathbf{u}}}\right)$ that satisfies the local problem $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}\right)\Big|_K = \left(\mathbf{u}\left(\bar{\hat{\mathbf{u}}}\right), \mathbf{Q}\left(\bar{\hat{\mathbf{u}}}\right)\right)\Big|_K$, we seek updates to the solution $\delta\hat{\mathbf{u}}$ by solving the following system:

$$\left(\frac{\partial \mathbf{r}_{\hat{u}}}{\partial \hat{\mathbf{u}}} + \frac{\partial \mathbf{r}_{\hat{u}}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \hat{\mathbf{u}}} + \frac{\partial \mathbf{r}_{\hat{u}}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \hat{\mathbf{u}}} \right) \delta\hat{\mathbf{u}} = -\mathbf{r}_{\hat{u}}, \tag{33}$$

DOI: 10.1002/nme

where we have used the chain rule to obtain the fully linearized system with respect to $\hat{\mathbf{u}}$. Here, all the terms involved such as residuals or Jacobians have to be computed using the current iterate $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}, \bar{\hat{\mathbf{u}}}\right)$. More details on the computation of these can be found in Appendix A. The iteration is repeated in the usual Newton–Raphson fashion until the update gets below a certain tolerance ($\|\delta\hat{\mathbf{u}}\| < tol$). To avoid the divergence of the iteration, a damped Newton strategy is implemented that limits the stepsize by $\alpha \leqslant 1$ to guarantee $\|\mathbf{r}_{\hat{u}}(\hat{\mathbf{u}})\| > \|\mathbf{r}_{\hat{u}}(\hat{\mathbf{u}} + \delta\hat{\mathbf{u}})\|$. The choice of α follows the usual bisection rule. After each iteration, we use $\alpha\delta\hat{\mathbf{u}}$ to update the solution. A basic algorithmic description can be found later.

In theory, we would expect the Newton–Raphson iteration to present quadratic convergence once the iterate is close to the solution. For this, the Jacobian matrix has to be properly computed, this meaning, we require the dependencies $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}\right)\Big|_K = \left(\mathbf{u}\left(\bar{\hat{\mathbf{u}}}\right), \mathbf{Q}\left(\bar{\hat{\mathbf{u}}}\right)\right)\Big|_K$ and the sensitivities $\partial\mathbf{u}/\partial\hat{\mathbf{u}}$ and $\partial\mathbf{Q}/\partial\hat{\mathbf{u}}$ to be solved exactly (up to numerical error).

3.3.3. Exact local solver. We now describe an iterative scheme to solve the local problem (32) and obtain the desired dependencies $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}\right)\Big|_K = \left(\mathbf{u}\left(\bar{\hat{\mathbf{u}}}\right), \mathbf{Q}\left(\bar{\hat{\mathbf{u}}}\right)\right)\Big|_K$. For this, two different approaches will be combined using a simple switch. The first approach is to linearize the different residuals that enter the problem before taking any minimization step. By this, we mean, from a current iterate $\bar{\mathbf{Z}} = \left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}\right)$, we look for updates $\delta\mathbf{Z} = (\delta\mathbf{u}, \delta\mathbf{Q})$ that solve the following linearized problem: given $\hat{u}_h \in M_h^p$, find $\delta\mathbf{Z} \in \mathbb{R}^m \times \mathbb{R}^{m\times d}$ such that

$$\begin{aligned}
\delta\mathbf{Z} = \underset{\delta\mathbf{C}\in\mathbb{R}^m\times\mathbb{R}^{m\times d}}{\arg\min}\ & \frac{1}{2}\left(\mathbf{r}_u^K + \frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\delta\mathbf{C}\right)^T X_V^{-1}\left(\mathbf{r}_u^K + \frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\delta\mathbf{C}\right), \\
\text{s.t}\quad & \mathbf{r}_\mathbf{q}^K + \frac{\partial\mathbf{r}_\mathbf{q}^K}{\partial\mathbf{Z}}\delta\mathbf{C} = 0, \\
\text{s.t}\quad & \mathbf{c}^T\left(\mathbf{r}_u^K + \frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\delta\mathbf{C}\right) = 0,
\end{aligned} \tag{34}$$

where the different terms that appear in this system are described in Appendix A. Notice that the variable $\bar{\mathbf{Z}}$ has been introduced to simplify the notation. The minimization now follows by deriving the Karush–Kuhn–Tucker conditions for $\delta\mathbf{Z}$:

$$\begin{bmatrix}
\left(\frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\right)^T X_V^{-1}\left(\frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\right) & \frac{\partial\mathbf{r}_\mathbf{q}^K}{\partial\mathbf{Z}}^T & \left(\frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\right)^T\mathbf{c} \\
\frac{\partial\mathbf{r}_\mathbf{q}^K}{\partial\mathbf{Z}} & 0 & 0 \\
\mathbf{c}^T\left(\frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\right) & 0 & 0
\end{bmatrix}
\begin{Bmatrix} \delta\mathbf{Z} \\ \mu \\ \lambda \end{Bmatrix}
= -\begin{Bmatrix}
\left(\frac{\partial\mathbf{r}_u^K}{\partial\mathbf{Z}}\right)^T X_V^{-1}\mathbf{r}_u^K \\
\mathbf{r}_\mathbf{q}^K \\
\mathbf{c}^T\mathbf{r}_u^K
\end{Bmatrix}. \tag{35}$$

Notice how $\hat{u}_h$ is fixed and just acts as a parameter. Also notice that λ plays the role of a Lagrange multiplier for the conservation whereas μ represents a set of Lagrange multipliers for the kinematic variables. This approach represents a constrained version of the well-known Gauss–Newton method (GN) for nonlinear least squares problems. The GN method is very robust in the sense that at each iteration, the computed update is a feasible descent direction of the problem. Also, it is well known that the convergence of this scheme can approach a quadratic rate, though this strongly depends on the value of the objective function at convergence. In particular, when it is non-zero, some terms are missing in the linearization that may slow or even prevent convergence.

DOI: 10.1002/nme

The second approach to solve (32) is to depart from the first order optimality conditions directly:

$$\begin{aligned}\left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T X_V^{-1}\mathbf{r}_u^K + \mu^T\left(\frac{\partial \mathbf{r}_\mathbf{q}^K}{\partial \mathbf{Z}}\right) + \lambda \mathbf{c}^T\left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right) &= 0,\\ \mathbf{r}_\mathbf{q}^K &= 0,\\ \mathbf{c}^T\mathbf{r}_u^K &= 0,\end{aligned} \tag{36}$$

and apply the Newton–Raphson iteration to it. This is also known as sequential quadratic programming (SQP) and yields the following system to solve

$$\begin{bmatrix} \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T X_V^{-1}\left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right) + \left(\frac{\partial^2 \mathbf{r}_u^K}{\partial \mathbf{Z}^2}\right)\cdot\left(X_V^{-1}\mathbf{r}_u^K + \lambda\mathbf{c}^T\right) & \frac{\partial \mathbf{r}_\mathbf{q}^K}{\partial \mathbf{Z}}^T & \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T\mathbf{c} \\ \frac{\partial \mathbf{r}_\mathbf{q}^K}{\partial \mathbf{Z}} & 0 & 0 \\ \mathbf{c}^T\left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right) & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta\mathbf{Z} \\ \mu \\ \lambda \end{Bmatrix} = -\begin{Bmatrix} \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T X_V^{-1}\mathbf{r}_u^K \\ \mathbf{r}_\mathbf{q}^K \\ \mathbf{c}^T\mathbf{r}_u^K \end{Bmatrix}, \tag{37}$$

where the vector of unknowns contains the update $\delta\mathbf{Z}$ and the approximate Lagrange multipliers. This way, the difference between (35) and (37) is only the second derivative terms that appear in the latter one. These derivatives account for the variation of the test space with respect to the solution and play a vital role in the convergence. Indeed, this might be the reason why the DPG scheme of [36] does not achieve quadratic convergence even once close to the solution (see [36] pp. 16). Notice also the second derivatives of $\mathbf{r}_\mathbf{q}^K$ have not been included since this residual is linear in all the arguments.

One of the most important properties of the SQP iteration is the rate of convergence, which is locally quadratic when close enough to the solution. Its main disadvantage is the cost associated with constructing the second derivatives. Hence, we propose to combine the Gauss–Newton with the SQP in order to take advantage of their respective properties. In particular, we propose a switch of schemes based on the size of the update $\|\delta\mathbf{Z}\|$. More sophisticated switches might be devised, but this works fine for the cases of interest here. The local solver iteration is described in the algorithm below.

Before moving on to the next step in the HDPG scheme, we would like to comment on the local solver just presented. In particular, we would like to point out that the problem to solve (32) is of the constrained nonlinear least squares kind. These problems have long been studied in the field of optimization; hence, very efficient algorithms exist to solve them (e.g., Levenberg–Marquardt algorithm). We did not explore other options than the two presented here, which proved to be very efficient at solving all types of elements, even with strongly under-resolved shocks in them as we will see in the Results section.

From an implementation point of view, the local solver cost can be reduced if the equations for the kinematic relationships are inverted before the minimization takes place. Namely, one can see from (28) that the relationship between these variables is linear, and more importantly, can be inverted locally to obtain $\mathbf{q}_h = \mathbf{q}_h(u_h, \hat{u}_h)$. This way, the minimization statement only involves u_h and $\hat{u}_h$, and the Lagrange multipliers μ can be dispensed with.

3.3.4. Local problem sensitivities $\frac{\partial u_h}{\partial \hat{u}_h}$, $\frac{\partial \mathbf{q}_h}{\partial \hat{u}_h}$. Once the local problem is solved, the sensitivities of the local solution to the boundary data, required to formulate system (33) have to be computed. To do so, we can use the first order optimality conditions, (36), and use the implicit function theorem

DOI: 10.1002/nme

to obtain the sensitivities. For this, let $\bar{\hat{\mathbf{U}}} \in \mathbb{R}^{t \times f}$ be the coefficients of $\hat{u}_h$ on ∂K, where f is the number of faces. The system to solve for the sensitivities reads as follows:

$$\begin{bmatrix} \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T X_V^{-1} \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right) + \left(\frac{\partial^2 \mathbf{r}_u^K}{\partial \mathbf{Z}^2}\right) \cdot \left(X_V^{-1} \mathbf{r}_u^K + \lambda \mathbf{c}^T\right) & \frac{\partial \mathbf{r}_\mathbf{q}^K}{\partial \mathbf{Z}}^T & \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T \mathbf{c} \\ \frac{\partial \mathbf{r}_\mathbf{q}^K}{\partial \mathbf{Z}} & 0 & 0 \\ \mathbf{c}^T \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right) & 0 & 0 \end{bmatrix} \begin{Bmatrix} \frac{\partial \mathbf{Z}}{\partial \bar{\hat{\mathbf{U}}}} \\ \frac{\partial}{\partial \bar{\hat{\mathbf{U}}}} \\ \frac{\partial \lambda}{\partial \bar{\hat{\mathbf{U}}}} \end{Bmatrix} = - \begin{Bmatrix} \left(\frac{\partial \mathbf{r}_u^K}{\partial \mathbf{Z}}\right)^T X_V^{-1} \left(\frac{\partial \mathbf{r}_u^K}{\partial \bar{\hat{\mathbf{U}}}}\right) + \left(\frac{\partial^2 \mathbf{r}_u^K}{\partial \mathbf{Z} \partial \bar{\hat{\mathbf{U}}}}\right) \cdot \left(X_V^{-1} \mathbf{r}_u^K + \lambda \mathbf{c}^T\right) \\ \frac{\partial \mathbf{r}_\mathbf{q}^K}{\partial \bar{\hat{\mathbf{U}}}} \\ \mathbf{c}^T \frac{\partial \mathbf{r}_u^K}{\partial \bar{\hat{\mathbf{U}}}} \end{Bmatrix}, \tag{38}$$

where the different terms that appear are properly described in Appendix A. Notice that the only condition for this to hold is that the Jacobian of the implicit mapping with respect to the variable we want to compute sensitivities for, has to be invertible. Notice this is just the matrix on the left of (38), that coincides with the matrix from the SQP iteration, and because of this, can be reused provided the convergence tolerance for the local solver is small enough to make the changes in the solution negligible. Our test indicates this strategy does not affect the overall convergence of the scheme.

Algorithm 1 Hybridized discontinuous Petrov–Galerkin (HDPG) scheme

Given: $\bar{\mathbf{u}}, \bar{\mathbf{Q}}, \bar{\hat{\mathbf{u}}}, \|\delta \hat{\mathbf{u}}_h\| = 10 \cdot tol$
while $\|\delta \hat{\mathbf{u}}_h\| \geq tol$ **do**
  for $j = 1 \rightarrow N_{ele}$ **do**
    Extract $\bar{\hat{\mathbf{U}}}$ from $\bar{\hat{\mathbf{u}}}$
    $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}, \frac{\partial \mathbf{u}}{\partial \bar{\hat{\mathbf{U}}}}, \frac{\partial \mathbf{Q}}{\partial \bar{\hat{\mathbf{U}}}}\right)$ = HDPG Local Solver $\left(\bar{\mathbf{u}}, \bar{\mathbf{Q}}, \bar{\hat{\mathbf{U}}}\right)$
  end for
  Solve the Global problem (Equation 33) $\rightarrow \delta \hat{\mathbf{u}}$
  Find damping α using bisection
  $\bar{\hat{\mathbf{u}}} \leftarrow \bar{\hat{\mathbf{u}}} + \alpha \delta \hat{\mathbf{u}}$
  $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{u}} + \alpha \frac{\partial \mathbf{u}}{\partial \hat{\mathbf{u}}} \delta \hat{\mathbf{u}}$, $\bar{\mathbf{Q}} \leftarrow \bar{\mathbf{Q}} + \alpha \frac{\partial \mathbf{Q}}{\partial \hat{\mathbf{u}}} \delta \hat{\mathbf{u}}$
end while

Algorithm 2 HDPG Local Solver

Given initial conditions for: $\mathbf{u}, \mathbf{Q}$
Given boundary data: $\hat{\mathbf{U}}$
while $\|\delta \mathbf{u}\| \geq tol$ **do**
  if $\|\delta \mathbf{u}\| \geq \mathcal{O}(1)$ **then**
    Solve the GN system (Equation 35) $\rightarrow \delta \mathbf{u}, \delta \mathbf{Q}$
  else
    Solve the SQP system (Equation 37) $\rightarrow \delta \mathbf{u}, \delta \mathbf{Q}$
  end if
  $\mathbf{u} \leftarrow \mathbf{u} + \delta \mathbf{u}$, $\mathbf{Q} \leftarrow \mathbf{Q} + \delta \mathbf{Q}$
end while
Compute sensitivities (Equation 38) $\rightarrow \frac{\partial \mathbf{u}}{\partial \bar{\hat{\mathbf{U}}}}, \frac{\partial \mathbf{Q}}{\partial \bar{\hat{\mathbf{U}}}}$
Return: $\mathbf{u}, \mathbf{Q}, \frac{\partial \mathbf{u}}{\partial \bar{\hat{\mathbf{U}}}}, \frac{\partial \mathbf{Q}}{\partial \bar{\hat{\mathbf{U}}}}$

DOI: 10.1002/nme

3.4. Extension to time dependent problems

Next, we would like to extend the HDPG scheme to an unsteady convection–diffusion problem defined by the following:

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) - \nabla \cdot (\epsilon \mathbf{q}) &= f, \quad \text{in } \Omega, \\ \mathbf{q} - \nabla u &= 0, \quad \text{in } \Omega, \\ u &= g_D, \quad \text{on } \partial\Omega. \end{aligned} \tag{39}$$

In particular, we introduce the usual polynomial spaces for the solution; however, we assume this solution is parameterized by the time t. Namely, we seek $(u_h(t), \mathbf{q}_h(t), \hat{u}_h(t)) \in V_h^p \times \mathbf{W}_h^p \times M_h^p(g_D)$. We can derive the weak formulation residuals by using integration by parts as follows:

$$\begin{aligned} r_{uK}(u_h, \mathbf{q}_h, v; \hat{u}_h) &= \left(\frac{\partial u_h}{\partial t}, v\right)_K + \langle \mathbf{F}(\hat{u}_h)\cdot\mathbf{n} + \epsilon\mathbf{q}_h\cdot\mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), v\rangle_{\partial K} \\ &\quad - (\mathbf{F}(u_h) - \epsilon\mathbf{q}_h, \nabla v)_K - (f, v)_K, \\ r_{\mathbf{q}K}(u_h, \mathbf{q}_h, \mathbf{w}; \hat{u}_h) &= (\mathbf{q}_h, \mathbf{w})_K + (u_h, \nabla\cdot\mathbf{w})_K - \langle \hat{u}_h, \mathbf{w}\cdot\mathbf{n}\rangle_{\partial K}, \\ r_{\hat{u}}(u_h, \mathbf{q}_h, \hat{u}_h, \mu) &= \langle \mathbf{F}(\hat{u}_h)\cdot\mathbf{n} + \epsilon\mathbf{q}_h\cdot\mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), \mu\rangle_{\partial\mathcal{T}_h}. \end{aligned} \tag{40}$$

In this paper, we will follow a method of lines approach in which the solution is discretized in time using standard ODE formulations and introduced in the definition of the residuals. The differential-algebraic nature of the residuals (notice only u_h presents time derivatives) represents an obstacle to the application of certain time stepping schemes; however, it fits naturally in the framework of implicit methods like the backwards difference formulae (BDF) or the diagonally implicit Runge–Kutta schemes (DIRK). As an example, we will describe here the BDF1 (Backward Euler) implementation. For this, we assume at the current time step s, the derivatives can be approximated by the formula:

$$\partial u_h/\partial t|_i \approx (u_h^s - u_h^{s-1})/\Delta t, \tag{41}$$

and the rest of the terms in the equations are computed at time s, too. The residuals then read as follows:

$$\begin{aligned} r_{uK}(u_h, \mathbf{q}_h, v; \hat{u}_h) &= \left(\frac{u_h}{\Delta t}, v\right)_K + \langle \mathbf{F}(\hat{u}_h)\cdot\mathbf{n} + \epsilon\mathbf{q}_h\cdot\mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), v\rangle_{\partial K} \\ &\quad - (\mathbf{F}(u_h) - \epsilon\mathbf{q}_h, \nabla v)_K - \left(f + \frac{u_h^{s-1}}{\Delta t}, v\right)_K, \\ r_{\mathbf{q}K}(u_h, \mathbf{q}_h, \mathbf{w}; \hat{u}_h) &= (\mathbf{q}_h, \mathbf{w})_K + (u_h, \nabla\cdot\mathbf{w})_K - \langle \hat{u}_h, \mathbf{w}\cdot\mathbf{n}\rangle_{\partial K}, \\ r_{\hat{u}}(u_h, \mathbf{q}_h, \hat{u}_h, \mu) &= \langle \mathbf{F}(\hat{u}_h)\cdot\mathbf{n} + \epsilon\mathbf{q}_h\cdot\mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), \mu\rangle_{\partial\mathcal{T}_h}, \end{aligned} \tag{42}$$

where the superscript s has been omitted for clarity. Notice in this definition, u_h^{s-1} is just data from the previous time step and can be grouped with the source term f. In order to apply the HDPG scheme to the unsteady problem, we just have to follow the algorithm described in Sections 3.2–3.3 using (42) to define the residuals.

4. NUMERICAL EXPERIMENTS

In this section, we present some results for HDPG compared against HDG, both for the same polynomial order p. In every case, the tests space is enriched up to the point where no difference is noticed from increasing Δp. In some cases, we will focus our attention on pure convective operators for which the HDPG formulation presented above is still valid, however, one can save computation by dispensing the kinematic variables $\mathbf{q}_h$.

DOI: 10.1002/nme

4.1. *One-dimensional linear convection problem*

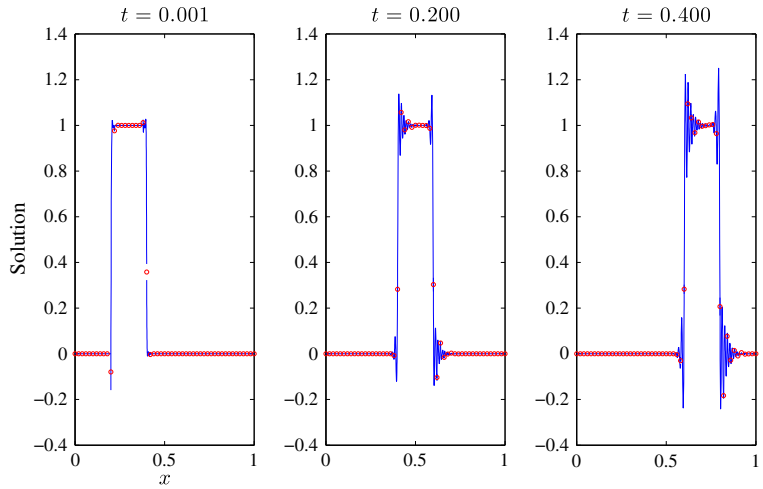
We first present the HDPG scheme applied to a linear convection problem:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, \quad \text{in } \Omega \in [0,1], \tag{43}$$

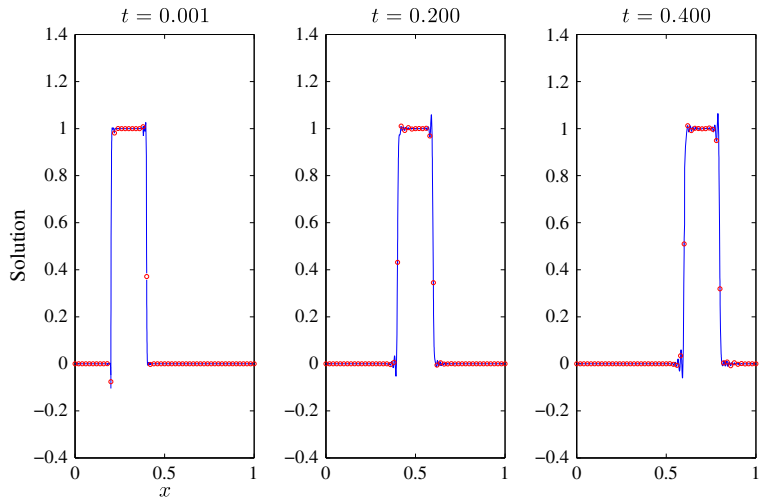
with initial condition

$$u(x,0) = \begin{cases} 1 & \text{if } x \in [0.2, 0.4] \\ 0 & \text{otherwise,} \end{cases} \tag{44}$$

and homogeneous Dirichlet conditions on the boundaries. The initial condition for u_h is interpolated from the analytical one and generates some initial oscillation that we will propagate down in time. The results for both HDG and HDPG using 50 elements of order $p = 5$ are included in Figure 1. The time stepping was carried out using a backwards Euler scheme with $\Delta t = 10^{-3}$. As we can see, the HDPG scheme produces less oscillatory solutions than the HDG scheme.



(a) HDG, $p = 5$



(b) HDPG, $p = 5$, $\Delta p = 5$

Figure 1. Linear convection of a hat function using (a) hybridizable discontinuous Galerkin (HDG) and (b) hybridized discontinuous Petrov–Galerkin (HDPG). In this case, $p = 5$, $\Delta p = 5$ and a backward Euler formula with $\Delta t = 10^{-3}$ is used.

DOI: 10.1002/nme

4.1.1. One-dimensional viscous Burgers' problem. Next, we present results for the Burgers' equation in 1D in cases where smooth initial conditions give rise to discontinuities. The equation to solve reads as follows:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = \epsilon \frac{\partial^2 u}{\partial x^2}, \quad \text{in } \Omega \in [0,1], \tag{45}$$

where the artificial viscosity used will be measured in terms of the so-called cell Peclet number: $Pe|_{\text{cell}} = (h/p)(c/\epsilon)$ that measures the ratio between resolution of the scheme and characteristic length of diffusion layers. In this formula, h represents a characteristic size of the element, and c represents a characteristic propagation velocity, which is equal to $|u|$ for Burgers' equation.

The first case we presented corresponds to the situation where an initial sinusoidal profile steepens into a stationary shock wave. For this case, the initial condition is set to be $u(x,0) = \sin(2\pi x)$ with homogeneous Dirichlet boundary conditions, and the viscosity is set to zero ($Pe|_{\text{cell}} = \infty$). The results for both HDG and HDPG using 25 elements of order $p = 3$ are shown in Figure 2. To time march, a BDF3 scheme with $\Delta t = 10^{-2}$ was used. The results indicate how the good stabilization properties of HDPG carry on to this nonlinear case as oscillation is strongly reduced in the vicinity of the shock.

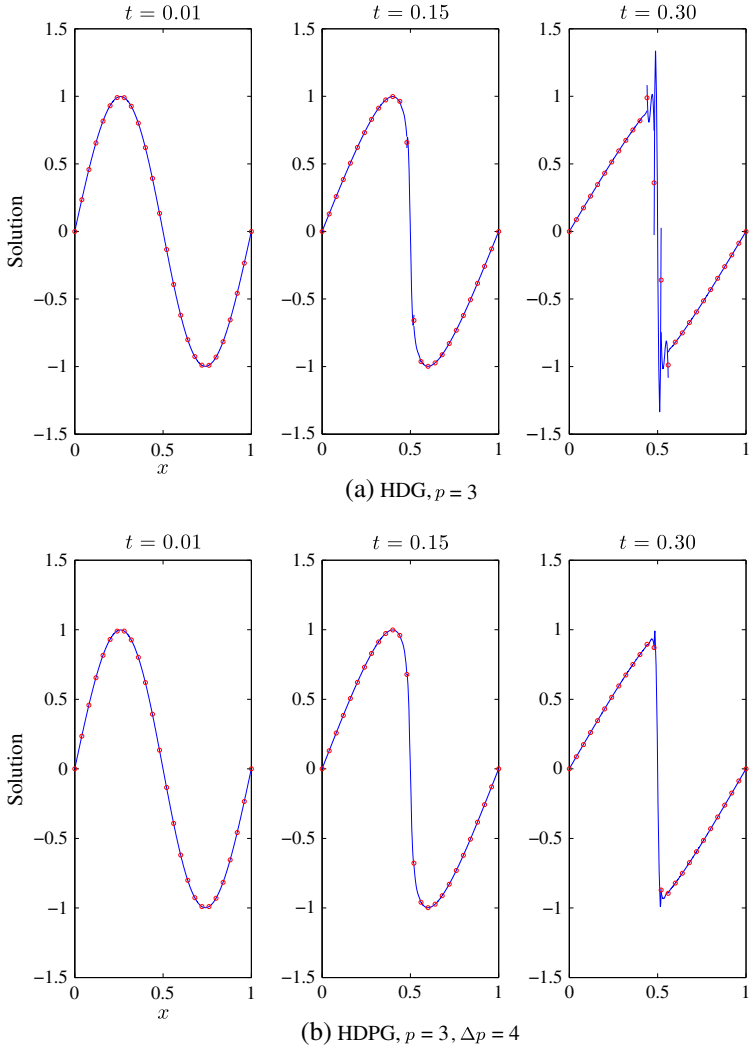


Figure 2. Burgers' equation solution for the case where a steady shock forms using (a) HDG and (b) HDPG. In this case, $p = 3$, $\Delta p = 4$ and a BDF3 scheme with $\Delta t = 10^{-2}$ is used.

DOI: 10.1002/nme

To further confirm this, we apply HDPG to the same equation with an initial condition consisting of a smoothed version of the hat function

$$u(x,0) = \begin{cases} 1 & \text{if } x \in [0.2, 0.5] \\ 0 & \text{otherwise,} \end{cases} \tag{46}$$

and homogeneous Dirichlet boundary conditions. This setting generates both a moving shock and an expansion fan that is integrated in time using a BDF3 scheme with $\Delta t = 10^{-2}$. For this case, 25 elements of order $p = 3$ are used, combined with constant viscosity ϵ such that $Pe|_{\text{cell}} = 10$. The results, included in Figure 3, show the evolution of the shock and fan when HDG and HDPG are used. As we can see, while both schemes propagate the shock at the right speed, thanks to being conservative, the HDPG solution is significantly less oscillatory.

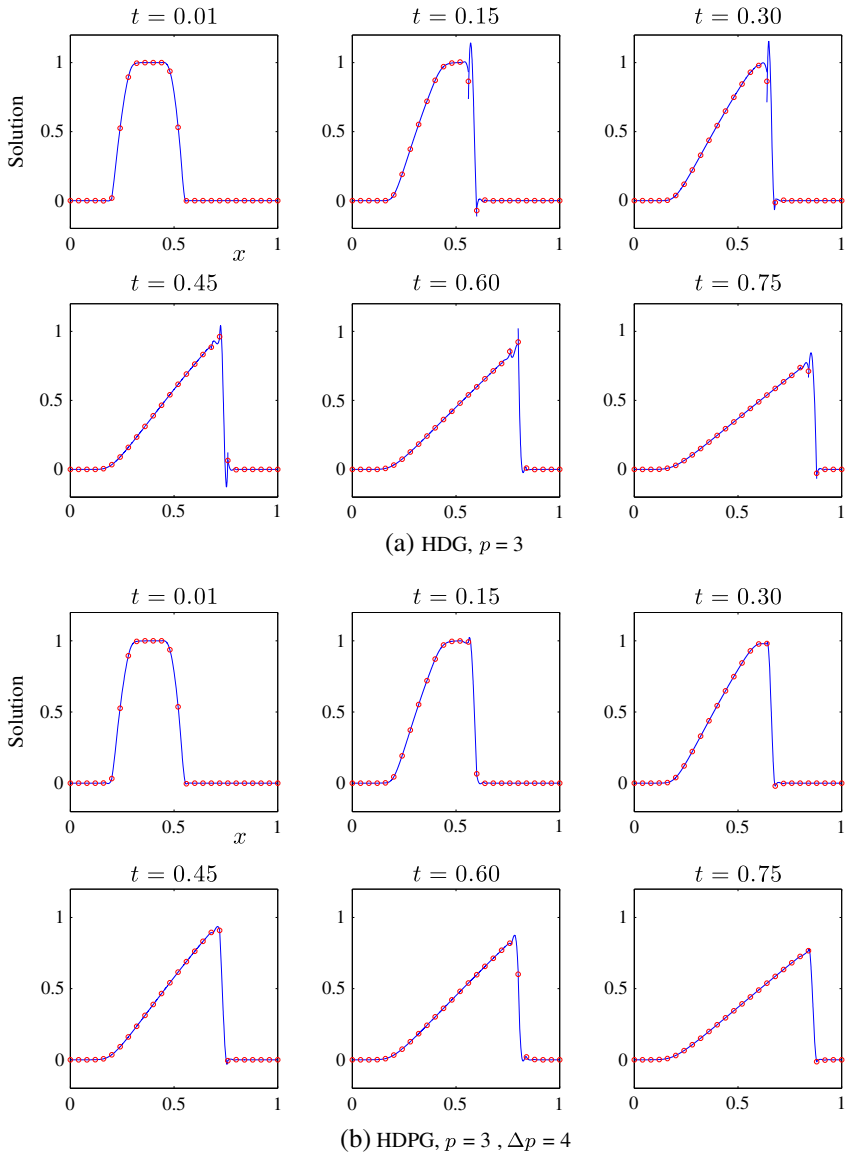


Figure 3. Burgers' equation solution for the case where a moving shock and an expansion fan form using (a) HDG and (b) HDPG. In this case, $p = 3$, $\Delta p = 4$ and a BDF3 scheme with $\Delta t = 10^{-2}$ is used.

DOI: 10.1002/nme

4.2. Peterson's example

The Peterson's example [32] is a famous test case in which DG methods are known to yield $p+1/2$ order of convergence. The DPG method proposed in [7,31] is the first of them that produces optimal convergence of order $p+1$. The problem to solve reads as follows:

$$\nabla \cdot (\mathbf{c}u) = 0, \quad \text{in } \Omega \in [0,1] \times [0,1], \tag{47}$$

where $\mathbf{c} = (0,1)$ and the boundary conditions are Dirichlet on the sides and bottom $u(x,y) = u_0$, and free outflow on the top. In this case, both $u_0(x) = x^2$ and $u_0(x) = \sin(6x)$ have been used as boundary condition, following Peterson [32] and Demkowicz *et al.* [31], respectively.

The results for the error and the convergence rate of the HDPG scheme, in this case, are summarized in Table I. Here, h is an indicator of the element size. As we can see, the expected $p+1/2$ convergence rate for HDG is achieved, while HDPG approaches $p+1$ for $\Delta p \geqslant 3$. This shows the enhanced stability of the proposed scheme. Notice for $\Delta p = 1$, the error of the HDPG scheme did not converge, even though the solver did not report conditioning issues with the matrices involved. The analysis carried out in [37] for the original DPG scheme partially explains this behavior.

4.2.1. Two-dimensional viscous Burgers' equation. In the last example, we apply the HDPG scheme to solve a two-dimensional Burgers' equation:

$$\frac{1}{2}\frac{\partial(u^2)}{\partial x} + \frac{\partial u}{\partial y} = \epsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad \text{in } \Omega \in [0,1] \times [0,1]. \tag{48}$$

Here, the boundary conditions need to be consistent with the hyperbolic character of the equation. In particular, we will set $u(x,y) = 1-2x$ on the $x=0$, $x=1$ and $y=0$ sides of the domain and extrapolation boundary conditions at $y=1$.

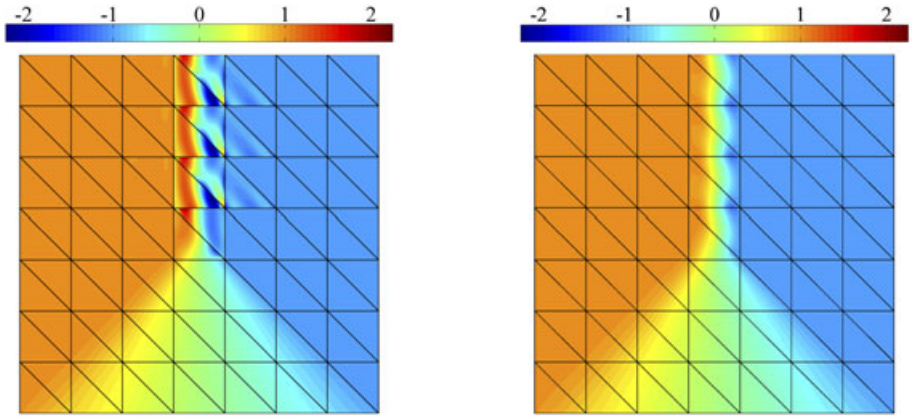
The results in Figure 4 compare HDG and HDPG for the case of a regular mesh of 98 triangular elements, using $p=4$ and no viscosity ($Pe|_{\text{cell}} = \infty$). As we can see, the HDPG solution is significantly less oscillatory with an enriched space of only order $\Delta p = 2$ higher.

These results might be influenced by the regularity of the mesh. In order to assess the method, we performed the computation of similar cases on an unstructured mesh. It is worth to mention that for

Table I. Error and convergence rate for Peterson's example using hybridizable discontinuous Galerkin (HDG) and hybridized discontinuous Petrov–Galerkin (HDPG) with $p=1$.

(a) $u_0 = x^2$					(b) $u_0 = \sin(6x)$				
Method	Δp	h	$\|u-u_h\|_2$	Order	Method	Δp	h	$\|u-u_h\|_2$	Order
HDG	–	0.167	3.19×10^{-3}	–	HDG	–	0.167	3.77×10^{-2}	–
HDG	–	0.083	1.18×10^{-3}	1.44	HDG	–	0.083	1.49×10^{-2}	1.33
HDG	–	0.042	4.04×10^{-4}	1.54	HDG	–	0.042	5.24×10^{-3}	1.51
HDG	–	0.021	1.56×10^{-4}	1.37	HDG	–	0.021	2.04×10^{-3}	1.37
HDG	–	0.010	5.36×10^{-5}	1.54	HDG	–	0.010	6.95×10^{-4}	1.55
HDPG	2	0.167	2.92×10^{-3}	–	HDPG	2	0.167	3.53×10^{-2}	–
HDPG	2	0.083	1.08×10^{-3}	1.43	HDPG	2	0.083	1.37×10^{-2}	1.37
HDPG	2	0.042	3.14×10^{-4}	1.78	HDPG	2	0.042	4.08×10^{-3}	1.74
HDPG	2	0.021	1.24×10^{-4}	1.35	HDPG	2	0.021	1.61×10^{-3}	1.34
HDPG	2	0.010	4.55×10^{-5}	1.44	HDPG	2	0.010	5.91×10^{-4}	1.44
HDPG	3	0.167	2.10×10^{-3}	–	HDPG	3	0.167	2.71×10^{-2}	–
HDPG	3	0.083	5.90×10^{-4}	1.83	HDPG	3	0.083	7.51×10^{-3}	1.85
HDPG	3	0.042	1.59×10^{-4}	1.90	HDPG	3	0.042	2.01×10^{-3}	1.90
HDPG	3	0.021	4.16×10^{-5}	1.93	HDPG	3	0.021	5.32×10^{-4}	1.92
HDPG	3	0.010	1.11×10^{-5}	1.91	HDPG	3	0.010	1.43×10^{-4}	1.90

DOI: 10.1002/nme

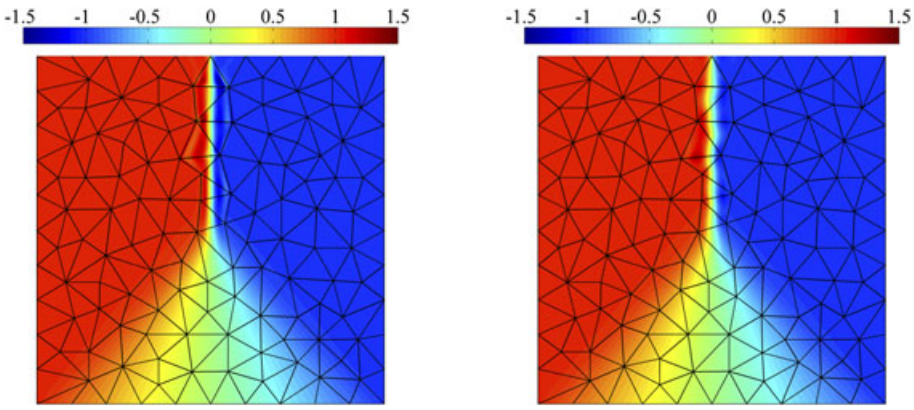


(a) HDG, $p = 4$, $Pe|_{cell} = \infty$ (b) HDPG, $p = 4$, $\Delta p = 2$, $Pe|_{cell} = \infty$

Figure 4. Solution to the Burgers equation in 2D using both (a) HDG and (b) HDPG on a structured mesh. Notice the reduced oscillation that HDPG introduces compared to HDG at the shock location.

Table II. Comparison of maximum relative oscillation (%) at the shock as a function of viscosity between HDG and HDPG for the space–time Burgers' equation using $p = 4$ and $\Delta p = 2$ on an unstructured grid.

$Pe\|_{cell}$	HDG oscillation (%)	HDPG oscillation (%)
2	0	0
10	44	30
50	90	42
100	110	43
1000	–	44
∞	–	44



(a) HDG, $p = 4$, $Pe|_{cell} = 10$ (b) HDPG, $p = 4$, $\Delta p = 2$, $Pe|_{cell} = 10$

Figure 5. Solution to the Burgers equation in 2D using both (a) HDG and (b) HDPG on an unstructured mesh. Notice the oscillation present in the HDG solution around the elements that contain the discontinuity, that is not present in the HDPG solution. Notice also the slight bending of the shock because of the mesh in the HDPG case, which indicates that the stabilization mechanism depends on the geometry of the element.

the case without viscosity, the HDG scheme failed to deliver a solution. In particular, divergence of the simulation occurred owing to the extreme oscillation around the discontinuity. To explore this phenomenon, different values of $Pe|_{\text{cell}} \in [2, \infty)$ were used to add extra stability to the scheme.

DOI: 10.1002/nme

This revealed that the HDPG scheme converged in all cases whereas the HDG scheme did not. The different overshoot present at the shock for both schemes, as a function of the artificial viscosity, is summarized in Table II. As we can see, once the viscous effects are small enough, the oscillation of the HDPG solution no longer grows, indicating that the stabilization mechanism introduced by the optimal test function is suitable for under-resolved situations. A visual comparison of the solution is included in Figure 5.

5. CONCLUSIONS

In this paper, we have presented the HDPG approach for scalar conservation laws. Our objective was to devise a method with the same complexity as the original HDG scheme, but with enhanced stability in the presence of discontinuities. To do so, the HDG scheme was combined with the theory of the optimal test functions, suitably modified to account for nonlinearity and to enforce conservation.

The scheme has been applied to linear convection and Burgers' equation in 1D and 2D, with and without the addition of artificial viscosity, and compared to the HDG scheme. The results indicate that HDPG delivers less oscillatory solutions in the presence of discontinuities, because of the stabilization role of the optimal test functions. In particular, optimal $p+1$ error estimates have been experimentally confirmed for the $p+1/2$ sub-optimal case constructed by Peterson.

We end the paper by noting that the application of HDPG to systems of conservation laws such as the Euler or Navier–Stokes equations is a subject of current research.

APPENDIX A: RESIDUAL AND DERIVATIVES EVALUATION

In this appendix, we will describe the different terms that enter the HDPG formulation. In what follows, the unknown $\mathbf{Z}$ that was introduced to alleviate the notation, will be split in the original components $\mathbf{u}$ and $\mathbf{Q}$. Let ϕ_i denote the ith basis function of the space $\mathcal{P}^p(K)$, of dimension m. Similarly, let ψ_i denote the ith basis function of the space $\mathcal{P}^{p+\Delta p}(K)$, of dimension n. Let $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{Q} \in \mathbb{R}^{m\times d}$ represent the vector of coefficients of the expansion of u_h and $\mathbf{q}_h$ in the basis for $\mathcal{P}^p(K)$ and $(\mathcal{P}^k(K))^d$, respectively.

$$u_h = \sum_{i=1}^{m} u_i \phi_i, \qquad q_{hj} = \sum_{i=1}^{m} q_{ij}\phi_i \quad j = 1,\dots,d.$$

Also, let ζ_{ij} denote the ith basis function for the space $\mathcal{P}^p(\mathcal{E}_K)$ at the jth local face. This space has dimension t at each face and can be broken in f face contributions. We denote by $\hat{\mathbf{U}} \in \mathbb{R}^{t\times f}$ the vector of coefficients of the expansion of $\hat{u}_h$ in the basis ζ_{ij}.

$$\hat{u}_h = \sum_{i=1}^{t}\sum_{j=1}^{f} \hat{U}_{ij}\zeta_{ij}.$$

We obtain the residuals of the local problem by integration against each element of the test space;

$$\begin{aligned} r^K_{ui} &= \left(\frac{u_h}{\Delta t}, \psi_i\right)_K + \langle \mathbf{F}(\hat{u}_h)\cdot\mathbf{n} + \epsilon\mathbf{q}_h\cdot\mathbf{n} + \tau(u_h,\hat{u}_h)(u_h-\hat{u}_h), \psi_i\rangle_{\partial K} \\ &\quad - (\mathbf{F}(u_h) - \epsilon\mathbf{q}_h, \nabla\psi_i)_K - \left(f + \frac{u_h^{s-1}}{\Delta t}, \psi_i\right)_K, \quad i = 1,\dots,n, \\ r^K_{\mathbf{q}ij} &= (q_{hj},\phi_i)_K + \left(u_h, \frac{\partial\phi_i}{\partial x_j}\right)_K - \langle \hat{u}_h, \phi_i\cdot n_j\rangle_{\partial K}, \quad i = 1,\dots,m \quad j = 1,\dots,d, \\ X_{Vik} &= (\psi_j,\psi_j)_K, \quad i = 1,\dots,n \quad k = 1,\dots,n, \end{aligned}$$

where, following the HDPG scheme presented earlier, the residual for the conservation law (r_{uK}) is integrated against polynomials of order Δp higher ($n > m$).

DOI: 10.1002/nme

With this in mind, we can compute the different Jacobians and second derivatives required in the local iteration as follows:

$$\frac{\partial r^K_{ui}}{\partial u_k} = \left(\frac{\phi_k}{\Delta t}, \psi_i\right)_K + \langle \frac{\partial \tau(u_h, \hat{u}_h)}{\partial u_h}\phi_k(u_h - \hat{u}_h), \psi_i\rangle_{\partial K} + \langle \tau(u_h, \hat{u}_h)\phi_k, \psi_i\rangle_{\partial K} - \left(\frac{\partial \mathbf{F}(u_h)}{\partial u_h}\phi_k, \nabla\psi_i\right)_K, \quad i = 1,\dots,n \quad k = 1,\dots,m,$$

$$\frac{\partial r^K_{ui}}{\partial q_{kl}} = \langle \epsilon\phi_k \cdot n_l, \psi_i\rangle_{\partial K}, \quad i = 1,\dots,n \quad k = 1,\dots,m \quad l = 1,\dots,d,$$

$$\frac{\partial r^K_{\mathbf{q}ij}}{\partial u_k} = \left(\phi_k, \frac{\partial \phi_i}{\partial x_j}\right)_K, \quad i = 1,\dots,m \quad j = 1,\dots,d \quad k = 1,\dots,m,$$

$$\frac{\partial r^K_{\mathbf{q}ij}}{\partial q_{kl}} = (\delta_{jk}\phi_k, \phi_i)_K, \quad i = 1,\dots,m \quad j = 1,\dots,d \quad k = 1,\dots,m \quad l = 1,\dots,d,$$

$$\frac{\partial^2 r^K_{ui}}{\partial u_k \partial u_r} = \langle \frac{\partial^2 \tau(u_h, \hat{u}_h)}{\partial u_h^2}\phi_k\phi_r(u_h - \hat{u}_h), \psi_i\rangle_{\partial K} + \langle 2\frac{\partial \tau(u_h, \hat{u}_h)}{\partial u_h}\phi_k\phi_r, \psi_i\rangle_{\partial K} - \left(\frac{\partial^2 \mathbf{F}(u_h^2)}{\partial u_h}\phi_k\phi_r, \nabla\psi_i\right)_K, \quad i = 1,\dots,n \quad k = 1,\dots,m \quad r = 1,\dots,m,$$

where only non-zero second derivatives have been computed.

Similarly, we can compute the terms required for the sensitivities as follows:

$$\frac{\partial r^K_{ui}}{\partial U_{kl}} = \langle \frac{\partial \mathbf{F}(\hat{u}_h)\cdot\mathbf{n}}{\partial \hat{u}_h}\zeta_{kl}, \psi_i\rangle_{\partial K} + \langle \frac{\partial \tau(u_h, \hat{u}_h)}{\partial \hat{u}_h}\zeta_{kl}(u_h - \hat{u}_h), \psi_i\rangle_{\partial K} - \langle \tau(u_h, \hat{u}_h)\zeta_{kl}, \psi_i\rangle_{\partial K}, \quad i = 1,\dots,n \quad k = 1,\dots,t \quad l = 1,\dots,f,$$

$$\frac{\partial r^K_{\mathbf{q}ij}}{\partial U_{kl}} = -\langle \zeta_{kl}, \phi_i \cdot n_j\rangle_{\partial K}, \quad i = 1,\dots,m \quad j = 1,\dots,d \quad k = 1,\dots,t \quad l = 1,\dots,f,$$

$$\frac{\partial^2 r^K_{ui}}{\partial u_r \partial U_{kl}} = \langle \frac{\partial^2 \tau(u_h, \hat{u}_h)}{\partial \hat{u}_h \partial u_h}\zeta_{kl}\phi_r(u_h - \hat{u}_h), \psi_i\rangle_{\partial K} + \langle \left(\frac{\partial \tau(u_h, \hat{u}_h)}{\partial \hat{u}_h} - \frac{\partial \tau(u_h, \hat{u}_h)}{\partial u_h}\right)\phi_r\zeta_{kl}, \psi_i\rangle_{\partial K}, \quad i = 1,\dots,m \quad j = 1,\dots,d \quad k = 1,\dots,t \quad l = 1,\dots,f, \quad r = 1,\dots,m,$$

again, only the non-zero terms have been computed.

Finally, we need to derive expressions for the computation of the global problem $\mathbf{r}_{\hat{u}} = 0$ and its derivatives. Following the usual finite element procedure, we introduce an index mapping from the DOFs of the faces, $\hat{U}_{ij}$, to the global DOFs of the space M_h^p, denoted by $\theta(i, j, k)$. The residual can then be computed by summation over all the elements as follows:

$$r_{\hat{u}_{\theta(i,j,k)}} = \sum_{k=1}^{e} \langle \mathbf{F}(\hat{u}_h)\cdot\mathbf{n} + \epsilon\mathbf{q}_h\cdot\mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), \zeta_{ij}\rangle_{K_k}$$

Similarly, we can compute the derivatives that enter (33):

$$\frac{\partial r_{\hat{u}_{\theta(i,j,k)}}}{\partial \hat{u}_{\theta(r,s,k)}} = \sum_{k=1}^{e} \langle \left(\frac{\partial \mathbf{F}(\hat{u}_h)\cdot\mathbf{n}}{\partial \hat{u}_h} + \frac{\partial \tau(u_h, \hat{u}_h)}{\partial \hat{u}_h}(u_h - \hat{u}_h) - \tau(u_h, \hat{u}_h)\right)\zeta_{rs}, \zeta_{ij}\rangle_{K_k} + \sum_{k=1}^{e} \langle \epsilon\phi_k \cdot n_l \frac{\partial Q_{kl}}{\partial \hat{U}_{rs}}, \zeta_{ij}\rangle_{K_k} + \sum_{k=1}^{e} \langle \left(\frac{\partial \tau(u_h, \hat{u}_h)}{\partial u_h}(u_h - \hat{u}_h) + \tau(u_h, \hat{u}_h)\right)\frac{\partial u_k}{\partial \hat{U}_{rs}}\phi_k, \zeta_{ij}\rangle_{K_k}.$$

DOI: 10.1002/nme

ACKNOWLEDGEMENTS

D. Moro would like to acknowledge the CajaMadrid Foundation for the Graduate Studies Scholarship that funded his work. N.C. Nguyen and J. Peraire gratefully acknowledge the support provided by the Singapore MIT Alliance as well as the Air Force Office of Scientific Research under the MURI program on biologically inspired flight. The authors would like to acknowledge Prof. J. Gopalakrishnan for his useful suggestions and comments and Dr. X. Roca for his comments and the mesh used for the 2D cases.

REFERENCES

1. Arnold D, Brezzi F, Cockburn B, Marini L. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2002; **39**(5):1749–1779.
2. Barter G, Darmofal D. Shock capturing with PDE-based artificial viscosity for DGFEM: part I. Formulation. *Journal of Computational Physics* 2010; **229**(5):1810–1827.
3. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics* 1997; **131**(2):267–279.
4. Baumann C, Oden J. A discontinuous HP finite element method for convection–diffusion problems. *Computer Methods in Applied Mechanical Engineering* 1999; **175**(3-4):311–341.
5. Cockburn B, Shu C. The local discontinuous Galerkin method for time-dependent convection–diffusion systems. *SIAM Journal on Numerical Analysis* 1998; **35**(6):2440–2463.
6. Cockburn B, Shu C. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing* 2001; **16**(3):173–261.
7. Demkowicz L, Gopalakrishnan J. A class of discontinuous Petrov-Galerkin methods. Part I: the transport equation. *Computer Methods in Applied Mechanical Engineering* 2010; **199**(23-24):1558–1572.
8. Hartmann R, Houston P. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics* 2002; **183**(2):508–532.
9. Hesthaven J, Warburton T. Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell's equations. *Journal of Computational Physics* 2002; **181**(1):186–221.
10. Klaij C, Van der Vegt J, Van der Ven H. Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations. *Journal of Computational Physics* 2006; **217**(2):589–611.
11. Lomtev I, Karniadakis G. A discontinuous Galerkin method for the Navier–Stokes equations. *International Journal for Numerical Methods in Engineering* 1999; **29**(5):587–603.
12. Peraire J, Persson PO. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing* 2008; **30**(4):1806–1824.
13. Reed N, Hill T. Triangle mesh methods for the neutron transport equation. *Technical Report LA2 UR-73-479*, Los Alamos Scientific Laboratory, 1973.
14. Cockburn B, Gopalakrishnan J, Lazarov R. Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis* 2009; **47**(2):1319–1365.
15. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. *Journal of Computational Physics* 2009; **228**(9):3232–3254.
16. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations. *Journal of Computational Physics* 2009; **228**(23):8841–8855.
17. Nguyen NC, Peraire J, Cockburn B. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics* 2011; **230**(10):3695–3718.
18. Soon SC, Cockburn B, Stolarski HK. A hybridizable discontinuous Galerkin method for linear elasticity. *International Journal for Numerical Methods in Engineering* 2009; **80**(8):1058–1092.
19. Cockburn B, Gopalakrishnan J. The derivation of hybridizable discontinuous Galerkin methods for Stokes flow. *SIAM Journal on Numerical Analysis* 2009; **47**:1092–1125.
20. Cockburn B, Gopalakrishnan J, Nguyen NC, Peraire J, Sayas F. Analysis of HDG methods for Stokes flow. *Mathematics of Computation* 2011; **80**:723–760.
21. Nguyen NC, Peraire J, Cockburn B. A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanical Engineering* 2010; **199**(9-12):582–597.
22. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2011; **230**(4):1147–1170.
23. Nguyen NC, Peraire J, Cockburn B. A hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations (AIAA Paper 2010-362). *Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, January 2010.
24. Nguyen NC, Peraire J, Cockburn B. Hybridizable discontinuous Galerkin methods. In *Lecture Notes in Computational Science and Engineering*, Vol. 76. Springer: Berlin, 2011; 63–84.
25. Nguyen NC, Peraire J, Cockburn B. A comparison of HDG methods for Stokes flow. *Journal of Scientific Computing* 2010; **45**:215–237.
26. Peraire J, Nguyen NC, Cockburn B. A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations (AIAA Paper 2010-363). *Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, FL, USA, 2010.

DOI: 10.1002/nme

27. Nguyen NC, Peraire J, Cockburn B. Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. *Journal of Computational Physics* 2011; **230**(19):7151–7175.
28. Güzey S, Cockburn B, Stolarski HK. The embedded discontinuous Galerkin method: application to linear shell problems. *International Journal for Numerical Methods in Engineering* 2007; **70**(7):757–790.
29. Peraire J, Nguyen NC, Cockburn B. An embedded discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations. *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, 2011.
30. Cockburn B, Dong B, Guzmán J, Restelli M, Sacco R. A hybridizable discontinuous Galerkin method for steady-state convection–diffusion-reaction problems. *SIAM Journal on Scientific Computing* 2009; **31**(5):3827–3846.
31. Demkowicz L, Gopalakrishnan J. A class of discontinuous Petrov–Galerkin methods. Part II: optimal test functions. *Numerical Methods for Partial Differential Equations* 2010; **27**(1):70–105.
32. Peterson TE. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis* 1991; **28**(1):133–140.
33. Demkowicz L, Gopalakrishnan J, Niemi A. A class of discontinuous Petrov–Galerkin methods. Part III: adaptivity. *Technical Report 10-1*, ICES, 2010.
34. Babuška I. Error-bounds for finite element method. *Numerische Mathematik* 1971; **16**(4):322–333.
35. Bottasso C, Micheletti S, Sacco R. A multiscale formulation of the discontinuous Petrov–Galerkin method for advective–diffusive problems. *Computer Methods in Applied Mechanical Engineering* 2005; **194**(25-26):2819–2838.
36. Chan J, Demkowicz L, Moser R, Roberts N. A class of discontinuous Petrov–Galerkin methods. Part V: solution of 1D Burgers and Navier–Stokes Equations. *Technical Report 10-25*, ICES, 2010.
37. Gopalakrishnan J, Qiu W. An analysis of the practical DPG method, 07 2011. http://arxiv.org/abs/1107.4293v1.

DOI: 10.1002/nme