49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2011, Orlando, Florida

AIAA 2011-197

A Hybridized Discontinuous Petrov-Galerkin Method for Compresible Flows

D. Moro*, N.C. Nguyen†, J. Peraire‡, J. Gopalakrishnan§

We present a hybridized discontinuous Petrov-Galerkin (HDPG) method for the solution of compressible flows. The HPDG method combines the efficiency of the hybridizable discontinuous Galerkin (HDG) method with the excellent stability of the discontinuous Petrov-Galerkin (DPG) method. This aim is achieved by using the DGP method to discretize the governing equations at the element level and the HDG method to glue the local solutions together. Moreover, we propose to enrich the test space with a constant function in order to make the HDPG method conservative. In the presence of under-resolved features such as shocks and boundary layers, the HDPG scheme is found to be more robust and stable than the HDG method. We present several numerical examples to demonstrate the proposed method.

I. Introduction

The development of robust, accurate, and efficient methods for the numerical solution of hyperbolic systems 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. The main difficulties in computing solutions with shocks are that (1) when a shock is formed it poses a source of instability in the shock region, which then leads to numerical instabilities if no treatment of shock waves is introduced; (2) it is hard to predict when and where new shocks arise, and track them as they propagate; (3) solution must satisfy the Rankine-Hugoniot jump condition and the entropy conditions; (4) solution should have sharp and clean shocks at the discontinuity interface; and (5) numerical treatment of shock waves should not cause deterioration in resolution and reduction of accuracy in domains where the solution is smooth. Although significant progress has been made over the years in both the theoretical and numerical investigations, capturing shocks, especially when shocks propagate and interact with one another, 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[1–4,8,9,11,14–17,30,32] for the numerical solution of hyperbolic systems of 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, in spite of all these advantages, DG methods have not yet made a more significant impact for practical applications. This is largely due to the main criticism that DG methods are computationally expensive. This cost is primarily associated to the large number of degrees of freedom caused by nodal duplication at the element boundary interfaces. More specifically, assuming about six linear tetrahedral elements per node, the number of unknowns in a DG system would approximately be 24 times the number of unknowns in the corresponding continuous Galerkin (CG) system for the same order of the approximating polynomial. The storage and computation cost of implicit DG methods are thus several times that of CG methods. However,

*MSc student, Department of Aeronautics and Astronautics, MIT
†Research Scientist, Department of Aeronautics and Astronautics, MIT
‡Professor, Department of Aeronautics and Astronautics, MIT
§Professor, Department of Mathematics, University of Florida

when used with explicit time-stepping schemes, DG methods[9,15] provide block-diagonal mass matrices to be inverted, which results in very low storage and efficient numerical schemes for many problems. One major disadvantage with explicit DG methods is that the timestep size is restricted by the smallest element in the mesh and the degree of polynomials used in representing the numerical solution. Even a few small elements can render the timestep size so small that it actually leads to time-consuming computations for many problems involving boundary layers and sharp features.

More recently, a new class of implicit DG methods — the so-called hybridizable discontinuous Galerkin (HDG) method — was first introduced for elliptic problems.[6] The HDG methods have already been extended to convection-diffusion systems,[24,25] linear and nonlinear elastodynamics,[21,34] incompressible and compressible flows[5,7,19,22,23,26–28] involving shock waves[18] and turbulence, and electromagnetics.[20] 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 degrees of freedom than standard DG methods. In fact, a variant of the HDG method — the so-called embedded DG method[13,29] — has the same global degrees of freedom as CG methods and has the stability properties of a DG method. This large reduction in the number of degrees of freedom can lead to significant savings for both computational time and memory storage. Another advantage of HDG methods is that their postprocessed 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.

Very recently, the Discontinuous Petrov-Galerkin method (DPG) was introduced for convection problems[11] and extended to linear convection-diffusion problems.[12] The main idea of the DGP 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 the 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. We present 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.

The paper is organized as follows. In Section 2, we introduce some notation used throughout the paper. In Section 3, we review the HDG method for hyperbolic systems of conservation laws. We then describe the HDPG method in Section 4 and present numerical results in Section 5. Finally, in Section 6, we provide some concluding remarks.

II. Notation

Let Ω be a physical domain in $\mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$ in $\mathbb{R}^{d-1}$. We denote by $\mathcal{T}_h$ a collection of disjoint elements (triangles and tetrahedrons) that partition Ω. We also denote by $\partial\mathcal{T}_h$ the set $\{\partial K : K \in \mathcal{T}_h\}$, that is, a collection of the boundaries of all elements in $\mathcal{T}_h$. We shall denote by $\boldsymbol{n}$ the outward unit normal of ∂K. For an element K of the collection $\mathcal{T}_h$, $F = \partial K \cap \partial\Omega$ is the boundary face if the $d-1$ Lebesgue measure of F is nonzero. For two elements K^+ and K^- of the collection $\mathcal{T}_h$, $F = \partial K^+ \cap \partial K^-$ is the interior face between K^+ and K^- if the $d-1$ Lebesgue measure of F is nonzero. Let $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$ denote the set of interior and boundary faces, respectively. We denote by $\mathcal{E}_h$ the union of $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$. Note that by definition $\partial\mathcal{T}_h$ and $\mathcal{E}_h$ are different. More precisely, an interior face is counted twice in $\partial\mathcal{T}_h$ but once in $\mathcal{E}_h$ and a boundary face is counted once in both $\partial\mathcal{T}_h$ and $\mathcal{E}_h$.

Let $\mathcal{P}^k(D)$ denote the set of polynomials of degree at most k on a domain D. We introduce discontinuous

finite element spaces

$$\mathcal{W}_h^k = \{w \in L^2(\mathcal{T}_h) \ : \ w|_K \in \mathcal{P}^k(K), \ \forall K \in \mathcal{T}_h\},$$
$$\boldsymbol{\mathcal{V}}_h^k = \{\boldsymbol{w} \in (L^2(\mathcal{T}_h))^m \ : \ \boldsymbol{w}|_K \in (\mathcal{P}^k(K))^m, \ \forall K \in \mathcal{T}_h\},$$
$$\boldsymbol{\mathcal{Q}}_h^k = \{\boldsymbol{W} \in (L^2(\mathcal{T}_h))^{m\times d} \ : \ \boldsymbol{W}|_K \in (\mathcal{P}^k(K))^{m\times d}, \ \forall K \in \mathcal{T}_h\},$$

for $\boldsymbol{w} = (w_i), 1 \le i \le m$, and $\boldsymbol{W} = (W_{ij}), 1 \le i \le m, 1 \le j \le d$. Here $L^2(D)$ is the space of square integrable functions on D. In addition, we introduce a traced finite element space

$$\boldsymbol{\mathcal{M}}_h^k = \{\boldsymbol{\mu} \in (L^2(\mathcal{E}_h))^m \ : \ \boldsymbol{\mu}|_F \in (\mathcal{P}^k(F))^m, \ \forall F \in \mathcal{E}_h\},$$

for $\boldsymbol{\mu} = (\mu_i), 1 \le i \le m$. Note that $\boldsymbol{\mathcal{M}}_h^k$ consists of functions which are continuous inside the faces $F \in \mathcal{E}_h$ and discontinuous at their borders.

For functions w and v in $L^2(D)$, we denote $(w,v)_D = \int_D wv$ if D is a domain in $\mathbb{R}^d$ and $\langle w,v\rangle_D = \int_D wv$ if D is a domain in $\mathbb{R}^{d-1}$. Likewise, for functions $\boldsymbol{w}$ and $\boldsymbol{v}$ in $(L^2(D))^m$, we denote $(\boldsymbol{w},\boldsymbol{v})_D = \int_D \boldsymbol{w}\cdot\boldsymbol{v}$ if D is a domain in $\mathbb{R}^d$ and $\langle \boldsymbol{w},\boldsymbol{v}\rangle_D = \int_D \boldsymbol{w}\boldsymbol{v}$ if D is a domain in $\mathbb{R}^{d-1}$. For functions $\boldsymbol{W}$ and $\boldsymbol{V}$ in $(L^2(D))^{m\times d}$, we denote $(\boldsymbol{W},\boldsymbol{V})_D = \int_D \mathrm{tr}(\boldsymbol{W}^T\boldsymbol{V})$ if D is a domain in $\mathbb{R}^d$ and $\langle \boldsymbol{W},\boldsymbol{V}\rangle_D = \int_D \mathrm{tr}(\boldsymbol{W}^T\boldsymbol{V})$ if D is a domain in $\mathbb{R}^{d-1}$, where tr is the trace operator of a square matrix. We finally introduce the following volume inner products

$$(w,v)_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (w,v)_K, \qquad (\boldsymbol{w},\boldsymbol{v})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\boldsymbol{w},\boldsymbol{v})_K, \qquad (\boldsymbol{W},\boldsymbol{V})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\boldsymbol{W},\boldsymbol{V})_K,$$

and boundary inner products

$$\langle w,v\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle w,v\rangle_{\partial K}, \qquad \langle \boldsymbol{w},\boldsymbol{v}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \boldsymbol{w},\boldsymbol{v}\rangle_{\partial K}, \qquad \langle \boldsymbol{W},\boldsymbol{V}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \boldsymbol{W},\boldsymbol{V}\rangle_{\partial K}.$$

All of the above notations and definitions are necessary for the description of the ideas in this paper.

III. The Hybridizable Discontinuous Galerkin Method

In this section, we review the hybridizable discontinuous Galerkin (HDG) Method for solving the hyperbolic system of conservation laws written as a first-order system of equations as

$$\begin{aligned} \boldsymbol{Q} - \nabla\boldsymbol{u} &= 0, && \text{in } \Omega\times(0,T], \\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla\cdot\boldsymbol{F}(\boldsymbol{u},\boldsymbol{Q}) &= \boldsymbol{f}, && \text{in } \Omega\times(0,T], \end{aligned} \tag{1}$$

where $\boldsymbol{u} = (u_i), 1 \le i \le m$, is a vector of m conserved quantities, $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{Q})$ are fluxes of dimension $m\times d$, and $\boldsymbol{f}$ is a source term. In general, the fluxes can be the written as $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{Q}) = \boldsymbol{F}^{\mathrm{inv}}(\boldsymbol{u}) + \boldsymbol{F}^{\mathrm{vis}}(\boldsymbol{u},\boldsymbol{Q})$, where $\boldsymbol{F}^{\mathrm{inv}}(\boldsymbol{u})$ are the inviscid fluxes and $\boldsymbol{F}^{\mathrm{vis}}(\boldsymbol{u},\boldsymbol{Q})$ are the viscous fluxes. The above system is supplemented with the following initial condition

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_0(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \Omega, \tag{2}$$

and the following general boundary condition

$$\boldsymbol{b}(\boldsymbol{u},\boldsymbol{Q},\boldsymbol{n}) = 0, \quad \text{on } \partial\Omega, \tag{3}$$

where $\boldsymbol{n}$ is the normal unit vector to the boundary $\partial\Omega$. The function $\boldsymbol{b}$ depends on $(\boldsymbol{u},\boldsymbol{Q})$ as well as $\boldsymbol{n}$ and is called the boundary flux operator. The form of the boundary flux operator depends on the types of boundary conditions (Dirichlet and Neumann boundary conditions, inflow and outflow boundary conditions, wall boundary conditions, etc.) applied on the boundary of the physical domain. We shall discuss these boundary conditions later in the paper.

A. Formulation

We first consider the governing equations (1) on any element $K \in \mathcal{T}_h$, multiply them with some test functions $(\boldsymbol{E}, \boldsymbol{w}) \in (\mathcal{P}^k(K))^{m\times d} \times (\mathcal{P}^k(K))^m$, and integrate the resulting equations by part. This results in the so-called *local problem*: $(\boldsymbol{Q}_h, \boldsymbol{u}_h) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k$ satisfies

$$\begin{aligned} (\boldsymbol{Q}_h, \boldsymbol{E})_K + (\boldsymbol{u}_h, \nabla\cdot\boldsymbol{E})_K - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial K} &= 0, \qquad && \forall\, \boldsymbol{E} \in (\mathcal{P}^k(K))^{m\times d}, \\ \Big(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w}\Big)_K - (\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{Q}_h), \nabla\boldsymbol{w})_K + \langle \widehat{\boldsymbol{q}}_h, \boldsymbol{w}\rangle_{\partial K} &= (\boldsymbol{f}, \boldsymbol{w})_K, \qquad && \forall\, \boldsymbol{w} \in (\mathcal{P}^k(K))^m, \end{aligned} \tag{4}$$

where the numerical flux $\widehat{\boldsymbol{q}}_h$ is an approximation to $\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{Q})\cdot\boldsymbol{n}$ over ∂K. Note that (4) is a Galerkin projection of the governing equations at the element level onto $(\mathcal{P}^k(K))^{m\times d} \times (\mathcal{P}^k(K))^m$. Second, we take the numerical flux to be

$$\widehat{\boldsymbol{q}}_h = \boldsymbol{F}(\widehat{\boldsymbol{u}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \quad \text{on } \partial K, \tag{5}$$

where $\widehat{\boldsymbol{u}}_h \in \boldsymbol{\mathcal{M}}_h^k$ is an approximation to the trace of the solution $\boldsymbol{u}$ on ∂K, and $\boldsymbol{S}$ is a stabilization matrix which has an important effect on both the stability and accuracy of the resulting scheme. Third, by enforcing the continuity of the L^2 projection of $\widehat{\boldsymbol{q}}_h$ across interior faces and imposing the boundary conditions, we obtain the so-called *global weak formulation*

$$\langle \widehat{\boldsymbol{q}}_h, \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \Big\langle \widehat{\boldsymbol{b}}_h, \boldsymbol{\mu}\Big\rangle_{\partial\Omega} = 0, \quad \forall\, \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}_h^k, \tag{6}$$

where $\widehat{\boldsymbol{b}}_h$ is the boundary numerical flux whose definition depends on the boundary conditions and will be given below.

We next insert (5) into (4) and sum the resulting equation over all elements to arrive at the semi-discrete HDG formulation: $(\boldsymbol{Q}_h, \boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k \times \boldsymbol{\mathcal{M}}_h^k$ satisfies

$$(\boldsymbol{Q}_h, \boldsymbol{E})_{\mathcal{T}_h} + (\boldsymbol{u}_h, \nabla\cdot\boldsymbol{E})_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0, \tag{7a}$$

$$\Big(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w}\Big)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{Q}_h), \nabla\boldsymbol{w})_{\mathcal{T}_h} + \langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \boldsymbol{w}\rangle_{\partial\mathcal{T}_h} = (\boldsymbol{f}, \boldsymbol{w})_{\mathcal{T}_h}, \tag{7b}$$

$$\langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \Big\langle \widehat{\boldsymbol{b}}_h, \boldsymbol{\mu}\Big\rangle_{\partial\Omega} = 0, \tag{7c}$$

for all $(\boldsymbol{E}, \boldsymbol{w}, \boldsymbol{\mu}) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k \times \boldsymbol{\mathcal{M}}_h^k$.

Finally, we consider to advance the semi-discrete system (7) in time using an implicit time integrator. For simplicity of exposition, we consider the Backward Euler method for time integration. We denote by $(\boldsymbol{Q}_h^n, \boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n)$ the numerical approximations to $(\boldsymbol{Q}(t^n), \boldsymbol{u}(t^n), \widehat{\boldsymbol{u}}(t^n))$ at time $t^n = n\Delta t^n$, where Δt^n is a timestep size at level n. Using the backward Euler method to discretize the time derivative in (7b), we obtain that $(\boldsymbol{Q}_h^n, \boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k \times \boldsymbol{\mathcal{M}}_h^k$ satisfies

$$(\boldsymbol{Q}_h^n, \boldsymbol{E})_{\mathcal{T}_h} + (\boldsymbol{u}_h^n, \nabla\cdot\boldsymbol{E})_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{u}}_h^n, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0, \tag{8a}$$

$$\Big(\frac{\boldsymbol{u}_h^n}{\Delta t^n}, \boldsymbol{w}\Big)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h^n, \boldsymbol{Q}_h^n), \nabla\boldsymbol{w})_{\mathcal{T}_h} + \langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n)\cdot\boldsymbol{n} + \boldsymbol{S}^n(\boldsymbol{u}_h^n - \widehat{\boldsymbol{u}}_h^n), \boldsymbol{w}\rangle_{\partial\mathcal{T}_h} = \Big(\boldsymbol{f}^n + \frac{\boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{w}\Big)_{\mathcal{T}_h}, \tag{8b}$$

$$\langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n)\cdot\boldsymbol{n} + \boldsymbol{S}^n(\boldsymbol{u}_h^n - \widehat{\boldsymbol{u}}_h^n), \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \Big\langle \widehat{\boldsymbol{b}}_h^n, \boldsymbol{\mu}\Big\rangle_{\partial\Omega} = 0, \tag{8c}$$

for all $(\boldsymbol{E}, \boldsymbol{w}, \boldsymbol{\mu}) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k \times \boldsymbol{\mathcal{M}}_h^k$ and $n \geq 1$, where $(\boldsymbol{Q}_h^0, \boldsymbol{u}_h^0, \widehat{\boldsymbol{u}}_h^0)$ is the L^2 projection of the initial data $(\boldsymbol{Q}_0, \boldsymbol{u}_0, \boldsymbol{u}_0)$ onto $\boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k \times \boldsymbol{\mathcal{M}}_h^k$.

We note that the use of higher-order backward difference formula (BDF) schemes such as the second-order accurate BDF2 scheme and the third-order accurate BDF3 scheme gives rise to a similar formulation as (8). The HDG method can also work with other implicit time-stepping methods such as the diagonally implicit Runge-Kutta (DIRK) methods and DG methods in time.

B. Boundary conditions

In this subsection, we show how the HDG method can handle various boundary conditions by appropriately defining the boundary numerical flux $\widehat{\boldsymbol{b}}_h^n$. We first consider the Dirichlet boundary condition in which the function $\boldsymbol{b}$ is given by

$$\boldsymbol{b}(t) = \boldsymbol{u}(t) - \boldsymbol{g}_D(t), \quad \text{on } \partial\Omega_D, \ \forall t \in (0, T], \tag{9}$$

where $\partial\Omega_D$ is a portion of the boundary $\partial\Omega$ and $\boldsymbol{g}_D$ is the Dirichlet boundary data. The corresponding boundary numerical flux is then defined as

$$\widehat{\boldsymbol{b}}_h^n = \widehat{\boldsymbol{u}}_h^n - \boldsymbol{g}_D^n, \quad \text{on } \partial\Omega_D. \tag{10}$$

Similarly, for the Neumann boundary condition in which $\boldsymbol{b}$ is given by

$$\boldsymbol{b}(t) = \boldsymbol{F}(\boldsymbol{u}(t), \boldsymbol{Q}(t)) \cdot \boldsymbol{n} - \boldsymbol{g}_N(t), \quad \text{on } \partial\Omega_N, \ \forall t \in (0, T], \tag{11}$$

we define the associated boundary numerical flux as

$$\widehat{\boldsymbol{b}}_h^n = \widehat{\boldsymbol{q}}_h^n - \boldsymbol{g}_N^n, \quad \text{on } \partial\Omega_N. \tag{12}$$

Here $\partial\Omega_N$ is another portion of the boundary $\partial\Omega$ and $\boldsymbol{g}_N$ is the Neumann boundary data. Other Neumann and mixed boundary conditions can be imposed in a similar manner.

We now consider other boundary conditions for viscous compressible fluid flows such as the inflow and outflow boundary conditions as well as wall boundary conditions. These conditions can be thought of as a combination of Dirichlet and Neumann conditions applied to different components, or combinations of components, of the solution vector. For instance, at the inlet section or outlet section of the flow, we need to either set the state variable $\boldsymbol{u}$ to the freestream condition $\boldsymbol{u}_\infty$ or set the viscous stresses to zero depending on the eigenvalues of the system. To this end, we define the boundary flux vector $\widehat{\boldsymbol{b}}_h$ as

$$\widehat{\boldsymbol{b}}_h^n = \boldsymbol{F}^{\text{vis}}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n) \cdot \boldsymbol{n} + \boldsymbol{A}^+(\widehat{\boldsymbol{u}}_h^n)(\boldsymbol{u}_h^n - \widehat{\boldsymbol{u}}_h^n) - \boldsymbol{A}^-(\widehat{\boldsymbol{u}}_h^n)(\boldsymbol{u}_\infty^n - \widehat{\boldsymbol{u}}_h^n), \tag{13}$$

where $\boldsymbol{F}^{\text{vis}}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n)$ are the viscous fluxes, which are very small at the inflow and outflow boundaries, and $\boldsymbol{A}^\pm = (\boldsymbol{A} \pm |\boldsymbol{A}|)/2$ and $\boldsymbol{A} = [\partial \boldsymbol{F}^{\text{inv}}(\boldsymbol{u})/\partial \boldsymbol{u}] \cdot \boldsymbol{n}$ denotes the Jacobian of the normal component of the inviscid fluxes.[33]

Finally, at the solid surface, we need to impose zero velocity and either a fixed temperature $T = T_w$ (isothermal wall) or zero heat flux $\partial T/\partial \boldsymbol{n} = 0$ (adiabatic wall). In order to do that, we first set

$$\widehat{b}_{h1}^n = \widehat{u}_{h1}^n - u_{h1}^n, \tag{14}$$

which means that we extrapolate the density. We then set

$$\widehat{b}_{hi}^n = \widehat{u}_{hi}^n, \quad 2 \le i \le m-1, \tag{15}$$

which means that we impose $\widehat{u}_{hi}^n = 0, 2 \le i \le m-1$, at the viscous solid wall. For the last component of $\widehat{\boldsymbol{b}}_h^n$, we need to distinguish between the isothermal wall and the adiabatic wall. For the isothermal wall, we set

$$\widehat{b}_{hm}^n = T_w - \widehat{T}_h(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n), \tag{16}$$

where $\widehat{T}_h(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n)$ is the approximate trace of the temperature and it is determined from $\widehat{\boldsymbol{u}}_h^n$ and $\boldsymbol{Q}_h^n$. For the adiabatic wall, we set

$$\widehat{b}_{hm}^n = \boldsymbol{F}_m^{\text{vis}}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n) \cdot \boldsymbol{n}, \tag{17}$$

which means that we impose zero heat flux at the solid wall.

C. Stabilization matrix

There are several possible choices for the stabilization matrix including the Roe scheme[33] and Lax-Friedrich scheme.[9] For the Roe scheme, we choose

$$\boldsymbol{S} = \boldsymbol{L}|\boldsymbol{\Lambda}|\boldsymbol{R}, \tag{18}$$

where $\boldsymbol{L}$, $\boldsymbol{R}$, and $\boldsymbol{\Lambda}$ are the matrices of the left and right eigenvectors, and eigenvalues of the Jacobian matrix $[\partial \boldsymbol{F}^{\text{inv}}(\widehat{\boldsymbol{u}}_h^n)/\partial \widehat{\boldsymbol{u}}_h^n] \cdot \boldsymbol{n}$, respectively. The second choice is the local Lax-Friedrich scheme

$$\boldsymbol{S} = \tau_{\max}^\ell \boldsymbol{I}, \tag{19}$$

where $\tau_{\max}^\ell$ is the local maximum speed of the system, and $\boldsymbol{I}$ is the identity matrix. The third choice is the global Lax-Friedrich scheme

$$\boldsymbol{S} = \tau_{\max}^g \boldsymbol{I}, \tag{20}$$

where $\tau_{\max}^g$ is the global maximum speed of the system. In general, the choice of the stabilization matrix becomes less critical for high k since numerical dissipation in the order of $O(h^{k+1})$ vanishes rapidly with increasing k.

D. Implementation

By applying the Newton-Raphson method to linearize the nonlinear system (8), we obtain the following linear system at every Newton iteration

$$\begin{bmatrix} \mathbf{A}^n & \mathbf{B}^n \\ \mathbf{C}^n & \mathbf{D}^n \end{bmatrix} \begin{pmatrix} \delta\mathbf{s}^n \\ \delta\widehat{\mathbf{u}}^n \end{pmatrix} = \begin{pmatrix} \mathbf{f}^n \\ \mathbf{g}^n \end{pmatrix}, \tag{21}$$

where $\delta\mathbf{s}^n$ and $\delta\widehat{\mathbf{u}}^n$ are the vectors of degrees of freedom of $(\delta\boldsymbol{Q}_h^n, \delta\boldsymbol{u}_h^n)$ and $\delta\widehat{\boldsymbol{u}}_h^n$, respectively, which are in that order the Newton increments of the current iterate $(\boldsymbol{Q}_h^n, \boldsymbol{u}_h^n)$ and $\widehat{\boldsymbol{u}}_h^n$. It is important to note that the matrix $\mathbf{A}^n$ has a block-diagonal structure due to the discontinuous nature of the approximation spaces. Therefore, it can be inverted at the element level to yield a block-diagonal matrix $(\mathbf{A}^n)^{-1}$. We can thus eliminate $\delta\mathbf{s}^n$ to obtain a reduced system in terms of $\delta\widehat{\mathbf{u}}^n$ as

$$\mathbf{K}^n \, \delta\widehat{\mathbf{u}}^n = \mathbf{r}^n, \tag{22}$$

where

$$\mathbf{K}^n = -\mathbf{C}^n(\mathbf{A}^n)^{-1}\mathbf{B}^n + \mathbf{D}^n, \qquad \mathbf{r}^n = \mathbf{g}^n - \mathbf{C}^n(\mathbf{A}^n)^{-1}\mathbf{f}^n. \tag{23}$$

This is the global system to be solved at every Newton iteration. Since $\delta\widehat{\boldsymbol{u}}_h^n$ is single-valued over faces of the elements, the final matrix system of the HDG method is smaller than that of many other DG methods. Moreover, the matrix $\mathbf{K}^n$ is compact in the sense that only the degrees of freedom between neighboring faces that share the same element are connected. To form $\mathbf{K}^n$ we do not need to explicitly compute the matrices $\mathbf{A}^n, \mathbf{B}^n, \mathbf{C}^n$, and $\mathbf{D}^n$. Instead we compute the elemental matrices and elemental vectors, and perform the standard finite element assembly to form the system (22).

For large problems, iterative solution methods are unavoidable. One of the key indicators of the cost in an iterative method is the cost of the matrix vector multiplication which is proportional to the number of non-zeros in the problem matrix. For HDG one not only obtains a smaller matrix with fewer globally coupled degrees of freedom, but for a given size matrix the number of nonzeros is smaller. This is because in standard DG methods the number of nonzero elements scales like $O(k^d)$, whereas in the HDG method the number of nonzeros scales like $O(k^{d-1})$. This has the potential for significantly smaller matrices and hence more efficient solution techniques.

IV. The Hybridized Discontinuous Petrov-Galerkin Method

In this section, we introduce the hybridized discontinuous Petrov-Galerkin (HDPG) method that combines the efficiency of the HDG method with the excellent stability of the DPG method.[11,12] The essential ingredients are a local Petrov-Galerkin projection of the underlying PDEs at the element level onto spaces of polynomials of degree k to parametrize the numerical solution in terms of the numerical trace; an enrichment of the test space with constant shape function to ensure conservativity; a judicious choice of the numerical flux to provide stability and consistency; and a global jump condition that enforces the continuity of the numerical flux to arrive at a global weak formulation in terms of the numerical trace. In particular, we focus our attention on devising the HDPG method for solving the steady-state version of the hyperbolic system (1):

$$\begin{aligned} \boldsymbol{Q} - \nabla\boldsymbol{u} &= 0, \qquad \text{in } \Omega, \\ \nabla\cdot\boldsymbol{F}(\boldsymbol{u},\boldsymbol{Q}) &= \boldsymbol{f}, \qquad \text{in } \Omega. \end{aligned} \tag{24}$$

We then extend the method to the time-dependent case (1) near the end of the section.

A. Formulation

1. *The global weak formulation*

In the HDPG method we seek an approximation $\widehat{\boldsymbol{u}}_h \in \boldsymbol{\mathcal{M}}_h^k$ such that

$$\left\langle \widehat{\boldsymbol{q}}_h(\widehat{\boldsymbol{u}}_h), \boldsymbol{\mu} \right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \left\langle \widehat{\boldsymbol{b}}_h(\widehat{\boldsymbol{u}}_h), \boldsymbol{\mu} \right\rangle_{\partial\Omega} = 0, \quad \forall\, \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}_h^k, \tag{25}$$

where, in the spirit of the HDG method, the interior numerical flux is

$$\widehat{\boldsymbol{q}}_h(\widehat{\boldsymbol{u}}_h) = \boldsymbol{F}(\widehat{\boldsymbol{u}}_h, \boldsymbol{Q}_h(\widehat{\boldsymbol{u}}_h))\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h(\widehat{\boldsymbol{u}}_h) - \widehat{\boldsymbol{u}}_h), \tag{26}$$

and the boundary numerical flux $\widehat{\boldsymbol{b}}_h$ depends on the particular boundary conditions. Both the boundary numerical flux $\widehat{\boldsymbol{b}}_h$ and the stabilization matrix $\boldsymbol{S}$ are chosen the same as those for the HDG method.

We see that the global weak formulation (25) is the same for both the HDG method and the HDPG method. However, the local problem that defines $(\boldsymbol{u}_h, \boldsymbol{Q}_h)$ as a function of $\widehat{\boldsymbol{u}}_h$ will be different for each method. The difference lies in a projection method applied to the governing equations at the element level. In particular, the HDG method uses the Galerkin projection as describe earlier, whereas the HDGP method employs the Petrov-Galerkin projection as discussed below.

2. The local Petrov-Galerkin projection

We begin by recalling that the trial space for the solution of the local problem is $\boldsymbol{\mathcal{U}}(K) \equiv (\mathcal{P}^k(K))^{m\times d} \times (\mathcal{P}^k(K))^m$ and that the restriction of $(\boldsymbol{Q}_h, \boldsymbol{u}_h)$ on K resides in $\boldsymbol{\mathcal{U}}(K)$. To define the Petrov-Galerkin projection we introduce a test space $\boldsymbol{\mathcal{S}}(K) \equiv (\mathcal{P}^{k+\Delta k}(K))^{m\times d} \times (\mathcal{P}^{k+\Delta k}(K))^m$ for any given positive integer $\Delta k \geq 1$. It is clear from their definition that the trial space is a subset of the test space.

To define the local problem we consider the governing equations (24) on any element $K \in \mathcal{T}_h$, multiply them with some test function $(\boldsymbol{E}, \boldsymbol{w}) \in \boldsymbol{\mathcal{S}}(K)$, and integrate the resulting equations by parts. We thus obtain that $(\boldsymbol{Q}_h, \boldsymbol{u}_h) \in \boldsymbol{\mathcal{U}}(K)$ satisfies

$$\begin{aligned} (\boldsymbol{Q}_h, \boldsymbol{E})_K + (\boldsymbol{u}_h, \nabla\cdot\boldsymbol{E})_K - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial K} &= 0, \\ -(\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{Q}_h), \nabla\boldsymbol{w})_K + \langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \boldsymbol{w}\rangle_{\partial K} - (\boldsymbol{f}, \boldsymbol{w})_K &= 0. \end{aligned} \tag{27}$$

To simplify the notation, we introduce $\boldsymbol{U}_h \equiv (\boldsymbol{Q}_h, \boldsymbol{u}_h)$ and $\boldsymbol{V} = (\boldsymbol{E}, \boldsymbol{w})$ and rewrite the above equation in a more compact form as

$$r_K\big(\boldsymbol{U}_h, \boldsymbol{V}; \widehat{\boldsymbol{u}}_h\big) = 0, \tag{28}$$

where r_K is given by

$$\begin{aligned} r_K\big(\boldsymbol{U}_h, \boldsymbol{V}; \widehat{\boldsymbol{u}}_h\big) &= (\boldsymbol{Q}_h, \boldsymbol{E})_K + (\boldsymbol{u}_h, \nabla\cdot\boldsymbol{E})_K - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial K} \\ &\quad -(\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{Q}_h), \nabla\boldsymbol{w})_K + \langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h, \boldsymbol{Q}_h)\cdot\boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \boldsymbol{w}\rangle_{\partial K} - (\boldsymbol{f}, \boldsymbol{w})_K. \end{aligned} \tag{29}$$

In this setting, $\boldsymbol{U}_h$ is the solution of the local problem (28), $\boldsymbol{V}$ is some appropriate test function, and $\widehat{\boldsymbol{u}}_h$ is the Dirichlet data to the local problem (28). As a result, the local problem (28) defines $(\boldsymbol{Q}_h, \boldsymbol{u}_h)$ as a function of $\widehat{\boldsymbol{u}}_h$. It remains to determine the test functions.

In the spirit of the DPG method, we choose the test functions $\boldsymbol{V} \in \boldsymbol{\mathcal{S}}(K)$ that are solutions of

$$(\boldsymbol{V}, \boldsymbol{Z})_K = r'_K(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{U}_h; \widehat{\boldsymbol{u}}_h), \quad \forall\, \boldsymbol{W} \in \boldsymbol{\mathcal{U}}(K), \forall \boldsymbol{Z} \in \boldsymbol{\mathcal{S}}(K), \tag{30}$$

where $r'_K(\cdot,\cdot,\boldsymbol{U}_h;\widehat{\boldsymbol{u}}_h)$ is the Fréchet derivative of $r_K(\cdot,\cdot;\widehat{\boldsymbol{u}}_h)$ with respect to the first argument $\boldsymbol{U}_h$. In essence, the equation (31) defines a mapping from the trial space $\boldsymbol{\mathcal{U}}(K)$ to the test space $\boldsymbol{\mathcal{S}}(K)$. More specifically, let $\boldsymbol{\phi}_i, 1 \leq i \leq n$, be basis functions of $\boldsymbol{\mathcal{U}}(K)$. Then the corresponding test functions $\boldsymbol{\varphi}_i, 1 \leq i \leq n$, are given by

$$(\boldsymbol{\varphi}_i, \boldsymbol{Z})_K = r'_K(\boldsymbol{\phi}_i, \boldsymbol{Z}, \boldsymbol{U}_h; \widehat{\boldsymbol{u}}_h), \quad \forall \boldsymbol{Z} \in \boldsymbol{\mathcal{S}}(K). \tag{31}$$

In general, the space spanned by the test functions $\boldsymbol{\varphi}_i, 1 \leq i \leq n$, does not contain the constant function. This implies that the Petrov-Galerkin method is not conservative.

To implement the Petrov-Galerkin method, we express $\boldsymbol{U}_h$ and $\widehat{\boldsymbol{u}}_h$ as

$$\boldsymbol{U}_h = \sum_{i=1}^{n} \mathrm{U}_i\ \boldsymbol{\phi}_i, \qquad \widehat{\boldsymbol{u}}_h = \sum_{j=1}^{N} \widehat{\mathrm{u}}_j\ \boldsymbol{\xi}_j, \tag{32}$$

where $\mathbf{U} = (\mathrm{U}_1, \ldots, \mathrm{U}_n)^T$ is the coefficient vector for $\boldsymbol{U}_h$ and $\widehat{\mathbf{u}} = (\widehat{\mathrm{u}}_1, \ldots, \widehat{\mathrm{u}}_N)^T$ is the coefficient vector for $\widehat{\boldsymbol{u}}_h$. Here $\boldsymbol{\xi}_j \in (\mathcal{P}^k(\{F : F \in \partial K\}))^m, 1 \leq j \leq N$, where $N = (d+1)\ \dim\ (\mathcal{P}^k(F))^m$, are basis functions defined on the faces of the boundary ∂K of an element $K \in \mathcal{T}_h$. Substituting (32) into (28) and (31) we obtain the following nonlinear system of equations as

$$\mathbf{J}(\mathbf{U};\widehat{\mathbf{u}})^T \mathbf{X}^{-1} \mathbf{r}(\mathbf{U};\widehat{\mathbf{u}}) = 0, \tag{33}$$

where, for any $\mathbf{W} = (\mathrm{W}_1, \ldots, \mathrm{W}_n)^T$,

$$\begin{aligned} \mathrm{r}_i(\mathbf{W};\widehat{\mathbf{u}}) &= r_K\Big(\textstyle\sum_{\ell=1}^n \mathrm{W}_\ell\, \boldsymbol{\phi}_\ell, \boldsymbol{\psi}_i; \sum_{j=1}^N \widehat{\mathrm{u}}_j\, \boldsymbol{\xi}_j\Big), && 1 \le i \le m, \\ \mathrm{J}_{ij}(\mathbf{W};\widehat{\mathbf{u}}) &= r'_K\Big(\boldsymbol{\phi}_j, \boldsymbol{\psi}_i, \textstyle\sum_{\ell=1}^n \mathrm{W}_\ell\, \boldsymbol{\phi}_\ell; \sum_{p=1}^N \widehat{\mathrm{u}}_p\, \boldsymbol{\xi}_p\Big), && 1 \le i \le m, 1 \le j \le n, \\ \mathrm{X}_{ij} &= (\boldsymbol{\psi}_j, \boldsymbol{\psi}_i)_K, && 1 \le i, j \le m. \end{aligned} \tag{34}$$

Here $\boldsymbol{\psi}_i, 1 \le i \le m$, are basis functions of $\boldsymbol{\mathcal{S}}(K)$. In the Petrov-Galerkin method, we need to solve (33) for $\mathbf{U}$ and compute $\boldsymbol{U}_h$ by (32) for any given $\widehat{\boldsymbol{u}}_h$.

It follows from (33) that $\mathbf{U}$ is the minimizer of

$$\mathbf{U}(\widehat{\mathbf{u}}) := \arg\min_{\mathbf{W}\in\mathbb{R}^n} \mathbf{r}(\mathbf{W};\widehat{\mathbf{u}})^T \mathbf{X}^{-1} \mathbf{r}(\mathbf{W};\widehat{\mathbf{u}}). \tag{35}$$

This minimization statement shows the connection between the Petrov-Galerkin method and the weighted least-squares method. Moreover, we can easily show that

$$\boldsymbol{U}_h(\widehat{\boldsymbol{u}}_h) := \arg \inf_{\boldsymbol{W}\in\boldsymbol{\mathcal{U}}(K)} \sup_{\boldsymbol{Z}\in\boldsymbol{\mathcal{S}}(K)} \frac{r_K(\boldsymbol{W}, \boldsymbol{Z}; \widehat{\boldsymbol{u}}_h)}{\|\boldsymbol{Z}\|_K}, \tag{36}$$

where $\|\boldsymbol{Z}\|_K = (\boldsymbol{Z}, \boldsymbol{Z})_K^{1/2}$. This inf-sup statement shows the connection between the Petrov-Galerkin method and the minimal residual method. Indeed, in our particular context, the three methods are equivalent.

As mentioned earlier, the PG method is not conservative. In order to make the method conservative, we propose to solve the following constrained optimization problem

$$\begin{aligned} \boldsymbol{U}_h(\widehat{\boldsymbol{u}}_h) := \arg \inf_{\boldsymbol{W}\in\boldsymbol{\mathcal{U}}(K)} \sup_{\boldsymbol{Z}\in\boldsymbol{\mathcal{S}}(K)} & \frac{r_K(\boldsymbol{W}, \boldsymbol{Z}; \widehat{\boldsymbol{u}}_h)}{\|\boldsymbol{Z}\|_K} \\ \text{s.t.} \quad & r_K\big(\boldsymbol{W}, \boldsymbol{C}; \widehat{\boldsymbol{u}}_h\big) = 0 \end{aligned} \tag{37}$$

where $\boldsymbol{C}$ is a non-zero constant function. This completes the definition of the local problem for the HDPG method.

B. Implementation

1. *The global matrix system*

We now describe the steps taken to implement the HDPG method. We first apply the Newton-Raphson method to linearize the global weak formulation (25): Given the current iterate $\widehat{\boldsymbol{u}}_h^*$, we find the Newton increment $\delta\widehat{\boldsymbol{u}}_h \in \boldsymbol{\mathcal{M}}_h^k$ as a solution of

$$a(\delta\widehat{\boldsymbol{u}}_h, \boldsymbol{\mu}; \widehat{\boldsymbol{u}}_h^*) = \ell(\boldsymbol{\mu}; \widehat{\boldsymbol{u}}_h^*), \quad \forall\, \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}_h^k, \tag{38}$$

where

$$\begin{aligned} a(\boldsymbol{\eta}, \boldsymbol{\mu}; \widehat{\boldsymbol{u}}_h^*) &= \left\langle \left(\frac{\partial \widehat{\boldsymbol{q}}_h}{\partial \boldsymbol{Q}_h} \frac{\partial \boldsymbol{Q}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{q}}_h}{\partial \boldsymbol{u}_h} \frac{\partial \boldsymbol{u}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{q}}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) \right) \boldsymbol{\eta}, \boldsymbol{\mu} \right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} \\ &\quad + \left\langle \left(\frac{\partial \widehat{\boldsymbol{b}}_h}{\partial \boldsymbol{Q}_h} \frac{\partial \boldsymbol{Q}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{b}}_h}{\partial \boldsymbol{u}_h} \frac{\partial \boldsymbol{u}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{b}}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) \right) \boldsymbol{\eta}, \boldsymbol{\mu} \right\rangle_{\partial\Omega}, \end{aligned} \tag{39}$$

$$\ell(\boldsymbol{\mu}; \widehat{\boldsymbol{u}}_h^*) = -\big\langle \widehat{\boldsymbol{q}}_h(\widehat{\boldsymbol{u}}_h^*), \boldsymbol{\mu} \big\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} - \big\langle \widehat{\boldsymbol{b}}_h(\widehat{\boldsymbol{u}}_h^*), \boldsymbol{\mu} \big\rangle_{\partial\Omega}, \tag{40}$$

for all $\boldsymbol{\mu}, \boldsymbol{\eta} \in \boldsymbol{\mathcal{M}}_h^k$. Note here that we have used the chain rule to calculate the derivative of $\widehat{\boldsymbol{q}}_h$ and $\widehat{\boldsymbol{b}}_h$ with respect to $\widehat{\boldsymbol{u}}_h$.

The weak formulation (38) gives rise to a matrix system of the form

$$\mathbf{A}\ \delta\widehat{\mathbf{u}} = \mathbf{f}, \tag{41}$$

where $\delta\widehat{\mathbf{u}}$ represents the vector of degrees of freedom of $\delta\widehat{\boldsymbol{u}}_h$. The matrix $\mathbf{A}$ and vector $\mathbf{f}$ can be formed by the usual finite element assembly procedure once the elemental matrices and vectors are computed as follows

$$\begin{aligned}
\mathrm{A}_{ij}^K &= \left\langle \left(\frac{\partial \widehat{\boldsymbol{q}}_h}{\partial \boldsymbol{Q}_h} \frac{\partial \boldsymbol{Q}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{q}}_h}{\partial \boldsymbol{u}_h} \frac{\partial \boldsymbol{u}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{q}}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) \right) \boldsymbol{\xi}_j, \boldsymbol{\xi}_i \right\rangle_{\partial \mathcal{T}_h \setminus \partial\Omega} \\
&\quad + \left\langle \left(\frac{\partial \widehat{\boldsymbol{b}}_h}{\partial \boldsymbol{Q}_h} \frac{\partial \boldsymbol{Q}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{b}}_h}{\partial \boldsymbol{u}_h} \frac{\partial \boldsymbol{u}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) + \frac{\partial \widehat{\boldsymbol{b}}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*) \right) \boldsymbol{\xi}_j, \boldsymbol{\xi}_i \right\rangle_{\partial\Omega}, \qquad (42) \\
\mathrm{f}_i^K &= -\left\langle \widehat{\boldsymbol{q}}_h(\widehat{\boldsymbol{u}}_h^*), \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h \setminus \partial\Omega} - \left\langle \widehat{\boldsymbol{b}}_h(\widehat{\boldsymbol{u}}_h^*), \boldsymbol{\xi}_i \right\rangle_{\partial\Omega}, \qquad (43)
\end{aligned}$$

for $1 \le i, j \le N$. In order to compute the elemental matrices and vectors we need to obtain $(\boldsymbol{Q}_h(\widehat{\boldsymbol{u}}_h^*), \boldsymbol{u}_h(\widehat{\boldsymbol{u}}_h^*))$ and their sensitivities $(\frac{\partial \boldsymbol{Q}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*), \frac{\partial \boldsymbol{u}_h}{\partial \widehat{\boldsymbol{u}}_h}(\widehat{\boldsymbol{u}}_h^*))$ by solving the local problem as we discuss below.

2. Solution of the local problem

We recall that the vector of degrees of freedom of $(\boldsymbol{Q}_h(\widehat{\boldsymbol{u}}_h^*), \boldsymbol{u}_h(\widehat{\boldsymbol{u}}_h^*))$ satisfies

$$\begin{aligned}
\mathbf{U}(\widehat{\mathbf{u}}^*) := \arg \min_{\mathbf{W} \in \mathbb{R}^n} \mathbf{r}(\mathbf{W}; \widehat{\mathbf{u}}^*)^T \mathbf{X}^{-1} \mathbf{r}(\mathbf{W}; \widehat{\mathbf{u}}^*) \\
\text{s.t. } \mathbf{c}^T \mathbf{r}(\mathbf{W}; \widehat{\mathbf{u}}^*) = 0,
\end{aligned} \qquad (44)$$

where $\mathbf{c}$ represents the vector of degrees of freedom for the constant basis function. We now introduce a Lagrange multiplier λ to relax the constraint and define the Lagrangian as

$$L(\mathbf{U}, \lambda) := \frac{1}{2} \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*)^T \mathbf{X}^{-1} \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*) + \lambda \mathbf{c}^T \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*). \qquad (45)$$

By setting the derivative of the Lagrangian to zero, we obtain the following nonlinear system of equations

$$\begin{aligned}
\mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*)^T \mathbf{X}^{-1} \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*) + \lambda \mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*)^T \mathbf{c} = 0, \\
\boldsymbol{c}^T \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*) = 0.
\end{aligned} \qquad (46)$$

This system has the structure of a nonlinear saddle point problem.

We now use the Newton-Raphson method to solve the above system: Given the current iterate $(\overline{\mathbf{U}}, \bar{\lambda})$, we find the increment $(\delta\mathbf{U}, \delta\lambda)$ as a solution of

$$\begin{aligned}
\big(\mathbf{J}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*)^T \mathbf{X}^{-1} \mathbf{J}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*) + \mathbf{H}(\overline{\mathbf{U}}, \bar{\lambda}; \widehat{\mathbf{u}}^*)\big) \delta\mathbf{U} + \mathbf{J}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*)^T \mathbf{c} \delta\lambda = -\mathbf{J}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*)^T (\mathbf{X}^{-1} \mathbf{r}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*) + \bar{\lambda} \mathbf{c}) \\
\boldsymbol{c}^T \mathbf{J}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*) \delta\mathbf{U} = -\boldsymbol{c}^T \mathbf{r}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*)
\end{aligned} \qquad (47)$$

where $\mathbf{H}(\overline{\mathbf{U}}, \bar{\lambda}; \widehat{\mathbf{u}}^*)$ is a matrix whose column vectors are given by

$$\mathbf{H}_i(\overline{\mathbf{U}}, \bar{\lambda}; \widehat{\mathbf{u}}^*) = \frac{\partial \mathbf{J}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*)}{\partial \mathrm{U}_i} (\mathbf{X}^{-1} \mathbf{r}(\overline{\mathbf{U}}; \widehat{\mathbf{u}}^*) + \bar{\lambda} \mathbf{c}), \quad 1 \le i \le n. \qquad (48)$$

The computation of this matrix requires the first derivatives of the Jacobian matrix $\mathbf{J}$ with respect to $\mathbf{U}$, which in turn requires the second derivatives of the residual vector $\mathbf{r}$. We then update the solution as

$$(\overline{\mathbf{U}}, \bar{\lambda}) = (\overline{\mathbf{U}}, \bar{\lambda}) + \alpha(\delta\mathbf{U}, \delta\lambda), \qquad (49)$$

where α is a Newton step size to ensure that the cost function is reduced at every Newton iteration. $\mathbf{H}$ is a necessary term in order to have quadratic convergence, however, in practice, the initial condition may lie outside the region of attraction and the full Newton iteration may diverge. To avoid this, during the first iterations, $\mathbf{H}$ is set to zero and only used once the length of $(\delta\mathbf{U}, \delta\lambda)$ is under a certain threshold. We repeat the process until convergence and then set $(\mathbf{U}, \lambda) := (\overline{\mathbf{U}}, \bar{\lambda})$.

Finally, in order to compute the sensitivities, we differentiate (46) with respect to $\widehat{\mathbf{u}}^*$ to obtain

$$\begin{aligned}
\big(\mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*)^T \mathbf{X}^{-1} \mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*) + \mathbf{H}(\mathbf{U}, \lambda; \widehat{\mathbf{u}}^*)\big) \frac{\partial \mathbf{U}}{\partial \widehat{\mathbf{u}}^*} + \mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*)^T \mathbf{c} \frac{\partial \lambda}{\partial \widehat{\mathbf{u}}^*} = -\mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*)^T \mathbf{X}^{-1} \frac{\partial \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*)}{\partial \widehat{\mathbf{u}}^*} - \mathbf{G}(\mathbf{U}; \widehat{\mathbf{u}}^*) \\
\boldsymbol{c}^T \mathbf{J}(\mathbf{U}; \widehat{\mathbf{u}}^*) \frac{\partial \mathbf{U}}{\partial \widehat{\mathbf{u}}^*} = -\boldsymbol{c}^T \frac{\partial \mathbf{r}(\mathbf{U}; \widehat{\mathbf{u}}^*)}{\partial \widehat{\mathbf{u}}^*}
\end{aligned} \qquad (50)$$

where $\mathbf{G}(\mathbf{U};\widehat{\mathbf{u}}^*)$ is a matrix whose column vectors are given by

$$\mathbf{G}_j(\mathbf{U};\widehat{\mathbf{u}}^*) = \frac{\partial \mathbf{J}(\mathbf{U};\widehat{\mathbf{u}}^*)}{\partial \widehat{\mathrm{u}}_j}\mathbf{X}^{-1}\mathbf{r}(\mathbf{U};\widehat{\mathbf{u}}^*), \quad 1 \le j \le N. \tag{51}$$

We note that the left-hand side of the system (50) is the same as that of (47) at convergence.

C. Extension to time-dependent problems

We extend the HDPG method described above to the time-dependent hyperbolic problem (1). We shall consider the backward Euler method for time integration as higher-order BDF schemes admit a similar implementation. As before, we denote by $(\boldsymbol{Q}_h^n, \boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n)$ the numerical approximations to $(\boldsymbol{Q}(t^n), \boldsymbol{u}(t^n), \widehat{\boldsymbol{u}}(t^n))$ at time $t^n = n\Delta t^n$, where Δt^n is a timestep size at level n. We seek an approximation $\widehat{\boldsymbol{u}}_h^n \in \boldsymbol{\mathcal{M}}_h^k$ such that

$$\big\langle \widehat{\boldsymbol{q}}_h^n(\widehat{\boldsymbol{u}}_h^n), \boldsymbol{\mu}\big\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \big\langle \widehat{\boldsymbol{b}}_h^n(\widehat{\boldsymbol{u}}_h^n), \boldsymbol{\mu}\big\rangle_{\partial\Omega} = 0, \quad \forall\, \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}_h^k, \tag{52}$$

where the interior numerical flux is

$$\widehat{\boldsymbol{q}}_h^n(\widehat{\boldsymbol{u}}_h^n) = \boldsymbol{F}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h(\widehat{\boldsymbol{u}}_h^n))\cdot\boldsymbol{n} + \boldsymbol{S}^n(\boldsymbol{u}_h^n(\widehat{\boldsymbol{u}}_h^n) - \widehat{\boldsymbol{u}}_h^n), \tag{53}$$

and the boundary numerical flux $\widehat{\boldsymbol{b}}_h^n$ depends on the particular boundary conditions. Both the boundary numerical flux $\widehat{\boldsymbol{b}}_h^n$ and the stabilization matrix $\boldsymbol{S}^n$ are chosen the same as those for the HDG method.

We follow the previous method of lines to define a local problem for any element $K \in \mathcal{T}_h$. In particular, using the backward Euler method to discretize the time derivative we find $(\boldsymbol{Q}_h^n, \boldsymbol{u}_h^n) \in \boldsymbol{\mathcal{U}}(K)$ such that

$$\begin{aligned}
(\boldsymbol{Q}_h^n, \boldsymbol{E})_K + (\boldsymbol{u}_h^n, \nabla\cdot\boldsymbol{E})_K - \langle \widehat{\boldsymbol{u}}_h^n, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial K} &= 0,\\
\Big(\frac{\boldsymbol{u}_h^n - \boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{w}\Big)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h^n, \boldsymbol{Q}_h^n), \nabla\boldsymbol{w})_K + \langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n)\cdot\boldsymbol{n} + \boldsymbol{S}^n(\boldsymbol{u}_h^n - \widehat{\boldsymbol{u}}_h^n), \boldsymbol{w}\rangle_{\partial K} - (\boldsymbol{f}^n, \boldsymbol{w})_K &= 0.
\end{aligned} \tag{54}$$

We now define $\boldsymbol{U}_h^n \equiv (\boldsymbol{Q}_h^n, \boldsymbol{u}_h^n)$ and introduce the associated residual

$$\begin{aligned}
r_K^n\big(\boldsymbol{U}_h^n, \boldsymbol{V}; \widehat{\boldsymbol{u}}_h^n\big) &= (\boldsymbol{Q}_h^n, \boldsymbol{E})_K + (\boldsymbol{u}_h^n, \nabla\cdot\boldsymbol{E})_K - \langle \widehat{\boldsymbol{u}}_h^n, \boldsymbol{E}\cdot\boldsymbol{n}\rangle_{\partial K}\\
&\quad + \Big(\frac{\boldsymbol{u}_h^n - \boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{w}\Big)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h^n, \boldsymbol{Q}_h^n), \nabla\boldsymbol{w})_K\\
&\quad + \langle \boldsymbol{F}(\widehat{\boldsymbol{u}}_h^n, \boldsymbol{Q}_h^n)\cdot\boldsymbol{n} + \boldsymbol{S}^n(\boldsymbol{u}_h^n - \widehat{\boldsymbol{u}}_h^n), \boldsymbol{w}\rangle_{\partial K} - (\boldsymbol{f}^n, \boldsymbol{w})_K,
\end{aligned} \tag{55}$$

for any given $\boldsymbol{Z} = (\boldsymbol{E}, \boldsymbol{w}) \in \boldsymbol{\mathcal{S}}(K)$. The local problem involves solving

$$\begin{aligned}
\boldsymbol{U}_h^n(\widehat{\boldsymbol{u}}_h^n) := \arg \inf_{\boldsymbol{W}\in\boldsymbol{\mathcal{U}}(K)} \sup_{\boldsymbol{Z}\in\boldsymbol{\mathcal{S}}(K)} &\frac{r_K^n(\boldsymbol{W}, \boldsymbol{Z}; \widehat{\boldsymbol{u}}_h^n)}{\|\boldsymbol{Z}\|_K}\\
\text{s.t.}\quad & r_K^n\big(\boldsymbol{W}, \boldsymbol{C}; \widehat{\boldsymbol{u}}_h^n\big) = 0
\end{aligned} \tag{56}$$

where $\boldsymbol{C}$ is a non-zero constant function.

It is obvious that the implementation can be carried out in the same way as the preceding one. We omit the detailed steps to save space. Below, we present numerical results to demonstrate the performance of the HDPG method.

V. Results in 1D

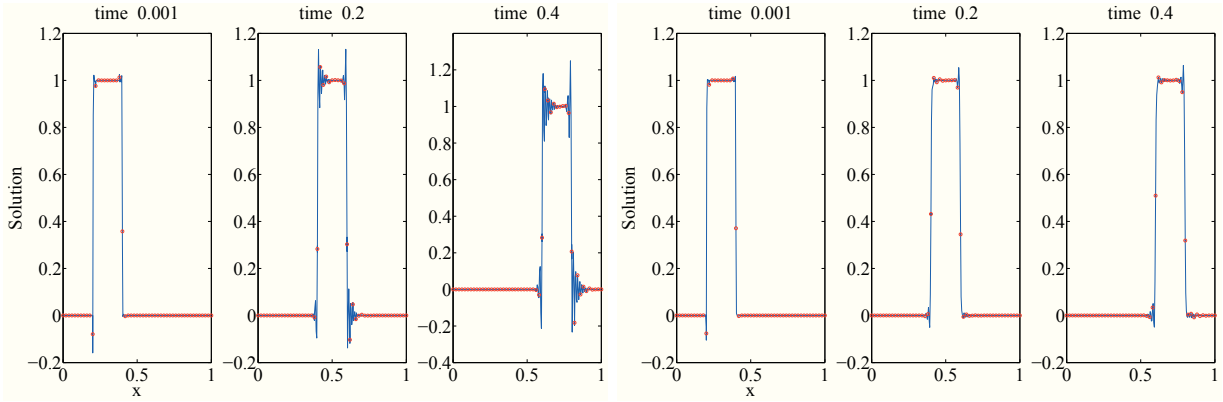
In this section, we present some 1D results obtained with the HDPG method applied to simple linear and non-linear hyperbolic conservation laws. The purpose is to illustrate the benefits that HDPG may yield in simple problems where discontinuous solutions are present.

A. Linear Convection

First of all, we apply the HDPG methodology to a linear convection problem with a sharp initial condition. We consider the equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{57}$$

where u is the magnitude advected in time along the x axis with a unit velocity. In this case, we are interested in discontinuous initial conditions ($u(x,0) = 1$ for $x \in (0.2, 0.4)$, $u(x,0) = 0$ elsewhere) and how the solution evolves in time. The boundary conditions are set to homogeneous Dirichlet at $x = 0$. Figure 1, shows the results for the convection of a piecewise constant function with polynomials of order $k = 5$ and 50 elements in the domain using HDG and HDPG (with $\Delta k = 5$). As we would would expect, oscillations appear near the discontinuities due to the lack of a dissipation mechanism. However, we note that the HDPG is far less oscillatory than the HDG solution for the same problem parameters.



(a) HDG

(b) HDPG with $\triangle k = 5$.

Figure 1: Convection of a hat function using HDG for polynomials of degree $k = 5$ and using a BDF scheme for time stepping with $\triangle t = 0.001$.

For this case, the test space is enriched with elements up to $k + \Delta k = 10$. As described in the previous section, this determines the space in which the maximization of stability takes place. Numerical experience indicates that beyond a certain Δk, the solutions do not improve further. Furthermore, the bigger the test space, the more expensive the computation becomes. Therefore, Δk is typically chosen so that no further stability is achieved.

B. Burgers equation

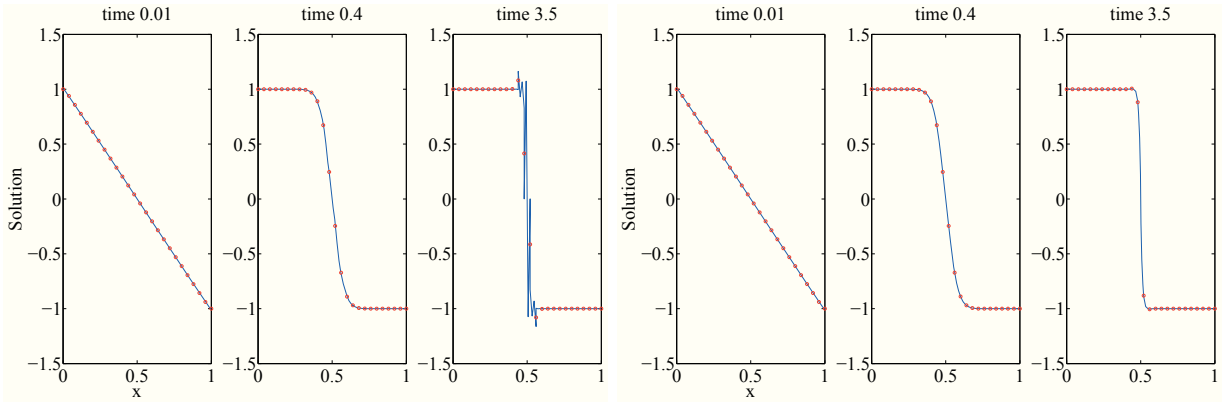
Here, we use the HDPG method to solve a non-linear problem that develops discontinuities in finite time even if the initial condition is continuous. We consider Burgers equation in 1D:

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2/2)}{\partial x} - \varepsilon \frac{\partial^2 u}{\partial x^2} = 0 \tag{58}$$

1. *Inviscid steady shock*

We compare the HDG and HDPG methods for the inviscid equation ($\varepsilon = 0$) with an initial smooth solution that evolves in time (BDF3 with $\Delta t = 0.01$) until a steady shock is formed. In this case, the boundary conditions are fixed ($u(0,t) = 1$, $u(1,t) = -1$) and the initial condition reads $u(x,0) = 1 - 2x$. The results for a discretization of 25 elements with order $k = 3$ are shown in figure 2. The last snapshot in each plot shows the solution at the stage beyond which the solution becomes stationary.

As we can notice, HDPG is capable of capturing the shock in one element plus a slight contribution of the neighbors (see 2b) without generating spurious oscillation. Meanwhile, HDG does produce a strong wiggling of the solution in the elements close to the shock.

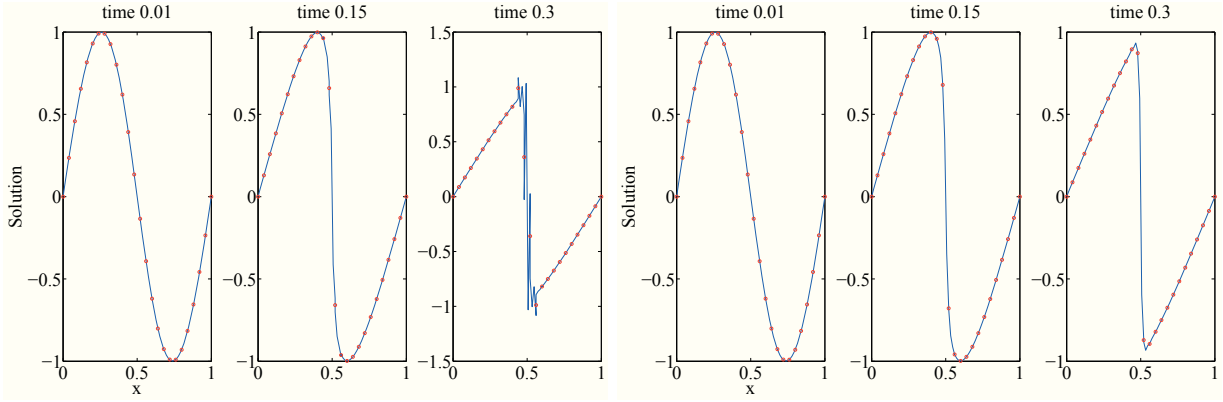


(a) HDG (b) HDPG with $\triangle k = 4$

Figure 2: Results for an inviscid steady shock with $k = 3$ and 25 elements.

2. Inviscid sawtooth shock profile

In this case, we consider also the inviscid equation with boundary conditions $u(0,t) = u(1,t) = 0$ and the initial condition $u(x,0) = \sin(2\pi x)$. The time stepping is the BDF3 method with the same time step ($\triangle t = 0.01$). We use 25 elements with $k = 3$. The results are shown in figure (3) for different times. It is observed that the HDG solution generates oscillations in the elements close to the shock but the HDPG solution is capable of capturing the discontinuity without any oscillations.



(a) HDG (b) HDPG with $\triangle k = 4$

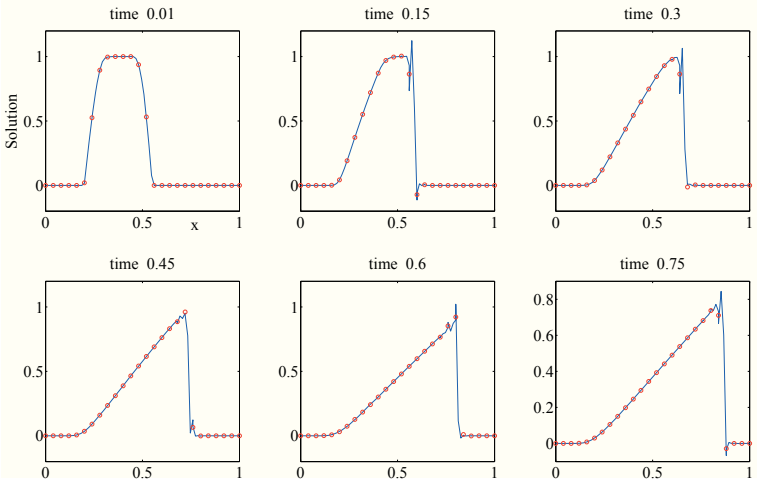
Figure 3: Results for an inviscid shock with $k = 3$ and 25 elements.

3. Viscous moving shock profile

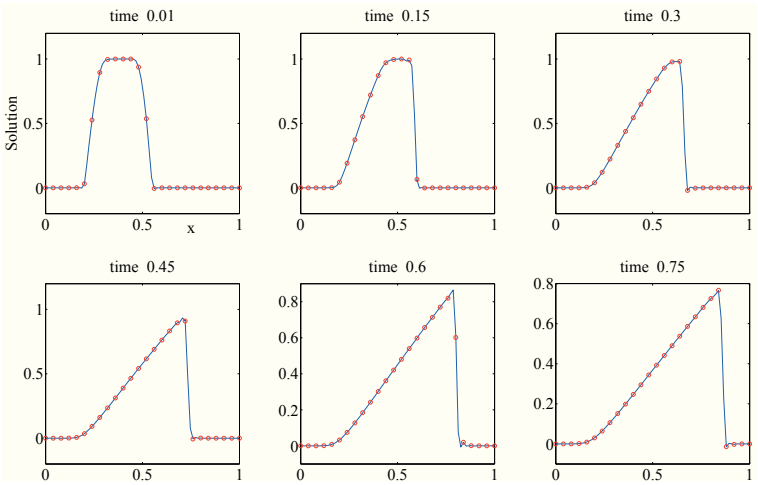
Here, we consider the viscous Burgers equation in the case where the viscosity is insufficient to resolve the discontinuities and compare the behavior of the HDG and HDPG methods. The initial conditions in this case are: $u(x,0) \simeq H(x-0.2) - H(x-0.5)$ where the Heaviside step function H has been smoothed in order to avoid oscillatory behavior in the initial step. This initial condition generates a shock that travels to the right of the domain followed by an expansion wave. The amount of viscosity added corresponds to an element Peclet number of 10 ($Pe|_{\text{element}} = \frac{h}{k}\frac{u}{\varepsilon} = 10$). For the time stepping, a BDF3 formula with $\Delta t = 0.01$ is used.

Figure 4 shows the results for both HDG and HDPG in this case. Again, the HDG scheme results in an oscillatory solution whereas the HDPG solution is non-oscillatory and has the correct propagation speed, as we would expect from a conservative scheme. We point out that for this moving shock problem, the HDPG solution without viscosity ($\varepsilon = 0$) exhibits oscillations although smaller that those obtained with

HDG. However, a smaller amount of viscosity (about five to ten times smaller that that required for the HDG method) is required for HDPG to eliminate oscillations.



(a) HDG



(b) HDPG with $\Delta k = 4$

Figure 4: Evolution of a smoothed hat function for the Burgers equation with $k = 3$ and 25 elements. Time stepping is carried out through a BDF3 formula with $\Delta t = 0.01$. The viscosity is selected so that $(Pe|_{\text{element}} = \frac{h}{k}\frac{u}{\varepsilon} = 10)$

VI. Results in 2D

In this section, we present two dimensional examples, first for a 2D scalar Burgers equation and then for the compressible Navier-Stokes equations.

A. Burgers equation in 2D

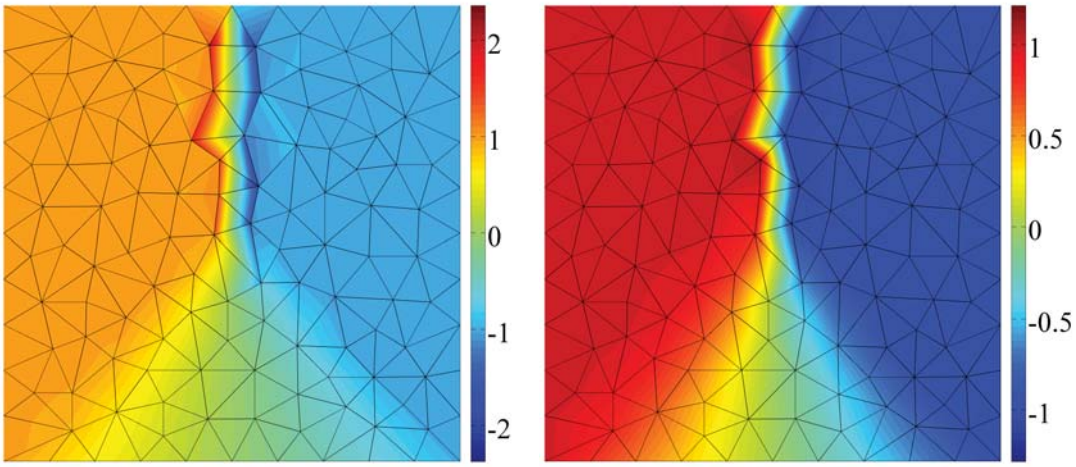
We consider the problem

$$\nabla \cdot \left[\begin{Bmatrix} u^2/2 \\ u \end{Bmatrix} - \varepsilon \nabla u \right] = 0 \quad \text{in } \Omega \tag{59}$$

with the boundary conditions:

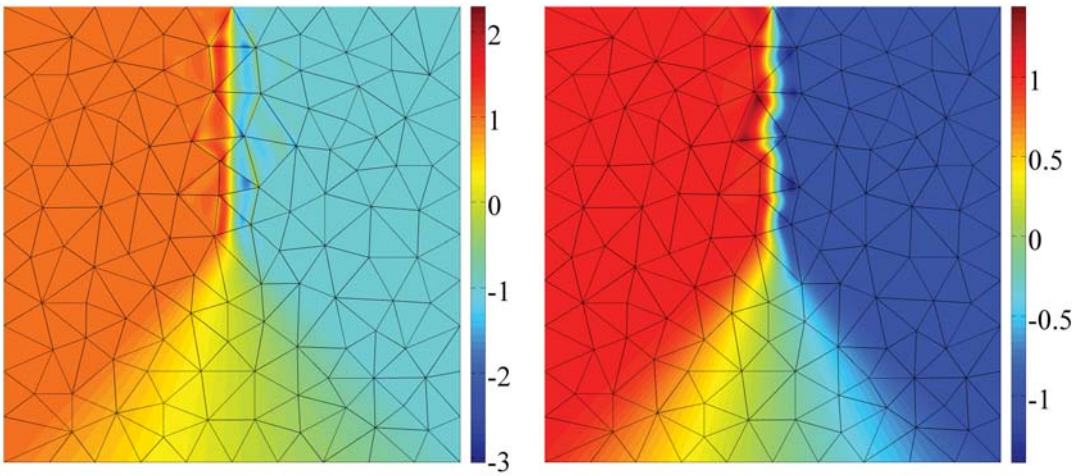
$$u(0, y) = 1, \qquad u(x, 0) = 1 - 2x, \qquad u(1, y) = -1. \tag{60}$$

We compare the solutions obtained using the HDG and HDPG methods. The exact solution consists of a compression wave that, given those boundary conditions, becomes a shock for $y = 0.5$ and continues as a shock until $y = 1$. We consider the purely inviscid case ($\varepsilon = 0$, $Pe = \infty$) and the case with very small viscosity corresponding to an element Peclet number of 100 ($Pe|_{\text{element}} = 100$). The results for the inviscid case for $k = 1$ are plotted in figure 5. We see that although overshoots are obtained for the HDG and HDPG solutions, they are over 100% in the HDG case and only about 20% in the HDG case. We then consider the viscous case with higher order polynomials $k = 3$ and ($Pe|_{\text{element}} = 100$). Again, comparing the HDG and HDPG solutions, we see that in the HDPG case the overshoot near the shock is several times smaller and the solution and the small oscillation is much more localized.



(a) HDG (b) HDPG with $\Delta k = 2$

Figure 5: Results for $k = 1$ and $Pe|_{\text{element}} = \infty$ for the case of an structured mesh



(a) HDG (b) HDPG with $\Delta k = 4$

Figure 6: Results for $k = 3$ and $Pe|_{\text{element}} = 100$ for the case of an structured mesh

We also show in figure 7 the solution obtained with HDPG with $k = 5$ and $Pe|_{\text{element}} = 100$. In this same setting, HDG *does not* converge.

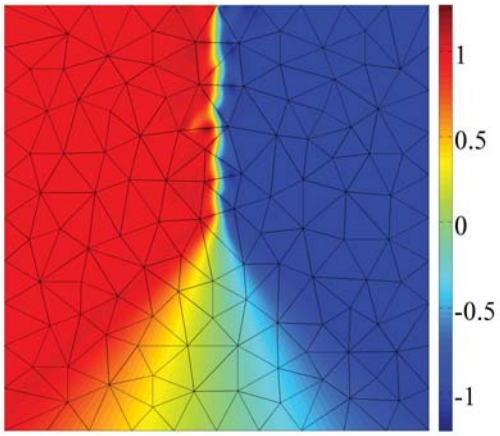


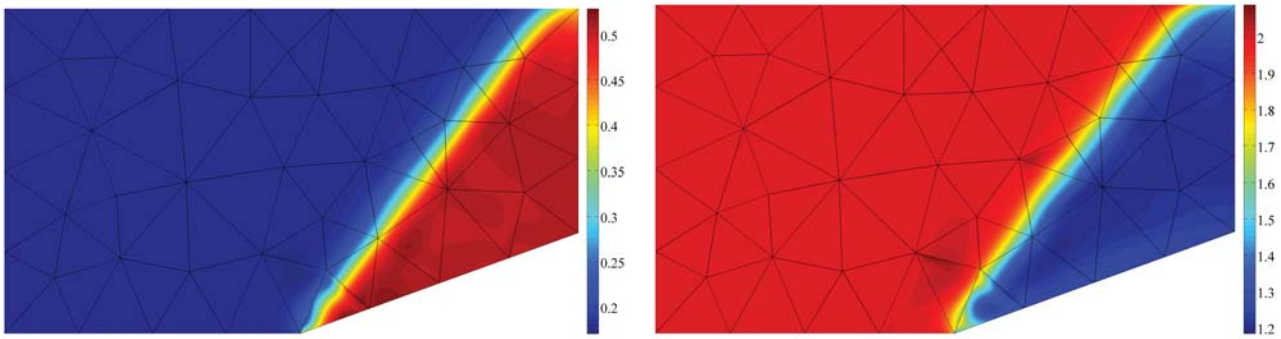
Figure 7: Result for $k = 5$ using HDPG with $\Delta k = 6$ and $Pe|_{\text{element}} = 100$.

B. Navier-Sokes equations in 2D

We consider the Navier-Stokes equations with constant values for the viscosity and thermal conductivity. The viscosity μ is given as a function of the element Peclet number, defined as $Pe|_{\text{element}} = \frac{h}{k}\frac{|\mathbf{v}|}{\mu}$, where h is a representative element size (kept constant over the domain) and $|\mathbf{v}|$ is the magnitude of the velocity. The thermal conductivity is chosen so that the Prandtl number is equal to 0.75. Since our objective here is to assess the merits of the HDPG method to capture shocks we impose slip-wall boundary conditions and employ values of the viscosity which are insufficient to fully resolve the viscous shocks. We note that this would not be a valid strategy to compute Euler solutions since too much dissipation is being applied in points where it is not required such as smooth regions of the flow, and in general the solution will not be as accurate. A better shock capturing strategy would apply viscosity selectively in regions where it is required.[2,18,31] This is subject of current research.

1. Wedge in supersonic flow

The first example we present is the flow past a 20° wedge immerse in a supersonic stream at $M_\infty = 2$. In this setting, an attached oblique shock forms that deflects the free stream flow to make it parallel to the wedge. Before and after the shock the flow is supersonic and uniform. Because of symmetry, only half of the domain is considered. Even though the HDPG is capable of producing a solution in the inviscid limit, this solution presents oscillations in the shock region that propagate downstream of the shock. In order to avoid this, a small amount of viscosity is introduced so that $Pe|_{\text{element}} = 10$. The results are plotted in figure 8. We see that except for the singularity at the wedge tip, the solution is captured cleanly using fourth order $k = 4$ polynomial elements. Note that at the wall, a zero shear stress condition has been imposed which means that the viscous stresses are only significant at the shock.

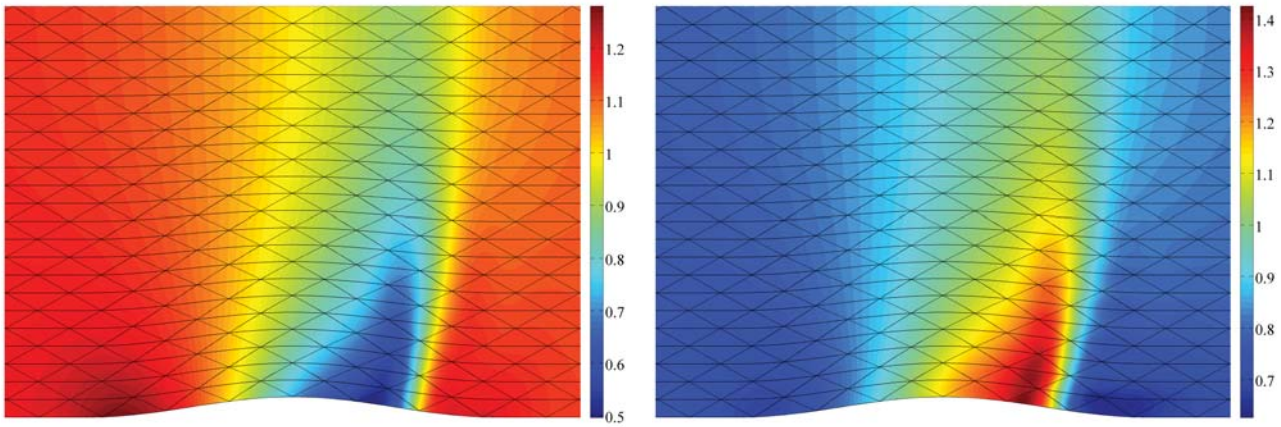


(a) Pressure (b) Mach number

Figure 8: Supersonic flow at $M_\infty = 2$ past a 20° wedge using HDPG with $k = 4$ and $\Delta k = 2$ and viscosity equivalent to $Pe|_{\text{element}} = 10$.

2. *Gaussian bump in a channel*

The second case is a transonic flow over a Gaussian bump. The inflow Mach number is $M_\infty = 0.8$ and the height of the bump represents 5% of the total height of the channel. The curvature of the bump is enough to generate a supersonic region on top of it that recovers to subsonic through a normal shock. As in the previous case, the viscosity is set so that $Pe|_{\text{element}} = 10$ and a slip-wall zero shear stress condition is imposed at the top and bottom walls of the channel. The results are plotted in figure 9 and further confirm the sub-cell shock capturing possibilities of HDPG with an order of magnitude less viscosity than other shock capturing DG methods.



(a) Pressure (b) Mach number

Figure 9: Transonic flow in a channel with a Gaussian bump in the lower surface using HDPG with $k = 3$ and $\Delta k = 2$. At the inlet $M_\infty = 0.8$. The viscosity is set so that $Pe|_{\text{element}} = 10$. The channel's length is roughly three times the length shown.

3. *Trefftz airfoil*

Finally, we present the solution for a Trefftz airfoil at $M_\infty = 0.8$ and zero angle of attack. As in the previous example, the flow presents a supersonic region in the upper surface that recovers to subsonic through a normal shock. Again, the viscosity is set so that $Pe|_{\text{element}} = 10$. The result for the pressure is plotted in figure 10. Notice how the HDPG method captures the shock within one element, even in a highly under-resolved mesh as the one used.

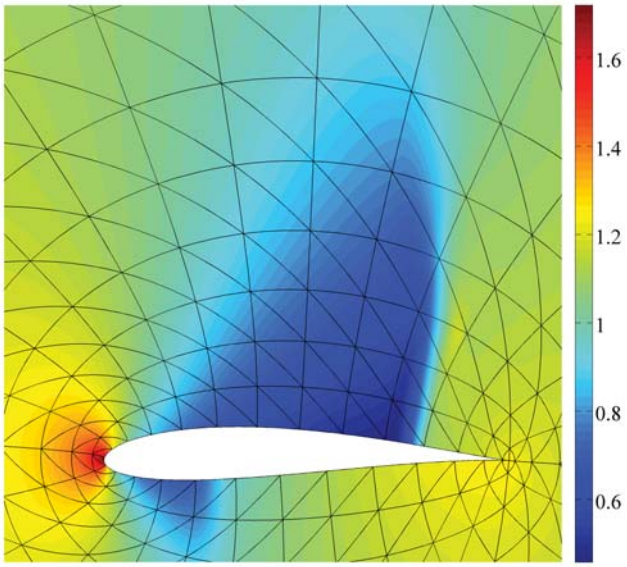


Figure 10: Pressure past a Trefftz airfoil in transonic regime at $M_\infty = 0.8$ and zero angle of attack using HDPG with $k = 3$ and $\Delta k = 2$ and $Pe|_{\text{element}} = 10$

VII. Conclusions and future work

We have presented a new hybridized discontinuous Galerkin method in which a minimization problem is solved locally for each problem. The proposed algorithm inherits the low number of globally coupled unknowns of the HDG methods but it is much more stable. In the presence of discontinuities, the required viscosity for HDPG method to capture shocks is an order of magnitude smaller than that required by HDG or other DG methods.

Future research will include the combination of the HDPG method with a discontinuity sensor to restrict the application of artificial viscosity to the elements affected by the shock. We will also consider more advanced and efficient techniques to solve the local minimization statement for the local problem.

VIII. Aknowledgements

The authors would like to acknowledge the support of the Singapore-MIT Alliance and the Air Force Office of Scientific Research under the MURI project on Biologically Inspired Flight for partially supporting this work. The first author would like to express his gratitude to the Fundacion CajaMadrid for the Graduate Studies Scholarship that supported his work.

References

[1]D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001.

[2]G. E. Barter and D. L. Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *J. Comput. Phys.*, 229(5):1810–1827, 2010.

[3]F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997.

[4]C. Baumann and J. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 175:311–341, 1999.

[5]B. Cockburn and J. Gopalakrishnan. The derivation of hybridizable discontinuous Galerkin methods for Stokes flow. *SIAM J. Numer. Anal.*, 47:1092–1125, 2009.

[6]B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 47:1319–1365, 2009.

[7]B. Cockburn, J. Gopalakrishnan, N. C. Nguyen, J. Peraire, and F.-J. Sayas. Analysis of HDG methods for Stokes flow. *Math. Comp.* To appear.

[8]B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for convection-diffusion systems. *SIAM J. Numer. Anal.*, 35:2440–2463, 1998.

[9]B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2001.

[10]C. Daraio, V. F. Nesterenko, E. B. Herbold and S. Jin Energy trapping and shock desintegration in composite granular medium *Phys. Rev. Lett.*, 92, 6, 2006.

[11]L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. part I: The transport equation. *Comput. Methods Appl. Mech. Engrg.*,199(23-24):1558–1572, 2010.

[12]L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. part I: Optimal test functions. *Numer. Methods Partial Differential Eq.*, 27(1): 70–105, 2011.

[13]S. Güzey, B. Cockburn, and H. Stolarski. The embedded discontinuous Galerkin methods: Application to linear shells problems. *Internat. J. Numer. Methods Engrg.*, 70:757-790, 2007.

[14]R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183:508–532, 2002.

[15]J. S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids i. time-domain solution of maxwell's equations. *J. Comput. Phys.*, 181(1):186–221, 2002.

[16]C. M. Klaij, J. J. W. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 217(2):589–611, 2006.

[17]I. Lomtev and G. E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 29:587–603, 1999.

[18]N. C. Nguyen and J. Peraire. A shock capturing HDG method for compressible flows. Submitted.

[19]N. C. Nguyen, J. Peraire, and B. Cockburn. A comparison of HDG methods for Stokes flow. *Journal of Scientific Computing*, 45(1-3):215–237, 2010.

[20]N. C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. Submitted.

[21]N. C. Nguyen, J. Peraire, and B. Cockburn. Implicit high-order HDG Methods for acoustics and elastodynamics Submitted.

[22]N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method the incompressible Navier-Stokes equations. *J. Comput. Phys.* Accepted.

[23]N. C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods, *in Spectral and High Order Methods for Partial Differential Equations*, (Editors J. S. Hesthaven and E. M. Ronquist). *Lecture Notes in Computational Science and Engineering*, 2011, Volume 76, pages 63-84.

[24]N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *J. Comput. Phys.*, 228:3232–3254, 2009.

[25]N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *J. Comput. Phys.*, 228:8841–8855, 2009.

[26]N. C. Nguyen, J. Peraire, and B. Cockburn. A hybridizable discontinuous Galerkin method for Stokes flow. *Comput. Methods Appl. Mech. Engrg.*, 199:582–597, 2010.

[27]N. C. Nguyen, J. Peraire, and B. Cockburn. A hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations (AIAA Paper 2010-362). In *Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, Janury 2010.

[28]J. Peraire, N. C. Nguyen, and B. Cockburn. A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations (AIAA Paper 2010-363). In *Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, Janury 2010.

[29]J. Peraire, N. C. Nguyen, and B. Cockburn. An Embedded discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. In *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, June 2011.

[30]J. Peraire and P.-O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[31]P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-112, Reno, NV, January 2006.

[32]W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[33]P. L. Roe Approximate riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.*, 43:357-372, 1981

[34]S.-C. Soon, B. Cockburn, and H. K. Stolarski. A hybridizable discontinuous Galerkin method for linear elasticity. *International Journal for Numerical Methods in Engineering*, 80(8):1058–1092, 2009.