51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0689

A Hybridized Multiscale Discontinuous Galerkin Method for Compressible Flows

N. C. Nguyen,[*] X. Roca,[†] D. Moro,[‡] and J. Peraire[§]
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

We introduce a hybridized multiscale discontinuous Galerkin (HMDG) method for the numerical solution of compressible flows. The HMDG method is developed upon extending the hybridizable discontinuous Galerkin (HDG) method presented in [30]. The extension is carried out by modifying the local approximation spaces on elements. Our local approximation spaces are characterized by two integers (n^*, k^*), where n^* is the number of subcells within an element and k^* is the polynomial degree of shape functions defined on the subcells. The selection of the value of (n^*, k^*) on a particular element depends on the smoothness of the solution on that element. More specifically, for elements on which the solution is smooth, we choose the smallest value $n^* = 1$ and the highest degree $k^* = k$. For elements containing shocks in the solution, we use the largest value $n^* = n$ and the lowest degree $k^* = 0$ so as to capture shocks without using artificial viscosity and limiting slopes/fluxes. The proposed method thus combines the accuracy and efficiency of high-order approximations with the robustness of low-order approximations. Numerical results are presented to demonstrate the performance of the proposed method.

I. Introduction

The numerical simulation of compressible flows has become an indispensable tool for many important applications such as aero-acoustics, vehicle design and turbomachinery. Although the ever increasing computer power allows us to solve complex problems that would have been intractable a few years ago, there are still many problems of practical interest for which the existing methods are inadequate. Therefore, the development of robust, accurate, and efficient methods for the numerical solution of the compressible Navier-Stokes equations in complex geometries remains a topic of considerable importance. Many complex flows are not amenable to low-order descriptions due to a number of intrinsic limitations in these schemes. Specifically, the description of shock propagation is particularly challenging as common low-order shock-capturing schemes necessarily smear the shock across the scale of interest. The wave-dispersion errors associated with low-order schemes are unacceptable for many applications. On the other hand, when high-order methods are used to describe shock waves and sharp discontinuities, they are generally less robust than their low-order counterparts due to the Gibbs phenomenon. Hence, it is desirable to develop numerical schemes that identify solution smoothness across the entire domain and apply suitable local approximations at element level to achieve both robustness and accuracy.

In recent years, discontinuous Galerkin (DG) methods [2–4,11,12,14,15,33–35] have attracted considerable attention because they possess a number of desirable properties for solving nonlinear hyperbolic systems of conservation laws. In particular, the DG methods work well on arbitrary meshes, result in stable high-order discretization of the convective and diffusive operators, allow for a simple and unambiguous imposition of boundary conditions, and are well-suited to adaptive strategies. However, the DG methods were criticized for providing sub-optimally convergent approximations for the flux as well as for producing a substantially larger amount of globally-coupled degrees of freedom (for the same mesh and polynomial degree of the approximation) in comparison to the well-established finite element methods for diffusion problems. In

[*] Research Scientist, Department of Aeronautics and Astronautics, M.I.T., 77 Massachusetts Avenue, AIAA Member.
[†] Postdoctoral Associate, Department of Aeronautics and Astronautics, M.I.T., 77 Massachusetts Avenue, AIAA Member.
[‡] PhD Candidate, Department of Aeronautics and Astronautics, M.I.T., 77 Massachusetts Avenue, AIAA Student Member.
[§] Professor, Department of Aeronautics and Astronautics, M.I.T., 77 Massachusetts Avenue, AIAA Associate Fellow.

response of these criticisms, the hybridizable discontinuous Galerkin (HDG) methods were first introduced in [10] and later extended in [7–9, 21–27].

The first HDG method for the compressible Euler and Navier-Stokes equations was introduced in [30]. This method possesses a number of desirable properties. First, in implicit formulations they reduce the globally coupled unknowns to the numerical trace of the solution on element boundaries, thereby leading to a significant reduction in the degrees of freedom. Second, they provide, for smooth (e.g. viscous-dominated) problems, approximations of all the variables which converge with the optimal order of $k+1$ in the L^2-norm. And third, they possess some superconvergence properties that allow us to compute a new approximate velocity which converge with order $k+2$ for $k \geq 1$.

In this paper, building on the previous work [30], we introduce a hybridized multiscale discontinuous Galerkin (HMDG) method for numerically solving compressible flows. The essential ingredients are (1) a local Galerkin projection of the underlying PDEs at the element level onto suitable local approximation spaces, (2) a judicious choice of the numerical flux to provide stability and consistency and (3) a global jump condition that enforces the continuity of the numerical flux to arrive at a global system in terms of the numerical trace. The local approximation spaces are characterized by two integers (n^*, k^*), where $n^* \in [1, n]$ is the number of subcells within an element and $k^* \in [0, k]$ is the polynomial degree of shape functions defined on the subcells. The selection of the value of (n^*, k^*) on a particular element depends on the smoothness of the solution on that element. More specifically, for elements on which the solution is smooth, we choose the smallest value $n^* = 1$ and the highest degree $k^* = k$. For elements containing shocks in the solution, we use the largest value $n^* = n$ and the lowest degree $k^* = 0$. The proposed method thus combines the accuracy and efficiency of high-order approximations with the robustness of low-order approximations. A key advantage of the HMDG method is that it can capture shocks without using artificial viscosity and limiting slopes/fluxes.

The paper is organized as follows. In Section 2, we introduce the notation used throughout the paper. We introduce the HMDG method for the compressible Euler equations in Section 3 and the compressible Navier-Stokes equations in Section 4. In Section 5, we briefly describe the smoothness and shock indicators used to determine the smoothness and shock location. In Section 6, we provide numerical results to assess the performance of the method.

II. Notation

Throughout this paper we shall denote scalar variables by italic letters with no boldface $(a, A, b, B, \text{etc.})$, vector variables by italic boldface lowercase letters $(\boldsymbol{a}, \boldsymbol{b}, \text{etc.})$, and second-order tensor variables by italic boldface uppercase letters $(\boldsymbol{A}, \boldsymbol{B}, \text{etc.})$. The identity tensor shall be denoted by $\boldsymbol{I}$. The components of $\boldsymbol{a}$ and $\boldsymbol{A}$ shall be denoted as a_i and A_{ij}, respectively. The symbols, $\cdot$, $\times$, $\otimes$, shall denote the usual scalar product, vector product, and tensor product, respectively. We shall use boldface roman uppercase letters $(\mathbf{A}, \mathbf{B}, \text{etc.})$ to denote matrices with entries $(A_{ij}, B_{ij}, \text{etc.})$ and boldface roman lowercase letters $(\mathbf{a}, \mathbf{b}, \text{etc.})$ to denote column vectors with elements $(a_i, b_i, \text{etc.})$. We shall also denote sets and spaces by calligraphic letters $(\mathcal{A}, \mathcal{B}, \text{etc.})$. In this paper, the tensor product notation and matrix product notation are interchanged, that is, $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b}$, $\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T$, $\boldsymbol{A} \cdot \boldsymbol{b} = \boldsymbol{A}\boldsymbol{b}$ and $\boldsymbol{A} \cdot \boldsymbol{B} = \boldsymbol{A}\boldsymbol{B}$.

A. Finite element mesh

Let Ω be a physical domain in $\mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$ in $\mathbb{R}^{d-1}$. We denote by $\mathcal{T}_h(\Omega)$ a collection of disjoint elements (triangles and tetrahedrons) that partition Ω. We also denote by $\partial\mathcal{T}_h(\Omega)$ 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 $\mathcal{T}_h(\Omega)$, $F = \partial K \cap \partial\Omega$ 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 $\mathcal{F}_h^o(\Omega)$ and $\mathcal{F}_h^\partial(\Omega)$ denote the set of interior and boundary faces, respectively. We denote by $\mathcal{F}_h(\Omega)$ the union of $\mathcal{F}_h^o(\Omega)$ and $\mathcal{F}_h^\partial(\Omega)$. Note that by definition $\partial\mathcal{T}_h(\Omega)$ and $\mathcal{F}_h(\Omega)$ are different. More precisely, an interior face is counted twice in $\partial\mathcal{T}_h(\Omega)$ but once in $\mathcal{F}_h(\Omega)$ and a boundary face is counted once in both $\partial\mathcal{T}_h(\Omega)$ and $\mathcal{F}_h(\Omega)$.

For every element K of the collection $\mathcal{T}_h(\Omega)$, we divide it into n^* subcells and denote by $\mathcal{S}_{n^*}(K)$ a collection of n^* disjoint subcells that partition K. We denote by $\partial\mathcal{S}_{n^*}(K) \equiv \{\partial S : \; S \in \mathcal{S}_{n^*}(K)\}$ a collection of the boundaries of all subcells in $\mathcal{S}_{n^*}(K)$. We also denote by $\mathcal{E}_{n^*}(K)$ the set of all faces in $\mathcal{S}_{n^*}(K)$.

B. Approximation spaces

Let $\mathcal{P}^\ell(D)$ denote the set of polynomials of degree at most ℓ on a domain D and $L^2(D)$ be the space of square integrable functions on D. For $K \in \mathcal{T}_h(\Omega)$ we introduce finite element spaces

$$\begin{aligned}\boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) &= \{\boldsymbol{a} \in [L^2(\mathcal{S}_{n^*}(K))]^m \ : \ \boldsymbol{a}|_S \in [\mathcal{P}^{k^*}(S)]^m, \ \forall S \in \mathcal{S}_{n^*}(K)\},\\ \boldsymbol{\mathcal{V}}^{k^*}(\mathcal{S}_{n^*}(K)) &= \{\boldsymbol{A} \in [L^2(\mathcal{S}_{n^*}(K))]^{m\times d} \ : \ \boldsymbol{A}|_S \in [\mathcal{P}^{k^*}(S)]^{m\times d}, \ \forall S \in \mathcal{S}_{n^*}(K)\},\end{aligned}$$

where $\boldsymbol{a} = (a_i), 1 \le i \le m$, and $\boldsymbol{A} = (A_{ij}), 1 \le i \le m, 1 \le j \le d$. For $K \in \mathcal{T}_h(\Omega)$, we further introduce finite element spaces on $\mathcal{E}_{n^*}(K)$ as

$$\boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K)) = \{\boldsymbol{y} \in [L^2(\mathcal{E}_{n^*}(K))]^m \ : \ \boldsymbol{y}|_E \in [\mathcal{P}^{k^*}(E)]^m, \ \forall E \in \mathcal{E}_{n^*}(K)\},$$

where $\boldsymbol{y} = (y_i), 1 \le i \le m$. Note that $\boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K))$ consists of functions which are continuous inside the faces $E \in \mathcal{E}_{n^*}(K)$ and discontinuous at their borders. It is important to note that the polynomial degree k^* varies with K.

In addition, we introduce a finite element space on $\mathcal{F}_h(\Omega)$ as

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

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

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

$$(\boldsymbol{a}, \boldsymbol{b})_{\mathcal{S}_{n^*}(K)} = \sum_{S\in\mathcal{S}_{n^*}(K)} (\boldsymbol{a}, \boldsymbol{b})_S, \quad (\boldsymbol{A}, \boldsymbol{B})_{\mathcal{S}_{n^*}(K)} = \sum_{S\in\mathcal{S}_{n^*}(K)} (\boldsymbol{A}, \boldsymbol{B})_S,$$

on $\mathcal{S}_{n^*}(K)$ for $K \in \mathcal{T}_h(\Omega)$, and

$$(\boldsymbol{a}, \boldsymbol{b})_{\mathcal{T}_h(\Omega)} = \sum_{K\in\mathcal{T}_h(\Omega)} (\boldsymbol{a}, \boldsymbol{b})_K, \qquad (\boldsymbol{A}, \boldsymbol{B})_{\mathcal{T}_h(\Omega)} = \sum_{K\in\mathcal{T}_h(\Omega)} (\boldsymbol{A}, \boldsymbol{B})_K,$$

on $\mathcal{T}_h(\Omega)$. We also define boundary inner products

$$\langle \boldsymbol{a}, \boldsymbol{b}\rangle_{\partial\mathcal{S}_{n^*}(K)} = \sum_{S\in\mathcal{S}_{n^*}(K)} \langle \boldsymbol{a}, \boldsymbol{b}\rangle_{\partial S}, \qquad \langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\partial\mathcal{S}_{n^*}(K)} = \sum_{S\in\mathcal{S}_{n^*}(K)} \langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\partial S},$$

and

$$\langle \boldsymbol{a}, \boldsymbol{b}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \boldsymbol{a}, \boldsymbol{b}\rangle_{\partial K}, \qquad \langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\partial K}.$$

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

III. The Euler Equations

A. Governing equations

We consider the Euler equations of gas dynamics defined over a domain $\Omega \subset \mathbb{R}^d$ written in nondimensional conservation form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla\cdot\boldsymbol{F}(\boldsymbol{u}) \ = \ 0, \qquad \text{in } \Omega, \tag{1}$$

where $\boldsymbol{u}$ is the m-dimensional vector of conserved dimensionless quantities (namely, density, momentum and energy) and $\boldsymbol{F}(\boldsymbol{u})$ are the inviscid fluxes of dimension $m \times d$. The nondimensional form of the Euler equations we use here can be found in Ref. [1]. Of course, the Euler equations (1) must be supplemented with appropriate initial conditions and boundary conditions at the inflow section, outflow section, and solid wall.

B. The HMDG method

We first introduce the numerical trace $\boldsymbol{\lambda}_h \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega))$ which is an approximation to the field variable $\boldsymbol{u}$ on the set $\mathcal{F}_h(\Omega)$. For any element $K \in \mathcal{T}_h(\Omega)$ we define a so-called *local problem* in which we seek an approximation $(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h) \in \boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K))$ such that

$$\begin{aligned} \left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w}\right)_{\mathcal{S}_{n^*}(K)} - (\boldsymbol{F}(\boldsymbol{u}_h), \nabla \boldsymbol{w})_{\mathcal{S}_{n^*}(K)} + \left\langle \widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}, \boldsymbol{w} \right\rangle_{\partial \mathcal{S}_{n^*}(K)} &= \boldsymbol{0}, \\ \left\langle \widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}, \boldsymbol{y} \right\rangle_{\partial \mathcal{S}_{n^*}(K) \backslash \partial K} + \langle \widehat{\boldsymbol{u}}_h - \boldsymbol{\lambda}_h, \boldsymbol{y} \rangle_{\partial K} &= \boldsymbol{0}, \end{aligned} \tag{2}$$

for all $(\boldsymbol{w}, \boldsymbol{y}) \in \boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K))$. The above problem can be solved at the element level whenever the numerical trace $\boldsymbol{\lambda}_h$ is known. In other words, the local problem defines $(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h)$ as a function of $\boldsymbol{\lambda}_h$. We would like to point out that the local problem (2) is nothing but the HDG discretization of the Euler equations (1) with a Dirichlet boundary condition $\boldsymbol{u} = \boldsymbol{\lambda}$ at the element level.

Of course, we need to define the numerical flux $\widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}$ such that it guarantees the stability of the local problem. Following [30, 31] the numerical flux is defined as

$$\widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n} = \boldsymbol{F}(\boldsymbol{u}_h) \cdot \boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h)(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \tag{3}$$

where $\boldsymbol{S}(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h)$ is the so-called stabilization matrix which plays an important role in the stability of the method. The choice of the stabilization matrix is discussed in [30, 31]. It should be obvious from (2) that the numerical flux $\widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}$ is also a function of the numerical trace $\boldsymbol{\lambda}_h$.

We finally need to determine $\boldsymbol{\lambda}_h$. This is done by requiring that $\boldsymbol{\lambda}_h \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega))$ is the solution of

$$\left\langle \widehat{\boldsymbol{F}}_h(\boldsymbol{\lambda}_h) \cdot \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h \backslash \partial \Omega} + \left\langle \widehat{\boldsymbol{b}}_h(\boldsymbol{\lambda}_h), \boldsymbol{\mu} \right\rangle_{\partial \Omega} = \boldsymbol{0}, \qquad \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega)). \tag{4}$$

Here $\widehat{\boldsymbol{b}}_h(\boldsymbol{\lambda}_h)$ is the boundary flux whose precise definition depends on the boundary conditions on $\partial\Omega$ and is discussed in detail in [30, 31]. Note that (4) is the global weak statement that enforces the continuity of the numerical flux across element interfaces and imposes the boundary conditions.

C. Implicit time discretization

Implicit time stepping methods are used to advance the system (2)-(4). For simplicity of exposition we consider the Backward-Euler scheme to discretize the time derivative since time integration using high-order backward difference formulae (BDF) schemes and diagonally implicit Runge-Kutta (DIRK) methods admits a similar procedure. At time level $t^j = j\Delta t$ we seek an approximation $(\boldsymbol{u}_h^j, \widehat{\boldsymbol{u}}_h^j) \in \boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K))$ such that

$$\begin{aligned} \left(\frac{\boldsymbol{u}_h^j - \boldsymbol{u}_h^{j-1}}{\Delta t}, \boldsymbol{w}\right)_{\mathcal{S}_{n^*}(K)} + (\boldsymbol{F}(\boldsymbol{u}_h^j), \nabla \boldsymbol{w})_{\mathcal{S}_{n^*}(K)} - \left\langle \widehat{\boldsymbol{F}}_h^j \cdot \boldsymbol{n}, \boldsymbol{w} \right\rangle_{\partial \mathcal{S}_{n^*}(K)} &= \boldsymbol{0}, \\ \left\langle \widehat{\boldsymbol{F}}_h^j \cdot \boldsymbol{n}, \boldsymbol{y} \right\rangle_{\partial \mathcal{S}_{n^*}(K) \backslash \partial K} + \left\langle \widehat{\boldsymbol{u}}_h^j - \boldsymbol{\lambda}_h^j, \boldsymbol{y} \right\rangle_{\partial K} &= \boldsymbol{0}, \end{aligned} \tag{5}$$

for all $(\boldsymbol{w}, \boldsymbol{y}) \in \boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{S}_{n^*}(K))$. The local problem (5) defines $(\boldsymbol{u}_h^j, \widehat{\boldsymbol{u}}_h^j)$ as a function of $\boldsymbol{\lambda}_h^j$.

The numerical flux at time t^j is defined as

$$\widehat{\boldsymbol{F}}_h^j \cdot \boldsymbol{n} = \boldsymbol{F}(\boldsymbol{u}_h^j) \cdot \boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h^j, \widehat{\boldsymbol{u}}_h^j)(\boldsymbol{u}_h^j - \widehat{\boldsymbol{u}}_h^j). \tag{6}$$

Of course, $\widehat{\boldsymbol{F}}_h^j$ depends implicitly on $\boldsymbol{\lambda}_h^j$ via the local problem (5).

Finally, the numerical trace $\boldsymbol{\lambda}_h^j \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega))$ satisfies

$$\left\langle \widehat{\boldsymbol{F}}_h^j(\boldsymbol{\lambda}_h^j) \cdot \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h \backslash \partial \Omega} + \left\langle \widehat{\boldsymbol{b}}_h^j(\boldsymbol{\lambda}_h^j), \boldsymbol{\mu} \right\rangle_{\partial \Omega} = \boldsymbol{0}, \qquad \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega)). \tag{7}$$

This is the global weak statement that enforces the continuity of the numerical flux across element interfaces and imposes the boundary conditions.

The implementation of the HMDG method can be carried in a similar fashion as that of the hybridized discontinuous Petrov-Galerkin (HDPG) method introduced in [20].

IV. The Navier-Stokes Equations

A. Governing equations

We consider the compressible Navier-Stokes equations written in conservation form as

$$\begin{aligned} \boldsymbol{q} - \nabla \boldsymbol{u} &= 0, \qquad \text{in } \Omega, \\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{F}(\boldsymbol{u}) + \boldsymbol{G}(\boldsymbol{u}, \boldsymbol{q})) &= 0, \qquad \text{in } \Omega, \end{aligned} \tag{8}$$

where $\boldsymbol{F}(\boldsymbol{u})$ and $\boldsymbol{G}(\boldsymbol{u}, \boldsymbol{q})$ are the inviscid and viscous fluxes of dimension m and $m \times d$, respectively. The nondimensional form of the Navier-Stokes equations as well as the definition of the inviscid and viscous fluxes can be found in Ref. [1]. Of course, the Navier-Stokes equations (8) must be supplemented with appropriate initial conditions and boundary conditions at the inflow section, outflow section, and solid wall.

B. The HMDG method

For any element $K \in \mathcal{T}_h(\Omega)$ we define a so-called *local problem* in which we seek an approximation $(\boldsymbol{q}_h, \boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h) \in \boldsymbol{\mathcal{V}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K))$ such that

$$\begin{aligned} (\boldsymbol{q}_h, \boldsymbol{v})_{\mathcal{S}_{n^*}(K)} + (\boldsymbol{u}_h, \nabla \cdot \boldsymbol{v})_{\mathcal{S}_{n^*}(K)} - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{v} \cdot \boldsymbol{n} \rangle_{\partial \mathcal{S}_{n^*}(K)} &= \boldsymbol{0}, \\ \left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w} \right)_{\mathcal{S}_{n^*}(K)} - (\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{q}_h), \nabla \boldsymbol{w})_{\mathcal{S}_{n^*}(K)} + \left\langle \widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}, \boldsymbol{w} \right\rangle_{\partial \mathcal{S}_{n^*}(K)} &= \boldsymbol{0}, \\ \left\langle \widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}, \boldsymbol{y} \right\rangle_{\partial \mathcal{S}_{n^*}(K) \backslash \partial K} + \langle \widehat{\boldsymbol{u}}_h - \boldsymbol{\lambda}_h, \boldsymbol{y} \rangle_{\partial K} &= \boldsymbol{0}, \end{aligned} \tag{9}$$

for all $(\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{y}) \in \boldsymbol{\mathcal{V}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{W}}^{k^*}(\mathcal{S}_{n^*}(K)) \times \boldsymbol{\mathcal{Y}}^{k^*}(\mathcal{E}_{n^*}(K))$. The above problem defines $(\boldsymbol{q}_h, \boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h)$ as a function of $\boldsymbol{\lambda}_h$. Again the local problem (9) is nothing but the HDG discretization of the Navier-Stokes equations (8) with a Dirichlet boundary condition $\boldsymbol{u} = \boldsymbol{\lambda}$ at the element level.

To ensure the stability of the local problem, the numerical flux is defined as

$$\widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n} = \boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{q}_h) \cdot \boldsymbol{n} + \boldsymbol{S}(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h)(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h), \tag{10}$$

It should be noted from (9) that the numerical flux $\widehat{\boldsymbol{F}}_h \cdot \boldsymbol{n}$ is also a function of the numerical trace $\boldsymbol{\lambda}_h$.

Finally, we find $\boldsymbol{\lambda}_h \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega))$ such that

$$\left\langle \widehat{\boldsymbol{F}}_h(\boldsymbol{\lambda}_h) \cdot \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h \backslash \partial \Omega} + \left\langle \widehat{\boldsymbol{b}}_h(\boldsymbol{\lambda}_h), \boldsymbol{\mu} \right\rangle_{\partial \Omega} = \boldsymbol{0}, \qquad \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}^k(\mathcal{F}_h(\Omega)). \tag{11}$$

This is the global weak formulation that enforces the continuity of the numerical flux across element interfaces and imposes the boundary conditions.

Implicit time stepping methods are then used to discretize the time derivative of (9) as described in the previous section. This results in a fully discrete nonlinear system which can then be solved by using the Newton-Raphson method and the hybridization technique as described in [20].

V. Smoothness and Shock Indicators

A. Smoothness indicator

As mentioned in the Introduction section, we determine the number of subcells n^* and polynomial degree k^* for our local approximation spaces based on the smoothness of the solution. In order to determine the smoothness of the solution, we use the discontinuity sensor introduced in [29, 32]. In particular, we express the solution within each element in terms of a hierarchical family of orthogonal polynomials. For smooth solutions, the coefficients in the expansion are expected to decay very rapidly. On the other hand, when the solution is not smooth, the strength of the discontinuity will dictate the rate of decay of the expansion coefficients.

Let s be an appropriate sensor variable (e.g. density, pressure). We express s within each element $K \in \mathcal{T}_h(\Omega)$ in terms of a hierarchical orthogonal basis as

$$s = \sum_{i=1}^{N} s_i \psi_i(\boldsymbol{x}), \tag{12}$$

where N is the total number of terms in the expansion and $\psi_i(\boldsymbol{x})$ are the orthogonal basis functions. We denote by $\bar{s}$ a truncated expansion of the above expression up to leading $M(\leq N)$ terms. We then define the smoothness indicator as

$$I(s) = \log_{10} \frac{(s - \bar{s}, s - \bar{s})_K}{(s, s)_K}. \tag{13}$$

The convergence rate of $I(s)$ to zero as $M \to N$ will determine the smoothness of the solution.

B. Shock indicator

In [19, 28], we propose to use the HDG method with artificial viscosity for shock capturing. In this paper, we use it as a shock indicator. More specifically, our shock indicator variable is specified as follows

$$s_K = \int_K f\Big(\frac{\ell \nabla \cdot \boldsymbol{v}}{c}\Big) d\boldsymbol{x}, \qquad \forall K \in \mathcal{T}_h(\Omega), \tag{14}$$

where ℓ is a length scale, $\boldsymbol{v}$ is the velocity field, $c = \sqrt{\gamma p/\rho}$ is the sound speed, and f is an analytic function given by

$$f(x) = \alpha \log(1 + \exp((\beta - x)/\alpha)), \tag{15}$$

for $\alpha = 0.1$ and $\beta = -0.5$.

The value of the shock indicator variable s_K will ultimately determine the strength of the shock at the element K. The element K is marked to contain shocks if it has $s_K \geq 1/2$. When shocks are detected on K, we set $k^* = 0$ and $n^* = n$ for a given user-specified constant n.

VI. Numerical Results

A. Inviscid transonic flow past a Kármán-Trefftz airfoil

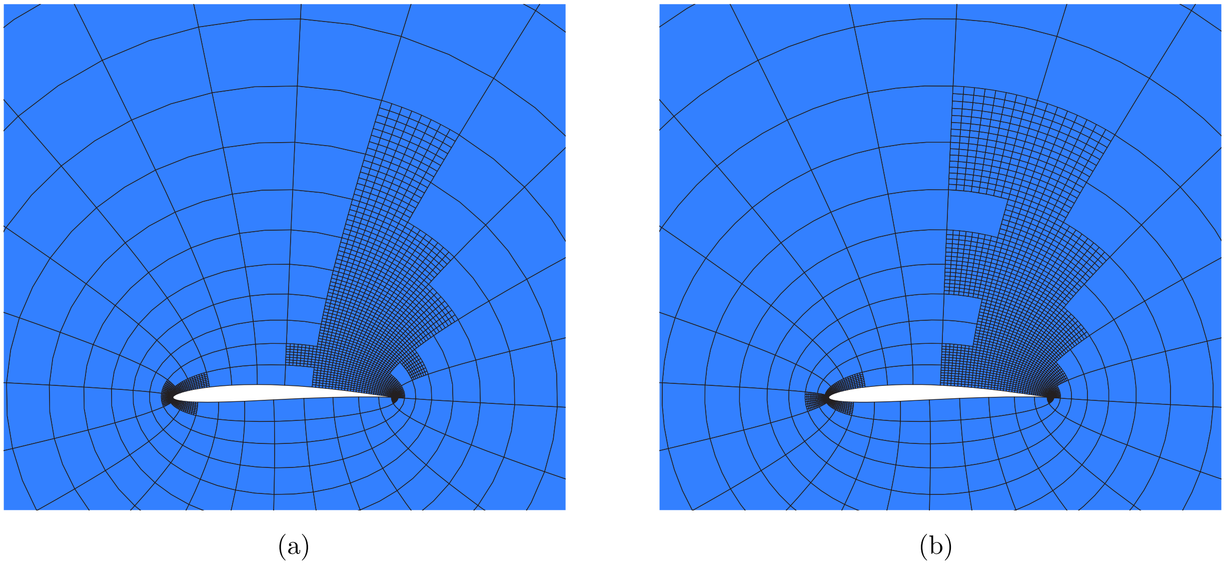
The first example involves the transonic flow past a Kármán-Trefftz airfoil at angle of attack $\alpha = 5^{\text{o}}$ and freestream Mach number $M_\infty = 0.8$. We perform a backward-Euler time integration from the free-stream solution to obtain the steady-state solution. During this time evolution, the method automatically detect shocks and modify the local approximation spaces at every time step. Figure 1 shows the zoom of the finite element meshes of 384 quadrilateral elements at the convergence of the steady-state solution for $k = 2$ and $k = 3$. Here we use polynomials of degree $k^* = k$ to represent the numerical solution on elements with smooth solution. For elements with non-smooth solution due to shock and leading/trailing edge, we divide each of them into $n^* = 8 \times 8$ subcells and use piecewise-constant polynomials of degree $k^* = 0$ to represent the numerical solution on those elements. Figure 2 displays the corresponding pressure contours. We observe that the method is capable of capturing transonic shock and that using $k = 3$ yields better results than $k = 2$.

B. Inviscid supersonic flow past a NACA 0012 airfoil

This example involves inviscid supersonic flow past a NACA 0012 airfoil at angle of attack $\alpha = 1.5^{\text{o}}$. Figure 3 shows the zoom of the finite element meshes of 893 triangular elements at the convergence of the steady-state solution for $M_\infty = 1.5$ and $M_\infty = 2.0$. Here we use polynomials of degree $k^* = k = 4$ to represent the numerical solution on elements with smooth solution. For elements with non-smooth solution due to shock and leading/trailing edge, we divide each of them into $n^* = 64$ subcells and use piecewise-constant polynomials of degree $k^* = 0$ to represent the numerical solution on those elements. Figure 4 depicts the corresponding Mach number contours. It is clear that our method can capture supersonic shocks robustly without using artificial viscosity and a flux/slop limiter.

C. Inviscid hypersonic flow past a circular cylinder

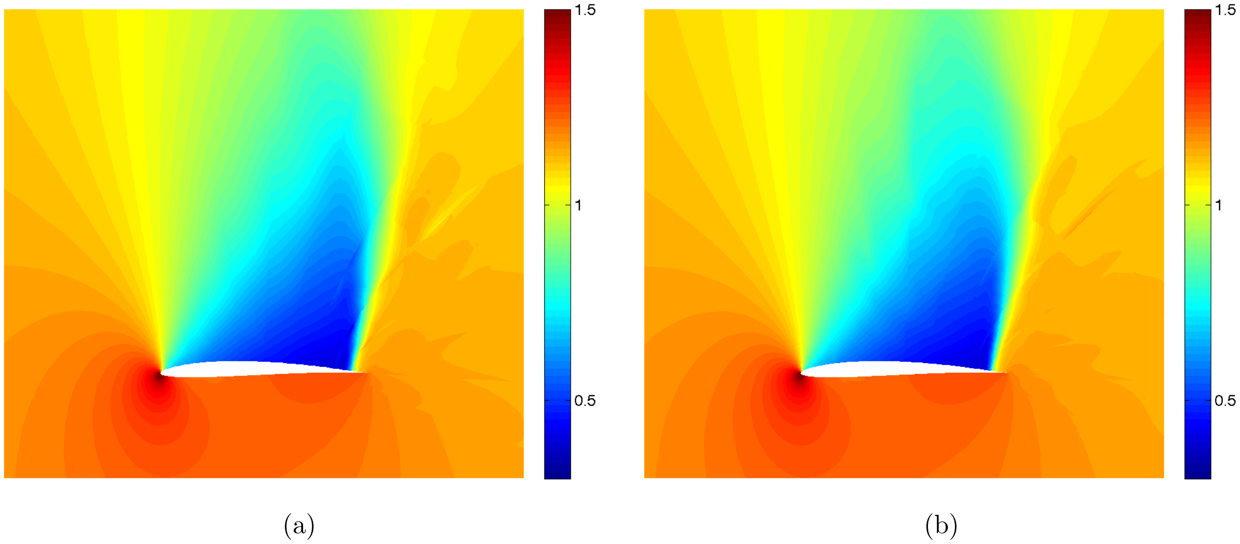
We consider inviscid hypersonic flow past a circular cylinder at two different free-stream Mach numbers $M_\infty = 10$ and $M_\infty = 20$. This test case serves to illustrate the robustness of our method for capturing very strong shocks in the hypersonic flow regime. We show in Figure 5 the finite element mesh of 338 quadrilateral elements used in our computation. Each element is further divided uniformly into $n^* = 64$ subcells for $M_\infty = 10$ or $n^* = 256$ subcells for $M_\infty = 20$. Note that piecewise-constant polynomials $k^* = 0$



(a)

(b)

Figure 1. The zoom of the finite element meshes near the Kármán-Trefftz airfoil for (a) $k = 2$ and (b) $k = 3$. Here we use polynomials of degree $k^* = k$ to represent the numerical solution on elements which have the shock indicator $s_K < 1/2$. For elements with $s_K \geq 1/2$, we divide each of them into $n^* = 8 \times 8$ subcells and use piecewise-constant polynomials of degree $k^* = 0$ to represent the numerical solution on those elements. Polynomials of degree k are used to represent the numerical trace on the edges of the elements.



(a)

(b)

Figure 2. Inviscid transonic flow past the Kármán-Trefftz airfoil at $\alpha = 5^o$ and $M_\infty = 0.8$: Pressure contours for (a) $k = 2$ and (b) $k = 3$.

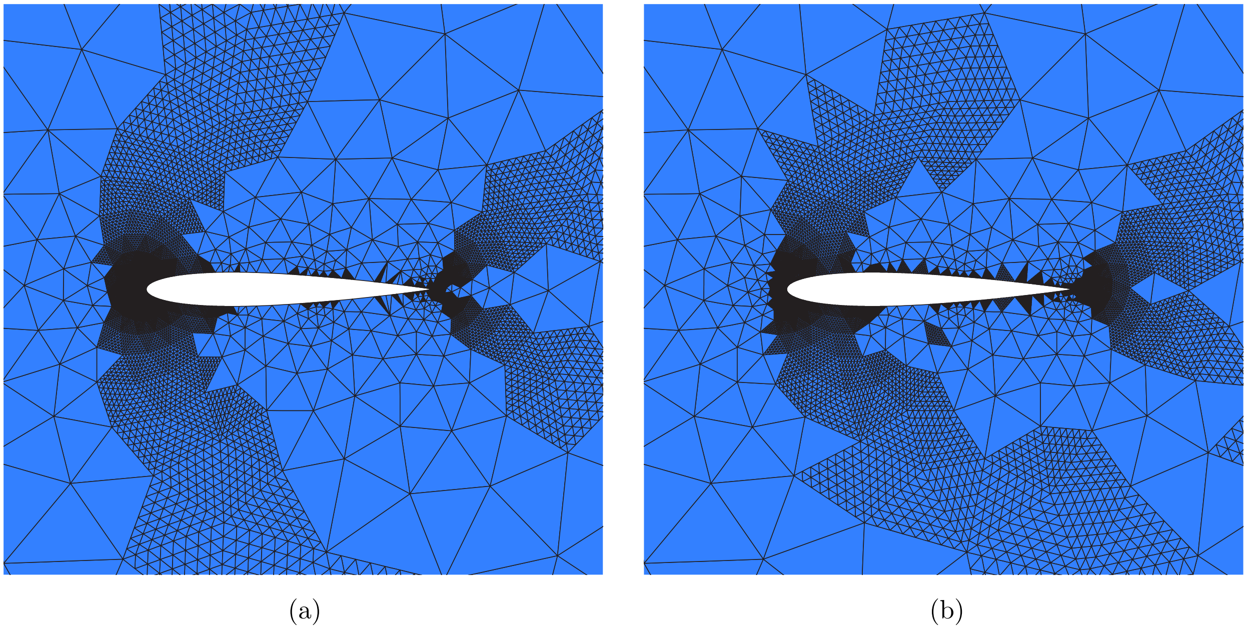


Figure 3. The zoom of the finite element meshes near the NACA 0012 airfoil for (a) $M_\infty = 1.5$ and (b) $M_\infty = 2.0$. Here we use polynomials of degree $k^* = k = 4$ to represent the numerical solution on elements which have the shock indicator $s_K < 1/2$. For elements with $s_K \geq 1/2$, we divide each of them into $n^* = 8 \times 8$ subcells and use piecewise-constant polynomials of degree $k^* = 0$ to represent the numerical solution on those elements. Polynomials of degree $k = 4$ are used to represent the numerical trace on the edges of the elements.

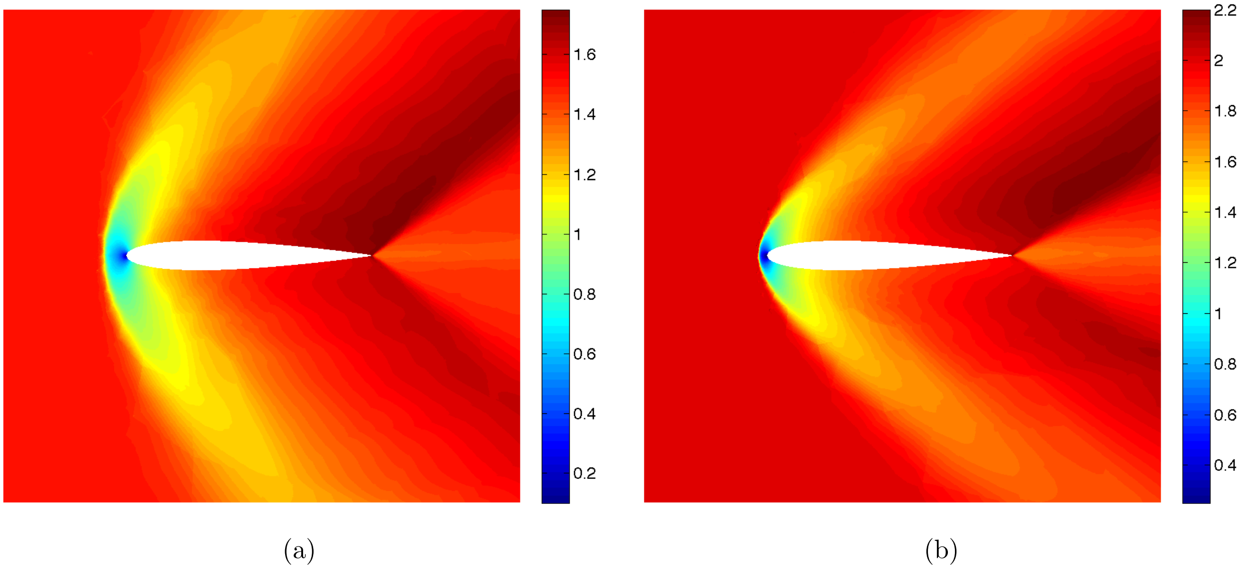


Figure 4. Inviscid supersonic flow past the NACA 0012 airfoil at $\alpha = 1.5^o$: Mach number contours for (a) $M_\infty = 1.5$ and (b) $M_\infty = 2.0$.

are used on all the subcells to represent the approximate solution. Polynomials of degree $k = 4$ are used to represent the numerical trace over the edges in both cases. We present in Figure 6 the Mach number contour and pressure contour for $M_\infty = 10$ and Figure 7 for $M_\infty = 20$. We observe that the proposed method can capture strong bow shocks. Moreover, increasing the number of subcells per element leads to sharper shock profile.

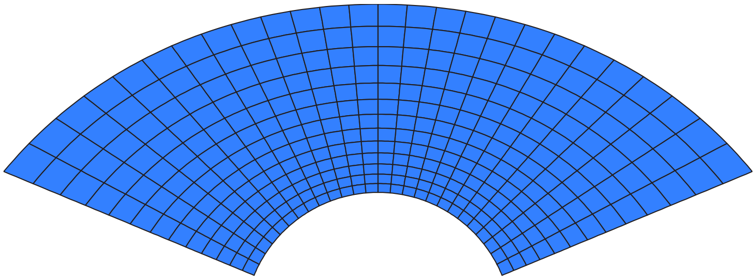
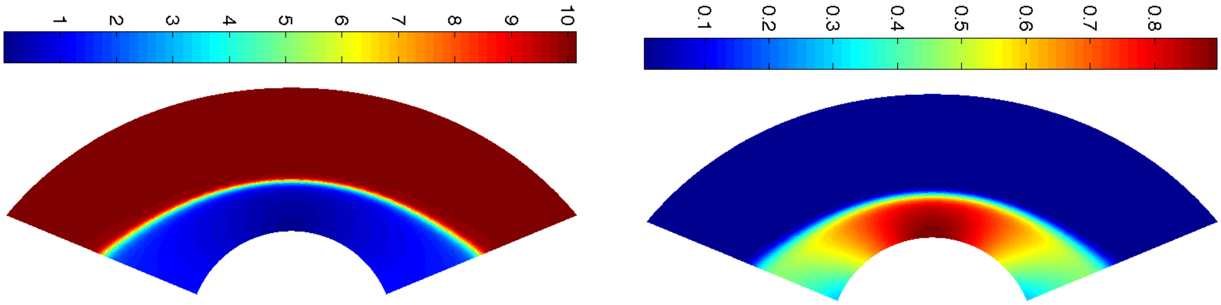
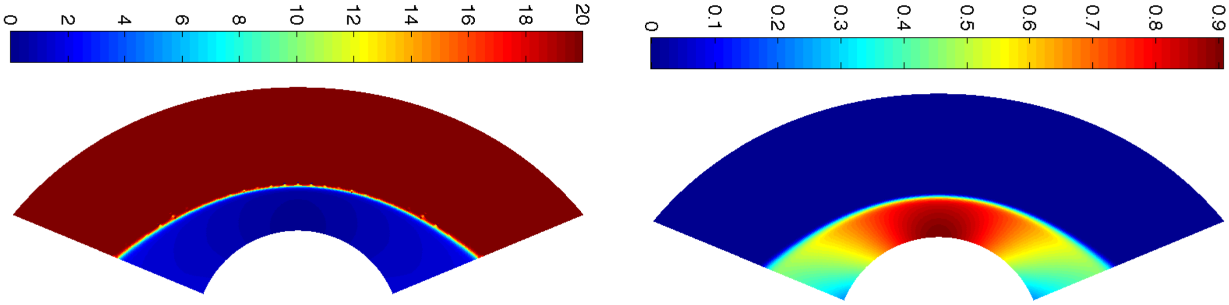


Figure 5. The finite element mesh for inviscid hypersonic flow past a circular cylinder. Each element is further divided into 64 subcells for $M_\infty = 10$ or 256 subcells for $M_\infty = 20$. Polynomials of degree $k^* = 0$ are used to represent the numerical solution on the subcells. Polynomials of degree $k = 4$ are used to represent the numerical trace on the edges of the elements.



(a) (b)

Figure 6. Inviscid hypersonic flow past a circular cylinder at $M_\infty = 10$: (a) Mach number contour and (b) pressure contour.



(a) (b)

Figure 7. Inviscid hypersonic flow past a circular cylinder at $M_\infty = 20$: (a) Mach number contour and (b) pressure contour.

D. Laminar viscous hypersonic flow past a circular half-cylinder

The hypersonic flow over a half-cylinder is a simple test problem that highlights the difficulties in surface heat transfer.[2] The flow conditions are free-stream Mach number $M_\infty = 17.605$, Reynolds number $Re = 376,930$, heat specific constant $\gamma = 1.4$, and Prandtl number $Pr = 0.72$. Boundary conditions are applied as follows.

The no-slip adiabatic-wall condition is used on the cylinder wall, the full-state condition is used on the outer boundary, and the extrapolation condition is use on the remaining boundary. Figure 8 shows the computational mesh of 338 quadrilateral elements. Each element is further divided uniformly into $n^* = 256$ subcells and piecewise-constant polynomials are used on all the subcells to represent the approximate solution. Polynomials of degree $k = 5$ are used to represent the numerical trace over the edges. Figure 9 displays the Mach number, temperature, pressure and density contours of the flow field. It is clear that the proposed method is able to capture very strong shock waves without using artificial viscosity and a flux/slop limiter. Moreover, the shock is very sharp and clean.

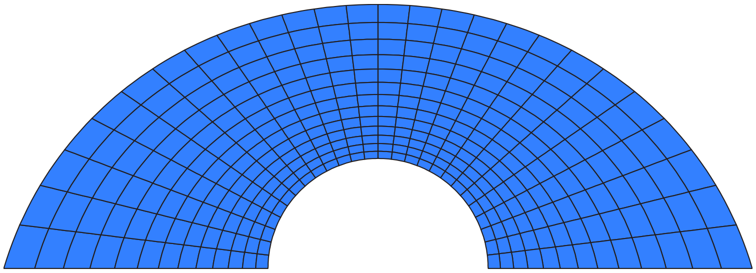
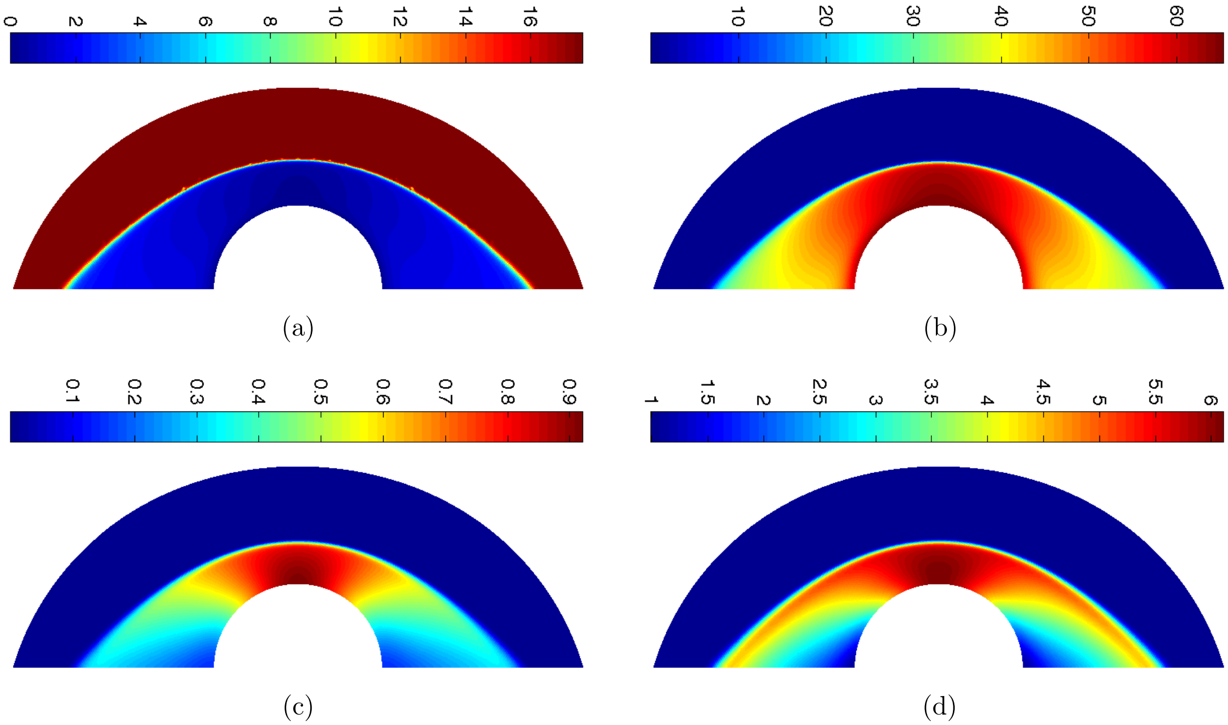


Figure 8. The finite element mesh for laminar viscous hypersonic flow past a circular half-cylinder at $M_\infty = 17.605$ and $Re = 376,930$. Each element is further divided into 256 subcells on which polynomials of degree $k^* = 0$ are used to represent the numerical solution. Polynomials of degree $k = 5$ are used to represent the numerical trace on the edges of the mesh.



(a)

(b)

(c)

(d)

Figure 9. Laminar viscous hypersonic flow past a circular half-cylinder at $M_\infty = 17.605$ and $Re = 376,930$: (a) Mach number contour, (b) temperature contour, (c) pressure contour, and (d) density contour.

E. Laminar viscous hypersonic flow past a circular half-cylinder: Shock-shock interaction

This example is similar to the previous example except for the flow conditions and boundary conditions. The flow conditions are Reynolds number $Re = 194,000$, heat specific constant $\gamma = 1.4$, and Prandtl number $Pr = 0.72$. The no-slip adiabatic-wall condition is used on the cylinder wall, the full-state condition is used on the outer boundary, and the extrapolation condition is use on the remaining boundary. For the full-state condition, we apply the following free-stream values

$$\rho^1_\infty = 1, \quad \rho^1_\infty u^1_\infty = 1, \quad \rho^1_\infty v^1_\infty = 0, \quad \rho^1_\infty E_\infty = 0.5277, \quad M^1_\infty = 8.03, \tag{16}$$

on one part of the outer boundary, and the following free-stream values

$$\rho^2_\infty = 3.3333, \quad \rho^2_\infty u^2_\infty = 3.1080, \quad \rho^2_\infty v^2_\infty = 0.6890, \quad \rho^2_\infty E_\infty = 1.7171, \quad M^2_\infty = 5.25, \tag{17}$$

on the remaining part. This example serves to illustrate the unique characteristics of the proposed method for solving complex interaction of coexisting discontinuous and smooth solution regions. The interaction results in the formation of a supersonic impinging jet, a series of shock waves, expansion waves, and shear layers in a local region of interaction. The supersonic impinging jet, which is bounded by two shear layers separating the jet from the upper and lower subsonic regions, impinges on the body surface, and is terminated by a jet bow shock just ahead of the surface. This impinging jet bow shock wave creates a small stagnation region of high temperature, pressure, and heating rates. Meanwhile, shear layers are formed to separate the supersonic jet from the lower and upper subsonic regions.

We show in Figure 10 the finite element mesh of 1911 quadrilateral elements. Each element is further divided uniformly into $n^* = 64$ subcells and piecewise-constant polynomials are used on all the subcells to represent the approximate solution. Polynomials of degree $k = 4$ are used to represent the numerical trace over the edges. We present in Figure 11 the Mach number, temperature, pressure and density contours of the flow field. It is clear that the proposed method is able to resolve the complex flow field and capture a wide range of shock phenomena including transonic, supersonic, and hypersonic shocks. Finally, we emphasize that these results are obtained without using artificial viscosity and a flux/slop limiter.

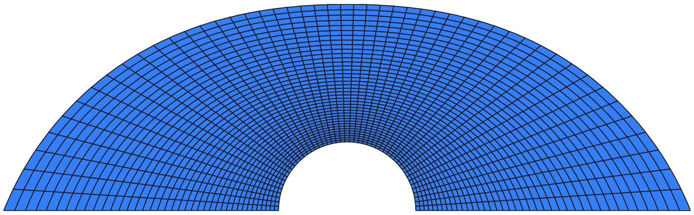


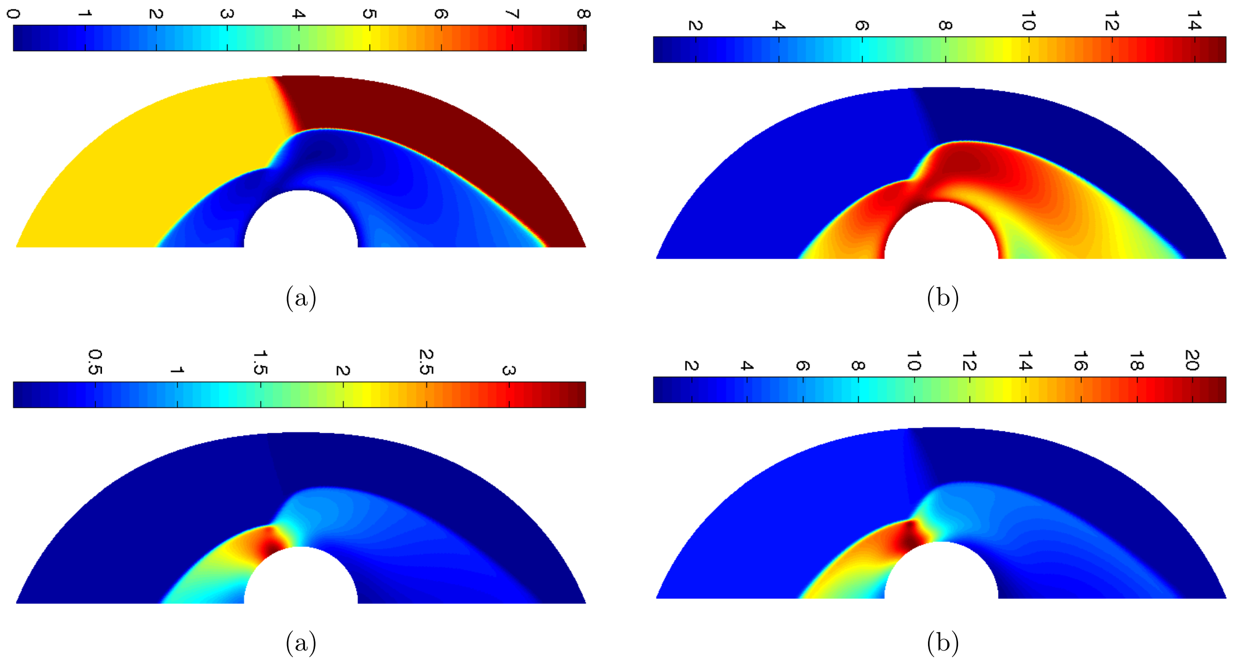
Figure 10. The finite element mesh consists of 1911 quadrilateral elements. Each element is further divided into 256 subcells on which polynomials of degree $k^* = 0$ are used to represent the numerical solution. Polynomials of degree $k = 4$ are used to represent the numerical trace on the edges of the mesh.

VII. Conclusions

In this paper, we have introduced a hybridized multiscale discontinuous Galerkin method for numerically solving compressible flows. We presented several test cases to illustrate the performance of the proposed method for a wide range of shock flows from transonic regime to hypersonic regime. The results are quite promising as the proposed method can capture very strong shocks and produce sharp shock profiles without using artificial viscosity and a flux/slop limiter. Despite the good results, there remains some important issues to be addressed. First, we need to improve our shock indicator because it did not perform well in the test cases considered here. Second, we may need to increase the approximation order in the shock region because piecewise-constant approximation is quite diffusive and affecting solutions in the smooth region. Future work will aim at tackling these issues and apply this method to other applications such as RANS flows.

Acknowledgments

N. C. Nguyen, X. Roca, and J. Peraire would like to acknowledge 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. D. Moro would like to acknowledge the support of the Caja-Madrid Foundation for the Graduate Studies Scholarship.



(a) (b)

(a) (b)

Figure 11. Laminar hypersonic flow past a circular half-cylinder at $M_\infty = 8.03$ and $Re = 194,000$: (a) Mach number contour, (b) temperature contour, (c) pressure contour, and (d) density contour.

References

[1] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher, *Computational Fluid Dynamics and Heat Transfer.* Hemisphere Publishing, New York, 1984.

[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] N. K. Burgess and D. J. Mavriplis. Robust computation of turbulent flows using a discontinuous galerkin method (AIAA Paper 2012-457). In *Proceedings of the 50th AIAA Aerospace Sciences Meeting and Exhibit*, Nashville, TN, January 9-12, 2012.

[5] B. Cockburn. Discontinuous Galerkin methods. *ZAMM Z. Angew. Math. Mech.*, 83:731–754, 2003.

[6] B. Cockburn. Discontinuous Galerkin Methods for Computational Fluid Dynamics. In R. de Borst E. Stein and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 3, pages 91–123. John Wiley & Sons, Ltd., England, 2004.

[7] B. Cockburn, B. Dong, and J. Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Math. Comp.*, 77:1887–1916, 2008.

[8] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, and R. Sacco. A hybridizable discontinuous galerkin method for steady-state convection-diffusion-reaction problems. *SIAM J. Scientific Computing*, 31(5):3827–3846, 2009.

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

[10] 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.

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

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

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

[14] R. Hartmann and P. Houston. An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 227:9670–9685, 2008.

[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] T. J. R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Comput. Methods Appl. Mech. Engrg.*, 54:341–355, 1986.

[17] 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.

[18] 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.

[19] D. Moro, N. C. Nguyen, and J. Peraire. Navier-Stokes Solution Using Hybridizable Discontinuous Galerkin Methods (AIAA Paper 2011-3060). In *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, June 2011.

[20] D. Moro, N. C. Nguyen, J. Peraire, and J. Gopalakrishnan. A hybridized discontinuous Petrov-Galerkin method for compressible flows (AIAA Paper 2011-197). In *Proceedings of the 49th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, January 2011.

[21] N.C. Nguyen, J. Peraire, and B. Cockburn, A hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 230:1147–1170, 2011.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[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, January 2010.

[28] N. C. Nguyen, and J. Peraire. An Adaptive Shock-Capturing HDG Method for Compressible Flows (AIAA Paper 2011-3407). In *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, June 2011.

[29] N. C. Nguyen, P. O. Persson, and J. Peraire, RANS Solutions using high order discontinuous Galerkin methods. In Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2007-0914, Reno, NV, January 2007.

[30] 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, January 2010.

[31] J. Peraire, N. C. Nguyen, and B. Cockburn. An Embedded Discontinuous Galerkin Method for the compressible Euler and Navier-Stokes equations (AIAA Paper 2011-3228). In *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, June 2011.

[32] 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.

[33] 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.

[34] P. O. Persson and J. Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008.

[35] 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.