20th AIAA Computational Fluid Dynamics Conference
27 - 30 June 2011, Honolulu, Hawaii

AIAA 2011-3060

An Adaptive Shock-Capturing HDG Method for Compressible Flows

N. C. Nguyen* and J. Peraire†

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

We introduce a hybridizable discontinuous Galerkin (HDG) method for the numerical solution of the compressible Euler equations with shock waves. By locally condensing the approximate conserved variables the HDG method results in a final system involving only the degrees of freedom of the approximate traces of the conserved variables. The HDG method inherits the geometric flexibility and high-order accuracy of discontinuous Galerkin methods, and offers a significant reduction in the computational cost. In order to treat compressible fluid flows with discontinuities, the HDG method is equipped with an artificial viscosity term based on an extension of existing artificial viscosity methods. Moreover, the artificial viscosity can be used as an indicator for adaptive grid refinement to improve shock profiles. Numerical results for subsonic, transonic, supersonic, and hypersonic flows are presented to demonstrate the performance of the proposed approach.

I. Introduction

The development of robust, accurate, and efficient methods for the numerical solution of hyperbolic nonlinear systems of conservation laws in complex geometries is a topic of considerable importance. Indeed, hyperbolic partial different equations (PDEs) 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 a hyperbolic system is the formation and propagation of discontinuities or shock waves even if initial and boundary data are smooth. The presence of such discontinuities is a major challenge for any attempt to provide physically correct and stable solution of hyperbolic conservation laws. Although significant progress has been made over the past years, the numerical approximation of hyperbolic PDEs remains an active research area with many challenging issues to be addressed. The compressible Euler equations are considered as one of the most important and widely investigated topics of hyperbolic conservation laws.

There are several methods developed for numerically solving the compressible Euler equations. First-order methods such as the Godunov scheme are stable, yet very dissipative because of their low order of accuracy. Second-order accurate schemes are often used for compressible flows because they are robust and accurate enough for many applications. High-order methods have gained increasing attention in recent years because of the need for high-order accuracy and low dissipation in applications such as direct numerical simulation, large eddy simulation, and computational aeroacoustics. The hybridizable discontinuous Galerkin (HDG) method is a new efficient high-order discontinuous Galerkin approach. The first HDG method was introduced for diffusion-reaction problems[8] and later analyzed in.[4,10,11] Several HDG methods are subsequently developed for biharmonic equations,[5] linear and nonlinear convection-diffusion problems,[6,31,32] linear elasticity,[42] Stokes flows,[7,9,12,33] incompressible Navier-Stokes equations,[29,30,34] compressible Navier-Stokes equations,[36] linear acoustic and elastodynamics,[28] and time-harmonic Maxwell's equations.[27]

In this paper, we develop a HDG method for the compressible Euler equations with particular emphasis on compressible flows with shock waves. There are four main ingredients in the HDG methodology. The first ingredient is a discontinuous Galerkin discretization of the governing equations at the element level which is called the *local solver* since it defines the conserved variables as a function of their approximate traces on the element boundary. The second ingredient is an explicit enforcement of the conservative property

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

of the numerical fluxes through a global weak formulation. The third ingredient is the hybridization or static condensation of the local solver in the global weak formulation to arrive, after a Newton-Rapson linearization, at a final system in terms of the approximate traces only. And the four ingredient is the artificial viscosity model added to the original governing equations for the purpose of shock capturing. The HDG method inherits the geometric flexibility and high-order accuracy of discontinuous Galerkin methods, and offers a significant reduction in the computational cost. The efficiency stems mainly from the fact that the approximate traces are defined and single-valued along the element border.

Artificial viscosity has been widely used in finite volume methods,[24] streamline upwind Petrov-Galerkin (SUPG) methods,[23] and spectral methods.[43] Recently, DG researchers have also employed artificial viscosity to capture shocks. Hartmann and Houston[20] used the magnitude of the residual to determine the amount of viscosity added to shock region. Persson and Peraire[37] introduced a sub-cell shock-capturing method based on the smoothness of an orthogonal expansion of the computed density for determining the shock region and amount of artificial viscosity added there. However, a drawback of this approach is that it may lead to oscillations in state gradients because the artificial viscosity is piecewise-constant. Recognizing this limitation by the Persson and Peraire's approach, Barter and Darmofal introduced a PDE-based artificial viscosity model[1] appended to the system of governing equations to obtain a smoother artificial viscosity. However, the PDE-based artificial viscosity approach is clearly more expensive since it solves an additional PDE for the artificial viscosity. Yet another approach proposed by Cook and Cabot[15–17] consists of adding artificial terms to the physical viscosity coefficients such as the dynamic viscosity, bulk viscosity, and thermal conductivity. The added artificial terms are determined based on the strain rate tensor and the internal energy. This approach was followed up with the work by Lele et al.[3, 19] in the context of compressible turbulence simulations. The approach was also adopted by Premasuthan et al.[40] for spectral difference method.

In this paper, we extend the previous work[1, 3, 15–17, 19, 37, 40] in some important ways. In our artificial viscosity model, the artificial viscosity is an analytic function of the dilatation. This function increases smoothly as the dilatation gets negative, so that artificial viscosity is only added to regions of strong negative dilatation. On the other hand, it vanishes as soon as the dilatation becomes positive, so that artificial viscosity is not added in regions of positive dilatation. In addition, we use the artificial viscosity as an indicator for adaptive grid refinement to improve shock profiles. We demonstrate the proposed approach through a number of test cases in subsonic, transonic, supersonic, and hypersonic regimes.

The paper is organized as follows. We introduce the HDG method for the compressible Euler equations in Section 2 and in the compressible Navier-Stokes equations in Section 3. In each section, we formulate the method, briefly describe its implementation, and discuss the choice of the stabilization matrix and the treatment of the boundary conditions. In Section 4 we provide numerical results to assess the performance of the method. Finally, in Section 5 we present some concluding remarks.

II. An Artificial Viscosity Model for Shock Capturing

A. Governing equations

In the presence of shock waves, a direct DG discretization of the compressible Euler equations will produce oscillatory solution — the so-called Gibbs phenomenon — which eventually causes the scheme to blow up. In order to capture shocks, we introduce an artificial viscosity term to the Euler equations. In particular, the compressible Euler equations of gas dynamics with an artificial viscosity term in the physical domain $\Omega \subset \mathbb{R}^d$ are written in nondimensional conservation form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{F}(\boldsymbol{u}) + \boldsymbol{G}(\varepsilon, \boldsymbol{u}, \nabla \boldsymbol{u})) = 0, \tag{1}$$

where $\boldsymbol{u} = (\rho, \rho v_i, \rho E)^T$ is the m-dimensional vector of conserved dimensionless quantities (namely, density, momentum and energy), $\boldsymbol{F}(\boldsymbol{u})$ are the inviscid fluxes of dimension $m \times d$, $\boldsymbol{G}(\varepsilon, \boldsymbol{u}, \nabla \boldsymbol{u})$ are the viscous fluxes of dimension $m \times d$ artificially added to the Euler equations for the purpose of capturing shocks, and ε is a scalar artificial viscosity. The system (1) must be supplemented with appropriate boundary conditions at the inflow and outflow boundaries and at the solid wall. We shall discuss these boundary conditions in Subsection 3.3.

The flux vectors in the above system are given by

$$\boldsymbol{F}_i = \begin{pmatrix} \rho v_i \\ \rho v_i v_j + \delta_{ij} p \\ \rho v_i H \end{pmatrix}, \qquad \boldsymbol{G}_i = -\varepsilon \frac{\partial \tilde{\boldsymbol{u}}}{\partial x_i}, \tag{2}$$

where $p = (\gamma - 1)\rho(E - 0.5v_i^2)$ is the static pressure, $H = E + p/\rho$ is the total enthalpy, δ_{ij} is the Kronecker delta symbol, γ is the specific heat ratio, and $\tilde{\boldsymbol{u}} = (\rho, \rho v_i, \rho H)^T$. Note in the definition of $\boldsymbol{G}$ that we choose $\partial\tilde{\boldsymbol{u}}/\partial x_i$ over $\partial \boldsymbol{u}/\partial x_i$ in order to preserve the total enthalpy.[1,18] The artificial viscosity ε is determined as follows.

B. Definition of the artificial viscosity

It is known that the presence of shock waves corresponds to strongly negative dilatation, that is, $\nabla \cdot \boldsymbol{v}$ is strongly negative. This physical observation is exploited by some researchers to determine the amount of artificial viscosity. In particular, Bhagatwala and Lele [3] design a shock sensor that is switched off for positive dilatation whereas it rises slowly as the negative dilatation increases in magnitude. Premasuthan et al[40] then modify the artificial viscosity formulation proposed in[3] to add artificial bulk viscosity that is scaled as the negative dilatation.

In this paper, we define the artificial viscosity as

$$\varepsilon = \varepsilon_0 f\Big(\frac{\ell \nabla \cdot \boldsymbol{v}}{c}\Big), \tag{3}$$

where ε_0 is a user-specified constant, ℓ is a characteristic length scale, $c = \sqrt{\gamma p/\rho}$ is the sound speed, and f is an analytic function. Not wanting to add viscosity at the wall, we specify ℓ as

$$\ell = \min(h_0, 10 d_w), \tag{4}$$

where h_0 is a representative size of the finite elements and d_w is the distance from the closest wall. To complete our artificial viscosity model we define f as

$$f(x) = \alpha \log(1 + \exp((\beta - x)/\alpha)), \tag{5}$$

for $\alpha = 0.1$ and $\beta = -0.5$. As plotted in Figure 1(a), the analytic function $f(x)$ is a smooth approximation of the ramp function $r(x) = \beta - x$ if $x \leq \beta$ and $r(x) = 0$ if $x > \beta$. It increases as the dilatation gets more and more negative, so that the artificial viscosity is added to regions of strong negative dilatation (compression shocks). On the other hand, it vanishes rapidly as the dilatation becomes positive, so that no artificial viscosity is added to regions of positive dilatation.

Note that the parameters ε_0 and h_0 play crucial yet different roles in our artificial viscosity model. While ε_0 ultimately determines the amount of artificial viscosity, h_0 determines the thickness of the artificial viscosity. Hence, one can control how viscosity is added to the shock region by varying these parameters. The two parameters will be left to be problem-dependent because they generally depend on the Mach number, the polynomial degree, the geometry, the computational grid, and of course the numerical scheme.

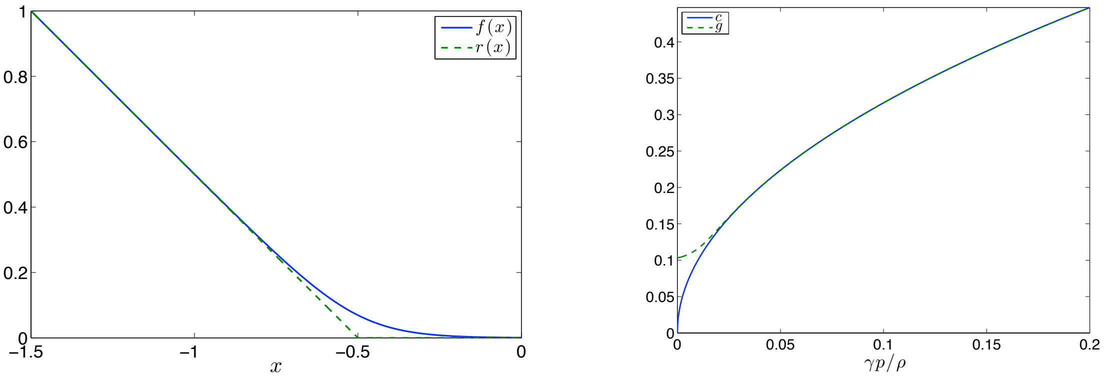
C. Modification for Hypersonic Applications

For problems with strong shock waves such as hypersonic applications, the pressure is typically very small at the shock so that the sound speed c is very stiff and close to zero there. In this case, the artificial viscosity model (3) becomes very sensitive to small pressure, which may cause the scheme to blow up. To address this issue, we propose a modified artificial viscosity as follows

$$\varepsilon = \varepsilon_0 f\Big(\frac{\ell \nabla \cdot \boldsymbol{v}}{g}\Big), \tag{6}$$

where g is a function of the sound speed and defined as

$$g(c) = c_\infty\{1 + 0.5\log(1 + \exp(2(c^2/c_\infty^2 - 1)))\}^{1/2}, \tag{7}$$



(a)

(b)

Figure 1. Plots of (a) the function f in (5) and (b) the function g in (7).

with $c_\infty = \sqrt{\gamma p_\infty/\rho_\infty}$ being the sound speed at far field. As shown in Figure 1(b) g is nothing but a smooth approximation of the sound speed c.

For transonic and supersonic applications, $g(c)$ and c behave in a similar manner. Therefore, the formula (6) can also be used for transonic and supersonic flows as well, and in fact produce similar results as the previous formula (3). However, for hypersonic applications, the modified version (6) is necessary because it is found to be more robust and stable than (3).

III. The Hybridizable Discontinuous Galerkin Method

A. Notation

The departure point of HDG method is to write the original system (1) as a first-order system of equations

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

To describe the HDG method for numerically solving the above system, we need to introduce some notation. Since the artificial viscosity ε in (8) is a function of $(\boldsymbol{u}, \boldsymbol{q})$, we shall write $\boldsymbol{G}(\boldsymbol{u}, \boldsymbol{q})$ in the remainder of this paper.

We denote by $\mathcal{T}_h$ a collection of disjoint regular elements K that partition Ω and set $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$. For an element K of the collection $\mathcal{T}_h$, $F = \partial K \cap \partial\Omega$ is the boundary face if the $d-1$ 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 $d-1$ measure of F is nonzero. We denote by $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$ the set of interior and boundary faces, respectively. We set $\mathcal{E}_h = \mathcal{E}_h^o \cup \mathcal{E}_h^\partial$.

Let $\mathcal{P}^k(D)$ denote the space of polynomials of degree at most k on a domain D and let $L^2(D)$ be the space of square integrable functions on D. We introduce the following discontinuous finite element approximation space

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

In addition, we introduce a finite element approximation space for the approximate trace of the solution

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

Note that $\boldsymbol{M}_h$ consists of functions which are continuous inside the faces (or edges) $F \in \mathcal{E}_h$ and discontinuous at their borders.

Finally, we define various inner products for our finite element spaces. We write $(w, v)_{\mathcal{T}_h} := \sum_{K \in \mathcal{T}_h} (w, v)_K$, where $(w, v)_D$ denotes the integral of $w\,v$ over the domain $D \subset \mathbb{R}^d$ for $w, v \in P_h$. We also write $(\boldsymbol{w}, \boldsymbol{v})_{\mathcal{T}_h} :=$

$\sum_{i=1}^m (w_i, v_i)_{\mathcal{T}_h}$ for $\boldsymbol{w}, \boldsymbol{v} \in \boldsymbol{W}_h^k$ and $(\boldsymbol{s}, \boldsymbol{r})_{\mathcal{T}_h} := \sum_{i=1}^m \sum_{j=1}^d (s_{ij}, r_{ij})_{\mathcal{T}_h}$ for $\boldsymbol{s}, \boldsymbol{r} \in \boldsymbol{Q}_h^k$ We then write $\langle \eta, \zeta \rangle_{\partial \mathcal{T}_h} := \sum_{K \in \mathcal{T}_h} \langle \eta, \zeta \rangle_{\partial K}$ and $\langle \boldsymbol{\eta}, \boldsymbol{\zeta} \rangle_{\partial \mathcal{T}_h} := \sum_{i=1}^m \langle \eta_i, \zeta_i \rangle_{\partial \mathcal{T}_h}$, for $\boldsymbol{\eta}, \boldsymbol{\zeta} \in \boldsymbol{M}_h^k$, where $\langle \eta, \zeta \rangle_D$ denotes the integral of $\eta \zeta$ over the domain $D \subset \mathbb{R}^{d-1}$.

B. Formulation

We begin by considering the governing equations (8) on one element K of $\mathcal{T}_h$ and seeking an approximation $(\boldsymbol{q}_h, \boldsymbol{u}_h) \in \boldsymbol{Q}_h^k \times \boldsymbol{W}_h^k$ such that it satisfies

$$\begin{aligned} (\boldsymbol{q}_h, \boldsymbol{s})_K + (\boldsymbol{u}_h, \nabla \cdot \boldsymbol{s})_K - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{s} \cdot \boldsymbol{n} \rangle_{\partial K} &= 0, \\ \left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w} \right)_K - (\boldsymbol{F}(\boldsymbol{u}_h) + \boldsymbol{G}(\boldsymbol{u}_h, \boldsymbol{q}_h), \nabla \boldsymbol{w})_K + \left\langle \widehat{\boldsymbol{H}}_h \cdot \boldsymbol{n}, \boldsymbol{w} \right\rangle_{\partial K} &= 0, \end{aligned} \tag{9}$$

for all $(\boldsymbol{w}, \boldsymbol{s}) \in (\mathcal{P}^k(K))^m \times (\mathcal{P}^k(K))^{m \times d}$. Here the numerical trace $\widehat{\boldsymbol{u}}_h$ (respectively, the numerical flux $\widehat{\boldsymbol{H}}_h$) is an approximation to $\boldsymbol{u}$ (respectively, $\boldsymbol{F}(\boldsymbol{u}) + \boldsymbol{G}(\boldsymbol{u}, \boldsymbol{q})$) over ∂K. The numerical flux $\widehat{\boldsymbol{H}}_h^n$ is defined as

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

where $\boldsymbol{S}$ is a *stabilization matrix* which has an important role in ensuring stability and accuracy of the method. The selection of the stabilization matrix is discussed in Subsection 3.3.

The above system is then discretized in time using backward difference formulaes (BDF) for the discretization of the time derivative. For instance, using the Backward-Euler scheme at time level t^n with timestep Δt^n we obtain

$$\begin{aligned} (\boldsymbol{q}_h^n, \boldsymbol{s})_K + (\boldsymbol{u}_h^n, \nabla \cdot \boldsymbol{s})_K - \langle \widehat{\boldsymbol{u}}_h^n, \boldsymbol{s} \cdot \boldsymbol{n} \rangle_{\partial K} &= 0, \\ \left(\frac{\boldsymbol{u}_h^n}{\Delta t^n}, \boldsymbol{w} \right)_K - (\boldsymbol{F}(\boldsymbol{u}_h^n) + \boldsymbol{G}(\boldsymbol{u}_h^n, \boldsymbol{q}_h^n), \nabla \boldsymbol{w})_K + \left\langle \widehat{\boldsymbol{H}}_h^n \cdot \boldsymbol{n}, \boldsymbol{w} \right\rangle_{\partial K} &= \left(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{w} \right)_K, \end{aligned} \tag{11}$$

for all $(\boldsymbol{w}, \boldsymbol{s}) \in (\mathcal{P}^k(K))^m \times (\mathcal{P}^k(K))^{m \times d}$, where $\boldsymbol{q}_h^n = \boldsymbol{q}_h(t^n)$, $\boldsymbol{u}_h^n = \boldsymbol{u}_h(t^n)$, and $\widehat{\boldsymbol{u}}_h^n = \widehat{\boldsymbol{u}}_h(t^n)$. At this point, we remark that one can directly solve (11) in an element-by-element fashion if $\widehat{\boldsymbol{u}}_h^n$ is known. As a result, the system (11) defines a mapping $\mathcal{L}_h$ (which we shall refer to as *local solver*) that maps every $\widehat{\boldsymbol{u}}_h^n \in \boldsymbol{M}_h^k$ to $(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n) \in \boldsymbol{Q}_h^k \times \boldsymbol{W}_h^k$. Hence, we can write

$$(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n) = \mathcal{L}_h(\widehat{\boldsymbol{u}}_h^n), \tag{12}$$

which indicates that $(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n)$ is a function of $\widehat{\boldsymbol{u}}_h^n$.

By summing the above system over all the elements and enforcing the consertivity of the numerical flux $\widehat{\boldsymbol{H}}_h^n$ and boundary conditions, we arrive at the following problem: Find $(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n) \in \boldsymbol{Q}_h^k \times \boldsymbol{W}_h^k \times \boldsymbol{M}_h^k$ such that

$$\begin{aligned} (\boldsymbol{q}_h^n, \boldsymbol{s})_{\mathcal{T}_h} + (\boldsymbol{u}_h^n, \nabla \cdot \boldsymbol{s})_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{u}}_h^n, \boldsymbol{s} \cdot \boldsymbol{n} \rangle_{\partial \mathcal{T}_h} &= 0, \\ \left(\frac{\boldsymbol{u}_h^n}{\Delta t^n}, \boldsymbol{w} \right)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h^n) + \boldsymbol{G}(\boldsymbol{u}_h^n, \boldsymbol{q}_h^n), \nabla \boldsymbol{w})_{\mathcal{T}_h} + \left\langle \widehat{\boldsymbol{H}}_h^n \cdot \boldsymbol{n}, \boldsymbol{w} \right\rangle_{\partial \mathcal{T}_h} &= \left(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{w} \right)_{\mathcal{T}_h}, \\ \left\langle \widehat{\boldsymbol{H}}_h^n \cdot \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h \backslash \partial \Omega} + \left\langle \widehat{\boldsymbol{B}}_h^n, \boldsymbol{\mu} \right\rangle_{\partial \Omega} &= 0, \end{aligned} \tag{13}$$

for all $(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{\mu}) \in \boldsymbol{Q}_h^k \times \boldsymbol{W}_h^k \times \boldsymbol{M}_h^k$. Here $\widehat{\boldsymbol{B}}_h^n$ is the numerical flux vector of dimension m and defined over the boundary $\partial \Omega$. Its precise definition depends on the types of boundary conditions and will be given below. Note that $\widehat{\boldsymbol{u}}_h^n$ is single-valued over each face since $\widehat{\boldsymbol{u}}_h^n$ belongs to $\boldsymbol{M}_h^k$. Furthermore, we have $\mathsf{P}([\![\widehat{\boldsymbol{H}}_h^n \cdot \boldsymbol{n}]\!]) = 0$, where P denotes the L^2 projection into $\boldsymbol{M}_h^k$ and $[\![\,]\!]$ denotes the usual jump operator. Therefore, the method is conservative since only the projection of $\widehat{\boldsymbol{H}}_h^n \cdot \boldsymbol{n}$ is needed in the above system.

Because $(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n)$ depends on $\widehat{\boldsymbol{u}}_h^n \in \boldsymbol{M}_h^k$ through the local solver (11), we can equivalently state that $\widehat{\boldsymbol{u}}_h^n \in \boldsymbol{M}_h^k$ is the solution of the following weak formulation

$$\left\langle \widehat{\boldsymbol{H}}_h^n \cdot \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h \backslash \partial \Omega} + \left\langle \widehat{\boldsymbol{B}}_h^n, \boldsymbol{\mu} \right\rangle_{\partial \Omega} = 0, \quad \forall \boldsymbol{\mu} \in \boldsymbol{M}_h^k. \tag{14}$$

Here both $\widehat{\boldsymbol{H}}_h^n$ and $\widehat{\boldsymbol{B}}_h^n$ should be understood as functions of $\widehat{\boldsymbol{u}}_h^n$ only since $(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n)$ is a function of $\widehat{\boldsymbol{u}}_h^n$ through the local solver $\mathcal{L}_h(\widehat{\boldsymbol{u}}_h^n)$. In essence, this weak formulation characterizes $\widehat{\boldsymbol{u}}_h^n$ as the *global* unknown, whereas the local solver (11) characterizes $(\boldsymbol{q}_h^n, \boldsymbol{u}_h^n)$ as the *local* unknowns. Hence, the HDG method produces a final algebraic system involving only the degrees of freedom of the approximate trace $\widehat{\boldsymbol{u}}_h^n$.

C. Stabilization Matrix

We propose here three schemes to define the stabilization matrix. In the first scheme, we choose

$$\boldsymbol{S} = \boldsymbol{L}|\boldsymbol{\Lambda}|\boldsymbol{R}, \tag{15}$$

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

$$\boldsymbol{S} = \lambda^{\ell}_{\max}\boldsymbol{I}, \tag{16}$$

where $\lambda^{\ell}_{\max} = |\boldsymbol{v}_h \cdot \boldsymbol{n}| + c_h$ is the local maximum speed of the system, and $\boldsymbol{I}$ is the identity matrix. The third scheme inspired by the global Lax-Friedrich method involves choosing

$$\boldsymbol{S} = (\|\boldsymbol{v}_\infty\| + c_\infty)\boldsymbol{I}, \tag{17}$$

where $\boldsymbol{v}_\infty$ and c_∞ is the velocity and sound speed at far field, respectively.

The choice of the stabilization matrix becomes less critical for high-order DG methods as numerical dissipation in the order of $O(h^{k+1})$ vanishes rapidly with increasing k. Because the third scheme is very simple to implement and less expensive than the other schemes, we shall employ it in the numerical examples unless we indicate otherwise.

D. Boundary Conditions

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 extrapolate it depending on the eigenvalues of the system. To this end, we define the boundary flux vector $\widehat{\boldsymbol{B}}_h$ as

$$\widehat{\boldsymbol{B}}_h = \boldsymbol{A}^+_n(\widehat{\boldsymbol{u}}_h)(\boldsymbol{u}_h - \widehat{\boldsymbol{u}}_h) - \boldsymbol{A}^-_n(\widehat{\boldsymbol{u}}_h)(\boldsymbol{u}_\infty - \widehat{\boldsymbol{u}}_h), \tag{18}$$

where $\boldsymbol{A}_n = \boldsymbol{A} \cdot \boldsymbol{n}$ and $\boldsymbol{A}^\pm_n = (\boldsymbol{A}_n \pm |\boldsymbol{A}_n|)/2$. Here $\boldsymbol{A}_n = [\partial \boldsymbol{F}/\partial \boldsymbol{u}] \cdot \boldsymbol{n}$ denotes the Jacobian of the inviscid normal flux to the boundary.

At the solid surface with slip condition, we must impose zero normal velocity. Henceforth, we set

$$\widehat{\boldsymbol{B}}_h = \boldsymbol{S}(\boldsymbol{b}_h - \widehat{\boldsymbol{u}}_h), \tag{19}$$

where the vector $\boldsymbol{b}_h$ is defined in terms of $\boldsymbol{u}_h$ as follows

$$\boldsymbol{b}_h[1] = \boldsymbol{u}_h[1], \quad \boldsymbol{b}_h[2, \dots, m-1] = \boldsymbol{v}_h - \boldsymbol{n}v_n, \quad \boldsymbol{b}_h[m] = \boldsymbol{u}_h[m]. \tag{20}$$

Here $\boldsymbol{v}_h = \boldsymbol{u}_h[2, \dots, m-1]$ are the velocity components of $\boldsymbol{u}_h$ and $v_n = \boldsymbol{v}_h \cdot \boldsymbol{n}$ is the normal component of the approximate velocity. Note that since $(\boldsymbol{v}_h - \boldsymbol{n}v_n) \cdot \boldsymbol{n} = 0$ we have $\widehat{\boldsymbol{v}}_h \cdot \boldsymbol{n} = 0$, where $\widehat{\boldsymbol{v}}_h = \widehat{\boldsymbol{u}}_h[2, \dots, m-1]$ are the velocity components of $\widehat{\boldsymbol{u}}_h$.

E. Implementation

The HDG method described here can be implemented in a straightforward manner following from the procedure outlined in our previous work.[29, 31, 32] For completeness, however, we provide a short discussion and refer to[29, 31, 32] for further details. By applying the Newton-Raphson method to linearize the nonlinear system (13), we obtain the following linear system at every Newton iteration

$$\begin{bmatrix} \mathbb{A}^n & \mathbb{B}^n & \mathbb{C}^n \\ \mathbb{D}^n & \mathbb{E}^n & \mathbb{F}^n \\ \mathbb{G}^n & \mathbb{H}^n & \mathbb{I}^n \end{bmatrix} \begin{pmatrix} \delta Q^n \\ \delta U^n \\ \delta \Lambda^n \end{pmatrix} = \begin{pmatrix} M^n \\ N^n \\ P^n \end{pmatrix}, \tag{21}$$

where $\delta Q^n, \delta U^n$ and $\delta \Lambda^n$ are the vectors of degrees of freedom of $\delta \boldsymbol{q}^n_h$, $\delta \boldsymbol{u}^n_h$ and $\delta \widehat{\boldsymbol{u}}^n_h$, respectively, which are in that order the Newton increments of $\boldsymbol{q}^n_h$, $\boldsymbol{u}^n_h$ and $\widehat{\boldsymbol{u}}^n_h$. Once the linear system is solved we can update the current solution $\boldsymbol{q}^n_h := \boldsymbol{q}^n_h + \gamma \delta \boldsymbol{q}^n_h$, $\boldsymbol{u}^n_h := \boldsymbol{u}^n_h + \gamma \delta \boldsymbol{u}^n_h$ and $\widehat{\boldsymbol{u}}^n_h := \widehat{\boldsymbol{u}}^n_h + \gamma \delta \widehat{\boldsymbol{u}}^n_h$, where γ is the damped Newton stepsize to guarantee the decrease of the residual at every Newton iteration. This procedure is repeated until the norm of the right-hand side vector in (21) is less than a specified tolerance.

To solve the above linear system, we note that the matrix $[\mathbb{A}^n \ \mathbb{B}^n; \mathbb{D}^n \ \mathbb{E}^n]$ has block-diagonal structure thanks to the local solver (11). Therefore, we can eliminate $(\delta Q^n, \delta U^n)$ to obtain a reduced system in terms of $\delta\Lambda^n$ as

$$\mathbb{K}^n \ \Lambda^n = R^n \tag{22a}$$

where

$$\mathbb{K}^n = -\begin{bmatrix} \mathbb{G}^n & \mathbb{H}^n \end{bmatrix} \begin{bmatrix} \mathbb{A}^n & -\mathbb{B}^n \\ \mathbb{D}^n & \mathbb{E}^n \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{C}^n \\ \mathbb{F}^n \end{bmatrix} + \mathbb{I}^n, \tag{22b}$$

and

$$R^n = P^n - \begin{bmatrix} \mathbb{G}^n & \mathbb{H}^n \end{bmatrix} \begin{bmatrix} \mathbb{A}^n & \mathbb{B}^n \\ \mathbb{D}^n & \mathbb{E}^n \end{bmatrix}^{-1} \begin{bmatrix} M^n \\ N^n \end{bmatrix}. \tag{22c}$$

This is the global system to be solved at every Newton iteration. Since $\widehat{\boldsymbol{u}}_h^n$ is defined and 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 $\mathbb{K}^n$ is compact in the sense that only the degrees of freedom between neighboring faces that share the same element are connected.

Note that one does not need to explicitly construct the full block-dgiagonal matrix $[\mathbb{A}^n \ \mathbb{B}^n; \mathbb{D}^n \ \mathbb{E}^n]$ to compute the stiffness matrix $\mathbb{K}^n$ and residual vector R^n. Instead, one can obtain the system (21) by employing the standard finite element assembly in which the global quantities are formed by assembling the corresponding elemental quantities. This procedure is given in.[29,31,32]

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. Numerical Results

In this section, we present numerical results for several steady-state test cases. Although the problems are steady, we use the backward Euler method with variable timestep to obtain the steady-state solution. In particular, we start from the initial freestream condition and gradually increase the timestep after every time steps. Once the solution on an initial grid is obtained, we refine the grid and directly use the Newton-Raphson method to compute the steady-state solution on the refined grid without time-marching. The number of Newton iterations required to converge to the tolerance 10^{-8} is less than 10. Note here that the results are obtained without using a homotopy sequence in the polynomial degree k and in the Mach number. In most examples, we shall use $k = 4$ to demonstrate that our method can capture shock robustly even though the solution is represented by high-order polynomials. Of course, we have also obtained the results for lower k, but we choose not to present them to save space. We would like to refer the reader to the previous work[1,37] which extensively discuss the effect of k and h on shock capturing with artificial viscosity in the DG context.

A. Ringleb Flow

We first consider the Ringleb flow to demonstrate the optimal accuracy of the HDG method. The Ringleb flow is an exact smooth solution of the Euler equations obtained using the hodograph method. For any given (x, y), we first obtain the radial velocity V by solving the following nonlinear equation

$$(x - 0.5L^2) + y^2 = \frac{1}{4\rho^2 V^4}, \tag{23}$$

where

$$\begin{aligned} c &= \sqrt{1 - V^2/5}, \\ \rho &= c^5, \\ L &= \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\ln\frac{1+c}{1-c}. \end{aligned} \tag{24}$$

We then compute

$$\begin{aligned} p &= c^7/\gamma, \\ \psi &= \sqrt{0.5/V^2 - (x - 0.5L)\rho}, \\ \theta &= \arcsin(\psi V), \\ v_1 &= V\cos(\theta), \\ v_2 &= V\sin(\theta). \end{aligned} \tag{25}$$

Since the exact solution can be determined for any spatial point, we take the domain Ω to be $(-2,-1)\times(1,2)$. The boundary condition is imposed by setting the exact state on the boundary of the domain and using (18). We consider triangular meshes that are obtained by splitting a regular $n \times n$ Cartesian grid into $2n^2$ triangles. On these meshes, we use polynomials of degree k to represent all the approximate variables with a nodal basis.[22] Since the solution is smooth we set $\varepsilon_0 = 0$, so that no viscosity is added.

We present in Table 1 the L^2 error and convergence rate of the numerical solution as a function of $h = 1/n$ and k. We observe that the approximate solution converges with the optimal order $k+1$. Hence, the convergence of the HDG method is optimal.

mesh	$k=1$		$k=2$		$k=3$		$k=4$	
$1/h$	error	order	error	order	error	order	error	order
2	4.35e-3	$--$	3.24e-4	$--$	2.35e-5	$--$	2.08e-6	$--$
4	1.10e-3	1.98	4.85e-5	2.74	1.43e-6	4.04	7.90e-8	4.72
8	2.80e-4	1.98	6.92e-6	2.81	8.63e-8	4.05	2.80e-9	4.82
16	7.06e-5	1.99	9.37e-7	2.88	5.18e-9	4.06	9.36e-11	4.90

Table 1. History of convergence of the approximate solution for the Ringleb flow.

B. Transonic Flow Past a RAE 2822 Airfoil

The second example involves the transonic flow past a RAE 2822 airfoil. The freestream mach number is 0.75 and the angle of attack is 3.00 degrees. Freestream conditions are prescribed at the outer boundaries. On the airfoil surface, inviscid wall boundary condition is used. A weak shock is formed on the upper surface nearer to the trailing edge. The finite element mesh consists of 1088 triangles as shown in Figure 4. Isoparametric elements with the polynomials of degree $k = 4$ are used to represent both the unknowns and geometry.

We study here the effect of the two parameters ε_0 and h_0 on the shock profiles. We present in Figure 2 the pressure and artificial viscosity obtained for four different cases: (a) $(\varepsilon_0, h_0) = (0.002, 0.25)$, (b) $(\varepsilon_0, h_0) = (0.002, 0.5)$, (c) $(\varepsilon_0, h_0) = (0.004, 0.25)$, and (d) $(\varepsilon_0, h_0) = (0.004, 0.5)$. We observe that the shock profile is smoother as we increase ε_0 and h_0 and that artificial viscosity is only added to the shock region. Moreover, increasing ε_0 leads to more viscosity added to the shock region, while increasing h_0 renders the artificial viscosity region thicker. In order to better visualize the effect of these two parameters, we plot the pressure coefficient C_p in Figure 3. We see that away from the shock region the pressure coefficient C_p are the same for all of the four cases. At the shock region, (a) and (b) produce sharper shock profiles (albeit at a slight oscillation) than (c) and (d). It appears that the best shock profile is associated with (c).

C. Transonic Flow Past a NACA 0012 Airfoil

The third example involves the transonic flow past a NACA 0012 airfoil at angle of attack $\alpha = 1.5^o$ and freestream Mach number $M_\infty = 0.8$. A weak shock is formed on the upper surface, while another even weaker shock is formed under the lower surface. Figure 5 shows the original unstructured grid of 916 elements used in the computation. Isoparametric elements with the polynomials of degree $k = 4$ are used to represent both the unknowns and geometry.

In this example, we aim to demonstrate the effectiveness and efficiency of adaptive grid refinement for improving the shock profiles. To this end, we show in Figure 6 the pressure and artificial viscosity obtained on the initial grid and three successive refinements. The parameters are set as (a) $(\varepsilon_0, h_0) = (0.003, 0.3)$ for the initial mesh with 916 elements, (b) $(\varepsilon_0, h_0) = (0.002, 0.2)$ for the first refinement with 1030 elements, (c) $(\varepsilon_0, h_0) = (0.0015, 0.15)$ for the second refinement with 1296 elements, and (d) $(\varepsilon_0, h_0) = (0.0010, 0.1)$ for the third refinement with 1940 elements. We observe that the shocks are better resolved with increasing the refinement level and that both the amount and thickness of viscosity are reduced as well. We further see that artificial viscosity is added only in shock regions and that grid refinement is carried out only in those regions. Figure 8 displays the artificial viscosity (without grid) on the third refinement. We observe that the artificial viscosity is very smooth and that viscosity added to the lower shock is significantly less than viscosity added to the upper shock. This is because the dilatation of the upper shock is much larger than that of the lower shock. Figure 7 shows the pressure coefficient distribution over the airfoil surface and its close-up at the lower shock. It is clear that grid refinement improves the shock profile by smoothing and sharpening it.

D. Supersonic Flow Past a NACA 0012 Airfoil

The purpose of this test case is to examine the effectiveness of the artificial viscosity model for stronger shocks. The geometry, initial mesh, and flow conditions of this example are similar to those in the previous example except that the freestream Mach number is now $M_\infty = 2.0$. The supersonic flow regime generates two different shocks: a bow shock at upstream and an oblique shock at downstream. This can be seen in Figure 9 which shows the mach number and artificial viscosity obtained on the initial mesh and three successive refinements. The parameters are set as (a) $(\varepsilon_0, h_0) = (0.002, 0.2)$ for the initial mesh with 916 elements, (b) $(\varepsilon_0, h_0) = (0.0015, 0.15)$ for the first refinement with 1680 elements, (c) $(\varepsilon_0, h_0) = (0.0008, 0.08)$ for the second refinement with 3376 elements, and (d) $(\varepsilon_0, h_0) = (0.0004, 0.04)$ for the third refinement with 6952 elements.

We emphasize that our approach can capture quite strong shocks on relatively coarse grids. Moreover, the shocks are significantly smoother and sharper as we refine the initial grid around the shock regions. The artificial viscosity is only added to the shock regions and reduced with increasing the refinement level. As depicted in Figure 8, the artificial viscosity on the third refinement is not only small, but also smooth and thin. The results clearly indicate that the amount of viscosity is sufficient to produce smooth and sharp shock profiles while it does not affect the computed solution away from the shock regions.

E. Supersonic Flow Past a Circular Bump

This test case involves inviscid supersonic flow in a channel with a 4% thick circular bump on the bottom side. The length and height of the channel are 3 and 1, respectively. The inlet Mach number is $M_\infty = 1.4$. Inlet/outlet conditions are prescribed at the left/right boundaries, while inviscid wall boundary condition is used on the top and bottom sides. Isoparametric elements with the polynomials of degree $k = 4$ are used to represent both the unknowns and geometry.

We present in Figure 10 the pressure and artificial viscosity obtained on the initial grid and three successive refinements. The parameters are set as (a) $(\varepsilon_0, h_0) = (0.0015, 0.4)$ for the initial mesh with 600 elements, (b) $(\varepsilon_0, h_0) = (0.001, 0.3)$ for the first refinement with 1120 elements, (c) $(\varepsilon_0, h_0) = (0.0002, 0.2)$ for the second refinement with 2378 elements, and (d) $(\varepsilon_0, h_0) = (0.0001, 0.1)$ for the third refinement with 5016 elements. We observe again that the shocks are very sharp after three refinements and that artificial viscosity is added only to the shock region. As depicted in Figure 11, the artificial viscosity on the third refinement is very thin and smooth. Therefore, it does not affect the computed solution away from the shock region.

F. Hypersonic Flow Past a Circular Cylinder

The last test case involves hypersonic flow past a circular cylinder at $M_\infty = 7$. This test case serves to illustrate the effectiveness of our artificial viscosity approach for strong shocks in the hypersonic flow regime. The cylinder wall is treated as inviscid wall boundary condition. The rest of the boundary is treated as inlet/outlet conditions. Figure 13 shows the initial mesh with only 392 elements. Isoparametric elements with the polynomials of degree $k = 4$ are used to represent both the unknowns and geometry. The modified artificial viscosity model (6) is used here to obtain all of the results because the flow regime is hypersonic. The stabilization matrix is set to $\boldsymbol{S} = 10\|\boldsymbol{v}_\infty\|\boldsymbol{I}$.

We show in Figure 12 the pressure, Mach number, and artificial viscosity on the initial grid and three successive refinements. The parameters are set as (a) $(\varepsilon_0, h_0) = (0.02, 0.25)$ for the initial mesh with 392 elements, (b) $(\varepsilon_0, h_0) = (0.015, 0.20)$ for the first refinement with 692 elements, (c) $(\varepsilon_0, h_0) = (0.01, 0.10)$ for the second refinement with 1422 elements, and (d) $(\varepsilon_0, h_0) = (0.006, 0.06)$ for the third refinement with 3071 elements. The results on the initial grid indicate that our approach can capture strong shocks well on a coarse mesh. It should be noted that the amount of artificial viscosity added to the shock is larger than those in the previous test cases. This can be attributed to the fact that the Mach number in this example is much higher than those of the previous examples. However, the use of adaptive grid refinement significantly improves the shock profiles: as the mesh is refined, the shock structures become sharper, the amount of artificial viscosity added is lesser, and the solutions become more accurate. This can be clearly seen in Figure 14 which plots the pressure along the centerline $y = 0$. Notice how the shock profile is significantly improved as the grid is refined.

V. Conclusions

In this paper, we have presented a HDG method with shock capturing capability for compressible flow. The HDG method inherits the geometric flexibility and optimal accuracy of discontinuous Galerkin methods, yet offers a significant reduction in the computational cost. This mainly stems from the fact that the number of non-zero elements of the HDG stiffness matrix scales like $O(k^{d-1})$, while in other DG methods this number scales like $O(k^d)$. In addition, we propose an artificial viscosity model that is shown to work well for a wide range of flow regimes from transonic to hypersonic flows. Although our approach is based on the previous work[3, 15, 40] which uses the dilatation to define artificial viscosity, our artificial viscosity model is different from and in fact easier to integrate than the previous work's models.[3, 15, 40] Finally, we employ the artificial viscosity as an indicator for adaptive grid refinement to improve the shock profile and thus increase accuracy. We emphasize that the artificial viscosity added to the shock region is very smooth and thin, so that it does not exert its influence beyond the shock region.

VI. Acknowledgements

We are grateful to Professor Bernardo Cockburn of UMN for his longstanding collaboration on this project. We would also like to thank Mr. David Moro of MIT for fruitful discussion. We 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.

References

[1] 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.

[2] 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.

[3] A. Bhagatwala and S.K. Lele. A modified artificial viscosity approach for compressible turbulence simulations. *Journal of Computational Physics*, 228(14):4965–4969, 2009.

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

[5] B. Cockburn, B. Dong, and J. Guzmán. A hybridizable and superconvergent discontinuous Galerkin method for biharmonic problems. *J. Sci. Comput.*, 40(1-3):141–187, 2009.

[6] 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.

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

[8] 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.

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

[10] B. Cockburn, J. Gopalakrishnan, and F.-J. Sayas. A projection-based error analysis of HDG methods. *Math. Comp.* To appear.

[11] B. Cockburn, J. Guzmán, and H. Wang. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Math. Comp.*, 78:1–24, 2009.

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

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

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

[15] A. W. Cook. Artificial fluid properties for large-eddy simulation of compressible turbulent mixing. *Physics of Fluids*, 19(5):055103, 2007.

[16] A. W. Cook and W. H. Cabot. A high-wavenumber viscosity for high-resolution numerical method. *Journal of Computational Physics*, 195(2):594–601, 2004.

[17] A. W. Cook and W. H. Cabot. Hyperviscosity for shockturbulence interactions. *Journal of Computational Physics*, 203(2):379–385, 2005.

[18] M. Drela, A. Merchant, and J. Peraire. Elimination of Spurious Loss in Euler Equation Computations. *AIAA Journal*, 38(3):411–417, 2000.

[19] B. Fiorina and S K. Lele. An artificial nonlinear diffusivity method for supersonic reacting flows with shocks. *Journal of Computational Physics*, 222(1):246–264, 2007.

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

[21] 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.

[22] 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.

[23] 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.

[24] A. Jameson. Analysis and design of numerical schemes for gas dynamics, 1: Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics*, 4:171–218, 1994.

[25] 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.

[26] 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.

[27] N. C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. *Journal of Computational Physics.* In Press.

[28] N. C. Nguyen, J. Peraire, and B. Cockburn. HDG methods for acoustics and elastodynamics: Superconvergence and postprocessing. *J. Comput. Phys.*, 230:3695–3718, 2011.

[29] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 230:1147–1170, 2011.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] S. Premasuthan, C. Liang, and A. Jameson. Computation of flows with shocks using spectral difference scheme with artificial viscosity (AIAA Paper 2010-1449). In *Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, Janury 2010.

[41] 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.

[42] 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.

[43] E. Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 26(1):30–44, 1989.

[44] J. Von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21:232–237, 1950.

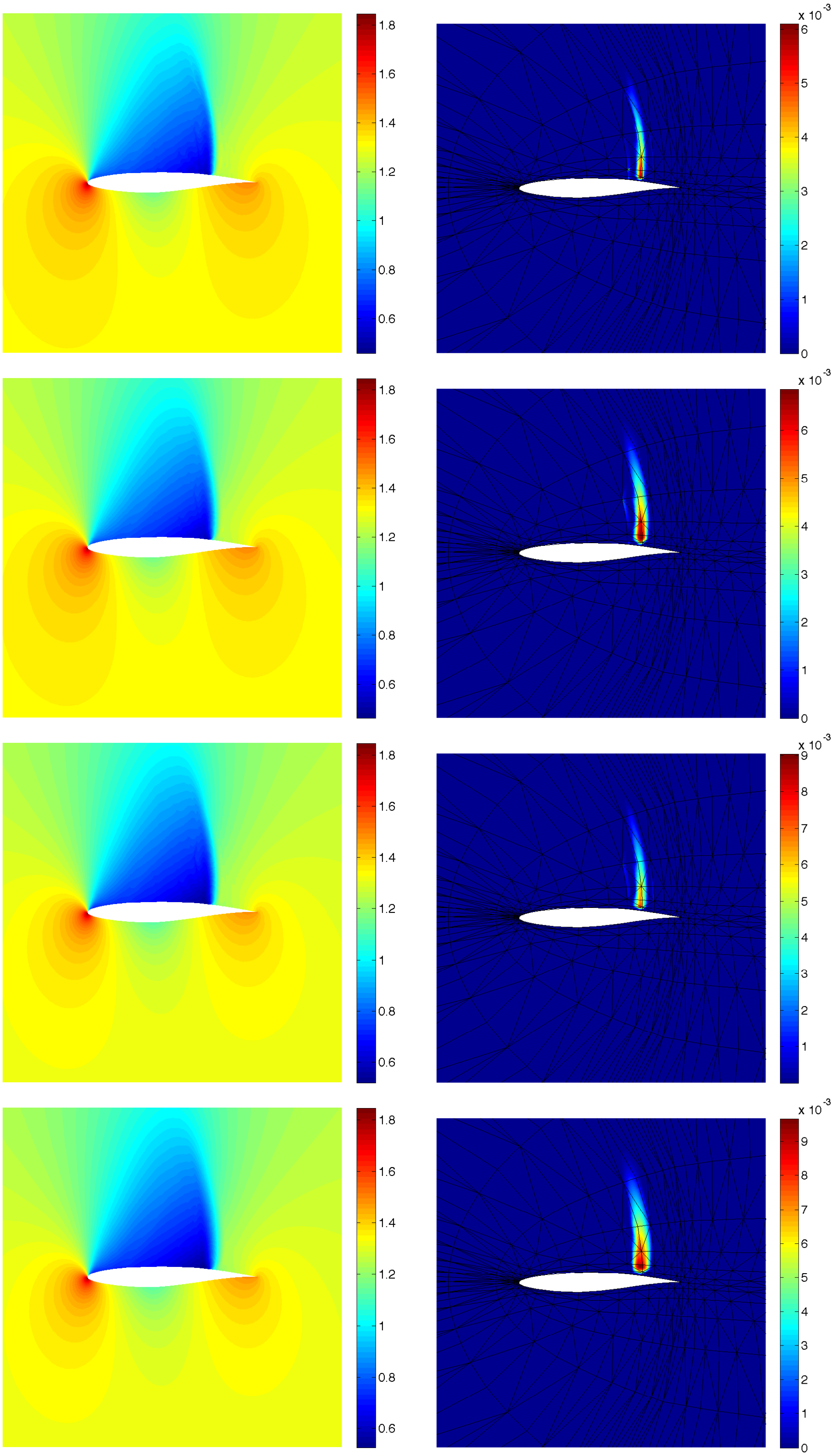


Figure 2. Pressure (left) and artificial viscosity (right) for transonic flow past a RAE 2822 airfoil: (a) $(\varepsilon_0, h_0) = (0.002, 0.25)$ **(top row), (b)** $(\varepsilon_0, h_0) = (0.002, 0.5)$ **(second row), (c)** $(\varepsilon_0, h_0) = (0.004, 0.25)$ **(third row), and (d)** $(\varepsilon_0, h_0) = (0.004, 0.5)$ **(bottom row).**

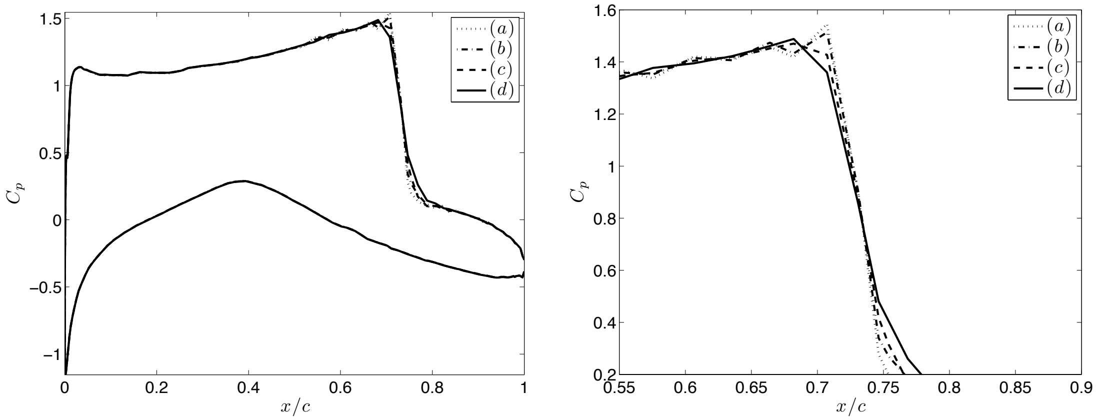


Figure 3. The pressure coefficient distribution over the RAE 2822 surface and its close-up at the shock profile.

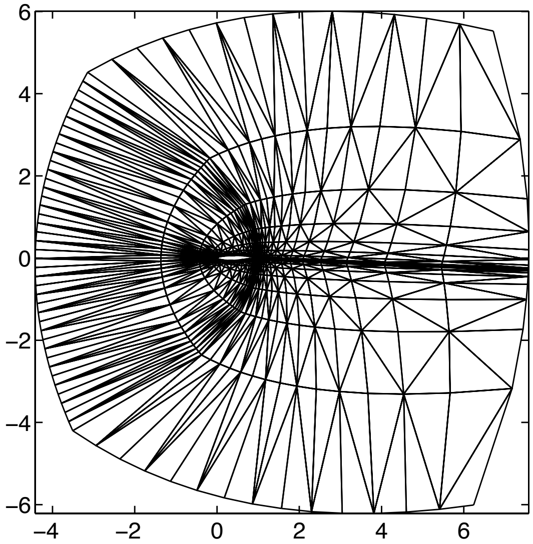


Figure 4. The finite element mesh for RAE 2822 geometry consists of 1088 triangles.

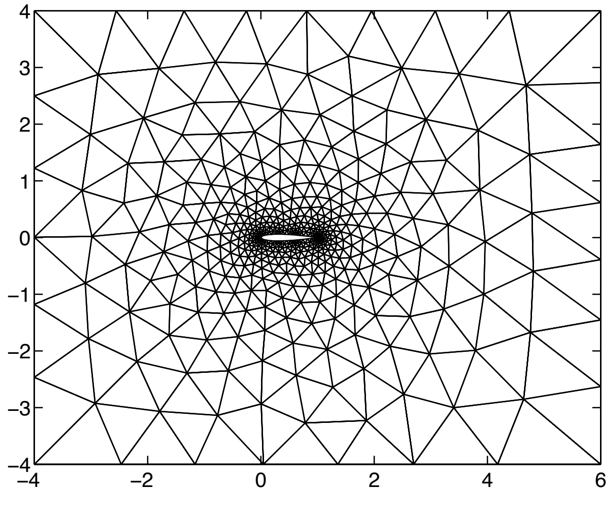


Figure 5. The initial mesh for NACA 0012 geometry.

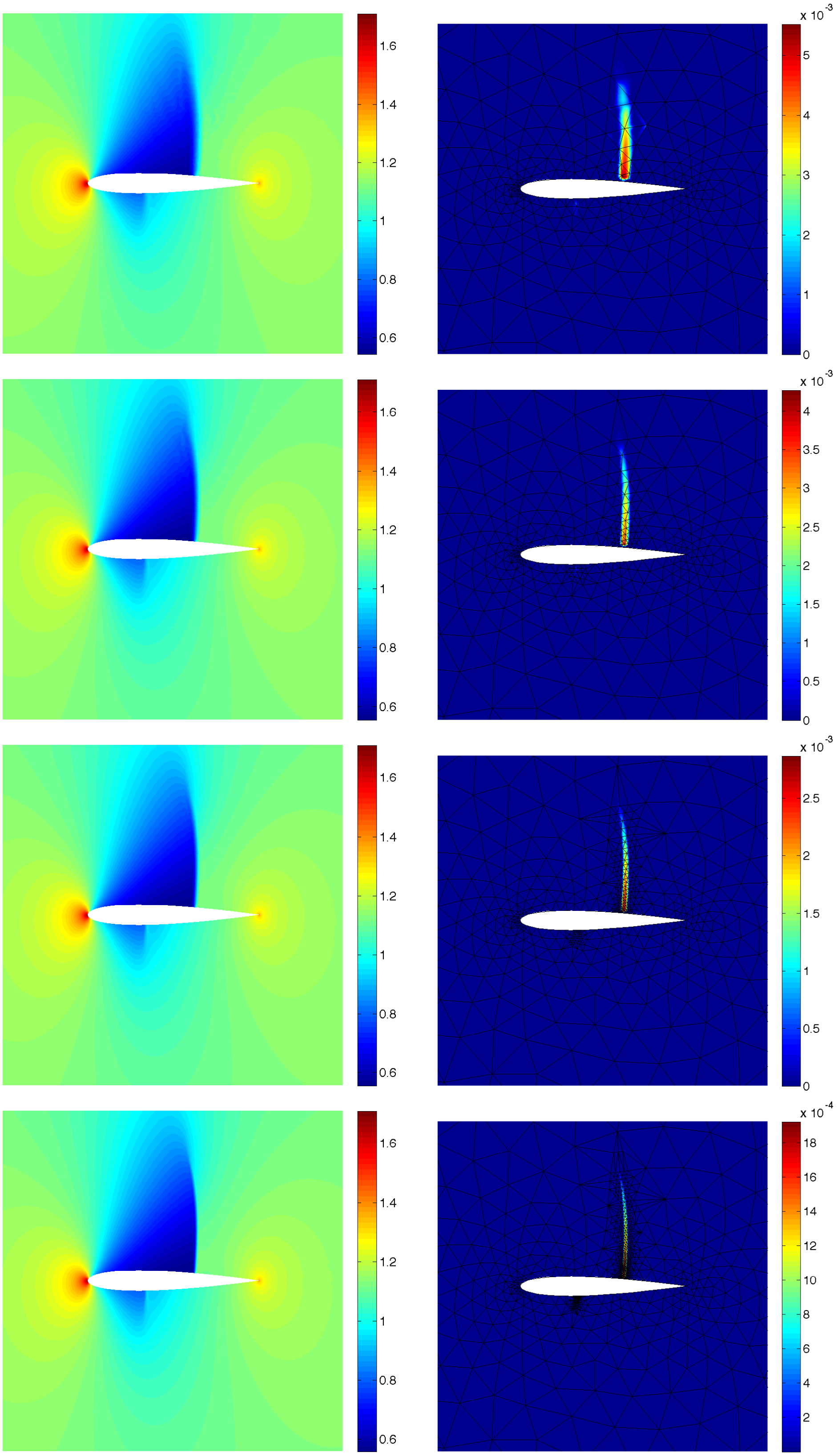


Figure 6. Pressure (left) and artificial viscosity (right) for transonic flow past a NACA 0012 airfoil: (a) the initial mesh (top), (b) the first refinement (second), (c) the second refinement (third), and (d) the third refinement (bottom).

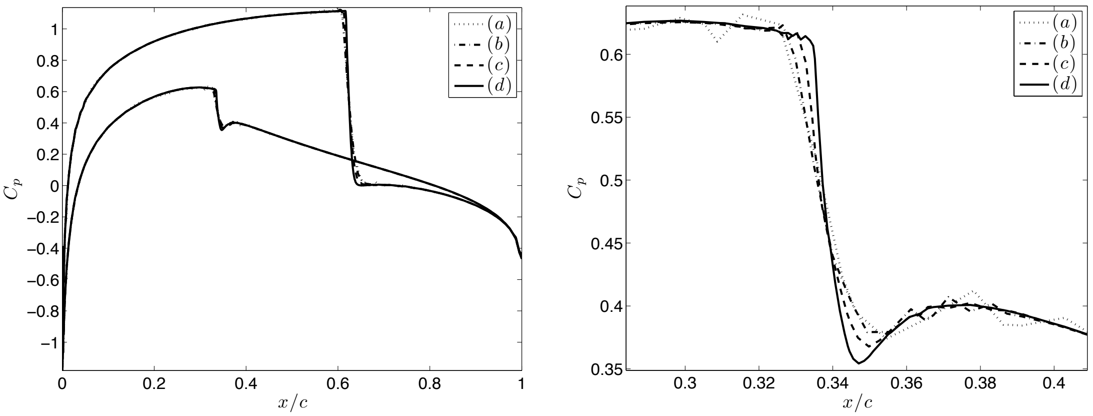


Figure 7. The pressure coefficient distribution over the NACA 0012 surface and its close-up at the shock profile of the lower surface for transonic flow past a NACA 0012 airfoil.

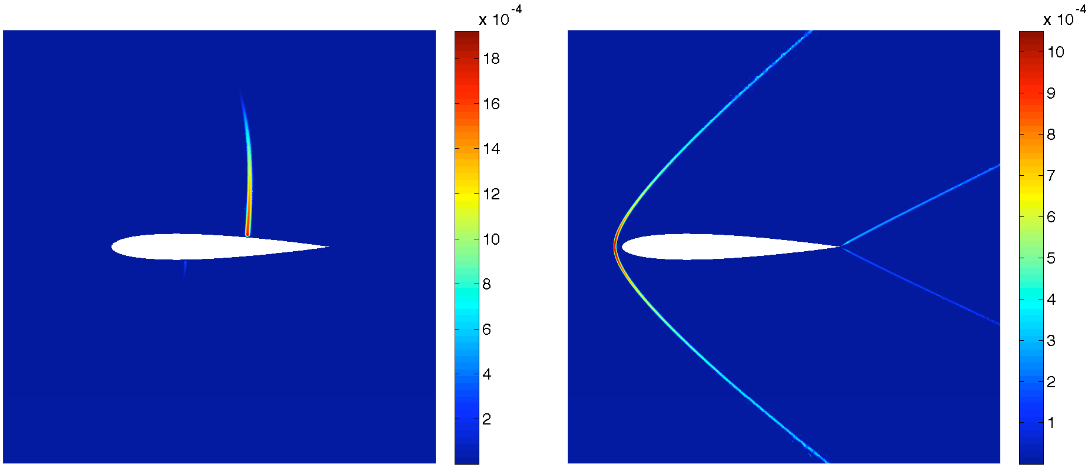


Figure 8. Plot of the artificial viscosity on the third refinement for transonic flow (a) and supersonic flow (b) past a NACA 0012 airfoil.

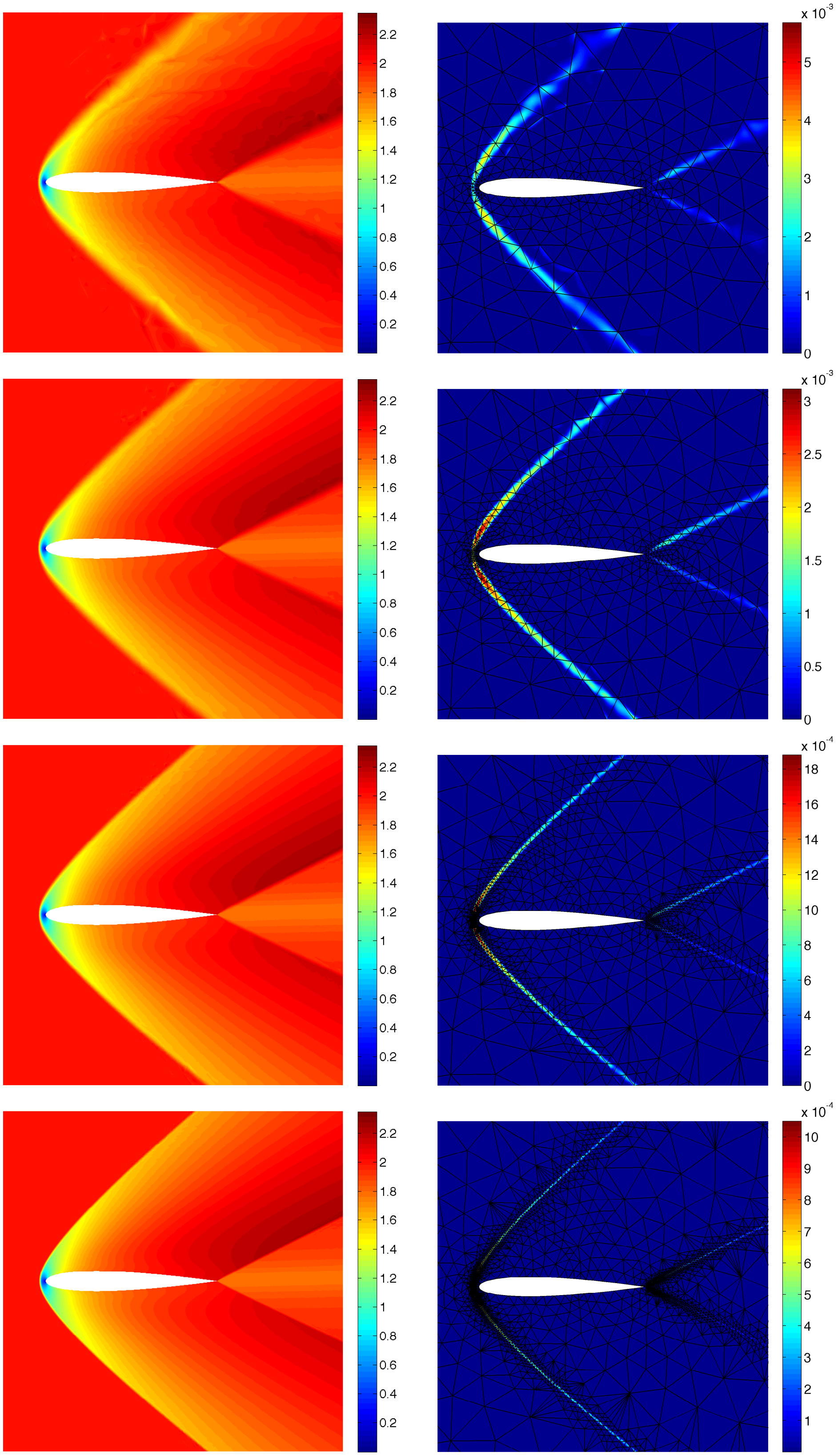


Figure 9. Mach number (left) and artificial viscosity (right) for hypersonic flow past a NACA 0012 airfoil: (a) the initial mesh (top), (b) the first refinement (second), (c) the second refinement (third), and (d) the third refinement (bottom).

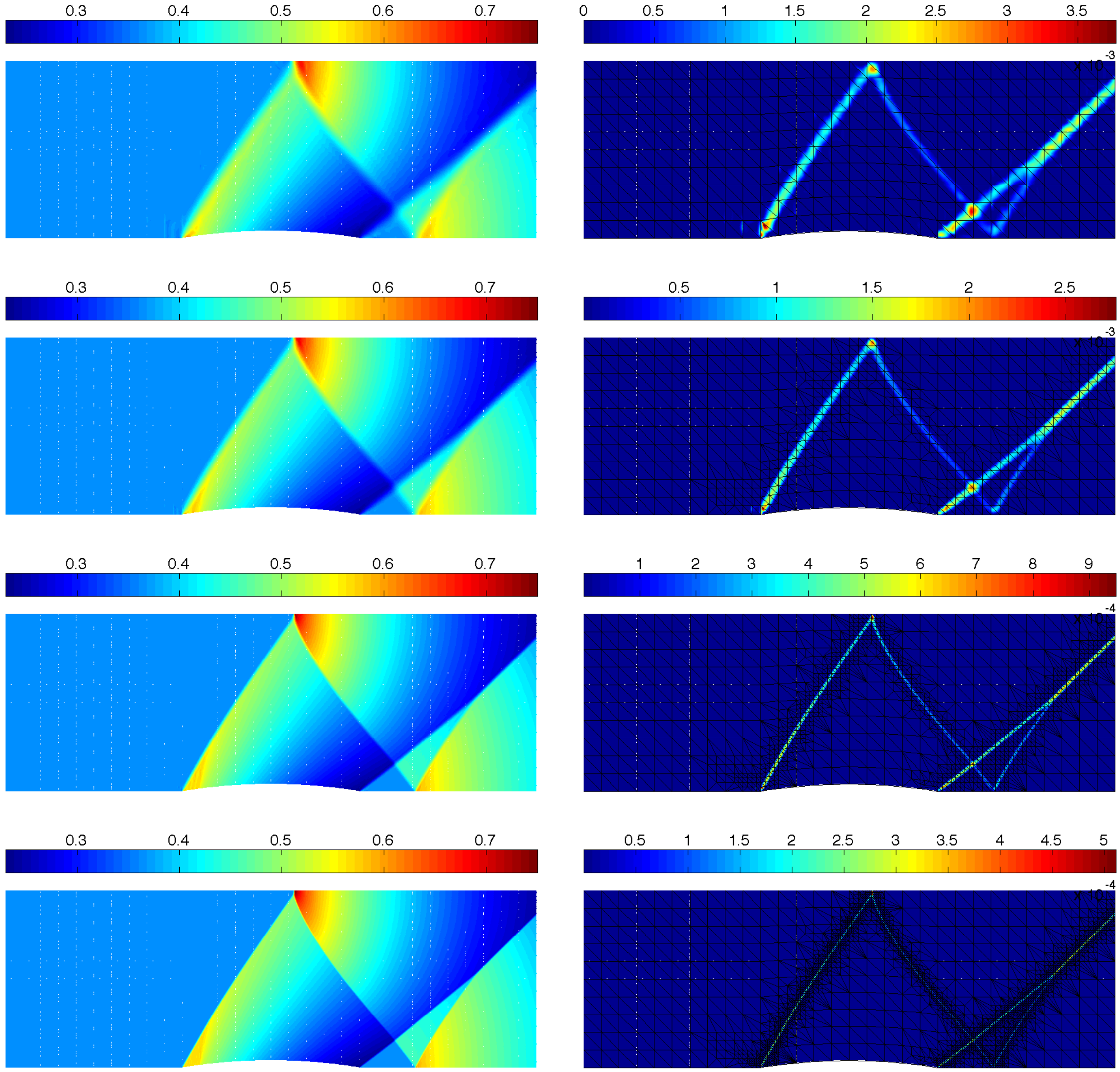


Figure 10. Pressure (left) and artificial viscosity (right) for supersonic flow past a circular bump: (a) the initial mesh (top), (b) the first refinement (second), (c) the second refinement (third), and (d) the third refinement (bottom).

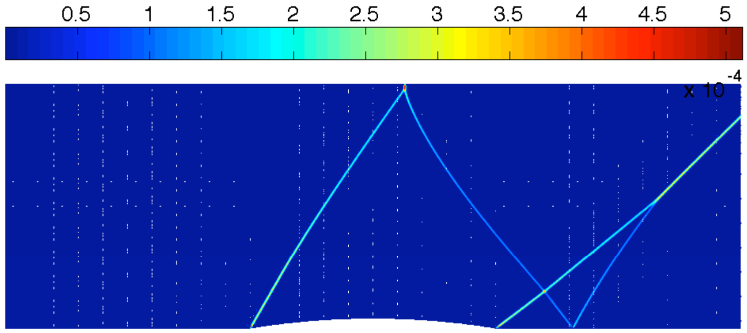


Figure 11. Plot of the artificial viscosity on the third refinement for supersonic flow past a circular bump.

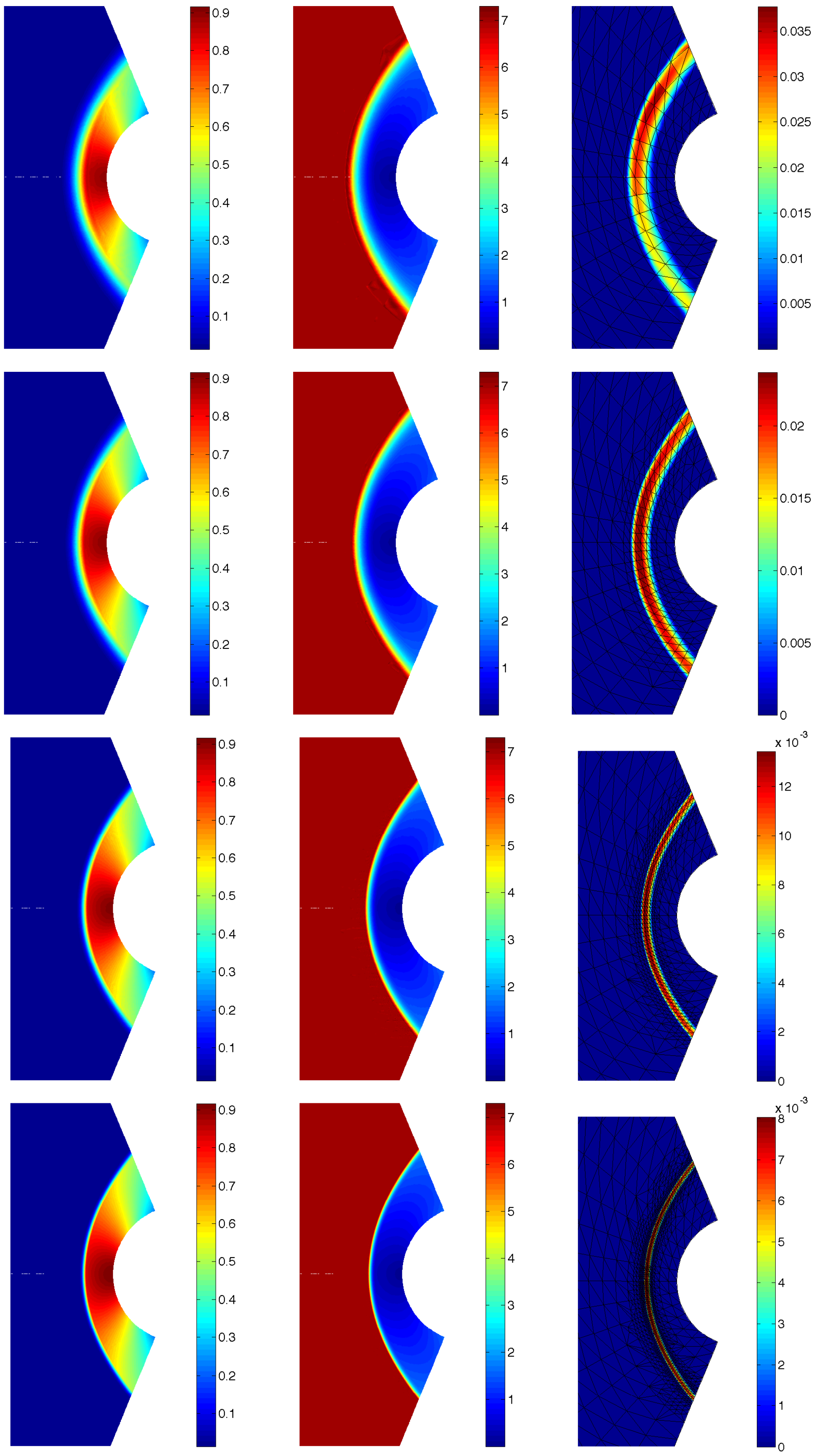


Figure 12. Pressure (left), Mach number (middle), and artificial viscosity (right) for hypersonic flow past a circular cylinder: (a) the initial mesh (top), (b) the first refinement (second), (c) the second refinement (third), and (d) the third refinement (bottom).

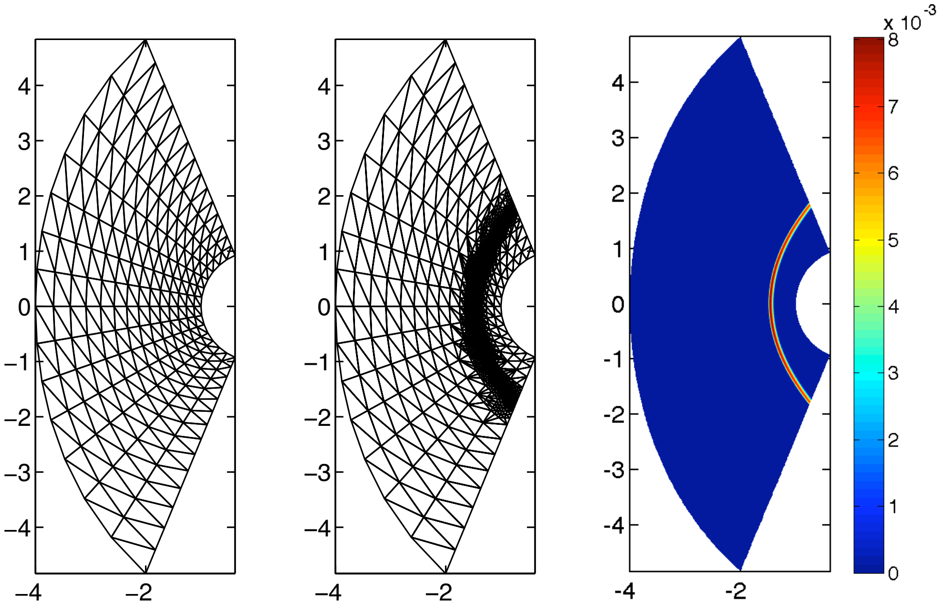


Figure 13. The initial mesh (left), the third refinement mesh (middle), and associated artificial viscosity (right) for hypersonic flow past a circular cylinder.

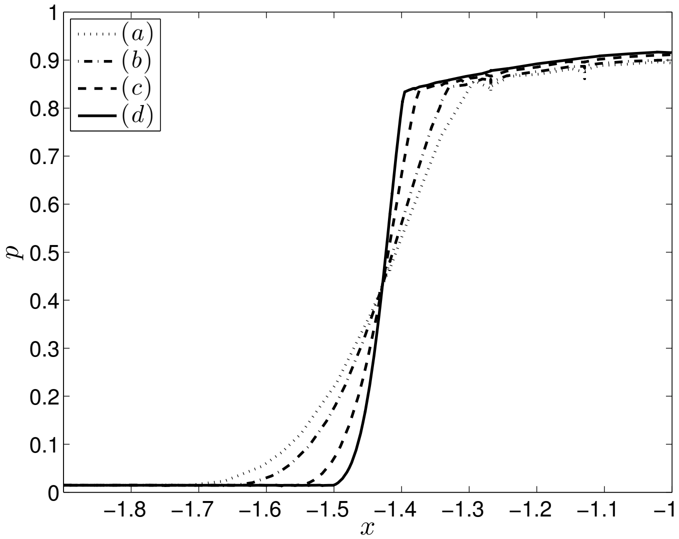


Figure 14. Plot of the pressure along the centerline $y = 0$ for hypersonic flow past a circular cylinder.