Continuous Artificial Viscosity Shock Capturing for Hybrid DG on Adapted Meshes

Yifan Bai* and Krzysztof J. Fidkowski†
University of Michigan, Ann Arbor, MI, 48188

One technique for capturing shocks with high-order methods is through artificial viscosity. Key considerations of this approach are (1) deciding the amount of artificial viscosity to add; (2) maintaining stability and efficiency of the nonlinear solver; and (3) ensuring the accuracy of the resulting solutions, particularly in the presence of strong shocks. To address (1), we compare a switch based on intra-element solution variation with one based on the difference between the solution and its low-order projection. To address point (2), we forego a complete linearization of the artificial viscosity contribution to the residual, in order to keep the residual Jacobian stencil compact. To address (3), we introduce the viscosity in a piecewise-continuous fashion to avoid spurious entropy production. Furthermore, we use output-based error estimation and mesh optimization to minimize the output error. We test the method on aerodynamic flow applications ranging from transonic to supersonic, discretized using standard discontinuous Galerkin (DG) and hybridized DG.

I. Introduction

High-order methods have enabled engineering computations at strict error tolerances for a variety of computational fluid dynamics applications requiring high accuracy. With increasing computational power and improvements in algorithms, these methods are becoming feasible for practical computations. The discontinuous Galerkin (DG) method is a popular high-order finite element method that builds on extensive work in Riemann solvers from the finite volume method. It enables rigorous high-order computations on general unstructured meshes, with a relatively simple implementation. However, DG remains expensive in degrees of freedom, and hence storage and CPU time. To tackle this, hybridized discontinuous Galerkin methods reduce the number of globally-coupled unknowns in an implicit solution through a static condensation procedure [1–4]. Specifically, element-interior unknowns are locally eliminated in favor of face unknowns, through a highly-parallelizable step.

Computational cost is only one issue, however. Another important consideration for high-order methods is robustness. While at first, a discontinuous approximation space appears well-suited for handling discontinuous features, in practice taking advantage of these discontinuities is difficult for general cases. The discontinuous Galerkin method, like other high-order discretizations, suffers from oscillations near discontinuities and other under-resolved features. Much work exists in stabilizing DG, including RKDG [5, 6], WENO [7, 8], and artificial viscosity [9–12]. This work is similar to more recent studies in smooth artificial viscosity [13, 14], but with differences in the smoothness indicator, the means of making the viscosity continuous, an extension to hybridized methods, and coupling with mesh adaptation.

II. Discretization

We consider conservation laws of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{H}}(\mathbf{u}, \nabla \mathbf{u}) \quad = \quad \mathbf{0}, \tag{1}$$

where $\mathbf{u}(\vec{x}, t) \in \mathbb{R}^s$ is the conservative state vector, s is the state rank, and $\vec{\mathbf{H}}(\mathbf{u}, \nabla \mathbf{u}) = \vec{\mathbf{F}}(\mathbf{u}) + \vec{\mathbf{G}}(\mathbf{u}, \nabla \mathbf{u})$ is the total, inviscid and viscous flux.

We consider two discretizations: discontinuous Galerkin (DG) and hybridized discontinuous Galerkin (HDG). In each of these, the computational domain Ω is divided into N_e elements, Ω_e, in a non-overlapping tessellation T_h. Inside element Ω_e, the state is approximated by polynomials of order p, with no continuity constraints on the element

*Graduate Student, Department of Aerospace Engineering, AIAA Student Member.
†Professor, Department of Aerospace Engineering, AIAA Associate Fellow.

boundary. Formally, we write: $\mathbf{u}_h \in \boldsymbol{\mathcal{V}}_h = [\mathcal{V}_h]^s$, where $\mathcal{V}_h = \left\{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^p \ \forall \Omega_e \in T_h\right\}$, and $\mathcal{P}^p$ denotes polynomials of order p on the reference space of element Ω_e.

A. Discontinuous Galerkin (DG)

The DG weak form of Eqn. 1 is obtained by multiplying by test functions in the same approximation space, integrating by parts, and coupling elements via single-valued fluxes that are functions of the states on the two adjacent elements:

$$\begin{aligned}\int_{\Omega_e} \mathbf{w}_h^T \frac{\partial \mathbf{u}_h}{\partial t}\, d\Omega - \int_{\Omega_e} \nabla \mathbf{w}_h^T \cdot \vec{\mathbf{H}}(\mathbf{u}_h, \nabla \mathbf{u}_h)\, d\Omega + \int_{\partial\Omega_e} \mathbf{w}_h^T \widehat{\mathbf{H}} \cdot \vec{n}\, ds \\ - \int_{\partial\Omega_e} \partial_i \mathbf{w}_h^{+T} \mathbf{K}_{ij}^{+} \left(\mathbf{u}_h^{+} - \widehat{\mathbf{u}}_h\right) n_j\, ds \quad = \quad 0 \quad \forall \mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h,\end{aligned} \tag{2}$$

where $(\cdot)^T$ denotes transpose, $\widehat{\mathbf{u}}_h = (\mathbf{u}_h^+ + \mathbf{u}_h^-)/2$, and on the element boundary $\partial\Omega_e$, $(\cdot)^+$, $(\cdot)^-$ denote quantities taken from the element or its neighbor (or boundary condition), respectively.

For the normal flux, $\widehat{\mathbf{H}} \cdot \vec{n}$, we use the Roe-approximate Riemann solver [15] unless specified otherwise and the second form of Bassi and Rebay (BR2) [16] for the viscous flux. Choosing a basis for the test and trial spaces yields the semi-discrete form, $\mathbf{M}\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0}$, where $\mathbf{M}$ is the mass matrix, and $\mathbf{U}$ and $\mathbf{R}$ are the discrete state vector and the discrete residual, respectively.

For steady cases, $\frac{\partial \mathbf{u}}{\partial t} = 0$, we solve the discretized system of non-linear equations $\mathbf{R}(\mathbf{U}) = \mathbf{0}$ with Newton iterations. The linearization of the left-hand-side is the residual Jacobian matrix, $\mathbf{A} \equiv \partial\mathbf{R}/\partial\mathbf{U}$, which is sparse and exhibits a nearest-neighbor block structure.

For unsteady cases, we integrate in time with a three-stage, third-order diagonally implicit Runge-Kutta (DIRK) method [17]. The update from $\mathbf{U}^n$ to $\mathbf{U}^{n+1}$ proceeds through a solution of n_{stage} systems:

$$\mathbf{M}(\mathbf{U}_i - \mathbf{U}^n) + \Delta t a_{ii} \mathbf{R}(\mathbf{U}_i) + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{R}(\mathbf{U}_j) = \mathbf{0}, \quad i = 1, \cdots, n_{\text{stage}} \tag{3}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \sum_{j=1}^{n_{\text{stage}}} b_j \mathbf{R}(\mathbf{U}_j), \tag{4}$$

where

$$a_{ij} = \begin{bmatrix} \alpha & 0 & 0 \\ \frac{1-\alpha}{2} & \alpha & 0 \\ \frac{-(6\alpha^2-16\alpha+1)}{4} & \frac{6\alpha^2-20\alpha+5}{4} & \alpha \end{bmatrix}, \quad b_i = \begin{bmatrix} \frac{-(6\alpha^2-16\alpha+1)}{4} \\ \frac{6\alpha^2-20\alpha+5}{4} \\ \alpha \end{bmatrix},$$

and $\alpha = 0.435866521508459$. Eqn. 3 can be solved with Newton iterations at each stage in a similar way as in the steady case, with a slight modification to the Jacobian matrix that does not affect its sparsity.

B. Hybridized and Embedded Discontinuous Galerkin (HDG)

The starting point for the HDG discretization is the conversion of Eqn. 1 to a system of first-order equations,

$$\vec{\mathbf{q}} - \nabla \mathbf{u} \quad = \quad \vec{\mathbf{0}}, \tag{5}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{H}}(\mathbf{u}, \vec{\mathbf{q}}) \quad = \quad \mathbf{0}, \tag{6}$$

where $\vec{\mathbf{q}}_h \in [\boldsymbol{\mathcal{V}}_h]^{\text{dim}}$ is the state gradient. Multiplying these two equations by test functions $\vec{\mathbf{v}}_h \in [\boldsymbol{\mathcal{V}}_h]^{\text{dim}}, \mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h$ and integrating by parts over an element Ω_e yields the weak form:

$$\int_{\Omega_e} \vec{\mathbf{v}}_h^T \cdot \vec{\mathbf{q}}_h\, d\Omega + \int_{\Omega_e} \nabla \cdot \vec{\mathbf{v}}_h^T \mathbf{u}_h\, d\Omega - \int_{\partial\Omega_e} \vec{\mathbf{v}}_h^T \cdot \vec{n}\, \widehat{\mathbf{u}}_h\, ds \quad = \quad 0 \quad \forall \vec{\mathbf{v}}_h \in [\boldsymbol{\mathcal{V}}_h]^{\text{dim}}, \tag{7}$$

$$\int_{\Omega_e} \mathbf{w}_h^T \frac{\partial \mathbf{u}_h}{\partial t}\, d\Omega - \int_{\Omega_e} \nabla \mathbf{w}_h^T \cdot \vec{\mathbf{H}}\, d\Omega + \int_{\partial\Omega_e} \mathbf{w}_h^T \widehat{\mathbf{H}} \cdot \vec{n}\, ds \quad = \quad 0 \quad \forall \mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h, \tag{8}$$

where $\widehat{\mathbf{u}}_h$ is a new independent unknown: the state on faces of the mesh. The system is closed by a weak enforcement of flux continuity across faces,

$$\int_{\sigma_f} \boldsymbol{\mu}_h^T \left\{ \widehat{\mathbf{H}} \cdot \vec{n}\big|_L + \widehat{\mathbf{H}} \cdot \vec{n}\big|_R \right\} ds \quad = \quad 0 \qquad \forall \boldsymbol{\mu}_h \in \boldsymbol{\mathcal{M}}_h, \tag{9}$$

where $\boldsymbol{\mathcal{M}}_h$ denotes the order-p approximation space on the faces $\sigma_f \in F_h$ of the mesh: $\boldsymbol{\mathcal{M}}_h = [\mathcal{M}_h]^s$, where $\mathcal{M}_h = \left\{ u \in L_2(\sigma_f) : u|_{\sigma_f} \in \mathcal{P}^p \ \forall \sigma_f \in F_h \right\}$, and the subscripts L and R refer to the left and right sides of a face. In HDG, the face approximations are independent and generally discontinuous at nodes and edges in three dimensions. This increases the size of the global system but yields well-defined blocks in the Jacobian matrix that simplify preconditioning.

The fluxes in Eqn. 8 are one-sided, meaning that they depend only on the state and gradient inside the element, and the face state,

$$\widehat{\mathbf{H}} \cdot \vec{n} = \vec{\mathbf{H}}(\hat{\mathbf{u}}_h, \vec{\mathbf{q}}_h) \cdot \vec{n} + \boldsymbol{\tau}(\hat{\mathbf{u}}_h, \mathbf{u}_h, \vec{n}), \qquad \boldsymbol{\tau} = \left| \frac{\partial}{\partial \mathbf{u}} (\widehat{\mathbf{F}} \cdot \vec{n}) \right|_{\mathbf{u}_h^*} (\mathbf{u}_h - \hat{\mathbf{u}}_h) + \eta \vec{\boldsymbol{\delta}}_h \cdot \vec{n}. \tag{10}$$

Note that $\boldsymbol{\tau}$ consists of a convective stabilization computed about the Roe-average state, $\mathbf{u}_h^*$, and a BR2 viscous stabilization [18], where η is set to a value that is at least the number of faces and $\vec{\boldsymbol{\delta}}_h$ is the BR2 auxiliary variable driven by the state jump $\mathbf{u}_h - \hat{\mathbf{u}}_h$.

Choosing bases for the trial/test spaces in Eqns. 7, 8, 9 gives a nonlinear system of ordinary differential equations,

$$\begin{aligned} \mathbf{R}^Q &= \mathbf{0}, \\ \mathbf{M}^U \frac{d\mathbf{U}}{dt} + \mathbf{R}^U &= \mathbf{0}, \\ \mathbf{R}^\Lambda &= \mathbf{0}, \end{aligned} \tag{11}$$

where $\mathbf{M}^U$ is the mass matrix. For a steady case, the ODEs reduce to a non-linear system of equations

$$\mathbf{R}^Q = \mathbf{0}, \quad \mathbf{R}^U = \mathbf{0}, \quad \mathbf{R}^\Lambda = \mathbf{0}, \tag{12}$$

with the Newton update system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{Q} \\ \Delta\mathbf{U} \\ \Delta\boldsymbol{\Lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{R}^Q \\ \mathbf{R}^U \\ \mathbf{R}^\Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{13}$$

where $\mathbf{Q}$, $\mathbf{U}$, and $\boldsymbol{\Lambda}$ are the discrete unknowns in the approximation of $\vec{\mathbf{q}}$, $\mathbf{u}$, and $\hat{\mathbf{u}}$, respectively. $[\mathbf{A}, \mathbf{B}; \mathbf{C}, \mathbf{D}]$ is the primal Jacobian matrix partitioned into element-interior and interface unknown blocks. Note that $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ contain both $\mathbf{Q}$ and $\mathbf{U}$ components. In addition, $\mathbf{A}$ is element-wise block diagonal, and hence easily invertible using element-local operations.

Statically condensing out the element-interior states gives a smaller system for the face degrees of freedom,

$$\underbrace{\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)}_{\boldsymbol{\mathcal{K}}} \Delta\boldsymbol{\Lambda} + \left(\mathbf{R}^\Lambda - \mathbf{C}\mathbf{A}^{-1}\left[\mathbf{R}^Q; \mathbf{R}^U\right]\right) = \mathbf{0}. \tag{14}$$

Solving this set of equations constitutes the global solution of the problem. Following the global solution for $\Delta\boldsymbol{\Lambda}$, an element-local back-solution yields the updates to $\mathbf{Q}$ and $\mathbf{U}$.

If the unsteady term is present, we integrate in time with the same DIRK scheme used for the DG disretization:

$$\begin{aligned} \mathbf{R}^Q(\mathbf{Q}_i, \mathbf{U}_i, \boldsymbol{\Lambda}_i) &= \mathbf{0}, \\ \mathbf{M}^U(\mathbf{U}_i - \mathbf{U}^n) + \Delta t a_{ii} \mathbf{R}^U(\mathbf{Q}_i, \mathbf{U}_i, \boldsymbol{\Lambda}_i) + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{R}^U(\mathbf{Q}_j, \mathbf{U}_j, \boldsymbol{\Lambda}_j) &= \mathbf{0}, \\ \mathbf{R}^\Lambda(\mathbf{Q}_i, \mathbf{U}_i, \boldsymbol{\Lambda}_i) &= \mathbf{0}, \quad i = 1, \cdots, n_{\text{stage}} \end{aligned} \tag{15}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \sum_{j=1}^{n_{\text{stage}}} b_j \mathbf{R}^U(\mathbf{Q}_j, \mathbf{U}_j, \mathbf{\Lambda}_j). \tag{16}$$

For the Newton iterations used to solve Eqn. 15 at each stage, the Jacobian matrix needs to be modified slightly from the steady case, but this does not affect the sparsity of $\mathcal{K}$.

III. Shock Capturing

A. The Euler Equations

Given the states $\mathbf{u} = [\rho, \rho\vec{v}, \rho E]$ for the Euler equations, where ρ is the density, $\vec{v}$ is the velocity and E is the total energy, the pressure can be found as $p = \frac{1}{\gamma-1}\left(\rho E - \frac{1}{2}|\vec{v}|^2\right)$, where γ is the specific gas constant. The inviscid flux vectors are

$$\vec{\mathbf{F}}(\mathbf{u}) = \begin{bmatrix} \rho\vec{v} \\ \rho\vec{v} \otimes \vec{v} + p\mathbf{I} \\ \rho\vec{v}H \end{bmatrix}, \tag{17}$$

where $H = E + \frac{p}{\rho}$ is the total enthalpy and $\mathbf{I}$ is the identity matrix of size dim $\times$ dim.

B. Artificial Viscosity

This section outlines the shock capturing approach using artificial viscosity. The starting point is the general form of an unsteady convection-diffusion partial differential equation, written in index notation,

$$\partial_t u_k + \partial_i F_{ik} = \partial_i (K_{ijkl} \partial_j u_l), \tag{18}$$

where k, l index the state rank, i, j index the spatial dimension, F_{ik} is the convective flux, and K_{ijkl} is the diffusivity tensor. Both F and K generally depend on the state and could depend on the position. For shock capturing, we augment the physical diffusivity with an extra tensor field,

$$K^{\text{stab}}_{ijkl}(\vec{x}) = T_{kl}\epsilon_{ij}(\vec{x}), \tag{19}$$

where $T_{kl} = \frac{\partial \tilde{u}_k}{\partial u_l}$ and $\tilde{u}_k = [\rho, \rho\vec{u}, \rho H]$ is the modified state vector that only appears for the Euler equations to devise a total enthalpy preserving stabilization term [12]. Numerical dissipation is added through both the convective and diffusive fluxes. The Roe flux function [15], is not total enthalpy preserving, and hence we also present test results for the van Leer–Hänel flux function [19] in this paper, which is designed to preserve the total enthalpy.

ϵ_{ij} is an artificial viscosity tensor field,

$$\epsilon_{ij}(\vec{x}) = C\frac{h_{ij}}{\bar{h}}\epsilon_0(\vec{x}), \tag{20}$$

where C is an $O(1)$ constant for adjusting the amount of stabilization, and $h_{ij}/\bar{h}$ is a smoothly varying anisotropy field that is obtained by averaging the mesh-implied element metric from elements to nodes. When needed at an arbitrary point in an element, the anisotropy tensor is interpolated from the linear nodes that make up the element. $\epsilon_0(\vec{x})$ is a smooth scalar that comes from averaging to nodes an element-based artificial viscosity, $\hat{\epsilon}_e$. The construction of $\hat{\epsilon}_e$ starts from a raw elemental artificial viscosity defined as

$$\epsilon_{0e} = \frac{\lambda_{\max}\bar{h}_e}{p}\bar{S}_e, \tag{21}$$

where $\bar{S}_e$ is a smoothness indicator computed from the states, which is either the resolution or the variation indicator in this paper, $\lambda_{\max}$ is the maximum wave speed in element e, p is the approximation order, and $\bar{h}_e$ is a measure of the size of the element

$$\bar{h}_e = [\det(\mathbf{M}_e)]^{1/(2\,\text{dim})}. \tag{22}$$

$\mathbf{M}_e$ is the element metric for element e.

This is followed by a Laplace smoothing of ϵ_e. The Laplace smoothing acts as a diffusion effect on the elemental smoothness indicator. The formulation is similar to a Jacobi smoother:

$$\tilde{\epsilon}_e^{k+1} = (1 - c_s)\epsilon_{0e} + \frac{c_s}{n_e} \sum_{t \in \mathcal{N}(e)} \tilde{\epsilon}_t^k, \quad k = 0, \cdots, n_{\text{smooth}} - 1 \tag{23}$$

$$\epsilon_e = \tilde{\epsilon}_e^{n_{\text{smooth}}} \tag{24}$$

where $\mathcal{N}(e)$ denotes the neighboring elements of element e (those that it shares an edge with), n_e is the number of the neighboring elements, and $0 < c_s \leq 1$ is a user-defined coefficient that adjusts the amount of diffusion. A bigger c_s introduces more diffusion. This is only used in our two-dimensional experiments. c_s is chosen to be 1.0 for the transonic cases and 0.9 for the hypersonic cases. $n_{\text{smooth}} = 10$ was found to be sufficient.

The final step in the calculation of $\hat{\epsilon}_e$ is a filter used for the hypersonic test cases presented in section VI to clip away spurious small values generated by the Laplace smoothing. We adopt the smooth filter definition from Barter and Darmofal [12], except we apply it before making the artificial viscosity continuous:

$$\hat{\epsilon}_e(\epsilon_e) = \begin{cases} 0, & \epsilon_e \leq \theta_L \\ \frac{1}{2}\theta_H \left(\sin\left[\pi \left(\frac{\epsilon_e - \theta_L}{\theta_H - \theta_L} - \frac{1}{2} \right) \right] + 1 \right), & \theta_L < \epsilon_e < \theta_H \\ \theta_H & \epsilon_e \geq \theta_H \end{cases}, \tag{25}$$

where θ_L and θ_H are the maximum and minimum values that $\hat{\epsilon}_e$ varies in between. $\theta_H = \lambda_{\max} \bar{h}_e / p$ and $\theta_L = 0.01\theta_H$ are used in this paper.

For unsteady cases, the artificial viscosity is calculated at each Newton iteration, and the updated viscosity values are used in the calculation of the residuals and the linearizations of the state variables.

The artificial viscosity is treated as constant when calculating the linearizations (no $\frac{\partial K}{\partial u}$ terms included). This freezing of the viscosity results precludes Newton convergence, but in practice it does not significantly increase the number of iterations required to obtain a solution, as many iterations are expended prior to the Newton convergence "bucket." The lack of an exact linearization, however, preserves the compact stencil, which is crucial for efficiency of the solver.

C. Smoothness Indicators

Both the resolution indicator and the variation indicator measure the smoothness of a selected quantity, which is chosen to be the density for our experiments, so that more artificial viscosity is added where discontinuities are presented in the solutions. The one-dimensional results presented are generated with the resolution indicator, whereas for the two-dimensional cases the type of indicator used will be specified.

1. Resolution Indicator

The resolution indicator takes advantage of the fact that for a smooth solution, the coefficients of the Fourier series decay rapidly. It is defined as the difference between a pth order quantity and its least-squares projection onto the space of $(p-1)$th order polynomials:

$$\bar{S}_e = \frac{f^2}{f+1}, \quad f = \frac{S_e}{S_0}, \quad S_e = \frac{\int_{\Omega_e} (u - \tilde{u})^2 \, d\Omega}{\int_{\Omega_e} u^2 \, d\Omega}, \tag{26}$$

where u is the chosen scalar for measuring regularity, $\tilde{u}$ is its $(p-1)$th projection, $S_0 = 10^{-c_0 - c_p p}$ is an order-dependent variation scale, and c_0 and c_p are constants that adjust the amount of stabilization. The bigger both of the constants are, the more stabilization is added. When the resolution indicator is used, the amount of stabilization is controlled by both c_0, c_p in a nonlinear mapping as well as the coefficient C in Eqn. (20) in a linear way. We did not attempt to fine-tune all three at the same time for this paper. $C = 1$ is used and c_0 and c_p are tuned for all one-dimensional examples since we tested for different polynomial orders for each case. For the two-dimensional examples, $c_0 = 0$ and $c_p = 1$ are used and C is tuned whenever more stabilization is needed.

2. Variation Indicator

The variation indicator is based on the variation of the solution in an element:

$$\bar{S}_e = \begin{cases} 0, & S_e < S^* - \Delta S \\ \frac{S_e}{2}\left[1 + \sin\left(\frac{\pi}{2\Delta S}(S_e - S^*)\right)\right], & S^* - \Delta S \leq S_e \leq S^* + \Delta S \\ S_e, & S_e > S^* + \Delta S \end{cases}, \tag{27}$$

$$S_e = \left[\frac{1}{|\Omega_e|}\int_{\Omega_e}\left(\frac{u}{\bar{u}_e} - 1\right)^2 d\Omega\right]^{1/2} \tag{28}$$

where u is the chosen scalar for measuring regularity, and $\bar{u}_e = \frac{1}{|\Omega_e|}\int_{\Omega_e} u d\Omega$, S^* and ΔS are user-defined parameters. The smooth scaling was presented by Persson and Peraire [10] and used in combination with the variation indicator by Ching et al. [14]. It preserves large values and clips down small ones. S^* is chosen to be 0.75 for our transonic cases and 1.25 for the hypersonic cases. ΔS is set to be 0.25 throughout.

IV. Mesh Adaptation

A. Output Error Estimation

The mesh adaptation process used in our in-house code for the experiments done in this paper is driven by the estimation of the output error. Let H denote the coarse space and h the fine space obtained by increasing the polynomial order by 1. An estimate of the error between the coarse and the fine space for our output of interest, J, can be found as the adjoint weighted residual. We use discrete adjoint solutions for this purpose. For DG, the output error estimate reads

$$J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) \approx -\delta\boldsymbol{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H), \tag{29}$$

where $\mathbf{U}_h^H$ is the state injected from the coarse to the fine state, and $\delta\boldsymbol{\Psi}_h$ is obtained by subtracting from the fine space adjoint the coarse space adjoint. For HDG, the output error estimate includes terms associated with the gradient and weak flux continuity equations,

$$J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) \approx -(\delta\boldsymbol{\Psi}_h^Q)^T \mathbf{R}_h^Q - (\delta\boldsymbol{\Psi}_h^U)^T \mathbf{R}_h^U - (\delta\boldsymbol{\Psi}_h^\Lambda)^T \mathbf{R}_h^\Lambda. \tag{30}$$

The localized error contributions on each element can be used as error indicators to drive the mesh adaptation process. For DG, the error indicator on element e is

$$\varepsilon_e \equiv \left|\delta\boldsymbol{\Psi}_{h,e}^T \mathbf{R}_{h,e}(\mathbf{U}_h^H)\right|. \tag{31}$$

The HDG error indicators ε_e^Q and ε_e^U can be found in a similar way but finding ε_e^Λ requires special considerations. A detailed discussion of the solution of the adjoint equations and the treatment of the error localization for HDG is presented in the second author's previous work [20].

We also tested the entropy variables in place of the output adjoint for the Euler equations [21]. They are defined by $\mathbf{v} \equiv \partial U/\partial \mathbf{u}$, where U is the entropy function chosen as

$$U = -\frac{\rho(\ln p - \gamma \ln \rho)}{\gamma - 1} \tag{32}$$

More details and discussions on entropy-adjoint adaptation can be found in the referenced work.

B. Mesh Optimization through Error Sampling and Synthesis (MOESS)

We outline the MOESS algorithm used in this paper in this section. The algorithm and a detailed discussion of its extension to HDG can be found in the second author's previous work [20, 22], which built on earlier work of Yano [23].

To form a continuous optimization problem, the information about the size and strteching of elements in a mesh, which is discrete, is encoded with a continuous Riemann metric $\mathcal{M}(\vec{x}) \in \mathbb{R}^{\dim\times\dim}$. The idea of the MOESS algorithm is to optimize the change in the current, mesh-implied metric $\mathcal{M}_0(\vec{x})$, introduced with a step matrix $\mathcal{S} \in \mathbb{R}^{\dim\times\dim}$ as,

$$\mathcal{M} = \mathcal{M}_0^{\frac{1}{2}} \exp(\mathcal{S}) \mathcal{M}_0^{\frac{1}{2}}, \tag{33}$$

given a target cost C_{target}, so that the estimated output error is minimized.

The step matrix field $\mathcal{S}(\vec{x})$ is represented by values at the mesh vertices, $\mathcal{S}_v$, in the implementation of MOESS. The optimization process in MOESS is achieved in multiple iteration steps. In each step, an error reduction to cost induction ratio is calculated at each vertex, the trace of the step matrices is modified so that the vertices with the largest values of this ratio are refined, and the vertices with the smallest values of the ratio are coarsened, the trace-free part of the step matrices is updated to preserve element shape at fixed area, and finally, the step matrices are manipulated to constrain the total cost. The process then repeats with the updated step matrices.

The rates of change of the total error and the cost with respect to the step matrices, $\frac{\partial \varepsilon}{\partial \mathcal{S}_v}$ and $\frac{\partial C}{\partial \mathcal{S}_v}$, are calculated from element-based models that relate the error indicator ε_e and elemental cost C_e to the step matrix on an element e. For DG, the error convergence model is

$$\varepsilon_e = \varepsilon_{e0} e^{\text{tr}(\mathcal{R}_e \mathcal{S}_e)}, \tag{34}$$

where $\mathcal{R}_e$ is an element-specific error rate tensor determined through an error sampling process. For HDG $\varepsilon_e = \varepsilon_e^U + \varepsilon_e^Q + \varepsilon_e^\Lambda$, and

$$\varepsilon_e^U = \varepsilon_{e0}^U e^{\text{tr}(\mathcal{R}_e^U \mathcal{S}_e)},\ \varepsilon_e^Q = \varepsilon_{e0}^Q e^{\text{tr}(\mathcal{R}_e^Q \mathcal{S}_e)},\ \varepsilon_e^\Lambda = \varepsilon_{e0}^\Lambda e^{\text{tr}(\mathcal{R}_e^\Lambda \mathcal{S}_e)}, \tag{35}$$

where $\mathcal{R}_e^U$, $\mathcal{R}_e^Q$, $\mathcal{R}_e^\Lambda$ are also found through error sampling. The cost model is related to the trace of the step matrix, which indicates the decrease of the area of an element. The local cost on element e is

$$C_e = C_{e0} e^{\frac{1}{2}\text{tr}(\mathcal{S}_e)}, \tag{36}$$

where C_{e0} is the current cost on element e, prior to refinement, measured by the number of degrees of freedom.

The updated Riemann metric field at the end of the optimization is used as the input to the Bi-dimensional Anisotropic Mesh Generator (BAMG) [24] to generate the updated mesh. In practice, several iterations of the mesh optimization and flow/adjoint solution are performed and the convergence of the targeted output is monitored.

V. One-Dimensional Results

A. Steady Linear Advection

We start by showing an example of a steady case of the linear advection equation,

$$a\frac{du}{dx} = f. \tag{37}$$

The source term f is chosen so that the exact solution is

$$u(x) = \frac{\tanh(256(x - 0.4)) + 3}{2}, \tag{38}$$

which contains a sharp variation at $x = 0.4$.

DG solutions are shown in Fig. 1. The artificial viscosity is able to reduce the overshoots in the solutions, even at very high orders, up to $p = 14$ tested.

B. Unsteady Linear Advection

We demonstrate the ability of the artificial viscosity to stabilize a transient solution with an example of the Zalesak "wave basket" [25] traveling with constant speed $a = 1$. The solutions after one period of the wave traveling are shown in Fig. 2. In this case, we test DG and HDG, and both methods preform similarly. The oscillations in the solutions are greatly reduced by the addition of the artificial viscosity.

C. Unsteady Inviscid Burgers

We compare the use of the continuous piece-wise linear artificial viscosity to the piece-wise constant artificial viscosity through an example of the 1D inviscid Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \tag{39}$$

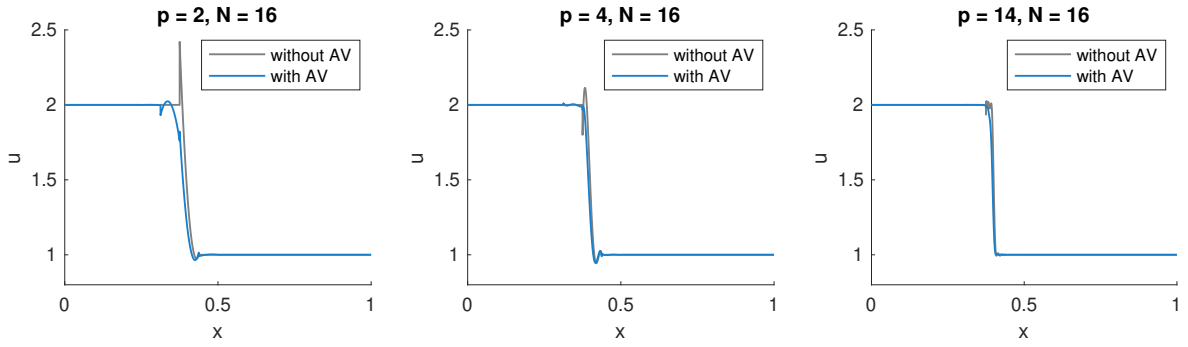
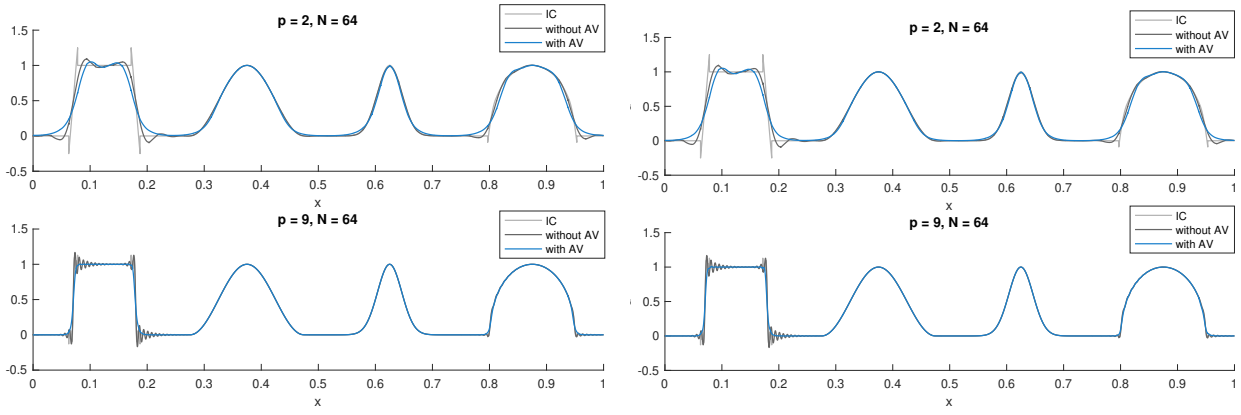


Fig. 1 DG solutions for steady advection: $a = 1.0$, $c_0 = 1.8$, $c_p = 0.3$, **with and without artificial viscosity (AV).**



(a) DG (b) HDG

Fig. 2 DG and HDG solutions for unsteady advection: $t = 1.0$, $c_0 = 1.0$, $c_p = 0.4$.

with an initial sinusoidal wave

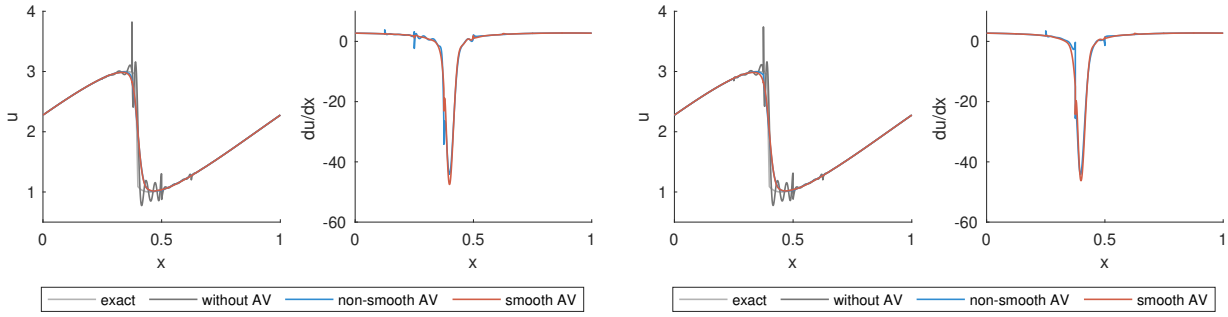
$$u_0(x) = \sin(2\pi(x - 0.5)) + 1, \tag{40}$$

and periodic boundary conditions. A shock wave will start to form at $t = \frac{1}{2\pi}$ as the wave propagates in space.

In Fig. 3, we show the solution values and first derivatives for the Burgers equation example. The derivatives of the solutions with the discontinuous artificial viscosity are more oscillatory for both the DG and HDG cases. The L_1 and L_2 errors in the solutions are shown in Fig. 4. At $t = 0.05$ the solution is smooth, and both DG and HDG are showing error convergence of the corresponding orders of the methods. At $t = 0.2$, a shock has formed, and the orders of error convergence for both DG and HDG drop to 1. The errors for the piece-wise constant artificial viscosity cases at $t = 0.2$ are in general bigger than those of the continuous artificial viscosity cases.

It is worth mentioning that the discontinuous artificial viscosity tends to reduce the stability of the Newton-Raphson iterations when the time step taken is relatively large. Although this instability can be alleviated by smaller time steps or under-relaxation of the Newton-Raphson iterations, it makes the solution with the discontinuous artificial viscosity more costly. This is a more serious issue when more artificial viscosity is added.

Moreover, if the amount of artificial viscosity is tuned down within the reasonable range, so that the capturing of the shock is sharper but more oscillatory, the difference between the errors for the piece-wise constant and the piece-wise linear artificial viscosity further increases.



(a) DG (b) HDG

Fig. 3 DG and HDG solutions to the Burgers equation: $N = 8$, $p = 9$, $t = 0.2.$, $c_0 = 2.0$, $c_p = 0.4$.

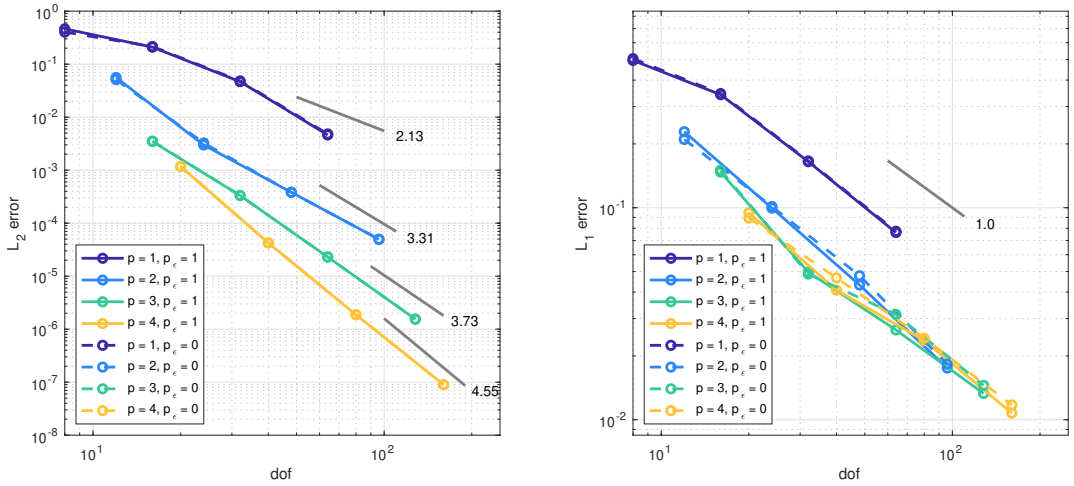
VI. Two-Dimensional Results

The two-dimensional test cases that we present are steady Euler cases with artificial viscosity, solved on unstructured triangular meshes with orders $p_\epsilon = 1$ and $p = 2$. Laplace smoothing is used for all cases unless specified otherwise.

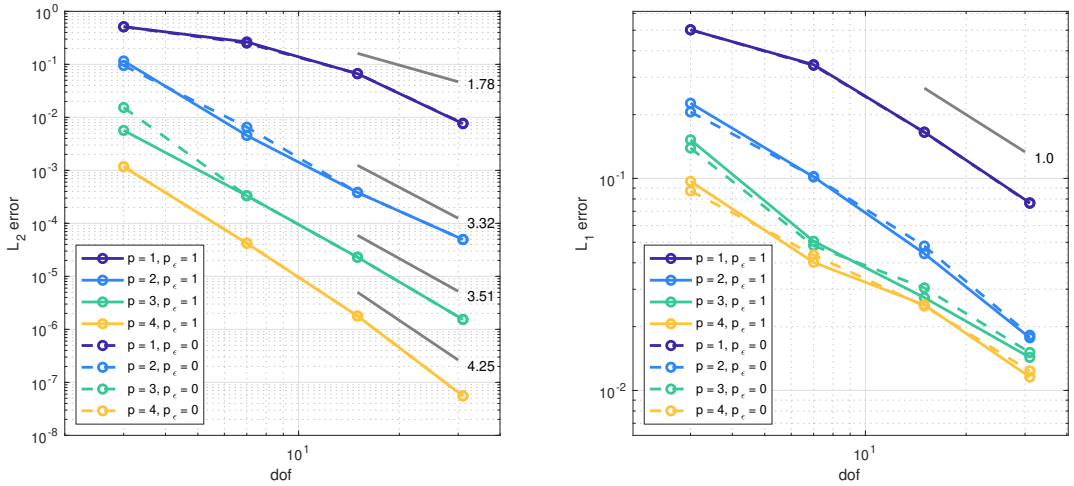
A. Transonic Airfoil

We demonstrate the solver's shock capturing ability and the effect of the Laplace smoothing with a transonic case at freestream Mach number $M = 0.8$ past a NACA 0012 airfoil at and an angle of attack $\alpha = 1.25°$. We run the case with both the resolution indicator ($C = 2$) and the variation indicator ($C = 0.5$) with and without Laplace smoothing for more than 10 adaptive iterations based on the drag adjoint. The drag convergence over the adaptive iterations is shown in Fig. 5. The adapted mesh and solution for the resolution indicator with Laplace smoothing are shown in Fig. 6. One main shock appears above the airfoil and a weak one appears below. Thin, anisotropic elements are placed along the shock interface by the mesh adaptation process.

Since the refined elements of the unstructured mesh are not perfectly aligned with the shock, the non-linear smooth indicators can cause oscillations in the artificial viscosity values along the shock. Therefore, oscillations can form along the shock in the solutions. Laplace smoothing of the artificial viscosity has the potential to alleviation this effect. To compare the solutions, a line probe is taken on the entropy field behind the shock for the solutions on the final adapted meshes. The entropy variations are presented in Fig. 7. Laplace smoothing is able to reduce the oscillations in the entropy for both the resolution and the variation indicators.

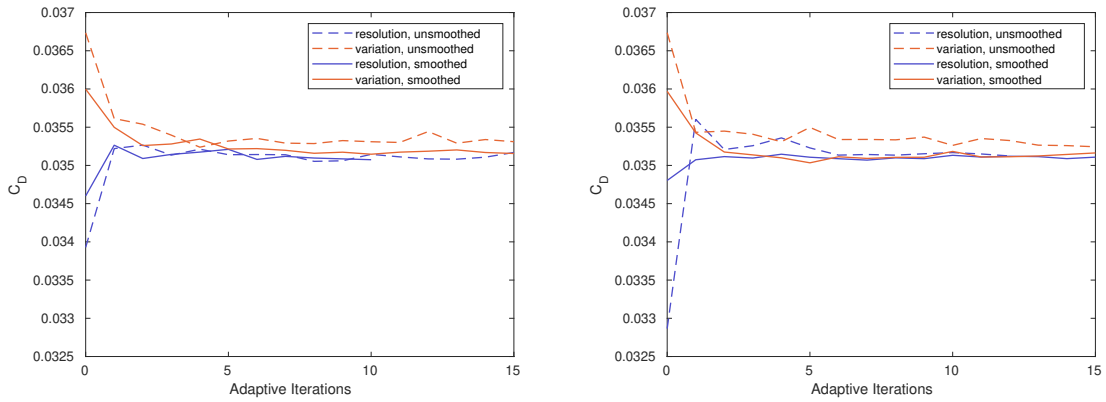


(a) DG, $t = 0.05$ on the left and $t = 0.2$ on the right.



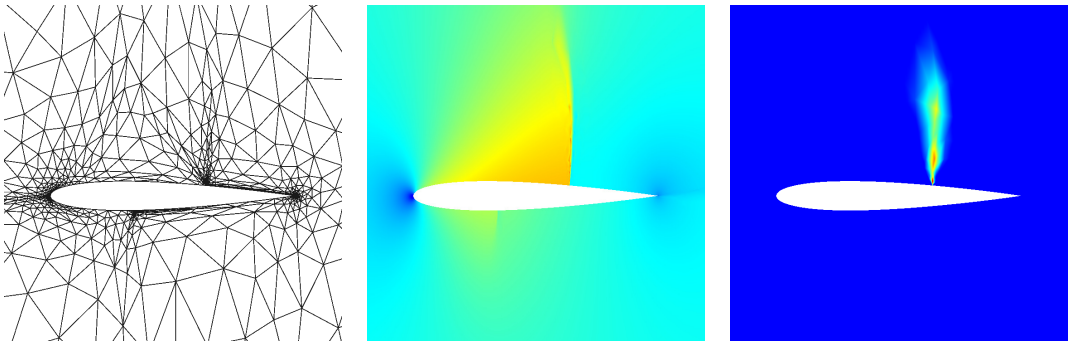
(b) HDG, $t = 0.05$ on the left and $t = 0.2$ on the right.

Fig. 4 Error convergence for DG and HDG solutions to Burgers equation.



(a) DG (b) HDG

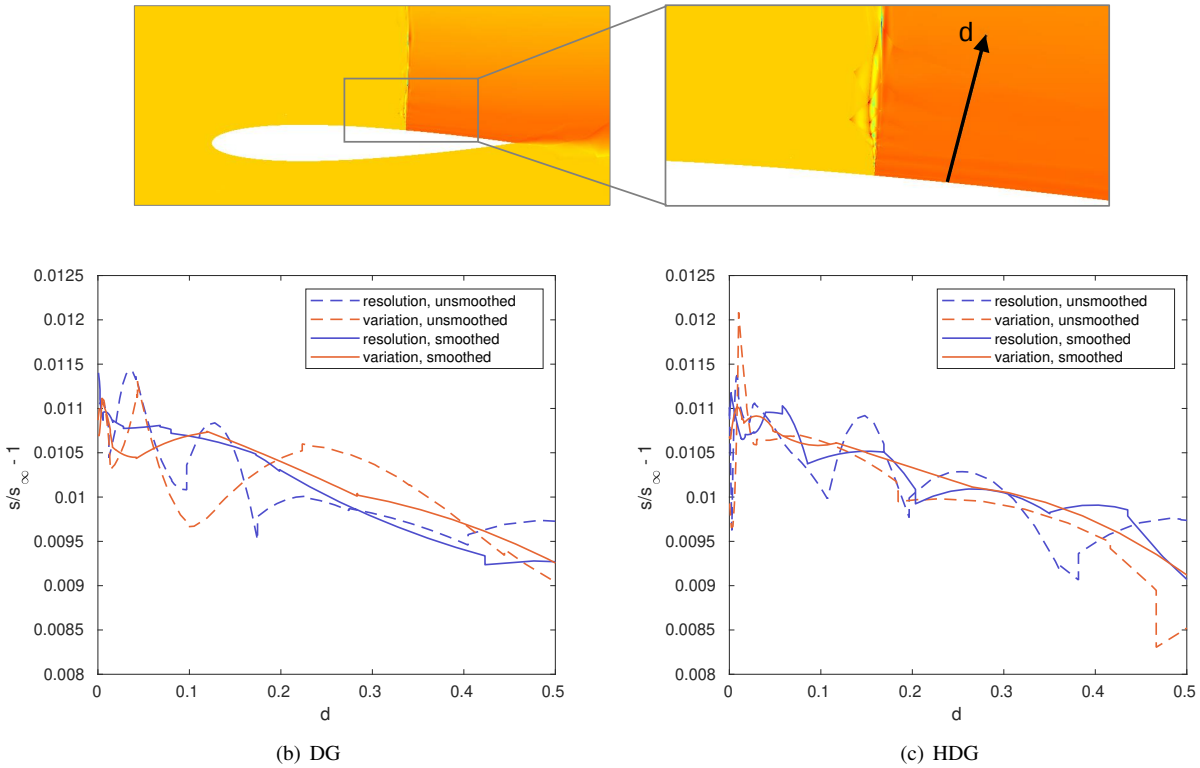
Fig. 5 Drag convergence for the naca case.



(a) Mesh (b) Mach number (c) Stabilization viscosity

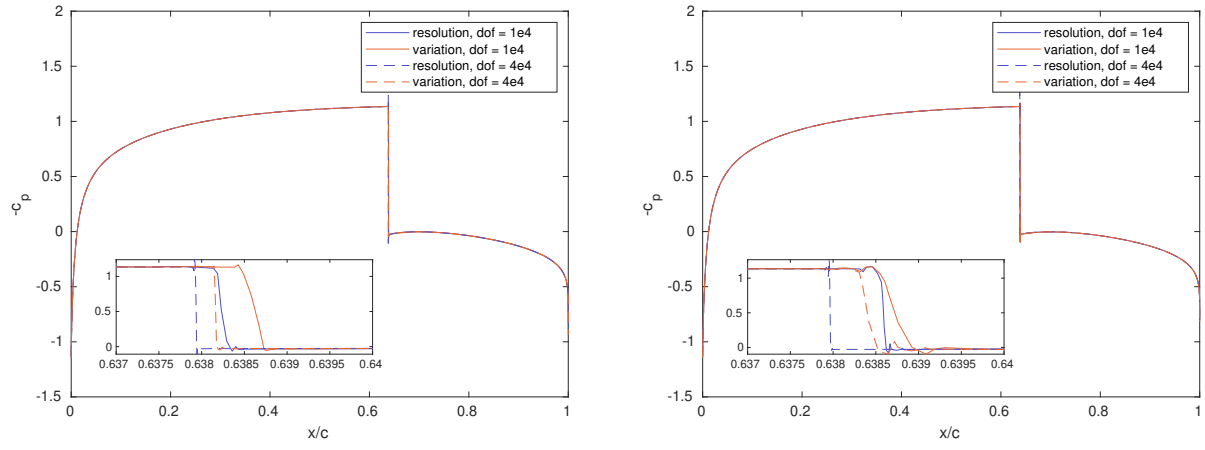
Fig. 6 DG Solution with the resolution indicator and Laplace smoothing. `dof` $= 1 \times 10^4$.

Fig. 8 shows the pressure coefficient plots on the upper surface of the airfoil. The difference between the location of the shock determined with the resolution and the variation indicators is less than 0.1% for both DG and HDG. The resolution of the shock improves as the mesh gets refined.



(b) DG (c) HDG

Fig. 7 Line probe for entropy behind the shock.



(a) DG (b) HDG

Fig. 8 Pressure coefficient distributions on the upper surface of the airfoil.

B. Transonic Gaussian Bump

We investigate the errors in the total enthalpy generated by the numerical scheme with a test case of $M = 0.7$ channel flow with a smooth Gaussian bump geometry. The resolution indicator is used with $C = 4$ for all the results presented for this case. The channel is bounded in the region $[-1.5, 1.5] \times [0, 0.8]$. The bump on the bottom of the channel is defined by

$$y = 0.0625e^{-(x/0.2)^2}.$$

The total enthalpy should be conserved across the shock for an exact inviscid solution. However, the added artificial viscosity as well as the inviscid flux function can serve as sources of total enthalpy. Fig. 9 shows a convergence study of the L_2 error in the total enthalpy, defined as

$$E_H = \sqrt{\frac{\int_\Omega (H/H_\infty - 1)^2 d\Omega}{\int_\Omega d\Omega}},$$

on uniformly refined meshes. The modification of the state vector in the stabilization term, i.e. a non-identity tensor T_{kl} in Eqn. 19, improves the total enthalpy solutions as well as the convergence rate. Although due to the non-linearity of the shock capturing method the convergence rate is still lower than 1. The van Leer–Hänel flux function further reduces the total enthalpy generation. However, the use of the van Leer–Hänel flux function is found to impair the stablilty of the numerical scheme.

In the results in Fig. 10 we compare the mesh adaptation with the drag adjoint and with the entropy variables. The drag and the error in the total enthalpy both converge as the adaptation progresses when using the drag adjoint and eventually hover around the optimal values. However, when the entropy variables are used for adaptation, the drag coefficient and the error in the total enthalpy both eventually begin to increase after one point despite the effort of the adaptation to put elements along the shock. This seems to be relieved when the degrees of freedom increases, especially in the drag coefficient.

Fig. 11 displays the meshes for the two adaptive iterations marked in Fig. 10. The mesh for iteration 9 contains more refined elements along the shock while the mesh in the rest of the flow region is significantly coarsened. The corresponds to the raised error in both drag and total enthalpy. This indicates that the adaptation based on the entropy variables leads to meshes overly focused on the shock, causing insufficient resolution in the rest of the flow domain, and eventually results in inaccurate output values. Adaptation based on the drag output, on the other hand, is able be balance the resolution addition throughout the domain, producing better more accurate output values and better total enthalpy preservation.

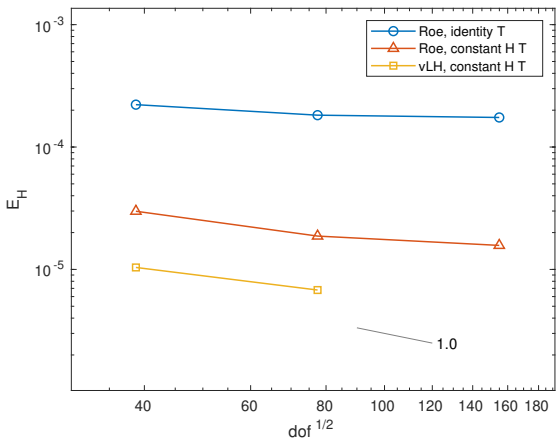


Fig. 9 Error convergence of on uniformly refined meshes for the bump case, using different diffusivity tensor transformations, as defined in Eqn. 19.

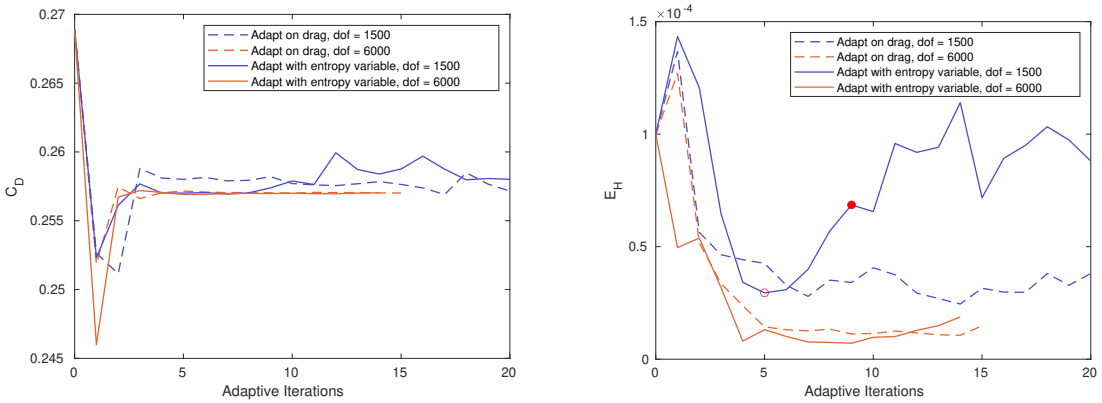
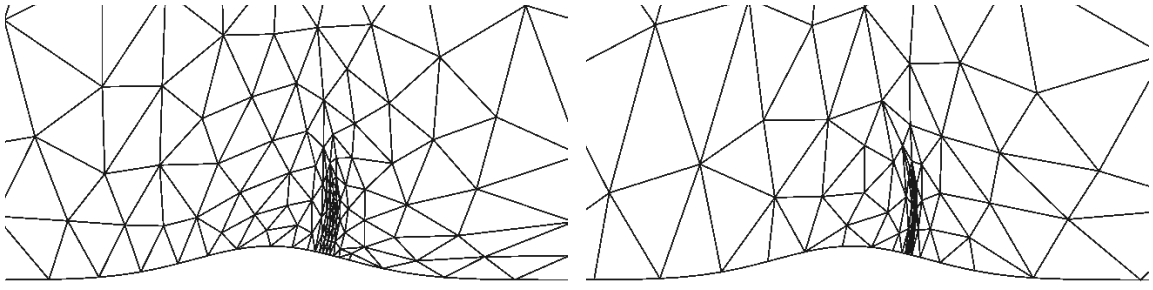


Fig. 10 Drag and total temperature convergence during the adaptation for the bump case.



(a) Iteration 5 ○ (b) Iteration 9 ●

Fig. 11 Meshes adapted using the entropy variables for the bump case.

C. Hypersonic Flow past a Cylinder

The last test case that we present is a $M = 5$ flow past a cylinder. A bow shock is expected to form in front of the cylinder. Part of the flow region behind the shock will be subsonic, bounded within the $M = 1$ contour lines. Supersonic outflow boundary conditions are used at the outflow boundaries. The definition for the error in the total enthalpy varies slightly from the last test case:

$$E_H = \frac{1}{D}\sqrt{\int_\Omega (H/H_\infty - 1)^2 d\Omega},$$

where $D = 2$ is the diameter of the cylinder. Since the main enthalpy error generates within the region behind the shock, the length scale that is of our concern is the cylinder diameter instead of the area of the flow region.

We first compare the results obtained by adapting on the total enthalpy output E_H with those obtained by adapting using the entropy variables. The Gaussian bump test case demonstrated that the entropy variables can intensively target the shock. The adaptive solution were run for more than 15 iteration starting from the initial structured mesh, until the outputs started hovering about certain values. The DG results with the resolution indicator are shown in Fig. 12. The same effect of the entropy variables is observed for the cylinder test case. The total enthalpy adjoint targets the shock and refines in the region behind the shock at the same time, while the entropy adjoint focuses extensively on the shock wave.

The drag and the total enthalpy errors are shown in Fig. 13. The values for drag and E_H are found by averaging the output values of the last 5 adaptive iterations. The "exact" value of the drag was obtained with adaptation on E_H and `dof` = 38400. The linear coefficient in Eqn. (20) has to be tuned to achieve convergence, as listed in Table 1. With a strong shock and the effect of the Laplace smoothing that smears the values to some extent, we resort to bigger stabilization values than the transonic cases. The variation indicator in general is more robust and requires less tuning. Moreover, HDG appears to be more sensitive to the amount of the stabilization, so the coefficient value was lowered on the coarsest mesh. The errors for a series of structured shock-fitted meshes are also plotted for comparison. To generate the shock-fitted meshes, the shock geometry is located as the $M = 3.5$ contour line. Uniform spacing is used in the tangent direction, and tanh spacing centered at the shock locations is used to find the grid point locations along the radial direction. The grid points are connected, and the elements are divided in half to form a triangular mesh. The flow solution is then found on the newly generated mesh and the mesh generation and solution process is repeated for a few times. For the shock-fitted meshes the elements are less skewed. $C = 32, 16$ are used for DG and HDG, respectively, with the resolution indicator, and $C = 6$ is used with the variation indicator.

indicator	DG	HDG
resolution	256	128 (coarsest) / 256 (otherwise)
variation	6	6

Table 1 C values in Eqn. (20) for the hypersonic cylinder case.

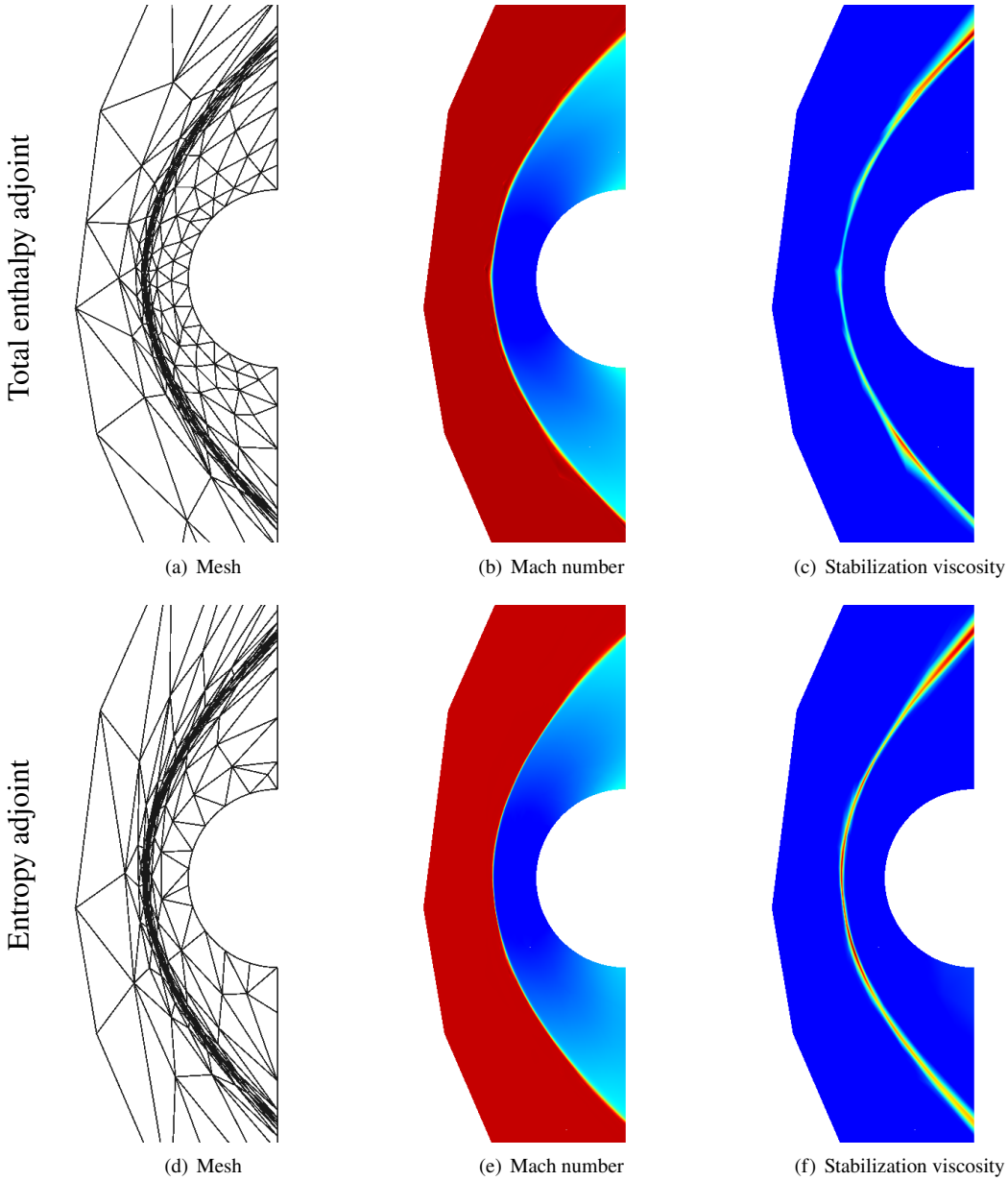
The mesh adaptation is evidently able to reduce the errors in both the total enthalpy and the drag, achieving convergence rates close to 2 for the adaptation on E_H. The adaptation with the entropy variables shows worse convergence of E_H. However, the error in drag is reasonable. The drag doesn't seem to be affected too much by the lack of resolution behind the shock for this case. The resolution and the variation indicator perform similarly for this case.

We also attempted the mesh adaptation with the drag adjoint for this case. However, the adaptation seems to focus on the adjoint features but not as much on the shock, leading to under-resolved shock and spurious solutions. The cause of this requires further investigation, which is ongoing.

VII. Conclusions

In this paper, we demonstrated through one-dimensional and two-dimensional simulations the ability of the continuous artificial viscosity to capture shocks in an adaptive-mesh setting for both the DG and HDG discretizations. We presented results that support generally agreed-upon ideas about the implementation of artificial-viscosity shock capturing:

1) The continuous artificial viscosity works better than piece-wise constant artificial viscosity in reducing oscillations near discontinuities. In the case of hypersonic flow, continuous artificial viscosity can potentially stabilize the cases that piece-wise constant artificial viscosity cannot.



(a) Mesh (b) Mach number (c) Stabilization viscosity

(d) Mesh (e) Mach number (f) Stabilization viscosity

Fig. 12 Adaptive DG results for $M = 5$ inviscid flow past a cylinder. `dof` = 2400

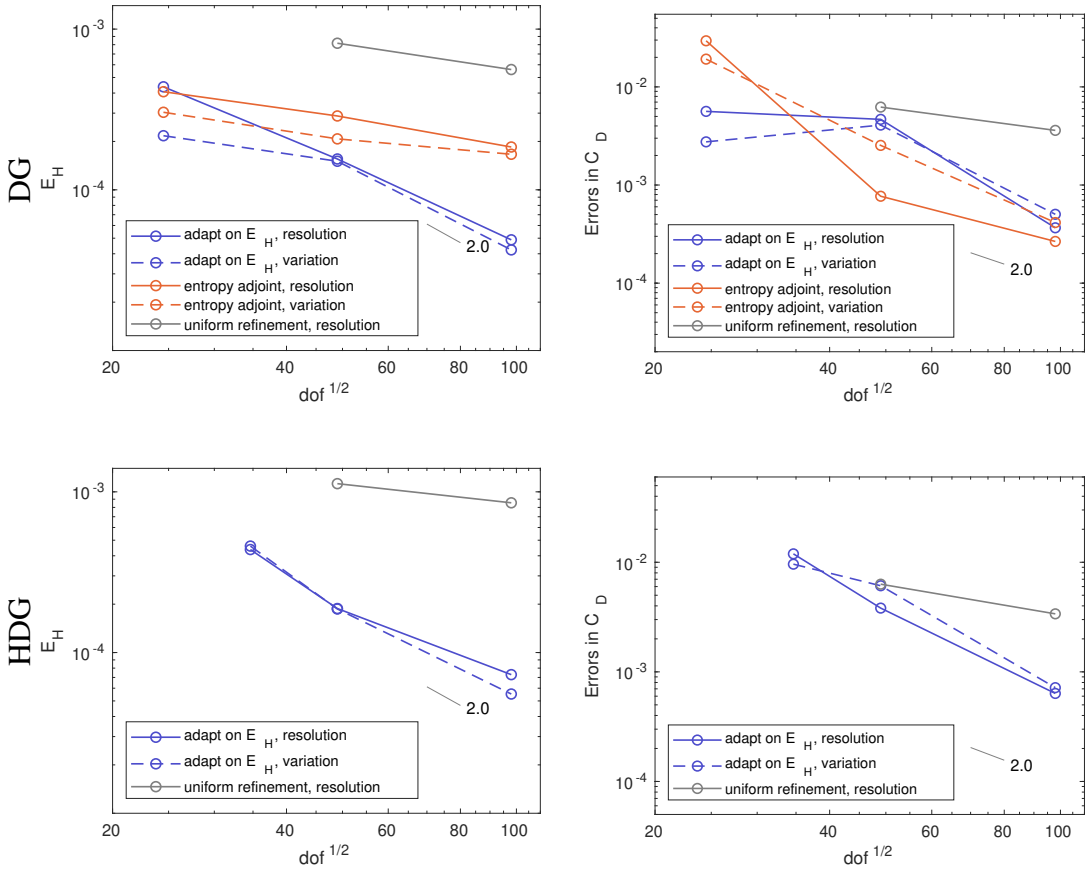


Fig. 13 Convergence plots for the cylinder case.

2) The use of the modified state vector in the added diffusion term is necessary to yield better total enthalpy preservation when dealing with the Euler equations.

We also drew a few conclusions:

1) The variation indicator can result in faster convergence most of the time, since the resolution indicator relies on a low-order projection and is more nonlinear. However, the resolution indicator distinguishes the shock better from the other variations in the flow field, e.g. expansion waves.
2) The mesh adaptation with the entropy adjoint tends to focus too much on the shock, at the cost of reducing the degrees of freedom in other areas that are key to total enthalpy preservation and accurate output computation. Adaptation on the L_2 error of the total enthalpy or an integrated force can lead to better results in many cases.
3) DG and HDG perform similarly in terms of the shock capturing quality for the experiments that we presented, with the exception that HDG is more sensitive to the amount of the artificial viscosity added. HDG is able to reduce the cost of solution without compromising the solution quality.

References

[1] Nguyen, N., Peraire, J., and Cockburn, B., "An Implicit High-Order Hybridizable Discontinuous Galerkin, Method for Linear Convection-Diffusion Equations," *Journal of Computational Physics*, Vol. 228, 2009, pp. 3232–3254.

[2] Peraire, J., Nguyen, N. C., and Cockburn, B., "A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations," AIAA Paper 2010-363, 2010.

[3] Schütz, J., and May, G., "A hybrid mixed method for the compressible Navier–Stokes equations," *Journal of Computational Physics*, Vol. 240, 2013, pp. 58–75.

[4] Woopen, M., Balan, A., May, G., and Schütz, J., "A comparison of hybridized and standard DG methods for target-based hp-adaptive simulation of compressible flow," *Computers & Fluids*, Vol. 98, 2014, pp. 3–16.

[5] Cockburn, B., and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261.

[6] Burbeau, A., Sagaut, P., and Bruneau, C., "A problem-independent limiter for high–order Runge-Kutta discontinuous Galerkin methods," *Journal of Computational Physics*, Vol. 169, 2001, pp. 111–150.

[7] Qiu, J., and Shu, C., "Hermite WENO schemes and their applications as limiters for Runge-Kutta discontinuous Galerkin, method: One-dimensional case," *Journal of Computational Physics*, Vol. 193, No. 1, 2003, pp. 115–135.

[8] Luo, H., Baum, J. D., and Löhner, R., "A hermite WENO-based limiter for discontinuous Galerkin method on Unstructured Grids," AIAA Paper 2007-0510, 2007.

[9] Neumann, J. V., and Richtmyer, R. D., "A method for the numerical calculation of hydrodynamic shocks," *Journal of Applied Physics*, Vol. 21, 1950, pp. 232–237.

[10] Persson, P.-O., and Peraire., J., "Sub-cell shock capturing for discontinuous Galerkin methods," AIAA Paper 2006-112, 2006.

[11] Hartmann, R., "Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, Vol. 51, No. 9–10, 2006, pp. 1131–1156.

[12] Barter, G. E., and Darmofal, D. L., "Shock capturing with PDE-based artificial viscosity for DGFEM: Part I, Formulation," *Journal of Computational Physics*, Vol. 229, 2010, pp. 1810–1827.

[13] Lv, Y., See, Y. C., and Ihme, M., "A general and robust high-order numerical framework for shock-capturing: entropy-bounding, shock detection and artificial viscosity," AIAA Paper 2015-0572, 2015.

[14] Ching, E. J., Lv, Y., Gnoffo, P., Barnhardt, M., and Ihme, M., "Shock capturing for discontinuous Galerkin methods with application to predicting heat transfer in hypersonic flows," *Journal of Computational Physics*, Vol. 376, 2019, pp. 54–75.

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

[16] Bassi, F., and Rebay, S., "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by B. Cockburn, G. Karniadakis, and C.-W. Shu, Springer, Berlin, 2000, pp. 197–208.

[17] Alexander, R., "Diagonally implicit Runge–Kutta methods for stiff ODE's," *SIAM Journal on Numerical Analysis*, Vol. 14, No. 6, 1977, pp. 1006–1021.

[18] Fidkowski, K. J., "A Hybridized Discontinuous Galerkin Method on Mapped Deforming Domains," *Computers and Fluids*, Vol. 139, No. 5, 2016, pp. 80–91. https://doi.org/10.1016/j.compfluid.2016.04.004.

[19] Hänel, D., Schwane, R., and Seider, G., "On the accuracy of upwind schemes for the solution of the Navier-Stokes equations," *8th Computational Fluid Dynamics Conference*, 1987, p. 1105.

[20] Fidkowski, K. J., and Chen, G., "Output-based mesh optimization for hybridized and embedded discontinuous Galerkin methods," *International Journal for Numerical Methods in Engineering*, Vol. 121, No. 5, 2019, pp. 867–887. https://doi.org/10.1002/nme.6248.

[21] Fidkowski, K. J., and Roe, P. L., "An entropy adjoint approach to mesh refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287.

[22] Fidkowski, K., "A local sampling approach to anisotropic metric-based mesh optimization," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0835.

[23] Yano, M., et al., "An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes," Ph.D. thesis, Massachusetts Institute of Technology, 2012.

[24] Borouchaki, H., George, P.-L., Hecht, F., Laug, P., and Saltel, E., "Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: Algorithmes," Ph.D. thesis, INRIA, 1995.

[25] Zalesak, S. T., "The design of Flux-Corrected Transport (FCT) algorithms for structured grids," *Flux-Corrected Transport*, Springer, 2005, pp. 29–78.