AIAA JOURNAL
Vol. 50, No. 12, December 2012

hp-Adaptive Discontinuous Galerkin Methods for the Navier–Stokes Equations

Nicholas K. Burgess* and Dimitri J. Mavriplis†
University of Wyoming, Laramie, Wyoming 82071

DOI: 10.2514/1.J051340

In this work, an *hp*-adaptive discontinuous Galerkin solver for the Navier–Stokes equations is developed. Previous work has focused on efficient solution techniques for discontinuous Galerkin discretizations. Recent work has focused on improving the robustness and efficiency of the discontinuous Galerkin solver for aerodynamic flows. Herein, an *hp*-adaptive approach is proposed, which seeks to place degrees of freedom within the domain in the manner most appropriate for the nature of the solution. Near discontinuities, the algorithm will refine the mesh, whereas, in regions where the solution is smooth, the algorithm will enrich the discretization order. This has two effects. First, degrees of freedom are placed where they are needed, thus addressing the efficiency of the method. Second, attempting to use high-order polynomials to capture solutions that are discontinuous is avoided, addressing the robustness of the method. The adaptation procedure is driven via a discrete adjoint-based goal-oriented error-estimation technique. The method is evaluated using two test cases, both of which are steady-state flows. The first test case is a laminar viscous flow, and the second one is an inviscid transonic flow. The second test case is employed to compare a few shock-capturing approaches, including artificial diffusion. Furthermore, comparisons are made between *hp*-adaptation and uniform *p*-enrichment.

Nomenclature

c_p	=	specific heat at constant pressure
E_t	=	total energy
h_x, h_y	=	mesh size in Cartesian directions
$\mathcal{L}_h$	=	functional evaluated with discrete solution
M	=	number of modes
P	=	pressure
Pr	=	Prandtl number
p	=	polynomial order
$\mathbf{R}_h$	=	discrete flow residual
S_{ij}	=	strain rate tensor components
s_k	=	element resolution indicator value
T	=	temperature
$\mathcal{T}_h$	=	mesh
u, v	=	Cartesian velocity components
$\mathbf{u}$	=	flow variables
$\mathbf{u}_h$	=	discrete flow variables
$\hat{\mathbf{u}}_h$	=	flow variable modal coefficients
$\mathbf{w}_h$	=	test function
Γ_b	=	boundary mesh edge
Γ_i	=	interior mesh edge
γ	=	ratio of specific heats
ε_c	=	computable error
ε_k	=	element-wise artificial diffusion coefficient
Λ_h	=	adjoint variable
μ	=	molecular viscosity
ρ	=	density
τ_{ij}	=	shear stress tensor components
ϕ_j	=	basis function
Ω_k	=	element domain

Presented as Paper 2011-490 at the 49th AIAA Aerospace Sciences Meeting, Orlando, FL, January 4–7, 2011; received 30 April 2011; revision received 23 February 2012; accepted for publication 7 March 2012.

*Ph.D. Candidate, Department of Mechanical Engineering.
†Professor, Department of Mechanical Engineering.

I. Introduction

HIGH-ORDER discontinuous Galerkin (DG) methods are receiving a great deal of attention for the computation of convection-dominated problems [1–8]. Recent work [9–14] has focused on the extension of these methods to convection–diffusion equations (e.g., Navier–Stokes equations). Additionally, a posteriori error estimation for functional outputs is becoming a mature technique for estimating the contribution of discretization error to simulation outputs [15–19]. These a posteriori error estimates are based on solutions to the so-called adjoint (dual) problem. Such error estimates provide a method to guide adaptive refinement techniques such that additional degrees of freedom are optimally placed for a simulation output. Recently, these techniques have been used to perform adaptive mesh refinement (i.e., *h*-refinement only) in the context of DG discretizations of the Reynolds-averaged Navier–Stokes equations [20–22]. Wang and Mavriplis [18] used these techniques for the *hp*-adaptation of high-speed shocked flows using a DG discretization of the compressible Euler equations.

In this work, we propose to use these techniques to adaptively enrich the discretization order and refine the mesh for DG discretizations of the Navier–Stokes equations on mixed-element meshes (i.e., meshes containing triangles and quadrilaterals). A discrete adjoint formulation is used to obtain the error estimates in the functional of interest. The formulation is based on a discrete adjoint approach using a fully dual (adjoint) consistent discretization. Oliver [21], Oliver and Darmofal [23], and Hartmann and Houston [24] have shown numerically that using dual-inconsistent discretizations can lead to suboptimal convergence of the primal solution, whereas using dual consistent or asymptotically dual consistent discretizations leads to optimal convergence ($\sim h^{p+1}$) of the primal problem and super convergence ($\sim h^{2p}$) of the output functional. Herein, we employ a modified symmetric interior penalty (SIP) method for the Navier–Stokes equations that is fully dual consistent [25].

DG methods are capable of generating high-order accurate solutions to the Euler and Navier–Stokes equations. However, this is only attained if the solution is smooth. Unfortunately, for aerodynamic applications, solutions are rarely smooth. Nonsmooth solutions can result from the expected discontinuities such as shock waves and contact discontinuities as well as from additional sources, which are not covered as thoroughly in the literature. For example, if the leading edge of an airfoil has been discretized with too few cells, oscillations can develop due to underresolution of smooth phenomena and cause the solver to fail as shown in [26].

A unique property of DG discretizations is that the order of accuracy and the number of degrees of freedom (DoFs) are coupled. This is in contrast to the traditional finite-volume or finite-difference techniques, which instead rely on extended stencils to increase the discretization order. This property causes additional difficulty for the development of limiters for DG methods compared to traditional CFD methods and poses a rather serious drawback for computing discontinuous solutions with DG. Although the addition of artificial diffusion can be used to treat nonsmooth solutions, no theoretical minimum bound on the required amount of artificial diffusion has been given, thus complicating the use of this approach. Furthermore, initial investigations into using an artificial diffusion method for shock capturing indicate that the functional accuracy is not necessarily improved when the discretization order is uniformly increased. However, artificial diffusion does have a role to play in combination with adaptation.

Although for nonadaptive techniques the coupling between discretization order and the number of degrees of freedom poses a rather serious problem for limiting the solution as shown numerically in [26], the coupling is actually an advantage in the context of an adaptive method. This property of DG discretizations allows for a flexible procedure by which resolution can be added to a problem. In this work, we examine the use of hp-adaptation for two purposes, the first of which is to place degrees of freedom within the domain as optimally as possible, the second of which is to improve solver robustness by not attempting to use high-order polynomial approximations in regions of the mesh where it is inappropriate (e.g., near discontinuities). Moreover, the constant presence of discontinuous solutions in practical problems of interest motivates one to examine adaptation techniques that take this into account as robustly as possible.

Although a large amount of work has been conducted to create efficient solvers for discontinuous Galerkin discretizations of the Euler and Navier–Stokes equations [27–30], much less attention has been paid to solver robustness. We view the proposed approach as both a method by which to place degrees of freedom optimally within the domain and as a method to improve the robustness of the DG solver via careful attention to how the degrees of freedom are added to the mesh. In regions where the solution is smooth, p-enrichment is used, whereas, in regions where the solution is not smooth, h-refinement is employed. Additionally, this work also discusses the accuracy and robustness of artificial diffusion, hp-adaptation, and a combined approach to shock capturing.

II. Governing Equations

The conservative form of the compressible Navier–Stokes equations describing the conservation of mass, momentum, and total energy including artificial diffusion fluxes in two dimensions are given as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot [\vec{F}_c(\mathbf{u}) - \vec{F}_v(\mathbf{u}, \nabla \mathbf{u}) - \vec{F}_{\text{ad}}(\varepsilon, \mathbf{u}, \nabla \mathbf{u})] = 0 \tag{1}$$

subject to the appropriate boundary and initial conditions within a domain Ω. In two dimensions, the flow variables and Cartesian components of the flux vectors are written as

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{Bmatrix}, \quad \mathbf{F}_{\mathbf{c}}^x = \begin{Bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ u(E_t + P) \end{Bmatrix}, \quad \mathbf{F}_{\mathbf{c}}^y = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ v(E_t + P) \end{Bmatrix},$$

$$\mathbf{F}_{\mathbf{v}}^x = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + c_p\left(\frac{\mu}{P_r}\right)\frac{\partial T}{\partial x} \end{Bmatrix}, \quad \mathbf{F}_{\mathbf{v}}^y = \begin{Bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} + c_p\left(\frac{\mu}{P_r}\right)\frac{\partial T}{\partial y} \end{Bmatrix} \tag{2}$$

where ρ is fluid density, $\vec{v} = (u, v)$ are the Cartesian velocity components, P is the fluid pressure, E_t is the total energy, T is the fluid temperature, and τ_{ij} is the viscous stress tensor. Assuming a Newtonian fluid, the viscous stress tensor takes the following form (with $x_i = x, y \quad ; i = 1, 2$):

$$\tau_{ij} = 2\mu S_{ij} \qquad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}$$
$$\text{for } i = 1, 2, \qquad j = 1, 2 \tag{3}$$

where μ is the fluid viscosity obtained via Sutherland's law. Further, using the ideal gas assumption, the pressure is given as

$$P = (\gamma - 1)\left[E_t - \frac{1}{2}\rho(u^2 + v^2)\right] \tag{4}$$

where $\gamma = 1.4$ is the ratio of specific heats. Additionally, the artificial diffusion fluxes are given by

$$\mathbf{F}_{\mathbf{ad}x} = \begin{Bmatrix} \varepsilon_k(|u| + a)\frac{h_x}{p+1}\frac{\partial \rho}{\partial x} \\ \varepsilon_k(|u| + a)\frac{h_x}{p+1}\frac{\partial \rho u}{\partial x} \\ \varepsilon_k(|u| + a)\frac{h_x}{p+1}\frac{\partial \rho v}{\partial x} \\ \varepsilon_k(|u| + a)\frac{h_x}{p+1}\frac{\partial \rho H}{\partial x} \end{Bmatrix}, \quad \mathbf{F}_{\mathbf{ad}y} = \begin{Bmatrix} \varepsilon_k(|v| + a)\frac{h_y}{p+1}\frac{\partial \rho}{\partial y} \\ \varepsilon_k(|v| + a)\frac{h_y}{p+1}\frac{\partial \rho u}{\partial y} \\ \varepsilon_k(|v| + a)\frac{h_y}{p+1}\frac{\partial \rho v}{\partial y} \\ \varepsilon_k(|v| + a)\frac{h_y}{p+1}\frac{\partial \rho H}{\partial y} \end{Bmatrix} \tag{5}$$

where a is the sound speed, and p is the order of accuracy. These fluxes are a modified version of those presented in [31]. The modifications of the artificial diffusion fluxes consist of adding anisotropic mesh metrics (h_x, h_y) and the convective eigenvalue scaling $(|u| + a, |v| + a)$ terms. The addition of these two terms resulted in a significant improvement of the shock capturing method robustness. The value of ε in each cell k is determined as

$$\varepsilon_k = \begin{cases} 0 & s_k \le s_0 - \kappa \\ \frac{1}{2}\varepsilon_0\left[1 + \sin\left(\frac{1}{2}\frac{\pi(s_k - s_0)}{\kappa}\right)\right] & s_0 - \kappa \le s_k \le s_0 + \kappa \\ \varepsilon_0 & s_k \ge s_0 + \kappa \end{cases}$$
$$s_0 = -4\log_{10}(c_{s_0} p) \tag{6}$$

where c_{s_0} and κ are user-defined coefficients, and s_k for the cell k is given by the resolution indicator:

$$s_k = \log_{10}\left(\frac{\int_{\Omega_k}(\rho_h - \tilde{\rho}_h)^2 \, \mathrm{d}\Omega_k}{\int_{\Omega_k} \rho_h^2 \, \mathrm{d}\Omega_k}\right)$$
$$\rho_h = \sum_{j=1}^{M_p} \hat{\rho}_j \phi_j$$
$$\tilde{\rho}_h = \sum_{j=1}^{M_{p-1}} \hat{\rho}_j \phi_j \tag{7}$$

where M_p is the number of modes in a discretization of order p, and M_{p-1} is the number of modes in a discretization of order $p - 1$. If the modifications to the artificial diffusion fluxes are not employed, the setting of artificial viscosity coefficients is overly difficult.

III. Spatial Discretization

The DG discretization is carried out by considering $\mathbf{u} \in \mathcal{V}$. Let the computational domain Ω be partitioned into a set of nonoverlapping simplex elements denoted $\mathcal{T}_{h,p}$ (where h and p are the 'size' and 'order' of the elements of $\mathcal{T}_{h,p}$). Subsequently, we will omit the subscript p in the notation for brevity and just let $\mathcal{T}_h$ represent the discretized domain. Let k denote an element $k \in \mathcal{T}_h$ on which we define a discrete function space $\mathcal{V}_h^p$, which is chosen such that $\mathcal{V}_h^p \subset \mathcal{V}$. Let the collection of interior faces of $\mathcal{T}_h$ be denoted I_h with f denoting a face $\in I_h$; also, let the boundary $\partial\Omega$ be discretized into a set of nonoverlapping faces $\mathcal{B}_h$ with a single boundary face denoted

$b \in \mathcal{B}_h$. Let $\mathbf{u}_h$ represent the discrete solution to $\mathbf{u}$. The discretization is given by finding $\mathbf{u}_h \in \mathcal{V}_h^p$, such that

$$\sum_{k\in\mathcal{T}_h}\int_{\Omega_k}\mathbf{w}_h^T\frac{\partial\mathbf{u}_h}{\partial t}+\mathbf{w}_h^T\nabla\cdot[\vec{F}_c(\mathbf{u}_h)-\vec{F}_v(\mathbf{u}_h,\nabla\mathbf{u}_h) - \vec{F}_{\text{ad}}(\varepsilon,\mathbf{u}_h,\nabla\mathbf{u}_h)]\,\mathrm{d}\Omega_k=0 \quad \forall\ \mathbf{w}_h\in\mathcal{V}_{h,p} \tag{8}$$

This can be written as

$$\sum_{k\in\mathcal{T}_h}\int_{\Omega_k}\mathbf{w}_h^T\frac{\partial\mathbf{u}_h}{\partial t}\,\mathrm{d}\Omega_k+\mathbf{R}_h(\mathbf{w}_h,\mathbf{u}_h,\nabla\mathbf{u}_h)=0, \quad \forall\ \mathbf{w}_h\in\mathcal{V}_{h,p} \tag{9}$$

where the $\mathbf{R}_h(\mathbf{w}_h,\mathbf{u}_h,\nabla\mathbf{u}_h)$ is the discrete spatial residual. The discrete spatial residual is just the collection of the spatial derivative terms in Eq. (8). The spatial residual is integrated by parts resulting in the following weak form:

$$\begin{aligned}\mathbf{R}_h(\mathbf{w}_h,\mathbf{u}_h,\nabla\mathbf{u}_h)=&-\sum_{k\in\mathcal{T}_h}\int_{\Omega_k}\nabla\mathbf{w}_h^T\cdot(\vec{F}_c(\mathbf{u}_h)-\vec{F}_v(\mathbf{u}_h,\nabla\mathbf{u}_h)\\&-\vec{F}_{\text{ad}}(\varepsilon,\mathbf{u}_h,\nabla\mathbf{u}_h))\,\mathrm{d}\Omega_k+\sum_{i\in I_h}\int_{\Gamma^i}\mathcal{H}_c(\mathbf{u}_h^+,\mathbf{u}_h^-,\vec{n})\\&-\mathcal{H}_v(\mathbf{u}_h,\mathbf{u}_h^-,\mathbf{w}_h^+,\mathbf{w}_h^-,\nabla\mathbf{u}_h^+,\nabla\mathbf{u}_h^-,\vec{n})\\&-\mathcal{H}_{\text{ad}}(\varepsilon^+,\varepsilon^-,\mathbf{u}_h^+,\mathbf{w}_h^+,\mathbf{w}_h^-,\mathbf{u}_h^-,\nabla\mathbf{u}_h^+,\nabla\mathbf{u}_h^-,\vec{n})\,\mathrm{d}s\\&+\sum_{b\in\mathcal{B}_h}\int_{\Gamma^b}\mathcal{H}_c^b(\mathbf{u}_h^b(\mathbf{u}_h^+),\vec{n})-\mathcal{H}_v^b(\mathbf{u}_h^b(\mathbf{u}_h^+),\mathbf{w}_h^+,\nabla\mathbf{u}_h^+,\vec{n})\\&-\mathcal{H}_{\text{ad}}^b(\varepsilon^+,\mathbf{u}_h^b(\mathbf{u}_h^+),\mathbf{w}^+,\nabla\mathbf{u}_h^+,\vec{n})\,\mathrm{d}s\end{aligned} \tag{10}$$

where $\mathcal{H}_c(\cdot,\cdot,\vec{n})$ is the convective numerical flux, $\mathcal{H}_v(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\vec{n})$ is the viscous numerical flux, and $\mathcal{H}_{\text{ad}}(,\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\vec{n})$ is the artificial diffusion flux on the interior faces Γ^i. The numerical fluxes $\mathcal{H}_c^b(\cdot,\vec{n})$, $\mathcal{H}_v^b(\cdot,\cdot,\cdot,\vec{n})$, and $\mathcal{H}_{\text{ad}}^b(\cdot,\cdot,\cdot,\cdot,\vec{n})$ denote boundary numerical fluxes (which are different from the interior numerical fluxes) on a boundary edge Γ^b. Hartmann [25] showed that using the Riemann solver on the boundary results in a dual inconsistent discretization. For a dually consistent discretization, the boundary numerical flux is taken as the native flux normal to the boundary evaluated at the boundary condition state $\mathbf{u}_h^b(\mathbf{u}_h^+)$. The interior convective numerical fluxes are chosen to be Riemann solvers, and current implementations include the flux difference splitting schemes of Rusanov [32] and Roe [33].

The numerical flux for the viscous term is obtained via a modified version of the SIP method presented in Ref. [34], which seeks to penalize the solution for being discontinuous at the element interfaces. For full details of the discretization including SIP method details see Refs. [34,35].

IV. Discrete Adjoint-Based Error Estimation

In this work, the adaptation procedure is driven by estimating the error in an output functional of interest. This is known as goal-oriented or adjoint-based adaptation. The objective of this procedure is to adapt the mesh based on an element's contribution to the error in the output functional. The predicted error may also be used to give a correction to the functional value.

A. Formulation

The following formulation is based on the approach described in [17]. Consider the functional of interest $\mathcal{L}(\mathbf{u})$ evaluated with the discrete flowfield variables. Furthermore, consider a coarse mesh $\mathcal{T}_H$ on which a flow solution $\mathbf{u}_H$ has been obtained and used to evaluate the functional $\mathcal{L}_H(\mathbf{u}_H)$ on the coarse (i.e., current) mesh. Given this flow solution and functional, we seek an approach by which we can estimate the functional on a globally refined mesh $\mathcal{T}_h$, without computing the flow solution on the globally refined mesh. Therefore, we expand the fine grid functional in a Taylor series about a solution projected from the coarse mesh to the fine mesh denoted by $\mathbf{u}_H^h$:

$$\mathcal{L}_h(\mathbf{u}_h)=\mathcal{L}_h(\mathbf{u}_H^h)+\left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)_{\mathbf{u}_H^h}(\mathbf{u}_h-\mathbf{u}_H^h)+\cdots \tag{11}$$

where $\mathcal{L}_h(\mathbf{u}_H^h)$ is the fine mesh functional evaluated with the coarse mesh solution projected to the fine mesh. The vector $\left(\partial\mathcal{L}/\partial\mathbf{u}_h\right)_{\mathbf{u}_H^h}$ is the sensitivity of the functional with respect to the solution evaluated at the same projected state. To eliminate the term involving the solution on the fine mesh we appeal to the constraint equation. For steady-state solutions, the temporal derivative is zero, and the residual defined by Eq. (10) can also be expanded about the projected solution:

$$\mathbf{R}_h(\mathbf{u}_h)=\mathbf{R}_h(\mathbf{u}_H^h)+\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}(\mathbf{u}_h-\mathbf{u}_H^h)+\cdots \tag{12}$$

The fine-level residual is constrained to be zero, which allows one to rearrange Eq. (12) to solve for the quantity involving the unknown fine solution as

$$(\mathbf{u}_h-\mathbf{u}_H^h)\approx-\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}^{-1}\mathbf{R}_h(\mathbf{u}_H^h) \tag{13}$$

Upon substitution of Eq. (13) into Eq. (11), one obtains the following expression for the estimate of the error in the functional:

$$\mathcal{L}_h(\mathbf{u}_h)-\mathcal{L}_h(\mathbf{u}_H^h)\approx-\left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}^{-1}\mathbf{R}_h(\mathbf{u}_H^h) \tag{14}$$

where the flow residual on the fine mesh $\mathbf{R}_h(\mathbf{u}_H^h)$ is nonzero because the coarse mesh flow solution projected to the fine mesh does not satisfy the discrete equations on the fine mesh. Next, we define the fine adjoint variable Λ_h as the variable satisfying

$$\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}^T\Lambda_h=\left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)^T \tag{15}$$

Therefore, the functional error can now be defined in terms of the adjoint variable:

$$\mathcal{L}_h(\mathbf{u}_h)-\mathcal{L}_h(\mathbf{u}_H^h)\approx-(\Lambda_h)^T\mathbf{R}_h(\mathbf{u}_H^h) \tag{16}$$

The adjoint solution should be expected to cost as much as the flow solution and thus it is undesirable to compute the fine grid adjoint directly. Therefore, the coarse adjoint solution is obtained via

$$\left[\frac{\partial\mathbf{R}_H}{\partial\mathbf{u}_H}\right]_{\mathbf{u}_H}^T\Lambda_H=\left(\frac{\partial\mathcal{L}_H}{\partial\mathbf{u}_H}\right)^T \tag{17}$$

which is solved on the coarse mesh. In this work, the fine mesh (level) $\mathcal{T}_h$ employed for error estimation contains the same number of elements as the original mesh but employs a discretization order of $p+1$, where the coarse mesh $\mathcal{T}_H$ employs a discretization order of p. Therefore, the transition of mesh levels $H\to h$ is equivalent to the transition of discretization orders $p\to p+1$. To obtain the functional error estimate, the coarse adjoint solution is projected onto the fine mesh by injection. Injection is the process of initializing the fine mesh solution with the coarse mesh solution without performing any interpolation to obtain the fine mesh solution. Because the fine mesh employs a discretization order of $p+1$, the injection operator is defined by setting the modes from 1 to M_p of $\Lambda_H^h=\Lambda_H$ and setting the remaining high-order modes to zero, where M_p is the number of modes in a discretization of order p. The injection operator is followed by a small number (≤ 5) of linear solution iterations on the fine mesh to generate the approximate fine mesh adjoint variable Λ_H^h. Wang and Mavriplis [18] used a patch-wise least-squares method to reconstruct the adjoint on the fine mesh. However, the reconstruction procedure is more complicated in the current case involving mixed element nonconforming meshes. Furthermore, several solution cycles on the fine mesh results in a relatively low cost operation and gives an approximate fine adjoint, which is based on the discrete fine-

mesh equations. Introducing the approximate fine adjoint solution results in the following error estimate:

$$\mathcal{L}_h(\mathbf{u}_h) - \mathcal{L}_h(\mathbf{u}_H^h) \approx -\underbrace{(\Lambda_H^h)^T \mathbf{R}_h(\mathbf{u}_H^h)}_{\varepsilon_c} \tag{18}$$

where ε_c is the computable error. In this work, the element-wise contributions of computable error ε_c are used as the adaptation criteria. Following Mani [19], the mesh is adapted based on attempting to eliminate a certain fraction of the total error, usually >90%. For this work, we have targeted 99% of the total error. The process forms a sorted list of the elements according to their contribution to the total error from the highest error to the lowest error. A loop over the queue is performed, and elements are flagged for refinement until the total amount of error processed exceeds the specified percentage of the total error. This ensures that only the elements with the highest contribution to the total error are refined for highly nonuniform error distributions. When the error has become more uniformly distributed, near-uniform refinement will occur. Once the elements have been tagged for adaptation, they are refined via either a p-enrichment or h-refinement procedure where appropriate.

V. Mesh Refinement Methods

Discontinuous Galerkin methods couple the order of accuracy with the number of degrees of freedom. This gives DG methods additional flexibility with regard to the placement of the degrees of freedom by an adaptation algorithm. In particular, DG methods have two paths by which to increase resolution in a given problem: h-refinement and p-enrichment. Oliver and Darmofal [20] and Oliver [21] developed an unsteady mesh adaptation procedure within the context of high-order DG discretizations. However, the mesh adaptations are performed at a fixed order of accuracy and thus only exploit one method of adding resolution to the problem. Wang and Mavriplis [18] and Wang [36] developed an hp-adaptive approach for DG discretizations of the compressible Euler equations on purely triangular meshes and demonstrated its effectiveness for computing both purely smooth flows and flows with discontinuities. Herein, we propose to extend the work of [18,36] to viscous flows on mixed-element meshes and enhance the discontinuity capturing ability of hp-adaptation, using a similar combined h-refinement and p-enrichment approach. In what follows, we describe each method of adaptation in isolation and then discuss how they are combined to give the overall hp-adaptive approach.

A. *h*-Refinement

Burgess et al. [26] showed that using quadrilateral elements in the highly stretched regions of the mesh is advantageous. As a result, the refinement process becomes more complicated than in the simpler case of conforming triangular meshes [18,36]. For meshes containing quadrilateral elements, it is convenient to allow for nonconforming interfaces (i.e., hanging nodes) in the mesh. Therefore, the triangles are now also refined such that they can have nonconforming interfaces. Refinement of both element types accomplished using isotropic splitting on a four-to-one basis with no more than a two-to-one discrepancy between the size of neighboring elements. Furthermore, although it is commonplace to smooth the refined meshes after they are generated, no smoothing is applied in this work because it can corrupt the structure of the anisotropic boundary-layer mesh.

The refinement pattern for triangles is depicted in Fig. 1. The triangle is refined using midpoint subdivision where a node is inserted at the midpoint of each edge on the triangle. This results in four children (4:1) for each subdivided element. The quadrilaterals are refined in an analogous manner as depicted in Fig. 2, with the exception that an additional node is placed at the center of the refined quadrilateral.

B. *p*-Enrichment

In contrast to h-refinement, p-enrichment refines the element in question by maintaining the current element size and connectivity. The p-enrichment procedure is much simpler than the h-refinement procedure and consists of simply increasing the discretization order from p to $p + 1$ on the element flagged for refinement. p-enrichment is implemented without regard to element type as depicted in Figs. 3a and 3b, and a jump of no more than one order is permitted between elements.

C. *hp*-Adaptation

It is well known that using high-order polynomials in the vicinity of discontinuities results in Gibbs phenomena, which can cause solver failure for the Navier–Stokes/Euler equations. Thus, we propose to combine h-refinement and p-enrichment to account for the presence of discontinuities in the solution. This procedure will use h-refinement near discontinuities and p-enrichment in smooth flow regions. The objective is to allow discontinuities to be captured using a low-order discretization while using high-order polynomials in smooth flow regions where the use of high-order polynomials is appropriate.

hp-adaptation is a hybridization of the h-refinement and p-enrichment techniques. These two techniques are used in tandem such that, if an element is to be refined, a decision must be made as to whether to use h-refinement or p-enrichment. The current implementation examines the smoothness of the primal solution to determine which type of adaptation is used for each element. The smoothness is determined by examining the jump indicator of [37]. The value of the jump indicator for an element is given by

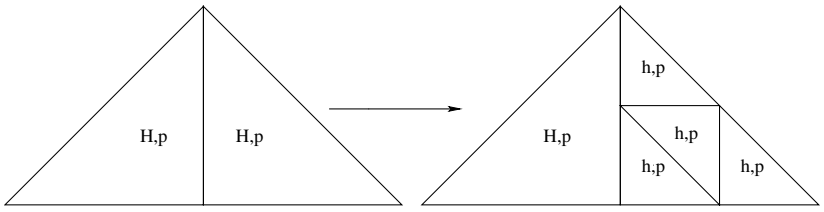


Fig. 1 Illustration of the triangle *h*-refinement pattern.

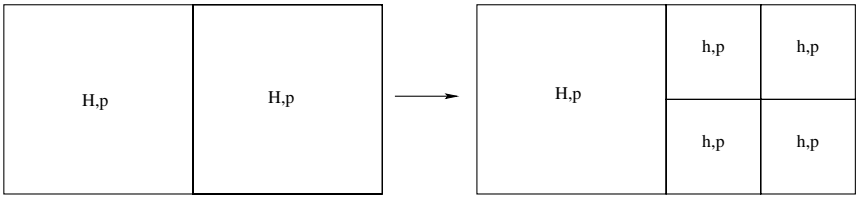
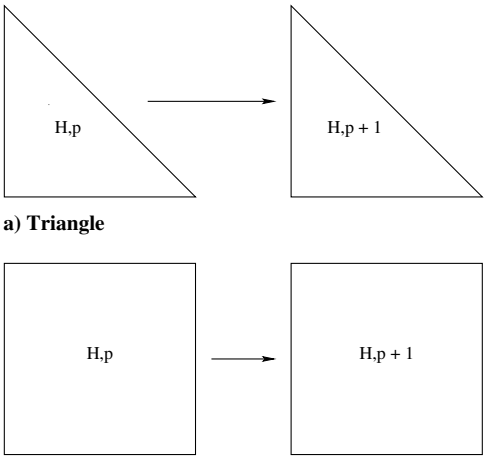


Fig. 2 Illustration of the quadrilateral *h*-refinement pattern.



a) Triangle

b) Quadrilateral

Fig. 3 Illustration of *p*-enrichment on both triangles and quadrilaterals.

$$(\tilde{s}_k)_f = \frac{1}{|\partial\Omega_k|}\int_{\partial\Omega_k}\left|\frac{[q]\cdot\vec{n}}{\{q\}}\right| \mathrm{d}s \tag{19}$$

where q is taken as each of the density, velocities, and pressure separately, resulting in an $(\tilde{s}_k)_f$ for each quantity:

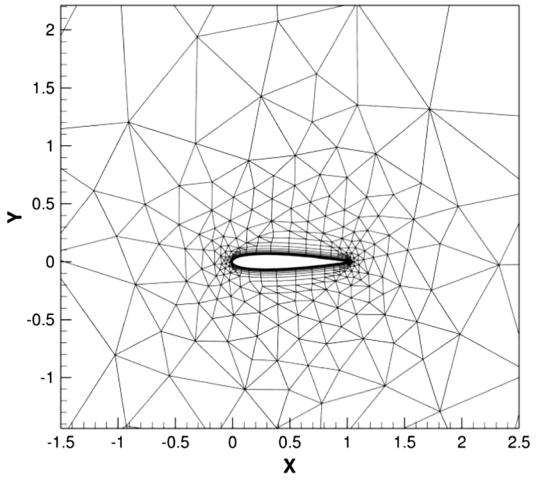
$$\tilde{s}_k = \max_{f=1\ldots nf}((\tilde{s}_k)_f) \tag{20}$$

The previous expressions simply state that we compute the jump of density, velocities, and pressure on each cell. We then take the maximum value of these jumps to use as the smoothness indicator for the cell. The choice between whether to adapt an element with *h*-refinement or *p*-enrichment is made by

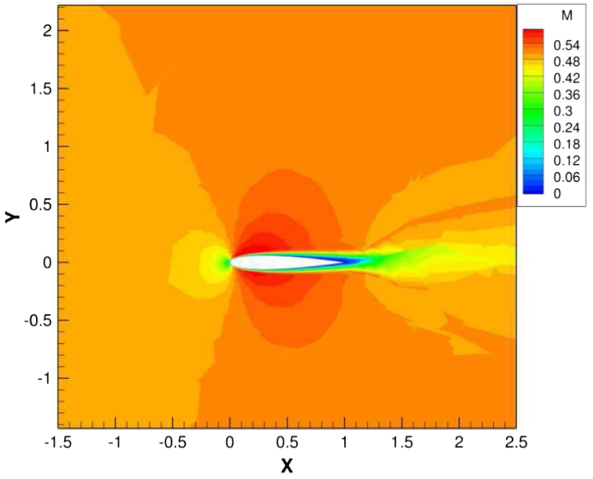
$$\begin{cases} \tilde{s}_k > \frac{1}{\kappa}, & h\text{ -refinement} \\ \tilde{s}_k < \frac{1}{\kappa}, & p\text{ -enrichment} \end{cases} \tag{21}$$

where, as in [18], $\kappa = 25$ is used throughout this work. In addition to selecting the adaptation mode based on the solution smoothness, a maximum discretization order in also enforced. When a cell reaches its maximum discretization order and further refinement is required, *h*-refinement is substituted for *p*-enrichment, even if the solution within the cell is smooth.

Though *hp*-adaptation is designed to place degrees of freedom optimally for a given objective functional, *hp*-adaptation can also be viewed as a technique to enhance the robustness of the DG solver. In essence, *hp*-adaptation seeks to design the mesh based on the solution, which for cases of underresolved phenomena, such as those encountered in [26], should result ultimately in a mesh of sufficient local resolution such that high-order polynomials can be used throughout. For flow features that will most likely remain underresolved for the entire simulation (e.g., shocks and contact discontinuities), the *hp*-adaptive scheme is capable of adding degrees of freedom while maintaining low *p* order in such a way as to

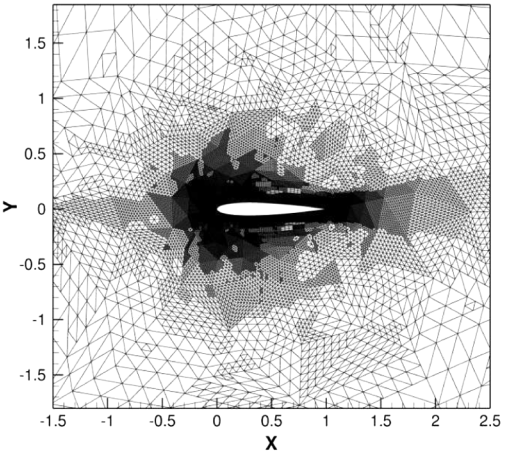


a) Initial Grid: 1,148 elements, $p = 1$

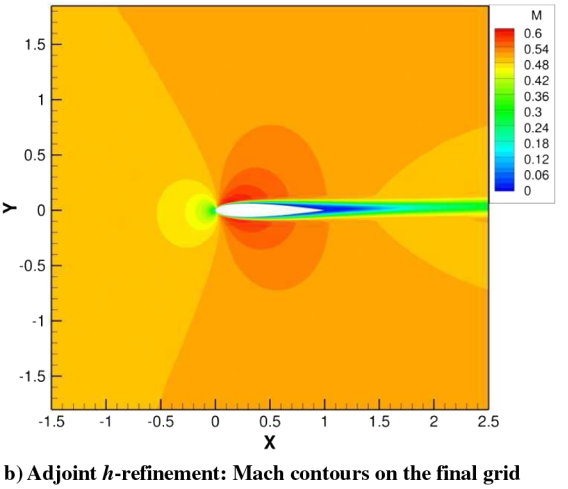


b) Mach Contours: Initial Grid

Fig. 4 Initial mesh and Mach number contours of the laminar flow over a NACA0012 airfoil with $p = 1$, $M_\infty = 0.5$, $\alpha = 1$ deg, and $Re = 5000$.

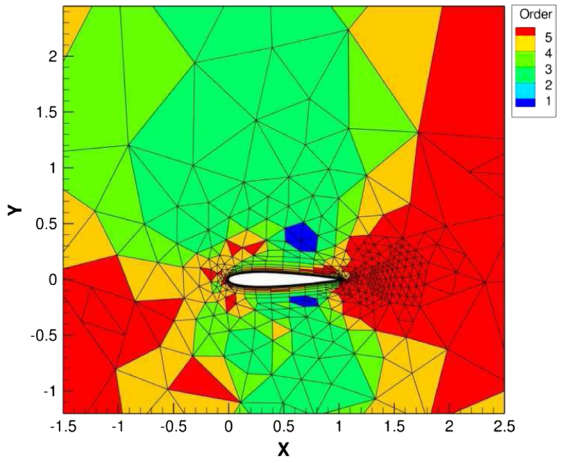


a) Adjoint *h*-refinement: Final grid with 174,145 elements, $p = 1$

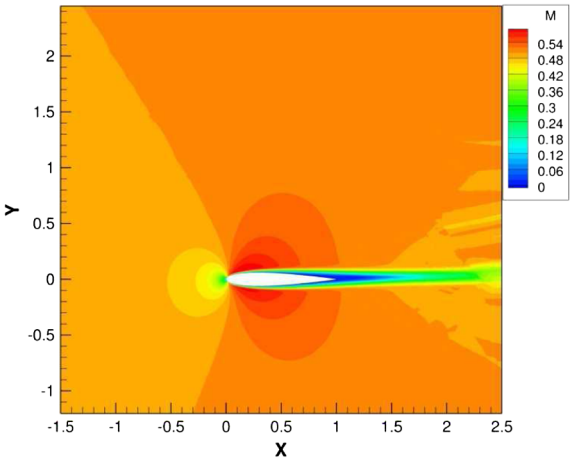


b) Adjoint *h*-refinement: Mach contours on the final grid

Fig. 5 Final mesh and Mach number contours of the laminar flow over a NACA0012 airfoil using adjoint *h*-adaptation with $p = 1$, $M_\infty = 0.5$, $\alpha = 1$ deg, and $Re = 5000$.

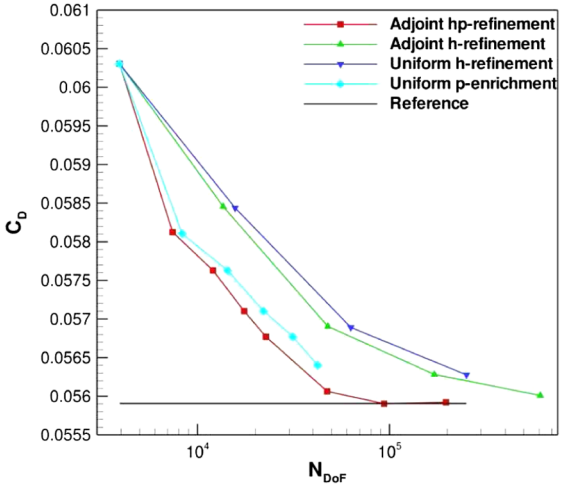


a) Adjoint *hp*-adaptation: Final grid with 6,776 elements, $p = 1$ to $p = 5$

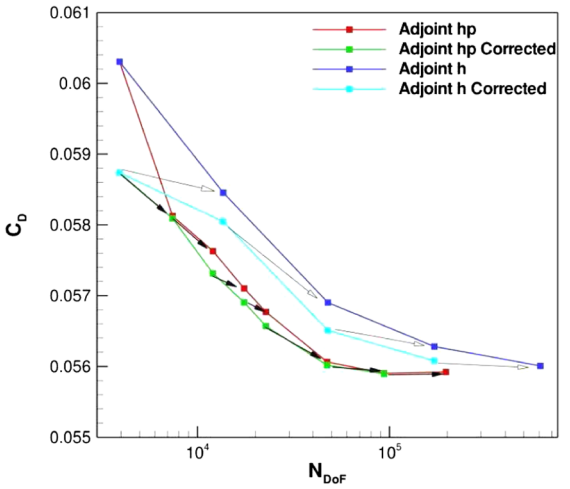


b) Adjoint *hp*-adaptation: Mach contours on the final grid

Fig. 6 Final mesh and Mach number contours of the laminar flow over a NACA0012 airfoil using adjoint *hp*-adaptation with $p = 1$ to $p = 5$, $M_\infty = 0.5$, $\alpha = 1$ deg, and $Re = 5000$.



a) Drag versus N_{DoF}



b) Adjoint Corrected Drag versus N_{DoF}

Fig. 7 Drag vs N_{DoF} for the laminar flow over a NACA0012 airfoil using various adaptation methods, with and without adjoint corrections.

avoid Gibbs phenomena, providing a natural way for the present DG solver to handle nonsmooth solutions robustly.

VI. Implicit Steady-State Solver

The flow equations arising at each adaptive step are solved using a Newton method where the linear system is solved approximately at each Newton step with a GMRES solver. By neglecting the temporal derivative, the system of equations in Eq. (9) becomes

$$\mathbf{R}_h(\mathbf{u}_h) = 0 \tag{22}$$

where $\mathbf{R}_h(\mathbf{u}_h)$ is the nonlinear residual. This set of nonlinear equations is solved using Newton's method:

$$\left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}\right]^n \Delta \mathbf{u}_h^{n+1} = -\mathbf{R}_h(\mathbf{u}_h^n) \qquad \mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \Delta \mathbf{u}_h^{n+1} \tag{23}$$

Newton's method will diverge if the initial guess is too far from the final solution. Thus, the flux Jacobian matrix is augmented with a damping term to increase robustness. The damped Newton iteration is given as

$$\left[\frac{[I]}{\Delta t^n} + \frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}\right]^n \Delta \mathbf{u}_h^{n+1} = -\mathbf{R}_h(\mathbf{u}_h^n) \qquad \mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \Delta \mathbf{u}_h^{n+1} \tag{24}$$

where Δt is an element-wise time step used as a damping factor:

$$\mathrm{CFL} = \min\left(\mathrm{CFL}_{\min}\left(\frac{\|\mathbf{R}_h\|_2^0}{\|\mathbf{R}_h\|_2^n}\right)^r, \mathrm{CFL}_{\max}\right)$$
$$\Delta t_k^n = \frac{\mathrm{CFL}|\Omega_k|}{|\partial\Omega_k|(|\vec{u}_k| + a_k)} \tag{25}$$

where the CFL is the Courant–Friedrichs–Lewy number, and a is the sound speed. Because of the block-sparse nature and size of the matrix, an iterative method is used to solve the linear system arising from the Newton's method. The linear adjoint problem is solved in a similar manner using a defect-correction approach where the same GMRES solver is used as the smoother during each defect-correction step. See Refs. [26,29] for details of the solver and a review of its performance.

VII. Numerical Results

The proposed hp-adaptive method has been evaluated using three test cases. The first two test cases are the laminar viscous flow over a NACA0012 airfoil. The first test case is presented to compare hp-adaptation with h-refinement as well as with both uniform h-refinement and uniform p-enrichment. The second test case is the transonic flow over a NACA0012 airfoil. This case is presented to demonstrate the accurate and robust shock capturing ability of the hp-adaptive method, and the results of hp-adaptation are compared with uniform p-enrichment using an artificial diffusion method.

Where appropriate, the computational grid may contain both triangles and quadrilaterals within the same grid. The adaptive algorithm addresses these grids by allowing for nonconforming interfaces between elements of all types. For all test cases, the adaptation is terminated when the functional of interest is grid-converged (i.e., the functional changes by less than 0.5% from one adaptation step to the next). The performance of the method is measured by considering how many DoFs are required to yield a grid-converged functional. The number of DoFs is determined as the total number of unknowns per equation in the domain. For example, a purely triangular mesh of 100 elements with a uniform discretization order of $p = 1$ would have 300 DoFs. We also show the computational cost of generating these results by showing functional or functional error versus wall time (i.e., CPU time or computational time).

All results have been computed using the Riemann solver of Roe [33] on the cell interfaces. All test cases are steady-state solutions, and the flow and adjoint equations have been converged such that their residuals have been reduced by 12 orders of magnitude at each stage of the adaptive process. In some sense, this represents the worst-case scenario for timing adaptive methods because one would probably only partially converge the intermediate steps before moving on to the next adaptive cycle.

The presented wall times for the various numerical results can vary significantly from test case to test case, which is the result of hardware replacements that could not be avoided.

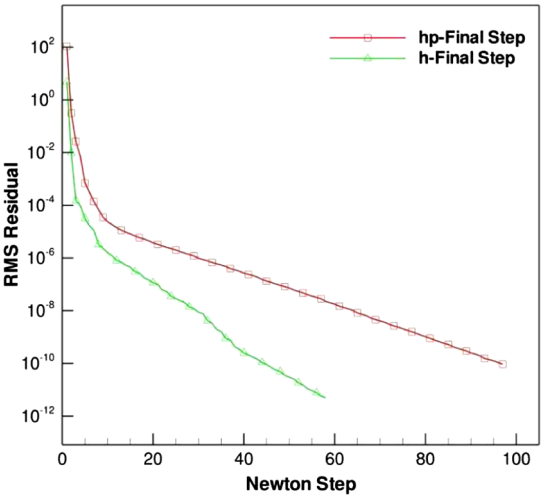
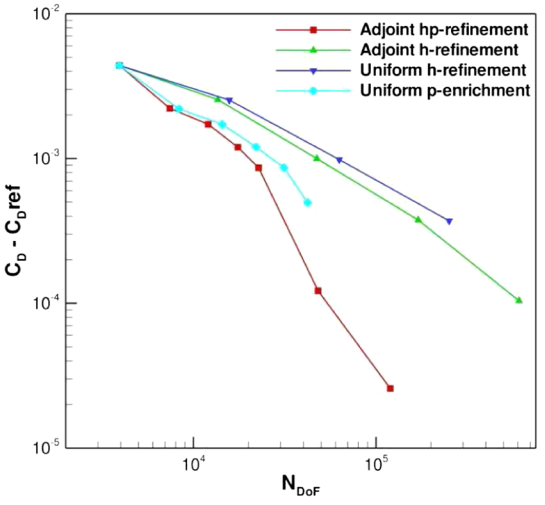


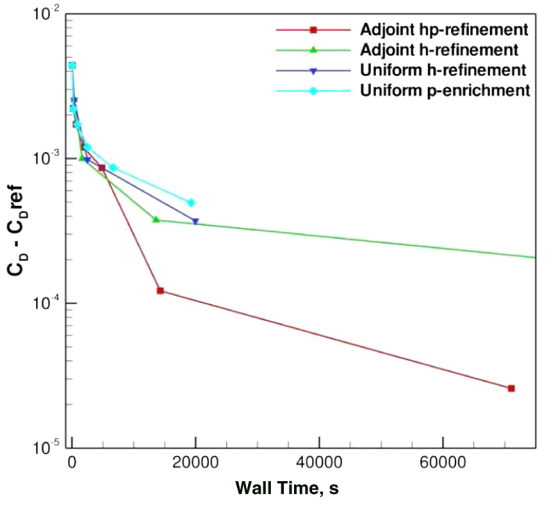
Fig. 9 Flow-solver iterative convergence for the laminar flow over a NACA0012 airfoil.

A. NACA0012 Airfoil: Drag-Based Adaptation

The first test case consists of the laminar flow over a NACA0012 airfoil. The flow conditions are $M_\infty = 0.5$, $\alpha = 1$ deg, and $Re = 5000$. The initial grid contains 1148 elements, with a uniform discretization order of $p = 1$, resulting in 3930 DoFs. This case is

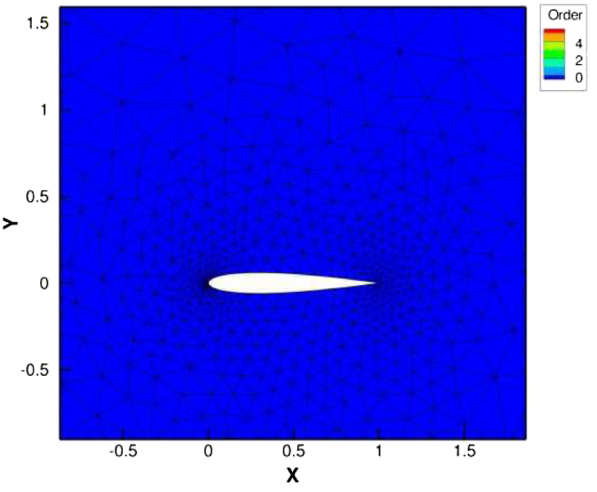


a) Drag Error versus N_{DoF}

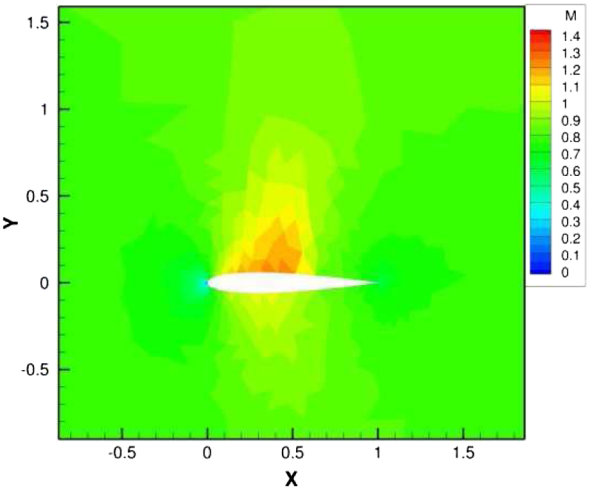


b) Drag error versus wall time

Fig. 8 Drag error for the laminar flow over a NACA0012 airfoil using various adaptation methods.



a) Initial grid: 1,556 elements, $p = 0$



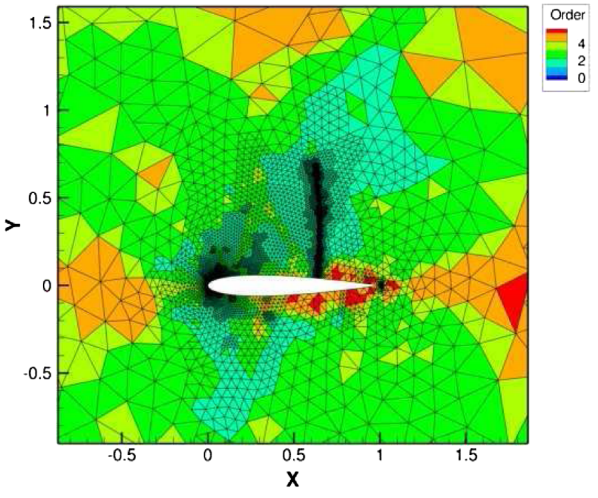
b) Mach contours: Initial grid

Fig. 10 Initial mesh and Mach number contours for the transonic flow over a NACA0012 airfoil with $p = 0$, $M_\infty = 0.8$, and $\alpha = 1.25$ deg.

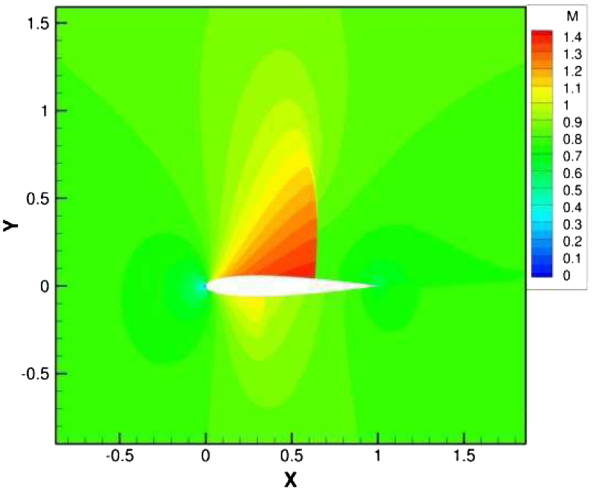
computed using adjoint-based h-adaptation with a discretization order of $p = 1$ as well as the hp-adaptive approach. Additionally, uniform h-refinement and uniform p-enrichment was performed for comparative purposes. When employing the hp-adaptive approach for this case, the maximum discretization order in the grid is set at $p = 5$; thus, any cell that requires refinement and already has a discretization order of $p = 5$ will be subdivided regardless of solution smoothness within a cell. This example is shown to illustrate the high efficiency of the hp-adaptive approach, i.e., hp-adaptation can produce very accurate functionals with respect to the reference solution using relatively few degrees of freedom when compared against uniform refinement/enrichment and h-refinement. We have chosen drag as the output functional for the adjoint adaptations.

Figures 4a, 4b, 5a, 5b, 6a, and 6b depict the initial and final grids using both adjoint h-refinement and adjoint hp-adaptation as well as Mach number contours on those grids. Note that the adjoint-based strategies target both the surface of the airfoil as well as the wake region downstream from the trailing edge. Both h-refinement and hp-adaptation Mach number contours look very similar at the final stage; however, examination of Figs. 7 and 8 show that the hp-adaptation results contain about one-third the number of DoFs compared to the h-refinement results. Figure 9 depicts the iterative convergence of the flow solver for the final adaptive step of the adjoint h-refinement and adjoint hp-adaptation; one should note that residuals are reduced by 12 orders of magnitude in both cases to eliminate any algebraic error from affecting the functional values.

Figure 7a depicts the computed drag versus the number of DoFs using adjoint-based h-refinement, adjoint-based hp-adaptation, uniform h-refinement, and uniform p-enrichment. The reference value in Fig. 7a was computed using the same DG solver with approximately 250,000 DoFs at a uniform discretization order of $p = 4$. Figure 7a clearly shows that the hp-adaptive method yields a grid-converged drag result with the fewest number of degrees of freedom compared to any of the refinement methods presented. Comparison of the hp-adaptive approach with the h-refinement approach shows that the hp-adaptive approach yields a grid-converged drag result with approximately one-third the number of DoFs used in the h-refinement approach. Figure 7b depicts the drag versus the number of DoFs using the error predicted according to Eq. (18) to correct the coarse level drag, where the arrows point from the coarse level corrected drag to the fine level drag that is being estimated. Although initially the corrected coarse level drag seems quite poor, as the refinement process continues and the drag becomes closer to grid-converged, the corrections improve significantly. At this point, the adjoint correction yields corrected coarse-level drag values that closely match the corresponding fine-level drag values. The increased effectiveness of the correction is explained by the fact that the error is predicted as a linear Taylor series expansion about the coarse level, and thus as the drag becomes closer to grid-converged, the linear Taylor series becomes a better approximation of the

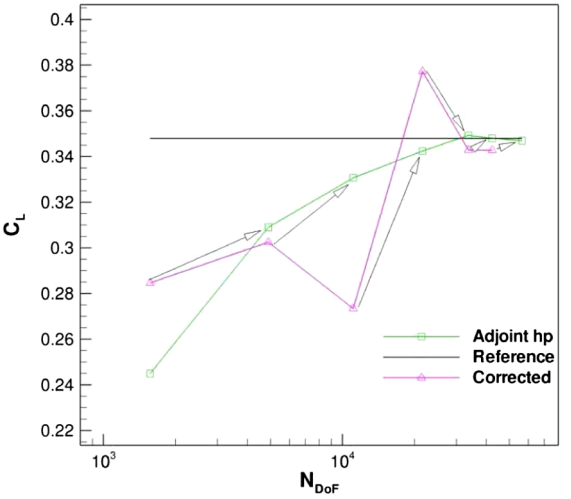


a) Adjoint *hp*-refinement: Final grid with 26,407 elements, $p = 0$ to $p = 5$

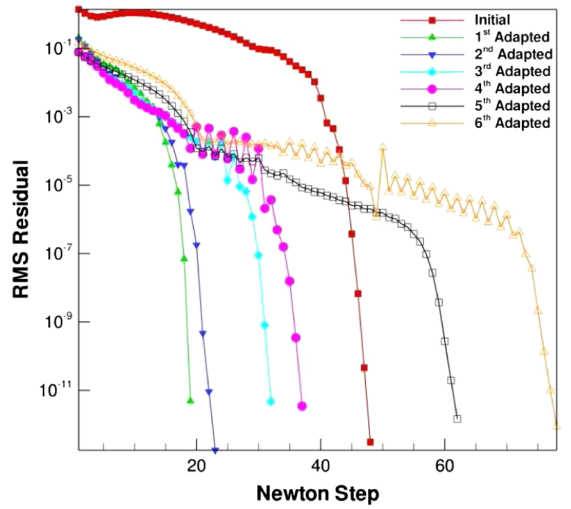


b) Adjoint *hp*-refinement: Mach contours on the final grid

Fig. 11 Final mesh and Mach number contours for the transonic flow over a NACA0012 airfoil ($M_\infty = 0.8$ and $\alpha = 1.25$ deg) using adjoint *hp*-adaptation; the discretization order varies from $p = 0$ to $p = 5$.



a) Adjoint *hp*-adaptation: Lift versus NDoF



b) Iterative convergence

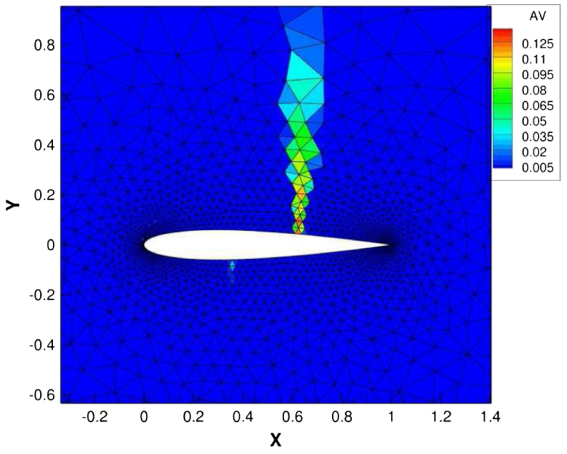
Fig. 12 Transonic NACA0012 lift vs N_{DoF} using *hp*-adaptation and iterative convergence.

behavior between coarse- and fine-level drag values. The adjoint hp-adaptive method clearly gives the most accurate drag result (i.e., closest to the reference solution) for a given number of DoFs. Note that the h-refinement case was terminated early because the number of DoFs became impractically high, and it was clear that the hp-adaptation results had become grid-converged using far fewer DoFs.

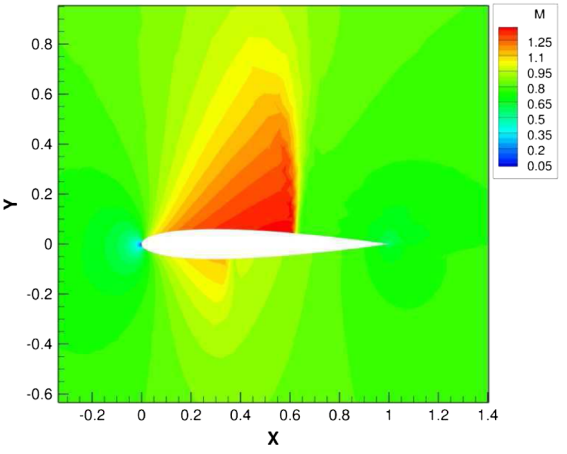
Figures 8a and 8b show the drag error versus the number of DoFs and the wall time. Figure 8a shows that the hp-adaptive method gives the lowest drag error per degree of freedom. The slope of these error curves was computed for both the h-refinement and hp-adaptation cases. Theoretically, for a uniform discretization order, the drag error should converge like $\mathcal{O}(h^{2p})$ (where $h = \sqrt{N_{\text{DoF}}}$ for two-dimensional flows). Superconvergence of the functional error is a property of dual-consistent discretizations [25]. Computation of the slope for both the uniform and adjoint h-refinement drag error curves in Fig. 8a yields the expected value of 2. Computation of the slope of the hp-adaptation drag error curve results in a slope of 8.8, which is a striking result. Even though only a fraction of the grid contains cells with a $p = 5$ discretization, we are able to get very close the theoretical slope of 10, and certainly, by including some $p = 5$ cells in the grid, we are achieving a higher slope than we could by using $p = 4$ and uniform mesh refinement. Figure 8b shows the drag error versus the wall time. Again, we see that hp-adaptation yields low-error functionals using a fraction of the wall time required by the h-refinement method. This is yet another measure of the efficiency of hp-adaptation. Recall that the wall time shown for the adjoint-based methods includes the fully converged adjoint and flow solutions.

B. Transonic NACA0012 Airfoil: Lift-Based Adaptation

As a second and final example, the hp-adaptive method is applied to an inviscid transonic flow around a NACA0012 airfoil. Although the previous two test cases contained smooth flow solutions, this case has both a strong and a weak shock wave. This final case represents a scenario where the hp-adaptive approach not only yields high efficiency but also enhanced robustness. The flow conditions for this case are $M_\infty = 0.8$ and $\alpha = 1.25$ deg. For comparative purposes, three refinement scenarios are employed for this test case. In the first scenario, the grid initially contains 1566 triangles with a uniform discretization order of $p = 0$, resulting in 1566 DoFs and is subsequently adapted using hp-adaptation in the absence of any artificial diffusion. The second scenario uses uniform p-enrichment and a grid with $N = 3086$ triangles and artificial diffusion to stabilize the high-order solution in the presence of shock waves. The final scenario employs hp-adaptation, where the grid initially contains 1566 triangles with a uniform discretization order of $p = 1$, which requires artificial diffusion to stabilize the solution in the vicinity of the shock waves. The objective functional for the adjoint-based refinements is lift. A reference solution was computed with a second-order finite-volume method using a mesh that contains $N = 200,000$ elements. This particular case was chosen to enable comparison between adaptation using $p = 0$ in the vicinity of the shock wave without any artificial diffusion, adaptation using $p = 1$ with artificial diffusion, and uniform p-enrichment using artificial

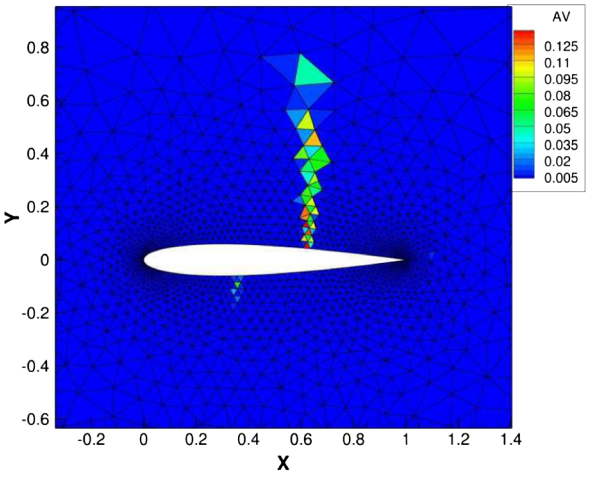


a) Initial artificial viscosity and grid: 3,086 elements, $p = 1$

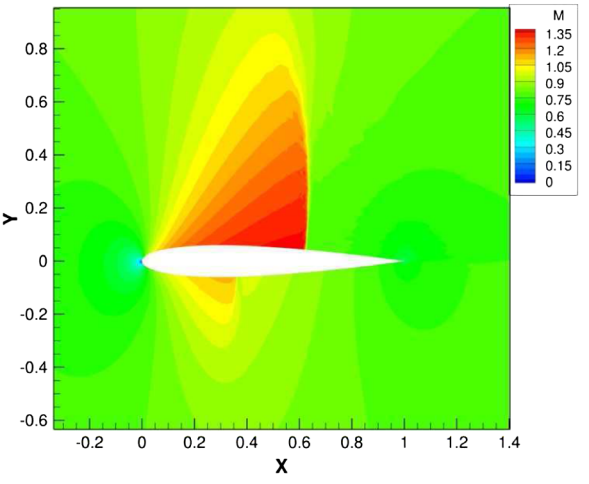


b) Mach contours: Initial grid

Fig. 13 Initial artificial viscosity and Mach number contours for the transonic flow over a NACA0012 airfoil with $p = 1$, $M_\infty = 0.8$, and $\alpha = 1.25$ deg.



a) Uniform p-enrichment: Artificial diffusion and final grid with 3,086 elements, $p = 4$



b) Uniform p-enrichment: Mach contours on the final grid

Fig. 14 Final artificial viscosity and Mach number contours for the transonic flow over a NACA0012 ($M_\infty = 0.8$ and $\alpha = 1.25$ deg) airfoil using uniform p-enrichment, the discretization order is $p = 4$.

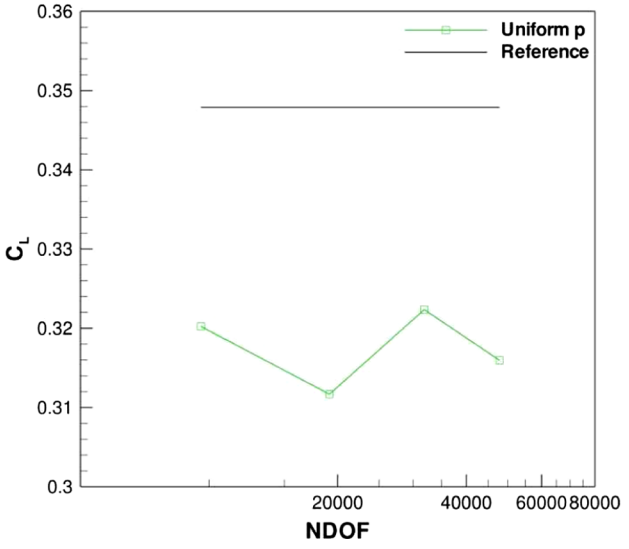
diffusion. It should be noted that using the artificial diffusion with high discretization orders can be difficult, due to the need to change the artificial diffusion settings from discretization order to discretization order.

Figures 10a, 10b, 11a, and 11b depict the grids and Mach number contours for this case at the initial and final stages of the adaptive process using hp-adaptation with $p = 0$ without artificial diffusion. At the final adaptive stage, we have a very sharply resolved shock wave free of oscillations. Figure 12a depicts the lift versus N_{DoF}, and it is clear that a grid-converged lift coefficient that closely matches the reference value is obtained using approximately 60,000 DoFs. The hp-adaptive algorithm has used less than one-third the number of DoFs used to compute the reference solution. Figure 12b depicts the iterative convergence for all adaptive cycles, indicating that a fully converged solution is obtained at every stage of the adaptive process. Furthermore, this approach is relatively robust, requiring less than 100 Newton iterations in all cases. Although there are $p = 0$ cells directly involved in the computation of the lift, we still achieve nicely grid-converged results with a very efficient algorithm.

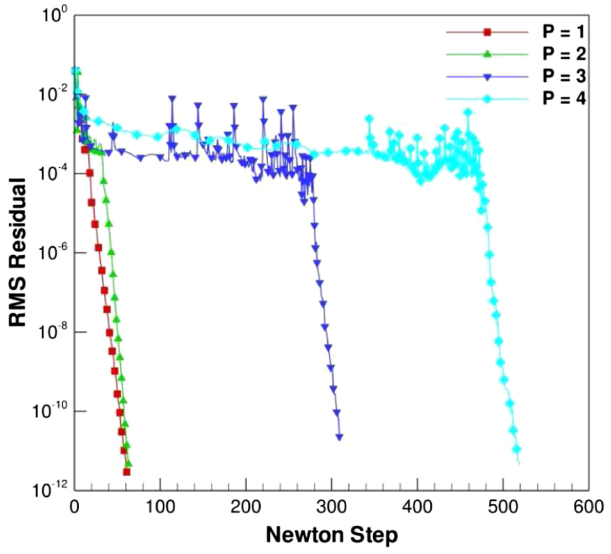
Although no explicit limiter has been used to generate these results, one can view the hp-adaptive approach as a form of limitation. Traditional slope limiters effectively reduce the order of accuracy locally. The idea behind slope limitation is to assume that high discretization order is appropriate everywhere in the grid and then to remedy those areas where it is not appropriate, corresponding to a top-down approach. hp-adaptation can be viewed as a bottom-up approach to limitation because hp-adaptation starts with low discretization order and moves toward high discretization order where appropriate. In the context of DG discretizations, the bottom-up approach has an advantage because it takes the coupling between order of accuracy and resolution into account naturally.

As a point of comparison, this flow was also computed using an artificial diffusion method, which was described in Sec. II. The flow was computed using $p = 1$ to $p = 4$ on a triangular mesh with 3189 elements, which corresponds to approximately 10,000 to 50,000 DoFs.

Figures 13a, 13b, 14a, and 14b depict the grid, artificial diffusion contours, and Mach number contours for this flow computed with uniform p-enrichment. Note how the shock wave sharpens when uniform p-enrichment is applied. This gives an indication that increasing p has increased the resolution of the flowfield. The lift versus N_{DoF} is depicted in Fig. 15a. One can clearly see that, with increasing discretization order, the lift does not converge to a fixed value as uniform p-enrichment is applied, despite the obvious increase in resolution. The conclusion is that, although the resolution is certainly increased, the artificial diffusion has compromised the accuracy of the higher-order result. Figure 15b depicts the iterative converge for this case using artificial diffusion and uniform p-enrichment. Clearly, computing shocks with this method can become quite expensive, especially when compared with the adjoint hp-adaptation convergence history in Fig. 12b. Additionally, for each

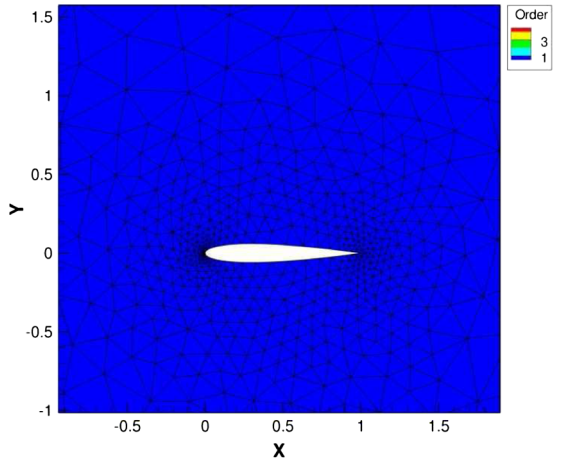


a) Lift versus N_{DoF}

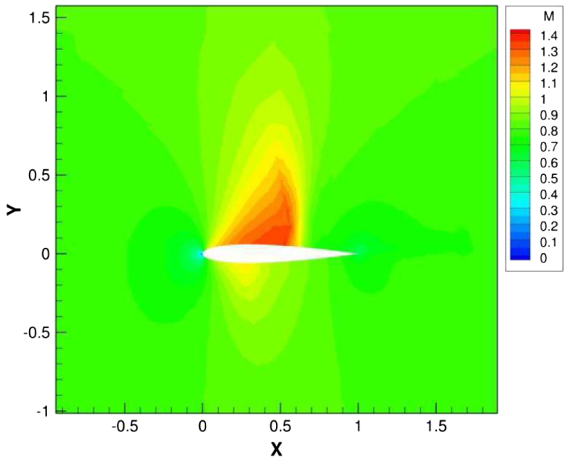


b) Iterative convergence

Fig. 15 Transonic NACA0012 lift vs N_{DoF} using artificial diffusion with $p = 1$ to $p = 4$ and iterative convergence.



a) Initial grid: 1,556 elements, $p = 1$



b) Mach contours: Initial grid

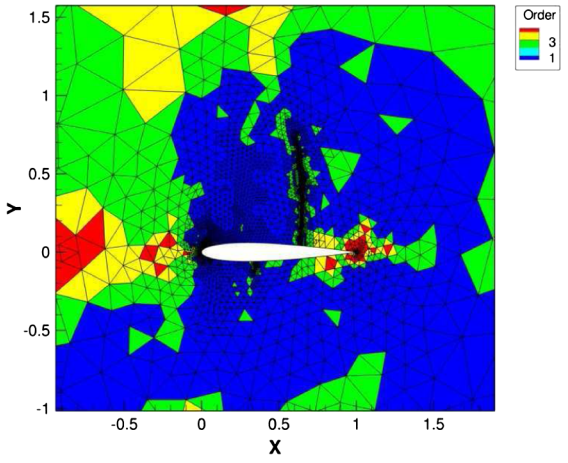
Fig. 16 Initial mesh and Mach number contours for the transonic flow over a NACA0012 airfoil with $p = 1$ and artificial diffusion, $M_\infty = 0.8$, and $\alpha = 1.25$ deg.

order of accuracy, the artificial diffusion parameters had to be adjusted sometimes by a factor of 2 or more to get the solution to converge. Adjusting the parameters is the cause of the poor functional grid convergence observed for this test case.

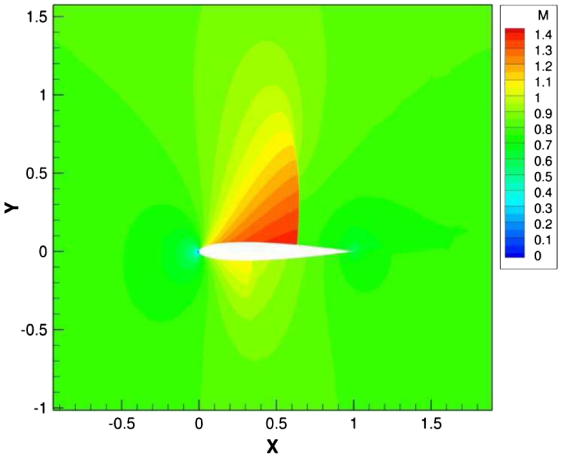
As a third and final test, adjoint hp-adaptation with a minimum value of $p = 1$ and artificial diffusion are combined and applied to this transonic flow test case. The motivation for having at least a $p = 1$ discretization is simply that the SIP method relies on at least a $p = 1$ discretization for the viscous terms; thus, if one desires to compute viscous flows with shock waves, this is a requirement that must be satisfied. Further, it is of interest to investigate if hp-adaptation can remedy the poor objective convergence that the uniform p-enrichment results, in the presence of artificial diffusion, have demonstrated.

Figures 16a, 16b, 17a, and 17b depict the grid and Mach number contours for this flow computed with adjoint hp-adaptation and artificial diffusion. One clearly sees that h-refinement is applied in the vicinity of the shock wave and p-enrichment is applied elsewhere. Also, the shock-wave thickness reduces dramatically during the adaptive process. Figure 18a shows the lift versus N_{DoF} for this case, and one can immediately notice that hp-adaptation remedies the grid-convergence issues that uniform p-enrichment has demonstrated. This remedy is mainly due to the fact that, in the areas where artificial diffusion is applied, mesh refinement is employed as opposed to p-enrichment. The application of localized mesh refinement in the vicinity of the shock wave results in a significantly more grid-converged functional than p-enrichment in this region of the mesh. The application of hp-adaptation required no adjustments in the artificial viscosity settings and therefore a grid-converged functional is obtained using this method of refinement. Comparison of Figs. 18a and 12a shows that using hp-adaptation with artificial diffusion uses significantly fewer degrees of freedom than hp-adaptation using $p = 0$ as the minimum discretization order. This improvement in efficiency is due to the uniform second-order accuracy of the smooth portions of the flow, resulting in a significantly better initial solution. Figure 18b depicts the iterative convergence of the flow solver. One should immediately note that the hp-adaptation results take far fewer Newton steps than the uniform p-enrichment results, especially for the final refinement in each case.

Figure 19a depicts the lift versus the computational time, which clearly shows that hp-adaptation yields more-accurate lift values at a reduced cost compared to higher-order shock capturing with artificial diffusion. However, the most efficient method involves combining the artificial diffusion with hp-adaptation using a minimum discretization order of $p = 1$. This combination produces a grid-converged functional using the least computational time. Furthermore, the application of hp-adaptation has remedied the poor functional convergence observed when applying artificial diffusion with uniform p-enrichment. Furthermore, during hp-adaptation, the shock-wave position can change, which can result in the shock wave entering a high-order element. Should this occur using hp-adaptation alone, the solver will fail due to Gibbs phenomena. However, hp-adaptation combined with artificial diffusion can deal with this scenario automatically, because the artificial viscosity will become

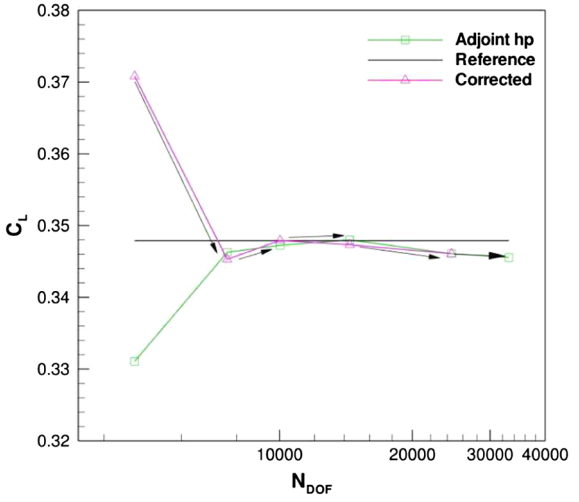


a) Adjoint *hp*-refinement: Final grid with 8,346 elements, $p = 1$ to $p = 4$

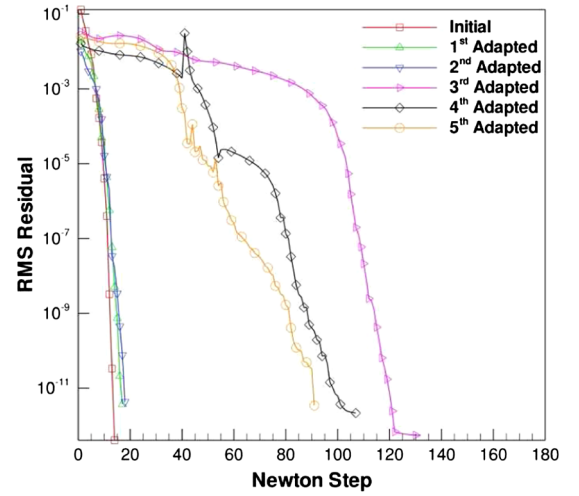


b) Adjoint *hp*-refinement: Mach contours on the final grid

Fig. 17 Final mesh and Mach number contours for the transonic flow over a NACA0012 airfoil using adjoint *hp*-adaptation ($p = 1$ to 4) and artificial diffusion.



a) Adjoint *hp*-adaptation: Lift versus N_{DoF}



b) Iterative convergence

Fig. 18 Transonic NACA0012 lift vs N_{DoF} using *hp*-adaptation and artificial diffusion and iterative convergence.

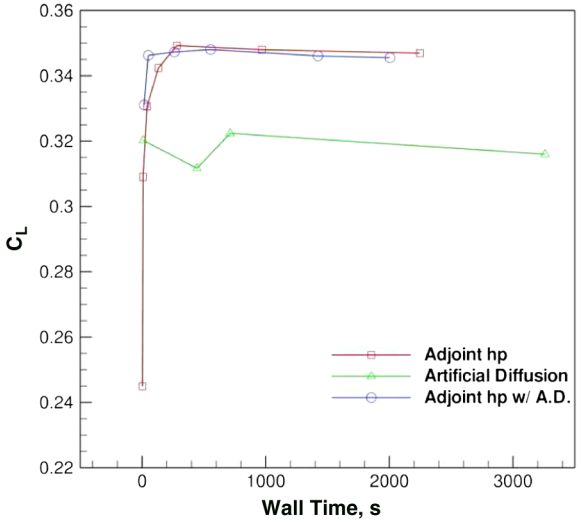


Fig. 19 Comparison of transonic NACA0012 lift vs wall time using artificial diffusion with uniform *p*-enrichment and *hp*-adaptation.

active and stabilize the element. Therefore, the combination of these procedures is by far the most robust method presented. Although the hp-adaptive approach is not the most elegant shock-capturing method for DG discretizations, hp-adaptation has some significant advantages; hp-adaptation gives robust and fast iterative convergence, and with each refinement the functional improves and eventually achieves a grid-converged value. The results indicate that the artificial diffusion should be combined with at least h-refinement to achieve grid-converged functional values. In particular, this work has shown hp-adaptation to be a very effective choice for combination with artificial diffusion.

VIII. Conclusions

An hp-adaptive high-order discontinuous Galerkin (DG) solver for the Navier–Stokes equations has been developed and applied to two test cases. The adaptive method presented is driven by a goal-oriented approach, which makes use of adjoint-based error estimation. The adaptive algorithm is capable of adapting both the grid and discretization order locally. The solver adapts the grid nonconformally to allow for its use on mixed-element meshes such as those shown in the numerical results. The hp-adaptive method has demonstrated high efficiency by computing high accuracy functionals using fewer degrees of freedom than the reference or uniform refinement solutions. Furthermore, a transonic test case has shown the robustness of an hp-adaptive approach to shock capturing. hp-adaptation has also been shown to remedy the grid convergence of functionals when capturing shock waves with an artificial diffusion method, which demonstrates how to use artificial diffusion to simultaneously obtain a robust yet accurate high-order DG flow solver.

Appendix A: Artificial Diffusion Settings

The artificial diffusion method implemented in this work requires the setting of three constants. Tables A1 and A2 show the values of these constants for the transonic flow case. Table A1 gives the constants when uniform p-enrichment is used, and Table A2 gives the constants for the hp-refinement case.

Table A1 Artificial diffusion constants for transonic flow using uniform *p*-enrichment

Order	ε_0	c_{s_0}	κ
1	2.0	1.0	4.0
2	1.0	1.5	3.5
3	.5	1.5	3.5
4	.5	.5	3.5

Table A2 Artificial diffusion constants for transonic flow using *hp*-adaptation

Adaptation step	ε_0	c_{s_0}	κ
1	2.0	1.0	4.0
2	2.0	1.0	4.0
3	2.0	1.0	4.0
4	2.0	1.0	4.0
5	2.0	1.0	4.0

Acknowledgments

The first author would like to acknowledge the contributions of Dimitri Mavriplis, Jay Sitaraman, Karthik Mani, and Brian Lockwood. This work was supported under the U.S. Air Force Office of Scientific Research grant FA9950-10-C-0051 and the National Science Foundation grant 0904936.

References

[1] Bassi, F., and Rebay, S., "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," *Journal of Computational Physics*, Vol. 138, No. 2, 1997, pp. 251–285. doi:10.1006/jcph.1997.5454

[2] Cockburn, B., and Shu, C.-W., "Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *SIAM Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261. doi:10.1023/A:1012873910884

[3] Nastase, C. R., and Mavriplis, D. J., "High-Order Discontinuous Galerkin Methods Using an hp-Multigrid Approach," *Journal of Computational Physics*, Vol. 213, No. 1, March 2006, pp. 330–357. doi:10.1016/j.jcp.2005.08.022

[4] Fidkowski, K. J., and Darmofal, D. L., "Development of a Higher-Order Solver for Aerodynamic Applications," *42nd Aerospace Sciences Meeting and Exhibit*, Reno NV, AIAA Paper 2004-0436, Jan. 2004.

[5] Nair, R., Choi, H.-W., and Tufo, H., "Computational Aspects of Scalable High-Order Discontinuous Galerkin Atmospheric Dynamical Core," *Computers and Fluids*, Vol. 38, No. 2, Feb 2009, pp. 309–319. doi:10.1016/j.compfluid.2008.04.006

[6] Wang, L., Mavriplis, D. J., and Anderson, W. K., "Adjoint Sensitivity Formulation for Discontinuous Galerkin Discretizations in Unsteady Inviscid Flow Problems," *AIAA Journal*, Vol. 48, No. 12, Dec. 2010, pp. 2867–2883. doi:10.2514/1.J050444

[7] Toulopoulos, I., and Ekaterinaris, J. A., "High-Order Discontinuous Galerkin Discretizations for Computational Aeroacoustics in Complex Domains," *AIAA Journal*, Vol. 44, No. 3, March 2006, pp. 502–511. doi:10.2514/1.11422

[8] Mascarenhas, B. S., Helenbrook, B. T., and Atkins, H. L., "Application of p-Multigrid to Discontinuous Galerkin Formulations of the Euler Equations," *AIAA Journal*, Vol. 47, No. 5, May 2009, pp. 1200–1208. doi:10.2514/1.39765

[9] Cockburn, B., and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 2440–2463. doi:10.1137/S0036142997316712

[10] Bassi, F., and Rebay, S., "Numerical Evaluation of Two Discontinuous Galerkin Methods for the Compressible Navier–Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 40, No. 1, Sept. 2002, pp. 197–207. doi:10.1002/fld.338

[11] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid Solution of High-Order Discontinuous Galerkin Discretizations of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 207, No. 1, Feb. 2005, pp. 92–113. doi:10.1016/j.jcp.2005.01.005

[12] Epshteyn, Y., and Rivière, B., "Estimation of Penalty Parameters for Symmetric Interior Penalty Galerkin Methods," *Journal of Computational and Applied Mathematics*, Vol. 206, No. 2, Sept. 2007, pp. 843–872. doi:10.1016/j.cam.2006.08.029

[13] Xia, Y., Luo, H., Norgaliev, R., and Cai, C., "A Class of Reconstructed Discontinuous Galerkin Methods for the Compressible Flows on

Arbitrary Grids," *49th AIAA Aerospace Sciences Meeting*, Orlando FL, AIAA Paper 2011-199, Jan. 2011.
[14] Shahbazi, K., "A Parallel High-Order Discontinuous Galerkin Solver for the Unsteady Incompressible Navier–Stokes Equations in Complex Geometries," Ph.D. Thesis, Univ. of Toronto, May 2007.
[15] Estep, D. J., "A Posteriori Error Bounds and Global Error Control for Approximation of Ordinary Differential Equations," *SIAM Journal on Numerical Analysis*, Vol. 32, No. 1, 1995, pp. 1–48. doi:10.1137/0732001
[16] Hartmann, R., and Houston, P., "Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations," *Journal of Computational Physics*, Vol. 183, No. 2, Dec. 2002, pp. 508–532. doi:10.1006/jcph.2002.7206
[17] Venditti, D. A., and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, No. 1, May 2003, pp. 22–46. doi:10.1016/S0021-9991(03)00074-3
[18] Wang, L., and Mavriplis, D. J., "Adjoint-Based h-p Adaptive Discontinuous Galerkin Methods for the 2D Euler Equations," *Journal of Computational Physics*, Vol. 228, No. 20, Nov. 2009, pp. 7643–7661. doi:10.1016/j.jcp.2009.07.012
[19] Mani, K., "Error Estimation and Adaptation for Functional Outputs in Time-Dependent Flow Problems," *Journal of Computational Physics*, Vol. 229, No. 2, Jan. 2010, pp. 415–440. doi:10.1016/j.jcp.2009.09.034
[20] Oliver, T. A., and Darmofal, D. L., "An Unsteady Adaptation Algorithm for Discontinuous Galerkin Discretizations of the RANS Equations," *18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, AIAA Paper 2007-3940, June 2007.
[21] Oliver, T. A., "A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds Averaged Navier–Stokes Equations," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, Sept. 2008.
[22] Hartmann, R., Held, J., and Leicht, T., "Adjoint-Based Error Estimation and Adaptive Mesh Refinement for the RANS and k–ω Turbulence Model Equations," *Journal of Computational Physics*, Vol. 230, No. 11, May 2011, pp. 4268–4284. doi:10.1016/j.jcp.2010.10.026
[23] Oliver, T. A., and Darmofal, D. L., "Analysis of Dual Consistency for Discontinuous Galerkin Discretizations of Source Terms," *SIAM Journal on Numerical Analysis*, Vol. 47, No. 5, 2009, pp. 3507. doi:10.1137/080721467
[24] Hartmann, R., and Houston, P., "An Optimal Order Interior Penalty Discontinuous Galerkin Discretization of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 227, No. 22, Nov. 2008, pp. 9670–9685. doi:10.1016/j.jcp.2008.07.015
[25] Hartmann, R., "Dual Consistency Analysis of Discontinuous Galerkin Discretizations," *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696. doi:10.1137/060665117
[26] Burgess, N. K., Nastase, C. R., and Mavriplis, D. J., "Efficient Solution Techniques for Discontinuous Galerkin Discretizations of the Navier–Stokes Equations on Hybrid Anisotropic Meshes," *48th AIAA Aerospace Sciences Meeting*, Orlando, FL, AIAA Paper 2010-1448, Jan. 2010.
[27] Nastase, C. R., and Mavriplis, D. J., "High-Order Discontinuous Galerkin Methods Using a Spectral Multigrid Approach," *43rd Aerospace Sciences Meeting and Exhibit*, Reno NV, AIAA Paper 2005-1268, Jan. 2005.
[28] Iacono, F., May, G., and Wang, Z., "Relaxation Techniques for High-Order Discretizations of Steady Compressible Inviscid Flows," *40th AIAA Fluid Dynamics Conference*, AIAA Paper 2010-4991, Chicago, June 2010.
[29] Shahbazi, K., Mavriplis, D. J., and Burgess, N. K., "Multigrid Algorithms for High-Order Discontinuous Galerkin Discretizations of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 228, No. 21, Nov. 2009, pp. 7917–7940. doi:10.1016/j.jcp.2009.07.013
[30] Haga, T., and Wang, Z., "Efficient Solution Techniques for High-Order Methods on 3-D Anisotropic Hybrid Meshes," *49th AIAA Aerospace Sciences Meeting*, Orlando FL, AIAA Paper 2011-45, Jan. 2011.
[31] Persson, P.-O., and Peraire, J., "Sub-Cell Shock Capturing for Discontinuous Galerkind Methods," *44th Aerospace Sciences Meeting and Exhibit*, Reno NV, AIAA Paper 2006-112, Jan. 2006.
[32] Davis, S. F., "Simplified Second-Order Godunov-Type Methods," *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, No. 3, 1988, pp. 445–473. doi:10.1137/0909030
[33] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5
[34] Burgess, N. K., "An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows," Ph.D. Thesis, University of Wyoming, Nov. 2011.
[35] Burgess, N. K., and Mavriplis, D. J., "An hp-Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows on Mixed-Element Meshes," *Proceeding of the 49th Aerospace Sciences Meeting*, AIAA, Orlando, FL, Jan. 2011;also AIAA Paper 2011-490.
[36] Wang, L., "Techniques for High-Order Adaptive Discontinuous Galerkin Discretizations in Fluid Dynamics," Ph.D. Thesis, Univ. of Wyoming, June 2009.
[37] Krivodonova, L., Xin, J., Remacle, J., Chevaugeon, N., and Flaherty, J., "Shock Detection and Limiting with Discontinuous Galerkin Methods for Hyperbolic Conservation Laws," *Applied Numerical Mathematics*, Vol. 48, Nos. 3–4, March 2004, pp. 323–338. doi:10.1016/j.apnum.2003.11.002

W. K. Anderson
Associate Editor