49th AIAA Aerospace Sciences Meeting and Exhibit, January 4–7, 2011, Orlando, FL

An *hp*-Adaptive Discontinuous Galerkin Solver for Aerodynamic flows on Mixed-Element Meshes

Nicholas K. Burgess *
and
Dimitri J. Mavriplis †
Department of Mechanical Engineering, University of Wyoming, Laramie WY 82071, USA

In this work we develop an *hp*-adaptive discontinuous Galerkin (DG) solver for aerodynamic flows. Previous work has focused on efficient solution techniques for discontinuous Galerkin discretizations. Recent work has focused on improving the robustness and efficiency of our discontinuous Galerkin solver for aerodynamic flows. Herein we propose an *hp*-adaptive approach 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 while in regions where the solution is smooth it will enrich the discretization order. This has two effects, first of all degrees of freedom are placed where they are needed thus addressing the efficiency of the method and second we avoid attempting to use high-order polynomials to capture solutions which are discontinuous, 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 three test cases all of which are steady state flows. Two of these are laminar viscous flows and one is an inviscid transonic flow. In addition the transonic flow has been computed using an artificial diffusion method and the *hp*-adaptive approach is compared to the artificial diffusion shock capturing method.

I. Introduction

High-order discontinuous Galerkin (DG) methods are receiving a great deal of attention for the computation of convection-dominated problems.[1–5] Recently there has been work done to extend these methods to convection-diffusion problems such as the Navier-Stokes equations.[6–13] In addition a posteriori error estimation for functional outputs is becoming a mature technique for estimating the contribution of discretization error to simulation outputs.[14–20] These a posteriori error estimates are based on solutions to the so called adjoint (dual) problem. These error estimates provide a method to guide adaptive refinement techniques to optimally place degrees of freedom within a problem. Recently these techniques have been used to perform adaptive mesh refinement (i.e. h-refinement only) of purely triangular meshes in the context of DG discretizations of the RANS equations.[17,18] Reference[19] has 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. References[18,21] have shown numerically that using dual-inconsistent discretizations can lead to sub-optimal convergence of the primal solution, while using dual consistent or asymptotically dual consistent discretizations leads to optimal convergence of the primal problem. Herein we use a modified symmetric interior penalty method (SIP) for the Navier-Stokes equations.

Discontinuous Galerkin (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. Non-smooth 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

*Ph.D. Candidate, Department of Mechanical Engineering, University of Wyoming, AIAA Student Member
†Professor, Department of Mechanical Engineering, University of Wyoming, AIAA Associate Fellow

in the literature. For example turbulence models used with the RANS equations often have non-smooth features associated with them. In our recent work we have noticed that the turbulence model of Spalart and Allmaras[22,23] has a discontinuity in its working variable at the interface between turbulent/non-turbulent regions of the flow. This discontinuity results in oscillations in the high-order solution, which can easily cause solver failure. This discontinuity is an example of a somewhat unexpected mode of solver failure, where a flow that one would initially assume is smooth in fact turns out to be non-smooth.

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 is what makes developing limiters for DG more difficult than for traditional CFD methods, and poses a rather serious drawback for computing discontinuous solutions with DG. While the addition of artificial viscosity can be used to treat non-smooth solutions, no theoretical minimum bound on the required amount of artificial viscosity 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 increased. However, artificial diffusion may have a place as a robustness enhancement method in combination with some type of adaptation.

While for non-adaptive techniques the coupling of discretization order and number of degrees of freedom poses a rather serious problem for limiting the solution as shown numerically in reference,[13] it 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 not appropriate (e.g. near discontinuities). This results in a minimum amount of adhoc parameters to tune, in fact there is only one which is required by the proposed *hp*-adaptive method. Further 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.

While a large amount of work has been conducted to create efficient solvers for discontinuous Galerkin(DG) discretizations of the Euler and Navier-Stokes equations,[3,12,13,24] 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 utilized while in regions where the solution is not smooth *h*-refinement is employed.

II. Governing Equations

The conservative form of the compressible Navier-Stokes equations describing the conservation of mass, momentum and total energy in two dimensions are given as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \left(\vec{\mathbf{F}}_c(\mathbf{u}) - \vec{\mathbf{F}}_v(\mathbf{u}, \nabla \mathbf{u}) \right) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) \tag{1}$$

subject to the appropriate boundary and initial conditions within a domain Ω.

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{Bmatrix}, \quad \mathbf{F_c}^x = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E_t + p) \end{Bmatrix}, \quad \mathbf{F_c}^y = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E_t + p) \end{Bmatrix},$$

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

$$\mathbf{S} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

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 form (with $\mathbf{x} = (x,y)$):

$$\begin{aligned} \tau_{ij} &= 2(\mu) S_{ij} \\ S_{ij} &= \frac{1}{2}\left(\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{u}_j}{\partial \mathbf{x}_i}\right) - \frac{1}{3}\frac{\partial \mathbf{u}_k}{\partial \mathbf{x}_k}\delta_{ij} \\ &\text{for } i = 1..2, j = 1..2 \end{aligned} \tag{3}$$

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

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

where $\gamma = 1.4$ is the ratio of specific heats. We have included a source term in our governing equations for generality, to include situations where there might be source terms involved. For example the turbulence models used with the RANS equations would have source terms. Throughout this work the source term will be zero.

III. Spatial Discretization

Before we proceed with the details it is important to explain the notation used in deriving the discretization. First a **bold face** symbol denotes a vector in the number of fields (i.e. the number of partial differential equations) while $\vec{()}$ denotes a vector in d spatial dimensions (herein $d = 2$). In addition a matrix will be denoted by $[\cdot]$. With regard to the discretization, given an exact solution u the corresponding discrete solution is denoted u_h, likewise for a continuous test function w the discrete test function is denoted by w_h. When deriving the discretization the test functions are written as vectors with same dimension as the solution, one must remember that while the inner products appear to give scalar equations the test function space has enough basis functions to give the proper number of equations (number of fields times number of modes on the element) on each element in the mesh. Again the source term (which is zero in this work) is included for the generality of the spatial discretization discussion.

The DG discretization is carried out by considering $\mathbf{u} \in \mathcal{V}$, where $\mathcal{V}$ is just the space which contains the exact solution. Let the computational domain Ω be partitioned into a set of non-overlapping 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 $\mathcal{V}_h^p \subset \mathcal{V}$. In addition 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 non-overlapping faces $\mathcal{B}_h$ with a single boundary face denoted b. 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

$$\sum_{k\in\mathcal{T}_h}\int_{\Omega_k} \mathbf{w}_h^T \frac{\partial \mathbf{u}_h}{\partial t} + \mathbf{w}_h^T \nabla\cdot\left[\vec{\mathbf{F}}_c(\mathbf{u}_h) - \vec{\mathbf{F}}_v(\mathbf{u}_h, \nabla\mathbf{u}_h)\right] - \mathbf{w}_h^T S(\mathbf{u}_h, \nabla\mathbf{u}_\mathbf{h})\, d\Omega_k = 0 \quad \forall \mathbf{w}_h \in \mathcal{V}_{h,p} \tag{5}$$

This can be written as

$$\sum_{k\in\mathcal{T}_h}\int_{\Omega_k} \mathbf{w}_h^T \frac{\partial \mathbf{u}_h}{\partial t} d\Omega_k + \mathbf{R}_h(\mathbf{w}_h, \mathbf{u}_h, \nabla\mathbf{u}_h) = 0, \quad \forall \mathbf{w}_h \in \mathcal{V}_{h,p} \tag{6}$$

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 and source terms in Eq. (5). The spatial residual is integrated by parts resulting in the following weak form

$$\begin{aligned} \mathbf{R}_h(\mathbf{w}_h, \mathbf{u}_h, \nabla\mathbf{u}_h) = &- \sum_{k\in\mathcal{T}_h}\int_{\Omega_k} \nabla\mathbf{w}_h^T \cdot \left[\vec{\mathbf{F}}_c(\mathbf{u}_h) - \vec{\mathbf{F}}_v(\mathbf{u}_h, \nabla\mathbf{u}_h)\right] - \mathbf{w}_h^T S(\mathbf{u}_h, \nabla\mathbf{u}_h)\, d\Omega_k \\ &+ \sum_{i\in I_h}\int_{\Gamma^i} \mathcal{H}_c(\mathbf{u}_h^+, \mathbf{u}_h^-, \vec{n})(\mathbf{w}_h^+ - \mathbf{w}_h^-) - \mathcal{H}_v(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{w}_h^+, \mathbf{w}_h^-, \nabla\mathbf{u}_h^+, \nabla\mathbf{u}_h^-, \vec{n}) ds \\ &+ \sum_{b\in\mathcal{B}_h}\int_{\Gamma^b} \mathcal{H}_c^b(\mathbf{u}_h^b(\mathbf{u}_h^+), \vec{n}) - \mathcal{H}_v^b(\mathbf{u}_h^b(\mathbf{u}_h^+), \mathbf{w}_h^+, \nabla\mathbf{u}_h^+, \vec{n}) ds \end{aligned} \tag{7}$$

where $\mathcal{H}_c(\cdot,\cdot,\vec{n})$ is the convective numerical flux and $\mathcal{H}_v(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,\vec{n})$ is the viscous numerical flux on the interior faces Γ^i. The numerical fluxes $\mathcal{H}_c^b(\cdot,\vec{n})$ and $\mathcal{H}_v^b(\cdot,\cdot,\cdot,\vec{n})$ denote boundary numerical fluxes (which are different from the interior numerical fluxes) on a boundary edge Γ^b. Reference[25] has shown 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,[26] Roe,[27] HLL[28] and HLLC.[29–31]

The numerical flux for the viscous term is obtained via a modified version of the symmetric interior penalty method (SIP) presented in references,[10–12,32] which seeks to penalize the solution for being discontinuous at the element interfaces. In fact the form of the SIP method used in this work is a hybridization of the one discussed in reference[12] and reference.[32] It is now convenient to introduce the following average and jump operators for both vector and scalar quantities. It is understood from the notation that for the jump of a vector we mean a vector in the physical space and not in the number of equations. The average operator is defined by

$$\begin{aligned} \{\phi\} &= \frac{1}{2}\left(\phi^+ + \phi^-\right) \\ \{\vec{\chi}\} &= \frac{1}{2}\left(\vec{\chi}^+ + \vec{\chi}^-\right) \end{aligned} \tag{8}$$

with the scalar and vector jump operators given by

$$\begin{aligned} [\![\varphi]\!] &= (\varphi^+ - \varphi^-)\vec{n} \\ [\![\vec{\chi}]\!] &= (\vec{\chi}^+ - \vec{\chi}^-)\cdot\vec{n} \end{aligned} \tag{9}$$

respectively. Note that the jump in a scalar quantity is a vector and the jump in a vector quantity is a scalar. Also note that a vector in fields is treated as a "scalar" by the jump operator and thus the jump of $\mathbf{u}_h$ is a matrix with the number of rows equal to the number of equations and the number of columns equal to the number of spatial dimensions in the domain. Using this notation the SIP numerical fluxes are given as

$$\begin{aligned} \mathcal{H}_v &= \{F_v(\mathbf{u}_h)\}\cdot[\![\mathbf{w}_h^T]\!] + \left\{[\mathbf{G}(\mathbf{u}_h)]^{T_{block}}\nabla\mathbf{w}_h\right\}\cdot[\![\mathbf{u}_h]\!] + \nu\left\{[\mathbf{G}(\mathbf{u}_h)]\right\}[\![\mathbf{u}_h]\!]\cdot[\![\mathbf{w}_h^T]\!] \\ \mathcal{H}_v^b &= F_v\left(\mathbf{u}_h^b\left(\mathbf{u}_h^+\right)\right)\cdot\vec{n}(\mathbf{w}_h^T)^+ + \left[\mathbf{G}\left(\mathbf{u}_h^b\left(\mathbf{u}_h^+\right)\right)\right]^{T_{block}}\nabla\mathbf{w}_h^+\cdot\vec{n}\left(\mathbf{u}_h^+ - \mathbf{u}_h^b\left(\mathbf{u}_h^+\right)\right) \\ &\quad + \nu\left[\mathbf{G}\left(\mathbf{u}_h^b\left(\mathbf{u}_h^+\right)\right)\right]\left(\mathbf{u}_h^+ - \mathbf{u}_h^b\left(\mathbf{u}_h^+\right)\right)\vec{n}\cdot(\mathbf{w}_h^T)^+\vec{n} \end{aligned} \tag{10}$$

where the matrix $[\mathbf{G}]$ is actually a block matrix with *dxd* blocks and with each block having *nxn* dimensions, where n is the number of equations (or fields). The $[\cdot]^{T_{block}}$ indicates a transposing of the blocks of $[\mathbf{G}]$. The blocks of $[\mathbf{G}]$ are defined such that

$$\begin{aligned} \mathbf{F_v}^x &= [G_{11}]\frac{\partial\mathbf{u}_h}{\partial x} + [G_{12}]\frac{\partial\mathbf{u}_h}{\partial y} \\ \mathbf{F_v}^y &= [G_{21}]\frac{\partial\mathbf{u}_h}{\partial x} + [G_{22}]\frac{\partial\mathbf{u}_h}{\partial y} \end{aligned} \tag{11}$$

The final piece of the discretization is the approximation of $\mathbf{u}\in\mathcal{V}$ by $\mathbf{u}_h\in\mathcal{V}_h^p$. In particular $\mathcal{V}_h^p$ is the space spanned by the polynomials $\{\phi_i, i=1..M\}$ where M is the number of modes required to specify a complete basis of order p. The set of functions we choose are Legendre polynomial based bubble functions $\phi_i\in L^2$ from reference.[33] The discrete solution $\mathbf{u}_h$ is given as a sum of coefficients times the basis functions

$$\mathbf{u}_h \equiv \mathbf{u}_h(\mathbf{x},t) = \sum_{j=1}^{M}\hat{\mathbf{u}}_j(t)\phi_j(\mathbf{x}) \tag{12}$$

where M is the number of modes defining the truncation level. The semi-discrete discontinuous Galerkin formulation (*i.e.* continuous in time) is given by choosing $\mathbf{w}_h$ to be each $\phi_i, \forall i = 1...M$, resulting in $nf*M$ equations for each element $k\in\mathcal{T}_h$ where nf is the number of equations (fields) in the system of Navier-Stokes equations.

The choice of penalty parameter can be rather adhoc as it is only required to be "large enough" to stabilize the scheme. However, Shahbazi in reference[10] derived an explicit expression for the penalty parameter for Poisson's

equation. A modified version of this expression given in reference[12] has been successfully implemented for this work. The value of the penalty parameter on an interface is

$$\nu = max\left(M^+, M^-\right) max\left(\frac{|\partial\Omega_k^-|}{|\Omega_k^-|}, \frac{|\partial\Omega_k^+|}{|\Omega_k^+|}\right) \tag{13}$$

where $|\Omega_k^\pm|$ and $|\partial\Omega_k^\pm|$ are the area and perimeter of elements $k^\pm$ respectively and where the $(\quad)^\pm$ denotes the elements on each side on the interface.

In this work the curved boundaries are mapped supper parametrically to order $p_{max}+1$ where p_{max} is the maximum order in the grid. The set of discrete equations is solved in modal space and the integrals are evaluated using Gaussian quadrature rules .[33–35] These quadrature rules require projection of the solution from the modal space to the quadrature points. To preserve the $p+1$ accuracy of the DG scheme the volume integrals are computed using a rule that integrates a polynomial of degree $2p$ exactly. To the same end surface integrations are carried out with a rule that integrates a polynomial of degree $2p+1$ exactly.[36]

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.

IV.A. Formulation

The following formulation is based on the approach described in reference.[16] Consider the functional of interest $\mathcal{L}(\mathbf{u})$ evaluated with the discrete flow-field 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 $\mathbf{u}_H^h$.

$$\mathcal{L}_h(\mathbf{u}_h) = \mathcal{L}_h(\mathbf{u}_H^h) + \left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)_{\mathbf{u}_H^h}(\mathbf{u}_h - \mathbf{u}_H^h) + \cdots \tag{14}$$

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 $\left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)_{\mathbf{u}_H^h}$ 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. (7) can also be expanded about the projected solution

$$\mathbf{R}_h(\mathbf{u}_h) = \mathbf{R}_h(\mathbf{u}_H^h) + \left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}(\mathbf{u}_h - \mathbf{u}_H^h) + \cdots \tag{15}$$

The fine level residual is constrained to be zero which allows one to re-arrange Eq. (15) to solve for the quantity involving the unknown fine solution as

$$(\mathbf{u}_h - \mathbf{u}_H^h) \approx -\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}^{-1}\mathbf{R}_h(\mathbf{u}_H^h) \tag{16}$$

Upon substitution of Eq. (16) into Eq. (14) one obtains the following expression for the estimate of the error in the functional

$$\mathcal{L}_h(\mathbf{u}_h) - \mathcal{L}_h(\mathbf{u}_H^h) \approx -\left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}^{-1}\mathbf{R}_h(\mathbf{u}_H^h) \tag{17}$$

where the flow residual on the fine mesh $\mathbf{R}_h(\mathbf{u}_H^h)$ are non-zero since 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

$$\left[\frac{\partial\mathbf{R}_h}{\partial\mathbf{u}_h}\right]_{\mathbf{u}_H^h}^T\Lambda_h = \left(\frac{\partial\mathcal{L}_h}{\partial\mathbf{u}_h}\right)^T \tag{18}$$

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

$$\mathcal{L}_h(\mathbf{u}_h) - \mathcal{L}_h(\mathbf{u}_H^h) \approx -(\Lambda_h)^T \mathbf{R}_h(\mathbf{u}_H^h) \tag{19}$$

The adjoint solution should be expected to cost as much as the flow solution and thus we do not want to compute the fine grid adjoint directly. Therefore the coarse adjoint solution is obtained via

$$\left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}\right]^T_{\mathbf{u}_H} \Lambda_H = \left(\frac{\partial \mathcal{L}_H}{\partial \mathbf{u}_H}\right)^T \tag{20}$$

which is solved using an hp-multgrid solver or an hp-multgrid preconditioned GMRES solver[3, 12, 13] on the coarse mesh. The coarse adjoint is then projected onto the fine mesh by injection and a few smoothing ($<= 5$) operations on the fine mesh. Reference[19] has 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 non-conforming meshes. Furthermore, several smoothing 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

$$\mathcal{L}_h(\mathbf{u}_h) - \mathcal{L}_h(\mathbf{u}_H^h) \approx -\underbrace{(\Lambda_H^h)^T \mathbf{R}_h(\mathbf{u}_H^h)}_{\varepsilon_c} \tag{21}$$

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 reference,[20] 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 non-uniform 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. hp-Adaptation

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. Reference[17, 18] has developed an unsteady mesh adaptation procedure for turbulent flows in 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. References[19, 37] have 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 reference[19, 37] to viscous flows on mixed-element meshes, 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.

V.A. h-Refinement

For viscous flows the authors have found it very advantageous to use quadrilateral elements in the highly stretched regions of the mesh.[13] As a result the refinement process becomes more complicated than in the simpler case of purely conforming triangular meshes.[19, 37] For meshes containing quadrilateral elements it is convenient to allow for non-conforming interfaces (i.e. hanging nodes) in the mesh. Therefore the triangles are now also refined such that they can have non-conforming interfaces. Refinement of both element types is done on a four to one basis with no more than a two to one discrepancy between the size of neighboring elements. Furthermore, while other techniques smooth their refined meshes after they are generated, we choose not to do this because it can corrupt the structure of the anisotropic boundary layer mesh.

The refinement pattern for triangles is depicted in Figure 1. The triangle is refined using mid-point subdivision where a node is inserted at the mid-point of each edge on the triangle. This results in four children (4:1) for each

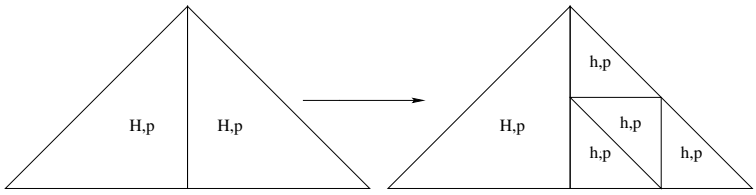


Figure 1. Illustration of the triangle *h*-refinement pattern.

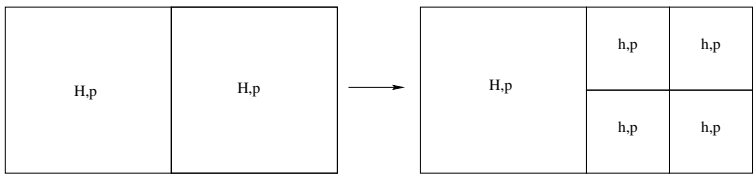


Figure 2. Illustration of the quadrilateral *h*-refinement pattern.

subdivided element. The quadrilaterals are refined in an analogous manner as depicted in Figure 2 with the exception that an additional node is placed at the center of the refined quadrilateral. The present method allows for the presence of hanging nodes for both triangular and quadrilateral elements as shown in Figures 1 and 2.

In an attempt to enforce a no more than 4:1 refinement rule any element with all but one of its edges flagged for refinement will have its final edge refined. If an an element has an edge marked for refinement and that edge is connected to a hanging node then the element is flagged for a full refinement.

The presence of hanging nodes complicates the inter-element surface integral compared to a conforming mesh. In our approach an edge connected by any two nodes is defined as a unique edge in the mesh. Thus triangles with a hanging node actually have four edges (similarity quadrilaterals can have up to six edges). The surface integral between non-conforming elements where one has a hanging node is accomplished by computing each edge integral separately and then pushing their contributions back into the elements on each side of the edges. While for the element with the hanging node the two edges that surround the hanging node have unique identification numbers in physical space, they have the same local edge number on the non-conforming element in the transformed space. This is to say that the edges surrounding a hanging node make up equal portions of a single edge in the transformed space. Furthermore, while for conforming edges the edge integral is the done over the full edge in transformed space, the edge integral for the non-conforming element is done over a fraction (usually 1/2) of the edge in transformed space. This results in a deformation and translation of the quadrature point locations on each side of an edge. The volume integrals are unaffected by the presence of hanging nodes.

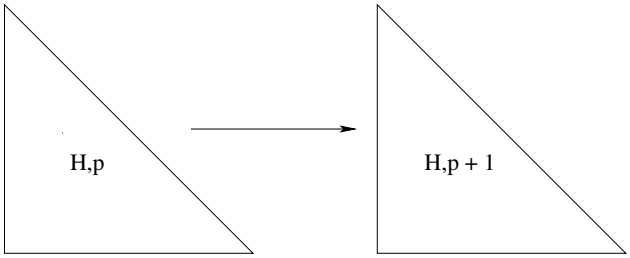
V.B. *p*-Enrichment

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. It is implemented without regard to element type as depicted in Figures 3(a) and 3(b), and a jump of no more than one order is permitted between elements.

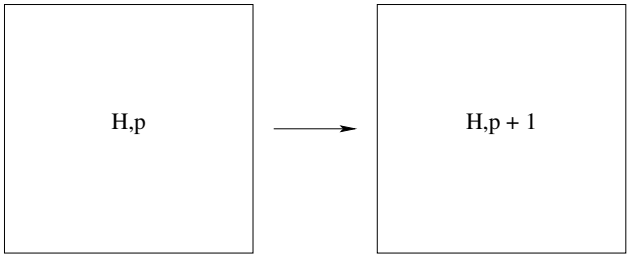
While *p*-enrichment induces no additional geometrical complexity, one does need to address how many quadrature points must be used to integrate the fluxes along the edges. In previous work we have integrated edge fluxes to $2p+1$ accuracy. When using a grid with variable discretization order it is necessary to use an integration rule that integrates to $2(max(p^+,p^-))+1$ accuracy where p^+ and p^- denote the element order on each side of an edge. The solutions on either side of the edge are evaluated using the number of modes from each element sharing the edge. The volume integrals remain unaffected by the variable discretization order throughout the mesh.

V.C. *hp*-Adaptation

It is well known that using high-order polynomials in the vicinity of discontinuities results in Gibbs phenomena, which for the Navier-Stokes/Euler equations can cause solver failure. Thus we propose to combine *h*-refinement and



(a) Triangle



(b) Quadrilateral

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

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 $p = 0$ discretization while using high-order polynomials in smooth flow regions where it is appropriate to do so.

The 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 reference.[38] The value of the jump indicator for an element is given by

$$(\tilde{s}_k)_f = \frac{1}{|\partial\Omega_k|} \int_{\partial\Omega_k} \left| \frac{[\![q]\!] \cdot \vec{n}}{\{q\}} \right| ds \tag{22}$$

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

$$\tilde{s}_k = \max_{f=1..nf} ((\tilde{s}_k)_f) \tag{23}$$

The above 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

$$\begin{cases} \tilde{s_k} > \frac{1}{\kappa}, & h\text{-refinement} \\ \tilde{s_k} < \frac{1}{\kappa}, & p\text{-enrichment} \end{cases} \tag{24}$$

where as in reference[19] $\kappa = 25$ is used throughout this work. In addition to selecting the adaptation mode based on the solution smoothness a maximum order in also enforced. When a cell reaches its maximum 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, it can also be viewed as a technique to enhance the robustness of the DG solver. In essence it seeks to design the mesh based on the solution, which for cases of under-resolved phenomena such as those encountered in reference[13] should result in a mesh of sufficient resolution such that high-order polynomials can be used throughout. For flow features that will most likely remain under-resolved 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 avoid Gibbs phenomena, providing a natural way for the present DG solver to begin handing non-smooth solutions robustly.

VI. The Implicit Steady State Solver

By neglecting the temporal derivative the system of equations becomes

$$\mathbf{R}_h(\mathbf{u}_h) = 0 \tag{25}$$

where $\mathbf{R}_h(\mathbf{u}_h)$ is the non-linear residual including source terms. This set of non-linear equations is solved using Newton's method:

$$\begin{aligned} \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}\right]^n \Delta \mathbf{u}_h^{n+1} &= -\mathbf{R}_h(\mathbf{u}_h^n) \\ \mathbf{u}_h^{n+1} &= \mathbf{u}_h^n + \Delta \mathbf{u}_h^{n+1} \end{aligned} \tag{26}$$

It is well known that 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:

$$\begin{aligned} \left[\frac{[I]}{\Delta t(n)} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}_p}\right]^n \Delta \mathbf{U}_p^{n+1} &= -\mathbf{R}(\mathbf{U}_p^n) \\ \mathbf{U}_p^{n+1} &= \mathbf{U}_p^n + \Delta \mathbf{U}_p^{n+1} \end{aligned} \tag{27}$$

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

$$\begin{aligned} CFL &= min\left(CFL_{min}\left(CFL_s\right)^n, CFL_{max}\right) \\ \Delta t(n)_k &= \frac{CFL|\Omega_k|}{|\partial\Omega_k|(|\vec{u}_k| + a_k)} \end{aligned} \tag{28}$$

where the CFL is the Courant-Friedrichs-Lewy number and a is the sound speed. Due to 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 iterative solver takes the form of either a linear hp-multigrid method or a multgrid preconditioned GMRES(MGPC-GMRES) method,[12,13] both of which use an under relaxed line-implicit Jacobi smoother. See reference[12,13] for details of the solver and a review of its performance.

VII. Numerical Results

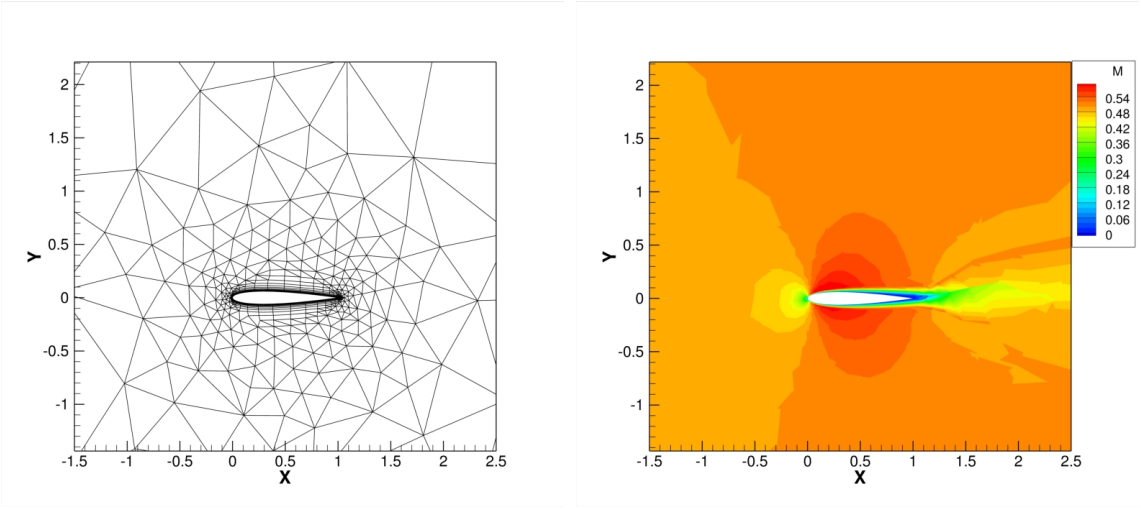
The proposed hp-adaptive method has been evaluated using three test cases. The first two test cases are the laminar viscous flow over a NACA0012 and a two element airfoil. The first test case is presented to compare hp-adaptation with h-refinement as well as with both uniform h-refinement and uniform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

Where appropriate these grids may contain both triangles and quadrilaterals within the same grid. The adaptive algorithm addresses this by allowing for non-conforming 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 .5% from one adaptation step to the next. The performance of the method is measured by considering how many degrees of freedom (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 vs wall time (i.e. CPU-time or computational time).

All results have been computed using the Riemann solver of Roe[27] on the cell interfaces. The equations are integrated implicitly using Newton's method with an exact linearization. The linear solver employed within each Newton iteration is the multgrid preconditioned GMRES (MGPC-GMRES) solver of references.[12,13] All test cases are steady state solutions and the flow and adjoint equations have been converged such that their residuals have been reduced by 13 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 intermediates steps before moving on to the next adaptive cycle.

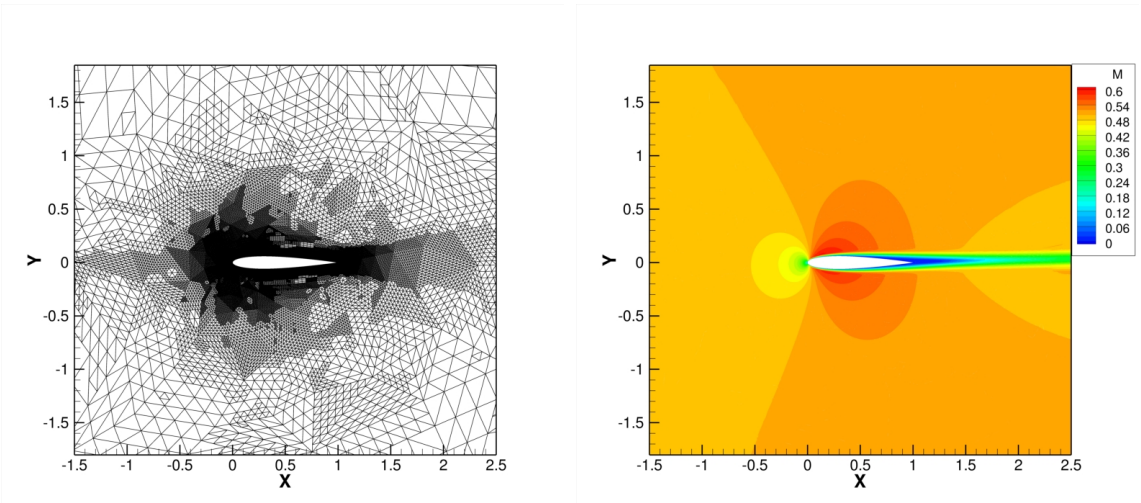
VII.A. NACA0012: Drag based Adaptation

The first test case consists of the laminar flow over a NACA0012 airfoil. The flow conditions are $M_\infty = .5$, $\alpha = 1^o$, and $Re = 5,000$. The initial grid contains 1,148 elements, with a uniform discretization order of $p = 1$ resulting in 3,930 DoFs. This case was 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 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 how smooth the solution is within that cell. The value of $p = 5$ was chosen as the grid maximum based on our experience with uniform p-enrichment. This example is shown to illustrate the high efficiency of the hp-adaptive approach, i.e. it 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.



(a) Initial Grid: 1,148 elements, $p = 1$ (b) Mach Contours: Initial Grid

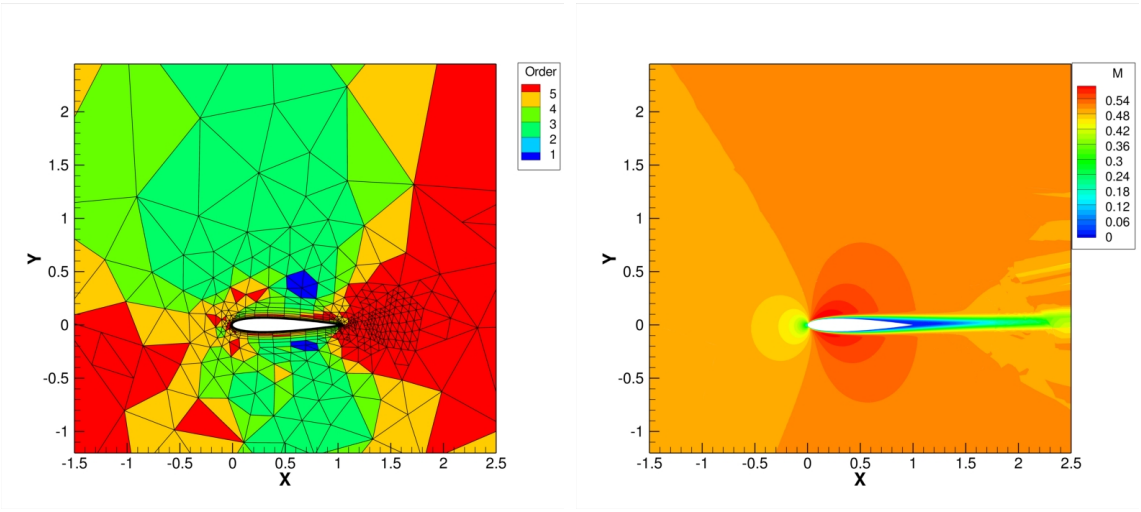
Figure 4. Initial mesh and Mach contours of the laminar flow over a NACA0012 airfoil with $p = 1$, $M_\infty = .5$, $\alpha = 1^o$, and $Re = 5,000$.



(a) Adjoint h-refinement: Final Grid with 174,145 elements, $p = 1$ (b) Adjoint h-refinement: Mach Contours on the Final Grid

Figure 5. Final mesh and Mach contours of the laminar flow over a NACA0012 airfoil using adjoint h-adaptation with $p = 1$, $M_\infty = .5$, $\alpha = 1^o$, and $Re = 5,000$.

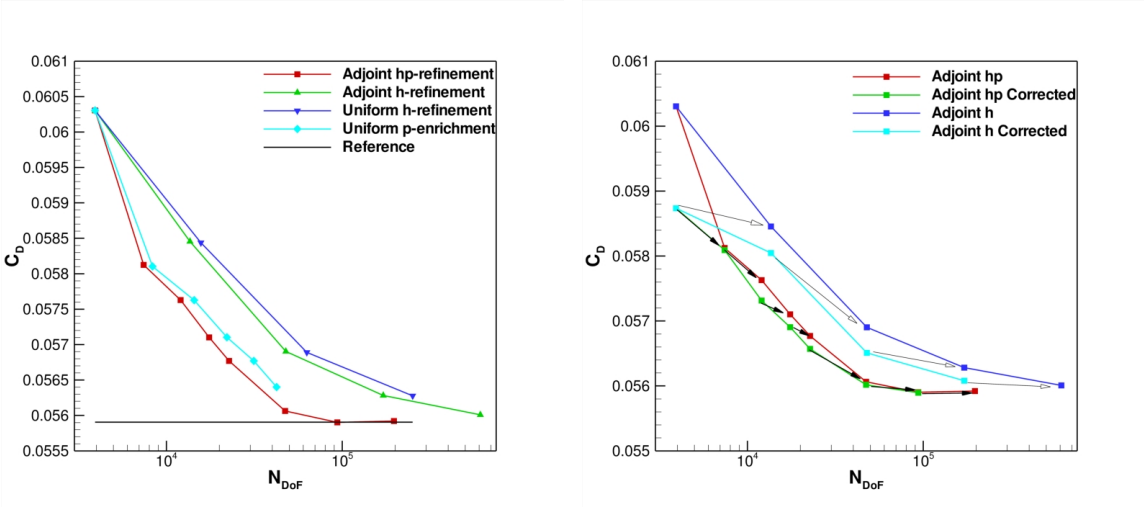
Figures 4(a)-6(b) depict the initial and final grids using both adjoint h-refinement and adjoint hp-adaptation as well as Mach contours on those grids. Note that the adjoint based strategies target both the surface of the airfoil as well as the wake region downstream from the trailing edge. Further note that both h-refinement and hp-adaptation Mach



(a) Adjoint hp-adaption: Final Grid with 6,776 elements, $p = 1$ to $p = 5$

(b) Adjoint hp-adaptation: Mach Contours on the Final Grid

Figure 6. Final mesh and Mach contours of the laminar flow over a NACA0012 airfoil using adjoint hp-adaptation with $p = 1$ to $p = 5$, $M_\infty = .5$, $\alpha = 1^o$, and $Re = 5,000$.



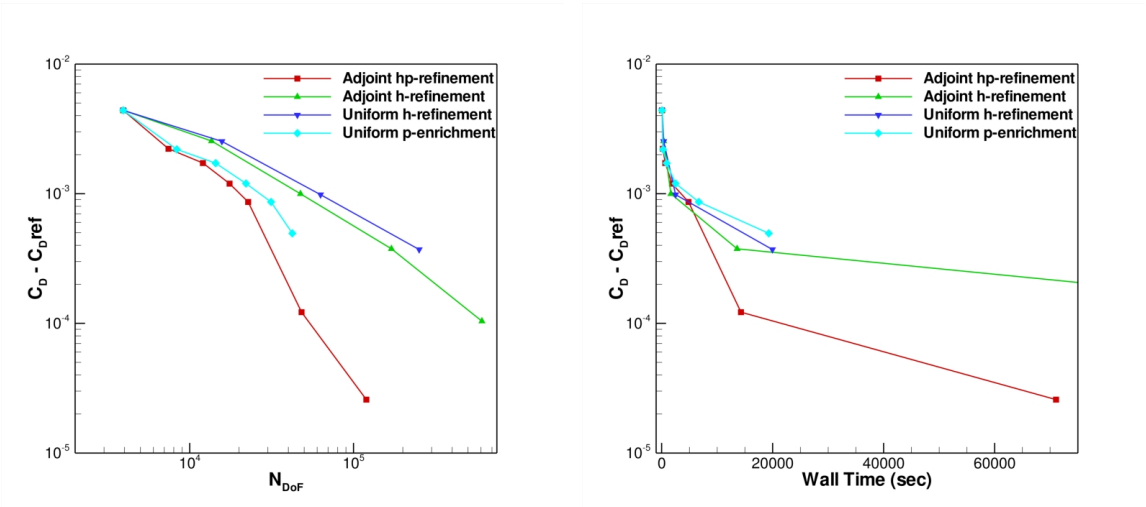
(a) Drag vs. NDoF

(b) Adjoint Corrected Drag vs. NDoF

Figure 7. Drag vs NDoF for the laminar flow over a NACA0012 airfoil using various adaptation methods, with and without adjoint corrections.

contours look very similar at the final stage, however examination of 7(a) shows that the hp-adaptation results contain about one third the number of DoFs compared to the h-refinement results.

Figure 7(a) depicts the computed drag vs. the number of DoFs using adjoint based h-refinement, adjoint based hp-adaptation, uniform h-refinement and uniform p-enrichment. The reference value depicted in Figure 7(a) was computed using the same DG solver with approximately 250,000 DoFs at a uniform discretization order of $p = 4$. Figure 7(a) 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 7(b) depicts the drag vs. the number of DoFs using the error predicted according to Eq. (21) to correct the coarse level drag, the arrows point from the coarse level corrected drag to the fine level drag it is attempting to predict. While initially the corrected coarse level drag seems quite poor as the refinement processes 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



(a) Drag Error vs. NDoF

(b) Drag Error vs. Wall Time

Figure 8. Drag Error for the laminar flow over a NACA0012 airfoil using various adaptation methods.

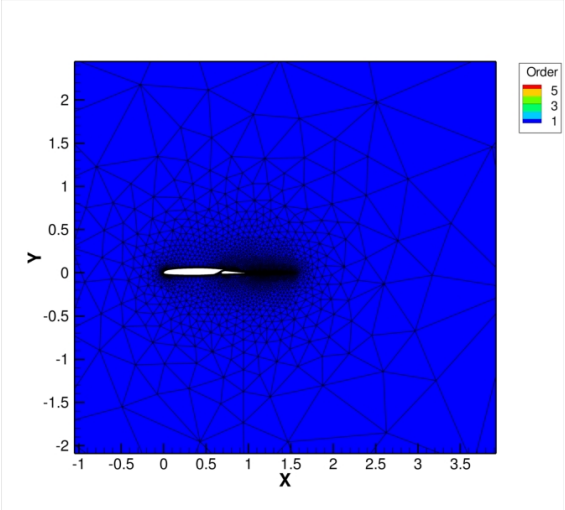
the error is predicted as a linear Taylor series expansion about the coarse level and thus as the drag becomes closer to grid converged the linear Taylor series becomes a better approximation of the behavior between coarse and fine level drag values. The adjoint *hp*-adaptive method clearly gives the 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 had become grid converged with far fewer DoFs.

Figures 8(a) and 8(b) show the drag error vs. the number of DoFs and the wall time. Figure 8(a) shows that the *hp*-adaptive method gives the lowest drag error per degree of freedom. The asymptotic slope of these error curves was computed for both the *h*-refinement and *hp*-adaptation cases. Theoretically for a uniform discretization order the drag should converge like $O(h^{2p})$. Therefore for the *h*-refinement curve in Figure 8(a) we expect that asymptotic slope to be 2 which it is. Computation of the asymptotic slope of the *hp*-adaptation 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 $p = 4$. Figure 8(b) shows the drag error vs. the wall time. Again we see that the *hp*-adaptation yields low error using a fraction of the wall time required by the *h*-refinement method. This is yet another measure of the efficiency of the *hp*-adaptation. Recall that the wall time shown for the adjoint based methods includes the adjoint solution as well as the flow solution.

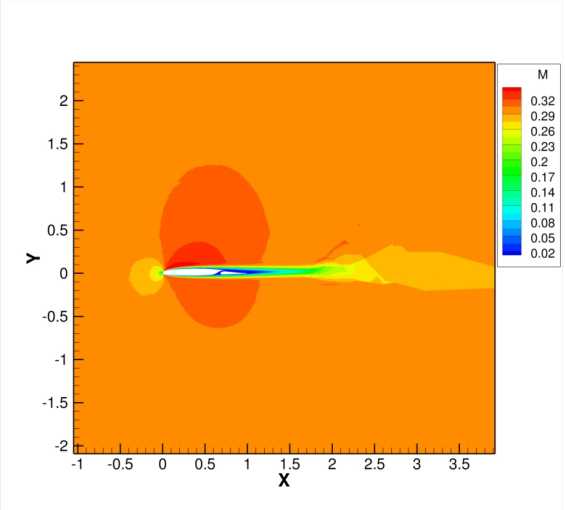
VII.B. Two-element Airfoil: Drag based Adaptation

The second test case is an example of applying the *hp*-adaptive method to a more complex geometry. This test consists of a two-element airfoil with the following flow conditions, $M_\infty = .3$, $\alpha = 1^o$, and $Re = 5,000$. The initial mesh consists of 6,921 elements. The grid is initialized to a uniform discretization order of $p = 1$ which results in 22,401 DoFs. Based on the results of the previous test case, this case is computed using only the *hp*-adaptive method where again the maximum discretization order is fixed at $p = 5$. Again we have chosen drag as the output functional for the adjoint based adaptation.

Figures 9(a)-10(b) depict the initial and final meshes along with the Mach number contours on each. Notice that there is a significant difference in the wake solutions between the initial and final meshes. Also notice that the *hp*-adaptation did not find it necessary to subdivide any elements because the initial mesh is relatively fine and the smoothness indicator did not detect any non-smooth phenomena, in this case changing the discretization order was sufficient to yield a grid converged result. The final mesh consists of the same number of cells as the original but with variable discretization order varying from $p = 1$ to $p = 5$ throughout the domain as depicted in Figure 10(a). Figure 11(a) shows the drag vs. NDoF for this case and clearly a grid converged drag value has been obtained after four adaptive steps using approximately 80,000 DoFs. Figure 11(b) shows the drag vs. wall time for this case, and a grid converged drag is generated in 53 min of computational time.

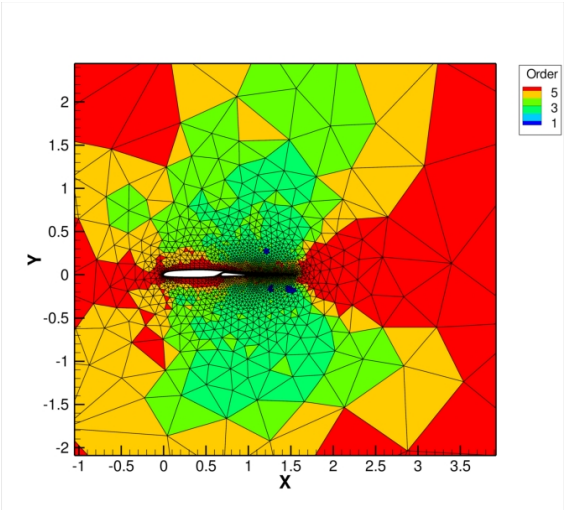


(a) Initial Grid: 6,921 elements, $p = 1$

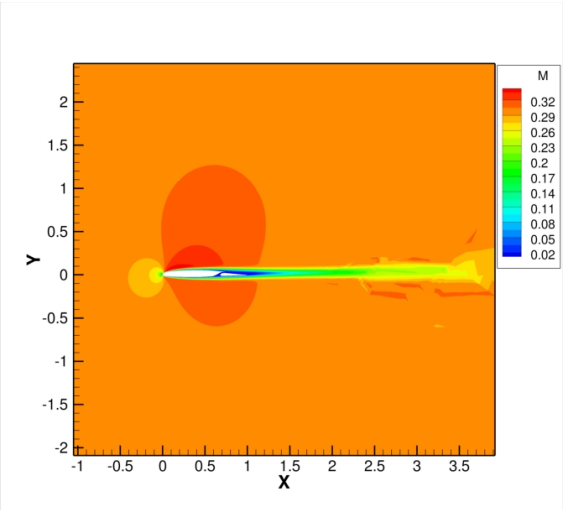


(b) Mach Contours: Initial Grid

Figure 9. Initial mesh and Mach contours of the laminar flow over a two-element airfoil with $p = 1$, $M_\infty = .3$, $\alpha = 1^o$, and $Re = 5,000$.



(a) Adjoint hp-adaption: Final Grid with 6,921 elements, $p = 1$ to $p = 5$



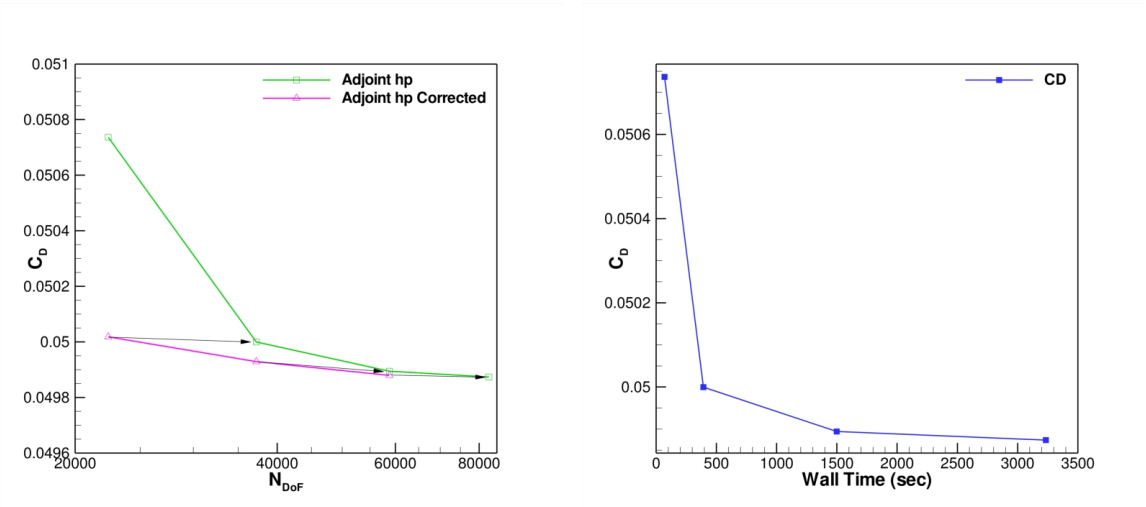
(b) Adjoint hp-adaptation: Mach Contours on the Final Grid

Figure 10. Final mesh and Mach contours of the laminar flow over a two-element airfoil using adjoint hp-adaptation with $p = 1$ to $p = 5$, $M_\infty = .3$, $\alpha = 1^o$, and $Re = 5,000$.

One should also note that the adjoint corrections give good results throughout the adaptive process. This is because the adaptive process was started on a much finer mesh than the NACA0012 test case and thus the process was started much closer to grid convergence than the previous case. The reason that the initial mesh contains so many cells is because the grid generator we have employed was unable to generate a mesh that was any coarser than the one shown.

VII.C. Transonic NACA0012: Lift based Adaptation

As a final example the hp-adaptive method is applied to a transonic flow around a NACA0012 airfoil. While the previous two test cases contained smooth flow solutions, this case has both a strong and a weak shock. This final case represents a scenario where the hp-adaptive approach not only yields high efficiency but it also yields enhanced robustness by allowing the solver to compute a flow it would otherwise be unable to compute without additional stabilization. The flow conditions for this case are $M_\infty = .8$ and $\alpha = 1.25^o$. The grid initially contains 1,566 triangles with a uniform discretization order of $p = 0$, resulting in 1,566 DoFs. The objective function in this case is lift. A reference solution was computed using a second order finite volume method with a 200,000 element mesh. Reference[19] com-

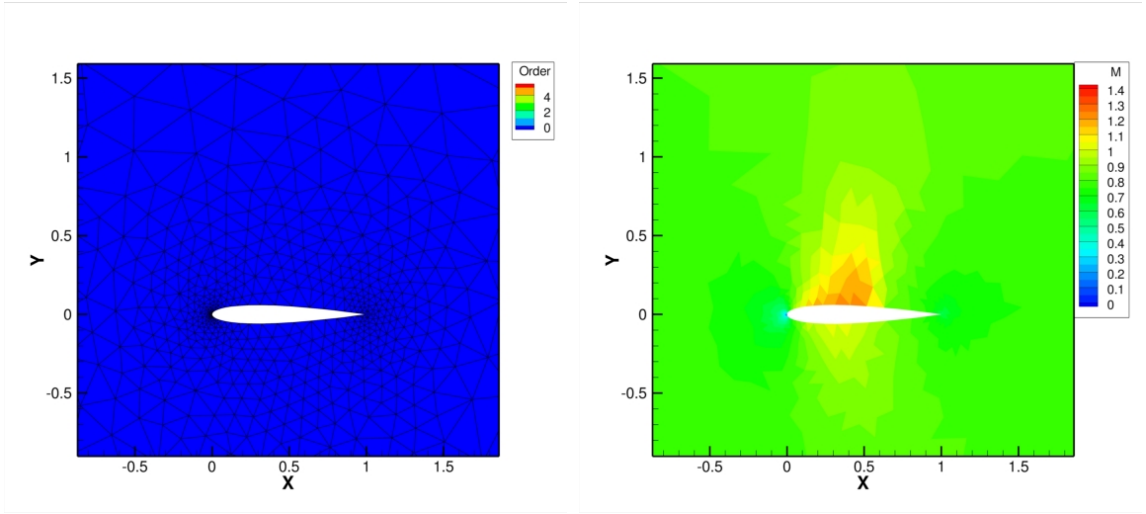


(a) Adjoint hp-adaptation: Drag vs NDoF

(b) Adjoint hp-adaptation: Drag vs Wall Time

Figure 11. Drag vs NDoF and vs Wall Time for the laminar flow over a two-element airfoil using hp-adaptation

puted a hypersonic case with an hp-adaptive method. This case is a significant departure from the case in reference[19] because here the shock touches the surface, thus some cells with $p = 0$ will be directly involved in the evaluation of the lift, which is not true of the case in reference.[19] We chose a transonic flow in order evaluate the method under these conditions. By initializing the grid with a discretization order of $p = 0$ the hp-adaptation is designed to maintain a $p = 0$ discretization near the shock waves and if refinement is required to use h-refinement in this region of the grid. At no point during the hp-adaptation is a limiter, artificial diffusion, or any form of explicit stabilization performed, instead we rely on the natural diffusion of the $p = 0$ discretization to ensure stability of the computation.

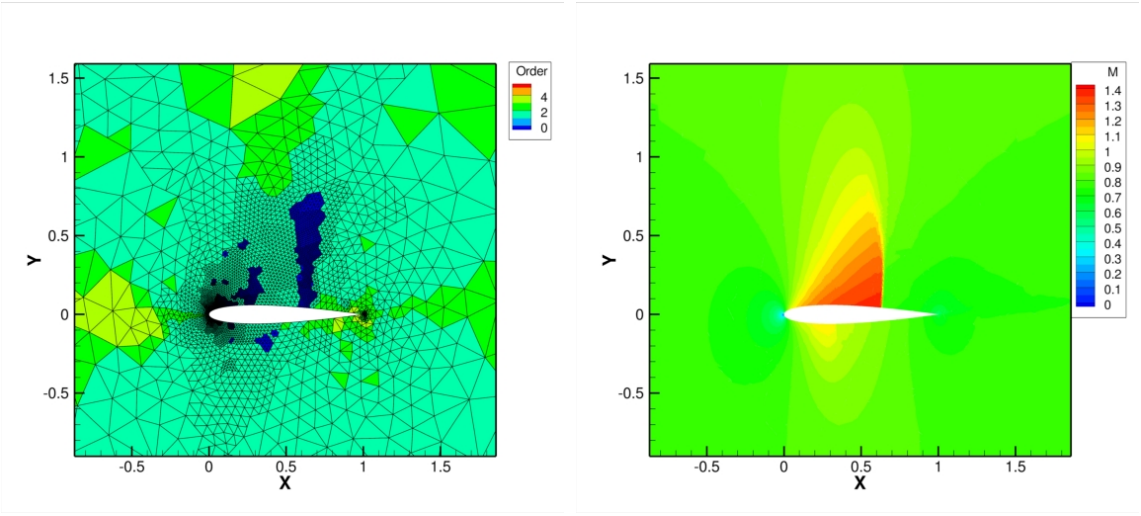


(a) Initial Grid: 1,556 elements, $p = 0$

(b) Mach Contours: Initial Grid

Figure 12. Initial mesh and Mach contours for the transonic flow over a NACA0012 airfoil with $p = 0$, $M_\infty = .8$ and $\alpha = 1.25^o$.

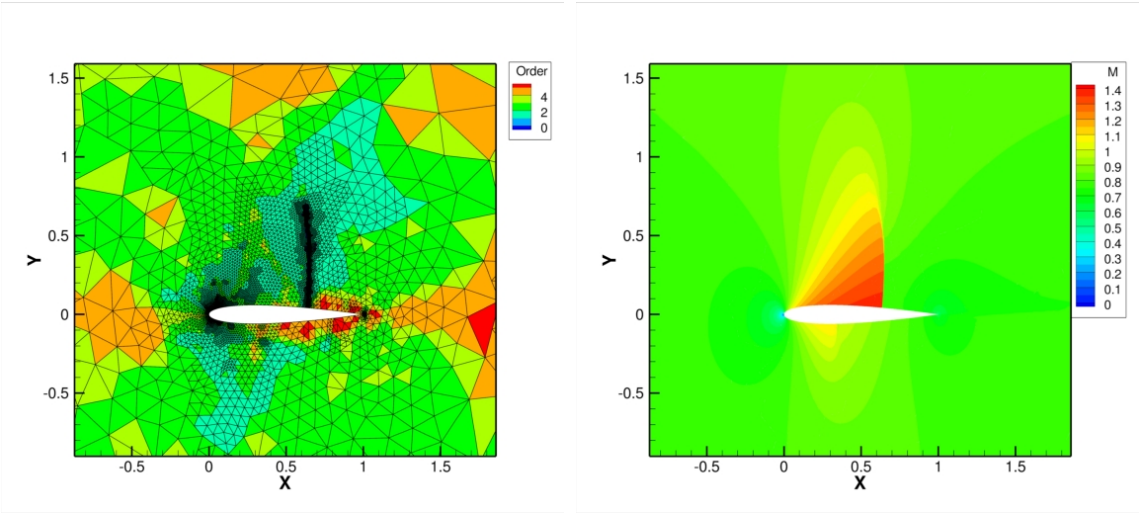
Figures 12(a)-14(b) depict the grids and Mach contours for this case at the initial, third, and final stages of the adaptive process. At the final adaptive stage we have a very sharply resolved shock free of oscillations. Figure 15(a) depicts the lift vs. NDoF and it is clear that a grid converged lift 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 15(b) depicts the iterative convergence for all adaptive cycles indicating that a fully converged solution is obtained at every stage of the adaptive process. Furthermore we see that it takes less than 100 Newton iterations in all cases, which shows how robust this method is. Furthermore even though there are $p = 0$ cells directly involved in the computation of the lift we still achieve nicely grid converged results with



(a) Adjoint hp-refinement: Intermediate Grid with 13,564 elements, $p = 1$ to $p = 3$

(b) Adjoint hp-refinement: Mach Contours on the Intermediate Grid

Figure 13. Intermediate mesh and Mach contours for the transonic flow over a NACA0012 ($M_\infty = .8$ and $\alpha = 1.25^o$) airfoil using adjoint hp-adaptation, the discretization order varies from $p = 1$ to $p = 3$.



(a) Adjoint hp-refinement: Final Grid with 26,407 elements, $p = 1$ to $p = 5$

(b) Adjoint hp-refinement: Mach Contours on the Final Grid

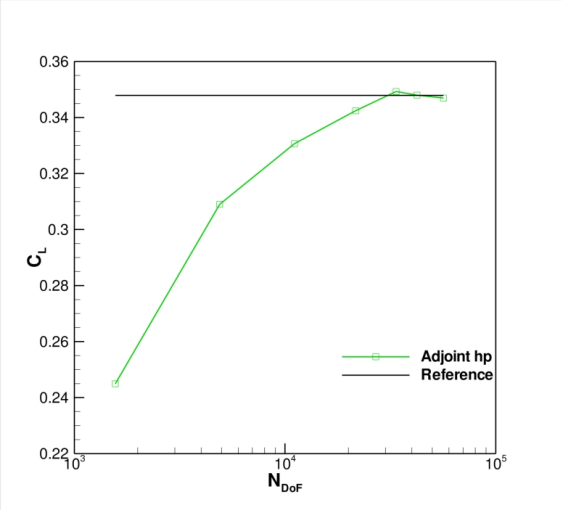
Figure 14. Final mesh and Mach contours for the transonic flow over a NACA0012 ($M_\infty = .8$ and $\alpha = 1.25^o$) airfoil using adjoint hp-adaptation, the discretization order varies from $p = 1$ to $p = 5$.

a very efficient algorithm.

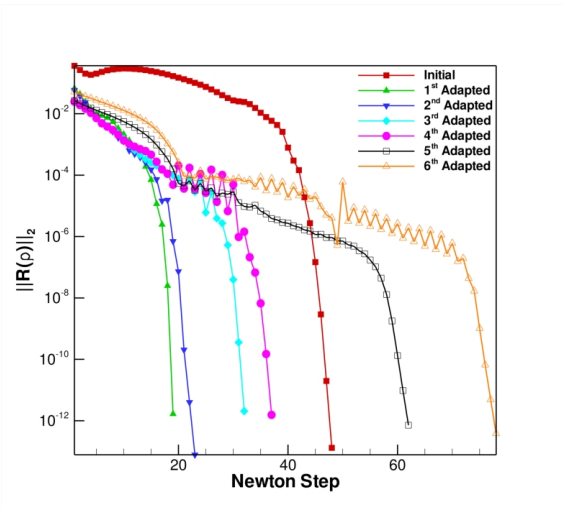
While no explicit limiter has been used to generate these results one can view the hp-adaptive approach as a form of limitation. Tradition slope limiters effectively reduce the order of accuracy locally. The idea behind slope limitation is to assume that high-order is appropriate everywhere in the grid and then to remedy those areas where high-order is not appropriate, which is a top down approach. The hp-adaptation can be viewed as a bottom up approach to limitation because it starts with low-order and moves towards high-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 is a modified version of the one found in reference.[39] The flow was computed using $p = 1$ to $p = 4$ on a triangular mesh with 3,189 elements, which corresponds to approximately 10,0000 to 50,000 DoFs.

The lift vs. NDoF is depicted in Figure 16(a), one can clearly see that with increasing order of accuracy the lift

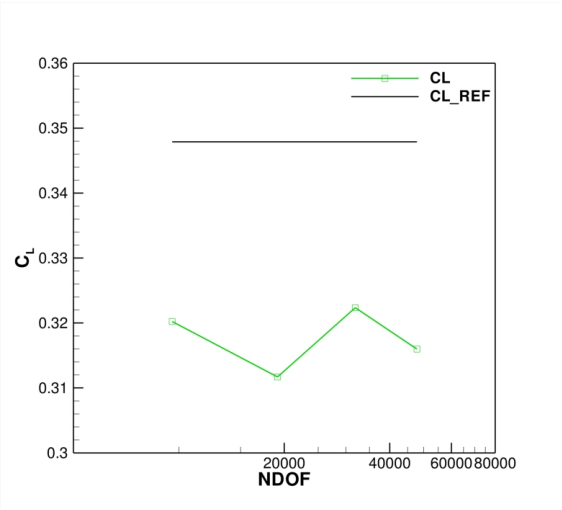


(a) Adjoint *hp*-adaptation: Lift vs. NDoF

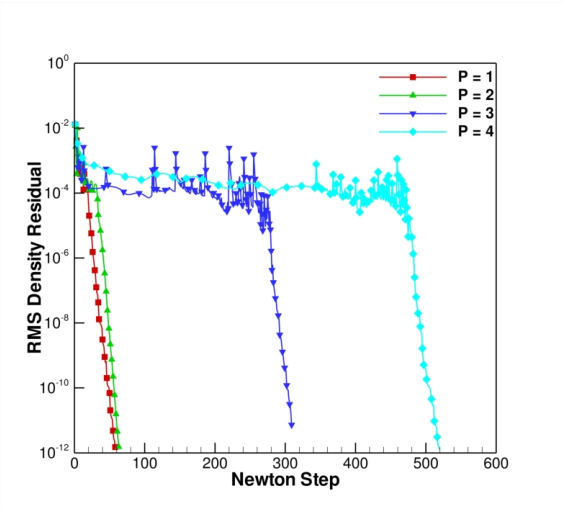


(b) Iterative Convergence

Figure 15. Transonic NACA0012 Lift vs NDoF using using *hp*-adaptation and iterative convergence



(a) Artificial Diffusion: Lift vs. NDoF



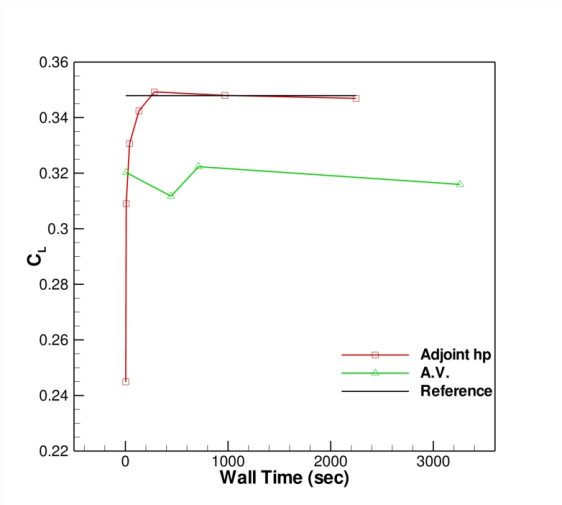
(b) Iterative Convergence

Figure 16. Transonic NACA0012 Lift vs NDoF using using artificial diffusion with $p = 1$ to $p = 4$ and iterative convergence

does not converge to a fixed value as uniform *p*-enrichment is applied. Figure 16(b) depicts the iterative converge for this case using the artificial diffusion. One can immediately see that computing shocks with this method can become quite expensive especially when compared with the adjoint *hp*-adaptation convergence history in Figure 15(b). Figure 17(a) depicts the lift vs. the computational time, which clearly shows that *hp*-adaptation generates more accurate lift values at a reduced cost compared to higher-order shock capturing with artificial diffusion. Additionally for each 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. While the *hp*-adaptive approach is not the most elegant shock capturing method for DG discretizations it has some significant advantages; it is easy to use (i.e. requires only 1 parameter to tune, which is the value of smoothness indicator that selects *h*-refinement vs. *p*-enrichment), gives robust and fast iterative convergence, and with each refinement the functional improves and eventually achieves a grid converged value.

VIII. Concluding Remarks and Future Work

An *hp*-adaptive high-order discontinuous Galerkin solver for the Navier-Stokes equations has been developed and applied to three test cases. The adaptive method presented is driven by a goal-oriented approach which makes use



(a) Lift vs. Wall Time

Figure 17. Comparison of Transonic NACA0012 Lift vs Wall Time using using artificial diffusion with $p = 1$ to $p = 4$ and hp-adaptation

of adjoint based error estimation. The adaptive algorithm is capable of adapting both the grid and order of accuracy locally. The solver adapts the grid non-conformally 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.

Future work will focus on investigating techniques for enhancing robustness that do not rely on hp-adaptation but could and probably should be used in conjunction with hp-adaptation. The techniques under investigation include filtering,[40] a new artificial viscosity method first developed in the context of the spectral difference method by references,[41,42] and a modification of the artificial viscosity method presented in.[39] In particular future work will focus on combining the technique of reference[39] with the hp-adaptation as well as removing some of the tunable parameters. Robustness enhancement along with hp-adaptation will be required for efficient and reliable computations of turbulent flows and future work will also focus on computing turbulent flows using hp-adaptation.

IX. Acknowledgments

The first author would like to acknowledge the contributions of Prof. Dimitri Mavriplis, Prof. Jay Sitaraman, Dr. Karthik Mani, and Mr. Brian Lockwood. This work was supported under AFOSR Grant FA9950-10-C-0051 and NSF Grant 0904936.

References

[1]Bassi, F. and Rebay, S., "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," *J. Comput. Phys.*, Vol. 138, 1997, pp. 251–285.

[2]Cockburn, B. and Shu, C.-W., "Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *SIAM J. Sci. Comput.*, Vol. 16, No. 3, 2001, pp. 173–261.

[3]Nastase, C. R. and Mavriplis, D. J., "High-Order discontinuous Galerkin Methods using an hp-muligrid approach," *J. Comput. Phys.*, Vol. 213, No. 1, mar 2006, pp. 330–357.

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

[5]Wang, L. and Mavriplis, D. J., "Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations," *J. Comput. Phys.*, Vol. 225, No. 2, Aug 2007, pp. 1994–2015.

[6]Bassi, F. and Rebay, S., "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations," *J. Comput. Phys.*, Vol. 131, No. 2, Mar 1997, pp. 267–279.

[7]Bassi, F. and Rebay, S., "Numerical Evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations," *Int. J. Numer. Meth. in Fluids*, Vol. 40, No. 1, Sept 2002, pp. 197–207.

[8]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," *J. Comput. Phys.*, Vol. 207, No. 1, Feb 2005, pp. 92–113.

[9]Diosady, L. T. and Darmofal, D. L., "Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations," *J. Comput. Phys.*, Vol. 228, No. 1, Mar 2009, pp. 3917–3935.

[10]Shahbazi, K., *A Parallel High-Order Discontinuous Galerkin Solver For The Unsteady Incompressible Navier-Stokes Equations in Complex Geometries*, Ph.D. thesis, University of Toronto, may 2007.

[11]Hartmann, R. and Houston, P., "Symmetric Interior Penalty DG Methods for the Compressible Navier-Stokes Equations I: Method Formulation," *Internal Journal of Numerical Analysis and Modeling*, Vol. 3, No. 1, 2006, pp. 1–20.

[12]Shahbazi, K., Mavriplis, D. J., and Burgess, N. K., "Multigrid Algorithms for High-Order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations," *J. Comput. Phys.*, Vol. 228, No. 21, Nov 2009, pp. 7917–7940.

[13]Burgess, N. K., Nastase, C. R., and Mavriplis, D. J., "Efficient Solution Techniques for Discontinuous Galerkind Discretizations of the Navier-Stokes Equations on Hybrid Anisotropic Meshes," *Proceeding of the 48th Aerospace Sciences Meeting, Orlando, FL*, Jan. 2010, AIAA Paper 2010-1448.

[14]Estep, D. J., "A posteriori error bounds and global error control for approximation of ordindary differential equations," *SIAM Journal of Numerical Analysis*, Vol. 32, No. 1, 1995, pp. 1–48.

[15]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.

[16]Venditti, D. A. and Darmofal, D. L., "Anisotropic grid adaption for functional outputs:application to two-dimensional viscous flows," *Journal of Computational Physics*, Vol. 187, No. 1, May 2003, pp. 22–46.

[17]Oliver, T. A. and Darmofal, D. L., "An Unsteady Adaptation Algorithm for Discontinuous Galerkin Discretizations of the RANS Equations," *Proceeding of the 18th AIAA CFD Conference, Miami, FL*, 2007, AIAA Paper 2007-3940.

[18]Oliver, T. A., *A high-order, adaptive, discontinuous Galerkin finit element method for the Reynolds averaged Navier-Stokes equations*, Ph.D. thesis, Massachusetts Institute of Technology, sept 2008.

[19]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.

[20]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.

[21]Oliver, T. A. and Darmofal, D. L., "Analysis of Dual Consistency for Discontinuous Galerkin Discretizations of Source Terms," *SIAM J. Numer. Anal.*, Vol. 47, No. 1, 2009, pp. 3507.

[22]Spalart, P. and Allmaras, S., "A one-equation turbulence model for Aerodynamic flows," *Le Recherche Aerospátiale*, Vol. 1, 1994, pp. 5–21.

[23]Oliver, T. A. and Darmofal, D. L., "Impact of Turbulence Model Irregularity on High-Order Discretizations," *Proceeding of the 47th Aerospace Sciences Meeting and Exhibit, Orlando FL*, Jan. 2009, AIAA Paper 2009-953.

[24]Nastase, C. R. and Mavriplis, D. J., "High-Order Discontinuous Galerkin Methods using a Spectral Multigrid Approach," *Proceedings of the 43rd Aerospace Sciences Meeting and Exihibit, Reno NV*, 2005, AIAA Paper 2005–1268.

[25]Hartmann, R., "Dual Consistency Analysis of Discontinuous Galerkin Discretizations," *SIAM Journal of Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.

[26]Davis, S. F., "Simplified Second-Order Godunov-Type Methods," *SIAM J. Sci. Statist. Comput.*, Vol. 9, No. 3, 1988, pp. 445–473.

[27]Roe, P. L., "Approximate Riemann Solvers, Parameter vectors, and Difference Schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.

[28]Harten, A., Lax, P. D., and Van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *SIAM Review*, Vol. 25, No. 1, 1983, pp. 35–61.

[29]Toro, F. E., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Applied Mechanics, Springer-Verlag, New York, NY, 1999.

[30]Batten, P., Clarke, N., Lambert, C., and Causon, D. M., "On the Choice of Wavespeeds for the HLLC Riemann Solver," *SIAM J. Sci. Comput.*, Vol. 18, No. 2, 1997, pp. 1553–1570.

[31]Batten, P., Leschiner, M. A., and Goldberg, U. C., "Average-State Jacobians and Implicit Methods for Compressible Viscous and Turbulent Flows," *J. Comput. Phys.*, Vol. 137, 1997, pp. 38–78.

[32]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.

[33]Solin, P., Segeth, P., and Zel, I., *High-Order Finite Element Methods*, Studies in Advanced Mathematics, Chapman and Hall, 2003.

[34]Dunavant, D. A., "Highr Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle," *Int. J. Numer. Meth. Engng.*, Vol. 21, 1985, pp. 1129–1148.

[35]Dunavant, D. A., "Economical Symmetrical Quadrature Rules for Complete Polynomials Over a Square Domain," *Int. J. Numer. Meth. Engng.*, Vol. 21, 1985, pp. 1777–1784.

[36]Cockburn, B., Hou, S., and Shu, C.-W., "The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case," *Math. Comput.*, Vol. 54, No. 545, 1990.

[37]Wang, L., *Techniques For High-Order Adaptive Discontinuous Galerkin Discretizations in Fluid Dynamics*, Ph.D. thesis, University of Wyoming, June 2009.

[38]Krivodonova, L., Xin, J., Remacle, J., Chevaugeon, N., and Flaherty, J., "Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws," *Appl. Numer. Math*, Vol. 48, No. 3-4, Mar 2004, pp. 323–338.

[39]Persson, P.-O. and Peraire, J., "Sub-Cell Shock Capturing for Discontinuous Galerkind Methods," *Proceedings of 44th Aerospace Sciences Meeting and Exihibit, Reno NV*, Jan. 2006, AIAA Paper 2006-112.

[40]Hesthaven, J. S. and Warburton, T., *Nodal Discontinuous Galerkin Methods*, Springer, New York, NY, 2008.

[41]Premasuthan, S., Liang, C., and Jameson, A., "A Spectral Difference method for viscous compressible flows with shocks," *Proceeding of the 19th AIAA Computational Fluid Dynamics Conference, San Antonio, Texax*, 2009, AIAA Paper 2009-3785.

[42]Premasuthan, S., Liang, C., and Jameson, A., "Computation of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity," *Proceeding of the 48th Aerospace Sciences Meeting, Orlando, FL*, 2010, AIAA Paper 2010-1449.