Journal of Computational Physics 228 (2009) 7643–7661

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Adjoint-based *h–p* adaptive discontinuous Galerkin methods for the 2D compressible Euler equations

Li Wang *, Dimitri J. Mavriplis

Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071-3295, United States

ARTICLE INFO

Article history:
Received 16 December 2008
Received in revised form 9 June 2009
Accepted 14 July 2009
Available online 25 July 2009

Keywords:
Discontinuous Galerkin methods
High-order methods
Adjoint-based error estimation
Output functional
Mesh refinement
hp-Multigrid approach
High speed flow
Compressible flow

ABSTRACT

In this paper, we investigate and present an adaptive Discontinuous Galerkin algorithm driven by an adjoint-based error estimation technique for the inviscid compressible Euler equations. This approach requires the numerical approximations for the flow (i.e. primal) problem and the adjoint (i.e. dual) problem which corresponds to a particular simulation objective output of interest. The convergence of these two problems is accelerated by an *hp*-multigrid solver which makes use of an element Gauss–Seidel smoother on each level of the multigrid sequence. The error estimation of the output functional results in a spatial error distribution, which is used to drive an adaptive refinement strategy, which may include local mesh subdivision (*h*-refinement), local modification of discretization orders (*p*-enrichment) and the combination of both approaches known as *hp*-refinement. The selection between *h*- and *p*-refinement in the *hp*-adaptation approach is made based on a smoothness indicator applied to the most recently available flow solution values. Numerical results for the inviscid compressible flow over an idealized four-element airfoil geometry demonstrate that both pure *h*-refinement and pure *p*-enrichment algorithms achieve equivalent error reductions at each adaptation cycle compared to a uniform refinement approach, but requiring fewer degrees of freedom. The proposed *hp*-adaptive refinement strategy is capable of obtaining exponential error convergence in terms of degrees of freedom, and results in significant savings in computational cost. A high-speed flow test case is used to demonstrate the ability of the *hp*-refinement approach for capturing strong shocks or discontinuities while improving functional accuracy.

1. Introduction

The use of high-order discontinuous Galerkin discretizations has become more widespread over the last decade for solving convection-dominated computational fluid dynamics problems [1–9]. The appeal of these methods relates to their favorable asymptotic accuracy properties, combined with compact stencils and favorable scalability properties on parallel computing architectures. Over the same period, practical formulations for goal-oriented a posteriori error estimation based on adjoint techniques have been developed and demonstrated successfully, initially within the context of a variational framework [10–15], and more recently for finite-volume discretizations [16–19]. The advantage of an objective-based adjoint error estimation approach is that it provides a well-founded technique for estimating the error in specific simulation outputs, which can then be used to guide adaptive refinement strategies for automatically producing accurate simulations at optimal cost with certifiable error bounds in the objective of interest.

* Corresponding author. Tel.: +1 307 399 6550; fax: +1 307 766 2695.
E-mail addresses: wangli@uwyo.edu (L. Wang), mavripl@uwyo.edu (D.J. Mavriplis).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.012

In this work, we combine high-order discontinuous Galerkin discretizations with adjoint error estimation techniques and adaptive refinement strategies to develop an efficient approach, which is capable of delivering a prescribed level of accuracy in specific engineering objectives of interest. We make use exclusively of a discrete adjoint formulation, and the objective error estimates are derived in a non-variational framework, although the resulting equations are identical to those obtained using a variational framework. One notable difference with some previous formulations is that we estimate only the change in the functional value between two successively refined or enriched mesh levels, as opposed to attempting to estimate the total error defined as the change between the current functional value and the value which would be obtained in the continuous limit [10,16]. The current approach, which has been demonstrated successfully for complex three-dimensional problems [19], is sufficient for adaptive mesh refinement purposes. In order to be competitive, the gains afforded by the reduced number of degrees of freedom for a given accuracy level achieved by the adaptive process, compared to a global refinement strategy, must outweigh the additional cost of the error estimation technique, as well as the cost of solving the flow and adjoint problem on multiple intermediate adaptively refined levels. To this end, the cost of the error estimation technique is designed to be equivalent to, or in most cases lower than, the cost of solving the analysis problem on the current refinement level. This is achieved by solving the adjoint problem on the same mesh as the analysis problem at each refinement level, and using reconstruction techniques to obtain a fine level adjoint field, as required in the error estimation formulation. Additionally, the adjoint problem is solved using the same efficient *h*–*p* multigrid algorithm used to solve the flow problem [7,9], thus guaranteeing similar convergence rates, since both problems contain the same eigenvalues, while the linear nature of the adjoint problem results in lower cost per iteration compared to the flow or analysis problem.

In the context of discontinuous Galerkin discretizations, accuracy enhancements can be achieved either by refining the mesh (h), or by raising the order of accuracy (p) of the discretization. In this work, we examine both *h*- and *p*-refinement approaches alone, as well as a combined *hp*-refinement approach, which has been shown to be optimal, in the sense that it has the potential to enable exponential error convergence [20]. The key to choosing between h and p lies in the ability to assess the local smoothness of the solution, and we make use of two previously described smoothness indicators for this purpose [1,8,21,22]. When properly implemented, the *hp*-adaptive refinement scheme has the potential to capture discontinuous phenomena such as strong shocks without the need for limiting, since the smoothness indicator acts to preserve low-order discretizations in such regions.

The outline of this paper is as follows. In Section 2 the governing equations are introduced and the Discontinuous Galerkin discretization is formulated. Section 3 describes key techniques for estimating the error in the functional with respect to its value on a globally refined or *p*-enriched fine mesh. This procedure avoids expensive computations of the flow and adjoint solutions on the fine mesh, while requiring a linear adjoint problem to be solved on the original coarse mesh. In Section 4, we employ various adaptive strategies to locally refine the coarse mesh in areas which most adversely influence the functional accuracy, as predicted by the aforementioned error estimation technique. Next, several numerical test cases are used to exhibit the performance of the proposed adaptive solution strategies in Section 5, including comparisons of overall accuracy and efficiency between the pure *h*-refinement, pure *p*-enrichment, and combined *hp*-refinement approaches. Additionally, the shock-capturing property of the *hp*-adaptive refinement strategy is demonstrated for a test case involving high speed flow over a half-circular cylinder. Finally, Section 6 summarizes the conclusions of this work.

2. Governing equations

The governing equations that we consider exclusively in this work are the two-dimensional steady-state compressible Euler equations in gas dynamics which can be written in the following conservative form:

$$\frac{\partial \mathbf{f}_1(\mathbf{u}(\mathbf{x}))}{\partial x} + \frac{\partial \mathbf{f}_2(\mathbf{u}(\mathbf{x}))}{\partial y} = 0 \quad \text{in } \Omega \tag{1}$$

where Ω is a two-dimensional bounded domain. Here, the vector of conservative flow variables $\mathbf{u}$ and the inviscid Cartesian flux components $\mathbf{f}_1$ and $\mathbf{f}_2$ are defined by

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{Bmatrix}, \quad \mathbf{f}_1 = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho e + p) u \end{Bmatrix}, \quad \mathbf{f}_2 = \begin{Bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (\rho e + p) v \end{Bmatrix} \tag{2}$$

respectively, where the notations ρ, p, and e denote the fluid density, pressure and specific total energy per unit mass, respectively. u and v represent the flow velocity components in the x and y coordinate directions. This system of equations is completed by the perfect gas equation of state given as,

$$p = (\gamma - 1)\left[\rho e - \frac{1}{2}\rho(u^2 + v^2)\right] \tag{3}$$

where γ is defined as the ratio of specific heats, which is 1.4 for air.

Consider a mesh, $\mathcal{T}_{h,p}$ (of average element size h and discretization order p), consisting with a sequence of non-overlapping partitions for the whole computational domain, Ω. The Discontinuous Galerkin formulation in weak form within each element $k \in \mathcal{T}_{h,p}$ for the steady-state system can be derived by multiplying the governing equations by a test function ϕ_j $(j = 1, \ldots, M)$, and then integrating over each element (e.g. k) as:

$$\mathbf{R}_{j,k}(\mathbf{u}_h) = -\int_k \left[\frac{\partial \phi_j}{\partial x}\mathbf{f}_1(\mathbf{u}_h) + \frac{\partial \phi_j}{\partial y}\mathbf{f}_2(\mathbf{u}_h)\right] dV + \int_{\partial k \setminus \partial \Omega} \phi_j \mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) dS + \int_{\partial k \cap \partial \Omega} \phi_j \mathbf{H}^b(\mathbf{u}_h^-(\mathbf{u}_h^+, \mathbf{n}), \mathbf{n}) dS = 0 \quad \forall k \in \mathcal{T}_{h,p} \tag{4}$$

where $\{\mathbf{R}_{j,k}, j = 1, \ldots, M; k = 1, \ldots, N\}$ denotes the vector of steady-state residuals within all elements in the computational domain (N: total number of elements), and $\mathbf{n} = (n_x, n_y)$ represents the unit normal vector outward to the boundary. The notations of $\mathbf{u}_h^+$ and $\mathbf{u}_h^-$ represent interior and exterior traces of the Galerkin finite-element approximation, $\mathbf{u}_h$, since the numerical solution is discontinuous between element interfaces. $\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$ and $\mathbf{H}^b(\mathbf{u}_h^-(\mathbf{u}_h^+, \mathbf{n}), \mathbf{n})$ represent numerical flux functions employed on all interior edges and the edges coinciding with physical boundaries of the computational domain $\partial\Omega$, respectively. Current implementations for the inter-element flux function $\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$ include the Riemann flux approximation of HLLC [23,24] and Lax–Friedrichs [25]. To satisfy the dual-consistency condition [26], it is required that there is no explicit dependence on $\mathbf{u}_h^+$ in the boundary flux function, $\mathbf{H}^b(\mathbf{u}_h^-(\mathbf{u}_h^+, \mathbf{n}), \mathbf{n})$. Here we set $\mathbf{H}^b(\mathbf{u}_h^-(\mathbf{u}_h^+, \mathbf{n}), \mathbf{n}) = \mathbf{f}_1(\mathbf{u}_h^-)n_x + \mathbf{f}_2(\mathbf{u}_h^-)n_y$, where $\mathbf{u}_h^-$ is determined to have the same density, total energy and tangential velocity $(u, v)_\parallel^t$ as $\mathbf{u}_h^+$, given by:

$$(u, v)_\parallel^t = (u^+, v^+)^t - (u^+ n_x + v^+ n_y)(n_x, n_y)^t \tag{5}$$

The Galerkin finite-element approximation in the weak formulation is expanded as a series of truncated basis functions [9] and solution coefficients as:

$$\mathbf{u}_h = \sum_{i=1}^{M} \tilde{\mathbf{u}}_{h_i} \phi_i(\mathbf{x}) \tag{6}$$

Since the set of basis functions is defined in a reference triangle $\overline{\Omega}$ spanning between $\{0 < \xi, \eta < 1\}$ [27], a coordinate mapping from the reference to a physical triangle is required for the computation of the first-order derivatives and integrals denoted in Eq. (4). The reference-to-physical transformation and the corresponding Jacobian associated with each element k are given by:

$$\mathbf{x}_k = \sum_{i=1}^{M} \tilde{\mathbf{x}}_{k_i} \phi_i(\xi, \eta) \quad J_k = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{7}$$

In the simple case of straight-sided elements the transformation is linear, thus the geometric mapping $\tilde{\mathbf{x}}_k$ can be evaluated by using only the element vertex coordinates. However, in the more complex cases of high-order curved elements, which are typically required at curved boundaries, additional surface nodes [28] must be included for determining the higher-order modes $(p > 1)$ of the geometric mapping coefficients, obtained by:

$$\tilde{\mathbf{x}}_k = \mathbf{\Phi}^{-1} \hat{\mathbf{x}}_{p_k} \tag{8}$$

where $\hat{\mathbf{x}}_{p_k} = \{\mathbf{x}_{c_k}, \mathbf{x}_{q_k}\}$ refers to the coordinates of physical quadrature points, consisting of the element vertices, $\mathbf{x}_{c_k}$ as well as additional surface points, $\mathbf{x}_{q_k}$, for the element k. $\mathbf{\Phi}$ denotes the projection mapping matrix which is constituted by the basis functions evaluated at the aforementioned quadrature points $(\hat{\xi}_{p_k} \leftarrow \hat{\mathbf{x}}_{p_k})$ in the reference triangle. The evaluation of the volume integrals in Eq. (4) is computed by use of Gaussian quadrature rules [29,30] which are exact for polynomial degree $2p$, and the surface integrals use Gauss–Legendre–Lobatto quadrature rules [27] which are exact for polynomial degree of $2p + 1$ [7,9].

The convergence of this non-linear system of equations is accelerated by an *hp*-multigrid approach driven by a linearized element Gauss–Seidel solver. Further details pertaining to the acceleration of the flow solver can be found in references [7,9]. Once the flow approximation is obtained, it is relatively simple to compute an objective functional of interest in engineering applications, such as lift, drag (L/D) or integrated temperature as chosen in the present work. These objectives are readily expressed as surface integrals of the flow-field variables over solid walls where forces or heating are concerned. The Cartesian force $\mathbf{F} = (F_x, F_y)$ acting on the boundaries of interest $(\partial\Omega_w)$ for the model problem is given as:

$$F_x = \int_{\partial\Omega_w} p n_x dS \quad F_y = \int_{\partial\Omega_w} p n_y dS \tag{9}$$

Then the lift and drag which are defined as the components of $\mathbf{F}$ perpendicular and tangential to the free-stream velocity, respectively, are derived as,

$$L = -F_x sin\alpha + F_y cos\alpha \quad D = F_x cos\alpha + F_y sin\alpha \tag{10}$$

where α is the angle of attack. The objective functional of integrated temperature on a solid surface is computed as:

$$T_I = \int_{\partial\Omega_w} T dS = \int_{\partial\Omega_w} \frac{p}{\rho} dS \tag{11}$$

3. Adjoint-based error estimation

In this section, we address and analyze adjoint-based error estimates for specific output functionals. The goal of this procedure is to obtain a spatial distribution of the functional error which can be used either to correct the current functional value or to drive an adaptive meshing procedure for reducing functional simulation error.

3.1. Formulation

Consider a coarse mesh $\mathcal{T}_H$, where the parameter H refers to an average element size as well as a low discretization order p for the current finite-element mesh (The absorption of the notation p is for conciseness). Let $J(\tilde{\mathbf{u}})$ denote an objective functional of interest, which is computed based on the flow-field variables in terms of the expansion coefficients, $\tilde{\mathbf{u}} = \{\tilde{\mathbf{u}}_{i,k}, i = 1,\ldots,M; k = 1,\ldots,N\}$ (refer to Eq. (6)). The objective functional can be evaluated on the coarse mesh, represented as $J_H(\tilde{\mathbf{u}}_H)$ by using the steady-state flow solution $\tilde{\mathbf{u}}_H$ that satisfies Eq. (4) on the coarse mesh. We seek an approach to approximate the objective functional on a globally refined mesh $\mathcal{T}_h$, as illustrated in Fig. 1, by either subdividing each element into four congruent elements (subdivided mesh), or by increasing the order of discretization from p to $p+1$ (enriched mesh). However, in practice we wish to avoid computing the flow solution on $\mathcal{T}_h$ directly due to the possibly high expense of solving the fine level flow problem. Therefore, we expand the output functional $J_h(\tilde{\mathbf{u}}_h)$ on the fine mesh with respect to the projected coarse grid flow solution, $\tilde{\mathbf{u}}_H^h = I_H^h \tilde{\mathbf{u}}_H$, based on a Taylor series expansion as:

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

where $J_h(\tilde{\mathbf{u}}_H^h)$ is the output functional on the fine mesh evaluated with the projected coarse mesh flow solution. The vector $\left(\frac{\partial J_h}{\partial \tilde{\mathbf{u}}_h}\right)_{\tilde{\mathbf{u}}_H^h}$ denotes the sensitivities of the fine level objective functional with respect to the fine level flow solution evaluated at the state $\tilde{\mathbf{u}}_H^h$. If the output functional consists of a surface integral, the sensitivities of the objective with respect to the flow solution are non-zero only on boundary elements which intersect the surface under consideration. Similarly, the residual vector $\mathbf{R} = \{\mathbf{R}_{j,k}, j = 1,\ldots,M; k = 1,\ldots,N\}$ can be expanded about the projected coarse level flow solution:

$$\mathbf{R}_h(\tilde{\mathbf{u}}_h) = \mathbf{R}_h(\tilde{\mathbf{u}}_H^h) + \left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{u}}_h}\right]_{\tilde{\mathbf{u}}_H^h} (\tilde{\mathbf{u}}_h - \tilde{\mathbf{u}}_H^h) + \cdots \tag{13}$$

where $\left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{u}}_h}\right]_{\tilde{\mathbf{u}}_H^h}$ is the full-Jacobian matrix of the fine level flow problem evaluated by using the projected coarse mesh flow solution state. We proceed to derive an approximation of the solution error, $\tilde{\mathbf{u}}_h - \tilde{\mathbf{u}}_H^h$, by re-arranging Eq. (13), given as,

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

where the fact that $\mathbf{R}_h(\tilde{\mathbf{u}}_h) = 0$ has been used, since $\tilde{\mathbf{u}}_h$ represents the finite-element flow solution on the fine mesh. Substituting Eq. (14) into Eq. (12), we obtain an expression to approximate the fine mesh functional as:

$$J_h(\tilde{\mathbf{u}}_h) \approx J_h(\tilde{\mathbf{u}}_H^h) - \left(\frac{\partial J_h}{\partial \tilde{\mathbf{u}}_h}\right)_{\tilde{\mathbf{u}}_H^h} \left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{u}}_h}\right]_{\tilde{\mathbf{u}}_H^h}^{-1} \mathbf{R}_h(\tilde{\mathbf{u}}_H^h) \tag{15}$$

where the flow residuals $\mathbf{R}_h(\tilde{\mathbf{u}}_H^h)$ are non-zero since the coarse level flow solution projected onto the refined mesh $\tilde{\mathbf{u}}_H^h$ does not satisfy the discretized flow equations in the fine space. In order to relate the functional error to the local residuals of the primal solution and to avoid computing the inverse of the Jacobian matrix directly, the fine adjoint variable, $\boldsymbol{\lambda}_h$, is introduced, satisfying:

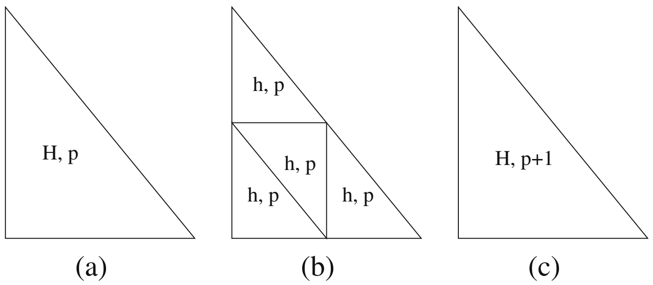


Fig. 1. Illustration of refined finite-element meshes. (a) Coarse mesh. (b) Subdivided mesh. (c) Enriched mesh.

$$\left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{u}}_h}\right]^T_{\tilde{\mathbf{u}}^h_H} (\boldsymbol{\lambda}_h)_{\tilde{\mathbf{u}}^h_H} = \left(\frac{\partial J_h}{\partial \tilde{\mathbf{u}}_h}\right)^T_{\tilde{\mathbf{u}}^h_H} \tag{16}$$

This particular approach corresponds to a discrete adjoint formulation, where the Jacobian of the discrete equations is used in the definition of the adjoint variables. The coefficient matrix of the system of equations constructed for the adjoint problem corresponds to the transpose of the full-Jacobian matrix of the flow equations and thus the adjoint solution scheme delivers similar convergence characteristics as the flow problem. Substituting Eq. (16) into Eq. (15) and re-arranging, we may write the functional on the fine mesh as:

$$J_h(\tilde{\mathbf{u}}_h) \approx J_h(\tilde{\mathbf{u}}^h_H) \underbrace{-(\boldsymbol{\lambda}_h)^T_{\tilde{\mathbf{u}}^h_H} \mathbf{R}_h(\tilde{\mathbf{u}}^h_H)}_{\varepsilon^*_a} \tag{17}$$

The term ε^*_a in Eq. (17) involves the fine mesh adjoint solution. In order to avoid solving the fine mesh adjoint problem, we first solve the discrete adjoint problem on the coarse mesh:

$$\left[\frac{\partial \mathbf{R}_H}{\partial \tilde{\mathbf{u}}_H}\right]^T_{\tilde{\mathbf{u}}_H} \boldsymbol{\lambda}_H = \left(\frac{\partial J_H}{\partial \tilde{\mathbf{u}}_H}\right)^T_{\tilde{\mathbf{u}}_H} \tag{18}$$

using an *hp*-multigrid approach driven by a linearized element Gauss–Seidel smoother [7,9] to accelerate convergence. An approximation to the fine level adjoint solution is then obtained by performing a reconstruction postprocessing procedure to the coarse level adjoint solution, $\boldsymbol{\lambda}_H$, onto the refined mesh via a patch-wise least square method [26], where the reconstructed adjoint solution at an element k ($k \in \mathcal{T}_h$) is solved by minimizing the following equations,

$$I((\boldsymbol{\lambda}^h_H)_{i_k}) = \sum_{l \in \mathcal{P}_k} \left\| \sum_{j=1}^{M^*} (\boldsymbol{\lambda}^h_H)_{j_k} \phi_j|_l - \sum_{j=1}^{M} (\boldsymbol{\lambda}_H)_{j_k} \phi_j|_l \right\|^2_{L_2} \quad i = 1, \cdots, M^* \tag{19}$$

with respect to each variable as:

$$\frac{\partial I\left((\boldsymbol{\lambda}^h_H)_{i_k}\right)}{\partial (\boldsymbol{\lambda}^h_H)_{j_k}} = 0 \quad i,j = 1, \cdots, M^* \tag{20}$$

where $\mathcal{P}_k$ represents the patch of element k consisting with all of its neighboring elements, and the subscript $(\cdot)_{i_k}$ denotes the ith expansion coefficient of the adjoint solution in element k. M^* denotes the number of modes required for the order of discretization on the fine mesh. Using the reconstructed adjoint expression in Eq. (17), we obtain the error expression:

$$J_h(\tilde{\mathbf{u}}_h) \approx J_h(\tilde{\mathbf{u}}^h_H) \underbrace{-(\boldsymbol{\lambda}^h_H)^T \mathbf{R}_h(\tilde{\mathbf{u}}^h_H)}_{\varepsilon_a} \underbrace{-((\boldsymbol{\lambda}_h)_{\tilde{\mathbf{u}}^h_H} - (\boldsymbol{\lambda}^h_H))^T \mathbf{R}_h(\tilde{\mathbf{u}}^h_H)}_{\varepsilon_r} \tag{21}$$

where ε_a and ε_r denote the computable adjoint correction and the remaining error, respectively. The computable error correction ε_a is expressed as the inner product of the local residuals with the reconstructed adjoint solution, and results in a spatial distribution of the functional error. The remaining error term is typically an order of magnitude smaller than the computable error correction, as shown in Section 5.1.2, and thus the remaining error can be safely absorbed into the adjoint correction without compromising the reliability of the adaptive algorithm. Therefore the functional approximation on a globally refined mesh becomes:

$$J_h(\tilde{\mathbf{u}}_h) \approx J_h(\tilde{\mathbf{u}}^h_H) - (\boldsymbol{\lambda}^h_H)^T \mathbf{R}_h(\tilde{\mathbf{u}}^h_H) \tag{22}$$

Next the variation of the discrete functional between coarse and fine levels, $J_h(\mathbf{u}_h) - J_H(\mathbf{u}_H)$, is derived by subtracting from both sides the coarse functional, $J_H(\mathbf{u}_H)$, written as:

$$\underbrace{J_h(\mathbf{u}_h) - J_H(\mathbf{u}_H)}_{\varepsilon_c} \approx \underbrace{J_h(\mathbf{u}^h_H) - J_H(\mathbf{u}_H)}_{\varepsilon_d} \underbrace{-(\boldsymbol{\lambda}^h_H)^T \mathbf{R}_h(\tilde{\mathbf{u}}^h_H)}_{\varepsilon_a} \tag{23}$$

In this form, the additional term ε_d is the error incurred between the evaluation of the functional on the fine level using the projected coarse mesh flow solution, and the functional evaluated on the coarse mesh level with the coarse mesh flow solution. Referring to Eq. (23), the major computational work at each adaptation cycle involves solving the flow and adjoint problems on the coarse mesh and the proper reconstruction procedure to the fine mesh level.

3.2. Refinement criteria

The adjoint correction ε_a provides a spatial functional error estimator on each element k in the current computational domain, given as,

$$\varepsilon_{a,k} = -(\boldsymbol{\lambda}^h_H)^T_k \mathbf{R}_{h,k}(\tilde{\mathbf{u}}^h_H) \tag{24}$$

where the local element-wise error indicator is expressed as the inner product of the local residual vector with the approximated adjoint variables within the children elements of a coarse element k. In order to adaptively reduce the error in the objective functional, an error-balancing refinement criterion [13] is employed in the current work where elements are flagged for local refinement if the inequality

$$|\varepsilon_{a,k}| > \frac{E_{tol}}{N} \tag{25}$$

holds, where E_{tol} is a positive user-desired global tolerance and N denotes the number of elements in $\mathcal{T}_H$. Eq. (25) provides a stopping criterion for the simulation when the specified error tolerance is achieved or when the local error indicator on the current mesh is within the maximum allowable equidistributed error level $\bar{\eta} = \frac{E_{tol}}{N}$ in each cell.

We note that alternate criteria for mesh refinement such as the use of decreasing error tolerance with increasing refinement levels have also been proposed [19], although the current simple criterion is best suited for validating the effectiveness of the error estimation strategy. If the termination criterion is not reached, we store a list of the indices of the flagged elements and then proceed to the local mesh refinement operation presented in the following section to improve the quality of the finite-element mesh.

4. Mesh refinement

In the context of DG discretizations, accuracy enhancements can be achieved either by refining the mesh (h), or by raising the order of accuracy (p) of the discretization. Therefore, the mesh refinement operation can be classified as h-refinement, p-enrichment and combined hp-refinement.

4.1. *h-Refinement*

h-Refinement involves locally refining the mesh by adding nodes to midpoints of each of the edges of flagged elements on the current coarse mesh. Thus an h-refinement procedure subdivides one element into four self-similar children elements (1:4 refinement), while keeping the original discretization order p fixed. New nodes on interior edges are simply added at the edge midpoints since all interior elements have straight-sided edges. However, for wall or surface boundary elements, the newly generated nodes and additional surface quadrature points required for curvilinear boundary conditions (c.f. Eqs. (7) and (8)) in the presence of higher-order discretizations ($p > 1$) must conform to the original geometry boundary. This is achieved by utilizing a quintic spline interpolation to recover the original surface shape. Fig. 2 illustrates an example of the meshes obtained by an h-refinement process on an airfoil using fixed discretization orders $p = 1$ and $p = 4$, respectively.

Upon subdividing one flagged element into four congruent children elements, hanging nodes may be generated on shared edges where only a single side of two neighboring elements has been marked for refinement. Several algorithms are employed in this work in order to eliminate the presence of hanging nodes as depicted in Fig. 3: a given element with a single hanging node is subdivided into two children elements (1:2 refinement); if two hanging nodes are generated in one element, a regular 1:4 subdivision is implemented. Since repeated implementation of the 1:2 refinement due to hanging nodes on a given element may result in poorly shaped elements, only a single level of 1:2 refinement is permitted, and 1:4 refinement is employed if subsequent refinement of these elements is required. Additional mesh optimization techniques [31] are also employed including an edge-swapping procedure, where the local topology of the mesh is changed based on a criterion which maximizes the minimum interior angle of the elements (a MaxMin triangulation), and a mesh smoothing procedure where the positions of the mesh points are relocated to the centroid of the surrounding mesh points which are connected to the current point, while the mesh topology remains unchanged.

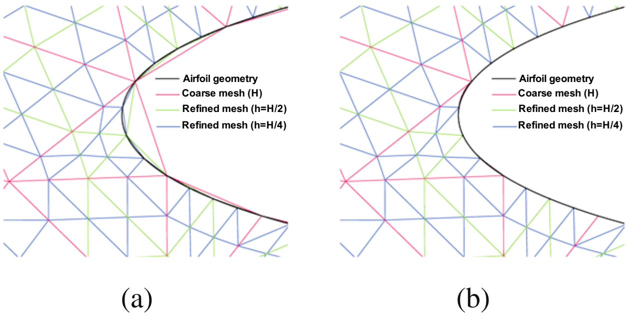


Fig. 2. An example of h-refinement for an original coarse mesh and its consecutively refined meshes near an airfoil boundary. (a) Refinement on a fixed discretization order $p = 1$. (b) Refinement on a fixed discretization order $p = 4$.

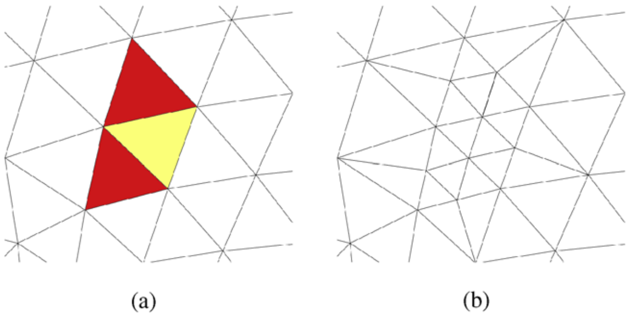


Fig. 3. An example of h-refinement to eliminate the presence of hanging nodes. (a) Original mesh where the red elements represent elements flagged for regular 1:4 refinement and the yellow element represents a forced 1:4 refinement in this element due to the generation of two additional nodes. (b) The subsequent h-refined mesh.

4.2. p-Enrichment

An alternate approach to mesh h-refinement is p-enrichment, where the local discretization order p is increased to $p+1$, while the underlying triangulation remains fixed. Note that if the p-enrichment occurs in a surface element where curved boundary conditions are required, additional surface points must be employed in the presence of higher-order solutions. These additional points are obtained by projecting additionally created boundary points onto the original boundary geometry description, as illustrated in Fig. 4, for the creation of a $p=4$ curved boundary element in the two-dimensional case. Moreover, during the process of adaptive p-enrichment, it is very common to have two neighboring elements with different discretization orders, as illustrated in Fig. 5 for an example of $p^+=2$ and $p^-=1$. Due to the fact that additional quadrature points required for higher-order discretizations must be included for obtaining the desired solution accuracy [32], the optimal number of quadrature points for calculating the surface integral at a shared elemental interface with $p^+ \neq p^-$ is determined by the higher order of discretizations. However, the number of quadrature points for the volume integral evaluations is selected only based on the discretization order of the computed element itself. For example, in the example illustrated in Fig. 5 the quadrature points for the volume integral are determined by p^+ and p^- for the left and the right elements, respectively, and those for the edge integral are determined by p^+ alone. In addition, the Galerkin solution approximations, i.e. $\mathbf{u}_q^+$ and $\mathbf{u}_q^-$, at each quadrature point are obtained by using their own expansion coefficients and basis functions, given by $\mathbf{u}_q^+ = \sum_{i=1}^{M^+(p^+)} \tilde{\mathbf{u}}_i^+ \phi_i^+$ and $\mathbf{u}_q^- = \sum_{i=1}^{M^-(p^-)} \tilde{\mathbf{u}}_i^- \phi_i^-$, respectively.

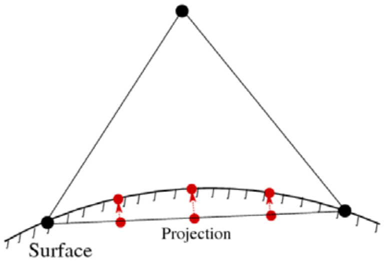


Fig. 4. Illustration of projection of additional surface points for creation of a $p=4$ curved boundary element in the two-dimensional case.

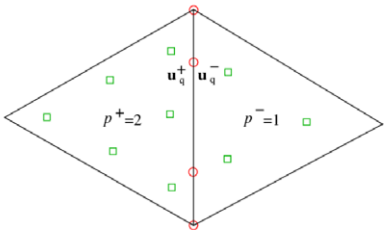


Fig. 5. Illustration of requirement for volume and edge quadrature points between two neighboring elements with $p^+=2$ and $p^-=1$. Square symbols: volume; circle symbols: edge.

p-enrichment is known to be capable of achieving exponential convergence in terms of number of unknowns for smooth solutions. However, this approach can not be employed directly for non-smooth solutions or for cases with shocks or singularities, since the higher-order discretizations ($p > 0$) produce unbounded oscillations near shocks and may result in loss of stability. In order to avoid higher-order discretizations in shock regions in the context of an adaptive scheme, the combined *h*–*p* refinement approach is invoked.

4.3. *hp*-Refinement

In the *hp*-refinement approach, a choice between *h*-refinement and *p*-enrichment must be made individually for each one of the elements flagged as large error contributors on the current coarse mesh. The goal of the combined *hp*-adaptive refinement scheme is to make use of *h*-refinement in regions with discontinuities or large flow gradients where high-order discretizations may not perform properly and to utilize *p*-enrichment in areas with smooth solution behavior to deliver high accuracy. This is accomplished by employing a smoothness indicator based on an element-wise or inter-element indicator.

4.3.1. *Element-wise indicator*

For cases with smooth solutions, the coefficients of the solution expansions in the DG methodology are assumed to have a similar decay rate ($\sim 1/n^2$) as the Fourier coefficients [1,8,21]. On the other hand, for non-smooth solutions, the discontinuities in the solutions deteriorate this decay rate, and thus a local indicator based on the measure of the decay rate of solution expansion coefficients can be utilized to determine a suitable sensor of the solution smoothness. Within each element *k* of a computational mesh, this element-wise smoothness indicator is defined as:

$$S_k = \frac{(\rho - \breve{\rho}, \rho - \breve{\rho})_k}{(\rho, \rho)_k} \tag{26}$$

In this expression, $(\cdot,\cdot)$ denotes the standard L_2 inner product within element *k*. ρ and $\breve{\rho}$ refer to one representative quantity of the flow variables (as represented by density in the expression), determined by using a full (for order *p*) or a truncated expansion (for order $p-1$), respectively, given by:

$$\rho = \sum_{i=1}^{M} \tilde{\rho}_i \phi_i \tag{27}$$

$$\breve{\rho} = \sum_{i=1}^{\overline{M}} \tilde{\rho}_i \phi_i \tag{28}$$

where M and $\overline{M}$ denote the total number of terms in the solution expansions for discretization orders *p* and $p-1$, respectively. Note that the same expansion coefficients $\tilde{\rho}_i$ are used in Eqs. (27) and (28). Given the fact that the definition of S_k involves square quantities, we expect that the value of S_k scales like $\sim\frac{1}{p^4}$ [8] and an automatic criteria for choosing between *h*-refinement and *p*-enrichment within a flagged element is given as:

$$\begin{cases} log_{10}(S_k) \geqslant s_0 - \textsc{l}, & h\text{-refinement} \\ log_{10}(S_k) < s_0 - \textsc{l}, & p\text{-enrichment} \end{cases} \tag{29}$$

where the parameter $s_0 \sim \frac{1}{p^4}$, and $\textsc{l}$ is an empirical parameter and set equal to 6.0 in our experience so as to capture non-smooth regions. Moreover, we use density in practice to determine the decay rate of expansion coefficients and other flow quantities such as Mach number can be also considered for this indicator.

4.3.2. *Inter-element indicator*

The other smoothness indicator used in this paper particularly for the cases of high speed flows in the next section is a jump discontinuity indicator [1,22] which measures the integral of the discontinuous jump of the flow-field variables over each edge of an element, given as,

$$S_k = \frac{1}{|\partial\Omega_k|} \int_{\partial\Omega_k} \left| \frac{u^+ - u^-}{\frac{1}{2}(u^+ + u^-)} \right| dS \tag{30}$$

where u^+ and u^- denote the inner and outer traces of the selected flow quantity. The discontinuity detection scheme is implemented as:

$$\begin{cases} S_k > \frac{1}{\mathcal{K}}, & h\text{-refinement} \\ S_k \leqslant \frac{1}{\mathcal{K}}, & p\text{-enrichment} \end{cases} \tag{31}$$

The parameter $\mathcal{K}$ is required to be sufficiently large to capture shocks of any strength and in our experience it is set equal to 25. The jump indicator performs better than the element-wise smoothness indicator for high speed flows and pressure has been found to be a reliable quantity to calculate inter-element jumps.

5. Numerical results

The proposed adaptive mesh strategies are first applied to a subsonic flow over an idealized four-element airfoil. Comparisons of the adapted meshes and the error convergence histories are demonstrated for *h*-refinement alone as well as for *p*-enrichment alone, and for the combined *hp*-adaptive approach. The functional error estimates are examined for each adaptation process of the various mesh adaptation strategies. Another test case involves strong shocks or discontinuities produced by hypersonic flow over a half-circular cylinder in order to demonstrate the shock-capturing properties of the *hp*-adaptive approach.

The input for the simulation consists of an initial coarse mesh $\mathcal{T}_{H,p}$ associated with a uniformly lower discretization order p, an objective functional of interest as well as a user-desired functional error tolerance. The output includes the refined mesh $\mathcal{T}_{\widehat{h},\widehat{p}}$ together with a polynomial degree distribution $\widehat{p}_k,\ k \in \mathcal{T}_{\widehat{h},\widehat{p}}$, and the final flow solution as well as final functional value. The performance of the adaptive process is measured in terms of the reduction of the error in the functional output versus the number of unknowns (i.e. degrees of freedom) or the computational cost in CPU time. Degrees of freedom (DOF) are computed as the total number of unknowns, excluding the number of subcomponents for the system of equations (i.e. 4 for the two-dimensional Euler equations). The computational time required for one adaptation cycle includes the accumulated expense of computing both the flow and adjoint problems for all previous adaptation cycles plus an additional flow solution for the current adapted mesh. On the other hand, the computational cost for the uniform refinement approach is only considered as the expense of solving a single flow problem on the uniformly refined mesh.

5.1. Subsonic flow over a four-element airfoil

The computational domain for this test case consists of an idealized four-element airfoil with a far-field outer boundary placed at a distance of 50 chord lengths away from the airfoil. The initial mesh illustrated in Fig. 6 contains 1508 unstructured triangular elements. A low free-stream Mach number of 0.2 as well as a zero angle of attack are prescribed for the flow field. The HLLC approximate Riemann solver [23,24] is used for the flux function at all interior edges/boundaries to resolve the discontinuity in the flow variables on each shared edge, and the particular wall boundary treatment described in Section 2 is employed at all wall/surface boundaries. The exact functional value is taken from a solution on a *h*-adapted mesh with a uniformly $p = 3$ discretization order, where the relative difference in the functional between the current and the previous adapted functional is within 10^{-4}.

5.1.1. hp-Multigrid solver

As mentioned previously, both the primal and the discrete adjoint problems must be solved at each mesh adaptation cycle, which consumes most of the computational cost (over 95%) throughout the adaptive simulation. Thus, in order to make the whole scheme competitive, we make use of efficient solution techniques including an *hp*-multigrid approach driven by an element Gauss–Seidel smoother to accelerate convergence for both the flow (primal) and adjoint (dual) problems. The Gauss–Seidel smoother requires storage of the full-Jacobian matrix of the flow equations and follows an ordered sweep across elements using the latest available neighboring information. The Jacobian matrix based on the final steady-state flow solution is then stored for the solution of the discrete adjoint problem.

The computed steady-state solution in terms of Mach number contours and the *x*-momentum component of the adjoint solution corresponding to the objective functional of lift on the initial mesh with a uniform $p = 4$ order of discretization are illustrated in Fig. 7(a) and (b), respectively. The *x*-momentum component of the adjoint solution corresponds to the sensitivity of the lift value to point sources of *x*-momentum in the flow field. The fact that regions upstream of the airfoil denoted

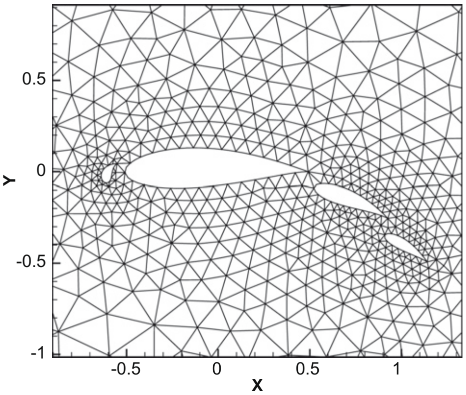
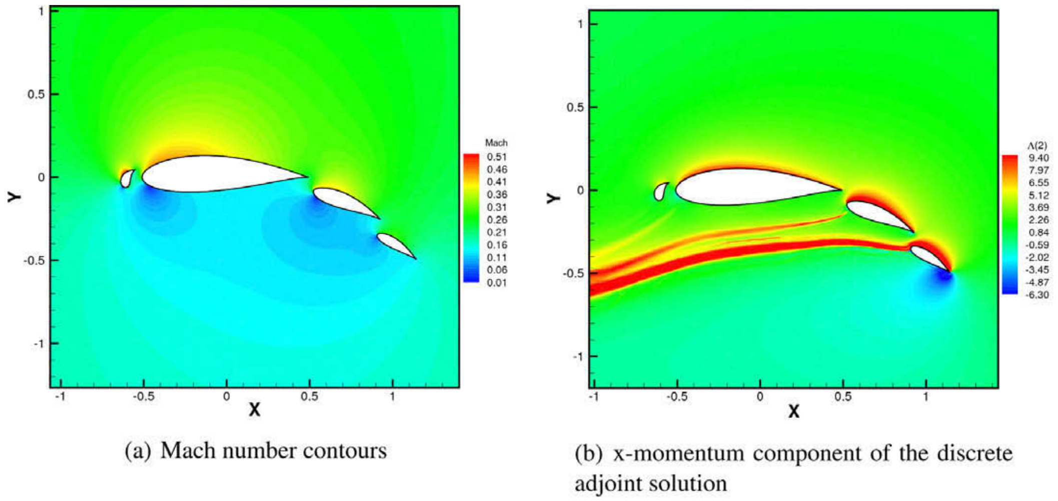


Fig. 6. Close-up view of the initial unstructured mesh (1508 elements) for the test case of subsonic flow over a four-element airfoil with free-stream Mach number of 0.2.



(a) Mach number contours

(b) x-momentum component of the discrete adjoint solution

Fig. 7. Flow and discrete adjoint solutions for subsonic flow over a four-element airfoil with free-stream Mach number of 0.2 at 0° angle of attack; lift is set to be the objective functional of interest.

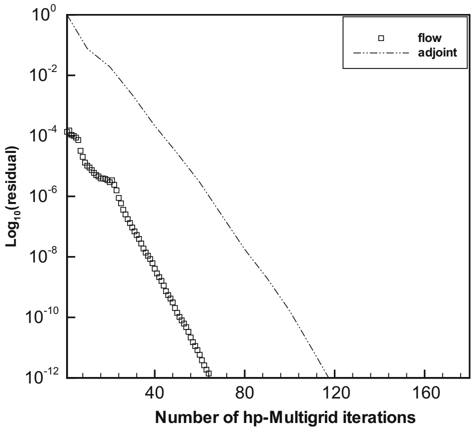


Fig. 8. Comparison of the convergence histories of both flow and adjoint (for lift output) solutions in terms of number of *hp*-multigrid iterations for subsonic flow over a four-element airfoil with free-stream Mach number of 0.2 at 0° angle of attack.

by the red contours in Fig. 7(b) have a significant effect on the lift output provides an illustration of the non-local effect due to the hyperbolic property of the governing equations.

Fig. 8 provides a comparison of convergence histories for both the flow and the adjoint solvers in terms of the number of *hp*-multigrid iterations. The discrete adjoint solver delivers a convergence rate which is asymptotically equivalent to that of the flow solver, since the adjoint problem is constructed as an exact dual to the primal problem.

5.1.2. Error prediction for two mesh levels

Before proceeding to the mesh adaptation test cases, it is important to verify that the computable adjoint error correction denoted as ε_a in Eq. (21) is capable of providing an acceptable approximation for the fine adjoint error correction, denoted as ε_a^* in Eq. (17), as well as providing an accurate prediction of the corresponding globally refined mesh value.

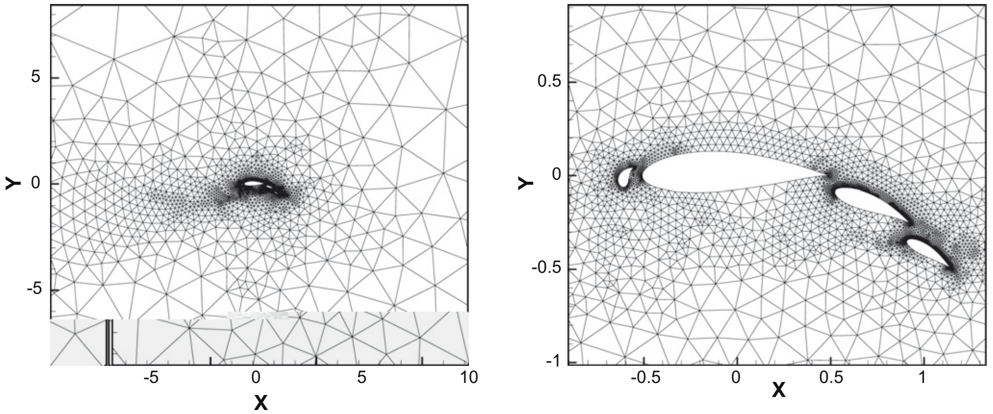
Tables 1 and 2 involve comparisons of the computed drag change between two mesh levels, where the fine level is obtained by increasing the grid resolution using a uniform 1:4 *h*-refinement (Case A) and by raising the discretization order p to $p+1$ globally (Case B), with the computable adjoint error correction ε_a and the corresponding term, ε_a^* evaluated using the fine level adjoint solution. We observe that the projected adjoint error correction provides good agreement with the fine adjoint error estimates in both cases, which demonstrate that the proposed reconstructive procedure for the coarse level adjoint sufficiently captures the behavior of the fine level adjoint. Moreover, the accuracy of the resulting estimates based on the computable adjoint error is satisfactory compared with the corresponding two-level functional error. For example, the computed ε_a value corresponds to 98.8% of the actual two-level error for Mesh 3 of Case A and 99.7% for Mesh 4 of Case B.

Table 1
Case A: comparisons of the adjoint error correction computed using fine and projected coarse adjoint solutions (ε_a^* and ε_a) with the true functional error between two mesh levels where grid resolution is increased by a uniform 1:4 refinement and the discretization order $p = 1$ is fixed for all runs. Mesh 1, 2 and 3 contain 1508, 6032 and 24128 elements, respectively.

Mesh	Error	ε_a^*	ε_a	% of error
1	−2.76903217E−3	−2.71897249E−3	−2.39419266E−3	86.5
2	−7.10679246E−4	−7.41825861E−4	−7.19017106E−4	98.8
3	−1.14173193E−4	−1.16541513E−4	−1.15579326E−4	98.8

Table 2
Case B: comparisons of the adjoint error correction computed using fine and projected coarse adjoint solutions (ε_a^* and ε_a) with the true functional errors between two mesh levels where the discretization order is uniformly increased from p to $p+1$ and the underlying grids (1508 elements) are fixed for all runs. Mesh 1, 2, 3 and 4 are uniformly discretized by $p = 0$, 1, 2, and 3, respectively.

Mesh	Error	ε_a^*	ε_a	% of error
1	−3.81409957E−2	−3.32547210E−2	−2.03270239E−2	53.3
2	−2.82996939E−3	−3.03437520E−3	−1.33580368E−3	47.2
3	−5.72914101E−4	−5.41733478E−4	−5.85074179E−4	97.9
4	−1.72394870E−4	−1.82852793E−4	−1.71881095E−4	99.7



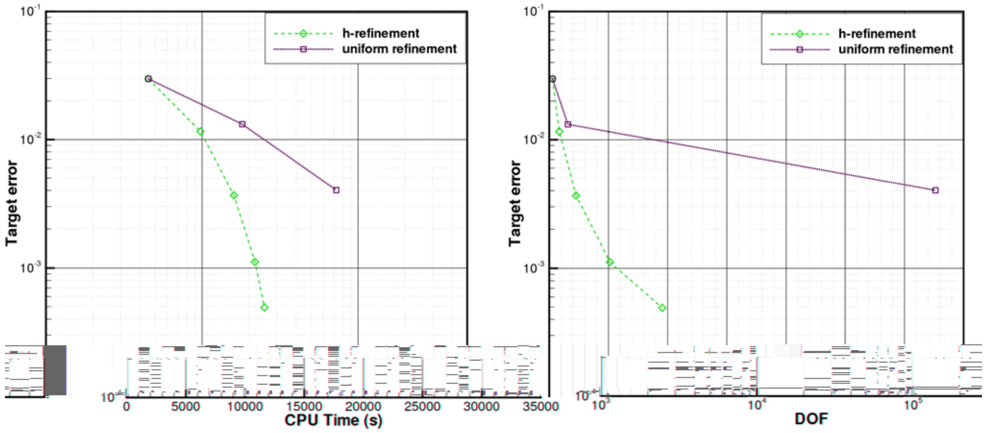
(a) Final h-adapted mesh (8,387 elements) (b) Close-up view of the final adapted mesh

Fig. 9. Final h-adapted mesh for the objective functional of lift and a fixed discretization order of $p = 1$ (i.e. second-order accurate) in the subsonic flow test case with free-stream Mach number of 0.2.

5.1.3. *h*-Adaptation for lift output

In this section, the performance of the h-adaptive mesh refinement algorithm driven by the adjoint-based error estimation is demonstrated for the case of inviscid subsonic flow over a four-element airfoil. The objective functional of interest is specified as lift, which is calculated from the static pressure distribution on the airfoil surface as described in Section 2. A uniform discretization order of $p = 1$ (i.e. second-order accurate) is fixed for all h-adaptive refinement levels and the error tolerance is set to be 10^{-3}. Fig. 9(a) and (b) depict the final h-adapted mesh with 8387 elements after four adaptation cycles. As expected, most of the refinement occurs around the airfoil surface, particularly near the leading and trailing edges. However, some h-refinement occurs upstream of the airfoil as shown in Fig. 9(a) due to the hyperbolic nature of the problem.

The convergence behavior of the functional error for the adaptive approach is compared with that achieved using uniform mesh refinement, where each element in the computational domain is refined by the aforementioned 1:4 ratio, regardless of the objective functional error contribution. Thus, the objective functional value obtained from the uniformly refined mesh represents the optimal functional error achievable on the corresponding h-adaptively refined mesh. Fig. 10 summarizes the results of the convergence of the functional error for both approaches. In terms of degrees of freedom as depicted in Fig. 10(a), the h-refinement approach achieves equivalent error levels as the uniform refinement approach at each adaptation cycle, while using fewer degrees of freedom. In other words, the uniform mesh refinement approach contains excessive resolution in areas of little influence on the functional accuracy. For example, at the second adaptation cycle, the number of unknowns required by the h-refinement approach is only 22% of the number employed by the uniform refinement method. This advantage is even more evident for the two following h-adaptation iterations, since the error convergence for the proposed approach displays roughly a factor of 4 over the uniform refinement. Moreover, in terms of the required CPU time



(b) Lift error vs. CPU time

(a) Lift error vs. degrees of freedom

Fig. 10. Comparisons of error convergence histories for the target functional of lift between the *h*-refinement and uniform mesh refinement approaches, using a fixed discretization order of $p = 1$ and a specified error tolerance of 10^{-3} in the subsonic flow test case with free-stream Mach number of 0.2.

shown in Fig. 10(b), the proposed approach incurs significantly lower computational cost resulting from the use of fewer degrees of freedom at equivalent accuracy, even though the computational cost of the adaptive approach includes all previous adaptive flow and adjoint solutions, while the uniform refinement case only includes the cost of the fine level flow solution alone.

Referring to Eq. (23), since the error estimate or correction produced by the adjoint technique, i.e. $\varepsilon_d + \varepsilon_a$, can be used for predicting the functional value on the next finer level, i.e. $J_h(\tilde{\mathbf{u}}_h) \approx J_H(\tilde{\mathbf{u}}_H) + \varepsilon_d + \varepsilon_a$, we examine the effectiveness of this adjoint-based error correction. Fig. 11 depicts the lift functional convergence histories on the *h*-adapted meshes, the functional values on the adapted meshes including the corresponding corrections, and the functional values computed on the uniformly refined meshes, in terms of degrees of freedom for this subsonic test problem. The correction produced by the adjoint-based error estimation becomes more accurate with increasing refinement levels, as seen by the fact that the last two approximated functionals (as represented by the third and the fourth star symbols), obtained by using the current functional with the added adjoint-based correction term, provide very accurate predictions for the finer functionals. The increasing effectiveness of the correction term on finer levels is explained by the fact that this is based on a linearization about the coarser level solution. The correction term can be used to predict the value of the functional on a finer level or can be used as an error estimate on the current solution.

5.1.4. *p*-Adaptation for drag output

In this case, the *p*-enrichment algorithm is performed using drag as the output. The same coarse mesh (Fig. 6) together with a uniform $p = 1$ distribution and an error tolerance of 5×10^{-4} are set as the initial inputs. The final *p*-adapted mesh depicted in Fig. 12 illustrates the variation of discretization orders ranging from $p = 1$ to $p = 4$ while operating on the same

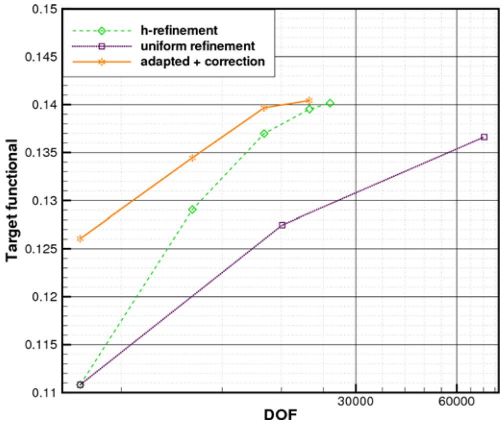


Fig. 11. Comparison of lift functional convergence as a function of degrees of freedom for the *h*-refinement and uniform mesh refinement approaches using a fixed discretization order of $p = 1$ in the subsonic flow test case with free-stream Mach number of 0.2.

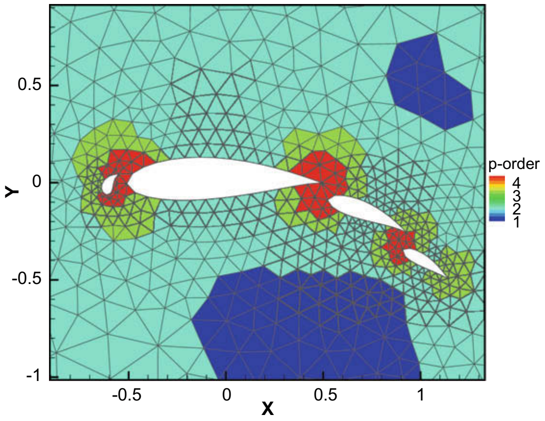
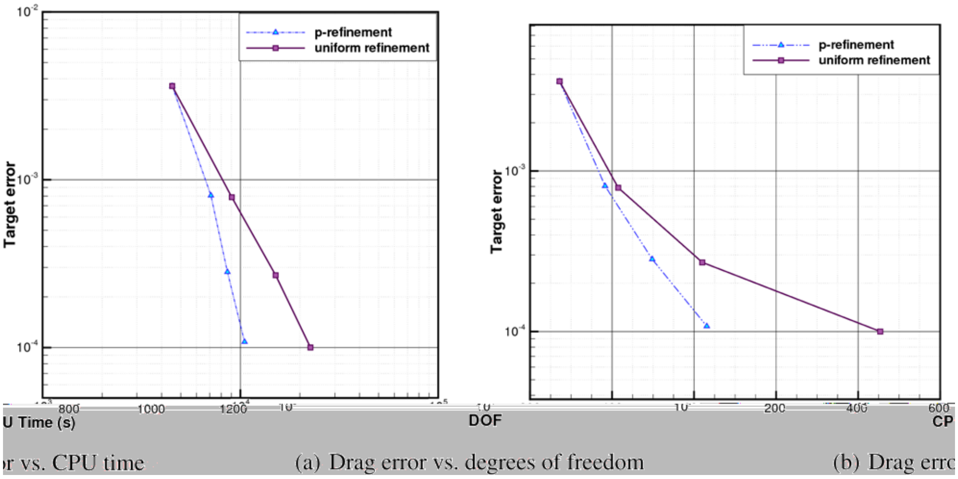


Fig. 12. Final p-adapted mesh (1508 elements; $p_{min} = 1,\ p_{max} = 4$) for the objective functional of drag in the test case of subsonic flow over a four-element airfoil with free-stream Mach number of 0.2.



r vs. CPU time (a) Drag error vs. degrees of freedom (b) Drag errc

Fig. 13. Comparisons of error convergence histories for the target functional of drag between the p-enrichment and uniform order refinement approaches, using a specified error tolerance of 5×10^{-4} in the subsonic flow test case with free-stream Mach number of 0.2.

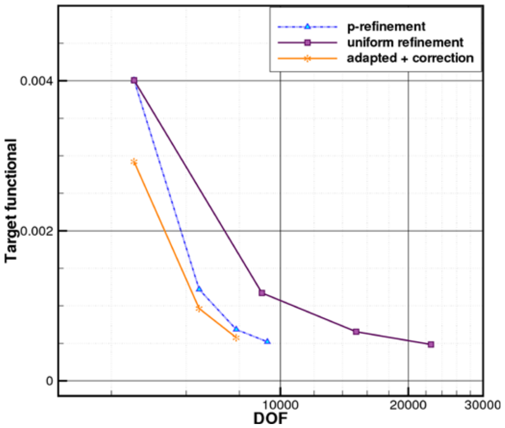


Fig. 14. Comparisons of drag functional convergence as a function of degrees of freedom for the adaptive p-enrichment and the uniform order refinement approaches in the subsonic flow test case with free-stream Mach number of 0.2.

initial mesh. Areas targeted for order refinement (the airfoil surface particularly around the leading and trailing edges) are similar to those in the lift-*h*-adaptation case.

The error convergence history of the proposed *p*-adaptive enrichment algorithm for this subsonic test case is also compared with that of the uniform order refinement approach, where the refined mesh is obtained by globally increasing the discretization order from p to $p+1$ with the underlying mesh fixed. The functional error obtained from the uniform order refinement method is used to determine the optimal achievable error reduction possible for one *p*-adaptation cycle. In terms of degrees of freedom, illustrated in Fig. 13(a), the proposed *p*-enrichment scheme is also capable of delivering equivalent error levels as the uniform refinement approach at each adaptation cycle, while requiring fewer degrees of freedom. In terms of CPU cost, as shown in Fig. 13(b), the *p*-enrichment scheme demonstrates superior performance compared to the uniform order-refinement scheme particularly on the last two adaptation cycles. For example, to achieve a 10^{-4} drag error, the uniform order refinement method incurs roughly twice the cost of the adaptive scheme in CPU time. The examination of the effectiveness of the error estimates or the correction term provided by the adjoint-based error estimation for the *p*-enrichment test case is demonstrated in Fig. 14. The functional value obtained from the purely *p*-adapted mesh at each refinement level is close to the functional obtained by the uniform order refinement. Moreover, the approximated functional for the next finer level provided by the functional with the added correction term on the previous coarse adaptation level (as represented by the orange line) is seen to become increasingly more accurate with additional refinement levels, and the finest objective functional is predicted very accurately on the previous adaptation level using the computable error correction term.

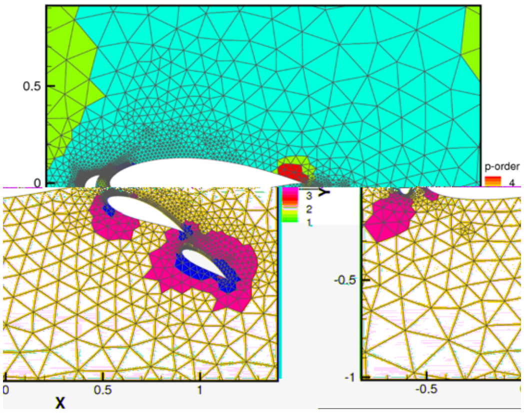
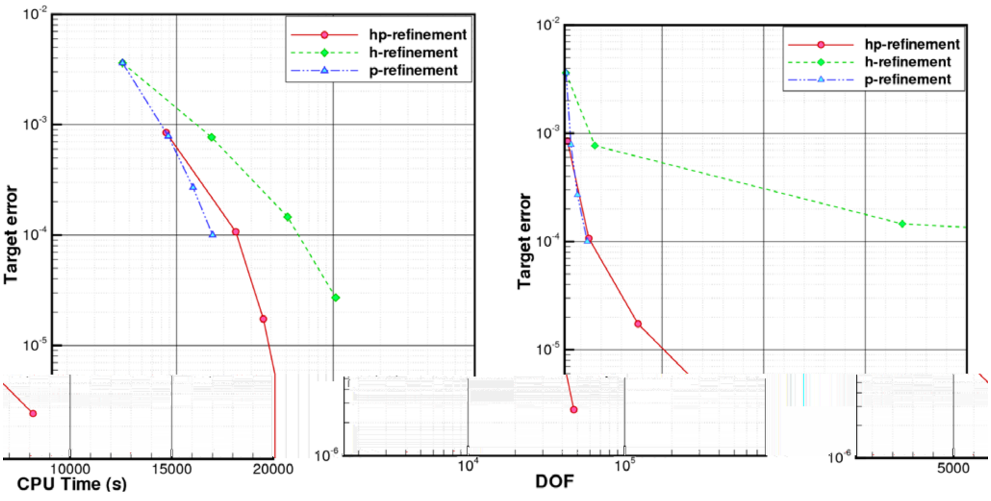


Fig. 15. Final *hp*-adapted mesh (7105 elements; $p_{min}=1, p_{max}=4$) for the objective functional of drag in the subsonic flow test case with free-stream Mach number of 0.2.



error vs. CPU time (a) Drag error vs. degrees of freedom (b) Drag

Fig. 16. Comparisons of error convergence histories for the target functional of drag between the *hp*-refinement, pure *h*-refinement and pure *p*-enrichment schemes, using a specified error tolerance of 10^{-5} in the subsonic flow test case with free-stream Mach number of 0.2.

5.1.5. hp-Adaptation for the drag output

The test case in this section involves the effective combination of both *h*- and *p*-refinement algorithms (i.e. the approach of *hp*-adaptation) for subsonic flow over the four-element airfoil, using drag as the objective functional. The element-wise smoothness indicator described in Section 4.3.1 is employed for this subsonic test case.

The *hp*-adaption test case starts with the same initial mesh (Fig. 6) together with a uniform discretization order of $p = 1$ and the desired error tolerance is specified as 10^{-5}. The final adapted mesh is shown in Fig. 15 with a total of 7105 elements and a variation of discretization orders ranging from $p = 1$ to $p = 4$. Areas of high *h*-refinement are concentrated near the airfoil leading edges, due to the presence of large gradients in these regions. On the other hand, areas of high order *p*-enrichment mainly occur near the surface of the last two airfoil-elements.

Fig. 16(a) compares the drag error convergence of *hp*-refinement with pure *h*-refinement and pure *p*-enrichment implemented using the same initial mesh and the specified error tolerance for this test case. In terms of degrees of freedom, the *h*-refinement approach represented by the green line requires more DOF for the same error reduction as the *p*- and *hp*-refine-

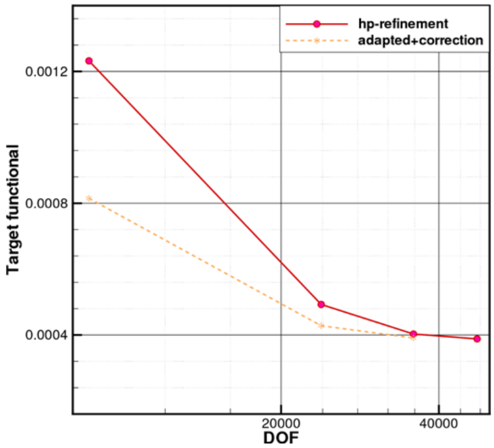
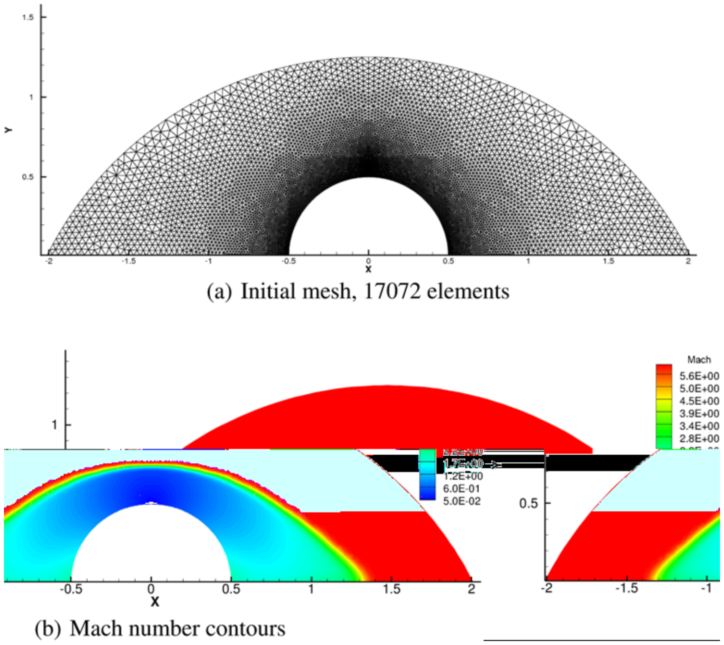


Fig. 17. Comparisons of drag functional convergence as a function of degrees of freedom for *hp*-refinement approach in the subsonic test case with free-stream Mach number of 0.2.



(a) Initial mesh, 17072 elements

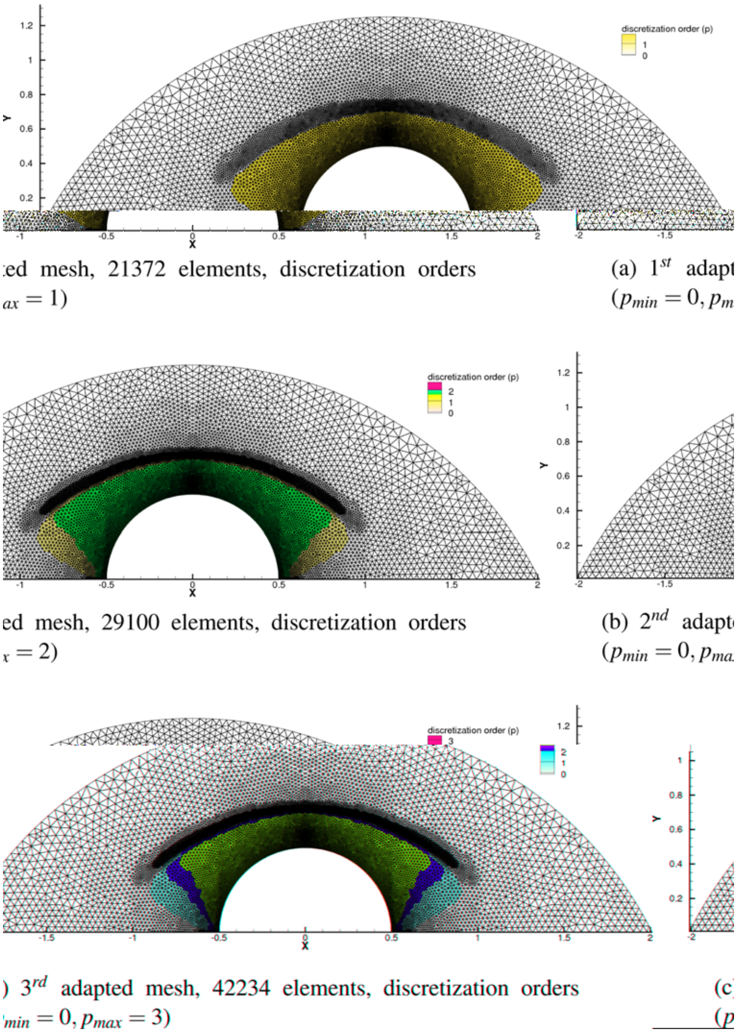
(b) Mach number contours

Fig. 18. Initial mesh and computed Mach number contours using a first-order accurate discretization ($p = 0$) for hypersonic flow over a half-circular cylinder with free-stream Mach number of 6.

ment approaches. Drag error is successively reduced to be less than 10^{-4} in the p-enrichment and hp-refinement algorithms and exponential error convergence rates are achieved in both adaptive schemes, however, the slope of the hp-enrichment (i.e slope = 5.3) exceeds the slope of 2.9 achieved using p-enrichment alone. In addition, for a fixed range of discretization orders ($p_{min} = 1, p_{max} = 4$), the hp-refinement scheme is capable of achieving roughly two orders of magnitude higher error reduction than the p-enrichment scheme. In terms of CPU cost, as illustrated in Fig. 16(b), the performance of the hp-refinement scheme demonstrates superior efficiency over the pure h-refinement scheme, while displaying similar cost compared to the p-enrichment scheme over the initial refinement levels. Fig. 17 exhibits the convergence of the functional for the hp-adaptive case in terms of DOF. It can be observed that the final output functional is accurately predicted by the corrected functional on the second and third adaptation cycles.

5.2. Hypersonic flow over a half-circular cylinder

The next test case involves the computation of hypersonic flow over a half-circular cylinder with a free-stream Mach number of 6 using the hp-adaptive scheme. For high-speed flows, it is well known that limiters or added artificial dissipation are required in order to robustly capture strong shocks with higher-order discretizations. On the other hand, the amount of dissipation added by a first-order spatial discretization scheme is sufficiently large to handle any shock using an appropriate Riemann solver. However, the dissipation added by a first-order accurate scheme is proportional to the element size, h, and therefore large numbers of small elements are required for high accuracy. Thus, the idea of the hp-refinement approach for



:ed mesh, 21372 elements, discretization orders (a) 1^{st} adapt
$ax = 1$) ($p_{min} = 0, p_m$

ed mesh, 29100 elements, discretization orders (b) 2^{nd} adapt
$x = 2$) ($p_{min} = 0, p_{ma}$

) 3^{rd} adapted mesh, 42234 elements, discretization orders (c
$_{min} = 0, p_{max} = 3$) (p

Fig. 19. hp-adapted meshes with the distributions of orders of discretization for hypersonic flow ($M_\infty = 6$) over a half-circular cylinder, using the target functional of integrated surface temperature.

problems with strong shocks is to provide enhanced accuracy through mesh refinement in shock regions identified by a smoothness indicator, and simultaneously to increase discretization orders in smooth areas to improve functional accuracy.

This test case employs integrated temperature on the cylinder surface as the objective functional due to the fact that surface heating is of significant interest for hypersonic vehicles, and the error tolerance of the functional output is specified as 5×10^{-4}. A first-order accurate scheme ($p = 0$) is used initially to ensure a stable solution, and the proposed *hp*-adaptive refinement strategy is implemented in areas where the spatial functional error has relatively large contributions. The switch between mesh subdivision (i.e. *h*-refinement) and variation of discretization order (i.e. *p*-enrichment) is based on the inter-element smoothness indicator discussed in Section 4.3.2. The far-field inlet boundary of the mesh is treated using a fully prescribed supersonic inflow condition. The inviscid fluxes of the flow solver are calculated using the Lax–Friedrichs flux function [25] and the flow solution obtained at the previous adaptation cycle is used as an initial condition for the next finer refinement cycle. In this test case, we examine the resolved shocks and the functional accuracy convergence throughout the adaptation process.

The adaptation starts with an initial mesh shown in Fig. 18(a) and a uniform $p = 0$ order of discretization. Since the mesh in shock regions is relatively coarse, discontinuities are not well resolved (spreading over 4–5 elements), as observed from the Mach number solution illustrated in Fig. 18(b). Fig. 19 illustrates the adapted meshes together with the distributions of discretization orders for all refinement levels. We observe that no refinement takes place ahead of the shock due to the fact that the uniform flow ahead of the shock wave is exactly represented by the first-order ($p = 0$) discretization in this region. Substantial *h*-refinement occurs around/in the shock regions where a first-order accurate scheme, represented by the white color on the plots, is maintained throughout all the *hp*-adaptation cycles, and higher-order discretizations are prescribed behind the shock and ahead of the cylinder. The final adapted mesh illustrated in Fig. 19(c) contains a dense distribution of *h*-refinement in shock areas and a range of discretization orders from $p = 0$ to $p = 3$ in the region between the shock and cylinder. It is also noted that the entire shock region is not refined, since lateral regions of the shock wave which are not close to the cylinder have little influence on the functional of interest. We also observe that a thin shock profile upstream of the cylinder is properly captured and resolved on the final *hp*-adapted mesh as illustrated by the Mach number contours in Fig. 20. Fig. 21 compares the shock profiles on the initial mesh and each of the three adapted meshes as a plot of pressure along the

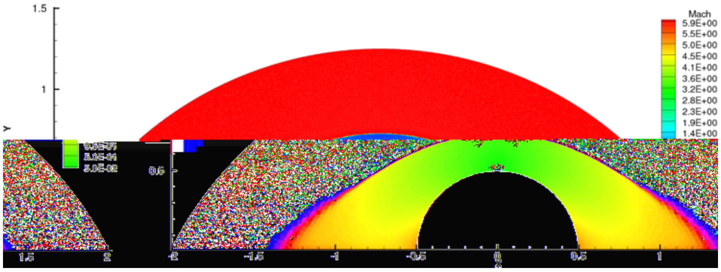


Fig. 20. Mach number contours on the final *hp*-adapted mesh for hypersonic flow ($M_\infty = 6$) over a half-circular cylinder.

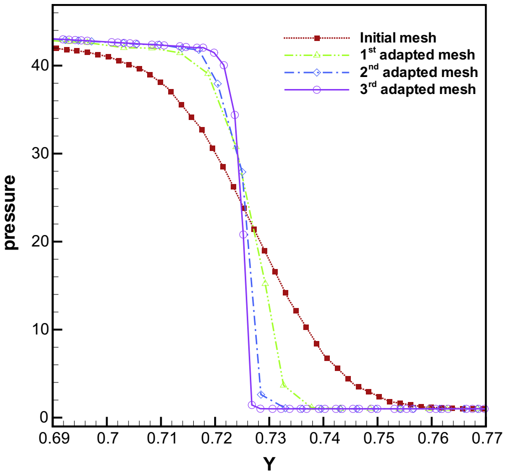


Fig. 21. Comparisons of pressure profiles at the centerline $x = 0$ between the initial and *hp*-adapted meshes for hypersonic flow over a half-circular cylinder ($M_\infty = 6$).

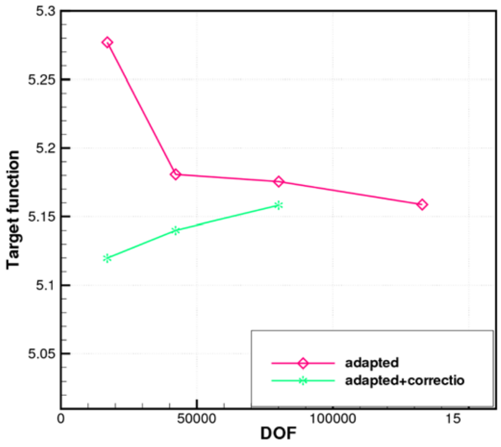


Fig. 22. Comparison of functional convergence in terms of degrees of freedom for hypersonic flow over a half-circular cylinder ($M_\infty = 6$), using the target functional of integrated surface temperature.

centerline $x = 0$. The original shock profile crosses a very wide distance, while the shock computed on the final adapted mesh is around 10 times thinner than the one resolved on the initial mesh. Note that in order to obtain a similar scale of the shock profile, the original mesh would require three successive full refinements, which produces a mesh with over 1 million degrees of freedom, whereas the number of unknowns in terms of degrees of freedom on the final adapted mesh is only about 6% of this triple-uniformly refined mesh.

Fig. 22 depicts the functional convergence for this test case. Although on the initial mesh ($p = 0$) the corrected functional represented by the first point (star symbol) of the green line does not predict the finer level functional very well, it does provide a notable improvement in the correct direction. Furthermore, the difference between the adapted functional value and corrected functional value significantly decreases with increasing adaptation cycles, and the last (third) corrected functional value predicts the final functional value very accurately.

6. Conclusions and future work

The adjoint-based error estimation and adaptivity techniques presented in this work hold great promise for improving the accuracy and reliability of engineering simulations where a small number of output objectives are of interest. The proposed *h*- or *p*-refinement algorithms demonstrate more efficient capabilities of error reduction than uniform mesh or order refinement, in terms of both degrees of freedom and computational cost. The *hp*-adaptation algorithm achieves exponential convergence which is faster than the pure *p*-enrichment algorithm, and furthermore, for shock problems where the pure *p*-enrichment scheme is not able to perform adequately, the *hp*-adaptation scheme demonstrates good capturing of strong shocks or discontinuities with improved accuracy. The key technique to assure proper performance of the *hp*-adaptive refinement scheme is to enable the proper choice between *p*-enrichment for smooth flow regions and *h*-refinement for non-smooth regions or in areas of discontinuities. It is notable that this approach enables the accurate solution of flows with strong shocks without the use of slope limiters, which can lead to loss of numerical convergence. However, the use of slope limiters or additional artificial dissipation may be useful in future work for enhancing overall accuracy and robustness. Future work will concentrate on these aspects of adaptive methods for high-speed flows, as well as more effective smoothness indicators, error tolerance prescriptions, and the extension to temporal adaptation using time-dependent adjoint methods.

Acknowledgments

This work was supported by NASA Grant NX07AC31A and AFOSR contract FA9550-09-C-0021.

References

[1] G.E. Barter, D.L. Darmofal, Shock capturing with higher-order, PDE-based artificial viscosity, AIAA Paper 2007-3823, June 2007.
[2] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
[3] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Appl. Mech. Eng. 35 (1998) 2440–2463.
[4] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, SIAM J. Sci. Comput. 16 (2001) 173–261.
[5] K.J. Fidkowski, T.A. Oliver, J. Lu, D. Darmofal, *p*-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 207 (2005) 92–113.
[6] H. Luo, J.D. Baum, R. Lohner, A *p*-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, J. Comput. Phys. 211 (2) (2006) 767–783.
[7] C.R. Nastase, D.J. Mavriplis, High-order discontinuous Galerkin methods using an *hp*-multigrid approach, J. Comput. Phys. 213 (1) (2006) 330–357.

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

[9] L. Wang, D.J. Mavriplis, Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations, J. Comput. Phys. 225 (2) (2007) 1994–2015.

[10] D. Estep, A posteriori error bounds and global error control for approximation of ordinary differential equations, SIAM J. Numer. Anal. 32 (1995) 1–48.

[11] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numer. 10 (2002).

[12] P. Houston, R. Hartmann, E. Suli, A posteriori error analysis for stabilised finite element approximations of transport problems, Comput. Methods Appl. Mech. Eng. 190 (11–12) (2000) 1483–1508.

[13] P. Houston, E. Suli, hp-Adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems, SIAM J. Sci. Comput. 23 (4) (2001) 1226–1252.

[14] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, J. Comput. Phys. 183 (2) (2002) 508–532.

[15] K.J. Fidkowski, D. Darmofal, A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 225 (2) (2007) 1653–1672.

[16] N.A. Pierce, M. Giles, Adjoint and defect error bounding and correction for functional estimates, J. Comput. Phys. 200 (2) (2004) 769–794.

[17] D.A. Venditti, D.L. Darmofal, Grid adaptation for functional outputs: application to two-dimensional inviscid flows, J. Comput. Phys 176 (1) (2002) 40–69.

[18] T. Barth, Numerical Methods and Error Estimation for Conservation Laws on Structured and Unstructured Meshes, Lecture Notes, Series: 2003–04, von Karman Institute for Fluid Dynamics, Brussels, Belgium, March 2003.

[19] M. Nemec, M.J. Aftosmis, M. Wintzer, Adjoint-based adaptive mesh refinement for complex geometries, AIAA Paper 2008-725, January 2008.

[20] P. Houston, E. Suli, A note on the design of hp-adaptive finite element methods for elliptic partial differential equations, Comput. Methods Appl. Mech. Eng. 194 (2–5) (2005) 229–243.

[21] C. Mavriplis, Adaptive mesh strategies for the spectral element method, Comput. Methods. Appl. Mech. Eng. 116 (1994) 77–86.

[22] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Appl. Numer. Math. 48 (3) (2004) 323–338.

[23] P. Batten, N. Clarke, C. Lambert, D.M. Causon, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. 18 (2) (1997) 1553–1570.

[24] E.F. Toro, M. Spruce, W. Spears, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1) (1994) 25–34.

[25] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, ICASE Report No. 97-65, NASA/CR-97-206253, 1997.

[26] J.C.-C. Lu, An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method, Doctoral Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2005.

[27] P. Solin, K. Segeth, I. Dolezel, High-Order Finite Element Methods, Studies in Advanced Mathematics, Chapman and Hall, 2003.

[28] C.R. Nastase, D.J. Mavriplis, A parallel *hp*-multigrid solver for three-dimensional discontinuous Galerkin discretizations of the Euler equations, AIAA Paper 2007-0512, January 2007.

[29] D.A. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle, Int. J. Numer. Meth. Eng. 21 (6) (1985) 1129–1148.

[30] D.A. Dunavant, Economical symmetrical quadrature rules for complete polynomials over a square domain, Int. J. Numer. Meth. Eng. 21 (10) (1985) 1777–1784.

[31] D.J. Mavriplis, Unstructured grid techniques, Annu. Rev. Fluid. Mech. 29 (1) (1997) 473–514.

[32] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (2) (1998) 199–224.