INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2017; **00**:1–34
Published online in Wiley Online Library (www.onlinelibrary.wiley.com). DOI: 10.1002/fld.4396

An *hp*-Adaptive Unstructured Finite Volume Solver for Compressible Flows

Alireza Jalali*
Carl Ollivier-Gooch†
Department of Mechanical Engineering
The University of British Columbia
Vancouver, BC, V6T 1Z4, Canada

Received . . .

1. INTRODUCTION

Higher-order discretizations of fluid dynamic equations have received a great deal of attention due to their potential advantages in obtaining more accurate solutions with lower cost. However, higher-order accuracy is only obtained in the smooth regions of the solution where there is no discontinuity in the solution or gradient. In aerodynamic applications, several sources of discontinuities exist, such as shocks, contact discontinuities, and the turbulence working variable at the edge of boundary layer (e.g. in Spalart-Allmaras turbulence model). These can deteriorate the order of accuracy in non-smooth regions and also cause solver failure in some cases. This motivates the idea of simultaneous mesh refinement (h-refinement) and order enrichment (p-enrichment) in higher-order compressible flow solvers. In other words, the order of solution approximation can be increased in those parts of the domain where the solution is smooth whereas the mesh resolution is enhanced in non-smooth regions in which a lower-order discretization is employed.

For compressible flow simulations, the combination of such a strategy with output-based adaptation (known as *hp*-adaptation) has been used by several Discontinuous Galerkin (DG) solvers where the number of degrees of freedom increases rapidly with the order of discretization. In this way, it is possible to optimally place degrees of freedom within a problem and achieve the required accuracy with minimal costs [1, 2, 3]. In addition, solver robustness is enhanced because a first-order discretization of flow field variables is employed near a shock wave without using slope limiters or artificial dissipation [4, 5]. *hp*-adaptive methods have been successfully applied to the other variants of Galerkin-based schemes such as hybridized-DG [6] and Petrov-Galerkin [7, 8].

In contrast to compact schemes such as DG and its variants, higher-order finite volume discretizations extend the stencil to obtain a more accurate estimate of the solution. In such schemes, there is no coupling between the number of degrees of freedom and order of discretization as the number of control volumes remains constant on a mesh with a fixed number of elements regardless of the order of accuracy. Nevertheless, a higher-order polynomial for solution approximation reconstructs a larger number of derivatives at each iteration [9]. Also, when using an implicit time advance scheme for the convergence to the steady-state solution, higher-order methods lead to a larger number of non-zero entries in the Jacobian matrix. This increases the memory usage and computational cost due to a larger storage requirement and also solving a denser linear system [10, 11] which must be preconditioned by a factorization method (e.g. Incomplete LU). As a result, *hp*-adaptation methodology, which has originally been designed for compact discretization schemes, also has potential advantages for higher-order finite volume flow solvers. In this way, the extra number of derivatives are only reconstructed in those regions where needed and the Jacobian matrix becomes sparser with a smaller bandwidth. Furthermore, this strategy is capable of automatic limiting for flows with inherent discontinuities as we can start from a lower-order solution everywhere and increase the order of polynomial in smooth regions while the mesh is refined near discontinuities. This can be viewed as an alternative to typical limiting approaches where the slope limiter, though designed to be active only near discontinuities, may deteriorate solution accuracy even in smooth regions [12] and/or hamper the convergence to the steady-state due to non-differentiability [13].

The goals of this paper are to extend the idea of *hp*-adaptation to unstructured finite volume methods and to demonstrate the capabilities of this scheme by applying it to two-dimensional compressible flow problems. In particular, we adapt the mesh resolution and discretization order based on an estimate of error in the computed solution at each level of refinement.

The quality of the adaptive schemes relies on the accuracy of error estimates. The conventional method of adaptation is based on feature-based criteria that highlight the distinctive features such as shock waves and boundary layers in the flow field [14, 15, 16]. This approach, which is simple and effective in some CFD applications, requires trial and error to determine the appropriate flow features and thus fails to provide a general and robust error estimate [17, 18]. Alternatively, the adaptation procedure can be carried out using residual-based approaches in which the truncation error of the fluid flow quantities constructs the adaptation criteria [19, 20, 21]. Moreover, output-based adaptation criteria based on solutions to the so-called adjoint (dual) problem, which is derived for an output of interest, have become very mature in recent years [22]. In this type of error estimates, the solution of the adjoint problem is multiplied by the local contribution of the truncation error estimate to provide information about the interaction of the error in different components of the solution [23] and subsequently those locations that require more resolution (typically h-refinement) for a more accurate estimate of the output quantity [24, 25]. It has been shown that the inclusion of the adjoint solution improves the effectiveness of the adaptation procedure over the traditional residual-based approaches [18]. These techniques have been used to perform adaptive mesh refinement (only h-refinement) in second-order unstructured finite volume methods for inviscid [26] and viscous laminar flows [18]. In addition, they have been used in the context of DG methods for inviscid [27], viscous laminar [28] and turbulent RANS [29, 30, 31] simulations.

In our work, we use residual-based and adjoint-based error estimation methods for hp-adaptation in our unstructured finite volume solver. In the former approach, we employ a higher-order residual operator to estimate the truncation error of a lower-order discretization scheme. The magnitude of the estimated truncation error is used as a local error indicator for h- or p-refinement. For the adjoint-based hp-adaptation, we compute the discrete adjoint solution obtained by a single linear system solve at each refinement level and multiply the adjoint solution by the higher-order estimate of the truncation error to find a different error indicator. The adjoint problem is formed by evaluating one order higher residual and Jacobian operators based on the lower-order control volume averages. In either approach, a certain fraction of control volumes that contribute most to the total error are flagged for refinement. The decision for h-refinement versus p-enrichment is based on local smoothness of the primal problem. It is worth mentioning that we allow for non-conforming interfaces (i.e., hanging nodes) in the mesh after h-refinement, allowing us treat triangles and quadrilaterals in the same way.

In what follows, we describe the governing equations for compressible flow problems in Section 2 followed by an overview of our higher-order unstructured finite volume solver in Section 3. The error indicators used for hp-adaptations are discussed in Section 4 and our hp-adaptation methods including h-refinement and p-enrichment in isolation and how they are combined together are explained in Section 5. We present our numerical results for inviscid and viscous compressible flow problems in Section 6 to highlight the efficiency and accuracy advantages obtained by hp-adaptation compared to uniform refinement of second- and higher-order accurate discretizations. Finally, Section 7 summarizes the paper and recommends future work.

2. GOVERNING EQUATIONS

The governing equations for compressible flows can be re-arranged as:

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

where U represents the vector of conserved variables, $\vec{F}_c$ is the vector of convective fluxes, $\vec{F}_v$ is the vector of viscous fluxes and S is the source term. For turbulent flows, we couple the conservation of mass, momentum and energy to the negative variant of the one equation turbulence model developed by Spalart and Allmaras (SA model) [32]. The solution, flux and source term vectors for the coupled system of RANS-SA in two-dimensions are given as:

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E_t \\ \rho\tilde{\nu} \end{pmatrix}, \quad F_c^x = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ u\left(E_t + P\right) \\ \rho u\tilde{\nu} \end{pmatrix}, \quad F_c^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ v\left(E_t + P\right) \\ \rho v\tilde{\nu} \end{pmatrix}$$

$$F_v^x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + c_p\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial x} \\ \frac{1}{\sigma}\left(\mu + f_n\rho\tilde{\nu}\right)\frac{\partial\tilde{\nu}}{\partial x} \end{pmatrix}, \quad F_v^y = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} + c_p\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial y} \\ \frac{1}{\sigma}\left(\mu + f_n\rho\tilde{\nu}\right)\frac{\partial\tilde{\nu}}{\partial y} \end{pmatrix}$$

$$S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho\left(P_t - D_t\right) + \frac{1}{\sigma}c_{b2}\rho\nabla\tilde{\nu}\cdot\nabla\tilde{\nu} - \frac{1}{\sigma}\left(\nu + \tilde{\nu}\right)\nabla\rho\cdot\nabla\tilde{\nu} \end{pmatrix} \tag{2}$$

where ρ is the fluid density, $\vec{V} = (u, v)$ are the Cartesian velocity components, E_t is the total energy, $\tilde{\nu}$ is the turbulence model working variable, P is the fluid pressure, μ is the dynamic viscosity, μ_T is the turbulent viscosity, ν is the kinematic viscosity, c_p is the specific heat at constant pressure, T is the fluid temperature, Pr and Pr_T are the Prandtl and turbulent Prandtl numbers respectively and τ_{ij} is the total viscous stress tensor including the Boussinesq approximated Reynolds stresses:

$$\tau_{ij} = 2\left(\mu + \mu_T\right)\dot{\gamma}_{ij} \tag{3}$$

$$\dot{\gamma}_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial V_k}{\partial x_k}\delta_{ij}$$

The fluid pressure can be related to the total energy by introducing the ideal gas equation of state given as:

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

where γ is the ratio of specific heats. We assume that the working fluid is air, $\gamma = 1.4$, $Pr = 0.72$ and $Pr_T = 0.9$.

Moreover, P_t and D_t denote the production and destruction of the turbulent viscosity; σ, c_{b2} and f_n are constants and functions related to its diffusion all given by Ref. [32]. For viscous laminar flow problems, the turbulence model equation is ignored and $\mu_T = 0$.

For inviscid flows, the Euler equations which are comprised of only convective fluxes ($F_v^x = F_v^y = S = 0$) are solved.

3. FINITE VOLUME FLOW SOLVER

For our computations, we use a higher-order unstructured finite volume solver. The elements of this solver including the spatial discretization and implicit solution strategy used to obtain the steady-state solution are described in this section briefly [33, 10].

3.1. Spatial discretization

To discretize the flow equations using the finite volume methods, we integrate the governing equations over each control volume and use the divergence theorem. In 2D, this process gives the finite volume discretization as:

$$A_i \frac{d\overline{U}_i}{dt} = -\oint_{CS_i} \left(\vec{F}_c\left(U\right) - \vec{F}_v\left(U, \nabla U\right)\right) \cdot \hat{n}\, ds + A_i \overline{S}_i \tag{5}$$

where $\overline{U}_i$ and $\overline{S}_i$ are the averages of the solution and source term vectors over control volume i, A_i is the control volume area, $\hat{n}$ is the outward unit normal vector at the surface of control volume and ds represents the infinitesimal length along the control volume boundary. The spatial accuracy of the discretization depends on the accuracy of solution reconstruction from which the flux integral and averaged source term are computed. In our solver, we use the K-exact least-squares solution reconstruction where each primitive variable of the flow field, ϕ, is approximated within a control volume by a piecewise polynomial about the control volume's reference point (x_i, y_i) as:

$$\begin{aligned} \phi_i^R\left(x,y\right) &= \left.\phi\right|_i + \left.\frac{\partial \phi}{\partial x}\right|_i \left(x - x_i\right) + \left.\frac{\partial \phi}{\partial y}\right|_i \left(y - y_i\right) + \left.\frac{\partial^2 \phi}{\partial x^2}\right|_i \frac{\left(x - x_i\right)^2}{2} \\ &\quad + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_i \left(x - x_i\right)\left(y - y_i\right) + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_i \frac{\left(y - y_i\right)^2}{2} + \ldots \end{aligned} \tag{6}$$

Note that a $(p+1)$-order accurate spatial discretization requires a polynomial degree of p. The coefficients of the Taylor series expansion are computed by conserving the mean value in the control volume and minimizing the error in predicting the mean value of nearby control volumes. This is obtained by solving a least-squares problem in which the conservation of mean acts as a hard constraint [9, 34]. For higher-order reconstruction, more neighboring control volumes are needed in the stencil. The control volumes are chosen based on their topological proximity to the reconstruction control volume as all neighbors at a given level (one layer of neighbors) are added together until the number of control volumes in the stencil reaches the number requested for each order of accuracy. The same strategy is used for boundary control volumes although they typically need more layers of neighbors to reach the requested number [9]. To compute the flux vector, we use Roe's scheme for convective fluxes [35]; we average the gradients and add a jump term for viscous fluxes [36]. The flux integral and averaged source term are evaluated by sufficiently accurate quadrature rules and Equation 5 can be re-written as:

$$A_i \frac{d\overline{U}_i}{dt} = -R\left(\overline{U}\right) \tag{7}$$

where $R\left(\overline{U}\right)$ is the discrete residual operator computed based on the control volume averages.

To implement the boundary conditions, the no-slip boundary conditions on the walls are applied by adding an extra constraint to the least-squares system of solution reconstruction for each boundary Gauss point [9] for viscous problems. In addition, the slip wall (for inviscid flows) and adiabatic wall condition are applied weakly by zeroing the momentum and heat flux components in Equation 2 for wall boundary faces. For inflow, we specify the values of total pressure, velocity components and turbulent viscosity while the static pressure is the only pre-specified condition at the outflow; other quantities are obtained from solution reconstruction. No point vortex correction is applied for the far-field boundaries.

3.2. Solution strategy

To find the steady-state solution for the system of governing equations, $R\left(\overline{U}\right) = 0$, it is possible to apply Newton's method directly. However, Newton's method will diverge if the initial guess is far from the real solution. Therefore, the linear system is augmented by a damping term which mimics the time derivative in the original time-dependent equations and prevents the evolution of non-physical solution at each iteration. The non-linear system of equations can be linearized at any time level n as:

$$\left[\frac{I}{\Delta t_i / A_i} + \frac{\partial R}{\partial \overline{U}}\right] \Delta\overline{U} = -R\left(\overline{U}^n\right) \tag{8}$$

where $\frac{\partial R}{\partial \overline{U}}$ is the discrete flux Jacobian, calculated explicitly to full order [10] in our solver, I is the identity matrix; Δt_i is the local time step, and $\Delta\overline{U}$ is the solution update vector. For robustness, an under-relaxation factor, ω^n , is used to ensure a physical solution at the next iteration:

$$\overline{U}^{n+1} = \overline{U}^n + \omega^n \Delta\overline{U} \tag{9}$$

Since we seek the steady-state solution of the flow equations, local time-stepping with a global CFL number can be used as:

$$\Delta t_i^n = \frac{CFL^n \cdot h_i}{\lambda_{max,\, i}} \tag{10}$$

where h_i is a characteristic cell size and $\lambda_{max,\, i}$ is the eigenvalue of the convective flux Jacobian with the largest magnitude. Note that for the system of hyperbolic equations, the distinct eigenvalues of the normal convective flux Jacobian are:

$$\lambda_1 = \vec{V} \cdot \hat{n} - a \qquad \lambda_2 = \vec{V} \cdot \hat{n} \qquad \lambda_3 = \vec{V} \cdot \hat{n} + a \tag{11}$$

in which a is the sound speed, $\vec{V} = (u, v)$ is the velocity vector and $\hat{n}$ is the outward unit normal vector at interfaces.

As we proceed, the *CFL* number must be increased to make the convergence to the steady-state solution faster. For this purpose, we use the exponential progression with under-relaxation strategy [37]:

$$CFL^{n+1} = \begin{cases} \beta \cdot CFL^n & \omega^n = 1 \\ CFL^n & \omega_{min} < \omega^n < 1 \\ \kappa \cdot CFL^n & \omega^n < \omega_{min} \end{cases} \tag{12}$$

Here, the parameters are set to $\beta = 1.5$, $\kappa = 0.1$ and $\omega_{min} = 0.01$.

To find the under-relaxation factor at each iteration, ω^n, a line search algorithm is incorporated with the implicit solver to minimize the norm of the unsteady residual vector defined as:

$$R_t\left(\tilde{U}\right) = \frac{I}{\Delta t}\left(\tilde{U} - \overline{U}^n\right) + R\left(\tilde{U}\right) \tag{13}$$

where $\tilde{U}$ is a future trial solution. It is possible to show that there exists a trial solution $\tilde{U}$ along the update vector direction $\Delta\overline{U}$ such that $\left\|R_t\left(\tilde{U}\right)\right\|_2 < \left\|R\left(\overline{U}^n\right)\right\|_2$. Therefore, the under-relaxation factor can be chosen such that the sufficient decrease condition is satisfied [37]. We start with the initial guess of $\omega^n = 1$ and check if the new solution state is physical *and* satisfies the decrease condition. If not, we halve the under-relaxation factor every time until the both conditions are satisfied. The physicality of the solution state is verified by checking the positivity of density and pressure at all quadrature points.

4. ERROR ESTIMATION

For adaptation, a reliable error indicator is required. For residual-based adaptation, a local estimate of the error in flow field quantities is used whereas this estimate is weighted by the solution of an adjoint problem for output-based adaptations. In our work, we use a higher-order operator to obtain an estimate of the truncation error.

Consider the following continuous non-linear problem for which $\overline{U}$ is the exact solution:

$$R\left(\overline{U}\right) = 0 \tag{14}$$

A lower-order discrete approximation of the exact solution, $\overline{U}_{p-1}$, satisfies the lower-order discrete non-linear residual:

$$R_{p-1}\left(\overline{U}_{p-1}\right) = 0 \tag{15}$$

The exact truncation error for the lower-order discrete problem is defined as the amount by which the discrete lower-order solution does not satisfy the continuous PDE. Such an error property can be estimated by applying a higher-order discrete operator, R_p, on the lower-order solution [23, 38]:

$$R\left(\overline{U}_{p-1}\right) \approx R_p\left(\overline{U}_{p-1}\right) \tag{16}$$

Therefore, the local contribution of control volume i to the total error can be obtained by the norm of the estimated truncation error in the corresponding control volume:

$$\epsilon_i = \left| R_p \left(\overline{U}_{p-1} \right) \right|_i \tag{17}$$

For adjoint-based adaptation, the error involved in the computation of an output functional based on a lower-order solution, $\mathcal{J}_{p-1}\left(\overline{U}_{p-1}\right)$, is estimated. For this purpose, a higher-order estimate of the output functional based on a higher-order solution can be obtained by writing a Taylor series expansion about the lower order solution projected to the higher-order space, $\overline{U}_{p-1\to p}$. Note that in the context of finite volume methods, the lower-order solution containing the control volume averages can be identically mapped into the higher-order space ($\overline{U}_{p-1\to p} = \overline{U}_{p-1}$). Therefore, the higher-order functional can be expanded as:

$$\mathcal{J}_p\left(\overline{U}_p\right) = \mathcal{J}_p\left(\overline{U}_{p-1}\right) + \left(\frac{\partial \mathcal{J}_p}{\partial \overline{U}_p}\right)_{\overline{U}_{p-1}} \left(\overline{U}_p - \overline{U}_{p-1}\right) + ... \tag{18}$$

where $\mathcal{J}_p\left(\overline{U}_{p-1}\right)$ and $\left(\frac{\partial \mathcal{J}_p}{\partial \overline{U}_p}\right)_{\overline{U}_{p-1}}$ are the higher-order functional and its sensitivity with respect to the higher-order solution both evaluated at the lower-order solution state. Considering that a higher-order solution is not in hand, a Taylor series expansion can be used for the higher-order residual to eliminate $\left(\overline{U}_p - \overline{U}_{p-1}\right)$ in Equation 18:

$$R_p\left(\overline{U}_p\right) = R_p\left(\overline{U}_{p-1}\right) + \left(\frac{\partial R_p}{\partial \overline{U}_p}\right)_{\overline{U}_{p-1}} \left(\overline{U}_p - \overline{U}_{p-1}\right) + ... \tag{19}$$

The higher-order solution satisfies its corresponding discrete operator, $R_p\left(\overline{U}_p\right) = 0$. So Equation 19 can be re-arranged to solve for the higher-order unknown solution as:

$$\left(\overline{U}_p - \overline{U}_{p-1}\right) \approx -\left(\frac{\partial R_p}{\partial \overline{U}_p}\right)^{-1}_{\overline{U}_{p-1}} R_p\left(\overline{U}_{p-1}\right) \tag{20}$$

Substituting Equation 20 into Equation 18 yields the following expression for the estimate of the error in the functional:

$$\mathcal{J}_p\left(\overline{U}_p\right) - \mathcal{J}_p\left(\overline{U}_{p-1}\right) \approx -\left(\frac{\partial \mathcal{J}_p}{\partial \overline{U}_p}\right)_{\overline{U}_{p-1}} \left(\frac{\partial R_p}{\partial \overline{U}_p}\right)^{-1}_{\overline{U}_{p-1}} R_p\left(\overline{U}_{p-1}\right) \tag{21}$$

Next, the higher-order discrete adjoint solution is defined as the variable Z_p such that:

$$\left(\frac{\partial R_p}{\partial \overline{U}_p}\right)^{T}_{\overline{U}_{p-1}} Z_p = \left(\frac{\partial \mathcal{J}_p}{\partial \overline{U}_p}\right)^{T}_{\overline{U}_{p-1}} \tag{22}$$

Note that both sides of Equation 22, which show the sensitivity of the higher-order residual and functional to the higher-order solution, are evaluated at the available lower-order solution. The functional error can be re-written in terms of the discrete adjoint solution as:

$$\mathcal{J}_p\left(\overline{U}_p\right) - \mathcal{J}_p\left(\overline{U}_{p-1}\right) \approx -Z_p^T\, R_p\left(\overline{U}_{p-1}\right) \tag{23}$$

For adaptation purposes, we require a local error indicator. Therefore, the magnitude of the contribution to the functional error from a particular control volume i can be approximated as:

$$\epsilon_i = \left|Z_p^T\, R_p\left(\overline{U}_{p-1}\right)\right|_i \tag{24}$$

5. ADAPTATION METHODS

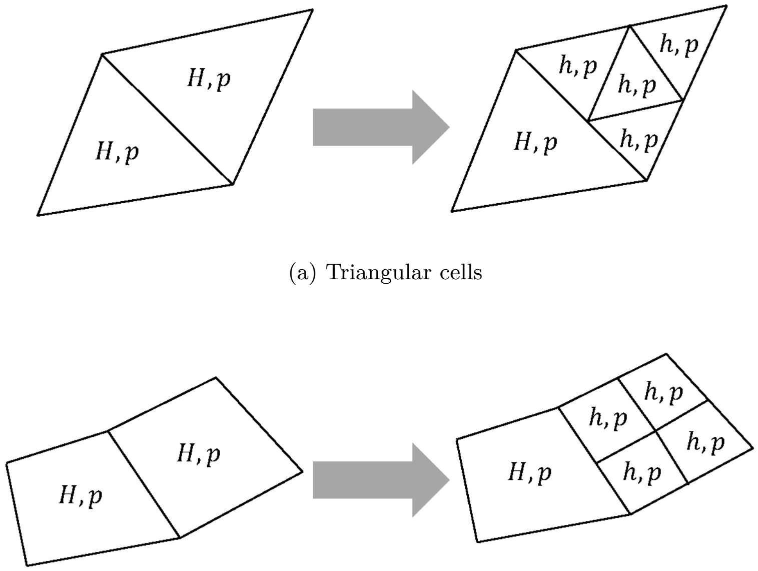
In our *hp*-adaptive process, we use both error indicators of Equation 17 and 24 to compare the effectiveness of the two estimates. Following Ceze and Fidkowski [2], we flag a certain percentage of control volumes, f^{adapt}, with the largest error indicators for refinement at each step of the adaptive procedure. For this purpose, a sorted list of the control volumes based on the value of their corresponding error indicator from the highest to the lowest is created. A loop over the list is executed and the control volumes are flagged for *p*-enrichment or *h*-refinement procedure where appropriate until the pre-specified target $(N_{CV} \times f^{adapt})$ is reached. This ensures that only the control volumes with the highest error magnitude are refined for highly non-uniform error distributions.

As described, accuracy enhancement in each control volume can be attained by decreasing the mesh size (h) or increasing the order of accuracy (p). In what follows, we discuss the details of each method in isolation and then how they are combined to give the hybrid *hp*-adaptive scheme for our unstructured finite volume solver.

5.1. h-*refinement*

To be able to handle triangular and quadrilateral cells in the same way, we use an isotropic non-conforming mesh refinement framework where hanging nodes are allowed regardless of the type of elements. In such a refinement, a triangle or quadrilateral is divided into four sub-cells which almost have the same size. A triangle is refined using mid-point subdivision so that a node is placed at the mid-point of the faces. An additional node is required at the center of the cell for a quadrilateral. Figure 1 shows the refinement pattern for both types of the cells in a four-to-one basis recognizing that the sub-cells created through refinement inherit the approximation order from the original cell.

The existence of hanging nodes complicates the computation of flux integrals and also selection of reconstruction stencils. Any face connecting two vertices is identified as a unique face on which integration quadrature points are placed. In addition, the indices of the cells at the two sides of such a face are stored to retrieve the indices of the first layer of neighbors in the process of stencil selection. In this approach, a triangle with one hanging node is represented by four faces. Similarly, a quadrilateral with one hanging node is represented by five faces. As a result, the number of control volumes in the first layer of neighbors increases by the number of hanging nodes that exist in a cell.



(a) Triangular cells

(b) Quadrilateral cells

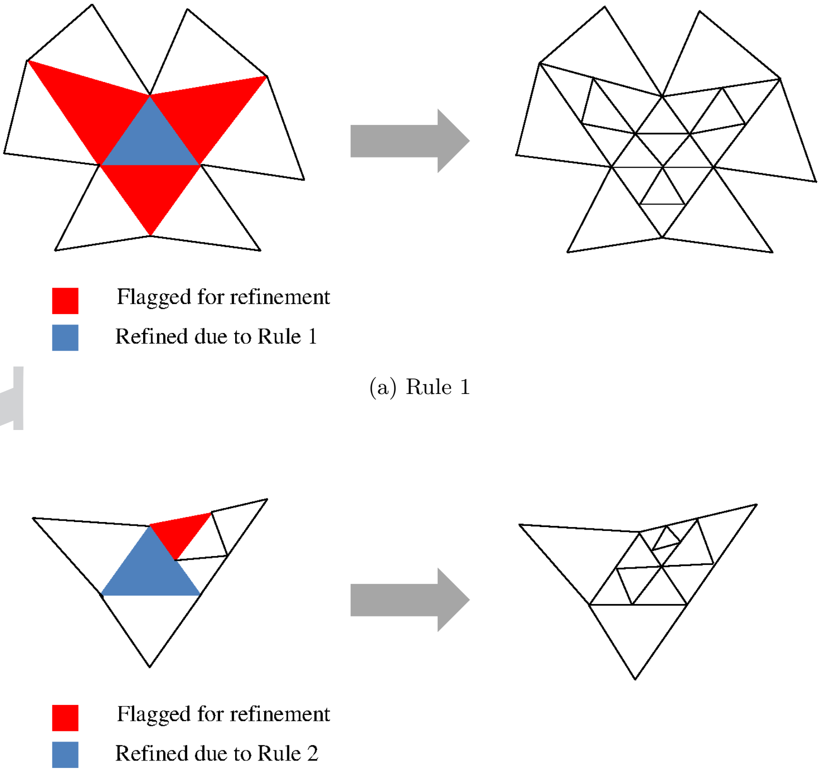
Figure 1. Schematic illustration of h-refinement pattern for 2D cells

To provide a smooth distribution of cell size throughout the mesh, we apply two standard rules for the refinement. First, a cell is automatically flagged for refinement if all of its faces are split due to the refinement of the neighbors. This rule has the advantage of limiting the number of hanging nodes to two and three for triangles and quadrilaterals, respectively in addition to removing potential un-refined holes from the mesh. The second rule enforces that all non-conforming faces have a 2:1 face length ratio by controlling the subdivision of half length non-conforming faces. If a fine cell with a half length face needs to be refined, the coarse cell on the other side containing the half length face must be divided as well to balance the face length ratio between adjacent cells. Figure 2 shows the implementation of the two rules for triangular cells.

5.2. p-*enrichment*

The implementation of p-enrichment is easier as it does not lead to any geometrical complexity and remains the same for triangular and quadrilateral cells. The primitive variables of the flow field are approximated throughout a control volume by a Taylor series expansion about the reference point (Equation 6). To enhance the accuracy by p-enrichment, the order of this polynomial is incremented by one and a larger least-squares system is solved to find the derivatives of the primitive variables at the reference point. To form the larger least-squares system, an extra layer of neighbors is added and a higher-order polynomial with more coefficients is reconstructed. Figure 3 shows the p-enrichment of a cell from $p = 1$ (second-order) to $p = 2$ (third-order) schematically.

We also apply some rules over the order of accuracy which mimic the rules used in h-refinement for uniform distribution of cell size and lead to uniform distribution of order



(a) Rule 1

(b) Rule 2

Figure 2. Schematic illustration of the h-refinement rules for triangular cells

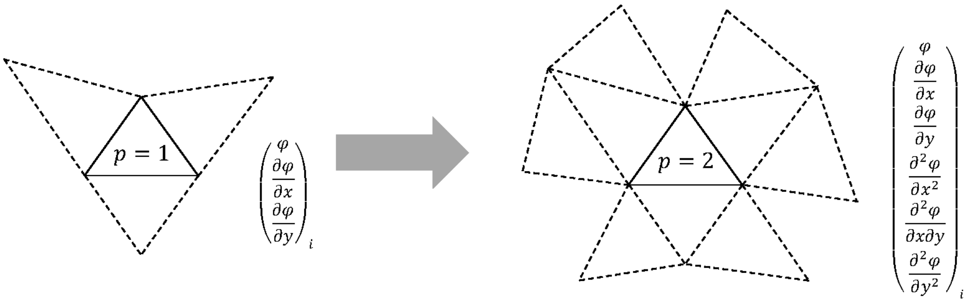

$p = 1$
$\begin{pmatrix} \varphi \\ \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \end{pmatrix}_i$
$p = 2$
$\begin{pmatrix} \varphi \\ \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \\ \frac{\partial^2 \varphi}{\partial x^2} \\ \frac{\partial^2 \varphi}{\partial x \partial y} \\ \frac{\partial^2 \varphi}{\partial y^2} \end{pmatrix}_i$

Figure 3. Schematic illustration of p-enrichment for an arbitrary cell from second-order to third-order

of accuracy throughout the mesh. First, if the order of accuracy in all neighboring cells is larger, we increment the order within the cell by one. Second, we modify the order of polynomial in neighboring control volumes not to end up with a jump of more than one order between adjacent cells. This is done by increasing the order of lower-order cells by one.

The number of integration quadrature points is determined based on the maximum order ($\max\left(p^{+}, p^{-}\right)$) of the two cells on each side of a face to ensure that the higher-order reconstructed polynomial is integrated sufficiently accurately along the faces.

5.3. hp-*refinement*

The final step in the adaptation procedure is the decision between *h*-refinement and *p*-enrichment. In finite element methods where the number of degrees of freedom scales with the order of accuracy, this decision is sometime made based on minimizing a cost function defined in terms of degrees of freedom [2]. Alternatively, this can be decided based on the smoothness of the primal solution so that the mesh size is reduced near discontinuities and order is increased in the smooth regions of the primal solution. To measure the level of solution smoothness, different smoothness indicators have been introduced in the literature [4]. One class of smoothness indicators, which is called element-wise indicator, assumes that the coefficients of smooth solution expansions have a similar decay rate as the Fourier coefficients. For finite element methods, it is possible to show that this decay rate scales as $\sim \frac{1}{p^4}$ [39]. So this indicator has been used extensively in several *hp*-adaptive finite element solvers (Petrov-Galerkin [8], discontinuous Galerkin [4] and hybridized discontinuous Galerkin [6]). In our work, which is based on finite volume discretization, the smoothness is determined by an inter-element jump indicator designed for shock detection in a DG solver [40]. This indicator has also been successfully used for the decision between mesh refinement and order enrichment in other *hp*-adaptive DG solvers [4, 5] and can be easily applied in our unstructured finite volume solver. The value of the jump for an arbitrary quantity of the flow field, ϕ, in a cell (cell i) is given as:

$$S_i = \frac{1}{|\partial\Omega_i|} \oint_{\partial\Omega_i} \left| \frac{\phi^{+} - \phi^{-}}{\frac{1}{2}\left(\phi^{+} + \phi^{-}\right)} \right| ds \tag{25}$$

where ϕ^{+} and ϕ^{-} are the reconstructed solutions at the two sides of a quadrature point, $|\partial\Omega_i|$ is the perimeter of the cell and ds is the infinitesimal length along the faces of the cell. For smooth solutions, the jump in the reconstructed solution and thus S_i are expected to be small whereas the indicator should return a value of $\mathcal{O}\left(1\right)$ near a sharp discontinuity such as a shock wave since the jump and average flow properties are of the same order of magnitude. Therefore, the choice between whether to apply *h*-refinement or *p*-enrichment is made by:

$$\begin{cases} S_i \geqslant \frac{1}{\mathcal{K}_\phi}, & h\text{-refinement} \\ S_i < \frac{1}{\mathcal{K}_\phi}, & p\text{-enrichment} \end{cases} \tag{26}$$

where $\mathcal{K}_\phi$ is a constant required to be large enough to capture discontinuities properly. In this work, we apply the jump indicator only on pressure for non-turbulent flows and on pressure *and* turbulence working variable of the SA model for the turbulent test case. The proper values of the jump indicator constants are found by several numerical experiments. In our solver, the values of of $200 \leq \mathcal{K}_P \leq 400$ for pressure and $50 \leq \mathcal{K}_{\tilde{\nu}} \leq 100$ for turbulence working variable seem to be effective choices for our problems.

Considering that our solver supports up to fifth-order discretizations ($p = 4$), once a control volume with the polynomial degree of $p_{max} = 3$ is flagged for extra refinement in smooth regions of the solution, we perform h-refinement instead of p-enrichment since we need one order higher discrete operator for error estimation (Equations 17 and 24).

6. NUMERICAL RESULTS

The *hp*-adaptation method proposed is evaluated for four different compressible flow problems: inviscid subsonic and transonic flows over the NACA 0012 geometry, turbulent subsonic flow over a flat plate and viscous laminar flow around the NACA 0012. In each case, the efficiency of the residual-based and adjoint-based *hp*-adaptive schemes are compared with second- and higher-order uniform refinements in terms of the number of degrees of freedom per equation and also CPU time required to obtain a grid-converged value of an output of interest. The presented number of degrees of freedom is the number of control volumes in each level and the wall clock time for the various test cases is based on simulations on a single core of an i7-4790 (3.60 GHz) CPU with memory of 16 GB (RAM) and cache levels of 8 KiB (L1), 256 KiB (L2) and 8 MiB (L3). In the adjoint-based adaptation method, the time includes the solution of both the primal and adjoint whereas only the primal solve time is included for the uniform refinement and residual-based adaptations.

6.1. Inviscid subsonic flow around NACA 0012

As our first test case, we consider the inviscid subsonic flow over the NACA 0012 with chord length of $c = 1$ and zero thickness at the trailing edge. The flow is characterized by the free stream Mach number of $Ma_\infty = 0.5$ and angle of attack of $\alpha = 2^\circ$. For residual-based adaptation, we use the L_2-norm of the truncation error obtained by the application of one order higher residual operator on all four components of the solution for each control volume. For adjoint-based adaptation, we consider the target functional of the pressure drag coefficient given by:

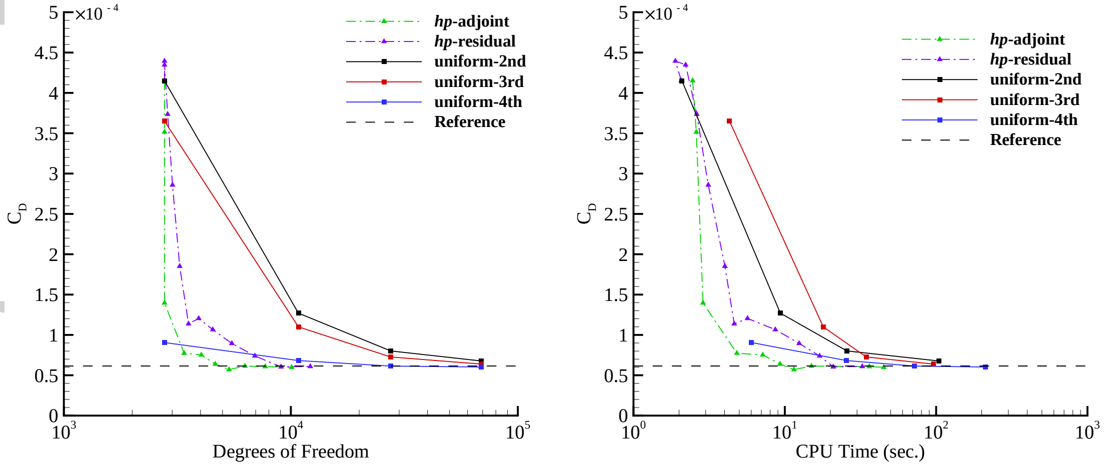
$$\mathcal{J}(U) = \int_{\partial\Omega} \psi \cdot (P\hat{n})\ ds \tag{27}$$

where $\hat{n}$ is the outward unit normal vector and $\psi = \frac{1}{C_\infty}(\cos\alpha, \sin\alpha)^T$along the wall boundaries and 0 everywhere else. C_∞ is a normalized reference value defined as:

$$C_\infty = \frac{1}{2}\gamma Ma_\infty P_\infty c \tag{28}$$

In the hypothetical case of infinitely far outer boundary, the drag force for this isentropic inviscid flow must converge to zero. However, we place the outer boundary $100c$ away from the wall boundaries and thus the pressure drag coefficient converges to a finite small value. The adaptation process starts with the second-order solution on a baseline coarse mesh with $2,776$ triangular cells. At each level of the adaptive procedure, $f^{adapt} = 15\%$ of control volumes with the largest error indicators are flagged for p-enrichment or h-refinement. Considering that the solution of this problem is smooth almost everywhere in the computational domain, we choose the upper bound of the pressure smoothness indicator, $\mathcal{K}_P = 200$, for the decision between p-enrichment and h-refinement.

The efficiency of the hp-adaptive refinement schemes is shown in Figure 4 in comparison with uniform refinement of different discretization order. A reference value for this problem is obtained by a fourth-order solution on a mesh with $197,891$ cells generated by uniform refinement. The adjoint-based adaptation outperforms the other refinement strategies as it converges to the grid converged drag coefficient after a few cycles with a smaller number of degree of freedom. The change in the values of the functional becomes small after 6 cycles of adaptations and remains almost constant in the last three consecutive cycles. The residual-based adaptation converges to the same value but after more refinement cycles and with a larger number of control volumes. In terms of the CPU time, the adjoint-based method takes slightly longer due to the extra adjoint solver. For this problem with a smooth solution, the fourth-order uniform refinement produces reasonably accurate output values with sufficient number of degrees of freedom which takes longer in term of CPU time. However, the second- and third-order uniform refinement procedures are far from the grid converged value except for the finest mesh with about $80,000$ control volumes.



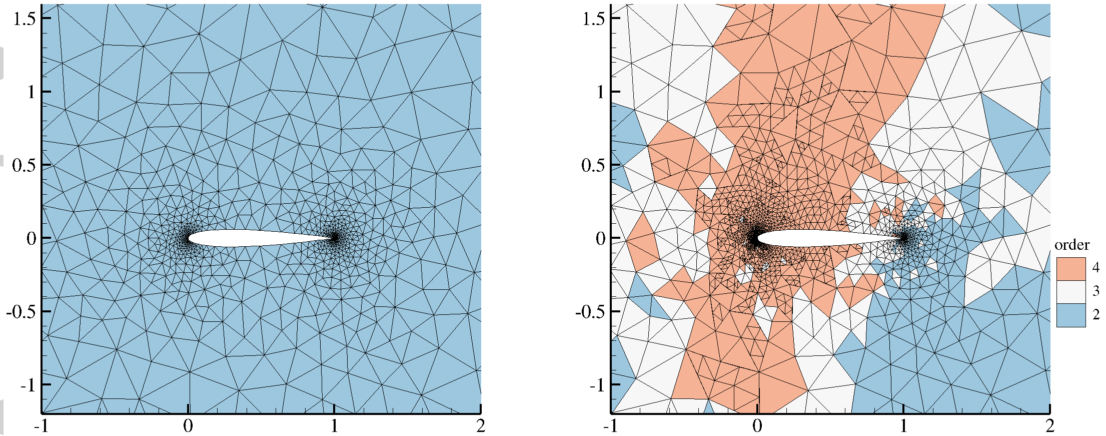
(a) Drag coefficient evolution with respect to degrees of freedom

(b) Drag coefficient evolution with respect to CPU time

Figure 4. Convergence of drag coefficient for inviscid subsonic test case ($Ma_\infty = 0.5$ and $\alpha = 2°$)

Figure 5 illustrates the baseline mesh and also the order of accuracy and mesh resolution after 6 cycles of the adjoint-based hp-adaptation for this test case. Note that the region near the leading edge in which the flow experiences high gradients towards the stagnation point is refined the most considering that the resolution of the initial mesh is not sufficient there. It is also worth-mentioning that the order of accuracy far from the wall remains equal to 2 which leads to less numerical complexity compared to uniformly higher-order discretizations.

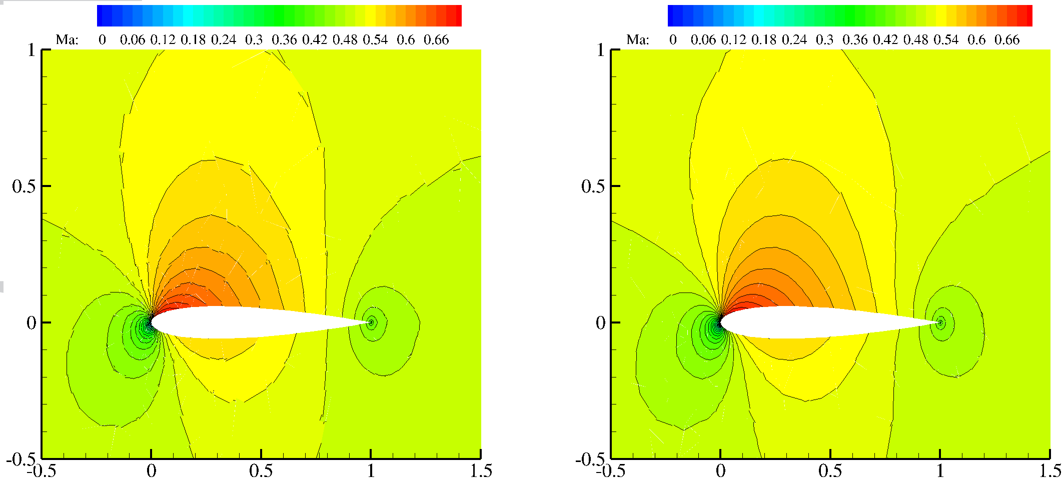
In addition, the contours of the Mach number on the two meshes with the prescribed order of accuracy for each control volume is shown in Figure 6. As expected, the hp-adapted mesh produces a smoother distribution of the Mach number near the airfoil.



(a) Baseline mesh with 2, 776 triangular cell (purely second-order)

(b) adjoint-based *hp*-adapted after 6 adaptation cycles (second- to fourth-order)

Figure 5. Mesh resolution and discretization order for inviscid subsonic test case ($Ma_\infty = 0.5$ and $\alpha = 2°$)



(a) Initial solution with purely second-order discretization on a mesh with 2, 776 control volumes

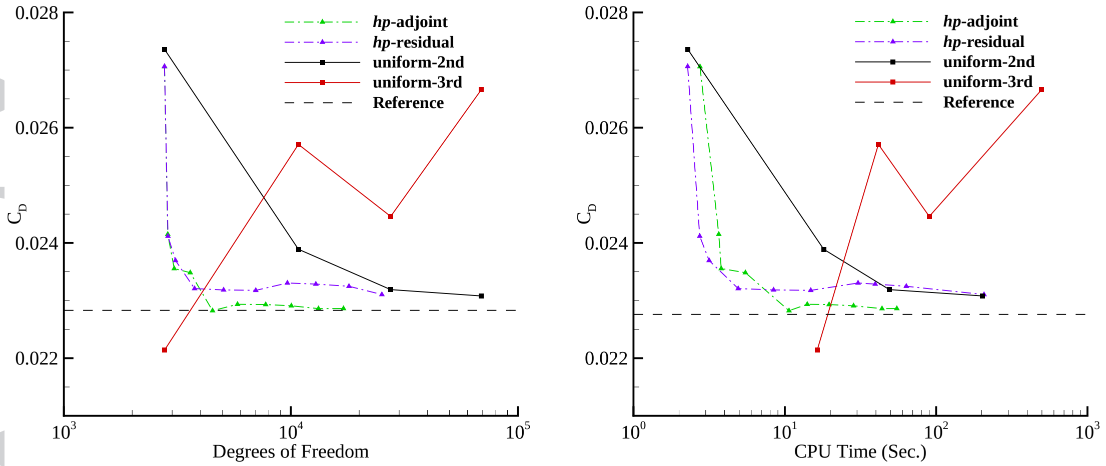
(b) adjoint-based *hp*-adapted solution after 6 levels of adaptation on a mesh with 5, 338 control volumes

Figure 6. Contours of Mach number for inviscid subsonic test case ($Ma_\infty = 0.5$ and $\alpha = 2°$)

6.2. *Inviscid transonic flow around NACA 0012*

As our second test case, we consider the transonic inviscid flow around the NACA 0012 with free stream Mach number of $Ma_\infty = 0.8$ and angle of attack of $\alpha = 1.25°$. For this test case, a strong shock and a weak shock appear on the upper and lower surfaces of the airfoil, respectively. Therefore, this is a good example to demonstrate the capability of the proposed *hp*-adaptation method for shock capturing. Similar to the inviscid subsonic case, we use the L_2-norm of the truncation error for residual-based adaptation and pressure drag coefficient as the functional of interest for the adjoint-based method. The other adaptation parameters are chosen as $f^{adapt} = 15\%$ and $\mathcal{K}_P = 300$. The smoothness indicator constant is set to a larger value to capture the weak shock on the lower surface although this leads to excessive refinement near the strong shock. As described, the *hp*-adaptation framework can be an alternative to slope limiting procedure for compressible problems with shocks and so can be used to enhance the robustness of an unstructured finite volume solver. This suggests we allow for first-order approximations near the shock as traditional slope limiters reduce the order of accuracy to one close to singularities. Instead, we do not use any slope limiter for problems with shocks. Although it is possible to start from a first-order solution everywhere and increasing the order in smooth regions of the flow field, this approach leads to the over-refinement of the initial mesh due to the the non-smooth behavior of the first-order solution. This is also repeated in the next cycles since the order is not increased in all of the *h*-refined cells (with order = 1). Instead, we start from a purely second-order solution on the same baseline mesh as the subsonic test case but we reduce the order of discretization by one (provided that the order is larger than 1) whenever a cell is flagged for *h*-refinement due to a non-smooth primal solution. In this way, we can attain the first-order approximation near the singularities after the first cycle. The order can be increased for any cell initially chosen for *h*-refinement (and obtained first-order accuracy) later in next adaptation cycles if that cell is yet chosen for refinement due to large error indicator but the smoothness indicator does not recognize the primal solution as non-smooth. Note that this is used as an extra option only for problems with shocks. For other problems with smooth solution almost everywhere, this strategy is not needed. Even if we use this option, the performance will not be significantly different as we rarely perform *h*-refinement in the initial phase of adaptation for those problems.

For uniform refinement studies, we only employ the second- and third-order discretizations with the slope limiter of Venkatakrishnan [41] and avoid the fourth-order discretization since it suffers from stability issues for such a problem with a strong shock. Figure 7 compares the efficiency of different refinement scenarios for this case. A reference drag coefficient is obtained by a second-order solution of this problem on an unstructured mesh with $308,470$ cells generated by uniform refinement. It is evident that the third-order discretization is not converging to a fixed value as the slope limiter degrades the accuracy of the solution. Furthermore, the tunable parameter in the slope limiter is adjusted for each mesh to get the solution to converge and this exacerbates the asymptotic convergence to a single value. Therefore, increasing the order of accuracy for problems with locally non-smooth solutions does not lead to more accurate answers as expected. On the other hand, the adjoint-based *hp*-adaptive method reaches the grid converged value after a small number of adaptation cycles so that the change in the drag coefficient becomes less than 0.2% in the last three consecutive cycles. Note that such a good estimate of the drag coefficient is found on an *hp*-adapted mesh with about $20,000$ control volumes. In term of the CPU time, the adjoint-based adaptation obviously outperforms uniform refinement considering the fact that the drag predicted by the second-order scheme on the finest mesh is not still sufficiently close to the reference value and takes about 5 times longer. The residual-based adaptation is also approaching the reference value but with a larger number of refinement cycles, degrees of freedom and CPU time compared the adjoint-based adaptation.



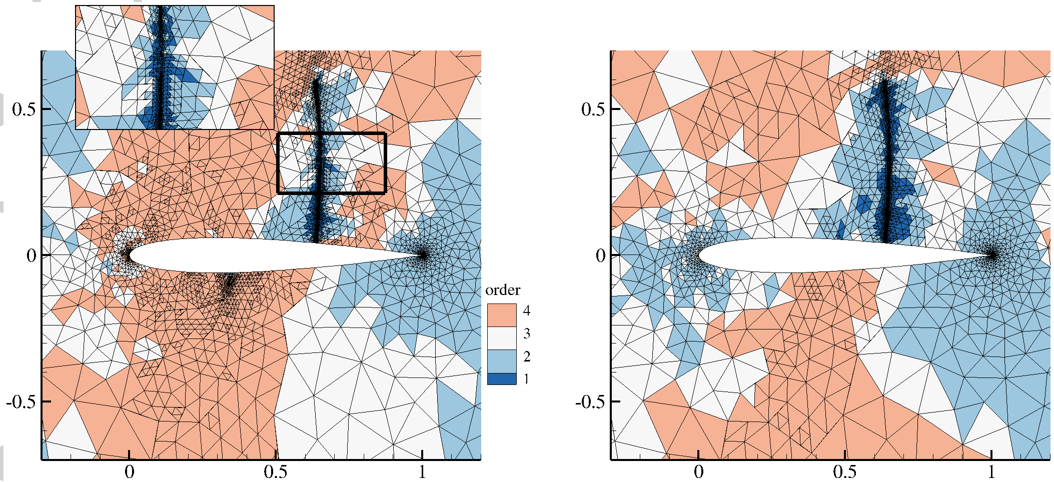
(a) Drag coefficient evolution with respect to degrees of freedom

(b) Drag coefficient evolution with respect to CPU time

Figure 7. Convergence of drag coefficient for inviscid transonic test case ($Ma_\infty = 0.8$ and $\alpha = 1.25°$)

The difference between the drag coefficients predicted by the two adaptation error indicators can be explained by the refinement pattern obtained from each one. Figure 8 compares the mesh resolution and local order of accuracy in the last cycle of the adjoint-based and residual-based adaptations. The residual-based indicator mainly focuses on the strong shock of the upper surface with minimal refinement of the leading and trailing edges. This behavior was previously reported by Woopen *et al.* [42] in residual-based mesh adaptation for their hybridized discontinuous Galerkin (HDG) method. In their study, the drag coefficients obtained on the residual-based adapted meshes differ by a constant amount from the values of the adjoint-based adapted meshes after a few refinement cycles. Conversely, the adjoint-based error indicator exhibits a more accurate refinement pattern as the the area close to the weak shock on the lower surface is refined properly in addition to the sufficient refinement of the leading and trailing edges. This implies that weighting the local residuals with the discrete adjoint solution provides a more accurate error indicator that captures all important features of the flow field during the refinement process. It should be noted that the first-order approximation is maintained close to the upper surface strong shock as magnified in Figure 8a which is ideal for regions with singular solution. Such a strategy is advantageous as it results in the efficiency of the solver in obtaining a grid converged output value with small number of degrees of freedom and also its robustness by employing a low-order solution in the singular regions of the flow field.

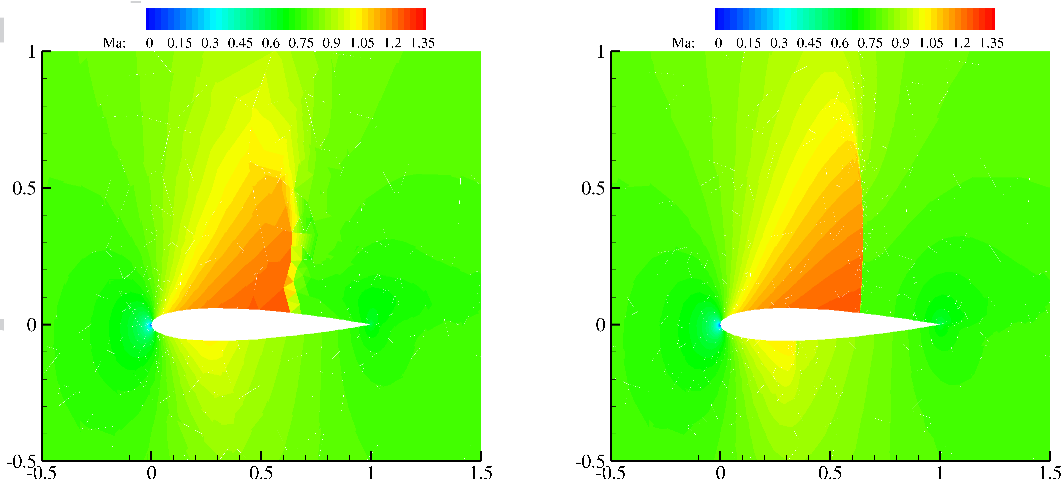
Figure 9 shows the contours of Mach number close to the airfoil for the initial and final adjoint-based adapted meshes of this test case. As expected, the two shocks sharpen as we proceed and increase the mesh resolution in their vicinity. Also, the use of first-order approximation in these regions prevents any overshoots in the final solution. In the smooth regions of the flow, the solution becomes smoother as we provide extra resolution by *p*-enrichment.



(a) Adjoint-based adaptation

(b) Residual-based adaptation

Figure 8. Final *hp*-adapted mesh and order for inviscid transonic test case ($Ma_\infty = 0.8$ and $\alpha = 1.25°$)

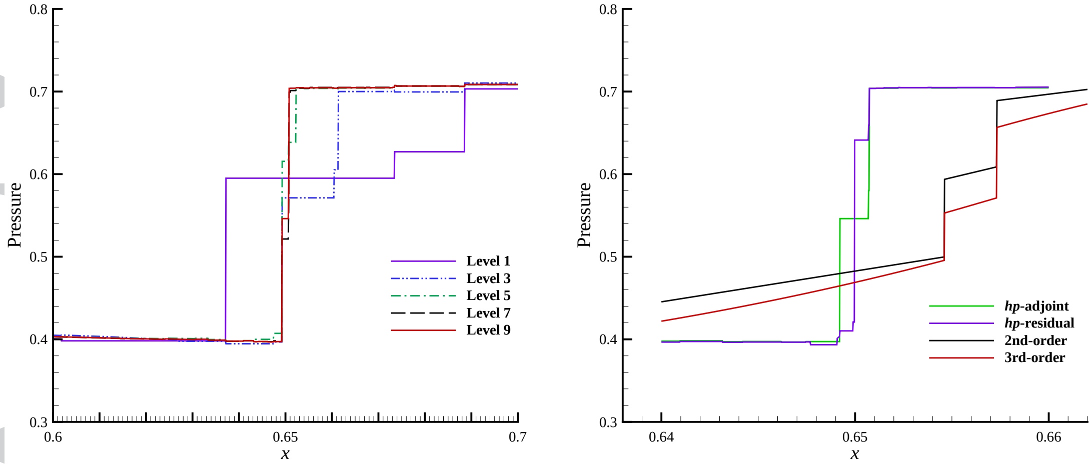


(a) adjoint-based *hp*-adapted after 1 adaptation cycle (first adapted mesh) on a mesh with 2,869 control volumes

(b) adjoint-based *hp*-adapted after 9 adaptation cycles (final adapted mesh) on a mesh with 17,056 control volumes

Figure 9. Contours of Mach number for inviscid transonic test case ($Ma_\infty = 0.8$ and $\alpha = 1.25°$)

Figure 10a compares the pressure profiles near the strong shock at $y = 0.3$ between different levels of adjoint-based *hp*-adaptation. The solution is sampled at equally spaced points spanning several control volumes. On the initial mesh, the shock spans a wide distance with several jumps in the magnitude of pressure. However, it gradually becomes thinner and sharper such that the shock computed on the last level is around 50 times thinner than the first one. This plot reveals the advantage of *hp*-adaptation in shock capturing in flows with discontinuities. Also, Figure 10b compares the pressure profile at $y = 0.3$ between the *hp*-adaptation methods after the final cycle of adaptation and second- and third-order solutions on the third uniformly refined mesh which has about the same number of control volumes.



(a) Adjoint-based levels

(b) Different methods

Figure 10. Comparison of pressure profiles near the upper surface strong shock at $y = 0.3$

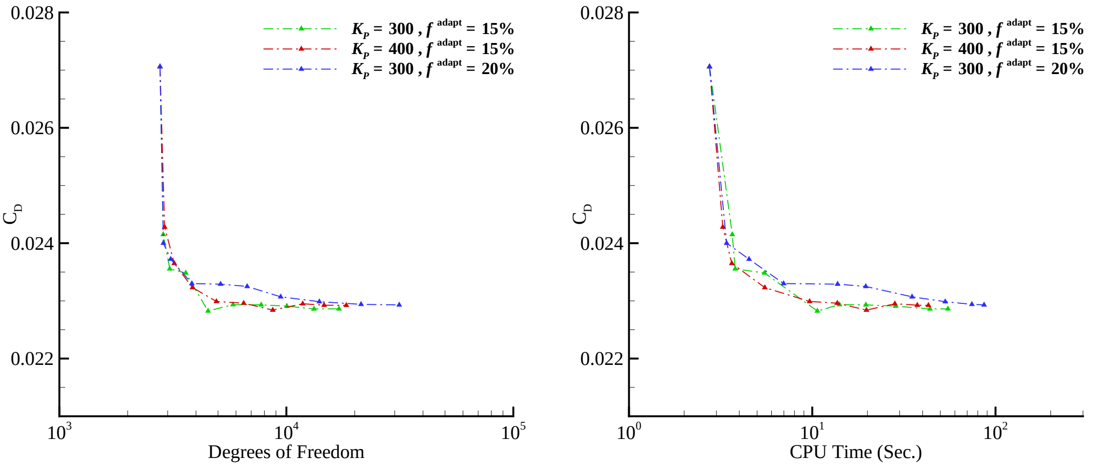
Note that the residual-based adaptation refines the region near the strong shock more than the adjoint-based as expected. This can be understood by the larger number of piecewise constant pressure values (first-order solutions) in the vicinity of the shock. It is also evident the second- and third-order methods do not estimate the location and strength of the shock quite accurately.

Finally, to investigate the effect of user-defined parameters on solver robustness, the convergence of drag coefficient is studied for three combinations (Figure 11). The difference in the final drag coefficient is less than 0.5 count of drag between different parameters. Also, increasing K_P and f^{adapt} results in a larger number of degrees of freedom as more control volumes are flagged at each level. It is seen that increasing the adaptation fraction by 5% increase the CPU time noticeably whereas increasing K_P slightly decreases the computation time due to having more control volumes with first-order discretization near the shock.

6.3. Turbulent subsonic flow over flat plate

As our first viscous problem, we compute the turbulent subsonic flow over a flat plate as described by the NASA Turbulence Modeling Resource (TMR) website [43]. Figure 12 shows the problem geometry and boundary conditions for this problem with $Re = 5 \times 10^6$ and $Ma_\infty = 0.2$. The range of the computational domain in the x-direction is $[-0.33, 2]$ with the leading edge of the flat plate at $x = 0$. The size of the domain in the y-direction is 1. At $y = 0$, the symmetry boundary condition is applied for $-0.33 \leq x \leq 0$ weakly by forcing the normal derivatives to be zero in the flux vectors and the no slip adiabatic boundary condition is imposed for $0 \leq x \leq 2$. Note that this causes a singularity in the solution at the leading edge. The free-stream value of the turbulent working variable is $\tilde{\nu}/\nu_\infty = 3.0$.

Similarly, the L_2-norm of the truncation error is used for residual-based adaptation. For adjoint-based adaptation, we consider the drag coefficient on the flat plate. The drag



(a) (b)

Figure 11. Effect of user-defined parameters in convergence of drag coefficient for inviscid transonic test case

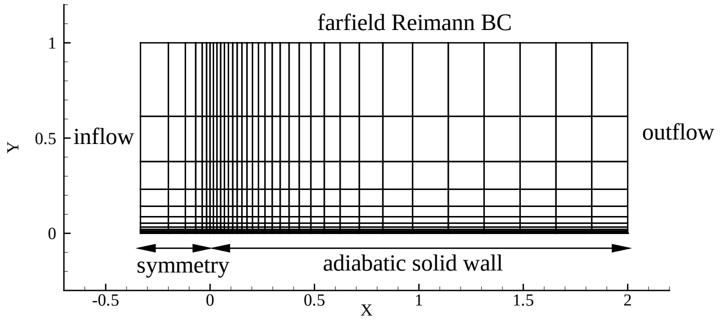


Figure 12. Turbulent flat plate test case mesh and geometry

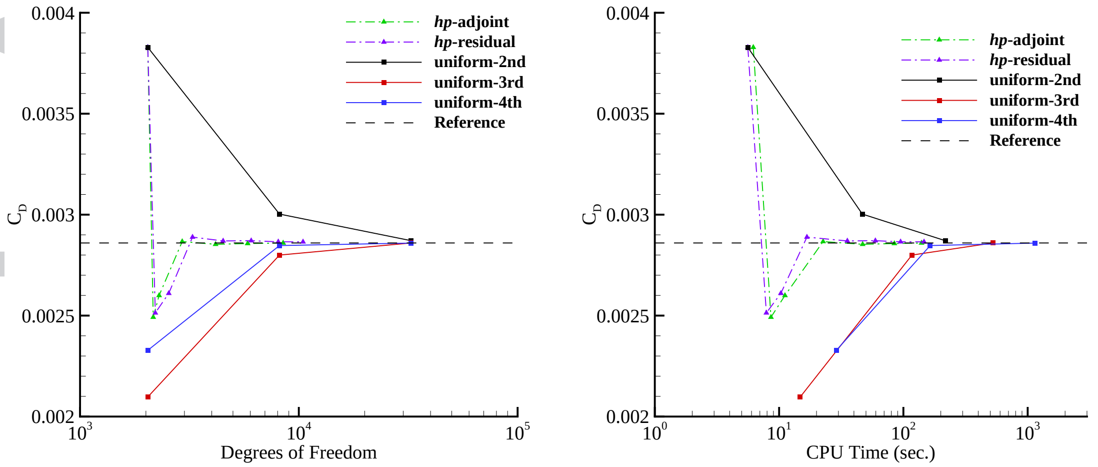
functional is defined as:

$$\mathcal{J}\left(U, \nabla U\right) = \int_{\partial\Omega} \psi \cdot \left(P\hat{n} - \tau\hat{n}\right)\, ds \tag{29}$$

in which $\hat{n}$ is the outward unit normal vector, τ is the viscous stress tensor and $\psi = \frac{1}{C_\infty}\left(\cos\,\alpha,\, \sin\,\alpha\right)^T$along the wall boundaries and 0 everywhere else. Note that for this case, the drag coefficient is reduced to viscous drag. For simplicity, we use constant dynamic viscosity.

For uniform refinement, we consider second- to fourth-order discretization on three nested quadrilateral meshes with $2,040$, $8,160$ and $32,640$ control volumes. The *hp*-adaptation starts with the second-order solution on the coarsest mesh. The adaptation parameters are set to $\mathcal{K}_P = 400$, $\mathcal{K}_{\tilde{\nu}} = 100$ and $f^{adapt} = 15\%$.

Figure 13 compares the efficiency of the *hp*-adaptation methods with uniform refinement of different discretization orders. The reference drag coefficient is the value obtained by fourth-order discretization on a very fine mesh with $130,560$ control volumes. The adaptive refinements are considerably more successful than uniform refinements, particularly the adjoint-based method which requires almost five times fewer control volumes to obtain the grid converged output. This is expected since adaptive refinement places sufficient resolution in the boundary layer without unnecessary refinement of other regions. Such a resolution can only be obtained on super fine meshes in the case of uniform refinement. More importantly, the CPU time on the final adjoint-based adapted mesh is 2, 4 and 9 times smaller than that of second-, third- and fourth-order discretizations on the finest mesh, respectively. This is also expected since the extra derivatives required for higher-order approximations are only computed in a small fraction of control volumes and also the Jacobian matrix has fewer entries. The residual-based adaptation is also effective for this case although it needs slightly more control volumes and CPU time compared to the adjoint-based method to reach the reference value.



(a) Drag coefficient evolution with respect to degrees of freedom

(b) Drag coefficient evolution with respect to CPU time

Figure 13. Convergence of drag coefficient for turbulent flat plate test case

Figure 14 shows the final adjoint-based adapted mesh and also order of accuracy for the control volumes close to the flat plate. Note that the adaptive refinement focuses on the region near the leading edge and also inside the boundary layer. It is also worth mentioning that the order of accuracy at the edge of the boundary layer where the derivative of the turbulence working variable is discontinuous does not increase even though the mesh is refined.

Figure 15 compares the turbulence working variable between the initial and final adjoint-based *hp*-adapted solutions. The viscous dominated part of the flow near the wall is refined properly without unnecessary refinement out of the viscous dominated region. The smooth contours of the turbulence working variable and the significantly higher peak value in the boundary layer show the improvement in the solution of the *hp*-adapted mesh compared to the initial mesh.

6.4. Laminar subsonic flow around NACA 0012

Now we turn our attention to viscous laminar subsonic flow around the NACA 0012 airfoil with a free stream Mach number of $Ma_\infty = 0.5$, Reynolds number of $Re = 5000$ with

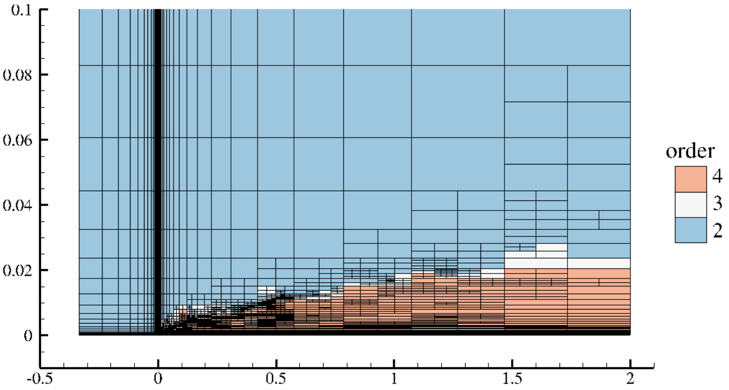
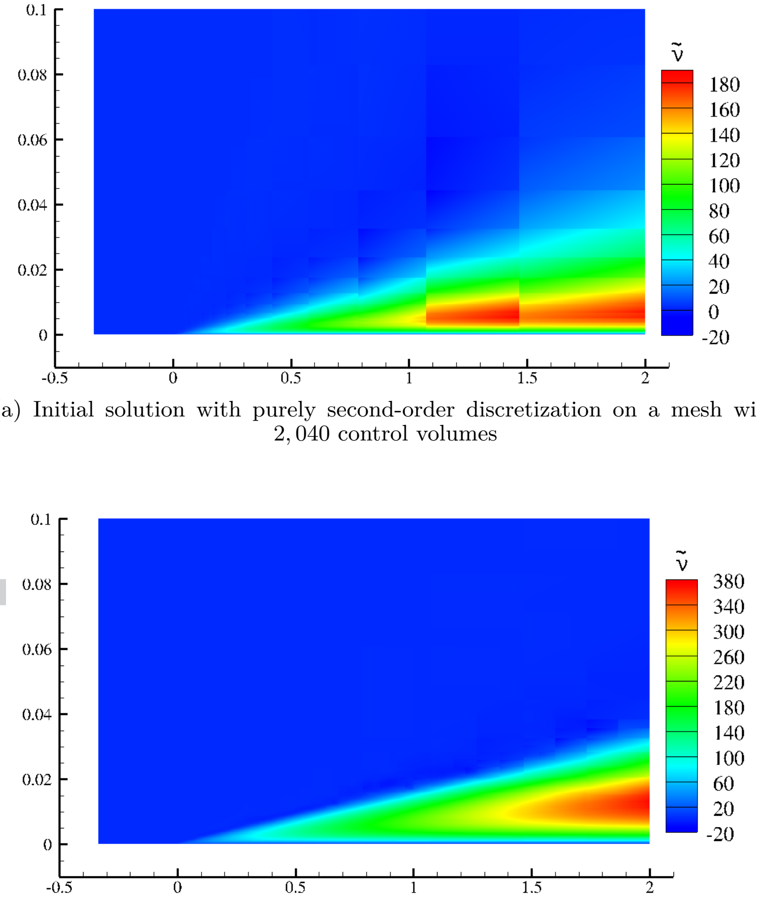


Figure 14. Mesh resolution and discretization order for the final adjoint-based *hp*-adapted turbulent flat plate test case ($Ma_\infty = 0.2,\ Re = 5 \times 10^6$)



(a) Initial solution with purely second-order discretization on a mesh with 2,040 control volumes

(b) adjoint-based *hp*-adapted final solution on a mesh with 8,478 control volumes

Figure 15. Contours of turbulence working variable for turbulent flat plate test case

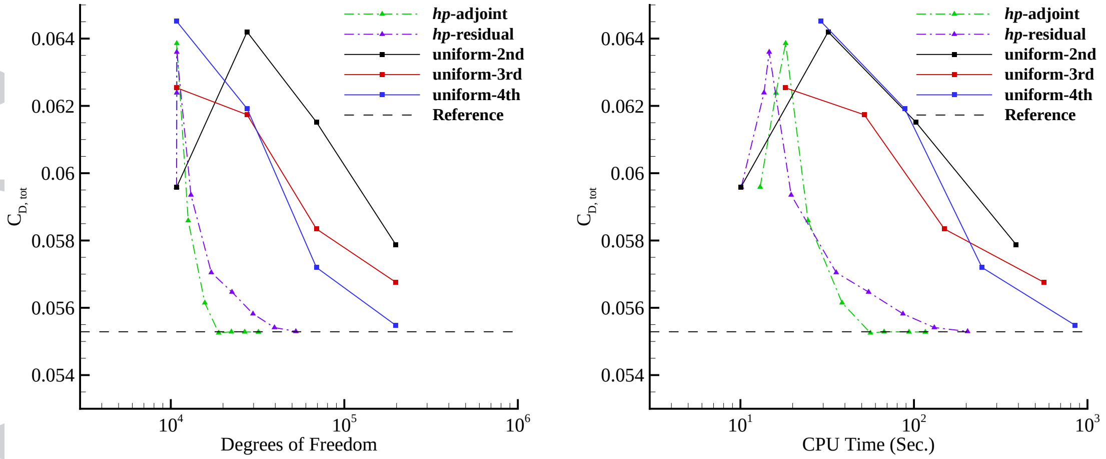
constant dynamic viscosity and angle of attack of $\alpha = 1°$. In this flow, a thin boundary layer appears on the surface of the airfoil and is followed by a wake downstream of the trailing edge. Similar to the turbulent flat plate subsonic case, we use the L_2-norm of the truncation error for residual-based adaptation and total drag coefficient as the functional of interest for the adjoint-based method. We start the adaptations with a second-order solution on an unstructured mesh with $10,797$ triangular cells. The adaptation parameters for this problem with smooth solution are set to $f^{adapt} = 10\%$ and $\mathcal{K}_P = 200$.

Figure 16 compares the efficiency of the *hp*-adaptation methods with uniform refinement of second-, third- and fourth-order discretizations. For this purpose, the convergence of the total drag coefficient versus the number of degrees of freedom and CPU time is plotted. For this problem, a reference drag coefficient is found by a fourth-order discretization on a mesh with $308,470$ cells. The advantage of *hp*-adaptation over uniform refinement is very clear for this test case with thin shear layers: the *hp*-adapted meshes reach grid convergence with about as many degrees of freedom as the first uniform refinement. At this resolution, the second-order discretization on the uniformly refined mesh has not yet begun to improve the results. The higher than second-order schemes (particularly fourth-order) converge faster as expected for this problem with smooth solution; however, the CPU time required to get close to the reference drag coefficient is more than six times larger compared with *hp*-adaptation. Similar to the previous test cases, the adjoint-based adaptation is more successful than the residual-based. Even though the residual-based converges to the same output value, it adds a number of unnecessary degrees of freedom in each cycle that do not improve functional accuracy. Note that the increase in the number of degrees of freedom for this problem typically originates from those fourth-order control volumes which need more resolution and thus are refined via *h*-refinement. On the other hand, the adjoint-based adaptation indicator optimally increases the resolution to be able to deliver a very accurate functional for which the adaptation is performed. The output value becomes almost constant in the last three cycles of the adjoint-based adaptation which highlights its efficiency. Note that the time required for solving the discrete adjoint problem is about 20 percent of the computation time. However, the total CPU time on grid converged meshes is smaller for adjoint-based adaptation since the residual-based method produces additional control volumes at each level and thus it takes longer to solve the primal problem. Table I shows the time required for solving the primal problem plus the higher-order residual evaluation which is needed for both adaptation methods and also the time required for solving the discrete adjoint problem only in adjoint-based adaptation.

	Residual-based		Adjoint-based		
Level	DoF	CPU time (sec.)	DoF	CPU time (sec.)	
		Primal + T.E.		Primal + T.E.	Adjoint
1	10797	10.12	10797	10.12	2.44
3	10824	14.65	10824	15.75	2.74
5	17109	35.64	15699	32.55	5.98
7	29754	86.43	22341	54.81	12.47
9	52569	203.72	31980	95.36	20.89

Table I. Breakdown of CPU time for *hp*-adaptation of laminar subsonic flow

Figure 17 depicts the mesh resolution and local order for the control volumes of the initial mesh and also adjoint-based *hp*-adapted mesh after 7 cycles. As required, the order

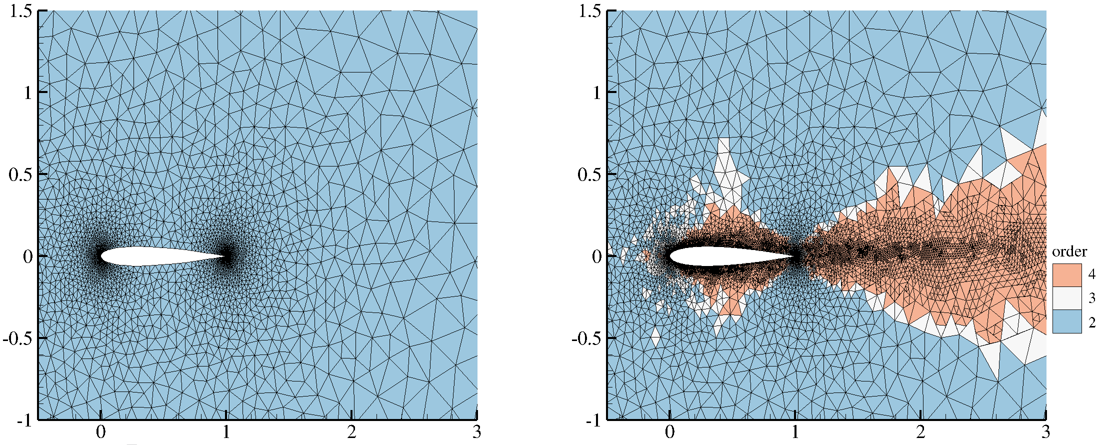


(a) Drag coefficient evolution with respect to degrees of freedom

(b) Drag coefficient evolution with respect to CPU time

Figure 16. Convergence of drag coefficient for laminar subsonic test case ($Ma_\infty = 0.5$, $Re = 5000$ and $\alpha = 1°$)

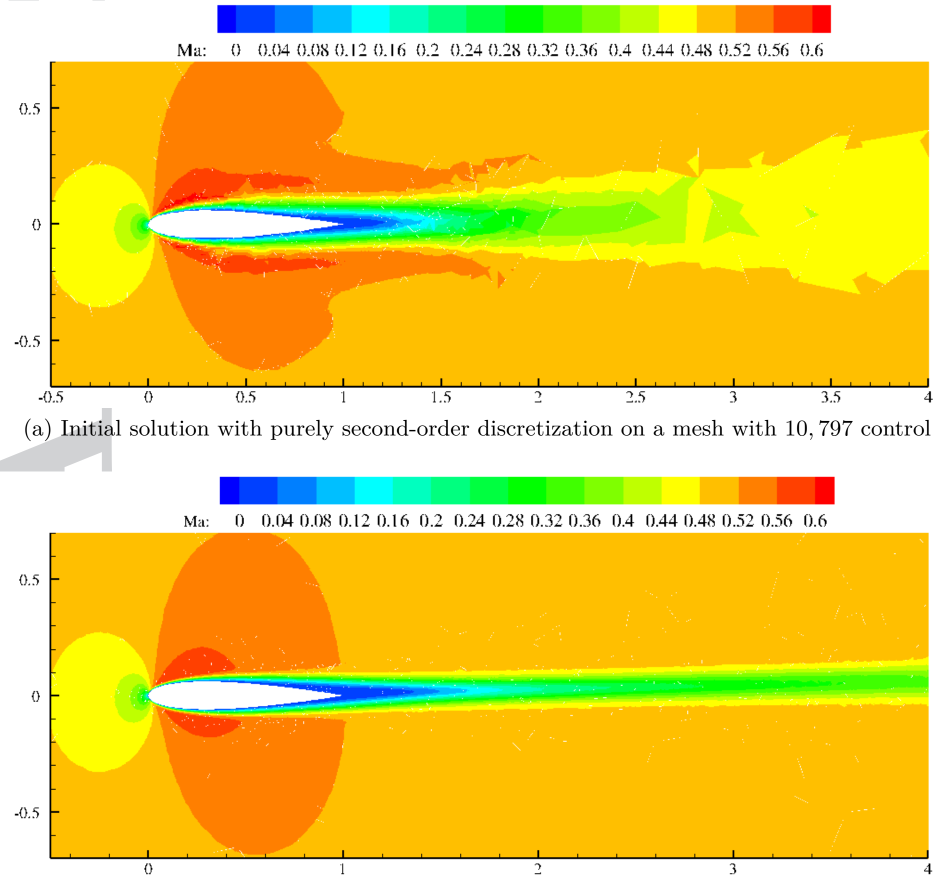
of accuracy increases in the boundary layer and wake regions via p-enrichment. Since some of the control volumes in these regions still contribute to the error the most even after the p-enrichment, we apply h-refinement so that we can reduce the error in the next cycles effectively and converge to the reference output value as fast as possible. To provide such a resolution with uniform refinement, we require considerably more cells in the mesh, most of which are far from the airfoil and do not contribute to the total error. Note that the mesh size and order do not change sufficiently far from the viscous dominated regions in the *hp*-adapted mesh.



(a) Baseline mesh with 10,797 triangular cells

(b) adjoint-based *hp*-adapted after 7 adaptation cycles (second- to fourth-order)

Figure 17. Mesh resolution and discretization order for laminar subsonic test case ($Ma_\infty = 0.5$, $Re = 5000$ and $\alpha = 1°$)



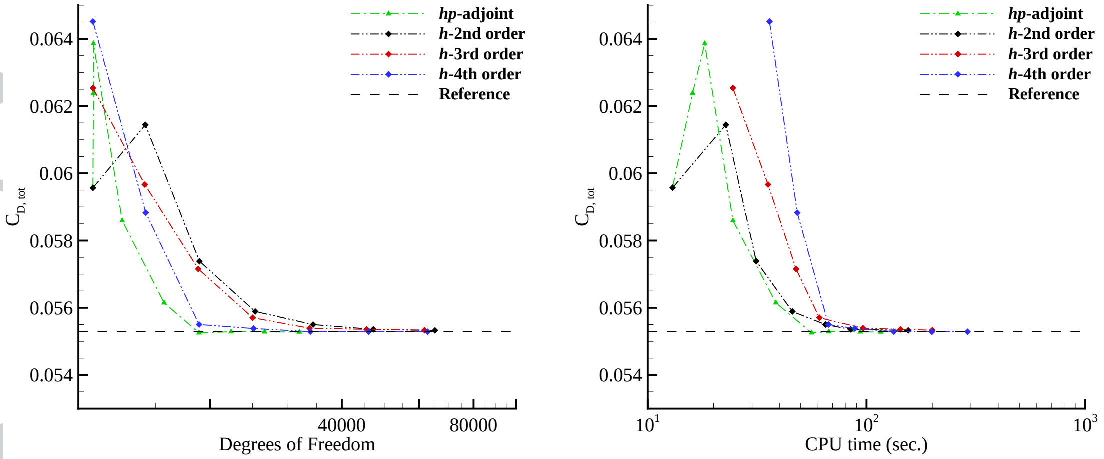
(a) Initial solution with purely second-order discretization on a mesh with $10,797$ control volumes

(b) adjoint-based *hp*-adapted final solution on a mesh with $31,980$ control volumes

Figure 18. Contours of Mach number for laminar subsonic test case ($Ma_\infty = 0.5$, $Re = 5000$ and $\alpha = 1^\circ$)

Moreover, the contours of Mach number on the initial and final *h*p-adapted meshes are shown in Figure 18. The longer wake region and smoother velocity distribution near the airfoil on the final mesh are clearly visible.

For this final test case with smooth solution, we compare the efficiency of *hp*-adaptation with pure *h*-adaptation of different orders. Considering that the adjoint-based adaptation is more efficient than residual-based method, we use adjoint-based error estimation for all cases with $f^{adapt} = 10\%$. Figure 19 compares the efficiency of different adjoint-based adaptation methods in terms of number of degrees of freedom and also CPU time. As seen, *hp*-adaptation outperforms pure *h*-refinement of different orders as it requires smaller number of cells to produce grid converged outputs. Even though *h*-refinement with the fourth-order scheme is comparable with *hp*-adaptation in terms of number of degrees of freedom, its computational cost is larger since higher-order polynomials should be reconstructed in all control volumes and the global Jacobian matrix is less sparse.



(a) Drag coefficient evolution with respect to degrees of freedom

(b) Drag coefficient evolution with respect to CPU time

Figure 19. Comparison of *h*- and *hp*-adaptive refinement for laminar subsonic test case

7. CONCLUSIONS

This paper presented an *hp*-adaptation method for an unstructured finite volume compressible flow solver. The flow solver approximates the primitive variables of the flow field using *K*-exact least-squares reconstruction. An implicit pseudo-time stepping scheme combined with a line search algorithm is employed to find the steady-state solution of the flow problems. For *hp*-adaptation, both residual-based and adjoint-based error indicators were examined. The former was obtained by a higher-order residual operator which estimates the truncation error while the adjoint-based indicator used the solution of the discrete adjoint problem to weight the truncation error. For adaptation, the control volumes which contribute to the total error the most are flagged for *h*-refinement or *p*-enrichment based on the smoothness of the solution. Mesh refinement is performed by local subdivision of cells and order enrichment is done by the reconstruction of a higher-order polynomial.

Numerical results were presented for inviscid and viscous flow problems. In each case, uniform refinement of second- and higher-order methods was compared against residual-based and adjoint-based adaptations in terms of the number of degrees of freedom and CPU time required to reach a reference value. In all of the cases, it was shown that the adjoint-based *hp*-adaptive method outperforms the other refinement procedures and is more successful in capturing the flow features that affect the accuracy of an output functional. Moreover, it was shown that the *hp*-adaptive method is capable of shock capturing and automatic limiting of flows with discontinuities (such as the transonic inviscid case) by mesh refinement and lowering the order of accuracy to one in those regions, simultaneously. For such flows where increasing the order of accuracy in non-smooth regions of the solution does not lead to more accurate solutions or sometimes causes solver failure, such a strategy is superior. Moreover, the advantage of the adaptive methods were sufficiently clear in the case of viscous laminar and turbulent cases where thin shear layers are present in the solution.

For these cases, the adaptive methods place enough resolution in the viscous dominated regions and thus provide a good estimate of functional values as opposed to the uniform refinements where many control volumes are needed to resolve those regions accurately. Also, for the viscous laminar test case, it was shown that the adjoint-based *hp*-adaptation outperforms adjoint-based *h*-refinement of different orders.

This work extended the idea of *hp*-adaptivity to unstructured finite volume methods. Future work will concentrate on the extension of *hp*-adaptations to our higher-order RANS flow solver used for the solution of turbulent flows on highly anisotropic meshes over aerodynamic configurations [33]. Moreover, we plan to extend this work to three-dimensional problems where the problems size is considerably larger and uniform refinement is not affordable.

ACKNOWLEDGEMENT

The first author would like to acknowledge the financial support of the University of British Columbia and the Canadian Natural Sciences and Engineering Research Council under Discovery Grant 194467.

REFERENCES

1. Stefano Giani and Paul Houston. High–order hp–adaptive discontinuous Galerkin finite element methods for compressible fluid flows. In *ADIGMA-A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, pages 399–411. Springer, 2010.
2. Marco Ceze and Krzysztof J. Fidkowski. Anisotropic hp-adaptation framework for functional prediction. *AIAA Journal*, 51(2):492–509, 2013.
3. Nicholas K. Burgess and Dimitri J. Mavriplis. An hp-adaptive discontinuous Galerkin solver for aerodynamic flows on mixed-element meshes. In *Proceedings of the Forty-Ninth AIAA Aerospace Sciences Meeting*, 2011.
4. Li Wang and Dimitri J Mavriplis. Adjoint-based hp adaptive discontinuous Galerkin methods for the 2d compressible Euler equations. *Journal of Computational Physics*, 228(20):7643–7661, 2009.
5. Nicholas K. Burgess and Dimitri J. Mavriplis. hp-adaptive discontinuous Galerkin methods for the Navier-Stokes equations. *AIAA Journal*, 50(12):2682–2694, 2012.
6. Michael Woopen, Arvind Balan, Georg May, and Jochen Schütz. A comparison of hybridized and standard DG methods for target-based hp-adaptive simulation of compressible flow. *Computers & Fluids*, 98:3–16, 2014.
7. W. Kyle Anderson, Behzad R. Ahrabi, and James C. Newman. Finite element solutions for turbulent flow over the NACA 0012 airfoil. *AIA Journal*, 54(9):2688–2704, 2016.
8. Behzad R. Ahrabi and W. Kyle Anderson. An adjoint-based hp-adaptive Petrov–Galerkin method for turbulent flows. In *Proceedings of Twenty-Second AIAA Computational Fluid Dynamics Conference*, 2015. AIAA Paper 2015-2603.
9. Carl F. Ollivier-Gooch and Michael Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, sep 2002.
10. Christopher Michalak and Carl Ollivier-Gooch. Globalized matrix-explicit Newton-GMRES for the high-order accurate solution of the Euler equations. *Computers and Fluids*, 39:1156–1167, 2010.
11. Amir Nejat and Carl Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227(4):2592–2609, 2008.

12. Christopher Michalak and Carl Ollivier-Gooch. Accuracy preserving limiter for the high-order accurate solution of the Euler equations. *Journal of Computational Physics*, 228(23):8693–8711, 2009.
13. Michael J. Aftosmis, Datta Gaitonde, and T. Sean Tavares. On the accuracy, stability, and monotonicity of various reconstruction algorithms for unstructured meshes. AIAA paper 94-0415, January 1994.
14. M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82(1):64–84, 1989.
15. Mahkame Sharbatdar and Carl Ollivier-Gooch. Anisotropic mesh adaptation: Recovering quasi-structured meshes. In *Proceedings of the Fifty-First AIAA Aerospace Sciences Meeting*, 2013.
16. Doug Pagnutti and Carl Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. *Computers and Structures*, 87:670–679, 2009.
17. X. Zhang, M.-G. Vallet, J. Dompierre, P. Labbe, D. Pelletier, J.-Y. Trepanier, R. Camarero, J. Lassaline, L. Manzano, and D. Zingg. Mesh adaptation using different error indicators for the Euler equations. In *Proceedings of the Fifteenth AIAA Computational Fluid Dynamics Conference*, 2001. AIAA Paper 2001-2549.
18. D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22–46, 2003.
19. M. Ainsworth and J. T. Oden. A unified approach to a posteriori error estimation using element residual methods. *Numerische Mathematik*, 65(1):23–50, 1993.
20. Haiyang Gao and Z.J. Wang. A residual-based procedure for hp-adaptation on 2D hybrid meshes. In *Proceedings of the Forty-ninth AIAA Aerospace Sciences Meeting*, 2011. AIAA Paper 2011-492.
21. N. Ganesh, N. V. Shende, and N. Balakrishnan. A residual estimator based adaptation strategy for compressible flows. In *Computational Fluid Dynamics 2006*, pages 383–388. Springer, 2009.
22. Krzysztof J. Fidkowski and David L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.
23. Niles Pierce and Michael Giles. Adjoint and defect error bounding and correction for functional estimates. *Journal of Computational Physics*, 200(2):769–794, 2004.
24. Michael Andrew Park. Adjoint-based, three-dimensional error prediction and grid adaptation. *AIAA Journal*, 42(9):1854–1862, 2004.
25. D. A. Venditti and D. L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *Journal of Computational Physics*, 164(1):204–227, October 2000.
26. D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *Journal of Computational Physics*, 175(1):40–69, February 2002.
27. Ralf Hartmann and Paul Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.
28. Ralf Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 51(9–10):1131–1156, 2006.
29. R. Hartmann., J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k–ω turbulence model equations. *Journal of Computational Physics*, 230(11):4268–4284, 2011.
30. T. Oliver and D. Darmofal. An unsteady adaptation algorithm for discontinuous Galerkin discretizations of the RANS equations. In *Proceedings of the Eighteenth AIAA CFD Conference*, 2007. AIAA Paper 2007-3940.
31. Todd A. Oliver. *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*. PhD thesis, Massachusetts Institute of Technology, 2008.
32. S. R. Allmaras, F. T. Johnson, and P. Spalart. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *Proceedings of the Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.
33. Alireza Jalali and Carl Ollivier-Gooch. Higher-order unstructured finite volume RANS solution of turbulent compressible flows. *Computer & Fluids*, 143:32–47, 2017.
34. Timothy J. Barth and Paul O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, January 1990.
35. P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
36. Alireza Jalali, Mahkame Sharbatdar, and Carl Ollivier-Gooch. Accuracy analysis of unstructured finite volume discretization schemes for diffusive fluxes. *Computers and Fluids*, 101:220–232, 2014.

37. M. Ceze and K. Fidkowski. Constrained pseudo-transient continuation. *International Journal for Numerical Methods in Engineering*, 102:1683–1703, 2015.
38. Mahkame Sharbatdar, Alireza Jalali, and Carl Ollivier-Gooch. Smoothed truncation error in functional error estimation and correction using adjoint methods in an unstructured finite volume solver. *Computers & Fluids*, 140:406–421, 2016.
39. Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *Proceedings of the Forty-fourth AIAA Aerospace Sciences Meeting and Exhibit*, 2006. AIAA Paper 2006-112.
40. L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J.E. Flaherty. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48:323–338, 2004.
41. V. Venkatakrishnan. Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118:120–130, 1995.
42. M. Woopen, G. May, and J. Schütz. Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 76(11):811–834, 2014.
43. Christopher Rumsey. Turbulence Modeling Resource, NASA Langley Research Center, http://turbmodels.larc.nasa.gov/, 2014.