Hp-Adaptivity on Anisotropic Meshes for Hybridized Discontinuous Galerkin Scheme

Aravind Balan, Michael Woopen, and Georg May

Graduate School AICES, RWTH Aachen University, 52062 Aachen, Germany

We present an efficient adaptation methodology on anisotropic meshes for the recently developed hybridized discontinuous Galerkin scheme for (nonlinear) convection-diffusion problems, including compressible Euler and Navier-Stokes equations. The methodology extends the refinement strategy of Dolejsi [8] based on an interpolation error estimate to incorporate an adjoint-based error estimate. For each element, we set the area using the adjoint-based error estimate, and we seek the anisotropy, of the element, which gives the smallest interpolation error in the $L^q$-norm ($q \in [1, \infty]$). For hp-adaptation, the local polynomial degree is also chosen in such a way that the configuration - element shape and the polynomial degree, gives the smallest interpolation error in the $L^q$-norm. Numerical results are shown for a scalar convection-diffusion case with a strong boundary layer, as well as for inviscid subsonic, transonic and supersonic and viscous subsonic flow around the NACA0012 airfoil, to demonstrate the effectiveness of the adaptation methodology.

I. Introduction

High order methods have become popular over the last decade due to their potential in giving more accurate results with lower cost compared to low order methods. The most popular high order method used to solve convection dominated flows is arguably the discontinuous Galerkin method [3, 1, 5, 2, 4]. One disadvantage of the DG method is the large number of degrees of freedom resulting from relaxing the continuity constraint of the solution across element interfaces. Recently, hybridization has been found useful in reducing the number of globally coupled degrees of freedom and hence the size of the global system that has to be solved at each iteration step. Here, the coupled unknowns, also known as the hybrid variables, have support only on the element interfaces. Hybridized DG discretizations were presented by Nguyen et al. for linear [14] and nonlinear [15] convection-diffusion equations. Peraire et al. [16] have extended the method for the Euler and Navier-Stokes equations and have shown its viability in aerodynamic flow simulations. In a slightly different approach, Egger et al. [9] have presented the hybridized formulation for the linear convection-diffusion equation, where the convection term is discretized with a DG method and the diffusion term with a hybrid mixed method. Schütz and May [12] have extended this for the compressible Euler and Navier-Stokes equations and have got promising results.

In most engineering applications, one may not be interested in full flow details, but rather in some specific quantities. In external aerodynamics, these may be lift or drag coefficients of airplanes. With the aim of getting accurate values for such functional quantities in the most efficient way, target-based error control methods have been developed. One such method is based on the adjoint solution of the original governing equations. In this method, an additional linear system is solved which then gives an estimate on the spatial error distribution contributing to the error in the target functional. This estimate can be used as a criteria for local adaptation, either h-adaptation (mesh refinement) or for p-adaptation (polynomial space enrichment), so that the error in the target functional is expected to get reduced. The adjoint solution acts as a weight in the error estimator, capturing the domain of dependence of errors in the target. Thus adjoint-based adaptation methods usually outperform purely feature-based methods.

Schütz and May [18] and Woopen et al. [22] devised and implemented an adjoint-based mesh adaptation procedure based on a class of adjoint-consistent hybridized finite element methods. The method has also

*Graduate Student
†Junior Professor
been extended to solve 3D problems and also turbulent flows using RANS approach by Woopen et al. [20], [21].

There are certain flows, for example, aerodynamic flows, which exhibit highly anisotropic features like boundary layers and shocks. In order to resolve these solution features efficiently, the mesh elements should also possess anisotropy (stretching in some direction) to align with the flow features. Anisotropic mesh elements along with the adjoint error estimation would provide an efficient distribution of the degrees of freedom in order to get an accurate value for the target functional. The first adjoint-based anisotropic refinement algorithm was devised by Venditti and Darmofal [6], which was applied to a finite volume scheme with piecewise linear reconstruction. They combined an adjoint based error estimate and a Hessian based method for anisotropic adaptation. This was extended to high order schemes by Fidkowski [10]. The basic idea is to construct mesh whose area is determined by the adjoint error estimator and the anisotropy is determined by requiring that the interpolation errors are the same along each edge for each mesh element. In a slightly different approach of Dolejsi [8], the anisotropy is chosen in such a way that the interpolation error in the $L^q$-norm ($q \in [1, \infty]$) is minimized. In the present work, we extend the adaptation strategy of Dolejsi [8] to incorporate an adjoint-based error estimate. We set the area of the element using the adjoint error estimate, and we seek the anisotropy which gives the smallest interpolation error in the $L^q$-norm ($q \in [1, \infty]$) on each element. For hp-adaptation, the polynomial degree is also chosen in such a way that the configuration - the element shape and the polynomial approximation degree, gives the smallest interpolation error in the $L^q$-norm.

The rest of the paper is organized as follows. The governing equations are introduced in Sec. II. The details of the numerical scheme, which includes the discretization of the weak formulation, time relaxation and the hybridization is given in Sec. III. Adjoint-based error estimation is briefly described in Sec. IV. In the subsequent sections, Sec. V and Sec. VI, the refinement methodologies for pure h-refinement and hp-refinement on anisotropic meshes are described, respectively. In the last section, Sec. VII, we show results for different test cases such as a scalar linear convection-diffusion problem, Euler and Navier-Stokes equations.

II. Governing Equations

A. Two-Dimensional Euler Equations

The Euler equations are comprised of the inviscid compressible continuity, momentum and energy equations. They are given in conservative form as

$$\partial_t w + \nabla \cdot f_c(w) = 0$$

with the conserved variables

$$w = (\rho, \rho u, \rho v, E)^T$$

where $\rho$ is the density, $u$ and $v$ the velocities in horizontal and vertical direction, and $E$ the total energy. The convective flux is given by

$$f_{c,1} = (\rho u, p + \rho u^2, \rho u v, u(E + p))^T$$
$$f_{c,2} = (\rho v, \rho u v, p + \rho u^2, v(E + p))^T.$$  

Pressure is related to the conservative flow variables $w$ by the equation of state

$$p = (\gamma - 1) \left( E - \frac{1}{2} \rho (u^2 + v^2) \right)$$

where $\gamma = c_p/c_v$ is the ratio of specific heats, generally taken as 1.4 for air. The speed of sound is $c$ which for ideal fluid is given by $c = \sqrt{\gamma \frac{p}{\rho}}$.

B. Two-Dimensional Navier-Stokes Equations

The Navier-Stokes equations are the viscous complement of the Euler-Equations for a Newtonian fluid. In conservative form they are given by

$$\partial_t w + \nabla \cdot (f_c(w) - f_c(w, \nabla w)) = 0.$$
The convective part of the Navier-Stokes equations coincides with the Euler equations. Thus, we only state the viscous flux in the following.

\[
    f_{v,1} = (0, \tau_{11}, \tau_{21}, \tau_{11}u + \tau_{12}v + kT) \quad (7)
\]

\[
    f_{v,2} = 0, \tau_{12}, \tau_{22}, \tau_{21}u + \tau_{22}v + kT \quad (8)
\]

The temperature is defined via the ideal gas law

\[
    T = \frac{\mu \gamma}{k \cdot \text{Pr}} \left( \frac{E}{\rho} - \frac{1}{2} (u^2 + v^2) \right) = \frac{1}{(\gamma - 1) \rho} \quad (9)
\]

where \( \text{Pr} = \frac{\mu c}{k} \) is the Prandtl number, which for air at moderate conditions is constant with a value of \( \text{Pr} = 0.72 \). \( k \) denotes the thermal conductivity coefficient. For a Newtonian fluid, the stress tensor is defined as follows

\[
    \tau = \mu \left( \nabla \tilde{w} + (\nabla \tilde{w})^T - \frac{2}{3} (\nabla \cdot \tilde{w}) \text{Id} \right) \quad (10)
\]

The variation of the molecular viscosity \( \mu \) as a function of temperature is determined by Sutherland’s law [19] as

\[
    \mu = \frac{C_1 T^{3/2}}{T + C_2} \quad (11)
\]

### III. Discretization

#### A. Notation

We tessellate the domain \( \Omega \) into a collection of non-overlapping elements, denoted by \( T_h \), such that \( \bigcup_{T \in T_h} T = \Omega \). For the element edges we regard two different kinds of sets, \( \partial T_h \) and \( \Gamma_h \), which are element-oriented and edge-oriented, respectively. The first is the collection of all element boundaries, which means that every edge appears twice. The latter however includes every edge just once. The reason for this distinction will become clear later. Please note that neither of these sets shall include edges lying on the domain boundary; these are denoted by \( \Gamma^b_h \).

We will distinguish between element-oriented inner products and edge-oriented inner products

\[
    (v, w)_{T_h} = \sum_{T \in T_h} \int_T vw \, dx, \quad \langle v, w \rangle_{\partial T_h} = \sum_{T \in T_h} \int_{\partial T} vw \, d\sigma, \quad \langle v, w \rangle_{\Gamma_h} = \sum_{e \in \Gamma_h} \int_e vw \, d\sigma.
\]

Let \( T^+ \) and \( T^- \) be two elements which share an interior edge. For any point on this edge, we introduce the so-called jump-operator which is defined as

\[
    [\varphi] = \varphi^- n^- + \varphi^+ n^+ \quad \text{for scalar } \varphi \quad (12)
\]

\[
    [\tau] = \tau^- \cdot n^- + \tau^+ \cdot n^+ \quad \text{for vector-valued } \tau \quad (13)
\]

for scalar \( \varphi \) and as

for vector-valued \( \tau \). Here \( n^+ \) and \( n^- \) denote the outward pointing normals on \( T^+ \) and \( T^- \), respectively, while \( \phi^\pm \) and \( \tau^\pm \) are the corresponding trace values.

#### B. Weak Formulation

A general convection-diffusion equation can be rewritten into a first-order system by introducing an additional unknown representing the gradient of the solution

\[
    q = \nabla w \quad (14)
\]

\[
    \nabla \cdot (f_e (w) - f_v (w, q)) = s (w, q) \quad (15)
\]

The main ingredient to get the hybridized weak formulation is to have an additional unknown, \( \lambda_h \), which approximates the solution on the edges of the mesh. In order to close the system, the continuity of the numerical fluxes across edges is required in a weak sense, resulting in an additional equation.
The weak formulation of the hybrid system, comprised of equations for the gradient \( q_h \), the solution itself \( w_h \) and its trace on the mesh skeleton \( \lambda_h \), is then given by:

Find \( x_h := (q_h, w_h, \lambda_h) \in X_h := (V_h, W_h, M_h) \) s.t. \( \forall y_h := (\tau_h, \varphi_h, \mu_h) \in X_h \)

\[
0 = N_h (x_h; y_h) \\
:= (\tau_h, q_h)_T + (\nabla \cdot \tau_h, w_h)_T - (\tau_h \cdot n, \lambda_h)_{\partial T_h} \\
- (\nabla \varphi_h, f_c(w_h) - f_v(w_h, q_h))_T + (\varphi_h, s(w_h, q_h))_T + \left( \varphi_h, \hat{f}_c - \hat{f}_v \right)_{\partial T_h} \\
+ \left( \mu_h, \hat{f}_c - \hat{f}_v \right)_{\Gamma_h} + N_h,\partial \Omega (q_h, w_h; \tau_h, \varphi_h).
\]

The respective function spaces are given by

\[
V_h = \{ v \in L^2 (\Omega) : v|_T \in P^p(T), T \in T_h \}^{c \times d} \\
W_h = \{ w \in L^2 (\Omega) : w|_T \in P^p(T), T \in T_h \}^c \\
M_h = \{ \mu \in L^2 (\Gamma_h) : \mu|_e \in P^p(e), e \in \Gamma_h \}^c.
\]

Thus, \( q, w \) and \( \lambda \) are approximated by piecewise polynomials of degree \( p \) which can be discontinuous across edges \( (q_h, w_h) \) or vertices \( (\lambda_h) \). We choose numerical fluxes comparable to the Lax-Friedrich flux and to the LDG flux for the convective and diffusive flux, respectively, i.e.

\[
\hat{f}_c (\lambda_h, w_h) = f_c (\lambda_h) \cdot n - \alpha_c (\lambda_h - w_h) \\
\hat{f}_v (\lambda_h, w_h, q_h) = f_v (\lambda_h, q_h) \cdot n + \alpha_v (\lambda_h - w_h)
\]

The boundary conditions are incorporated in an adjoint consistent manner (cf. [18]) by evaluating the analytical fluxes with the boundary conditions \( w_{\partial \Omega} \) applied to \( w_h \) and \( f_v,\partial \Omega \) to \( f_v \), i.e.

\[
N_h,\partial \Omega (q_h, w_h; \tau_h, \varphi_h) := \left( \tau_h \cdot n, w_{\partial \Omega} (w_h) \right)_{T_h} \\
+ \left( \varphi_h, (f_c (w_{\partial \Omega} (w_h)) - f_v,\partial \Omega (f_v (w_{\partial \Omega} (w_h), q_h))) \cdot n \right)_{T_h}.
\]

C. Relaxation and Hybridization

As relaxation method, a damped version of the Newton’s method has been used as given below.

\[
\left( N_h^n (x_h) - \frac{1}{\Delta t} \text{Id} \right) \delta x_h^n = -N_h (x_h^n).
\]

(12)

where \( \delta x_h^n \) is used to update for the solution until the residual \( N_h (x_h^n) \) drops below a certain threshold.

\[
x_h^{n+1} = x_h^n + \delta x_h^n
\]

(13)

Please note, that Newton’s method is formally recovered for \( \Delta t \to \infty \).

Using an appropriate polynomial expansion for \( \delta q_h, \delta w_h \) and \( \delta \lambda_h \), the linearized global system is given in matrix form as

\[
\begin{bmatrix}
A & B & R \\
C & D & S \\
L & M & N
\end{bmatrix}
\begin{bmatrix}
\delta Q \\
\delta W \\
\delta \Lambda
\end{bmatrix}
= \begin{bmatrix}
F \\
G \\
H
\end{bmatrix}
\]

(14)

where the vector \( [\delta Q, \delta W, \delta \Lambda]^T \) contains the expansion coefficients of \( \delta x_h \) with respect to the chosen basis. Please note that this vector is organized in such a manner that the degrees of freedom related to \( \delta q_h \) and \( \delta w_h \) are grouped together element-wise.

In order to carry on with the derivation of the hybridized method, we want to formulate that system in terms of \( \delta \Lambda \) only. Therefore we split it into

\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\begin{bmatrix}
\delta Q \\
\delta W
\end{bmatrix}
= \begin{bmatrix}
F \\
G
\end{bmatrix}
- \begin{bmatrix}
R \\
S
\end{bmatrix} \delta \Lambda
\]

(15)
and
\[
\begin{bmatrix}
L & M
\end{bmatrix}
\begin{bmatrix}
\delta Q \\
\delta W
\end{bmatrix} + N\delta \Lambda = H. \tag{16}
\]

Substituting Eq. (15) into Eq. (16) yields the hybridized system
\[
\left(\begin{bmatrix}
N - L & M
\end{bmatrix}
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^{-1}
\begin{bmatrix}
R \\
S
\end{bmatrix}
\right)\delta \Lambda = H - \begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^{-1}
\begin{bmatrix}
F \\
G
\end{bmatrix} \tag{17}
\]

The workflow is as follows: First, the hybridized system is assembled and then being solved for \(\delta \Lambda\). Then, \(\delta Q\) and \(\delta W\) can be reconstructed inside the elements via Eq. (15). It is very important to note that it is not necessary to solve the big system given by Eq. (15). In fact, the matrix in Eq. (15) is block diagonal whereat each block is associated to one element. Thus, both the assembly of the hybridized matrix in Eq. (17) and the reconstruction of \(\delta Q\) and \(\delta W\) can be done in an element-wise fashion.

### IV. Adjoint-Based Error Estimation

Often, the focus is not on the accuracy of the solution \(w_h\) of a partial differential equation but on some functional \(J : X \rightarrow \mathbb{R}\) acting on the solution situated in a suitable Hilbert Space \(X\). In aerospace applications lift and drag are often used as target functionals. In order to estimate the error of the approximated functional, we expand the target functional in a Taylor series as follows,
\[
e_h := J_h (x) - J_h (x_h) = J_h' [x_h] (x - x_h) + O \left( \|x - x_h\|^2 \right). \tag{18}
\]
Here \(x_h\) is the approximation to \(x\) in \(X_h\). We proceed in a similar manner with the error in the residual, i.e.
\[
N_h (x; y_h) - N_h (x_h; y_h) = N_h' [x_h] (x - x_h; y_h) + O \left( \|x - x_h\|^2 \right). \tag{19}
\]
As our discretization is consistent the first term \(N_h (x; y_h)\) vanishes.

Substituting Eq. (19) into Eq. (18) and neglecting the quadratic terms yield the so-called adjoint equation
\[
N_h' [x_h] (y_h; z_h) = J_h' [x_h] (y_h) \tag{20}
\]
with the dual solution \(z_h = (\tilde{q}_h, \tilde{w}_h, \tilde{\lambda}_h)\). \(z_h\) represents the link between variations in the residual and in the target functional.

Weighting the residual with the obtained adjoint solution results in an global error estimate
\[
e_h \approx \eta := N_h (x_h; z_h) \tag{21}
\]
which can then be restricted to a single element to yield a local indicator for mesh refinement, i.e.
\[
\eta_T := \left| N_h (x_h; z_h) \right|_T \tag{22}
\]
so that \(\eta \leq \sum_{T \in \mathcal{J}_h} \eta_T\) holds.

Please note, that the functionals \(N_h\) and \(J_h\) and their jacobians have to be evaluated in a somewhat richer space than \(X_h\), namely \(\bar{X}_h \supset X_h\). Otherwise, the weighted residual \(N_h (x_h; z_h)\) would be identical zero, as
\[
N_h (x_h; y_h) = 0 \quad \forall y_h \in X_h. \tag{23}
\]
This can be achieved by either mesh refinement or a higher polynomial degree for the ansatz functions. In our setting, especially when using a hierarchical basis, the latter is advantageous with respect to implementation effort and efficiency.

### V. Anisotropic H-Adaptation

The methodology is an extension of the mesh adaptation strategy of Dolejsi [8], incorporating the adjoint-based error estimate in finding the area of the mesh element. The original method of Dolejsi tries to generate
anisotropic meshes such that the interpolation error in the $L^q$-norm ($q \in [1, \infty)$) is under a given tolerance, and the number of degrees of freedom is the smallest possible. Equivalently, one can also set the area of the element and find the anisotropy which gives the minimum interpolation error in the $L^q$-norm.

The fundamental concept used in the mesh adaptation process is the mesh-metric duality [13]. The triangulation, $T_h$, can be characterized by the Riemannian metric field, which is a field of $2 \times 2$ (in 2D) symmetric, positive definite matrices, $M(\mathbf{x})$, over the physical domain. The mesh is then the image in the Euclidean space of a uniform mesh in a Riemannian metric space, where the distances are calculated. The length of a segment $\mathbf{ab}$ under the metric, $M$, is computed using a straight line parameterization and is given by

$$l_{M} = \int_{0}^{1} \sqrt{\mathbf{ab}^T M(\mathbf{a} + s\mathbf{ab}) \mathbf{ab}} \, ds$$

In a metric-conforming triangulation, the mesh will have edges which have the lengths close to unity under the given metric. This means

$$l_{M}(e) \approx 1$$

for every edge $e$ in the triangulation. For a given metric, $M$, the metric-conforming triangulation is not unique. However, they will have similar approximation properties since the edges have to meet the length constraints given by the metric. Note that in practice, the mesh generator creates such a mesh for which Eq. (25) is satisfied in a least square sense for the entire domain. Now, for a given mesh, we can also construct the unique metric which conforms with the mesh and we call this metric the implied metric. For each simplex mesh element $i$, the implied metric $M_i$ is a unique metric under which all the edges, $e_k$ ($k = 1, 2, 3$) of the elements are unit length. This means

$$e_k^T M_i e_k = 1, \quad k = 1, 2, 3$$

If the implied metric is denoted as,

$$M_i = \begin{pmatrix} a & b \\ b & c \end{pmatrix},$$

for the mesh element with edges $e_k$, and $(x_{e_k,1}, y_{e_k,1})$ and $(x_{e_k,2}, y_{e_k,2})$ are the vertices of each edge $e_k$, then Eq. 26 can be written as

$$a(x_{e,2} - x_{e,1})^2 + 2b(x_{e,2} - x_{e,1})(y_{e,2} - y_{e,1}) + c(y_{e,2} - y_{e,1})^2 = 1 \quad k = 1, 2, 3$$

The above equation is solved to get the components $(a, b, c)$ of the metric $M_i$. The anisotropy (stretching ratio and the orientation) of the mesh element can be defined with the help of the spectral decomposition of the metric, $M_i$. If we decompose the metric $M_i$ as,

$$M_i = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}^T \begin{pmatrix} \frac{1}{h_1} & 0 \\ 0 & \frac{1}{h_2} \end{pmatrix} \begin{pmatrix} \frac{1}{h_1} & 0 \\ 0 & \frac{1}{h_2} \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

then the eigen vectors give the direction of the principle axes of the ellipse, which encloses the mesh element, and the eigen values give the inverse of the square of the principle lengths, $h_1$ and $h_2$. Such an ellipse is shown in Fig. 1, with the major axis inclined at an angle $\theta$ and with the principal lengths as $h_1$ and $h_2$. The area, $I$ of the mesh element enclosed in the ellipse is evaluated [8] to be

$$I = \frac{\sqrt{3}}{4} h_1 h_2$$

If we define the aspect ratio of the mesh element as

$$\beta = h_2/h_1,$$

then $\{I, \beta, \theta\}$ uniquely defines the mesh element and the metric. The Aspect ratio, $\beta$ and the orientation $\theta$, together constitute the anisotropy of the triangle. Thus the metric completely defines the mesh by encoding in it both the area, $I$ and the anisotropy, $(\beta, \theta)$ of each mesh element in the physical space. In our adaptation
methodology, we evaluate the implied metric of the current mesh and we find the metric of the desired mesh by modifying the area and the anisotropy given by the implied metric. This new metric is then fed to the mesh generator which creates the new mesh which conforms with the given metric. Usually, we have to give the current mesh and the metric evaluated at each element or the vertices, as input to the mesh generator and it creates the new mesh. We have used BAMG mesh generator [11] for which the metric should be evaluated at the vertices of the mesh. This is done using a volume-weighted averaging of the elemental discrete metric.

A. Algorithm for H-Adaptation

The algorithm for h-adaptation works as follows.

Algorithm A

1. At each adaptation step, find the implied metric of the current mesh using Eq. (28), and evaluate the area of the mesh elements.

2. Determine the area, $I$, of the mesh elements for the desired mesh using adjoint-based error estimate and the isotropic size of the current mesh as explained in section B.

3. Find the anisotropy $\{\beta, \theta\}$ of the elements for the desired mesh, using the concept of the minimization of interpolation error, as explained in section D. Note that, in this process, we need to find the $(p+1)$-st derivative of the solution variable. Since the solution is only of degree $p$, we reconstruct the solution to polynomial degree $p + 1$. This is achieved using a patch reconstruction as given in section C.

4. Having found $\{I, \beta, \theta\}$, the metric $M$ is evaluated for all the mesh elements and is used as the input to the mesh generator to produce the desired mesh, conforming to the given metric. This new mesh is used for the next computation. This process is repeated till we get the mesh which keeps the target functional within the given error tolerance.

B. Finding the area of the desired mesh

The area, $I$, of the desired mesh element is calculated from the area, $I_e$ of the current mesh, and the adjoint error estimate such that the error in the target functional is expected to get reduced in the next iteration. A user specified mesh fraction, $f_h$ is used to find the cut-off error, $c_{err}$ and those elements which have error higher than $c_{err}$ get refined and those with lower error get coarsened. This is done in the following heuristic...
way for each element. For refinement, the area is reduced by a factor \( r_f \) as given below.

\[
r = \frac{\log_{10}(err) - \log_{10}(c_{err})}{\log_{10}(max_{err}) - \log_{10}(c_{err})}
\]

\[
r_f = (r_{\text{max}} - 1)r^2 + 1
\]

\[
I = \frac{I_c}{r_f}
\]

For coarsening, the area \( I \) is increased by a factor \( c_f \) as given below.

\[
c = \frac{\log_{10}(err) - \log_{10}(c_{err})}{\log_{10}(min_{err}) - \log_{10}(c_{err})}
\]

\[
c_f = (c_{\text{max}} - 1)c^2 + 1
\]

\[
I = I_c c_f
\]

where \( err \) is the local error given by the adjoint-based error estimate on the mesh element, \( max_{err} \) and \( min_{err} \) are the maximum and minimum of the element-wise error in the entire domain, respectively. \( r_{\text{max}} \) and \( c_{\text{max}} \) are the maximum refinement factor and the minimum coarsening factor, respectively. This means the area of the element with the maximum error gets reduced by a factor of \( r_{\text{max}} \) and that with the minimum error gets coarsened by a factor of \( c_{\text{max}} \).

C. Patch reconstruction

To find the \((p+1)\)-st derivative of the solution variable we need to reconstruct the solution from polynomial degree \( p \) to \( p+1 \). For this we use an \( H^1 \) patch reconstruction in which a patch is defined around each element and the solution is reconstructed in this patch. For hp-adaptivity, we also need to reconstruct the solution to degree \( p+2 \). For each element \( k \), the patch, defined as \( D(k) \), consists of the element and all the neighboring elements as given in Fig. 2. For the solution variable \( u \), we define the reconstructed solution \( \hat{u} \in P^{(p+r)}(D(k)), r = 1, 2 \).

\[
(\hat{u}, \phi)_{H^1} = (u, \phi)_{H^1}, \quad \forall \phi \in P^{(p+r)}(D(k))
\]

Figure 2: Patch for the element \( k \)

Fig. 3 compares the original solution and the reconstructed solutions for the boundary layer test case defined in section A. Here, the original solution which is in \( p = 1 \) is reconstructed to both \( p + 1 = 2 \) and \( p + 2 = 3 \) discretization space. We observe that the solution gets smoother with higher order reconstructions.
D. Finding the anisotropy of the desired mesh

We use the methodology proposed by Dolejsi [8] to find the anisotropy of the triangle. The basic concept is to find the anisotropy of the triangle, with a given area, which minimizes an interpolation error function in the $L_q$-norm, $q \in [1, \infty)$. We briefly reproduce here some of the essential details that are required to find the anisotropy of the triangle, as described in the paper of Dolejsi [8]. Those details include the definition of an interpolation error function and finding a set of parameters which can be used to derive an estimate for the interpolation error function. The set of parameters is called as the anisotropy of the interpolation error function, and they are in turn used to find the anisotropy of the triangle for which the interpolation error function is minimum in the $L_q$-norm. At first we define the directional derivative and the maximum directional derivative of a function $u$, which is needed to find the anisotropy of the interpolation error function of the same function $u$.

1. Directional derivative

For a sufficiently smooth function $u$, its $k$-th directional derivative along the direction $(\cos \phi, \sin \phi)$ is defined as

$$u^{(k,\phi)} = \sum_{l=0}^{k} \binom{k}{l} \frac{\partial^k u}{\partial x^l \partial y^{k-l}} (\cos \phi)^l (\sin \phi)^{k-l}$$ (39)
We define $\tilde{u}^{(p+1,\phi_m)}$ as the maximum $(p+1)$st directional derivative of the (reconstructed) anisotropy variable $\tilde{u}$, $\phi_m$ as the corresponding angle and $\tilde{u}^{(p+1,\phi_m)}_\perp$ as the value of the directional derivative perpendicular to $\phi_m$. We also define $\tilde{A}_p$ as the scaled maximum directional derivative of $\tilde{u}$, and $\tilde{\rho}_p$ as the ratio of the maximum derivative and the derivative along its perpendicular direction as given below.

$$
\tilde{A}_p = \frac{1}{(p + 1)!} \tilde{u}^{(p+1,\phi_m)} \quad \tilde{\rho}_p = \frac{\tilde{u}^{(p+1,\phi_m)}_\perp}{\tilde{u}^{(p+1,\phi_m)}}
$$

Note that, to get the high order directional derivatives using Eq. 39 in the physical space, we first evaluate the derivatives in the reference space, and then they are transformed into the physical space.

2. Interpolation error function

Given a sufficiently smooth function $u$, a point $\bar{x} = (\bar{x}, \bar{y}) \in \Omega$, and the polynomial degree, $p$, then a projector operator $\pi_{x,p}$ is defined such that

$$
\frac{\partial^k (\pi_{x,p} u(\bar{x}))}{\partial x^i \partial y^{k-i}} = \frac{\partial^k u(\bar{x})}{\partial x^i \partial y^{k-i}} \forall l = 0, ..., k \quad \forall k = 0, ..., p
$$

This means the projection $\pi_{x,p} u$, and the function $u$ has the same partial derivatives upto order $p$. Now, using Taylor series expansion around point $\bar{x}$, one can write for any point $x = (x, y)$,

$$
u(x) = \sum_{k=0}^{p+1} \frac{1}{k!} \left( \sum_{l=0}^{k} \binom{k}{l} \frac{\partial^k u(\bar{x})}{\partial x^l \partial y^{k-l}} (x - \bar{x})^l (y - \bar{y})^{k-l} \right) + O(|x - \bar{x}|^{p+2})
$$

Using the projection operator $\pi_{x,p}$ defined in Eq. 41, we can rewrite the above equation as

$$u(x) - \pi_{x,p} u(x) \approx E_{x,p}(x)
$$

where $E_{x,p}$ is the interpolation error function of degree $p$ located at $\bar{x}$, defined as

$$E_{x,p}(x) = \frac{1}{(p + 1)!} \sum_{l=0}^{p+1} \binom{l}{p+1} \frac{\partial^{p+1} u(\bar{x})}{\partial x^l \partial y^{p+1-l}} (x - \bar{x})^{l} (y - \bar{y})^{p+1-l}
$$

We aim to seek a triangle, having barycentre at $\tilde{x}$, with a given area, which has the least interpolation error, $E_{x,p}(x)$, in the $L_q$-norm, $q \in [1, \infty)$. For this, a set of parameters called the anisotropy of the interpolation error function needs to be defined. This is done in the next section.

3. Anisotropy of the interpolation error function

An estimate for the interpolation error function, $E_{x,p}(x)$ can be derived, in terms of three parameters $(A_p, \rho_p, \phi_p)$, as

$$|E_{x,p}(x)| \leq A_p \left( (x - \tilde{x})^T Q_{\phi_p} D_{\rho_p} Q_{\phi_p}^T (x - \tilde{x}) \right)^{\frac{p+1}{2}} \forall x \in \Omega
$$

where $A_p > 0$, $Q_{\phi_p}$ is the rotation through angle $\phi_p$, and $D_{\rho_p}$ is a diagonal matrix as given below.

$$Q_{\phi_p} = \begin{bmatrix} \cos \phi_p & -\sin \phi_p \\ \sin \phi_p & \cos \phi_p \end{bmatrix}, \quad D_{\rho_p} = \begin{bmatrix} 1 & 0 \\ 0 & \rho_p^{-\frac{p}{2}} \end{bmatrix}
$$

$A_p$, $\rho_p$ and $\phi_p$ represent the size, the aspect ratio and the orientation of the interpolation error function $E_{x,p}(x)$, respectively. They are defined in such a way that the estimate (Eq. 45) is sharp, in the sense that there exists $x \in \Omega$ such that in Eq. 45, the equality holds. The triplet $(A_p, \rho_p, \phi_p)$ is called as the anisotropy of the interpolation error function $E_{x,p}(x)$.

In most of the cases, $A_p$ is equal to $\tilde{A}_p$ - the scaled maximum directional derivative of $u$, and $\rho_p$ is equal to $\tilde{\rho}_p$, the ratio of the derivatives as given in Eq. 40. But in general, we may have to modify $\tilde{A}_p$ and, $\tilde{\rho}_p$ to get $A_p$ and $\rho_p$ that would satisfy Eq. (45). For detailed information on the determination of $\{A_p, \rho_p, \phi_p\}$ from $\{A_p, \tilde{\rho}_p, \phi_m\}$, we refer to the paper of Dolejsi [8]. Note that, for pure $h$-adaptations, we have used $\{A_p, \rho_p, \phi_p\} = \{A_p, \tilde{\rho}_p, \phi_m\}$. But for $hp$-adaptations, since we compare the interpolation error bounds for choosing the polynomial degrees, we want to be in the safer side in evaluating the bound and hence we modify $\{\tilde{A}_p, \tilde{\rho}_p, \phi_m\}$ to get $\{A_p, \rho_p, \phi_p\}$ that would satisfy Eq. (45) as sharply as possible.
4. Anisotropy of the triangle

The aim is to seek a triangle, with a given isotropic size, \( I \), which gives the least interpolation error in the \( L_q \)-norm, \( q \in [1, \infty) \). In other words, given the isotropic size, \( I \), find the anisotropy \( \{ \beta, \theta \} \) of the triangle for which \( \| E_{\bar{x},p} \|_{L^q(K)} \) is minimum. It turns out to be that, such an anisotropy \( \{ \beta, \theta \} \) for the triangle, for which the interpolation error is minimum, can be found using the anisotropy of the interpolation error function, \( \{ A_p, \rho_p, \phi_p \} \), defined in the previous section. Strictly speaking, the minimization is done over the ellipse, \( E \) which encloses the triangle, \( K \). But since, \( K \subset E \), we have the bound
\[
\| E_{\bar{x},p} \|_{L^q(K)} \leq \| E_{\bar{x},p} \|_{L^q(E)}.
\]

If \( K \) is the triangle defined by \( \{ I, \beta, \theta \} \), and \( \{ \beta, \theta \} \) is evaluated as,
\[
\beta = \rho_p^{\frac{1}{q+1}}, \quad \theta = \phi_p - \frac{\pi}{2},
\]
then it can be proved that \( \| E_{\bar{x},p} \|_{L^q(E)} \) is the minimum [8]. Furthermore, the bound for the interpolation error function \( E_{\bar{x},p} \) on element \( K \), is given as follows.
\[
\| E_{\bar{x},p} \|_{L^q(K)} \leq \omega \tag{48}
\]
where, for any \( q \in [1, \infty) \)
\[
\omega = c_{p,q} A_p \rho_p^{-0.5} I^{\frac{q+1}{q+2}} \tag{49}
\]
\[
c_{p,q} = \left( \frac{2\pi}{q(p+1)+2} \left( \frac{1}{\pi} \right)^\frac{q(p+1)+2}{2} \right)^{1/q} \tag{50}
\]
and for \( q = \infty \),
\[
\omega = A_p \rho_p^{-0.5} I^{\frac{p+1}{2}} \tag{51}
\]
and \( I \) is the isotropic size of the mesh element. In our case, we find the isotropic size using the adjoint error estimate as given in section B.

E. Generation of the metric and the mesh

Once \( \{ I, \beta, \theta \} \) are known, \( \{ h_1, h_2, \theta \} \) can be evaluated. From \( \{ h_1, h_2, \theta \} \), the metric \( \mathcal{M} \) can be evaluated using Eq. 29. The mesh generator BAMG is used to produce the new mesh which conforms to the metric. Since BAMG uses a slightly different parametric representation for the geometry, we have changed it to the one used in the NETGEN mesh generator. In this way the meshes generated by BAMG are conforming with the geometry representation in the NETGEN and hence they can be directly used in NETGEN for the next simulation. BAMG is also used to get a linearly interpolated solution on the new mesh from the solution on the current mesh. This new solution can be used for the next simulation to make the convergence faster.

VI. Anisotropic Hp-Adaptation

We extend the hp-adaptation strategy of Dolejsi to incorporate adjoint error estimate to choose the isotropic size. The original method of Dolejsi generates anisotropic meshes such that the interpolation error in the \( L^q \)-norm \( (q \in [1, \infty)) \) is under a given tolerance, and the number of degrees of freedom in minimum. Equivalently, one can also set the isotropic area of the element and find the anisotropy and the polynomial degrees which gives the minimum interpolation error. In our method, we set the isotropic size of the element using the adjoint error estimate like in the case of anisotropic h-refinement, and choose the configuration (anisotropy and the polynomial approximation) which gives the least interpolation error estimate.

A. Algorithm for Hp-Adaptation

The algorithm for hp-adaptation works as follows.

Algorithm B

1. We set the isotropic size, \( I \), of the triangle, \( K \) using the adjoint error estimate as given in section B.
2. If the anisotropy variable, \( u \) on element \( K \) is a polynomial of degree \( p_k \), then we consider three cases: the reconstructed variable \( \hat{u} \) to be polynomial of degrees \( p = p_k - 1, p_k, p_k + 1 \). In all the three cases, the maximum \((p + 1)\)st scaled directional derivative, \( \hat{A}_p \), the ratio \( \hat{\rho}_p \) and the angle \( \phi_m \) are found. This is used to calculate the anisotropy, \( \{A_p, \rho_p, \phi_p\} \) of the interpolation function \( E_{k,p} \) as given in section D.

Note: For \( p = p_k \) and \( p = p_k + 1 \), we need to reconstruct \( u \) to the spaces \( P^{p_k+1} \) and \( P^{p_k+2} \) to find the \((p + 1)\)st derivative. This is done using the patch reconstruction given in section C.

3. Evaluate the interpolation error bound \( \omega \) for the above three cases of polynomial approximations using Eq. 49 and Eq. 50 if the bound is measured in \( L_q, q \in [1, \infty) \), or by using Eq. 51 for \( L_q, q = \infty \).

4. Select, for each element, the \( p \) and the corresponding mesh, defined by the isotropic size \( I \), and the anisotropy information - \( \beta \) and \( \theta \), (using Eq. 47) which gives the smallest error bound \( \omega \). Note that, similar to the mesh fraction \( f_h \) in choosing the isotropic size, a user specified mesh fraction, \( f_p \) is also used now to find a cut-off error and those elements with a lower error than this cut-off error would retain the original polynomial degree, \( p_k \), even if the interpolation error bound suggests \( p_k+1 \). In other words, only if the element has an error (specified by the adjoint error estimator) above a certain cut-off error, will get p-refined.

VII. Numerical Results

In this section, the adaptation (both h and hp) is applied for different test cases to demonstrate the effectiveness of anisotropic h and hp-refinement method. The solver used is the Hybridized DG method as discussed in section (ref). The discretized equations are solved using the damped Newton method (section reference) and the resulting linear sub-problems are solved using restarted GMRES with ILU preconditioner. GMRES is also used for the linear adjoint problems. At first, a scalar convection diffusion is considered, where there is a strong boundary layer which is expected to be resolved well using anisotropic meshes. Then, in order to verify the method for aerodynamic applications, we solve Euler or Navier-Stokes equations over the NACA0012 airfoil.

Four cases of flow over NACA0012 airfoil are considered.

- Subsonic inviscid flow, \( M_\infty = 0.5, \alpha = 2^\circ \)
- Transonic inviscid flow, \( M_\infty = 0.8, \alpha = 1.25^\circ \)
- Supersonic inviscid flow, \( M_\infty = 1.5, \alpha = 0^\circ \)
- Subsonic viscous flow, \( M_\infty = 0.5, \alpha = 1.0^\circ, Re = 5000 \)

The geometry of the airfoil is defined by

\[
y = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4) \tag{52}\]

with \( x \in [0, 1] \). The initial mesh (Fig. 4) with 2155 triangular elements was created using the Netgen mesh generator [17]. The far field is a circle, centered at the airfoil mid chord with a radius of 1000 chords. This mesh has been used as the initial mesh for all the above test cases. In order to assess the effect of discretization orders on the error convergence rate, all the test cases are done for different solution spaces \((p = 1, 2, 3)\). Mach number has been used as the anisotropy variable for all the NACA0012 test cases. In addition to the anisotropic adaptation, we also do isotropic adaptation and the results are compared. For isotropic adaptation, we use fixed fraction marking strategy, and those elements which are marked are bisected to form child elements. We need to specify mesh fraction as a parameter for isotropic refinement. Typically 5 – 10% is used. In the convergence plots shown in the section on numerical results, ‘iso’ denotes isotropic refinement. Isotropic refinement is done using the Netgen mesh generator itself and no other mesh generators such as BAMG is used for this. For the anisotropic refinements, we have used BAMG mesh generator as explained before.

Now we give a short recap on the parameters that are to be defined for the anisotropic adaptation. For the h-adaptation, there are three parameters. They are the mesh fraction \( f_h \) that determines the cut-off error for refinement and coarsening, the maximum refinement size \( r_{\text{max}} \), and the maximum coarsening size \( c_{\text{max}} \).
In our simulations, we have used \( f_h \) to be around 0.3 for h-adaptation and around 0.1 for hp-adaptation, \( r_{max} \) in the range of 4 – 10 and \( c_{max} \) in the range of 2 – 6. For the same test case, but for different polynomial approximation degrees, we have used the same parameters. This makes the comparison fair, and we can check the impact of high order approximations on the efficiency in terms of the degrees of freedom. For hp-adaptation, in addition to the parameters just mentioned for the h-adaptation, there is only one more parameter - \( f_p \), the mesh fraction that determines the cut-off error for p-refinement. In our simulations, we have used \( f_p \) to be in the range of 0.2 – 0.3 and found to be working well. Exact values that were used for each test case are mentioned in the section of numerical results. For all the cases, interpolation error bound was estimated for \( q = 2 \), (\( L^2 \)-norm). For hp-adaptation on isotropic meshes, a jump sensor as given by Dolejsi [7], is used to choose between h and p-refinement.

### A. Scalar Boundary Layer

We consider the scalar convection-diffusion equation

\[
\nabla \cdot (w, w) - \epsilon \Delta w = s \quad (x, y) \in \Omega = [0,1]^2 \\
w(x, y) = 0 \quad (x, y) \in \partial \Omega
\]

We take the solution as

\[
w(x, y) = \left( x + \frac{e^{x/\epsilon} - 1}{1 - e^{1/\epsilon}} \right) \cdot \left( y + \frac{e^{y/\epsilon} - 1}{1 - e^{1/\epsilon}} \right),
\]

which forms a boundary layer for lower values of \( \epsilon \). Substituting this solution into the convection-diffusion equation gives a source function which we then use to solve the equation. The target functional of interest is the weighted total boundary flux i.e.

\[
J = \int_{\partial \Omega} \psi (w - \epsilon n \cdot q) d\sigma
\]

where the weighting function, \( \psi = \cos(2\pi x) \cos(2\pi y) \).

We have chosen \( \epsilon = 0.005 \), in which case there is a strong boundary layer on the top right corner of the domain. The presence of such a strong gradient in the solution makes it a suitable candidate to perform hp-adaptations. A uniform mesh with 512 elements (Fig. 6) has been used as the initial mesh. Anisotropic adaptations (both h and hp) have been performed and the results are compared with isotropic h-adaptations. Fig. 5 compares the convergence in error for isotropic h-adaptation and anisotropic (h and hp) adaptations. For h-adaptation, solution is approximated using polynomial of degree 2. For the anisotropic h-adaptation, \( f_h = 0.3 \), \( r_{max} = 8 \), \( c_{max} = 4 \) are used as the parameters, and a mesh fraction of 5% is used for isotropic
adaptation. For anisotropic hp-adaptation, the parameters used are $f_p = 0.3$, $f_h = 0.1$, $r_{max} = 8$, $c_{max} = 4$. With anisotropic h-refinement one could get an error of the order of $10^{-9}$ with just 1417 elements whereas the isotropic refinement needed 6931 elements to reach that error level. Fig. 5 ref shows the initial mesh (512 elements) and the adapted mesh after 2 adaptations (1417 elements). We can see the highly skewed elements near the boundary layer on the right side and on the top side. The solution is well-resolved with the adapted mesh as shown in Fig. 7. With hp-adaptation, we get the boundary layer resolved with almost the same number of mesh elements (554 elements) as that of the initial mesh (512 elements). In Fig. 8, the polynomial map is shown for the hp-adaptation. As expected in hp-adaptivity, we observe that elements on the boundary layer get higher polynomial degree and the interior gets the lower polynomial degree. In terms of the error convergence too, hp-adaptation works well as evident from Fig. 5. Fig. 9 shows the solution near the boundary layer on both an h-adapted mesh (1417 elements) and an h-adapted mesh (554 elements). On both of these meshes, we get almost the same error level of $\approx 10^{-9}$. On the hp-adapted mesh, even though its much coarser than the h-adapted mesh, we get a smooth solution and the same error level because of the higher polynomial approximation in the region.

Figure 5: Scalar boundary layer, $\epsilon = 0.005$, Comparison of error convergence for isotropic and anisotropic h- and hp-adaptations. ‘iso’ denotes isotropic refinement. For hp, $p = 2, \ldots, 6$

Figure 6: Meshes for the scalar boundary layer case, $\epsilon = 0.005$, $p = 2$, h-adaptation
(a) Solution on initial mesh of 512 elements  
(b) Solution on h-adapted mesh of 1417 elements

Figure 7: Scalar boundary layer, $\epsilon = 0.005$, $p = 2$, h-adaptation

(a) Polynomial map

Figure 8: Scalar boundary layer, $\epsilon = 0.005$, hp-adaptation, 3rd adapted step, 554 elements
B. Subsonic Inviscid Flow over NACA0012 Airfoil

The steady state Euler equations have been solved for the flow around the NACA 0012 airfoil with a free stream Mach number of $M_\infty = 0.5$ and an angle of attack of $\alpha = 2^\circ$. The drag coefficient has been used as the target functional for the adjoint-based adaptations. Both anisotropic and isotropic adaptations are performed on an initial mesh with 2155 mesh elements (Fig. 4). For anisotropic h-adaptation the parameters used are $f_h = 0.3$, $r_{max} = 10$, $c_{max} = 2$ and for isotropic refinement, a mesh fraction of 5% was used for the refinement. For anisotropic hp-adaptation, $f_p = 0.2$, $f_h = 0.1$, $r_{max} = 8$, $c_{max} = 4$ are used as the adaptation parameters. Fig. 10 shows the Mach number contours obtained on the h-adapted mesh (2655 elements) for $p = 3$ discretization. As expected, the adapted mesh has h-refined regions near the trailing edge where there is a singularity, and the leading edge due to the high gradient of the solution, which would result in more accurate drag coefficient values. Fig. 11 shows the Mach number contours and the corresponding polynomial map in the case of hp-adaptation. Fig. 12 shows the polynomial map for isotropic and anisotropic hp-refinement. It is interesting to note that the polynomial distribution look similar though both use different approach in choosing the polynomial degree; isotropic refinement uses the jump indicator and anisotropic refinement uses the interpolation error estimate. Fig. 13 compares the convergence of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp-adaptation.

C. Transonic, Inviscid Flow over NACA0012 Airfoil

We consider transonic, inviscid flow around the NACA0012 airfoil with a free-stream Mach number of $Ma_\infty = 0.8$, angle of attack, $\alpha = 1.25^\circ$. The main features of this flow are the strong shock on the upper side of the airfoil, a weak shock on the lower side and a singularity at the trailing edge. Euler equations are solved with shock capturing term active. Drag coefficient is used as the target functional. Initial mesh with 2155 elements is shown in Fig. 4.

For anisotropic h-adaptation the parameters used are $f_h = 0.3$, $r_{max} = 6$, $c_{max} = 4$ and for isotropic
refinement, a mesh fraction of 2% was used for the refinement. For anisotropic hp-adaptation, $f_p = 0.2$, $f_h = 0.1$, $r_{max} = 4$, $c_{max} = 4$ are used as the adaptation parameters. Fig. 14 shows the h-adapted mesh (4394 elements) and the Mach number contours on it for $p = 2$ discretization. As one would expect, both shocks (upper and lower) regions got refined with anisotropic meshes and the shocks are captured sharply. Fig. 15 shows the hp-adapted mesh, Mach number contours and the corresponding polynomial map for the hp-adaptive case. Fig. 16 compares the convergence of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp-adaptation. To compute the error in the drag coefficient, a reference value of $c_{d,ref} = 2.265319 \times 10^{-2}$ was used which was obtained from the adapted (isotropic) mesh with 76740 degrees of freedom and solution approximation degree of $p = 2$. In this case, high order ($p > 1$) does not seem to perform better than lower order method ($p = 1$) for both isotropic and anisotropic adaptations. This is probably due to the presence of singularity and the strong shock attached to the airfoil. However, if we compare isotropic and anisotropic refinement, then certainly there is an advantage for anisotropic refinement as evident from the convergence plot.
D. Supersonic, Inviscid Flow over NACA0012 Airfoil

We consider supersonic, inviscid flow around the NACA0012 airfoil with a free-stream Mach number of $Ma_\infty = 1.5$, angle of attack, $\alpha = 0^\circ$. The governing equations are the Euler equations. For this case, a strong and detached bow shock is formed in front of the airfoil. This has to be captured sharply with the help of anisotropic mesh elements. The target functional considered is again the drag coefficient, $c_d$. Initial mesh with 2155 elements is shown in Fig. 4.

For the anisotropic h-adaptation the parameters used are $f_h = 0.3$, $r_{\text{max}} = 6$, $c_{\text{max}} = 4$ and for isotropic refinement, a mesh fraction of 2% was used for the refinement. For anisotropic hp-adaptation, $f_p = 0.2$, $f_h = 0.1$, $r_{\text{max}} = 4$, $c_{\text{max}} = 4$ are used as the adaptation parameters. Fig. 17 shows the h-adapted mesh (7380 elements) and the Mach number contours on it for $p = 2$ discretization. We can see how the whole mesh got aligned with the flow features, especially the highly anisotropic mesh at the bow shock region. The mesh outside the Mach cone got coarsened since they don’t contribute to the error in the drag coefficient. Fig. 15 shows the hp-adapted mesh, Mach number contours and the corresponding polynomial map for the hp-adaptive case. Fig. 19 compares the convergence of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp-adaptation. The region outside the Mach cone...
cone got coarser mesh and lower polynomial degree which is consistent with the physics of the flow. This region is not ‘aware’ of the presence of the airfoil due to the supersonic nature of the flow. As expected, the region between the airfoil and the shock got higher polynomial degree. To compute the error in the drag coefficient, a reference value of $c_{d,ref} = 9.629886 \times 10^{-2}$ was used which was obtained from an isotropically adapted fine mesh with 411348 degrees of freedom for solution approximation of degree $(p=2)$. For the isotropic refinement, there doesn’t seem to be any difference in the convergence rate for different polynomial approximations. For anisotropic refinement, $p = 2$, have better convergence rate than $p = 1$. With higher $p$, $p > 2$, the convergence does not seem to get better, probably due to the low regularity of the dominant flow features. This is also reflected in the hp-adaptation, where the overall convergence rate is not really superior to pure h-adaptation with $p = 2$.

E. Subsonic, Viscous Flow over NACA0012 Airfoil

We consider subsonic, viscous flow around the NACA0012 airfoil with a free-stream Mach number of $M_\infty = 0.5$, angle of attack, $\alpha = 1^\circ$ and Reynolds number of $Re = 5000$. The essential feature of this flow is the thin, laminar boundary layer over the airfoil. Governing equations are the Navier-Stokes equations (ref). Drag coefficient is used as the target functional. Initial mesh with 2155 elements is shown in Fig. 4. For anisotropic h-adaptation, the parameters used are $f_h = 0.3$, $r_{max} = 6$, $c_{max} = 4$. For anisotropic hp-adaptation, $f_p = 0.3$, $f_h = 0.1$, $r_{max} = 6$, $c_{max} = 4$ are used as the parameters. Fig. 20 shows the h-adapted mesh (11945 elements) and the Mach number contours on it for $p=2$ discretization. We can see the highly skewed mesh elements on the boundary layer and the Mach number contours on the layer which are very well captured. An interesting feature of the flow that is very-well captured with the help of anisotropic elements is also shown as streamlines in Fig. 21. It is the recirculation of flow near the trailing edge on the upper side of the airfoil. In the mesh, above the airfoil, we can see a thin layer where there is a strong anisotropy and this perfectly passes through the region where the streamlines change its direction from left to right.

Fig. 22 shows the adapted mesh, Mach number contours and the corresponding polynomial map for the hp-adaptive case. The boundary layer region around the airfoil, the stagnation region near the leading edge and the wake in the region downstream of the airfoil get higher polynomial degree. Similar to the pure h-adaptation, we can also see the a layer of skewed meshes near the trailing edge to resolve the recirculation of flow (Fig. 23). Fig. 24 shows the convergence of the error in the drag coefficient for the isotropic and anisotropic refinement for $p = 1, 2, 3$ and for the anisotropic hp-adaptation. To compute the error in the drag coefficient, a reference value of $c_{d,ref} = 5.531683 \times 10^{-2}$ was taken from a fine adapted mesh with 511905 degrees of freedom and the solution approximation degree $p = 4$. Anisotropic refinement is found to
Figure 15: Inviscid transonic flow around NACA0012, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, Hp-adaptation, $p = 1,..,6$

Figure 16: Inviscid transonic flow, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, Comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic refinement. For hp-adaptation, $p = 1,..,6$
Figure 17: Inviscid supersonic flow around NACA0012, $M_\infty = 1.5$, $\alpha = 0^\circ$, $p = 2$, h-adaptation with drag coefficient as target.

be much more efficient than the isotropic one in terms of the number of degrees of freedom. Also, increasing the polynomial degree makes the convergence rate better. We observe that the Hp-adaptivity works quite well for this test case and the convergence plot makes a perfect envelope over the h-adaptive plots.

**VIII. Conclusions**

An adaptation strategy on anisotropic meshes is presented for general convection-diffusion equations, including both Euler and Navier-Stokes equations in 2D. For the scalar boundary layer test case, and the subsonic flow (both inviscid and viscous) around the NACA0012 airfoil, both h and hp adaptations on anisotropic meshes could perform quite well and they were significantly more efficient than the adaptations on isotropic meshes in terms of the number of degrees of freedom required to achieve the same level of error. We could also observe that the anisotropic meshes could resolve the flow features, like the shocks and the flow recirculation, very well. Increasing the polynomial order could also make the adaptations more efficient. Overall, as one would expect, hp-adaptations performed better than the pure h-adaptations for those test cases. However, for the transonic and the supersonic inviscid flows, even though the shocks and the other flow features were resolved well, overall gain in efficiency in reducing the target error were not much with either by increasing the polynomial degree or by using the hp-refinement. Future work would aim at exploring ways to tackle flows with shocks and find if hp-adaptivity can work better also for such cases. Extending the methodology for 3D cases is also a part of the future work.
Figure 18: Inviscid supersonic flow around NACA0012, $M_\infty = 1.5$, $\alpha = 0^\circ$, hp-adaptation with drag coefficient as target, $p = 1,.., 6$.

**Acknowledgments**

Financial support from the Deutsche Forschungsgemeinschaft (German Research Association) through grant GSC 111, is gratefully acknowledged.
Figure 19: Supersonic flow, $M_\infty = 1.5$, $\alpha = 0^\circ$. Comparison of error convergence for isotropic and anisotropic adaptations. ‘iso’ denotes isotropic refinement. For hp-adaptation, $p = 1, .., 6$

Figure 20: Viscous subsonic flow around NACA0012, $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, $p = 2$, h-adaptation with drag coefficient as target.

References


Figure 21: Viscous subsonic flow around NACA0012, $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, $p = 2$, h-adaptation with drag coefficient as target.


Figure 22: Viscous subsonic flow around NACA0012, $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, hp-adaptation with drag coefficient as target. $p = 1, \ldots, 6$


Figure 23: Viscous subsonic flow around NACA0012, $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$, hp-adaptation with drag coefficient as target. $p = 1, \ldots, 6$

Figure 24: Viscous laminar flow, $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re = 5000$. Comparison of error convergence for isotropic and anisotropic adaptations. 'iso' denotes isotropic refinement. For hp-adaptation, $p = 1, \ldots, 6$