A p-adaptation method for compressible flow problems using a goal-based error indicator

D. Ekelschot, D. Moxey, S.J. Sherwin and J. Peiró[1]

Department of Aeronautics, Imperial College London, London SW7 2AZ, UK.

Abstract

An accurate calculation of aerodynamic force coefficients for a given geometry is of fundamental importance for aircraft design. High-order spectral/hp element methods, which use a discontinuous Galerkin discretisation of the compressible Navier-Stokes equations, are now increasingly being used to improve the accuracy of flow simulations and thus the force coefficients. To reduce error in the calculated force coefficients whilst keeping computational cost minimal, we propose a p-adaptation method where the degree of the approximating polynomial is locally increased in the regions of the flow where low resolution is identified using a goal-based error estimator as follows.

Given an objective functional such as the aerodynamic force coefficients, we use control theory to derive an adjoint problem which provides the sensitivity of the functional with respect to changes in the flow variables, and assume that these changes are represented by the local truncation error. In its final form, the goal-based error indicator represents the effect of truncation error on the objective functional, suitably weighted by the adjoint solution. Both flow governing and adjoint equations are solved by the same high-order method, where we allow the degree of the polynomial within an element to vary across the mesh.

We initially calculate a steady-state solution to the governing equations using

URL: `(d.ekelschot12,d.moxey,s.sherwin,j.peiro)@imperial.ac.uk` (D. Ekelschot, D. Moxey, S.J. Sherwin and J. Peiró)

[1] Corresponding author

a low polynomial order and use the goal-based error indicator to identify parts of the computational domain that require improved solution accuracy which is achieved by increasing the approximation order. We demonstrate the cost-effectiveness of our method across a range of polynomial orders by considering a number of examples in two- and three-dimensions and in subsonic and transonic flow regimes. Reductions in both the number of degrees of freedom required to resolve the force coefficients to a given error, as well as the computational cost, are both observed in using the p-adaptive technique.

Keywords: High-order spectral/hp element method, discontinuous Galerkin, adjoint methods, p-adaptation, goal-based error estimation.

1. Introduction

A problem of significant interest to the aeronautics industry is the development of numerical methods that are capable of accurately determining the lift or drag coefficient of a given wing geometry, while keeping the computational cost as low as possible. The value of these coefficients is highly dependent on the surrounding flow properties, as well as the geometry under consideration. The key to obtaining accurate values for these coefficients therefore lies in determining the areas within the domain that have the greatest effect on the value of the lift or drag coefficient. In other words, determining the sensitivity of the lift or drag coefficients with respect to its surroundings tells us where the local accuracy of the solution should be enhanced. Increasing the resolution in these regions permits us to evaluate the quantity of interest more accurately and improve the efficiency of the simulation.

Goal-based error estimation is a technique that is based around this philosophy, providing an indication of the accuracy of a numerical solution that is based on a pre-defined target quantity of interest, such as the lift and drag coefficients. It relies on the concept of duality, in which an adjoint problem is derived from the governing equations. The solution to this adjoint problem represents the sensitivity to an infinitesimal perturbation on the target, and a local error indicator is defined as the inner product of the residual and the corresponding adjoint variable.

The resulting error indicator provides a way to adaptively increase computational resolution only in the regions of the domain where additional accuracy is needed, which keeps computational costs lower. There is a rich catalogue of literature available regarding the application of goal-based error estimation for mesh adaptation, which is discussed in the review by Fidkowski and Darmofal [1]. As a brief overview, the idea of using adjoint equations for goal-based error estimation with the finite element method has been outlined by Becker and Rannacher [2] and Larson and Barth [3]. Giles and Pierce [4] described how to make use of the concept of duality for optimising objective functionals for typical computational fluid dynamics problems such as lift and drag force coefficients. Applications of goal-based error estimation to compressible inviscid flow problems using a finite element discretisation is further described in [5, 6]. Furthermore, goal-based adaptation for inviscid supersonic flow problems discretised using a finite volume discretisation was presented by Venditti and Darmofal [7].

However, the aforementioned works rely on h-refinement to drive the adaptation process, whereby elements within the mesh that represent the computational domain are subdivided, thereby reducing their size, increasing resolution and obtaining solutions of greater accuracy. More recently however, the use of high-order finite element methods, such as the spectral/hp element method, is becoming increasingly popular in the investigations of these aeronautics problems. These methods typically utilise high-order polynomial approximations on each element, as opposed to the traditional linear shape functions. They therefore possess a variety of properties that make them attractive in fluid research applications, including low numerical diffusion and dispersion characteristics, highly-scalable parallel implementations on modern hardware and the ability to obtain higher accuracy solutions at levels of computing time comparable to more traditional, lower order finite element and finite volume methods.

The use of high-order methods opens an alternative route to drive the adaptive process, which is the focus of this paper. Instead of subdividing elements, we may instead choose to vary the polynomial order, P, within a given element in order to either increase or decrease the computational resolution. Whilst high-order methods have been used in combination with goal-based error estimation and h-refinement, as can be seen in, for example [8], p-refinement has received far less attention, and work has classically focused on elliptic problems. Demkowicz et al. [9] proposed a fully automatic h- and p-adaptation strategy that was initially applied to elliptic problems where the projection based interpolation error of a fine reference solution was minimised. This was later on extended by incorporating a dual problem in the work of Solín and Demkowicz [10]. The potential benefits of using p-adaptation for compressible flows has been discussed by Li and Jameson [11], who compare h- and p-adaptation for external flow problems in the framework of the spectral differences. They found that p-adaptation provides the highest accuracy with respect to the number of degrees of freedom and CPU time. Recently, Giorgiani et al. [12] studied the propagation of waves in a harbour for which they proposed a p-adaptive hybridisable discontinuous Galerkin (HDG) approach. They were able to show that this approach outperforms the classical continuous Galerkin approach. However, solutions with shocks require h-adaptation instead of p-adaptation to avoid numerical oscillations. More recently, the combination of h- and p-adaptation for compressible flows has been explored in reference [13].

The purpose of this work is to build on this body of knowledge and describe a novel scheme that can highlight the ability of p-adaptation, in combination with goal-based error estimation, to more accurately calculate aerodynamic force coefficients at a low computational cost. In particular, we present a high-order spectral/hp discontinuous Galerkin formulation of the compressible Euler and Navier-Stokes equations, which describes the underlying equations, adjoint problem and their implementation in detail. We will then consider a variety of examples of inviscid and viscous external flows in both two- and three-dimensions, lying in both subsonic and transonic flow regimes, in order to examine the effectiveness of the p-adaptive method.

We conclude the introduction with a brief outline. First, the governing equations are introduced in section 2. This is followed by a description of the goal-based error indicator in section 3 and a derivation of the continuous adjoint equations for the compressible Navier-Stokes equations in section 4. An outline of the high-order discretisation of the governing equations and corresponding adjoint equations is given in section 5. Finally, the effectiveness of goal-based p-adaptation is assessed using a set of two- and three-dimensional numerical examples in section 6. This set consist of three cases of two-dimensional flow past a NACA0012 aerofoil section: a subsonic inviscid flow ($Ma = 0.4$) at an incidence of five degrees, a subsonic laminar flow ($Ma = 0.1, Re = 5\,000$) at an angle of incidence of two degrees, and a transonic inviscid flow ($Ma = 0.8$) at an incidence of 1.25 degrees. The set is completed by a case involving the three-dimensional inviscid flow ($Ma = 0.5$) past an ellipsoid at an incidence of three degrees. Based on these results, we draw conclusions on the performance of the proposed p-adaptation strategy in section 7.

2. Governing equations

We consider a compressible flow in which the physical laws of conservation of mass, momentum and energy for fluids in a domain, Ω, are described using the compressible Navier-Stokes equations

$$\boldsymbol{R}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \left\{ \boldsymbol{f}_i^c(\boldsymbol{u}) - \boldsymbol{f}_i^v(\boldsymbol{u}, \nabla \boldsymbol{u}) \right\} = \boldsymbol{0}; \qquad \boldsymbol{u} \in \Omega \tag{1}$$

in a two-dimensional Cartesian frame of reference with coordinates (x_1, x_2). The vector of conserved variables is given by $\boldsymbol{u} = \{u_1, u_2, u_3, u_4\}^t = \{\rho, \rho v_1, \rho v_2, \rho E\}^t$ where ρ is the density, v_1 and v_2 are the Cartesian components of the velocity $\vec{v}$, and E is the total energy. Here $\boldsymbol{R}(\boldsymbol{u}, \nabla \boldsymbol{u})$ is used to denote the differential operator representing the governing equations with components $\boldsymbol{R} = \{R_1, R_2, R_3, R_4\}^t$.

The Cartesian components of the convective fluxes, $\boldsymbol{f}_1^c$ and $\boldsymbol{f}_2^c$, are given by

$$\boldsymbol{f}_1^c = \left\{ \begin{matrix} \rho v_1 \\ p + \rho v_1^2 \\ \rho v_1 v_2 \\ \rho v_1 H \end{matrix} \right\}, \quad \boldsymbol{f}_2^c = \left\{ \begin{matrix} \rho v_2 \\ \rho v_1 v_2 \\ p + \rho v_2^2 \\ \rho v_2 H \end{matrix} \right\} \tag{2}$$

where H is the total enthalpy and P is the pressure. The viscous fluxes are given by

$$\boldsymbol{f}_1^v = \left\{ \begin{matrix} 0 \\ \tau_{11} \\ \tau_{21} \\ v_1 \tau_{11} + v_2 \tau_{21} - k \frac{\partial T}{\partial x_1} \end{matrix} \right\}, \quad \boldsymbol{f}_2^v = \left\{ \begin{matrix} 0 \\ \tau_{12} \\ \tau_{22} \\ v_1 \tau_{12} + v_2 \tau_{22} - k \frac{\partial T}{\partial x_2} \end{matrix} \right\} \tag{3}$$

where T is the temperature, k is the thermal conductivity and τ is the tensor of viscous stresses, defined component-wise as

$$\begin{aligned} \tau_{11} &= \frac{4}{3}\mu \left(\frac{\partial v_1}{\partial x_1} - \frac{1}{2}\frac{\partial v_2}{\partial x_2} \right) \\ \tau_{12} = \tau_{21} &= \mu \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} \right) \\ \tau_{22} &= \frac{4}{3}\mu \left(\frac{\partial v_2}{\partial x_2} - \frac{1}{2}\frac{\partial v_1}{\partial x_1} \right). \end{aligned} \tag{4}$$

The total enthalpy given in the convective flux definition is defined as

$$H = E + \frac{p}{\rho} \tag{5}$$

To close the system, the pressure for a perfect gas is given by

$$p = (1 - \gamma)\rho \left(E - \frac{v_1^2 + v_2^2}{2} \right) \tag{6}$$

where $\gamma = \frac{c_v}{c_p}$ is the ratio of specific heats and its value for air is $\gamma = 1.4$. We will also consider inviscid flow problems where the effects of viscosity and heat conduction are neglected. In this case, we are dealing with the compressible Euler equations for which the viscous flux terms are neglected in equation (1).

2.1. Boundary conditions

For compressible viscous flow, fluid wets a solid wall and thus we apply a no-slip boundary condition at the wall boundary such that

$$v_1 = v_2 = 0 \tag{7}$$

Further we assume an isothermal wall condition where the temperature at the wall, T_w, is prescribed, i.e.

$$T = T_w. \tag{8}$$

We will also consider compressible inviscid flows for which the viscous fluxes, $\boldsymbol{f}_1^v$ and $\boldsymbol{f}_2^v$ in equation (1), are neglected. In this case, we consider a slip wall boundary condition meaning that

$$\vec{v} \cdot \vec{n} = 0 \tag{9}$$

The use of a finite domain to model flows about aerofoils in free flight requires suitable conditions at the far-field boundary to appropriately enforce free-stream flow conditions and to ensure that flow disturbances propagate across the far-field boundary without spurious reflections that might lead to significant numerical errors. Here we impose far-field boundary conditions through the utilisation of exact or approximate Riemann solvers to calculate the fluxes normal to the far-field boundary. The calculation of these fluxes is described in more detail in section 5.1.

3. Goal-based error indicator

The adaptation approach adopted here is to increase the polynomial order of the elemental approximation in parts of the computational domain where the presence of numerical errors affecting the value of a quantity of interest, such as lift or drag, is highlighted by a goal-based error indicator.

A goal-based indicator estimates the error in the quantities of interest represented by a “goal” or “target” functional using duality techniques [1, 2, 3, 4] in which the adjoint to the governing equations describes the sensitivity of the target functional to perturbations of representative parameters of the problem. We consider a target functional that we assume depends on the state variables only and denote it by $J(\boldsymbol{u})$. The error in target functional is determined both by the error in the state variables and the truncation error that is introduced when approximating the target functional $J(\boldsymbol{u})$.

Following the approach proposed by Giles and Pierce [4], we write the error in the target functional, ϵ_J, as

$$\begin{aligned} \epsilon_J = \{J_\delta(\boldsymbol{u}+\delta\boldsymbol{u}) - J(\boldsymbol{u})\} &= \{J_\delta(\boldsymbol{u}+\delta\boldsymbol{u}) - J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta))\} \\ &+ \{J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta)) - J(\boldsymbol{u})\} \end{aligned} \tag{10}$$

where $J(\boldsymbol{u})$ denotes the exact solution for the target functional and we have used the subscript δ to indicate that we are dealing with a discrete operator. We observe that the discrete target functional J_δ is evaluated using different terms that require definition. The first evaluation uses the discrete solution $\boldsymbol{u}+\delta\boldsymbol{u}$, where $\delta\boldsymbol{u}$ represents the discretisation error incurred through numerical approximation of the exact solution $\boldsymbol{u}$. J_δ is also evaluated using $\boldsymbol{u}(\boldsymbol{x}_\delta)$, the exact solution at the discrete coordinates.

The right-hand side of equation (10) is written in this particular form to illustrate the two contributions of error in the defined target functional. The first contribution, $J_\delta(\boldsymbol{u}+\delta\boldsymbol{u}) - J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta))$, is due to the error in the discrete solution, $\delta\boldsymbol{u}$, and the second contribution, $J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta)) - J(\boldsymbol{u})$, is the truncation error in approximating the target functional [4].

We are interested in the first contribution to the error since it is the only one we can evaluate, and consider that the variation in J can be written as

$$\delta J = J_\delta(\boldsymbol{u} + \delta\boldsymbol{u}) - J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta)) \tag{11}$$

This variation in the target functional, δJ, is approximated as

$$\delta J \approx \left\{ \frac{\partial J_\delta}{\partial \boldsymbol{u}} \right\}^t \delta\boldsymbol{u} \tag{12}$$

where for two-dimensional compressible flows, $\left\{ \frac{\partial J_\delta}{\partial \boldsymbol{u}} \right\}$ is a column vector with four entries given by

$$\left\{ \frac{\partial J_\delta}{\partial \boldsymbol{u}} \right\}_i = \frac{\partial J_\delta}{\partial u_i}\,; \quad i = 1, .., 4 \tag{13}$$

Furthermore, the discretisation error leads to a non-zero residual vector, denoted by $\delta\boldsymbol{R}$, which is a column vector with four entries. We evaluate the effect of the discretisation error on the governing equations in a similar way to equation (11) and write

$$\delta\boldsymbol{R} = \boldsymbol{R}_\delta(\boldsymbol{u} + \delta\boldsymbol{u}) - \boldsymbol{R}_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta)) \tag{14}$$

Using a Taylor series expansion truncated to first order, we approximate the variation $\delta\boldsymbol{R}$ as

$$\delta\boldsymbol{R} \approx \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} \right] \delta\boldsymbol{u} \tag{15}$$

In this case $\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} \right]$ is a 4 × 4 matrix with entries given by

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} \right]_{i,j} = \frac{\partial R_i}{\partial u_j}\,; \quad i, j = 1, ..., 4 \tag{16}$$

To facilitate the identification of the nature of the various terms involved in the analysis that follows, we will use $\{\circ\}$ to denote a column vector with four entries and $[\circ]$ for a 4×4 matrix. Rewriting equation (15) we get

$$\delta \boldsymbol{u} \approx \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right]^{-1} \delta \boldsymbol{R} \tag{17}$$

We can then substitute this expression in equation (12) to link the variation in target functional δJ to the residual $\delta \boldsymbol{R}$

$$\delta J \approx \left\{\frac{\partial J_\delta}{\partial \boldsymbol{u}}\right\}^t \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right]^{-1} \delta \boldsymbol{R} \tag{18}$$

where the term $\left\{\frac{\partial J}{\partial \boldsymbol{u}}\right\}^t \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right]^{-1}$ represents a vector of four entries which we generally write as

$$\boldsymbol{\psi}^t = \left\{\frac{\partial J_\delta}{\partial \boldsymbol{u}}\right\}^t \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right]^{-1} \tag{19}$$

Here, $\boldsymbol{\psi}^t$ incorporates the sensitivities of the local residual with respect to the error in target functional, δJ.

3.1. Discrete approximation of the functional error

Since the exact solution, $\boldsymbol{u}$, required for the evaluation of $\delta \boldsymbol{R}$ in the target functional error equation (18) is not known, we estimate the error using two discretisation levels as presented in [14]. In our case, these levels are obtained by using the same mesh, but at two different polynomial orders. We denote the solution obtained using a high polynomial order by $\boldsymbol{u}_H$ and the solution at low polynomial order by $\boldsymbol{u}_L$. In order to estimate the error in target functional, we extrapolate the low-order solution, $\boldsymbol{u}_L$ onto the high-order solution using the following notation

$$\boldsymbol{u}_H^L \equiv \boldsymbol{I}_H^L \boldsymbol{u}_L \tag{20}$$

where $\boldsymbol{I}_H^L$ represents the low-to-high extrapolation operator. The low-order target functional is $J_\delta(\boldsymbol{u}_H)$ and we can use the extrapolated solution $\boldsymbol{u}_H^L$ to measure the output error in the low-order solution relative to the high-order solution by

$$\delta J \approx J_\delta(\boldsymbol{u}_H^L) - J_\delta(\boldsymbol{u}_H) \tag{21}$$

We can approximate $\delta\boldsymbol{R}$ using the governing equation $\boldsymbol{R}_\delta(\boldsymbol{u}_H) = \boldsymbol{0}$ so that

$$\delta\boldsymbol{R} \approx \boldsymbol{R}_\delta(\boldsymbol{u}_H^L) - \boldsymbol{R}_\delta(\boldsymbol{u}_H) = \boldsymbol{R}_\delta(\boldsymbol{u}_H^L) \tag{22}$$

Now we can substitute equation (21) and (22) into equation (18) to link the approximate output error to the local residual induced by the extrapolated solution $\boldsymbol{u}_H^L$ and obtain

$$J_\delta(\boldsymbol{u}_H^L) - J_\delta(\boldsymbol{u}_H) = \boldsymbol{\psi}^t\boldsymbol{R}_\delta(\boldsymbol{u}_H^L) = \sum_{e=1}^{N_{el}} \left(\boldsymbol{\psi}^t\boldsymbol{R}_\delta(\boldsymbol{u}_H^L)\right)_e \tag{23}$$

where N_{el} is the number of elements in the mesh. This gives a practical method for the calculation of the error as the sum over the elements in the mesh of the inner product of the sensitivities with the residuals evaluated using the projection of the low-order solution onto the high-order solution. This inner product is calculated for each element separately.

To define the goal-based error indicator, we take the absolute value of the inner product of the sensitivities with the corresponding residuals for each element [1, 14], this is

$$\epsilon_e = \left\|\boldsymbol{\psi}^t\boldsymbol{R}_\delta(\boldsymbol{u}_H^L)\right\|_e \tag{24}$$

Up to this point, the sensitivities denoted by $\boldsymbol{\psi}^t$ are unknown and they represent a set of weights that tell us how important the local approximation is with respect to the error in target functional. If $J(\boldsymbol{u})$ and $\boldsymbol{R}(\boldsymbol{u})$ were discretised operators, we could rewrite equation (19), and derive the discrete approximation for $\boldsymbol{\psi}$. However, in the continuous approach followed here, the sensitivities are obtained by deriving the continuous adjoint equations defining $\boldsymbol{\psi}$ and employing a consistent discretisation to obtain their solution. The advantage of this continuous approach is that the adjoint equations are independent of the form of the discretised flow equations [15]. They are also similar to the governing equations and are discretised using the same numerical methods utilised to solve the flow equations. The next section describes the derivation of the continuous adjoint equations. Their solution provides us with the sensitivities $\boldsymbol{\psi}$ to construct the goal-based error indicator, ϵ_e.

4. Continuous adjoint formulation

Equations (18) and (19) show that the adjoint variables $\boldsymbol{\psi}^t$ link the gradient of the target functional to the local gradient of the variables. To derive the continuous adjoint equations to the compressible Navier-Stokes equations (1), our target functional is either of the lift or drag coefficients, against which we want to determine the flow sensitivity. We therefore consider external flow problems such as the simulation of compressible flow past an aerofoil wing section which is illustrated in figure 1. The aerodynamic coefficients of lift and drag for a wing section result from the integral over the surface of the pressure and the viscous shear stresses at the surface. In general, we can define our target functional as

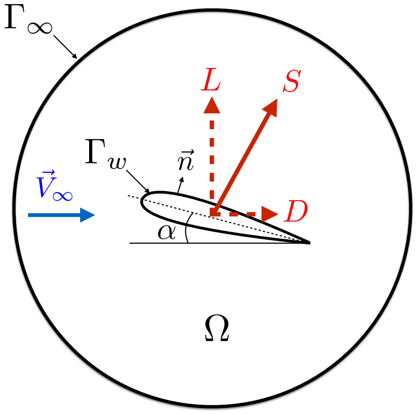


Figure 1: Sketch of the computational domain for the flow past an aerofoil.

$$J(\boldsymbol{u}, \nabla\boldsymbol{u}) = \int_{\Gamma_w} [p(\boldsymbol{u})\vec{n} - \boldsymbol{\tau}(\boldsymbol{u}, \nabla\boldsymbol{u})\vec{n}] \cdot \vec{\theta}\, d\Gamma \tag{25}$$

where Γ_w represents the wall boundary and $\vec{\theta}$ is the force projection vector. For instance, its value for the lift force is $\vec{\theta}_l = [-\sin\alpha, \cos\alpha]^t$ and for the drag force is $\vec{\theta}_d = [\cos\alpha, \sin\alpha]^t$. The pressure is denoted by $p(\boldsymbol{u})$, and $\boldsymbol{\tau}(\boldsymbol{u}, \nabla\boldsymbol{u})$ is the tensor of viscous stresses defined as

$$\boldsymbol{\tau}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \begin{bmatrix} \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \end{bmatrix} \tag{26}$$

where the expressions of the stress components are given in equation (4).

Now the constrained optimisation problem is to

$$\begin{aligned} &\text{minimize} && J(\boldsymbol{u}, \nabla \boldsymbol{u}) \\ &\text{subject to} && \boldsymbol{R}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \mathbf{0} \end{aligned} \tag{27}$$

Using the method of Lagrange multipliers, this is equivalent to the minimisation of the augmented problem

$$L\left(\boldsymbol{u}, \nabla \boldsymbol{u}\right) = J\left(\boldsymbol{u}, \nabla \boldsymbol{u}\right) - \int_{\Omega} \boldsymbol{\psi}^t \ \boldsymbol{R}\left(\boldsymbol{u}, \nabla \boldsymbol{u}\right) \ d\Omega \tag{28}$$

where the values of the multipliers $\boldsymbol{\psi} = [\psi_1, \psi_2, \psi_3, \psi_4]^t$ might be chosen freely without changing the value of the functional since $\boldsymbol{R}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \mathbf{0}$.

We now proceed to the derivation of the continuous adjoint equations and boundary conditions that will permit us to calculate the values of the multiplier, $\boldsymbol{\psi}$, and evaluate expression (24) to obtain our goal-based error indicator, ϵ_e. This requires the linearisation of the functional, L, which depends on the state variables, $\boldsymbol{u}$, and its first derivatives. In the following, we will refer to the first derivatives as $\boldsymbol{u}_{x_1} = \frac{\partial \boldsymbol{u}}{\partial x_1}$ and $\boldsymbol{u}_{x_2} = \frac{\partial \boldsymbol{u}}{\partial x_2}$. Here $\boldsymbol{u}_{x_1}$ and $\boldsymbol{u}_{x_2}$ are column vectors which have four entries. An extremum of this functional is a zero of its variational derivative [16]. This is expressed as

$$\begin{aligned} \delta L &= \left\{ \frac{\partial J}{\partial \boldsymbol{u}} \right\}^t \delta \boldsymbol{u} + \sum_{i=1}^{2} \left\{ \frac{\partial J}{\partial \boldsymbol{u}_{x_i}} \right\}^t \delta \boldsymbol{u}_{x_i} \\ &- \int_{\Omega} \boldsymbol{\psi}^t \left\{ \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} \right] \delta \boldsymbol{u} + \sum_{i=1}^{2} \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_{x_i}} \right] \delta \boldsymbol{u}_{x_i} \right\} d\Omega = 0 \end{aligned} \tag{29}$$

Substituting the expression (25) into equation (29), linearising the compressible

Navier-Stokes operator given in equation (1) and bringing the integral term to the right-hand side leads to

$$
\begin{aligned}
&\int_{\Gamma_w} \left\{ \vec{\theta}\cdot\vec{n}\left\{\frac{\partial p}{\partial \boldsymbol{u}}\right\}^t - \left\{\frac{\partial}{\partial \boldsymbol{u}}\left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right)\right\}^t\right\}\delta\boldsymbol{u}\; d\Gamma \\
&- \int_{\Gamma_w}\sum_{i=1}^{2} \left\{\frac{\partial}{\partial \boldsymbol{u}_{x_i}}\left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right)\right\}^t \delta\boldsymbol{u}_{x_i}\; d\Gamma \\
&= \int_{\Omega} \boldsymbol{\psi}^t\left\{\sum_{i=1}^{2}\frac{\partial}{\partial x_i}\left[\frac{\partial \boldsymbol{f}_i^c}{\partial\boldsymbol{u}} - \frac{\partial \boldsymbol{f}_i^v}{\partial\boldsymbol{u}}\right]\delta\boldsymbol{u} - \sum_{i,j=1}^{2} \frac{\partial}{\partial x_j}\left[\frac{\partial \boldsymbol{f}_i^v}{\partial \boldsymbol{u}_{x_j}}\right]\delta\boldsymbol{u}_{x_i}\right\} d\Omega
\end{aligned}
\tag{30}
$$

where the Jacobian of the Cartesian components of the convective and diffusive fluxes are given by $\frac{\partial \boldsymbol{f}_i^c}{\partial \boldsymbol{u}}$ and $\frac{\partial \boldsymbol{f}_i^v}{\partial \boldsymbol{u}}$. These Jacobians are 4×4 matrices and are given in Appendix A. The expressions of the 4×4 matrices $\frac{\partial \boldsymbol{f}_i^v}{\partial \boldsymbol{u}_{x_j}}; i, j = 1, 2,$ arising from the differentiation of the diffusive fluxes with respect to the first derivatives of $\boldsymbol{u}$ are also given in Appendix A.

To lift all the derivative terms onto the adjoint variable $\boldsymbol{\psi}$, we apply Gauss' theorem to the integral over the domain, Ω, on the right-hand side of equation (30) to obtain

$$
\begin{aligned}
&\int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{\left\{\frac{\partial p}{\partial \boldsymbol{u}}\right\}\vec{\theta}\cdot\vec{n} - \left\{\frac{\partial}{\partial \boldsymbol{u}}\left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right)\right\}\right\}\; d\Gamma \\
&- \int_{\Gamma_w}\sum_{i=1}^{2} \delta\boldsymbol{u}_{x_i}^t \left\{\frac{\partial}{\partial \boldsymbol{u}_{x_i}}\left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right)\right\}\; d\Gamma \\
&= \int_{\Omega} \delta\boldsymbol{u}^t\left\{-\sum_{i=1}^{2}\left[\frac{\partial \boldsymbol{f}_i^c}{\partial\boldsymbol{u}} - \frac{\partial \boldsymbol{f}_i^v}{\partial\boldsymbol{u}}\right]^t\frac{\partial \boldsymbol{\psi}}{\partial x_i} - \sum_{i,j=1}^{2}\left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t\frac{\partial^2 \boldsymbol{\psi}}{\partial x_i^2}\right\}\; d\Omega \\
&+ \int_{\Gamma} \delta\boldsymbol{u}^t\left\{\left[\frac{\partial \boldsymbol{f}_n^c}{\partial\boldsymbol{u}} - \frac{\partial \boldsymbol{f}_n^v}{\partial\boldsymbol{u}}\right]^t\boldsymbol{\psi} + \sum_{i,j=1}^{2}\left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t\frac{\partial \boldsymbol{\psi}}{\partial x_i}n_i\right\}\; d\Gamma \\
&- \int_{\Gamma}\sum_{i=1}^{2}\delta\boldsymbol{u}_{x_i}^t\left\{\left[\frac{\partial \boldsymbol{f}_n^v}{\partial \boldsymbol{u}_{x_i}}\right]^t\boldsymbol{\psi}\right\}\; d\Gamma
\end{aligned}
\tag{31}
$$

To derive the adjoint equations and appropriate boundary conditions, it is helpful to split the boundary, Γ, into two components: the wall boundary, Γ_w, and the rest, i.e. the far-field boundary for external flows, which is denoted by Γ_f. Equation (31) now reads

$$\begin{aligned}
&\int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{ \left\{ \frac{\partial p}{\partial \boldsymbol{u}} \right\} \vec{\theta}\cdot\vec{n} - \left\{ \frac{\partial}{\partial \boldsymbol{u}} \left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta} \right) \right\} \right\} \, d\Gamma \\
- &\int_{\Gamma_w} \sum_{i=1}^{2} \delta\boldsymbol{u}_{x_i}^t \left\{ \frac{\partial}{\partial \boldsymbol{u}_{x_i}} \left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right)\right\} \, d\Gamma \\
= &\int_\Omega \delta\boldsymbol{u}^t \left\{ -\sum_{i=1}^2 \left[\frac{\partial \boldsymbol{f}_i^c}{\partial\boldsymbol{u}} - \frac{\partial \boldsymbol{f}_i^v}{\partial\boldsymbol{u}}\right]^t \frac{\partial\boldsymbol{\psi}}{\partial x_i} - \sum_{i,j=1}^2 \left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \frac{\partial^2\boldsymbol{\psi}}{\partial x_i^2}\right\} \, d\Omega \\
+ &\int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^c}{\partial\boldsymbol{u}} - \frac{\partial \boldsymbol{f}_n^v}{\partial\boldsymbol{u}}\right]^t \boldsymbol{\psi} + \sum_{i,j=1}^2 \left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \frac{\partial\boldsymbol{\psi}}{\partial x_i} n_i \right\} \, d\Gamma \\
- &\int_{\Gamma_w} \sum_{i=1}^{2} \delta\boldsymbol{u}_{x_i}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \boldsymbol{\psi} \right\} \, d\Gamma \\
+ &\int_{\Gamma_f} \delta\boldsymbol{u}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^c}{\partial\boldsymbol{u}} - \frac{\partial \boldsymbol{f}_n^v}{\partial\boldsymbol{u}}\right]^t \boldsymbol{\psi} + \sum_{i,j=1}^2 \left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \frac{\partial\boldsymbol{\psi}}{\partial x_i} n_i \right\} \, d\Gamma \\
- &\int_{\Gamma_f} \sum_{i=1}^{2} \delta\boldsymbol{u}_{x_i}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \boldsymbol{\psi} \right\} \, d\Gamma
\end{aligned} \tag{32}$$

To derive the adjoint equations for compressible inviscid problems, we use the same procedure as presented for the compressible Navier-Stokes equations. However, in this case the viscous stresses are neglected and the target functional simplifies to

$$J(\boldsymbol{u}) = \int_{\Gamma_w} p\vec{n}\cdot\vec{\theta}\, d\Gamma \tag{33}$$

Since the viscous flux terms are neglected equation (32) simplifies to

$$\begin{aligned}
\int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{ \left\{ \frac{\partial p}{\partial \boldsymbol{u}} \right\} \vec{\theta}\cdot\vec{n} \right\} \, d\Gamma \;\; = \;\; &\int_\Omega \delta\boldsymbol{u}^t \left\{ -\sum_{i=1}^2 \left[\frac{\partial \boldsymbol{f}_i^c}{\partial\boldsymbol{u}} \right]^t \frac{\partial\boldsymbol{\psi}}{\partial x_i} \right\} \, d\Omega \\
+ &\int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^c}{\partial\boldsymbol{u}} \right]^t \boldsymbol{\psi} \right\} \, d\Gamma \\
+ &\int_{\Gamma_f} \delta\boldsymbol{u}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^c}{\partial\boldsymbol{u}} \right]^t \boldsymbol{\psi} \right\} \, d\Gamma
\end{aligned} \tag{34}$$

Depending on the set of governing equations that are considered, we can look at equation (32) or (34) and identify what terms we need to cancel to satisfy $\delta L = 0$ given that the perturbations $\delta \boldsymbol{u}$, $\delta \boldsymbol{u}_{x_1}$ and $\delta \boldsymbol{u}_{x_2}$ are arbitrary.

4.1. Adjoint equation

Requiring that the integrand in the integral over the domain, Ω, in equation (32) vanishes we obtain the *adjoint equation* to the compressible Navier-Stokes equations for $\boldsymbol{\psi}$ as

$$\hat{\boldsymbol{R}}(\boldsymbol{\psi}) = -\sum_{i=1}^{2}\left[\frac{\partial \boldsymbol{f}_i^c}{\partial \boldsymbol{u}} - \frac{\partial \boldsymbol{f}_i^v}{\partial \boldsymbol{u}}\right]^t \frac{\partial \boldsymbol{\psi}}{\partial x_i} - \sum_{i,j=1}^{2}\left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \frac{\partial^2 \boldsymbol{\psi}}{\partial x_i^2} = \mathbf{0} \tag{35}$$

For inviscid flows we neglect the viscous terms to obtain the adjoint compressible Euler equations,

$$\hat{\boldsymbol{R}}(\boldsymbol{\psi}) = -\sum_{i=1}^{2}\left[\frac{\partial \boldsymbol{f}_i^c}{\partial \boldsymbol{u}}\right]^t \frac{\partial \boldsymbol{\psi}}{\partial x_i} = \mathbf{0} \tag{36}$$

We solve for $\boldsymbol{\psi}$ using a similar discretisation to that used for the governing flow equations and obtain the sensitivity of the target functional with respect to the state variables.

4.2. Adjoint boundary conditions

Focusing now on the integrals over the wall to cancel the integrands, multiplying the terms $\delta \boldsymbol{u}_{x_i}; i = 1, 2$ in equation (32), the values at the wall should be such that

$$\left[\frac{\partial \boldsymbol{f}_n^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \boldsymbol{\psi} = \frac{\partial}{\partial \boldsymbol{u}_{x_i}}\left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right) \; ; \quad i = 1, 2 \tag{37}$$

Further, the cancellation of the terms corresponding to the perturbation $\delta \boldsymbol{u}$ leads to

$$\left\{\frac{\partial p}{\partial \boldsymbol{u}}\right\}\vec{\theta}\cdot\vec{n} - \left\{\frac{\partial}{\partial \boldsymbol{u}}\left((\boldsymbol{\tau}\vec{n})\cdot\vec{\theta}\right)\right\} = \left[\frac{\partial \boldsymbol{f}_n^c}{\partial \boldsymbol{u}} - \frac{\partial \boldsymbol{f}_n^v}{\partial \boldsymbol{u}}\right]^t \boldsymbol{\psi} + \sum_{i,j=1}^{2}\left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}}\right]^t \frac{\partial \boldsymbol{\psi}}{\partial x_i} n_i \tag{38}$$

For the compressible viscous flow problems considered in this article, we are dealing with an isothermal wall with a no-slip condition and thus the viscous flux normal to the wall can be written as

$$\boldsymbol{f}_n^v|_w = \left\{ \begin{array}{c} 0 \\ \boldsymbol{\tau}\vec{n} \\ k\left(\sum_{i=1}^{2} \dfrac{\partial T}{\partial x_i} n_i\right) \end{array} \right\}_w \tag{39}$$

We can linearise the viscous flux normal to the wall with respect to the first order derivatives of $\boldsymbol{u}$, given in equation (39), and substitute the result into equation (37) and we obtain the boundary conditions for the adjoint problem at the wall, which are given by

$$\psi_2 = \theta_1\,; \qquad \psi_3 = \theta_2\,; \qquad \psi_4 = 0 \tag{40}$$

Since the variable ψ_1 is not prescribed at the wall, we are considering a "do nothing" boundary condition for ψ_1. To finalise the derivation of the boundary conditions for the adjoint equations for compressible laminar flow, we deal now with the integrals over the far-field boundary Γ_f. The integrand multiplying the perturbation $\delta\boldsymbol{u}_{x_i}$ could be set to zero by imposing that $\boldsymbol{\psi} = \boldsymbol{0}$. This, together with the condition that $\sum_{i=1}^{2} \boldsymbol{\psi}_{x_i} n_i = \boldsymbol{0}$ at the far field, leads to the cancelation of the term for the perturbation $\delta\boldsymbol{u}$.

Considering compressible inviscid flow, we require the integrands multiplying the integrals over Γ_w in equation (34) need to cancel the integrands multiplying the terms $\delta\boldsymbol{u}$. Therefore we write

$$\int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{ \left\{ \frac{\partial p}{\partial \boldsymbol{u}} \right\} \vec{\theta}\cdot\vec{n} \right\}\, d\Gamma = \int_{\Gamma_w} \delta\boldsymbol{u}^t \left\{ \left[\frac{\partial \boldsymbol{f}_n^c}{\partial \boldsymbol{u}} \right]^t \boldsymbol{\psi} \right\}\, d\Gamma \tag{41}$$

The convective normal flux at the wall can be written as

$$\boldsymbol{f}^c_n|_w = \left\{ \begin{array}{c} 0 \\ p\vec{n} \\ 0 \end{array} \right\}_w \tag{42}$$

We linearise the convective flux normal to the wall with respect to $\boldsymbol{u}$ and substitute this in equation (41). We obtain the wall boundary condition for the adjoint problem which for a slip boundary is given by

$$\psi_2 n_1 + \psi_3 n_2 = (\theta_1 n_1 + \theta_2 n_2) \, . \tag{43}$$

Finally, for the compressible Navier-Stokes equations, the integrands at the far field multiplying $\delta\boldsymbol{u}$ should vanish which is achieved by setting $\boldsymbol{\psi} = \boldsymbol{0}$ at the far-field.

5. High-order discontinuous Galerkin discretisation

The compressible Navier-Stokes equations (1) are discretised using the high-order spectral/hp element framework `Nektar++` [17], which supports unstructured two- and three-dimensional simulations. To begin the discretisation, we subdivide the computational domain Ω into N_{el} non-overlapping elements, so that $\Omega = \bigcup_{e=1}^{N_{el}} \Omega^e$, where $\bigcap_{e=1}^{N_{el}} \Omega^e = \emptyset$. For convenience, the local element coordinates, defined as $(x_1, x_2) \in \Omega^e$, are mapped onto standard element coordinates $(\xi_1, \xi_2) \in \Omega^{st}$, defined as

$$\Omega^{st} = \{(\xi_1, \xi_2) \mid -1 \leq \xi_1, \xi_2 \leq 1\},$$

through a bijective mapping $M_e : (\xi_1, \xi_2) \mapsto (x_1, x_2)$, as illustrated in figure 2. In this way, we can express the approximate solution $\boldsymbol{u}_\delta$ in terms of a set of local expansion modes that have beneficial orthogonality properties within the standard element domain. We use a discontinuous Galerkin approach where the solution is allowed to be discontinuous between the elements. Denoting the

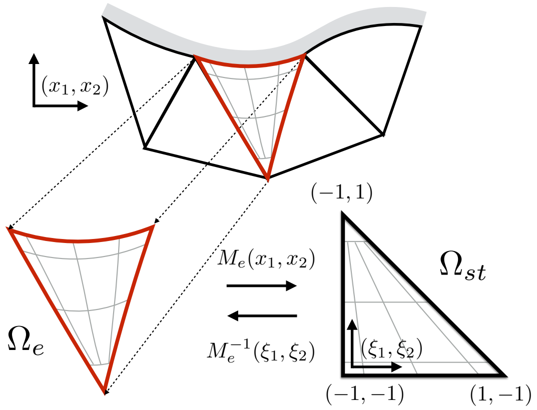


Figure 2: The mapping that relates the local coordinates (x_1, x_2) to the reference coordinates (ξ_1, ξ_2).

space of all polynomials of degree P defined on the standard element Ω^{st} by $\mathcal{P}_P(\Omega^{st})$, we seek an approximate solution using a set of polynomials defined in a standard space Ω^{st} where all mathematical operations will be performed.

The chosen modal expansion basis is constructed from a family of orthogonal Jacobi polynomials that are augmented with linear basis functions, as outlined in [18]. On each element, this allows us to achieve a natural separation of boundary modes, which have support on faces and edges of elements, and interior modes, which are zero on the element boundary. This partitioning makes this choice of basis well-suited for the construction of a scheme that is variable in polynomial order, since no interpolation is required to add contributions arising through the edges of an element. This will be discussed in more detail in section 5.3. A notable point, which can be seen in figure 2, is that in the spectral/hp formulation, every element type (including triangles and tetrahedra) uses modal functions that are defined on a collapsed coordinate space $[-1,1]^d$ for $d = 2, 3$. This allows us to exploit a tensor product expansion of the basis functions, taking the form $\phi_{pq}(\vec{\xi}) = \phi_p^a(\vec{\xi})\phi_{pq}^b(\vec{\xi})$ in two dimensions. It also leads to a set of collapsed coordinate quadrature points, as opposed to a nodal spacing of points within the element as is seen in other formulations.

Finally, the approximate solution within an element, $\boldsymbol{u}_e^\delta$, is written as an expansion in terms of the modal functions. For a quadrilateral element, this takes the form

$$\boldsymbol{u}_e^\delta(M^{-1}(x_1,x_2)) = \sum_{p=0}^{P}\sum_{q=0}^{Q} \phi_{pq}(\xi_1,\xi_2)\bar{u}_{pq} \tag{44}$$

where the array $\bar{u}_{pq}$ denotes the degrees of freedom that are solved for.

5.1. Discontinuous Galerkin discretisation of the governing equations

Using the mixed formulation for a DG discretisation of the compressible Navier-Stokes equations proposed in reference [19], we first determine the first derivatives using auxiliary variables $\boldsymbol{g} = \nabla \boldsymbol{u} = \left[\frac{\partial \boldsymbol{u}}{\partial x_1}, \frac{\partial \boldsymbol{u}}{\partial x_2}\right]$. A mixed formulation is then given by

$$\begin{aligned} \boldsymbol{g} - \nabla \boldsymbol{u} &= \boldsymbol{0} \\ \sum_{i=1}^{2} \frac{\partial}{\partial x_i}\left\{\boldsymbol{f}_i^c(\boldsymbol{u}) - \boldsymbol{f}_i^v(\boldsymbol{u},\boldsymbol{g})\right\} &= \boldsymbol{0} \end{aligned} \tag{45}$$

where the first equation of the coupled system (45) is discretised so that we obtain a solution for $\boldsymbol{g}_e^\delta$ using

$$\sum_{e=1}^{N_{el}}\left\{\int_{\Omega_e} \phi^e \boldsymbol{g}_e^\delta \, d\Omega - \int_{\Gamma_e} \phi^e \boldsymbol{u}_e^\delta \vec{n}\, d\Gamma + \int_{\Omega_e} \nabla\phi^e \boldsymbol{u}_e^\delta \, d\Omega\right\} = \boldsymbol{0} \tag{46}$$

where $\boldsymbol{g}_e^\delta$ and $\boldsymbol{u}_e^\delta$ are both approximated using modal expansions of the form (44). Once a solution is found for $\boldsymbol{g}_e^\delta$, we then substitute it in the discretised form of the second equation of (45) which is given by

$$\begin{aligned} - & \sum_{e=1}^{N_{el}}\int_{\Omega_e}\sum_{i=1}^{2}\frac{\partial\phi^e}{\partial x_i}\left\{\boldsymbol{f}_i^c(\boldsymbol{u}_e^\delta) - \boldsymbol{f}_i^v(\boldsymbol{u}_e^\delta,\boldsymbol{g}_e^\delta)\right\} d\Omega \\ + & \sum_{e=1}^{N_{el}}\int_{\Gamma_e} \phi^e \boldsymbol{f}_n^c(\boldsymbol{u}_e^\delta)\; d\Gamma - \sum_{e=1}^{N_{el}}\int_{\Gamma_e} \boldsymbol{f}_n^v(\boldsymbol{u}_e^\delta,\boldsymbol{g}_e^\delta)\; d\Gamma = \boldsymbol{0} \end{aligned} \tag{47}$$

where the convective and diffusive fluxes normal to the boundary are

$$\boldsymbol{f}^c_n(\boldsymbol{u}^\delta_e) = \sum_{i=1}^{2} \boldsymbol{f}^c_i(\boldsymbol{u}^\delta_e) n_i, \qquad \boldsymbol{f}^v_n(\boldsymbol{u}^\delta_e, \boldsymbol{g}^\delta_e) = \sum_{i=1}^{2} \boldsymbol{f}^v_i(\boldsymbol{u}^\delta_e, \boldsymbol{q}^\delta_e) n_i \tag{48}$$

In a discontinuous Galerkin discretisation, the values are discontinuous at the element interfaces and thus the integrals over the interfaces are not uniquely defined. Therefore, the fluxes appearing in the second term of the left-hand side of equation (46) and the second and third terms of the left-hand side of equation (47) are replaced by numerical fluxes calculated through the solution of a one-dimensional Riemann problem in the normal direction to the interface Γ_e. Hence, we replace the second term on the left-hand side of equation (47) with the numerical flux function, $\mathcal{H}^c$, so that

$$\boldsymbol{f}^c_n(\boldsymbol{u}^\delta_e) \approx \mathcal{H}^c(\boldsymbol{u}^\delta_{ex}, \boldsymbol{u}^\delta_{in}; \vec{n}) \tag{49}$$

where $\boldsymbol{u}_{ex}$ and $\boldsymbol{u}_{in}$ denote the values of the external and internal variables at the interface with respect to the e^{th} element, and $\vec{n}$ is the outer normal at the interface.

A set of different Riemann solvers, including the approximate Roe Riemann solver, the Harten-Lax-van Leer contact solver and the exact Riemann solver [20], are implemented in `Nektar++` to evaluate the fluxes at the interface. For the viscous flux terms, we have selected the local discontinuous Galerkin (LDG) flux amongst the different flux functions for the diffusion operator discussed and compared in [18, 21]. Therefore we use the external state variables in equation (46) to calculate the flux at the boundary and write

$$\boldsymbol{u}^\delta_e \vec{n}\big|_{\Gamma_e} = \boldsymbol{u}^\delta_{ex} \vec{n} \tag{50}$$

Further, we replace the term $\boldsymbol{f}^v_n(\boldsymbol{u}^\delta_e, \boldsymbol{g}^\delta_e)$ with the internal first derivatives as

$$\boldsymbol{f}^v_n(\boldsymbol{u}^\delta_e, \boldsymbol{g}^\delta_e)\big|_{\Gamma_e} = \sum_{i=1}^{2} \boldsymbol{g}_{i,in} n_i \tag{51}$$

5.2. Discontinuous Galerkin discretisation of the adjoint equations

For the adjoint equations, we again apply the approach described by Bassi and Rebay [19] by introducing the auxiliary term $\boldsymbol{g} = \nabla\boldsymbol{\psi} = \left[\frac{\partial\boldsymbol{\psi}}{\partial x_1}, \frac{\partial\boldsymbol{\psi}}{\partial x_2}\right]$ that represents the first derivatives of the adjoint variables. We compose a coupled system as follows

$$\begin{aligned} \boldsymbol{g} - \nabla\boldsymbol{\psi} &= \mathbf{0} \\ -\sum_{i=1}^{2}\left[\frac{\partial\boldsymbol{f}_i^c}{\partial\boldsymbol{u}} - \frac{\partial\boldsymbol{f}_i^v}{\partial\boldsymbol{u}}\right]^t \frac{\partial\boldsymbol{\psi}}{\partial x_i} - \sum_{i,j=1}^{2}\left[\frac{\partial\boldsymbol{f}_j^v}{\partial\boldsymbol{u}_{x_i}}\right]^t \frac{\partial\boldsymbol{g}_i}{\partial x_i} &= \mathbf{0} \end{aligned} \tag{52}$$

Using a similar procedure to that used to get equation (46), we obtain the discrete form to solve for the first order derivatives of the adjoint variables

$$\sum_{e=1}^{N_{el}} \left\{\int_{\Omega_e} \phi^e \boldsymbol{g}_e^\delta \, d\Omega - \int_{\Gamma_e} \phi^e \boldsymbol{\psi}_e^\delta \vec{n} \, d\Gamma + \int_{\Omega_e} \nabla\phi^e \boldsymbol{\psi}_e^\delta \, d\Omega\right\} = \mathbf{0} \tag{53}$$

where $\boldsymbol{g}_e^\delta$ and $\boldsymbol{\psi}_e^\delta$ are defined as in equation (44). Once a solution is found for $\boldsymbol{g}_e^\delta$, we substitute this in the discretised form of the second equation of the coupled system (52) which is now written as

$$\begin{aligned} &\sum_{e=1}^{N_{el}} \left\{\int_{\Omega_e} \sum_{i=1}^{2} \frac{\partial\phi^e}{\partial x_i} \left\{\left[\frac{\partial\boldsymbol{f}_i^c}{\partial\boldsymbol{u}} - \frac{\partial\boldsymbol{f}_i^v}{\partial\boldsymbol{u}}\right]^t \boldsymbol{\psi}_e^\delta\right\} d\Omega - \oint_{\Gamma_e} \phi^e \left\{\left[\frac{\partial\boldsymbol{f}_n^c}{\partial\boldsymbol{u}} - \frac{\partial\boldsymbol{f}_n^v}{\partial\boldsymbol{u}}\right]^t \boldsymbol{\psi}_e^\delta\right\} d\Gamma\right\} \\ &\quad + \sum_{e=1}^{N_{el}} \left\{\int_{\Omega} \sum_{i,j=1}^{2} \frac{\partial\phi^e}{\partial x_i} \left[\frac{\partial\boldsymbol{f}_j^v}{\partial\boldsymbol{u}_{x_i}}\right]^t \boldsymbol{g}_{i,e}^\delta \, d\Omega - \int_{\Gamma_e} \phi^e \sum_{i,j=1}^{2} \left[\frac{\partial\boldsymbol{f}_j^v}{\partial\boldsymbol{u}_{x_i}}\right]^t \boldsymbol{g}_{i,e}^\delta n_i \, d\Gamma\right\} = \mathbf{0} \end{aligned} \tag{54}$$

We also substitute the term on the boundary, $\left[\frac{\partial\boldsymbol{f}_n^c}{\partial\boldsymbol{u}}\right]^t \boldsymbol{\psi}_e^\delta$, by a numerical flux function. To guarantee consistency, we derive an adjoint Roe-averaged numerical flux function which is used in the numerical discretisation of the adjoint equations. Similarly to equation (49) we have

$$\left[\frac{\partial\boldsymbol{f}_n^c}{\partial\boldsymbol{u}}\right]^t \boldsymbol{\psi}_e^\delta\bigg|_{\Gamma_e} \approx \hat{\mathcal{H}}(\boldsymbol{\psi}_{ex}^\delta, \boldsymbol{\psi}_{in}^\delta; \vec{n}) \tag{55}$$

where $\hat{\mathcal{H}}$ denotes the adjoint Roe-averaged numerical flux. The derivation of $\hat{\mathcal{H}}$ is given in Appendix B.

For the flux terms that follow from integrating the viscous terms by parts, we use again the LDG method in order to be consistent with the discretisation of the governing equations. The term $\sum_{i=1}^{2} \boldsymbol{\psi}_{e_i}^{\delta} n_i$ replaced by a numerical flux in the same way as given in equation (50). Using the LDG approach, this term is similarly determined as in equation (50), by taking the adjoint of the external state and writing it as

$$\boldsymbol{\psi}_e^{\delta} \vec{n} \Big|_{\Gamma_e} = \boldsymbol{\psi}_{ex}^{\delta} \vec{n} \tag{56}$$

Furthermore, we substitute the last integral term on the left-hand side of equation (47) to obtain

$$\sum_{i,j=1}^{2} \left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}} \right]^t \boldsymbol{g}_{e_i}^{\delta} n_i \Big|_{\Gamma_e} = \sum_{i,j=1}^{2} \left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}} \right]^t \frac{\partial \boldsymbol{\psi}_{in_i}^{\delta}}{\partial x_i} n_i \tag{57}$$

5.3. Variable polynomial order

To accommodate a variable polynomial order distribution between elements, we need to appropriately evaluate the fluxes at the elemental interfaces, since they appear twice using a different polynomial approximation. One needs to ensure that the fluxes are calculated using the highest polynomial order of any two adjacent elements, as illustrated in figure 3.

Using the lowest number of integration points may lead to numerical instabilities. This can be explained by the fact that the element with the higher order has unconstrained degrees of freedom [22]. To ensure conservation and stability, the continuity of the total flux is required between two adjacent elements and therefore

$$\int_{\Gamma_{f_r}} \boldsymbol{f}(\boldsymbol{u}_{ex}^{\delta}) d\Gamma = \int_{\Gamma_{f_l}} \boldsymbol{f}(\boldsymbol{u}_{in}^{\delta}) d\Gamma \tag{58}$$

where $\boldsymbol{f}(\boldsymbol{u}_{ex}^{\delta})$ and $\boldsymbol{f}(\boldsymbol{u}_{in}^{\delta})$ represent the numerical flux on the edge between two elements which are each approximated using different polynomial orders. Once it is determined that the order is different, the coefficients are copied directly

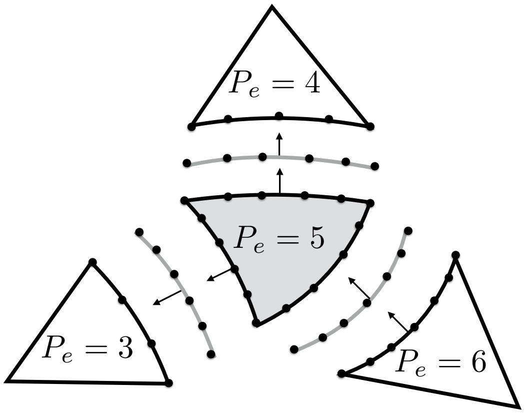


Figure 3: The higher order edge is used to calculated the advective fluxes and the result is used on the lower order adjacent edge.

onto the higher resolved side, but fewer coefficients have to be set on the other side. Since the modified basis defined in the previous section is not orthogonal, we instead project the higher order interface onto a space of orthogonal polynomials and filter it in this space, so as to remove the high-order frequencies. This can be done because the coefficients in orthogonal space are not coupled, so the high-order frequencies can be removed without changing the mean solution by setting the higher order coefficients equal to zero.

Once the degree of the orthogonal expansion is decreased, a reverse projection is carried out and the coefficients in the original modified basis are obtained. The newly calculated flux values are then used to determine the boundary integral for the lower order element. The projection procedure is described in the following where we omit the subscript e since all operations in the analysis take place within the element. The solution within an element as given in equation (44), can be expressed using a matrix formulation as

$$\boldsymbol{u}^{\delta} = B\bar{\boldsymbol{u}} \tag{59}$$

where B is the matrix of the modified expansion basis and the vector $\bar{\boldsymbol{u}}$ represents the corresponding modified coefficients. The solution can also be expressed using an orthogonal basis with coefficient vector $\bar{\boldsymbol{u}}_o$ as

$$\boldsymbol{u}^\delta = B_o\bar{\boldsymbol{u}}_o = B\bar{\boldsymbol{u}} \tag{60}$$

and the array of orthogonal coefficients is calculated from

$$\bar{\boldsymbol{u}}_o = B_o^{-1}B\bar{\boldsymbol{u}} \tag{61}$$

where B_o represent the matrix of the orthogonal expansion basis. The filtered coefficients that belong to the modified basis are obtained using

$$\bar{\boldsymbol{u}}_f = B^{-1}B_o\bar{\boldsymbol{u}}_{o,f} \tag{62}$$

where $\bar{\boldsymbol{u}}_f$ represents the filtered coefficient corresponding to the modified basis functions. In this way, the information contained in the high frequency components is removed without altering the mean value and the boundary integral for the lower order element can be obtained using $\bar{\boldsymbol{u}}_f$.

6. Numerical examples

This section aims to illustrate the performance and cost-effectiveness of the goal-based p-adaptive method when applied to external compressible flow cases, by recording and comparing both the number of degrees of freedom and the CPU time per time step. For each case under consideration, we generate a mesh that remains fixed across all tests and vary the polynomial order within the elements of that mesh, but ensure that the resulting high-order mesh conforms to the boundary of the computational domain.

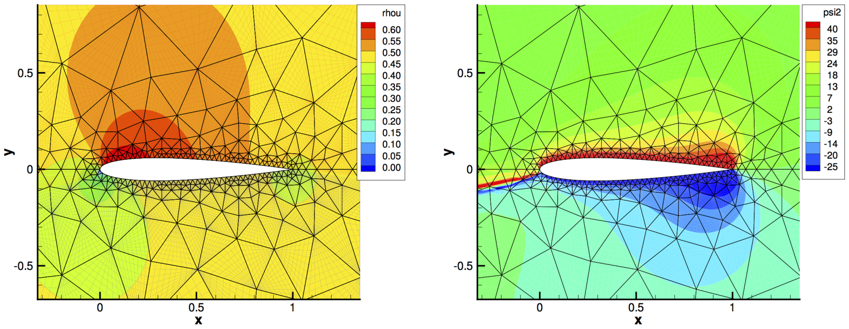
We begin by examining a well-known test case of inviscid and laminar subsonic flow over a NACA0012 aerofoil. Transonic flow regimes are then investigated over the same geometry to examine the behaviour of the method in the presence of shocks. Finally, we consider a subsonic inviscid flow over an ellipsoid, in order to demonstrate the application of the method to three-dimensional geometries discretised using high-order unstructured meshes.

6.1. Subsonic inviscid flow past a NACA0012 wing section ($Ma = 0.4$, $\alpha = 5°$).

We consider the application of p-adaptation to obtain improved estimates of the lift coefficient, c_l, for a NACA0012 profile in subsonic inviscid flow with a free-stream Mach number $Ma = 0.4$ at an incidence $\alpha = 5°$. The flow solution under these conditions is smooth and we expect an exponential decay of the numerical error as the polynomial order, P, is increased. Therefore this case is well suited to illustrate the ability of p-adaptation to increase numerical resolution whilst keeping CPU time costs low.

Considering compressible inviscid flow, we solve the adjoint equations given in equation (36) with the adjoint boundary conditions at the wall presented in equation (43). An initial steady-state solution is obtained using a unstructured mesh of 874 triangles with a uniform polynomial order $P = 3$. We assume that a steady-state solution has been reached when the lowest of the values of the L_2 norm of the residuals of the conservative variables is smaller than 10^{-8}. Another steady-state solution is then obtained in the same mesh using a constant polynomial order $P = 5$. The residuals are evaluated by interpolating the solution for $P = 3$ onto the solution for $P = 5$ and subtracting the interpolated solution from the higher order solution. The steady-state adjoint solution is obtained using the same mesh with $P = 5$ in order to determine the importance of the local residual with respect to the lift coefficient. Here we apply the same criterion of convergence towards steady state used for the direct problem. The solutions to the governing and adjoint equations are depicted in figure 4.

Figure 4a shows the x-momentum contours and figure 4b shows the contours of the corresponding adjoint solution ψ_2. The adjoint solution presented in figure 4b can be interpreted as follows: a positive perturbation in x-momentum (for example) results in an increase in lift where the ψ_2 component is positive. Conversely, it decreases the lift where the corresponding adjoint component is negative. Using the adjoint solution, we evaluate the element-wise error indicator in equation (24) to obtain a new polynomial distribution which is shown in



(a) Solution for x-momentum (ρu).

(b) Adjoint variable ψ_2 .

Figure 4: The solution to the governing (4a) and adjoint (4b) equations for inviscid flow past a NACA0012 aerofoil ($Ma = 0.4$, $\alpha = 5°$).

figure 5b. Here, the reference lift coefficient c_l^{ref} is computed from the solution obtained using a constant $P = 9$ everywhere and we define the error in lift coefficient as $\epsilon_{c_l} = \|c_l^{\text{ref}} - c_l\|$, where c_l is the value for each of our simulations.

We analyse the effect of different polynomial order distributions, both constant in space and using our p-adaptive method and, to assess the performance of each simulation, we must define metrics for comparison against the solution using constant polynomial order $P = 9$. As a measure of the computational cost, we use CPU time per time step normalised by the CPU time per time step for the reference case, which we denote by t_{CPU}. Although the solver is designed to run in parallel, in this instance we determine the value of t_{CPU} by running the simulations on a single CPU, in order to avoid potential load balancing issues and give a fair assessment of the performance of each simulation. Additionally, we note that due to the exploratory nature of this work, further computational improvements may be achievable through optimisation of the underlying code.

To determine the resolution requirements needed to attain a given error against the reference value, we consider the number of quadrature points across the mesh, which we denote by N_Q and is defined as

$$N_Q = \sum_{e=1}^{N_{el}} N_{p,e}^{\xi_1} N_{p,e}^{\xi_2} \tag{63}$$

where N_{el} is the number of elements, and $N_p^{\xi_1}$ and $N_p^{\xi_2}$ represent the number of integration points in the ξ_1 and ξ_2 coordinate directions respectively. The value N_Q has been chosen because it is representative of the number of degrees of freedom in the discrete solution and it is also very easy to calculate.

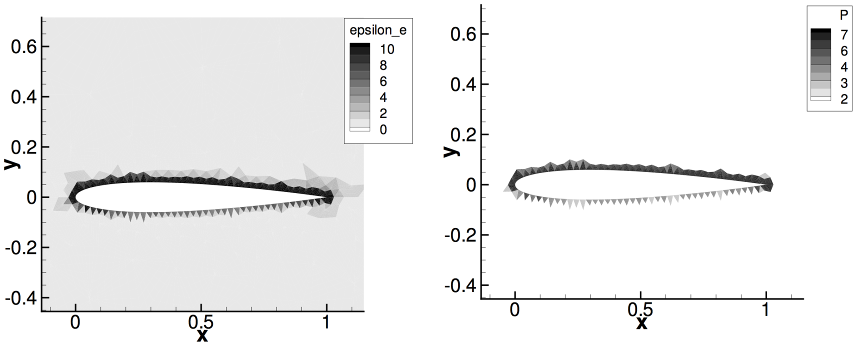
Table 1 compares the error in lift coefficient obtained using uniform polynomial refinement, i.e. a sequence of solutions with increasing constant polynomial order, and goal-based p-adaptation. Table 1 indicates that for the adaptive simulation with $3 \leq P \leq 9$ we obtain an error between the $P = 5$ and $P = 7$ simulations, but with a reduction of 35% and 62% fewer degrees of freedom respectively. The CPU time per time step is reduced by 46% for the $3 \leq P \leq 9$ case compared to the reference case while achieving a similar level of accuracy.

Uniform *P*	$P = 2$	$P = 3$	$P = 5$	$P = 7$	$P = 9$
N_Q	10488	17480	36708	62928	96140
t_{CPU}	0.25	0.27	0.48	0.70	1
c_l	0.6447	0.6574	0.659605	0.65972	0.65972
ϵ_{c_l}	1.5×10^{-2}	2.3×10^{-3}	6×10^{-5}	2.0×10^{-6}	–
Variable *P*	$2 \leq P \leq 4$	$3 \leq P \leq 6$	$3 \leq P \leq 9$		
N_Q	17844	21422	23892		
t_{CPU}	0.35	0.40	0.54		
c_l	0.659584	0.65969	0.65971		
ϵ_{c_l}	5.00×10^{-3}	8.25×10^{-4}	4.0×10^{-6}		

Table 1: Comparison of the error in lift coefficient for uniform polynomial refinement and goal-based p-adaptation for subsonic inviscid flow. The error is calculated with respect to the solution obtained with constant polynomial order $P = 9$.

6.2. Subsonic laminar flow past a NACA0012 aerofoil ($Ma = 0.1$, $Re = 5\,000$, $\alpha = 2°$).

We turn our attention now towards obtaining improved estimates of drag. Our target functional is the drag coefficient, c_d, evaluated over the surface of a



(a) The goal-based error indicator ϵ_e.

(b) The distribution of polynomial orders derived from ϵ_e.

Figure 5: Goal-based p-adaptation for inviscid flow past a NACA 0012 aerofoil ($Ma = 0.4$, $\alpha = 5°$).

NACA0012 aerofoil. The flow conditions correspond to subsonic laminar flow, with free-stream values $Ma = 0.1$ and $Re = 5\,000$, at an incidence $\alpha = 2°$.

We adopt a similar strategy to that used for the previous test case. The reference steady-state solution for this case is obtained using a constant polynomial order $P = 9$. To illustrate that the method is applicable to different mesh types, we use a structured mesh of 3012 quadrilateral elements here.

Figures 6a and 6b show the steady-state solutions to the direct and adjoint problems, respectively. The adjoint solution (figure 6b) shows a high sensitivity in the boundary layer which, as expected, indicates that viscous drag is the dominant factor determining the sensitivity of c_d in this case. Again we define our error as $\epsilon_{c_d} = \|c_d^{\text{ref}} - c_d\|$ against the reference drag coefficient c_d^{ref} and perform a series of simulations at both constant and variable polynomial order.

Figure 7 shows the relationship between the resolution requirements and the obtained error for the constant and variable polynomial order simulations. This figure shows the advantage of the variable polynomial order and the use of the goal-based error estimator, as the error in drag coefficient decreases significantly faster for the goal-based p-adaptation strategy compared to increasing the polynomial order uniformly across the domain. For example, starting with a low-order ($P = 3$) steady-state solution, the error in drag coefficient can be decreased by one order of magnitude by increasing the N_Q by a factor of 1.4 using adjoint-based p-adaptation. However, for uniform polynomial refinement, to obtain the same error reduction requires an increase of N_Q by a factor of 3.

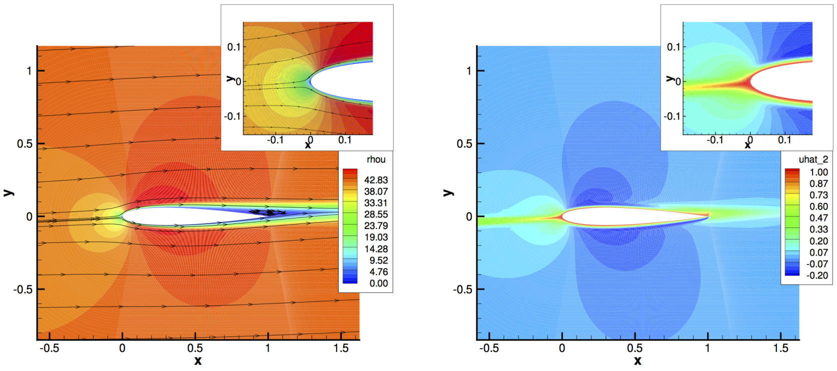
The CPU time per time step is reduced as well using a goal-based p-adaptive strategy. Table 2 confirms this, by showing the data of figure 7 alongside the recorded CPU time. For example the CPU time per time step is reduced by 54% for the $3 \leq P \leq 9$ case compared to the reference case while a similar level of accuracy is achieved as the reference case. The results show a broad increase in performance against the inviscid case considered in the previous section, due to both the sensitivity of c_d in a more localised region near the boundary layer, and the additional computational cost of evaluating the viscous tensor terms for constant polynomial order simulations. However, the results give a clear indication of the increase in performance: the $3 \leq P \leq 8$ simulation exhibits the same error as the solution obtained using constant $P = 7$ but at half of the computational cost.

Constant *P*	$P = 3$	$P = 5$	$P = 7$	$P = 9$	
N_Q	75300	147588	243972	364452	
t_{CPU}	0.28	0.29	0.64	1	
c_d	0.01507	0.01616	0.01629	0.01632	
ϵ_{c_d}	1.20×10^{-3}	1.57×10^{-4}	2.69×10^{-5}	$-$	
Variable *P*	$3 \leq P \leq 5$	$3 \leq P \leq 6$	$3 \leq P \leq 7$	$3 \leq P \leq 8$	$3 \leq P \leq 9$
N_Q	78183	82286	88165	96013	113909
t_{CPU}	0.31	0.32	0.34	0.36	0.45
c_d	0.01599	0.01624	0.01628	0.01629	0.01631
ϵ_{c_d}	3.19×10^{-4}	7.55×10^{-5}	3.47×10^{-5}	2.71×10^{-5}	5.63×10^{-6}

Table 2: Comparison of the error in drag coefficient for uniform polynomial refinement and goal-based p-adaptation for subsonic laminar flow. The error is calculated with respect to the solution obtained using constant $P = 9$.

6.3. Transonic inviscid flow past a NACA 0012 *($Ma = 0.8$, $\alpha = 1.25°$).*

This case corresponds to a transonic flow, with $Ma = 0.8$ past a NACA 0012 aerofoil at at an incidence $\alpha = 1.25°$. Salient features of this flow are a strong



(a) The x-momentum solution (ρu). **(b)** The corresponding adjoint variable ψ_2.

Figure 6: The solution to the governing and adjoint equations for compressible laminar flow past a NACA0012 aerofoil ($Ma = 0.1$, $Re = 5\,000$, $\alpha = 2°$).

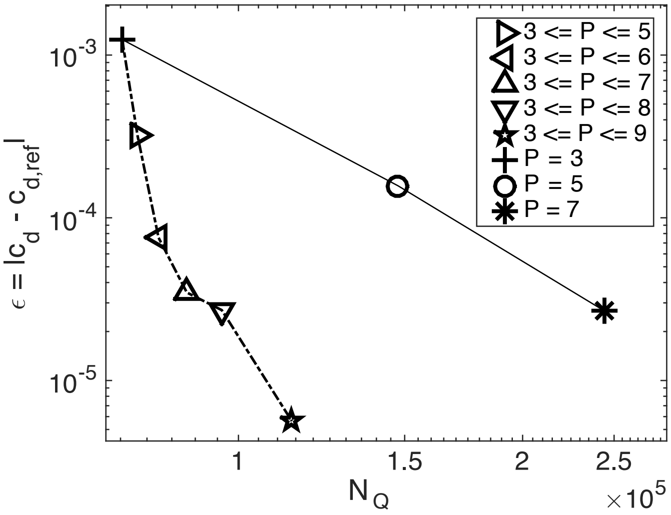


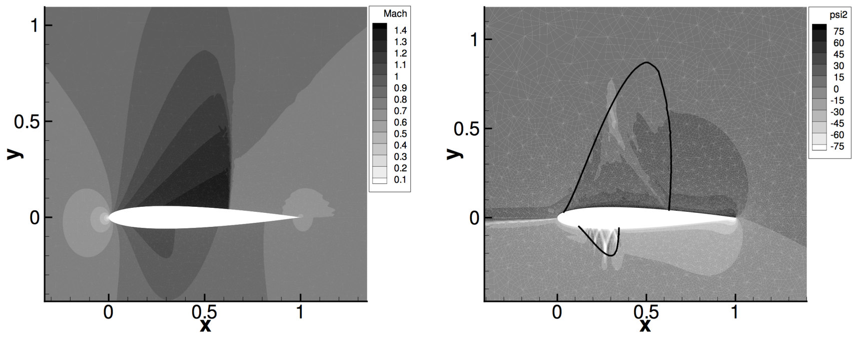
Figure 7: Comparison of goal-based p-adaptation with uniform polynomial refinement for subsonic laminar flow.

shock on the suction side of the aerofoil at 65% of the chord and a weaker shock on the pressure side at approximately 35% of the chord, both measured from the aerofoil's leading edge.

The target functional is the lift coefficient and it is evaluated following the same methodology used in the previous two cases. The computational domain is discretised into an unstructured mesh consisting of 8 446 triangles. The direct and adjoint steady-state solutions calculated using constant polynomial order $P = 6$ are shown in figures 8a and 8b, respectively. This is the reference solution for this case. The presence of dissipation terms that filter the high-order frequencies of the solution in the vicinity of shocks to ensure stability, does not justify the use of solutions at higher polynomial orders as reference.

The contour map of the adjoint variable ψ_2 in figure 8b shows clear traces of the characteristics carrying acoustic perturbations, i.e. flow compressions and expansions, throughout the supersonic regions which are enclosed by the Mach lines ($Ma = 1$) delineated in black in the figure. This is a very good illustration of the ability of the adjoint formulation to capture the mathematical essence of the problem. The adjoint solution also highlights the region at the foot of the strong shock on the suction side of the aerofoil as the major contributor to drag. In general, the sensitivity is highest in the vicinity of the shock but reduces significantly away from it.

The presence of shocks in the flow field requires shock capturing to stabilise the simulations. This is accomplished here through the introduction of appropriate solution-dependent dissipative terms. We use the discontinuity sensor proposed in [23] which identifies the presence of a shock by quantifying the smoothness of the solution within an element through comparison of solutions at two different polynomial orders. Large differences between these are interpreted as high-frequency oscillations triggered by the approximation of the shock, a discontinuity in inviscid flow, via polynomial functions that are continuous within



(a) Mach number

(b) Adjoint variable for ψ_2.

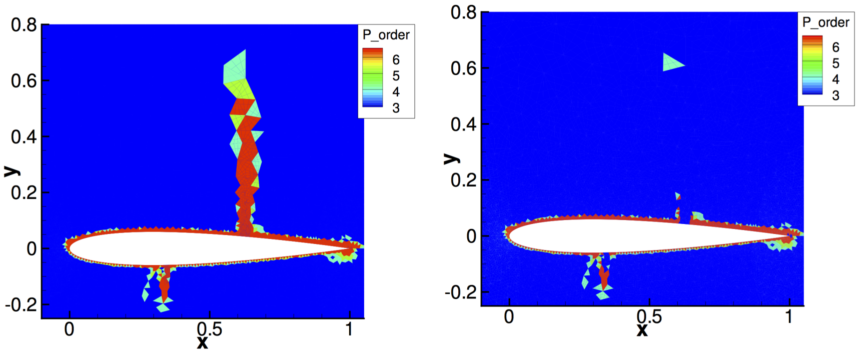
Figure 8: The solutions to the governing (8a) and adjoint (8b) equations for transonic inviscid flow past a NACA 0012 ($Ma = 0.8$, $\alpha = 1.25°$).

the element. The sensor is defined, for a generic variable u_i, as

$$S_e = \log_{10} \frac{\|u_{i,e}^{P} - u_{i,e}^{P-1}\|_{L_2}}{\|u_{i,e}^{P}\|_{L_2}} \tag{64}$$

Here we have used the density as the sensing variable, i.e. $u_i = \rho$. The discontinuity sensor is used to selectively apply the dissipative terms that dump numerical oscillations by effectively lowering the polynomial order approximation in those elements affected by the shock.

To assess the performance of the p-adaptive method in this context, we perform two simulations using constant order polynomials $P = 3$ and $P = 6$, and two p-adaptive simulations with polynomial orders in the range $3 \leq P \leq 6$. In the first p-adaptive simulation we apply the adaptation criterion as before in the whole domain without any special treatment of the regions affected by the shock. For the second p-adaptive simulation, we use the sensor (64) to identify the shock location and keep the polynomial order constant with a value $P = 3$ in those elements within the vicinity of the shock. The resulting distribution of polynomial orders from the p-adaptive simulations without and with polynomial order restriction are shown in figures 9a and 9b, respectively.



(a) Unrestricted polynomial order.

(b) Polynomial order restricted to $P = 3$ at shocks.

Figure 9: Comparison between the polynomial distributions obtained for the inviscid transonic flow case using the goal-based error indicator with restriction of the polynomial order to $P = 3$ at shocks (9b) and without (9a).

Table 3 presents the errors in the lift coefficients computed in these four cases with respect to a reference solution published by Yano and Darmofal [24, 25] that gives a value $c_l^{\text{ref}} = 0.35619$. Analysing the values of the relative error in table 3, it can be concluded that not much is gained by using solely p-adaptation using the goal-based error indicator. The comparison between the two polynomial distributions with and without restrictions in the polynomial order of the elements in the vicinity of the shock shows that the decrease in error is roughly comparable.

	$P = 3$	$3 \leq P \leq 6$	$3 \leq P \leq 6$	$P = 6$
			(shock filter)	
N_Q	168920	193460	189798	472976
t_{CPU}	0.65	0.72	0.70	1
c_l	0.333	0.336	0.334	0.336
ϵ_{c_l}	0.0232	0.0202	0.0222	0.0202

Table 3: Comparison of the error in lift coefficient using goal-based p-adaptation for transonic inviscid flow. The error is calculated with respect to the value $c_l^{\text{ref}} = 0.35619$ evaluated in references [24, 25].

Since we are using a shock capturing scheme that effectively reduces the order of the polynomial in the vicinity of a discontinuity through the addition of artificial viscosity, these results indicate that we achieve little gain in accuracy by allowing the order of the polynomial to increase in those elements near the shock. Therefore, these results also suggest that it would be preferable to increase the resolution there by keeping the polynomial degree fixed and decreasing the element size.

6.4. Subsonic inviscid flow past an ellipsoid ($Ma = 0.5$, $\alpha = 3^\circ$).

Finally we apply the goal-based p-adaptation technique to obtain improved estimates of lift for a three-dimensional inviscid subsonic flow past an ellipsoid of revolution with a free-stream Mach number $Ma = 0.5$. The lengths of the semi-axes of the ellipsoid are $0.2r$, r and $0.2r$, with $r = 0.25$. The flow considered is symmetric, the symmetry plane contains the major semi-axis which is at an incidence of 3° with the free-stream velocity vector. The ellipsoid is located in the centre of a $[10, 10, 10]$ box. The computational domain is discretised into an unstructured mesh, composed of 23 366 boundary-conforming high-order tetrahedra, which is shown in figure 10 for a constant polynomial order $P = 4$.

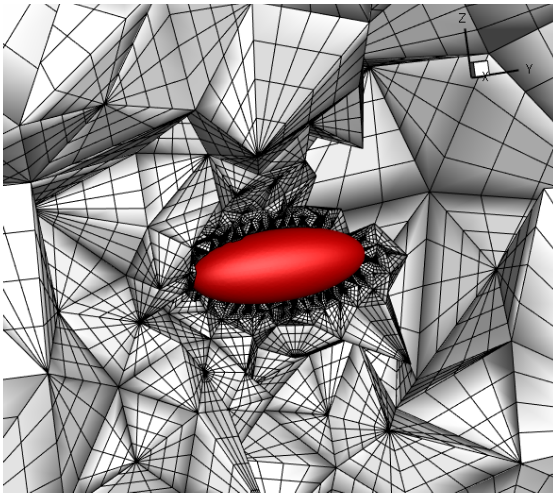


Figure 10: The tetrahedral mesh around an ellipsoid of revolution shown here with the degrees of freedom corresponding to a constant polynomial order $P = 4$.

The reference value of the lift coefficient, c_l^{ref}, is calculated from a steady-state solution obtained using a constant polynomial order $P = 4$. This permits us to afford the computational cost of this three-dimensional simulation. The solutions for the velocity components together with their corresponding adjoint variables are depicted in figure 11.

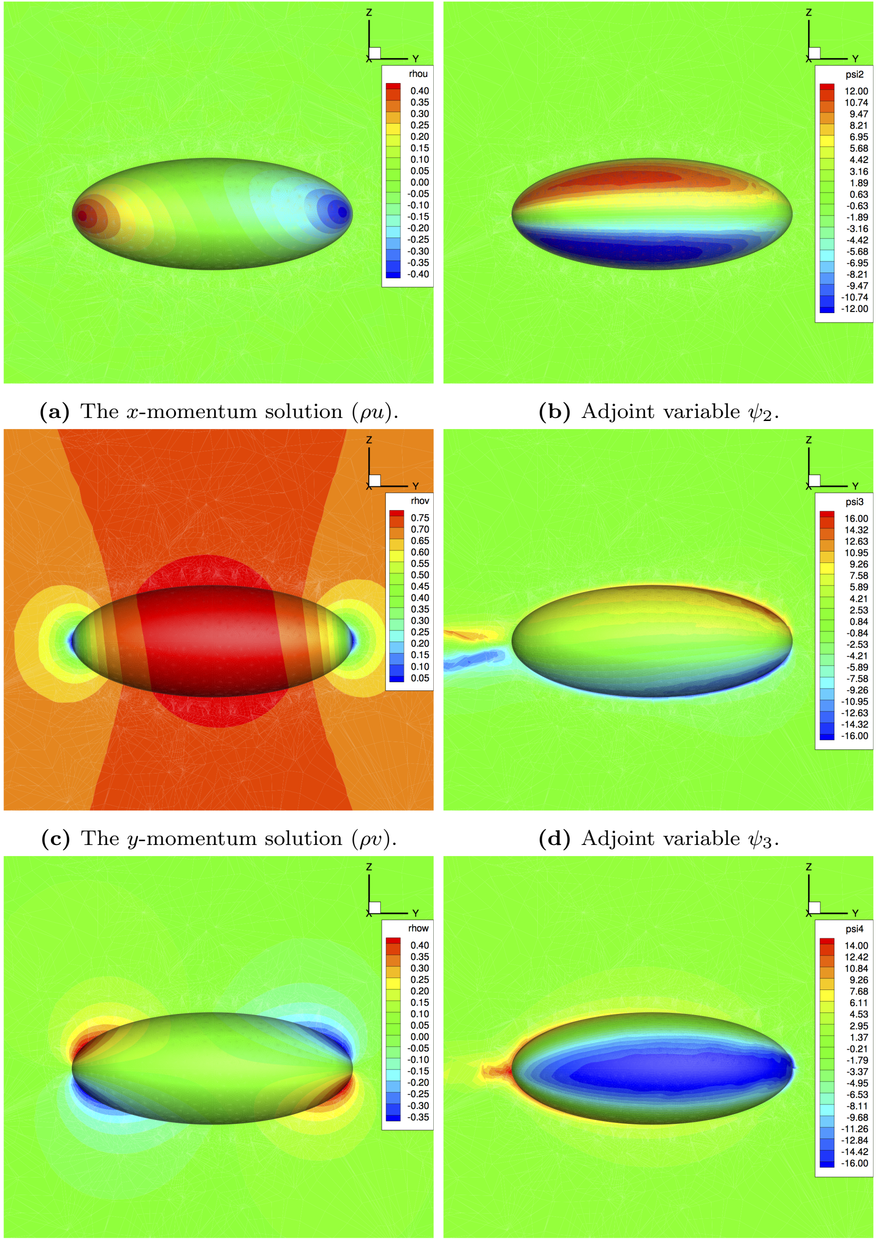
As in the previous examples, we compute a goal-based error indicator using equation (24). The absolute value of this error indicator is used to define a set of threshold values according to which we decide whether to increase the polynomial order or not at an element. After applying this strategy, the corresponding variable polynomial order distribution is shown in figure 12.

Since this case uses three-dimensional curvilinear elements, the choice of quadrature order can affect stability as shown in [26]. We therefore increase the number of integration points by a factor of 2, compared to the previous two-dimensional simulations, in each direction in order to avoid aliasing effects. The number of quadrature points, N_Q, in three dimensions is

$$N_Q = \sum_{e=1}^{N_{el}} 2^3 N_e^{\xi_1} N_e^{\xi_2} N_e^{\xi_3}, \tag{65}$$

where $N_e^{\xi_i}$ is the number of quadrature points, in the parametric direction ξ_i, within element e, and depends on P.

Table 4 shows a comparison of the error in lift coefficient calculated using goal-based p-adaptation and constant polynomial orders $P = 2$ and $P = 4$. The difference between the value of c_l obtained using goal-based p-adaptation and that using constant $P = 4$ is very small. However the goal-based p-adaptation strategy requires less than half of the resolution compared to the constant $P = 4$ solution and, in terms of CPU time per time step, it is decreased by approximately 30%. The error in the lift coefficient, c_l, is also reduced by two orders of magnitude compared to the $P = 2$ solution.



(a) The x-momentum solution (ρu).

(b) Adjoint variable ψ_2.

(c) The y-momentum solution (ρv).

(d) Adjoint variable ψ_3.

(e) The z-momentum solution (ρw).

(f) Adjoint variable ψ_4.

Figure 11: The solution for the velocity components in each Cartesian direction (x to z from top to bottom in the left-hand side column) and for their corresponding adjoint variable (ψ_2 to ψ_4 from top to bottom in the right-hand side column) for the flow past an ellipsoid of revolution at an incidence of 3°.

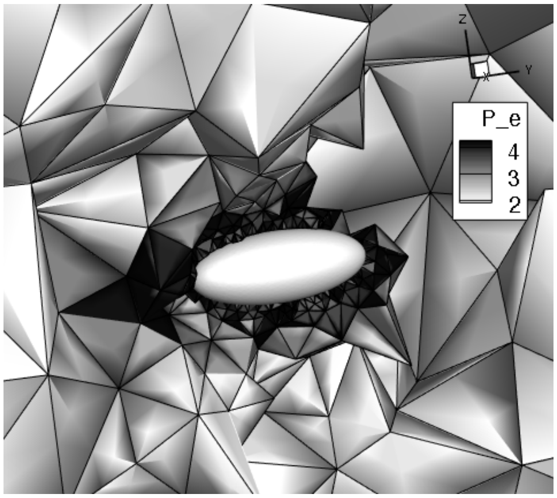


Figure 12: The new polynomial distribution based on the goal-based error indicator for the inviscid flow past an ellipsoid of revolution.

	$P = 2$	$2 \leq P \leq 4$	$P = 4$
N_Q	5.9×10^6	12.8×10^6	25.7×10^6
t_{CPU}	0.25	0.66	1.0
c_l	0.9658	1.0006	1.0005
ϵ_{c_l}	0.0347	10^{-4}	-

Table 4: Comparison of the error in lift coefficient obtained using goal-based p-adaptation for three-dimensional inviscid flow past an ellipsoid. The error is calculated with respect to the reference lift coefficient corresponding to $P = 4$.

7. Conclusions

We have presented a discontinuous Galerkin goal-based p-adaptation method for compressible flow problems. Overall, our results in both two- and three-dimensions demonstrate that this method is both a suitable and a promising adaptation strategy for sufficiently smooth inviscid and laminar flow problems. As shown in table 1 for our inviscid test case, the goal-based p-adaptive solution requires around 30% of the degrees of freedom and 46% less CPU time per time step to achieve the same order of accuracy as the reference simulation. The laminar test case also illustrates particularly well the benefits of using the sensitivity information following from the adjoint solution, since the boundary layer regions is clearly highlighted in the adjoint solution. For the three-dimensional subsonic inviscid case, we can draw similar conclusions as for the two-dimensional subsonic inviscid and laminar test cases. The error is reduced by two orders of magnitude compared to the $P = 2$ solution using goal-based p-adaptation. In this case approximately 50% fewer integration points, N_Q, were used compared to the reference case ($P = 4$). The CPU time reduction of 30% achieved is less compared to the two-dimensional cases. This can be ascribed to the de-aliasing strategy that was chosen by doubling the number of integration points in each spatial direction.

Unsurprisingly, the proposed p-adaptive strategy does not perform particularly well for flows with shocks, since one typically wants to avoid increasing the local approximation order in the vicinity of shocks. However, the adjoint solution presented in figure 8b illustrates that the error in aerodynamic force coefficient depends mainly on the accuracy of the solution at the root of the strong shock on the suction side. The importance of the accuracy of the solution reduces significantly when moving away from the surface according to the adjoint solution shown in figure 8b. It is therefore recommended to incorporate h-adaptation or mesh deformation, particularly at origin of the shock on the suction side of the aerofoil. However, as long as the solution is sufficiently smooth, goal-based p-adaptation is a suitable strategy to improve an initial coarse solution, both in terms of the number of quadrature points, N_Q, representing the mesh resolution, and of the CPU time per time step.

One of the challenging issues that could form the focus of future work in this area is the use of adjoint-based p-adaptation for unsteady flow problems. The main difficulty lies on the calculation of the unsteady adjoint solution. To enable the linearisation of the solution to the governing equations, we either need to store it in memory for each time step or recalculate it from a stored solution at a checkpoint in time using a checkpointing scheme. The latter strategy is often used for engineering problems that require both large memory storage and many time steps, like the ones presented in this paper, since the combination of these two factors makes the storage of the solution at each time step very impractical. A checkpointing scheme is proposed by Griewank et al. [27] which requires the *a priori* knowledge of the number of time steps. Wang et al. [28] introduced an optimal dynamic checkpointing algorithm for which the number of time steps does not need to be known *a priori* and only uses a fixed number of checkpoints for an arbitrary number of time steps. This dynamic checkpointing strategy is computationally more costly but it requires less memory to store the sensitivity fields for unsteady flows. Considering the scale of the computations we have dealt with here, the authors believe that the checkpointing scheme proposed by Wang et al. [28] is a suitable strategy for adjoint-based p-adaptation of unsteady compressible flows.

Acknowledgements

This work was partly supported by the Seventh Framework Programme of the European Commission as part of the ITN project ANADE under Grant Contract PITN-GA-289428.

Appendix A

The compressible Navier-Stokes equations are linearised to derive the corresponding adjoint equations. The expressions of the Jacobians of the convective and diffusive fluxes are given in the following sections.

Linearisation of the convective fluxes

The convective fluxes are linearised with respect to the vector of conserved variables $\boldsymbol{u} = [\rho, \rho v_1, \rho v_2, \rho E]^t$. The convective fluxes are given in equation (2). The Jacobian matrices for the convective fluxes are

$$\frac{\partial \boldsymbol{f}_1^c}{\partial \boldsymbol{u}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(\gamma-1)q^2}{2} - v_1^2 & (3-\gamma)v_1 & -(\gamma-1)v_2 & \gamma - 1 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ v_1\left(\frac{(\gamma-1)q^2}{2}\right) & H - (\gamma-1)v_1^2 & -(\gamma-1)v_1 v_2 & \gamma v_1 \end{bmatrix} \tag{66}$$

$$\frac{\partial \boldsymbol{f}_2^c}{\partial \boldsymbol{u}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -v_1 v_2 & v_2 & v_1 & 0 \\ \frac{(\gamma-1)q^2}{2} - v_2^2 & -(\gamma-1)v_1 & (3-\gamma)v_2 & \gamma - 1 \\ v_2\left(\frac{(\gamma-1)q^2}{2}\right) & -(\gamma-1)v_1 v_2 & H - (\gamma-1)v_2^2 & \gamma v_2 \end{bmatrix} \tag{67}$$

where $q^2 = v_1^2 + v_2^2$ and the total enthalpy is indicated by H.

Linearisation of the diffusive fluxes

It is more convenient to perform the differentiation of the diffusive fluxes if we express them in terms of the primitive variables $\tilde{\boldsymbol{u}} = [\rho, v_1, v_2, p]^t$. Since the diffusive fluxes are dependent on the first derivatives of the velocity component rather than the momentum components it becomes easier to linearise with respect to $\tilde{\boldsymbol{u}}$. However for consistency, we have to apply the chain rule to obtain $\frac{\partial \vec{\boldsymbol{f}}^v}{\partial \boldsymbol{u}}$. The Jacobians obtained when linearising the diffusive fluxes with respect to the vector of conservative variables, $\boldsymbol{u}$, can be written as

$$\frac{\partial \vec{\boldsymbol{f}}_i^v}{\partial \boldsymbol{u}} = \frac{\partial \tilde{\boldsymbol{u}}}{\partial \boldsymbol{u}} \frac{\partial \vec{\boldsymbol{f}}_i^v}{\partial \tilde{\boldsymbol{u}}} \tag{68}$$

where $\frac{\partial \tilde{\boldsymbol{u}}}{\partial \boldsymbol{u}}$ is given by

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial \boldsymbol{u}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{v_1}{\rho} & \frac{1}{\rho} & 0 & 0 \\ -\frac{v_2}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{(\gamma-1)q^2}{2} & -v_1(\gamma-1) & -v_2(\gamma-1) & \gamma-1 \end{bmatrix} \tag{69}$$

The derivatives of the viscous fluxes, $\frac{\partial \boldsymbol{f}_1^v}{\partial \boldsymbol{u}}$ and $\frac{\partial \boldsymbol{f}_2^v}{\partial \boldsymbol{u}}$, are 4×4 Jacobian matrices given by

$$\frac{\partial \boldsymbol{f}_1^v}{\partial \tilde{\boldsymbol{u}}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_1}\left(\frac{1}{\rho}\right) & \tau_{11} & \tau_{12} & \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_1}\left(\frac{p}{\rho^2}\right) \end{bmatrix} \tag{70}$$

$$\frac{\partial \boldsymbol{f}_2^v}{\partial \tilde{\boldsymbol{u}}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_2}\left(\frac{1}{\rho}\right) & \tau_{21} & \tau_{22} & \frac{\gamma\mu}{(\gamma-1)Pr} \frac{\partial}{\partial x_2}\left(\frac{p}{\rho^2}\right) \end{bmatrix} \tag{71}$$

The derivatives of the diffusive fluxes with respect to the first derivatives are given by

$$\frac{\partial \boldsymbol{f}_1^v}{\partial \tilde{\boldsymbol{u}}_{x_1}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{4\mu}{3} & 0 & 0 \\ 0 & 0 & \mu & 0 \\ \frac{kp}{R\rho^2} & \frac{4\mu v_1}{3} & \mu v_2 & \frac{k}{R\rho} \end{bmatrix} \quad \frac{\partial \boldsymbol{f}_1^v}{\partial \tilde{\boldsymbol{u}}_{x_2}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{2\mu}{3} & 0 \\ 0 & \mu & 0 & 0 \\ 0 & \mu v_2 & -\frac{2\mu v_1}{3} & 0 \end{bmatrix} \tag{72}$$

$$\frac{\partial \boldsymbol{f}_2^v}{\partial \tilde{\boldsymbol{u}}_{x_1}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & -\frac{2\mu}{3} & 0 & 0 \\ 0 & -\frac{2\mu v}{3} & \mu v_1 & 0 \end{bmatrix} \quad \frac{\partial \boldsymbol{f}_2^v}{\partial \tilde{\boldsymbol{u}}_{x_2}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \frac{4\mu}{3} & 0 \\ \frac{kp}{R\rho^2} & \mu v_1 & \frac{4\mu v_2}{3} & \frac{k}{R\rho} \end{bmatrix} \tag{73}$$

Appendix B

The Roe-averaged Riemann flux in the normal direction is given by

$$\mathcal{H}^c(\boldsymbol{u}_{ex}, \boldsymbol{u}_{in}) = \frac{1}{2}\left(\boldsymbol{f}^c_{n,in} + \boldsymbol{f}^c_{n,ex}\right) - \frac{1}{2}\|\Lambda\| T^{-1}\Delta\boldsymbol{u} \tag{74}$$

where $\Delta\boldsymbol{u} = \boldsymbol{u}_{ex} - \boldsymbol{u}_{in}$ indicate the solution at the internal and external interface respectively. Furthermore, Λ and T follow from diagonalising the Jacobian matrix of the convective flux. Hence, Λ denotes the diagonal 4×4 eigenvalue matrix and T denotes a 4×4 matrix containing the right eigenvectors.

We use equation (74) to construct an adjoint Roe-averaged Riemann solver. First we want to obtain the appropriate formulation for T^{-1}. For the adjoint problem, the Jacobian is transposed and the sign is negative. The transposed Jacobian for the flux in the normal direction gives

$$\left[\frac{\partial \boldsymbol{f}^c_1}{\partial \boldsymbol{u}}\right]^t = \begin{bmatrix} 0 & -u^2 + \frac{\gamma-1}{2}q^2 & -uv & u\left(\frac{\gamma-1}{2}q^2 - H\right) \\ 1 & (\gamma-3)u & v & H - (\gamma-1)u^2 \\ 0 & -(\gamma-1)v & u & -(\gamma-1)uv \\ 0 & (\gamma-1) & 0 & \gamma u \end{bmatrix} \tag{75}$$

In equation (7), u and v represent the velocity components normal and parallel to the interface respectively. The Riemann problem is transformed into a one-dimensional one by mapping the local quantities onto the normal of the interface. The matrix of right eigenvectors of the transposed Jacobian, $-\left[\frac{\partial \boldsymbol{f}^c_n}{\partial \boldsymbol{u}}\right]^t$, is

$$T = \begin{bmatrix} \frac{q^2}{2} + \frac{cu}{\gamma-1} & u^2 - H & -v & \frac{q^2}{2} - \frac{cu}{\gamma-1} \\ -u - \frac{c}{\gamma-1} & -u & 0 & -u + \frac{c}{\gamma-1} \\ -v & 0 & 1 & -v \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{76}$$

The inverse of T is given by

$$T^{-1} = \frac{\gamma - 1}{2c^2} \begin{bmatrix} 1 & (c-u) & v & (H-uc) \\ -2 & -2u & -2v & -q^2 \\ 2v & 2uv & 2v^2 + \frac{2c^2}{\gamma-1} & 2vH \\ 1 & (c+u) & v & (H+uc) \end{bmatrix} \tag{77}$$

We substitute T in the definition of the Roe-averaged Riemann flux (74) and, using the same analogy as for the governing equations, we write an adjoint Roe-averaged numerical flux function as

$$\hat{\mathcal{H}}(\boldsymbol{\psi}_{ex}, \boldsymbol{\psi}_{in}) = -\frac{1}{2}\left(\left\{\left[\frac{\partial \boldsymbol{f}^c_n}{\partial \boldsymbol{u}}\right]^t \boldsymbol{\psi}^\delta_e\right\}_{in} - \left\{\left[\frac{\partial \boldsymbol{f}^c_n}{\partial \boldsymbol{u}}\right]^t \boldsymbol{\psi}^\delta_e\right\}_{ex}\right) - \frac{1}{2}\|\Lambda\|T^{-1}\Delta\boldsymbol{\psi} \tag{78}$$

References

[1] K. J. Fidkowski, D. L. Darmofal, Review of output-based error estimation and mesh adaptation in computational fluid dynamics, AIAA Journal 49 (4) (2011) 673–694.

[2] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numerica 10 (2001) 1–102.

[3] M. G. Larson, T. J. Barth, A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems, Technical Report NAS-99-010, NAS (1999).

[4] M. B. Giles, N. A. Pierce, Adjoint equations in CFD: Duality, boundary conditions and solution behaviour, AIAA Paper 97-1950 (1996).

[5] P. Houston, R. Hartmann, E. Süli, Adaptive discontinuous Galerkin finite element methods for compressible fluid flows, in: M. Baines (Ed.), Numerical methods for Fluid Dynamics VII, ICFD, 2001, pp. 347–353.

[6] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws, SIAM Journal of Scientific Computing 24 (2002) 979–1004.

[7] D. A. Venditti, D. L. Darmofal, Grid adaptation for functional outputs: Application to two-dimensional inviscid flows, Journal of Computational Physics 176 (1) (2002) 40–69.

[8] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, Journal of Computational Physics 183 (2) (2002) 508–532.

[9] L. Demkowicz, W. Rachowicz, P. Devloo, A fully automatic *hp*-adaptivity, Journal of Scientific Computing 17 (1) (2002) 117–142.

[10] P. Solín, L. Demkowicz, Goal-oriented *hp*-adaptivity for elliptic problems, Computer Methods in Applied Mechanics and Engineering 193 (1) (2004) 449–468.

[11] L. Y. Li, Y. Allaneau, A. Jameson, Comparison of *h*- and *p*-adaptations for spectral difference methods, in: 40th AIAA Fluid Dynamics Conference and Exhibit, 2010.

[12] G. Giorgiani, S. Fernández-Méndez, A. Huerta, Goal-oriented *hp*-adaptivity for elliptic problems, International Journal for Numerical Methods in Fluids 72 (1) (2013) 1244–1262.

[13] M. Woopen, A. Balan, G. May, J. Schütz, A comparison of hybridized and standard DG methods for target-based *hp*-adaptive simulation of compressible flow, Computers and Fluids 98 (2014) 3–16.

[14] K. J. Fidkowski, High-order output-based adaptive methods for steady and unsteady aerodynamics, VKI Lecture Notes (2013).

[15] A. Jameson, S. Shankaran, L. Martinelli, Continuous adjoint method for unstructured grids, AIAA Journal 46 (5) (2008) 1226–1239.

[16] I. M. Gelfand, S. V. Fomin, Calculus of variations, Dover publications, 1963.

[17] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. de Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. M. Kirby, S. J. Sherwin, Nektar++: An open-source spectral/hp element framework, Computer Physics Communications 192 (2015) 205–219.

[18] G. E. Karniadakis, S. J. Sherwin, Spectral/hp element methods for computational fluid dynamics, 2nd Edition, Oxford Science Publications, 2005.

[19] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, Journal of Computational Physics 131 (1996) 267–279.

[20] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics: A practical introduction, Springer, 2009.

[21] R. M. Kirby, G. E. Karniadakis, Selecting the numerical flux in discontinuous Galerkin methods for diffusion problems, SIAM Journal of Scientific Computing 22-23 (1-3) (2005) 385–411.

[22] S. J. Sherwin, A high order Fourier/unstructured discontinuous Galerkin method for hyperbolic conservation laws, in: R. Jeltsch, M. Fey (Eds.), Hyperbolic Problems: Theory, Numerics, Applications, Vol. 130 of International Series of Numerical Mathematics, Birkhäuser Basel, 1999, pp. 875–884.

[23] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin method, in: 44th AIAA Aerospace Sciences Meeting and Exhibit (AIAA Paper 2006-112), 2006.

[24] M. Yano, D. L. Darmofal, Case C1.3 Flow over the NACA 0012 airfoil: subsonic inviscid, transonic inviscid, and subsonic laminar flows. Contribution

to the First Workshop of High-order CFD Methods held at the 50th AIAA Aerospace Sciences Meeting, Nashville, Tennessee, January 7–8 (2012).

[25] Z. J. Wang, K. J. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (2013) 811–845.

[26] G. Mengaldo, D. De Grazia, D. Moxey, P. E. Vincent, S. J. Sherwin, Dealiasing techniques for high-order spectral element methods on regular and irregular grids, Journal of Computational Physics 299 (2015) 56–81.

[27] A. Griewank, A. Walther, Revolve: An implementation of check-pointing for the reverse or adjoint mode of computational differentiation, ACM Transactions on Mathematical Software 26 (2000) 19–45.

[28] Q. Wang, P. Moin, G. Iaccarino, Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation, SIAM Journal of Scientific Computing 31 (4) (2009) 2549–2567.