Comparisons of p-adaptation strategies based on truncation- and discretisation-errors for high order discontinuous Galerkin methods

Moritz Kompenhans[a,1], Gonzalo Rubio[a,2], Esteban Ferrer[a,3], Eusebio Valero[a,4]

[a] *ETSIAE (School of Aeronautics)*
Universidad Politécnica de Madrid
Plaza de Cardenal Cisneros 3, 28040 Madrid
Spain

Abstract

In this work, local p-adaptation strategies for high order discontinuous Galerkin spectral element methods are compared. The principal aim of the paper is to determine the advantages and drawbacks of various sensors, based on truncation or discretisation errors, to detect the regions that require adaption. A well-established discretisation error based approach, estimated through the decay of the energy associated to the approximated solution modes, is compared to recently proposed truncation error adaptation methodologies (i.e. isotropic and anisotropic versions). The truncation error technique detects the regions that require refinement by estimating the truncation error in meshes with varying degrees of freedom. This comparison is particularly

[1] moritz.kompenhans@upm.es
[2] g.rubio@upm.es
[3] esteban.ferrer@upm.es
[4] eusebio.valero@upm.es

interesting since the truncation error is related to the discretisation error through a Discretisation Error Transport Equation, where the truncation error appears as a source term for the discretisation error.
The comparisons included quantify the accuracy of the flow solutions resulting from meshes adapted with these sensors and hence provide guidelines into which sensors are better suited to adapt meshes for inviscid and viscous flows. All the adaption strategies are tailored to high order methods and particularly implemented and tested in a discontinuous Galerkin solver.

Results include an inviscid NACA0012 and a viscous flat plate boundary layer. Output functionals (e.g. lift, drag) resulting from adapted meshes are compared in terms of the number of degrees of freedom, providing a critical assessment of the performance of each sensor. Namely, it is shown that both truncation error adaptation methodologies provide meshes with polynomial order distributions that lead to fewer degrees of freedom than when using the discretisation error based adaptation. In addition, the examples illustrate the outperforming advantage of anisotropic over isotropic adaptation.
Keywords: anisotropic p-adaptation, high order discontinuous Galerkin, Tau-estimation (τ-estimation), truncation error, discretisation error

---

1. Introduction

Computational Fluid Dynamic or CFD simulations require large numbers of Degrees of Freedom (DoF) to capture the complex physics governed by the Navier-Stokes equations. This set of non-linear partial differential equations

accommodate for a wide range of temporal and spatial scales that can render its numerical resolution difficult. The range of spatial scales necessitates a varying number of DoF to attain good numerical accuracy. A naive approach may consider adapting all flow regions uniformly to attain higher accuracy overall, but it is also possible to increase/decrease the DoF locally. The latter strategy can reduce the overall number of DoF and consequently minimise the computational cost. Historically, mesh adaptation methods for fluid flows have increased the number of mesh elements or grid nodes (h-refinement), which can be seen as the natural strategy when low order methods (e.g. Finite Volumes, Finite Elements) are considered.

High order methods (e.g. Spectral or discontinuous Galerkin) offer an alternative to low order methods and to the classical h-refinement strategy. Such methods enable the use of high order polynomials inside each computational element to approximate the numerical solution, which decreases the error exponentially when smooth flows are considered [50, 25, 12].

Discontinuous Galerkin (DG) methods have seen increasing popularity during recent years and have been used to solve a variety of problems including compressible [5, 29, 32, 33, 28] and incompressible [4, 37, 9, 43, 14, 15, 13] flows. DG methods enable discontinuities in the numerical solution between mesh elements, which facilitates local adaptation strategies since mesh discontinuities (e.g. hanging nodes) and polynomial discontinuities can be handled naturally and accurately. Local adaptation techniques for DG methods are relatively new and may consider h or p refinement independently. When

the numerical solution is smooth, p-refinement outperforms h-refinement since the former leads to an exponential decrease of the error. However, when flow discontinuities are present in the numerical solution (e.g. shock waves or geometric discontinuities), then h-refinement becomes advantageous.

The previous discussion underlines three of the main difficulties when dealing with local mesh adaptation in the framework of high order methods. First, it is necessary to locate the flow regions that, when adapted, will improve the accuracy of the numerical solution. Second, when high order techniques are used to solve the underlying equations, then it is necessary to select the adaptation strategy (h or p refinement) that minimises the error locally. Third, if the flow presents an anisotropic character (e.g. boundary layers), then the adaptation strategy can take advantage of an anisotropic adaptation to refine/coarsen the mesh following particular directions. Examples of h and p refinement strategies can be found in Mavriplis [31], Van der Vegt *et al.* [48] or Roy *et al.* [38].

To determine the region that requires adaptation (i.e. refinement or coarsening) various methods exist. Historically, the so called "feature based adaptation" has been broadly applied to improve the accuracy of numerical solutions [11, 1]. These methods identify regions for adaptation by quantifying flow gradients. This approach does not rely on a direct relation between the computed feature and the numerical errors and therefore the resulting mesh does not guarantee a reduction of the error. Extension to high order methods [31, 34, 3] may use the polynomial energy decay for each mode (that

approximate the numerical solution) to account for the flow gradients (i.e. quantifying the solution smoothness). These methods can also be seen as discretisation error based, since the polynomial energy decay for each mode is a measure of the discretisation error (i.e. the difference between the exact solution of the PDE and the exact solution of the discretised PDE) [8]. Discretisation error based adaptation methods neglect the fact that the discretisation error can be transported throughout the mesh domain, resulting in regions with large discretisation errors that have actually small discretisation error contributions [51, 21].

The truncation error technique appeared as a cost efficient methodology to select regions that require adaptation. The truncation error is defined as the difference between the discrete Partial Differential Equation (PDE) and the exact PDE operator, both applied to the exact solution of the problem [35]. This may be stated as

$$\tau^N = \mathcal{R}^N(u) - \mathcal{R}(u), \tag{1}$$

where τ^N denotes the truncation error for a polynomial order N, $\mathcal{R}$ is the partial differential operator, $\mathcal{R}^N$ the discrete partial differential operator (of order N) and u represents the exact solution.

The discretisation error and the truncation error are related through the Discretisation Error Transport Equation (DETE) [39] such that the truncation error acts as a local source for the discretisation error. This relationship

provides a compelling argument for the use of the truncation error, instead of the discretisation error, as a sensor for a mesh adaptation algorithm. In addition the DETE shows that truncation errors do not suffer from transported and cumulative effects (as for discretisation errors) and hence can be better suited to select regions for adaption.

Examples of τ-estimation method for low order spatial discretisations can be found in [6, 7, 20, 46] for finite differences, and in [45, 44, 16, 19, 17] for finite volumes. More recently the method has been extended by the authors to continuous high order [41] and discontinuous Galerkin formulations [42, 26].

Finally, adjoint methods enable the estimation of the discretisation error associated with functional outputs. This family of methods has been extended to high order discontinuous Galerkin methods [49, 23, 22, 24] but require the solution of the dual problem and generally the explicit storage of an embedded grid. Comparisons between adjoint methods for adaptation and truncation error can be found in [18, 10]. In addition, an explicit relationship between adjoint methods, discretisation error and truncation errors have been derived by the authors [26]. The latter shows that the truncation error controls the discretisation error and also any output functional error. Consequently, adaption strategies based on the truncation error should provide more solutions with lower discretiation errors and more accurate output functionals. Adjoint methodologies being a wide field of research, they are not considered in this work.

In summary, this paper includes comparisons of p-refinement strategies for adaptation using discontinuous Galerkin methods. An isotropic discretisation error adaption [31, 34, 3], based on the decay of the energy (i.e. L_2 norm) associated to the different modes, is compared to truncation error implementations. In particular, we compare isotropic and anisotropic truncation error methodologies, as well as two definitions for the truncation error, i.e. non-isolated and isolated [42].

The paper is organised as follows. The first section details the discontinuous Galerkin formulation and introduces various definitions for solution errors that are necessary to define the adaptation strategy. The second section details the adaptation strategies that consider discretisation, truncation and isolated truncation errors. In the third section, the adaptation strategies are compared for an inviscid NACA0012 and a viscous flat plate boundary layer test case. The comparison includes the number of degrees of freedom required by each method to attain a similar level of accuracy. Finally, the paper provides conclusions of the compared strategies.

2. Mathematical background

In this section the mathematical background of the problem is introduced. First, the Discontinuous Galerkin Spectral Element Method (DGSEM) is briefly advanced. Second, the errors that will serve as a basis for the refinement criteria are defined.

2.1. Discontinuous Galerkin Spectral Element Method

Discontinuous Galerkin methods were first developed [36] to solve conservation laws of the type: $\mathcal{F}(u) = u_t + \nabla \cdot \mathbf{f} = 0$. To apply the discontinuous Galerkin method, we first tessellate the physical domain Ψ to obtain its discrete version Ψ_h. This variational technique requires multiplying the conservation law at an elemental level Ω by a test function (typically polynomials) and integrate. By doing so, it is ensured that the residual of the equation is orthogonal (in the L_2 norm) to the space spanned by the test functions. Summing over all elemental contributions in the mesh domain (Ψ_h) and integrating by parts leads to the weak formulation of the problem and includes volume and surface contributions. Since in the discontinuous framework, discontinuities are allowed at edge interfaces, dual valued functions co-exist at the interfaces between neighbouring elements. To resolve the discontinuity associated to the convective fluxes, a Roe approximate Riemann solver is used (see Toro's monograph [47] for details). In addition, discontinuities associated to the viscous terms are resolved using a Bassi-Rebay formulation [5, 2]. Finally, boundary conditions are weakly imposed at domain boundaries. The resulting volume and surface integrals are numerically approximated using Gauss quadrature rules and lead to a discrete set of equations for each nodal point.

A nodal variant of discontinuous Galerkin technique that uses a quad/hexa mesh topology and tensor product expansions for the polynomial spaces is known as Discontinuous Galerkin Spectral Element Method (DGSEM), as

detailed in Kopriva [28]. The compressible Navier-Stokes equations can be discretised using the DGSEM approach [27, 28, 26]. The temporal terms are discretised using a third order Runge-Kutta method and are converged until steady state, which is reached once the the infinite norm of the conservation law residual (for all flow equations) falls below a tolerance not larger than 10^{-8} .

2.2. Definitions for discretisation and truncation errors

Let us consider the high order discretisation of a Partial Differential Equation (PDE) of the following form:

$$\int_{\Psi} \mathcal{F}(u)\phi dx = 0 \rightarrow \sum_{el \in \Psi_h} \mathcal{R}^N(u^N) = 0, \tag{2}$$

where ϕ is the test function, u^N represents the discrete solution of the PDE, N is the degree of the polynomial used to represent the solution inside each spectral element $el \in \Psi_h$ and $\mathcal{R}^N$ represents the discrete spatial partial differential operator. Henceforth, we assume that the solution is converged in time such that $u_t = 0$ and consequently $\mathcal{R}^N(u^N)$ is independent of time.

The DGSEM allows a straightforward separation between interior and inter-element contributions. Following [42] we define the *isolated* discrete partial differential operator

$$\hat{\mathcal{R}}^N(u^N) = \mathcal{R}^N_{\partial el}(u^N) + \mathcal{R}^N_{el}(u^N), \tag{3}$$

where $\mathcal{R}^N_{\partial el}(u)$ and $\mathcal{R}^N_{el}(u)$ represent the inter-element (surface terms) and interior (volume terms) contributions respectively, and the discrete partial differential operator

$$\mathcal{R}^N(u^N) = \mathcal{R}^{N\ *}_{\partial el}(u^N) + \mathcal{R}^N_{el}(u^N) = 0, \tag{4}$$

where $\mathcal{R}^{N\ *}_{\partial el}(u)$ represents the inter-element (surface terms) after solving the Riemann problem. From a practical point of view, $\hat{\mathcal{R}}^N(u^N)$ can be computed by evaluating $\mathcal{R}^N(u^N)$ without solving the Riemann problem arising at element edge discontinuities and leading to the surface terms, which means that each of the elements is isolated from its neighbors (i.e. minimising convective and diffusive effects).

The discretisation error, ϵ^N, and the truncation error, τ^N, corresponding to (2) are defined as

$$\begin{aligned} \epsilon^N &= u - u^N, \\ \tau^N &= \mathcal{R}^N(u), \end{aligned} \tag{5}$$

where u^N denotes the discrete numerical solution and u its continuous counterpart. Furthermore from (3) Rubio *et al.* [42] defined the *isolated* truncation error as

$$\hat{\tau}^N = \hat{\mathcal{R}}^N(u). \tag{6}$$

To apply the discrete operator, $\mathcal{R}^N$, or the isolated discrete operator, $\hat{\mathcal{R}}^N$,

to a continuous solution, a simple injection to the quadrature nodes is performed.

It should be noticed that the definitions of the three errors (5) and (6) include the exact solution of the problem, u, which is generally unknown. In the next section, we introduce error estimation methods that are subsequently used as adaptation refinement criteria.

3. Adaptation Process

In this section, we firstly detail the adaptation criteria for p-adaptation to secondly explain the algorithm used to refine/coarse the polynomial space.

3.1. Refinement criteria

One of the main difficulties when dealing with local mesh adaptation is to locate the flow regions that when adapted will improve the numerical solution. On the one hand, a good refinement criteria will generate a mesh with minimum error for a given number of degrees of freedom. On the other hand, the computational cost for the computing the refinement criteria has to be small such that the adaptation process is efficient.

This paper targets the point showing the evolution of output functionals under adaptation algorithms based on three different refinement criteria. Each of the criteria will be based in the estimation of the errors defined in the previous section, i.e. discretisation error, truncation error and isolated truncation error.

As stated in the introduction, the discretisation error and the truncation error are related through the Discretisation Error Transport Equation (DETE) [39] such that the truncation error acts as a local source for the discretisation error. Consequently, the truncation error is expected to be a better sensor for mesh adaptation. Besides the isolated truncation error (see previous section) only considers the interior element contributions of the truncation error [42] and, as a result, is expected to perform even better than the truncation error.

To estimate the error introduced in the previous section, we use a posteriori estimations. A posteriori methods require the computation of a converged approximate solution to compute the estimation of the error. It will be shown that this requirement can be relaxed in the truncation error and isolated truncation error estimations since non time-converged solution may be used. However, for the sake of simplicity but without loss of generality, we assume only time converged solutions. The approximate solution is obtained by solving the same problem with a higher polynomial order on each element. Then, the error can be estimated for all polynomial orders lower than specified, e.g. τ_P^N is the truncation error estimation using a fine simulation with order P to estimate the coarse error with polynomial order N, when $N < P$.

Discretisation error based. The discretisation error based refinement criterion was introduced by Mavriplis [30] and more recently used by Barosan *et al.* [3]. This criterion necessitates of the evaluation and extrapolation of

the spectrum of a discretisation based on Legendre polynomials (i.e. modal basis functions). The approximate solution of order P in each element can be written as

$$u^P = \sum_{n=0}^{P} a_n L_n, \tag{7}$$

where L_n is the Legendre polynomial of degree n and a_n is the spectral coefficient (i.e. modal energy). The estimation of the discretisation error for the Legendre polynomial approximation reads

$$\epsilon_P^N = \left(\frac{a_N^2}{\frac{1}{2}(2N+1)} + \int_{N+1}^{\infty} \frac{[a(n)]^2}{\frac{1}{2}(2N+1)} \right), \tag{8}$$

where function $a(n)$ is a least squares best fit of the last six points of the spectrum. To perform the least squares best fit an exponential decay, $a(n) \sim ce^{-\sigma n}$, is assumed. Note that this method estimates the solution error and it is a measure of the discretisation error, but may also viewed as an estimator for the smoothness (i.e. determines the order polynomial necessary) and hence as a "feature based adaptation" method.

Let us note that to apply this method to our nodal DGSEM approach, we need to project our nodal basis into a modal space constituted by Legendre polynomials (e.g. using a generalised Vandermonde matrix [25]). This is an element-wise operation with a reduced computational cost.

Truncation error based. The truncation error based refinement criterion is based on the τ-estimation method. The expressions to estimate the trun-

cation error were first deduced by Fraysse *et al.* [16] for finite volume schemes and subsequently extended to spectral Chebyshev collocation method and to DGSEM by Rubio *et al.* [41, 42]. This estimation method is quasi-*a priori* as it uses a non fully time-converged solution, $\tilde{u}^P$, to perform the estimation. For non-converged solutions and non-linear partial differential operators the approximated truncation error becomes

$$\tau_P^N \equiv \mathcal{R}^N\left(\tilde{u}^P\right) - \bar{I}_P^N \mathcal{R}^P\left(\tilde{u}^P\right), \tag{9}$$

where $\bar{I}_P^N$ is the transfer operator of the residual from order P to N, that defined as

$$\bar{I}_P^N \equiv \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} I_P^N \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{u^P}\right)^{-1}. \tag{10}$$

The difference between the exact and the approximate truncation error reads

$$\tau_P^N = \tau^N - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon_{it}^P\right)^2, \tag{11}$$

where ϵ_{it}^P represents the iteration error, i.e. the difference between the converged and the non-converged approximate solutions. If the second term in the RHS of (9) is not computed (notice the high cost of its computation), the difference between the exact and the approximate truncation error reads:

$$\tau_P^N = \tau^N - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \left(\epsilon^P + \epsilon_{it}^P\right) + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon_{it}^P\right)^2. \tag{12}$$

A proof of (11) is included in the Appendix and more information can be found in [42, 40]. It should be noticed that to apply the discrete operator, $\mathcal{R}^N$, to a solution of different order, u^P, it is necessary to evaluate this solution at the Gauss-Legendre nodes of order N, i.e. to interpolate to a coarse grid. For compactness, the notation in this work omits the interpolant such that $\mathcal{R}^N u^P = \mathcal{R}^N I_P^N u^P$.

Isolated truncation error based. The isolated truncation error based refinement criterion is also based on the τ-estimation method. Rubio *et al.* [42] shown that for non-converged solutions and non-linear partial differential operators, the approximated isolated truncation error becomes

$$\hat{\tau}_P^N \equiv \hat{\mathcal{R}}^N \left(\tilde{u}^P\right) - \hat{I}_P^N \mathcal{R}^P \left(\tilde{u}^P\right). \tag{13}$$

Besides $\hat{I}_P^N$, the transfer operator of the residual from order P to N for the isolated truncation error is,

$$\hat{I}_P^N \equiv \left.\frac{\partial \hat{\mathcal{R}}^N}{\partial u^N}\right|_{\tilde{u}^P} I_P^N \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P} \right)^{-1} \tag{14}$$

and the difference between the exact and the approximate truncation error reads

$$\hat{\tau}_P^N = \hat{\tau}^N - \left.\frac{\partial \hat{\mathcal{R}}^N}{\partial u^N}\right|_{u} \epsilon^P + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon_{it}^P\right)^2. \tag{15}$$

If the second term in the RHS of (13) is not computed (notice the high cost of its computation), the difference between the exact and the approximate

truncation error reads:

$$\hat{\tau}_P^N = \hat{\tau}^N - \left.\frac{\partial \hat{\mathcal{R}}^N}{\partial u^N}\right|_u \left(\epsilon^P + \epsilon_{it}^P\right) + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon_{it}^P\right)^2 . \tag{16}$$

The interested reader is referred to the Appendix of this document for a proof of (15). Additional information can be found in [42, 40]. From a practical point of view, $\hat{\mathcal{R}}^N(u^N)$ can be computed by evaluating $\mathcal{R}^N(u^N)$ without solving the Riemann problem for the surface terms, which means that each of the elements is isolated from its neighbors. For a more detailed explanation, the reader is referred to [40].

Remarks on the accuracy and computational cost of the estimations.

- Accuracy:

  The three error estimators described in this section, (8), (9) and (13), are accurate only if the asymptotic rate of convergence has been reached. In particular, it can be explicitly seen in (11) and (15) that the discretisation error in the fine mesh (polynomial order P), acts as a source of inaccuracy in the estimation. A similar result might be obtained for the discretisation error estimation. Therefore it is assumed that the discretisation error in the fine mesh, ϵ^P, is lower than the error being approximated ϵ^N, τ^N or $\hat{\tau}^N$, which is true for spectral methods in the asymptotic rate of convergence. If the asymptotic rate of convergence has not been reached, the sensors will indicate where the discretisation

error in the fine mesh, ϵ^P, is high. This analysis also explains why the coarse to fine estimation ($P < N$) results in inaccurate estimations. In that case, ϵ^P, the error made in the estimation will be usually higher than quantity being estimated ϵ^N, τ^N or $\hat{\tau}^N$.

As far as the iteration error ϵ^P_{it} is concerned, it has a first order or second order effect depending if the second term in the RHS of (9) and (13) is computed or if it is not, respectively. However, this source of inaccuracy can be reduced up to roundoff by converging the solution in the fine mesh $\tilde{u}^P \to u^P$ so $\epsilon^P_{it} \to 0$.

- Computational cost:

  It was previously stated that the estimation methods considered are *a posteriori*. That means that a solution, u^P should be available to perform the estimation. It is shown in [42] that the error estimation cost is negligible compared to the cost of converging the solution, u^P. Although the aim of this paper is to compare the efficiency of the sensors to locate the zones that require refinement, some methods to reduce the computational cost of the *a posteriori* approach are reviewed here.

  A "correction term" is introduced in (9) and (13) that permits the use of a non converged solution $\tilde{u}^P$. This is also known as quasi-*a priori* τ-estimation. Although the computation of the extra term involves solving a linear system of equations of dimension the number of DoF

of the problem, it is shown in [42] that it results in an overall reduction of the cost of the method. It should be noticed that the adaptation algorithms that make use of this "correction term" are optimised such that the linear system of equations is only solved once, even if the errors are estimated in several coarse (with lower polynomial order, $N < P$) meshes [26]. Finally, the computational cost of the truncation error estimation can be further reduced by extrapolating the estimations for polynomial orders higher than P. This enables the use of a lower polynomial order for the fine mesh and it is accurate if the asymptotic rate of convergence has been reached. For a more detailed analysis of the cost reduction strategies, the interested reader is referred to authors' paper [26].

In summary, *a priori* and *a posteriori* methods provide very similar estimates in terms of accuracy whilst reductions in computational cost are noticeable for *a priori* methods. In this work we focus on accuracy and compare the required number of degrees of freedom for each method. Henceforth, we do not consider the *a priori* approach (since it provides very similar results to the *a posteriori*) and do not compare computational costs.

3.2. Adaptation Algorithms

The common philosophy of all the algorithms presented is to equidistribute the error selected as refinement criteria, i.e. discretisation error, truncation error and isolated truncation error. All start with the computa-

tion of a solution with a high order polynomial P to then use this solution to compute the criteria in all the coarser meshes $N < P$. The elements in the adapted mesh use the lowest possible polynomial order that satisfies the required refinement criteria. Finally the high order polynomial solution P is interpolated into the new mesh and the simulation restarted.

In what follows, we first detail the discretisation error based estimation. Subsequently, the isotropic and anisotropic approaches for the truncation error and isolated truncation error based are introduced.

Discretisation Error Adaptation Algorithm. The discretisation error based adaptation algorithm uses the estimate (8) that is based on the spectral coefficients. The complete algorithm is detailed in Algorithm (1) and is performed for each mesh element. In part A, the simulation is converged until steady state and the coefficients a_k, and ϵ_P^N estimated. In part B, a new polynomial order is chosen such that the discretisation error estimation ϵ_P^N is below the desired threshold $\epsilon_{required}$. Having selected the polynomial that fulfills the threshold, the mesh is adapted by refining or coarsening the necessary element. In part C, the solution is interpolated into the new adapted mesh and the simulation is restarted and continued until convergence.

Isotropic Truncation / Isolated Truncation Error Adaptation Algorithm. The isotropic truncation error based adaptation algorithm relies on the estimate (9) that is based on the τ-estimation method. The complete algorithm is detailed in Algorithm 2. In part A, the simulation is converged

Part A - Estimation
Integrate in time on the fine mesh P until steady state;
for $N < P$ **do**
    Calculate the coefficient a_k;
    Estimate ϵ_P^N;
end
Part B - Adaptation
for $N < P$ **do**
    if $\epsilon_P^N \leq \epsilon_{required}$ **then**
        $P_x^{new} = N$;
        $P_y^{new} = N$;
        exit;
    end
end
Part C - Simulation
Interpolate converged solution to new adapted mesh;
Continue the simulation;

Algorithm 1: Discretisation error (feature based) algorithm on each mesh element

until steady state and the truncation error is estimated for all coarser meshes $N < P$. In part B, the new polynomial order is selected. If the estimation fulfills the requirement $\|\tau_P^{(N,N)}\|_{L_\infty}$ for a certain polynomial order, then this polynomial order is set for the new adapted mesh. In part C, the new solution is interpolated into the new mesh and the simulation restarted.

The algorithm can be modified to use the isolated truncation error by using (13) instead of (9).

Anisotropic Truncation / Isolated Truncation Error Adaptation Algorithm. The anisotropic truncation error based adaptation algorithm uses the estimate (9) that is based on the τ-estimation method. The com-

Part A - Estimation
Integrate in time on the fine mesh P until steady state;
for $N < P$ **do**
  Estimate the truncation error $\tau_P^{(N,N)}$;
end
Part B - Adaptation
for $N < P$ **do**
  if $\|\tau_P^{(N,N)}\|_{L_\infty} \leq required$ **then**
    $P_x^{new} = P_y^{new} = N$;
  end
end
Part C - Simulation
Interpolate converged solution to new adapted mesh;
Continue the simulation;

Algorithm 2: Isotropic τ-estimation adaptation

plete procedure is detailed in Algorithm 3. In part A, the simulation is converged until steady state and the truncation error is estimated for all coarser meshes ($N_x < P, N_y < P$). In part B, the new polynomial order is chosen such that the error threshold is satisfied using a minimum number of local degrees of freedom (LDoF), for each mesh element. In part C, the solution is interpolated into the new mesh and the simulation restarted.

The algorithm can be used for the isolated truncation error by using (13) instead of (9).

To clarify the adaptation procedure, we introduce an example for purely illustrative purposes of an isolated element mesh in Figure 1. Selecting $P = 8$ as the reference, we calculate the truncation error $\tau_P^{(N_x,N_y)}$ using lower polynomials ($N_x = 1, ..., 7$ $N_y = 1, ..., 7$) in each direction (showed in contour colors in the left figure). In this case, a required truncation error of $required = 10^{-2}$

Part A - Estimation
Integrate in time on the fine mesh P until steady state;
for $N_x < P$ **do**
  for $N_y < P$ **do**
    Estimate the truncation error $\tau_P^{(N_x,N_y)}$;
  end
end
Part B - Adaptation
$LDoF_{ref} = (P+1)(P+1)$;
for $N_x < P$ **do**
  for $N_y < P$ **do**
    $DoF_{new} = (N_x+1)(N_y+1)$
    if $\|\tau_P^{(N_x,N_y)}\|_{L_\infty} \leq required\ and\ \ DoF_{new} \leq LDoF_{ref}$ **then**
      $P_x^{new} = N_x$;
      $P_y^{new} = N_y$;
      $LDoF_{ref} = (P_x^{new}+1)(P_y^{new}+1)$;
    end
  end
end
Part C - Simulation
Interpolate converged solution to new adapted mesh;
Continue the simulation;

Algorithm 3: Anisotropic τ-estimation adaptation for each mesh element

cannot be achieved by polynomial combinations of low order (eg. red, yellow and light green areas). However, the error is below the required threshold in the upper right (blue) area.

Having identified the combinations (N_x, N_y) that fulfill the threshold criterion, the combination that requires the lowest number of LDoF is retained. Figure 1 (right) shows the number of LDoF for each combination. It can be seen that $N_x = 4$ and $N_y = 5$ guarantees the required threshold error, whilst minimising the local number of degrees of freedom to LDoF=30.

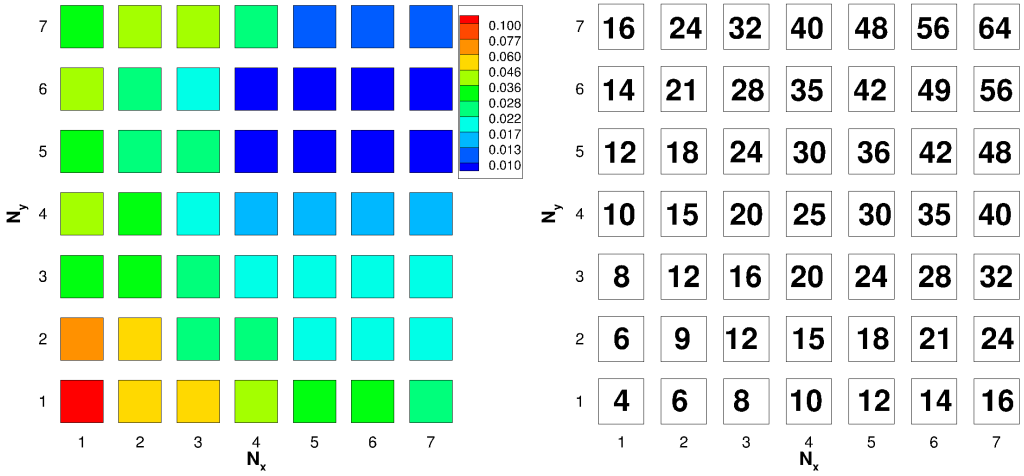

Figure 1: Truncation error estimation (in log scale) for different combinations of polynomial orders (left) and corresponding LDoF (right).

4. Numerical Results

The presented adaptation algorithms are applied to two test cases. First we analyse results for an inviscid NACA0012 airfoil (i.e. Euler equations), followed by a Navier-Stokes flat plate boundary layer simulation.

4.1. Inviscid NACA0012 airfoil

A NACA0012 airfoil is simulated at an angle of attack $\alpha = 0°$ using the Euler equations and a Mach number $M = 0.3$. We set slip wall boundary conditions at all airfoil surfaces and start with meshes that are not symmetric, see Figure 2. For unsymmetric meshes, the lift coefficient is not necessarily zero, but since we are considering a symmetric airfoil at zero angle of attack, the numerical lift should tend towards zero as the mesh is refined. This test case determines the advantages and limitations of the various adaptation

strategies proposed. The NACA case is converged in time until the maximum norm of the residual equation fall below 10^{-8} using a uniform polynomial order P=8.

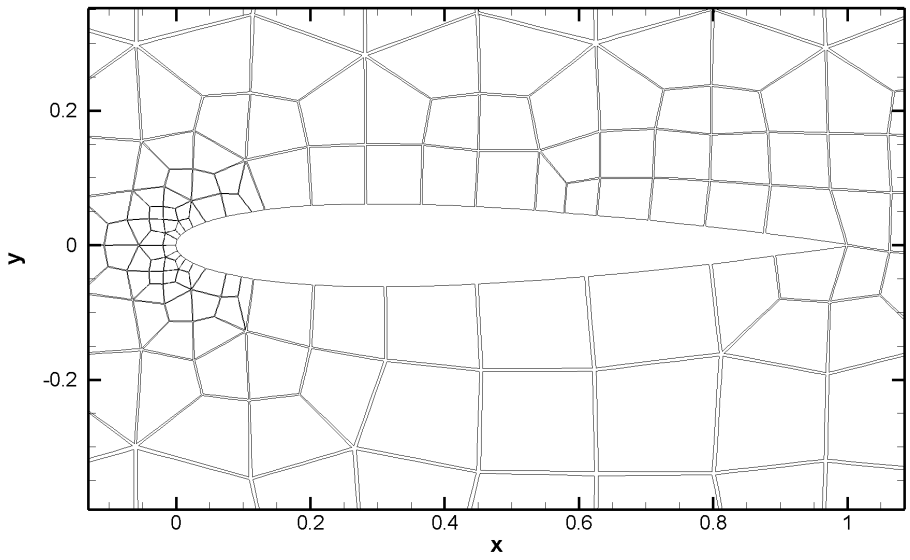

Figure 2: Initial NACA0012 mesh used for the adaptation.

First, Figure 3 shows the adapted meshes (nodal Legendre-Gauss points) obtained when using the discretisation error based adaptation approach (or "feature based") with thresholds 10^{-3} (top) and 10^{-4} (bottom). It can be seen that the discretisation error provides a very fine mesh at the leading edge whilst maintaining a relatively coarse resolution at the trailing edge.

Second, the adapted meshes resulting from using the truncation error adaptations are shown in Figure 4. The left figure shows the adaptation result when using the truncation error while the right figures show the results

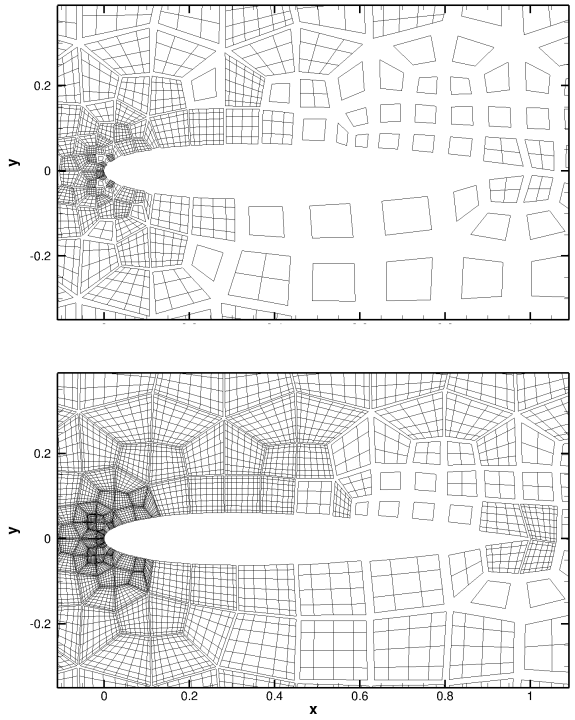

Figure 3: Adapted meshes (polynomial distributions) using the discretisation error ("feature based") strategy with thresholds 10^{-2} (top) and 10^{-3} (bottom). The computation Legendre-Gauss nodes are shown in each element.

based on the isolated truncation error for a threshold 10^{-4} (top) and 10^{-5} (bottom). It can be seen that both truncation error based adaptation strategies (non-isolated and isolated) provide meshes with a high concentration of degrees of freedom near the leading and the trailing edge. In Figure 5 the three refinement criteria are compared using thresholds that provide similar numbers of DoF in the adapted meshes. It can be noticed that the discretisation error based adaptation provides a coarser mesh near the trailing edge whilst similar meshes are obtained near the leading edge.

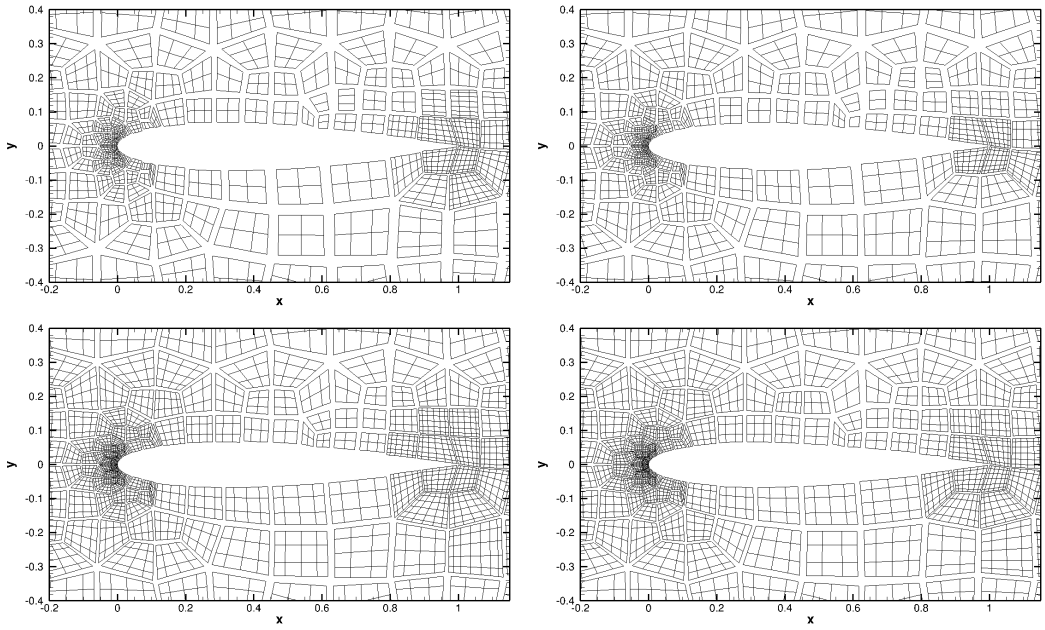

Figure 4: Adapted meshes (polynomial distributions) based on truncation error, TE (left) and isolated truncation error, ITE (right) for thresholds 10^{-4} (top) and 10^{-5} (bottom). The computation Legendre-Gauss nodes are shown in each element.

Figure 6 summarise lift and drag coefficients against the global number of degrees of freedom (DoF) using the various adaptation strategies. The adaptation thresholds are maintained for all methods to 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5}. We also include uniformly adapted meshes for reference. The figure shows convergence for the lift (left) and drag (right) in all cases once the asymptotic behaviour is reached (i.e. small enough required thresholds).

Firstly, comparing the discretisation and truncation error adaptations show that the truncation error requires fewer number of DoF to attain similar accuracy in the selected output functionals. Secondly, comparing truncation error to isolated truncation error adaptation shows that the isolated

version requires fewer DoF that the non-isolated version. Thirdly, comparing isotropic to anisotropic adaptation strategies (for both truncation and isolated truncation) shows a clear reduction of overall number of degrees of freedom to attain similar levels of accuracy.

In summary, for a similar level of accuracy, i.e. $\left|\frac{C_D^{Adapted}-C_D^{P=8}}{C_D^{P=8}}\right|< 1$, the following results are obtained. For the discretisation error based adaptation a 48% reduction of DoF compared to the uniform refinement. For the truncation error based adaptation a 70% reduction of DoF compared to the uniform refinement and a 10% reduction of the anisotropic truncation error compared to the isotropic approach. For the isolated truncation error based adaptation a 84% reduction of DoF compared to the uniform refinement and a 16% reduction of the anisotropic isolated truncation error compared to the isotropic approach.

This test case shows that the anisotropic isolated truncation error criteria outperforms the other criteria considered for adaptation.

Finally, according to Wang *et al.* [50], who summarised results from various high order solvers for a NACA0012 at $\alpha = 2°$, the high-order asymptotic convergence cannot be attained when using uniform refinement at the trailing edge of the airfoil. Indeed, the airfoil trailing edge results in a flow discontinuity that limits the solution smoothness and hence the exponential convergence when enriching the polynomial solution space. The truncation error has been proven to scale well for cases in the asymptotic range (see Rubio *et al.* [42]) while limitations have been observed in problems with sin-

gularities. In cases with discontinuities, h-refinement is preferred to enhance the solution accuracy. Nonetheless, the adaptation strategies based on the truncation errors have shown to perform well even for this challenging case.

4.2. Flat plate boundary layer

This section compares all presented adaptation strategies for a viscous Navier-Stokes simulation. We select a flat plate boundary layer case at Reynolds number per unit length Re=500 and Mach number M=0.2. The length of the plate is $L_x = 6$ and the leading edge singularity is included in the simulation (and is located at the point $(x, y) = (4, 0)$). The finest simulation used a uniform polynomial order $P = 8$ and was converged until the equation residual fell below 10^{-10}.

Using the discretisation error based adaptation method results in the meshes shown in Figure 7. For this criterion the thresholds 10^{-1}, 10^{-2}, 10^{-3} are depicted. Besides a high refinement around the singularity, which is common to all the schemes, it can be seen that this approach also refines the far field.

Applying the adaptation based on the truncation error and the isolated truncation error results in the meshes depicted in Figure 8. Here, we use the same thresholds as before (10^{-1}, 10^{-2}, 10^{-3}) for the adaptation. The truncation error shows a higher refinement around the singularity in comparison to the isolated truncation error. This leads to fewer overall number of DoF for the isolated truncation error case than for the truncation error

when the same threshold is considered. Both truncation error based adaptations provide meshes with fewer DoF than provided by the discretisation error (feature based) approach is used.

Finally, we depict in Figure 9 the drag coefficient C_D provided by the adapted meshes and compare the results to the solutions using uniform polynomials. Let us note that the maximum polynomial order for all adapted meshes is set to $P^{new} = 8$, such that this can be considered as the optimal solution.

The approaches based on the truncation error achieve similar drag coefficients with fewer global number of DoF than when using uniform adaptation or the discretisation error ("feature based") strategy. When using the discretisation error approach, the refinement in the far field (see Figure 7) results in a large number of DoF that are not required to improve the accuracy in the drag coefficient, which leads to an oversized mesh. This confirms that the truncation error acts as a better criteria to identify regions for refinement than the discretisation error. Indeed, the discretisation error method refines elements in the far field that have limited influence on the drag coefficient.

The anisotropic character of the boundary layer solution provides anisotropic meshes that significantly reduce the number of DoF whilst maintaining good a level of accuracy.

In summary, for a similar level of accuracy, i.e. $\left|\frac{C_D^{Adapted} - C_D^{P=8}}{C_D^{P=8}}\right| < 0.001$, the following results are obtained. For the discretisation error based adaptation a 25% reduction of DoF compared to the uniform refinement. For the

truncation error based adaptation a 84% reduction of DoF compared to the uniform refinement and a 27% reduction of the anisotropic truncation error compared to the isotropic approach. For the isolated truncation error based adaptation a 85% reduction of DoF compared to the uniform refinement and a 16% reduction of the anisotropic isolated truncation error compared to the isotropic approach.

The previous boundary layer results used a mesh with elements aligned with the flow direction, which lead to clearly dominant unidirectional adaptations. To further assess the advantages and limitations of the anisotropic approaches, a 45 degrees tilted mesh is selected, see Figure 10 (top shows the anisotropic adaptation for the truncation error and bottom for the isolated truncation error). Tilting the entire mesh by 45 degrees results in a mesh with many DoFs. In particular, the tilted isotropic adapted meshes have 29% and 40% extra degrees of freedom compared to their aligned with the flow isotropic adapted counterparts. However, the tilted anisotropic adapted meshes have 33% and 48% extra degrees of freedom compared to their aligned with the flow anisotropic adapted counterparts. This result means that the alignment worsens more the anisotropic refinement than to the isotropic refinement, as expected. It should be noticed that, although the results are not as good as for the aligned with the flow mesh, the anisotropic adaption is still preserved when the mesh is not aligned with the flow.

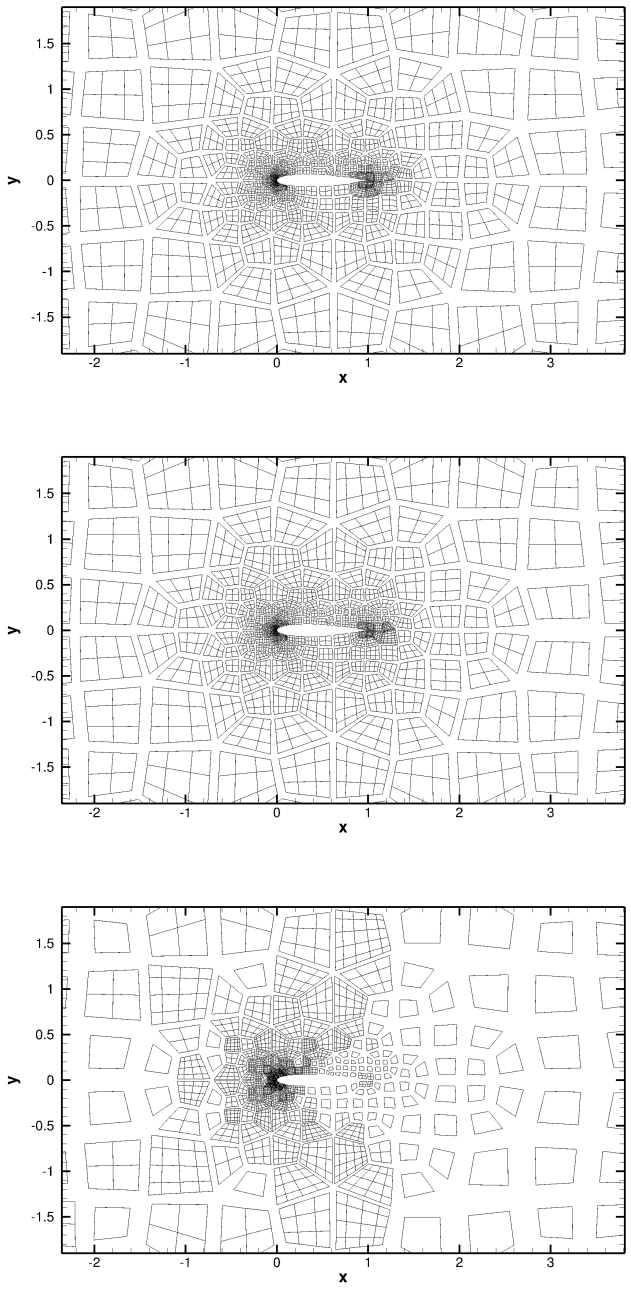

Figure 5: Adapted meshes (polynomial distributions) based on different refinement criteria with comparable final number of DoF. Top is anisotropic truncation error based (10^{-4} threshold, 20480 DoF). Middle is anisotropic isolated truncation error based (10^{-5} threshold, 21444 DoF). Bottom is isotropic discretisation error based (10^{-2} threshold, 19096 DoF). The computation Legendre-Gauss nodes are shown in each element.

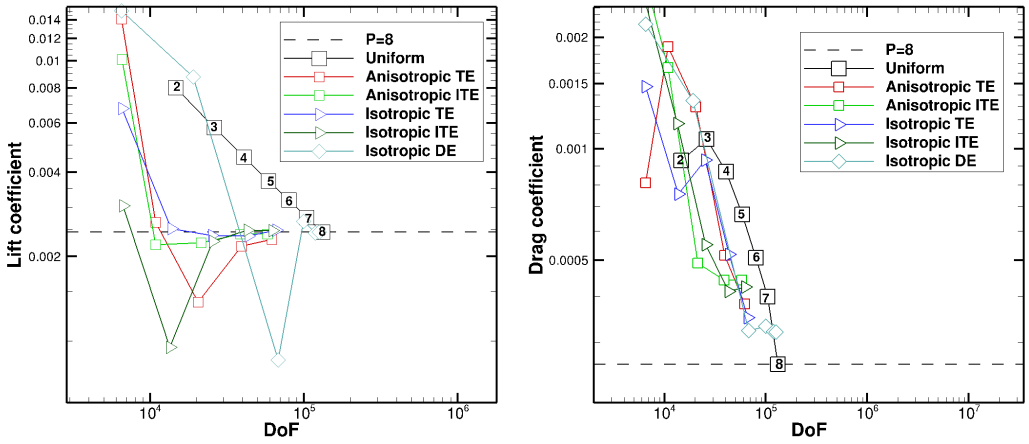

Figure 6: Lift and drag coefficients (pressure components only) for NACA0012 based on different adaptation algorithms. Uniform refinement, Discretisation error (DE), Truncation error (TE) and Isolated Truncation error (ITE) strategies are shown.

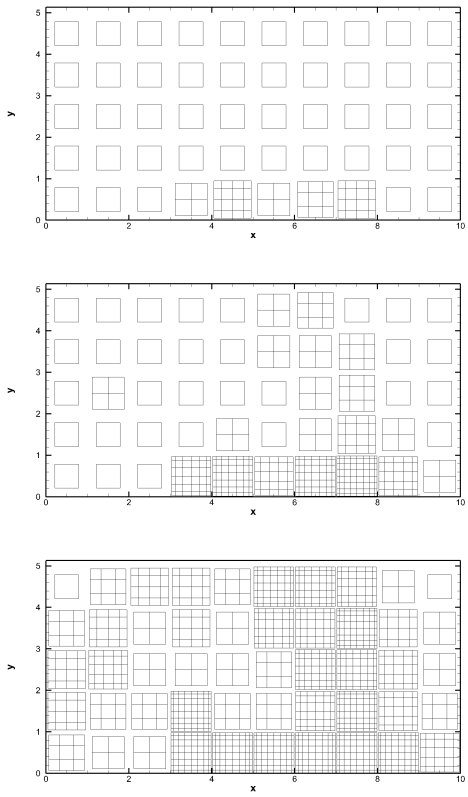

Figure 7: Adapted meshes (polynomial distributions) based on discretisation error approach for the, adaptation criteria 10^{-1} (top), 10^{-2} (middle), 10^{-3} (bottom). The computation Legendre-Gauss nodes are shown in each element. The computation Legendre-Gauss nodes are shown in each element. The computation Legendre-Gauss nodes are shown in each element.

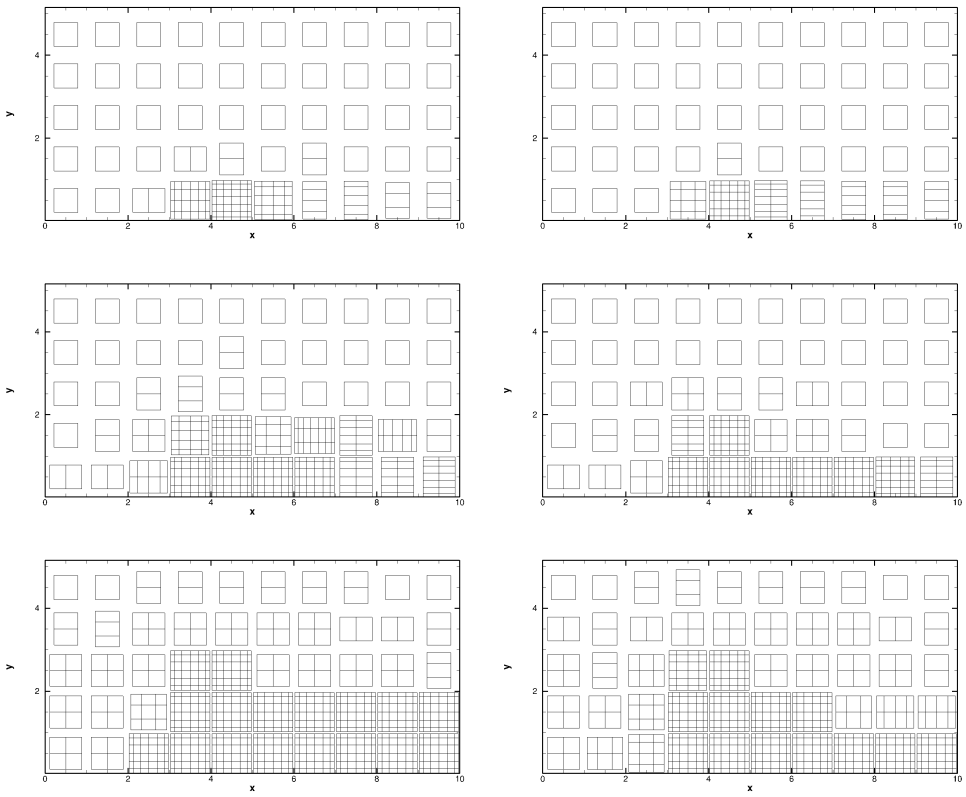

Figure 8: Adapted meshes (polynomial distributions) based on the truncation error (left) and the isolated truncation error (right), adaptation criteria 10^{-1} (top), 10^{-2} (middle), 10^{-3} (bottom). The computation Legendre-Gauss nodes are shown in each element.

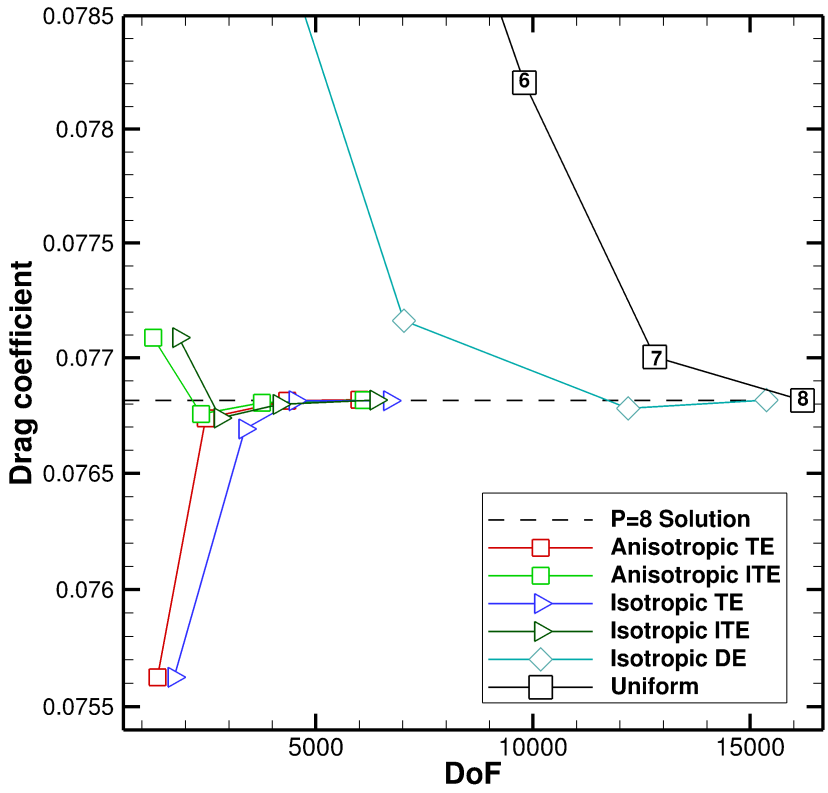

Figure 9: Drag coefficient C_D (including pressure and viscous components) and DoF based on different adaptation strategies. Uniform refinement, Discretisation error (DE), Truncation error (TE) and Isolated Truncation error (ITE) strategies are shown.

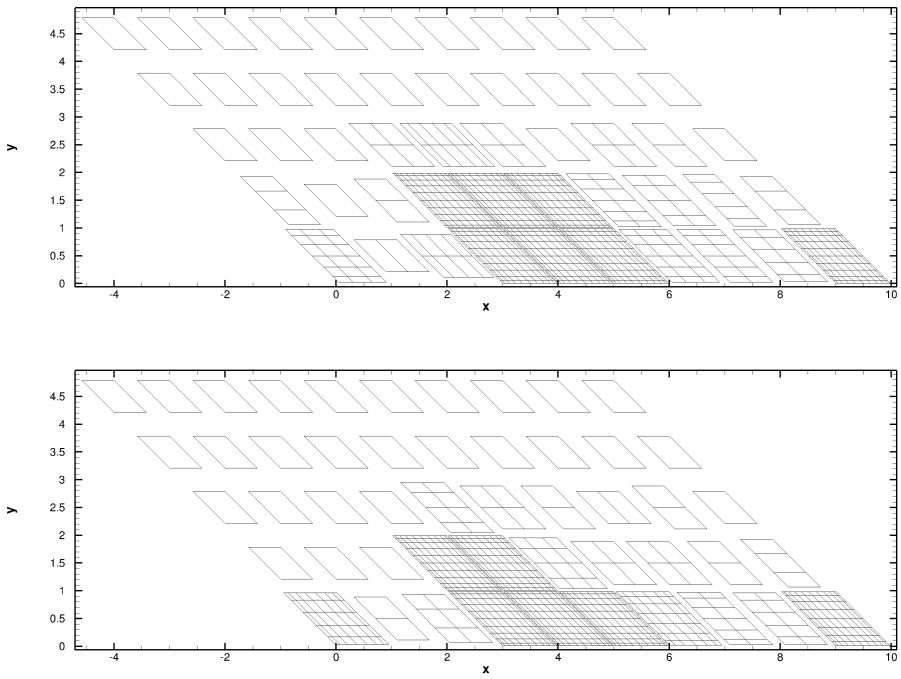

Figure 10: Adaptation for the boundary layer simulation for a threshold of 10^{-2} and Re = 100. Top: Truncation Error (TE) based Bottom: Isolated Truncation Error (ITE) based.

5. Conclusions

This work presents comparisons of local p-adaptation strategies for high order discontinuous Galerkin methods. The regions for adaptation (refinement or coarsening) are selected via three different procedures. A discretisation error sensor, also called "feature based", that quantifies the decay of the energy associated to the different modes is compared to two variants of truncation error approaches. The first truncation error approach relies on the estimation of the truncation error based on all terms appearing in the discrete discontinuous Galerkin variational formulation, whilst the second (the isolated truncation error) only takes into account volume terms.

These adaptation strategies are compared for an inviscid symmetric NACA0012 airfoil and a viscous flat plate boundary layer. The results show that truncation error based strategies require fewer degrees of freedom (approximately 80% less than uniform refinement), to attain similar accuracy, than the discretisation error approach (approximately 40% less than uniform refinement).

Furthermore, it is shown that flow regions with high discretisation errors do not necessarily require high resolution to increases the overall accuracy (defined in terms of output functionals such as lift and drag). This suggests that refinement in these areas may be un-necessary. The truncation error is identified as a better criteria for mesh adaptation to improve the overall accuracy of certain output values e.g. lift and drag.

Finally, anisotropic adaptations based on the truncation error outperform isotropic versions (approximately 15% fewer degrees of freedom than isotropic

versions) specially when considering flows with preferential directions such as boundary layers.

Appendix

Proof of the quasi-a priori τ-estimation formula, (9)

Substituting the definitions of the iteration error, $\tilde{u}^P = u^P - \epsilon^P_{it}$, and the discretization error, $u^P = u - \epsilon^P$, onto the estimate of the truncation error (9) and, using Taylor series, we obtain

$$\tau^N_P = \mathcal{R}^N(u) - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P_{it} - \bar{I}^N_P \mathcal{R}^P\left(\tilde{u}^P\right) + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon^P_{it}\right)^2, \tag{17}$$

or equivalently

$$\tau^N_P = \tau^N - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P_{it} - \bar{I}^N_P \mathcal{R}^P\left(\tilde{u}^P\right) + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon^P_{it}\right)^2. \tag{18}$$

Again, using Taylor series and the definition of the iteration error,

$$\tau^N_P = \tau^N - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P_{it} + \bar{I}^N_P \left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{u^P} \epsilon^P_{it} + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon^P_{it}\right)^2. \tag{19}$$

Taking into account that, by definition $u - \tilde{u}^P = \epsilon^P + \epsilon^P_{it}$, it can be seen that for $\bar{I}^N_P = \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{\tilde{u}^P} I^N_P \left(\left.\frac{\partial \mathcal{R}^P}{\partial u^P}\right|_{\tilde{u}^P}\right)^{-1}$ we have

$$\tau^N_P = \tau^N - \left.\frac{\partial \mathcal{R}^N}{\partial u^N}\right|_{u^N} \epsilon^P + \mathcal{O}\left(\epsilon^P\right)^2 + \mathcal{O}\left(\epsilon^P_{it}\right)^2. \tag{20}$$

Quasi-a priori τ-estimation formula for the isolated truncation error, (13), can be proved following the same procedure.

Acknowledgements

The authors would like to acknowledge the European Commission for the financial support of the ANADE project (Advances in Numerical and Analytical tools for DEtached flow prediction) under grant contract PITN-GA-289428 and the NNATAC project (New Numerical and Analytical Tools for Aerodynamic flow Control) under grant agreement PIAP-GA-2012-324298. In addition, the authors acknowledge the computer resources and technical assistance provided by the Centro de Supercomputación y Visualización de Madrid (CeSViMa).

References

[1] M. Ainsworth. A posteriori error estimation for discontinuous Galerkin finite element approximation. *SIAM Journal on Numerical Analysis*, 45(4):1777–1798, 2007.

[2] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39(5):1749–1779, 2001.

[3] I. Barosan, P.D. Anderson, and H.E.H. Meijer. Application of mor-

tar elements to diffuse-interface methods. *Computers and Fluids*, 35(10):1384–1399, 2006.

[4] F. Bassi, A. Crivellini, D. A. Di Pietro, and S. Rebay. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 218(2):794 – 815, 2006.

[5] F. Bassi and 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(2):267 – 279, 1997.

[6] M. J. Berger. *Adaptive finite difference methods in fluid dynamics*. Courant Institute of Mathematical Sciences New York University, 1987.

[7] K. Bernert. τ-Extrapolation—Theoretical foundation, numerical experiment, and application to Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 18(2):460–478, 1997.

[8] C. G. Canuto, Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Scientific Computation. Springer, 2010.

[9] B. Cockburn, G. Kanschat, and D. Schötzau. An Equal-Order DG Method for the Incompressible Navier-Stokes Equations. *Journal of Scientific Computing*, 40(1):188–210, 2008.

[10] J. M. Derlaga, T. Phillips, C. J. Roy, and J. Borggaard. Adjoint and truncation error based adaptation for finite volume schemes with error estimates. In *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech. American Institute of Aeronautics and Astronautics, January 2015.

[11] R. P. Dwight. Heuristic a Posteriori Estimation of Error Due to Dissipation in Finite Volume Schemes and Application to Mesh Adaptation. *Journal of Computational Physics*, 227(5):2845–2863, 2008.

[12] E. Ferrer. *A high order Discontinuous Galerkin-Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes for simulating cross-flow turbines*. PhD thesis, University of Oxford, 2012.

[13] E. Ferrer, D. Moxey, R.H.J. Willden, and S. Sherwin. Stability of projection methods for incompressible flows using high order pressure-velocity pairs of same degree: Continuous and discontinuous Galerkin formulations. *Communications in Computational Physics*, 16(3):817–840, 2014.

[14] E. Ferrer and R.H.J. Willden. A high order discontinuous Galerkin finite element solver for the incompressible Navier–Stokes equations. *Computers & Fluids*, 46(1):224–230, 2011.

[15] E. Ferrer and R.H.J. Willden. A high order discontinuous Galerkin - Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes. *Journal of Computational Physics*, 231(21):7037–7056, 2012.

[16] F. Fraysse, J. de Vicente, and E. Valero. The estimation of truncation error by τ-estimation revisited. *Journal of Computational Physics*, 231(9):3457–3482, 2012.

[17] F. Fraysse, G. Rubio, J. de Vicente, and E. Valero. Quasi-a priori mesh adaptation and extrapolation to higher order using tau-estimation. *Aerospace Science and Technology*, 38(0):76 – 87, 2014.

[18] F. Fraysse, E. Valero, and J. Ponsin. Comparison of mesh adaptation using the adjoint methodology and truncation error estimates. *AIAA Journal*, 2012.

[19] F. Fraysse, E. Valero, and G. Rubio. Quasi-a priori truncation error estimation and higher order extrapolation for non-linear partial differential equations. *Journal of Computational Physics*, 253(0):389 – 404, 2013.

[20] S. R. Fulton. On the accuracy of multigrid truncation error estimates. *Electronic Transactions on Numerical Analysis*, 15:29–37, 2003.

[21] X. Gu and T. Shih. Differentiating between source and location of error for solution-adaptive mesh refinement. In *15th AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 2001.

[22] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation

and adaptive mesh refinement for the RANS and k-ω turbulence model equations. *J. Comput. Phys.*, 230(11):4268–4284, 2011.

[23] R. Hartmann, J. Held, T. Leicht, and F. Prill. Discontinuous galerkin methods for computational aerodynamics 3D adaptive flow simulation with the dlr padge code. *Aerospace Science and Technology*, 14(7):512 – 519, 2010.

[24] R. Hartmann and T. Leicht. Generalized adjoint consistent treatment of wall boundary conditions for compressible flows. *J. Comput. Phys.*, 300:754–778, 2015.

[25] G. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods for CFD*. Numerical mathematics and scientific computation. Oxford University Press, 1999.

[26] M. Kompenhans, G. Rubio, E. Ferrer, and E. Valero. Adaptation strategies for high order discontinuous Galerkin methods based on Tau-estimation. *Journal of Computational Physics*, 306:216–236, 2015.

[27] D. A. Kopriva. Multidomain spectral solution of compressible viscous flows. *Journal of Computational Physics*, 115(1):184 – 199, 1994.

[28] D. A. Kopriva. *Implementing spectral methods for partial differential equations: algorithms for scientists and engineers*. Springer, 1st edition, 2009.

[29] B. Landmann, M. Kessler, S. Wagner, and E. Krämer. A parallel high-order discontinuous Galerkin code for laminar and turbulent flows. *Computers & Fluids*, 37(4):427 – 438, 2008.

[30] C. Mavriplis. A posteriori error estimators for adaptive spectral element techniques. In Pieter Wesseling, editor, *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics*, volume 29 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 333–342. Vieweg+Teubner Verlag, 1990.

[31] C. Mavriplis. Adaptive mesh strategies for the spectral element method. *Computer Methods in Applied Mechanics and Engineering*, 116(14):77 – 86, 1994.

[32] N. C. Nguyen, P. O. Persson, and J. Peraire. RANS solutions using high order discontinuous Galerkin methods. *45th AIAA Aerospace Science Meeting and Exhibit, Reno, Nevada*, 2007.

[33] T. A. Oliver and D. L. Darmofal. An unsteady adaptation algorithm for discontinuous Galerkin discretizations of the RANS equations. *18th AIAA Computational Fluid Dynamics Conference, Reno, Nevada*, 2007.

[34] P. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. *AIAA Aerospace Sciences Meeting and Exhibit*, 44th, 2006.

[35] T. S Phillips and C. J Roy. Residual methods for discretization error estimation. *AIAA Paper*, 3870:27–30, 2011.

[36] H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[37] B. Riviere. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[38] C. J. Roy. Strategies for driving mesh adaptation in CFD, AIAA 2009-1302, invited paper for session on Error Estimation and Control, 47th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 5-8, 2009.

[39] C. J. Roy. Review of discretization error estimators in scientific computing. *AIAA Paper*, 126, 2010.

[40] G. Rubio. *Truncation error estimation in the Discontinuous Galerkin Spectral Element Method*. PhD thesis, Universidad Politecnica de Madrid, 2015.

[41] G. Rubio, F. Fraysse, J. de Vicente, and E. Valero. The estimation of truncation error by τ-estimation for chebyshev spectral collocation method. *Journal of Scientific Computing*, 57(1):146–173, 2013.

[42] G. Rubio, F. Fraysse, D. A. Kopriva, and E. Valero. Quasi-a priori trun-

cation error estimation in the DGSEM. *Journal of Scientific Computing*, pages 1–31, 2014.

[43] K. Shahbazi, P. F. Fischer, and C. R. Ethier. A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics*, 222(1):391 – 407, 2007.

[44] T. I. P. Shih and B. R. Williams. Development and evaluation of an a posteriori method for estimating and correcting grid-induced errors in solutions of the Navier-Stokes equations. *AIAA Paper*, 1499:2009, 2009.

[45] A. Syrakos, G. Efthimiou, J. G. Bartzis, and A. Goulas. Numerical experiments on the efficiency of local grid refinement based on truncation error estimates. *Journal of Computational Physics*, 231(20):6725 – 6753, 2012.

[46] A. Syrakos and A. Goulas. Finite volume adaptive solutions using SIMPLE as smoother. *International Journal for Numerical Methods in Fluids*, 52(11):1215–1245, 2006.

[47] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer, Berlin, New York, 1997.

[48] J. J. W. van der Vegt and H. van der Ven. Space time discontinuous galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. general formulation. *Journal of Computational Physics*, 182(2):546 – 585, 2002.

[49] L. Wang and D. J. Mavriplis. Adjoint-based h-p adaptive discontinuous galerkin methods for the 2D compressible euler equations. *Journal of Computational Physics*, 228(20):7643 – 7661, 2009.

[50] Z.J. Wang, K. 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, and M. Visbal. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[51] Gary P Warren, W Kyle Anderson, James L Thomas, and Sherrie L Krist. Grid convergence for adaptive methods. *AIAA paper*, 1592:1991, 1991.