Quasi-*a priori* mesh adaptation and extrapolation to higher order using τ-estimation

F. Fraysse, G. Rubio, J. de Vicente and E. Valero

Universidad Politécnica de Madrid, E.T.S.I.Aeronáuticos, Madrid, Spain

Abstract

Improving the balance between cost and accuracy of computational fluid dynamics solvers by local mesh adaptation has become a topic of increasing interest. Numerical error based adaptation sensors proved to be robust and converge faster than sensors simply based on features of the flow field. However, this family of sensors known as residual-based sensors usually suffer from a high computational cost, e.g. the adjoint or Richardson extrapolation and are *a posteriori* indicators, which thus implies an initial solution to the problem. In this paper, we develop a mesh adaptation indicator accurate in the early stages of the iteration procedure, based on the τ-estimation technique. τ-estimation allows for an accurate truncation error estimation and can therefore be used to correct the discretized equations through the so-called τ-extrapolation methodology. We introduce the τ-estimation technique for non-converged solutions and derive all the necessary conditions to be used in the context of mesh adaptation and extrapolation of practical problems. Then, we apply this methodology to two-dimensional test cases, Euler flow past a NACA0012 airfoil in different Mach regimes, with a special focus on computational time. We demonstrate that quasi-*a priori* τ-estimation can substantially reduce the cost when combined with mesh adaptation or τ-extrapolation in comparison to classic adaptation sensors.

Keywords: mesh adaptation, truncation error, error estimation

1. Introduction

In the past decades, due to the increasing demand for complex fluid flow simulations, great effort has been done by the computational fluid dynamics

(CFD) community in order to increase the accuracy and reduce the calculation costs. It is now well understood that numerical errors play a crucial role in the balance between accuracy and computational time.

In the finite volume community, it is well known that accuracy can be easily gained by using a denser discrete mesh or by using a higher order numerical scheme as the discretization errors are of the order $\mathcal{O}(h^p)$, where h is a characteristic mesh length and p is the formal order of the scheme.

The simplest refinement anyone can think of is to divide all cells in the domain. This is referred to as uniform refinement. Although it does improve the solution vastly, it is easy to realise that a huge unwanted effort is required in doing so. For example, in the far-field region of an airfoil, cell division does not usually bring much improvement.

One great advantage of an unstructured approach to solve partial differential equations is that it provides more flexibility in the generation of the mesh, especially for complex configurations involving for instance multi-element airfoils. Unstructured meshes are also more suited in relocating nodes or (de-)refining regions.

These methods used to increase the accuracy are often referred to as r-, h- and p-refinement, and can be combined for example as hr- or hp-refinement, the latter being more often employed in the finite element community.

r-refinement, which consists in moving nodes by keeping the overall number constant, has been considered in the unstructured community [1, 2, 3, 4], however the unstructured approach allows for more tuning at a reasonable cost. Dynamic remeshing consists in generating new independent meshes, during the solution process, taking into account that the generation of unstructured meshes using a Delaunay-type algorithm is very efficient. An important contribution is due to Mavriplis [5], using an adaptive remeshing strategy for Euler two-dimensional flows with a multigrid strategy. In this approach, a coarse unstructured mesh is first generated, fine enough to capture flow properties, and regions where high gradients occur. Then a new finer mesh is created using Delaunay's triangulation and smoothed by a Laplacian operator. New meshes are then created until the desired level of accuracy is reached. This method naturally builds a sequence of meshes which can then be used in a full multigrid algorithm. Remeshing an unstructured mesh is much less demanding than the equivalent in a structured approach, and the smoothing step allows a sequence of good quality meshes ensuring optimal accuracy, but still require substantial computational time especially for complex geometries.

Another common approach in the unstructured community is called local refinement (h-refinement), where the topology of the initial mesh is kept, which obviates the use of high order transfer operators. Local refinement has been widely studied and many articles can be found in the literature. Parthasarathy and Kallinderis gave a great contribution to local refinement/derefinement techniques [7, 6] where they present adaptation methodologies for three-dimensional meshes and applied them to an Euler transonic test-case, showing good improvement of the accuracy. The algorithm is implemented in tandem with multigrid: coarse mesh levels are naturally generated by the adaptation procedure, and each time the mesh is adaptively refined, an additional level is added to the multigrid system. Later [8], three-dimensional viscous calculations have been performed using semi-unstructured prismatic meshes: an unstructured tessellation is used along the surface whereas a structured clustering is performed along the wall normal to accurately capture turbulence features. In this approach local refinement is performed only in the lateral direction, along the body surface, whereas only displacement of the nodes is considered in the normal direction, thus satisfying minimum spacings imposed by the turbulence model and simplifying the algorithm with respect to a fully three-dimensional adaptive approach. This approach has also been chosen by Braaten and Connell [9] for the three-dimensional Navier-Stokes equations.

The selection of a reliable adaptation parameter is a key aspect in order to reduce the errors in the computation. Classical methods such as feature-based compute a sensor based on the flow field variables [11, 10, 14]. However, their lack of mathematical foundation makes it difficult to apply them in a user-independent way to problems that differ greatly from established experience.Therefore, in the last years special attention has been paid to indicators based on numerical errors, such like the family of residual-based adaptation indicators or the goal-oriented adjoint methodology [12, 13].

One of the main drawbacks of these sensors is that they are *a posteriori* estimators, in the sense that a solution to the flow equations has to be computed first. In this work, we propose to extend previous work of Fraysse *et al.* [16, 17, 18] on the application of τ-estimation [15] to mesh adaptation and higher order extrapolation using non-converged solutions to steady-state Euler equations. The outline of this paper is as follows: in Section 2, the quasi-*a priori* truncation error estimator is introduced as well as all the necessary conditions to get optimal accuracy. In Section 3, we present some implementation details to practical problems. Finally in Section 4, we apply the dynamic τ-estimation to mesh adaptation and τ-extrapolation on two-dimensional test cases.

2. Mathematical background

Let us consider the discretization of a partial differential equation (PDE), representing symbolically for instance the Euler equations, on a mesh Ω_h indexed by a mesh size parameter h of the following form:

$$\mathcal{R}(\boldsymbol{U}) = 0 \Rightarrow \mathcal{R}^h(\boldsymbol{U}^h) = 0 \tag{1}$$

where, $\boldsymbol{U}^h$ is the discrete solution (converged) of the PDE and $\mathcal{R}^h$ represents the discrete residual (the spatial discretization). The discretization error ϵ^h and the local truncation error τ^h corresponding to Eq. 1 are defined as follows:

$$\begin{aligned} \epsilon^h &= \mathcal{I}^h\boldsymbol{U} - \boldsymbol{U}^h \\ \tau^h &= \mathcal{R}^h(\mathcal{I}^h\boldsymbol{U}) \end{aligned} \tag{2}$$

where $\mathcal{I}^h$ represent a continuum-to-mesh transfer operator, e.g. simple injection.

Introducing a coarser mesh level denoted by Ω_H, with a mesh ratio of $\rho = h/H < 1$, the coarse mesh equation of a full approximation storage multigrid technique reads,

$$\mathcal{R}^H(\widehat{\boldsymbol{U}}^H) = \tau_h^H \quad \text{with } \widehat{\boldsymbol{U}}^H = \widehat{\mathcal{I}}_h^H(\epsilon_{it}^h + \widetilde{\boldsymbol{U}}^h) \tag{3}$$

$$\tau_h^H = \mathcal{R}^H(\widehat{\mathcal{I}}_h^H\widetilde{\boldsymbol{U}}^h) - \mathcal{I}_h^H(\mathcal{R}^h(\widetilde{\boldsymbol{U}}^h)) = T_1 + T_2 \tag{4}$$

In Equation 4, $\widehat{\mathcal{I}}_h^H$ and $\mathcal{I}_h^H$ stand for restriction operators (applied over the solution and the residual respectively) and $\widetilde{\boldsymbol{U}}^h$ is the current approximation of the solution on the fine mesh, such that it differs to the converged solution by the iteration error, $\epsilon_{it}^h = \boldsymbol{U}^h - \widetilde{\boldsymbol{U}}^h$.

As stated by Brandt [15], it is remarkable that τ_h^H is a truncation error estimation on the coarse mesh Ω_H. Our objective is to use τ_h^H to estimate τ^H in the early stages of the iteration procedure, e.g. when $\epsilon_{it}^h \neq 0$ thus $T_2 \neq 0$. If this estimation can be performed with sufficient accuracy far before the solution is converged, it can provide an inexpensive error estimator to be used in mesh adaptation or extrapolation to higher accuracy.

The following theorem provides the conditions to be fulfilled for an accurate truncation error estimation using τ_h^H.

Theorem 1 (Truncation Error Estimate). *Assume that there exists* $n, p, q, r, s \geq 1$ *such that if* $\boldsymbol{U} \in \mathcal{C}^{n+p+q}(\Omega)$*, the truncation error (2) satisfies:*

- ***(A1)*** *Local truncation error of order* p*:* $\tau^h = h^p \mathcal{I}^h \boldsymbol{V} + \mathcal{O}(h^{p+q})$*, with* $V \in \mathcal{C}^q(\Omega)$*,*
- ***(A2)*** *Local discretization error of order* r*:* $\epsilon^h = h^r \mathcal{I}^h \boldsymbol{W} + \mathcal{O}(h^{r+s})$*, with* $W \in \mathcal{C}^s(\Omega)$*,*
- ***(A3)*** *Fine to coarse transfer operator of the solution of order* t*:* $\widehat{\mathcal{I}}_h^H \mathcal{I}^h \boldsymbol{U} = \mathcal{I}^H \boldsymbol{U} + \mathcal{O}(h^t)$*, with* $\boldsymbol{U} \in \mathcal{C}^t(\Omega)$*,*
- ***(A4)*** *Fine to coarse transfer operator of the residual such that:*

$$\mathcal{I}_h^H = \mathcal{J}^H \widehat{\mathcal{I}}_h^H \mathcal{J}^{h^{-1}}$$

where $\mathcal{J}^H = \left.\dfrac{\partial \mathcal{R}^H}{\partial \boldsymbol{U}^H}\right|_{\mathcal{I}^H \boldsymbol{U}}$ *is the Jacobian matrix.*

Then,

$$\tau_h^H = (1 - \rho^p)\tau^H + \mathcal{O}(\max(h^{\min(t, r+s)}, ||\epsilon_{it}^h||^2)) \qquad (5)$$

Proof

Let us first decompose the current solution approximation as $\widetilde{\boldsymbol{U}}^h = \mathcal{I}^h \boldsymbol{U} - \epsilon^h - \epsilon_{it}^h$. Then the first term T_1 of Equation 4 becomes:

$$T_1 = \mathcal{R}^H(\widetilde{\mathcal{I}}_h^H(\mathcal{I}^h \boldsymbol{U} - \epsilon^h - \epsilon_{it}^h)) \qquad (6)$$

with assumption **(A3)** and linearizing about $\mathcal{I}^H \boldsymbol{U}$, Equation 6 yields:

$$T_1 = \mathcal{R}^H(\mathcal{I}^H \mathcal{I}^h \boldsymbol{U}) - \mathcal{J}^H \widehat{\mathcal{I}}_h^H \epsilon^h - \mathcal{J}^H \widehat{\mathcal{I}}_h^H \epsilon_{it}^h + \mathcal{O}(\max(h^{2r}, h^t, ||\epsilon_{it}^h||^2)) \qquad (7)$$

As far as T_2 in Equation 4 is concerned, by using the same decomposition as in Equation 6 and linearizing about $\mathcal{I}^h \boldsymbol{U}$, it becomes:

$$T_2 = \mathcal{I}_h^H(\mathcal{R}^h(\mathcal{I}^h\boldsymbol{U}) - \mathcal{J}^h\epsilon^h - \mathcal{J}^h\epsilon_{it}^h) + \mathcal{O}(\max(h^{2r}, h^t, ||\epsilon_{it}^h||^2)) \tag{8}$$

Finally, by using the discrete error transport equation $\tau^h = R^h(\boldsymbol{U}^h + \epsilon^h) = \mathcal{J}^h\epsilon^h + \mathcal{O}(h^{2r})$, the relation between fine and coarse discretization errors $\widehat{\mathcal{I}}_h^H\epsilon^h = \rho^r\epsilon^H + \mathcal{O}(h^{r+\min(s,t)})$ and assumption **(A4)**, we obtain

$$\tau_h^H = (1-\rho^p)\tau^H + \mathcal{O}(\max(h^{\min(t,r+s)}, ||\epsilon_{it}^h||^2))$$

□

With an accurate truncation error estimation in hand, a direct application is the so-called τ-extrapolation which consists in solving the following problem

$$\mathcal{R}^H(\bar{\boldsymbol{U}}^H) = \frac{1}{1-\rho^p}\tau_h^H \tag{9}$$

Then it can be demonstrated that,

$$\bar{\epsilon}^H = \mathcal{I}^H\boldsymbol{U} - \bar{\boldsymbol{U}}^H = \mathcal{O}(\max(h^{\min(t,r+s)}, ||\epsilon_{it}^h||^2)) \tag{10}$$

From Equation 5, it can be seen that an accurate truncation error estimation can be obtained in the first steps when solving iteratively a non-linear PDE at the following conditions:

- Grid Ω^H has to be representative of Ω^h, e.g. embedded,
- Grids Ω^h and Ω^H are fine enough that we are working in the asymptotic range,
- The restriction operator acting on the solution ($\widehat{\mathcal{I}}_h^H$) is at least one order higher than the formal order of accuracy p, $t > p$,
- The discretization error decreases equally of faster than the truncation error $r \geq p$,
- The Jacobian $\mathcal{J}^h$ has to be inverted to reach quadratic convergence of the quasi-*a priori* truncation error estimate towards its converged value.

Then, Equation 10 states that a higher order solution can be obtained by solving again the set of equations with an additional source term (truncation error estimation) on the right-hand-side. Based on the previous developments, we derive in Section 3 a quasi-*a priori* mesh adaptation and τ-extrapolation procedure.

3. Applications and implementation details

Mesh adaptation and extrapolation to higher order have been reported previously in Fraysse *et al.* [17, 18]. In the first journal paper, a mesh adaptation sensor based on truncation error has been developed and compared to the adjoint methodology and classic feature-based indicator for the Euler equations. The second one has been devoted to the extension of truncation error-based mesh adaptation to the set of Navier-Stokes equations as well as higher order extrapolation. However, both approaches have been performed *a posteriori*: first the solution was computed on the initial mesh and then mesh adaptation and/or τ-extrapolation was applied. In this section, we detail how to perform τ-estimation in the early stages of the iteration to convergence and apply it to mesh adaptation and higher order extrapolation.

The CFD code used in the present work is the DLR TAU-Code [19]. It solves the Reynolds Averaged Navier-Stokes equations on unstructured hybrid meshes by employing a second-order finite volume discretization. The multigrid strategy implemented in TAU uses the full approximation scheme algorithm to compute the correction term on the coarse meshes. The coarse meshes are obtained by agglomeration of the fine mesh dual cells. When the primary mesh is composed of quadrilaterals (or hexahedrons for 3D computations), then the advancing front method is capable of agglomerating four quadrilaterals (eight hexahedrons) to create a coarse quadrilateral (hexahedral), as in a structured solver. However, when the primary mesh is unstructured, the agglomeration algorithm creates coarse mesh elements that do not necessarily maintain the fine mesh characteristics. In the context of multigrid, this situation is not a strong limitation, particularly in TAU, where the coarse mesh fluxes are computed with a first-order accuracy. However, in the context of truncation error estimation, it is of importance that the truncation error is identical between fine and coarse meshes (see Fraysse *et al.* [16]). This goal clearly cannot be accomplished if the elements differ from fine to coarse mesh. To circumvent this issue and to obtain estimations of the truncation error for unstructured meshes, in the following analysis, the fine mesh (where the flow solution is actually computed) was obtained from the coarse mesh by bisecting all of the edges. This method also allows for the use of injection to restrict the solution from fine to coarse meshes, thus preventing the use of complex interpolations and assuring that no accuracy is lost ($t = \infty$).

The estimation of the truncation error is obtained by computing the following expression:

$$\frac{1}{1-\rho^p}\tau_h^H = \mathcal{R}^H(\widehat{\mathcal{I}}_h^H \widetilde{\boldsymbol{U}}^h) - \mathcal{J}^H \widehat{\mathcal{I}}_h^H \mathcal{J}^{h^{-1}} \mathcal{R}^h(\widetilde{\boldsymbol{U}}^h) = T_1 + T_2 \qquad (11)$$

In the current work, the Euler equations are solved using the second order Jameson-Schmidt-Turkel scheme with scalar dissipation, so that $p = 2$ and given that Ω^h is embedded in Ω^H by bisection of all the edges, $\rho = 0.5$.

From Equation 11, it can be seen that the first term T_1 is straightforward to compute as it just consists in evaluating the residual on the coarse mesh from the current fine mesh approximation of the solution. The second term on the other side requires the inversion of the Jacobian matrix $\mathcal{J}^h$, it is achieved in the following way:

- Solve the linear system: $-\mathcal{J}^h \phi_1^h = \mathcal{R}^h(\widetilde{\boldsymbol{U}}^h)$,
- Restrict $\widetilde{\boldsymbol{U}}$ and ϕ_1^h using $\widehat{\mathcal{I}}_h^H$,
- Multiply by coarse mesh Jacobian $T_2 = \mathcal{J}^H \widehat{\mathcal{I}}_h^H \phi_1^h$.

The Jacobian matrices are built analytically inside the DLR TAU-Code and the linear system is solved using the ILU(0)-GMRES from the PETSc libraries [20].

4. Results

In this section we introduce the results associated to the local mesh adaptation and extrapolation to higher order based on dynamic τ-estimation. Three two-dimensional inviscid NACA0012 test cases are analyzed, representative of several flow regimes. The far-field conditions considered here are, subsonic, transonic and supersonic (see Table 1 and Figure 1(b)-(d)). Results on these test cases have been reported previously in the context of feature-based [11] and adjoint-based [21, 17] mesh adaptation.

Table 1: Far field conditions

Test case	Mach number	Angle of attack α
Subsonic	0.4	5°
Transonic	0.95	0°
Supersonic	1.5	1°

The subsonic test case is obtained using a free-stream Mach number of $M_\infty = 0.4$ and an angle of attack of $\alpha = 5^\circ$. Mach number contours can be seen in Figure 1(b). The main feature of this test case is a zone where the flow accelerates on the suction side right after the stagnation point. The transonic flow free-stream Mach number is set to $M_\infty = 0.95$ and the angle of attack to $\alpha = 0^\circ$. Mach number contours can be seen in Figure 1(c): a fish-tailed oblique shock and a recompression in the wake yielding to another normal shock wave. Finally, the last test case is devoted to a supersonic flow. The free-stream Mach number is set to $M_\infty = 1.5$ and the angle of attack to $\alpha = 1^\circ$ (see Figure 1(d)). The main features of this test case are a bow shock upstream as well as a fish-tailed one. Adapted meshes from converged solutions both for the feature-based and TE-based indicator are shown in Figure 2.

We focus the analysis on adaptation and τ-extrapolation based on a non-converged solution. For details concerning the mesh adaptation algorithms, we refer the reader to Fraysse *et al.* [17]. Thus, for each test case, the flow solution is iterated to various values of solver tolerance (based on L_2 norms of two successive approximations), from 1 to 10^{-6}. For each solver tolerance, adaptation using the new proposed dynamic truncation error estimation as well as feature-based adaptation are compared to their converged versions (10^{-10}).

In this paper, we emphasize the analysis on the influence of the iteration error. Then, only one local adaptation cycle is considered, based on a fixed increase of 40% of new nodes. In order to test the accuracy of the quasi-*a priori* adaptation procedure, a converged solution is computed on the locally adapted meshes and force coefficients (lift, drag and pitching moment) are output. The reference force coefficient values used for comparison are based on the converged solution of the uniformly refined mesh (the original mesh containing 11750 nodes in Figure 1(a) from which each edge is bisected yielding a 46540 nodes mesh).

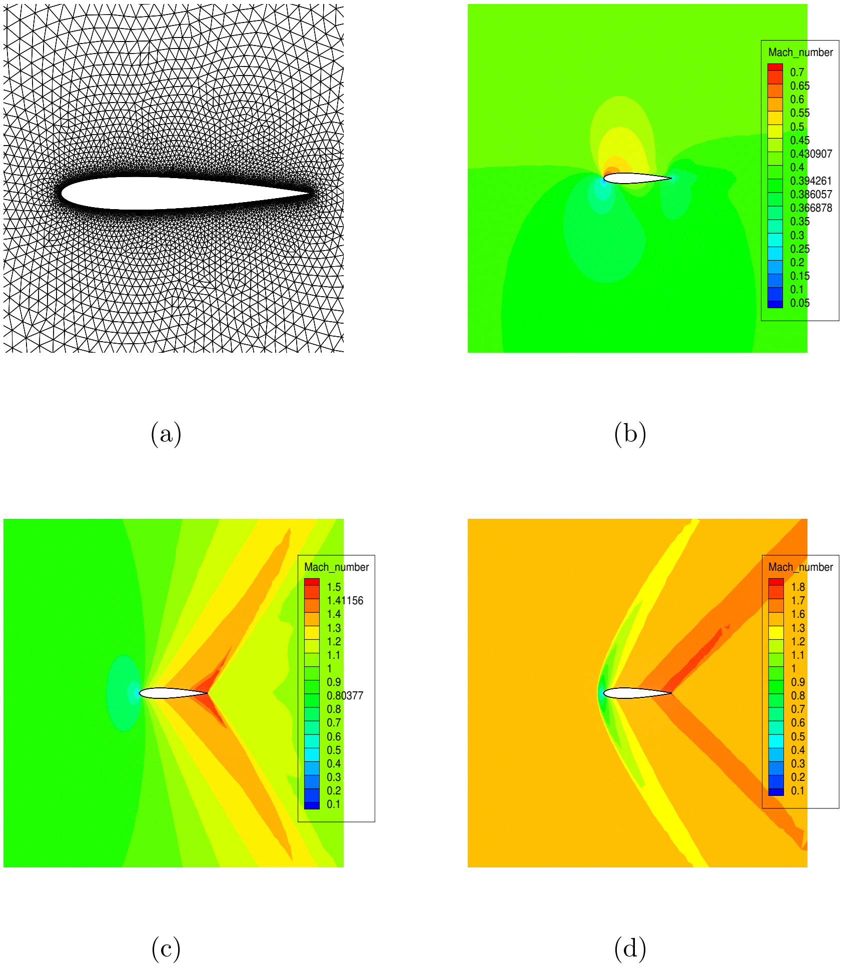
For each test-case, the following study is performed:

- dynamic τ-estimation,
- dynamic τ-estimation-based adaptation and comparison to feature-based and uniform adaptations,
- dynamic τ-extrapolation,
- detailed analysis of computational efficiency.

As discussed in Section 2 and in more details in Fraysse *et al.* [16], the conditions in order to get an accurate estimation of the truncation error are:

1. two consistent meshes Ω^h and Ω^H,
2. a high order restriction operator ($s > p = 2$) to transfer the solution from the initial mesh Ω^h to the coarse mesh Ω^H,
3. a restriction operator to restrict the second term T_2 of Equation 4 which satisfies: $\mathcal{I}_h^H = \mathcal{J}^H \widehat{\mathcal{I}}_h^H \mathcal{J}^{h^{-1}}$.

The first condition is ensured by first generating the coarse mesh Ω^H, in this work it contains 2995 nodes, and uniformly adapting it to get the initial mesh Ω^h, containing 11750 nodes. Building the initial mesh in this way allows also to fulfill the second point, as in this case a simple injection of the solution from Ω^h to Ω^H can be done. Finally, the third point requires, as described in Section 3 the solution of a linear system. In the following, we study the influence of the treatment of T_2 from Equation 4 on the quasi-*a priori* τ-estimation, adaptation and τ-extrapolation.



(a)

(b)

(c)

(d)

Figure 1: Two-dimensional NACA0012 test cases. (a): Initial mesh. Mach number contours of the (b): subsonic, (c): transonic and (d): supersonic test cases.

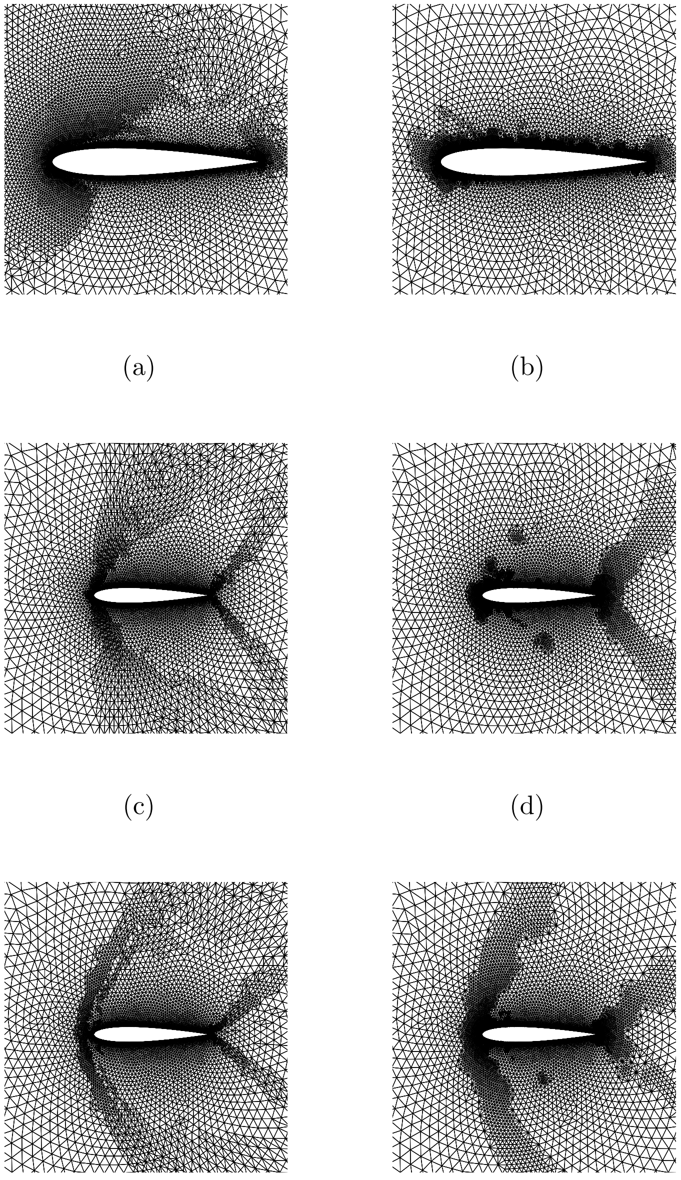


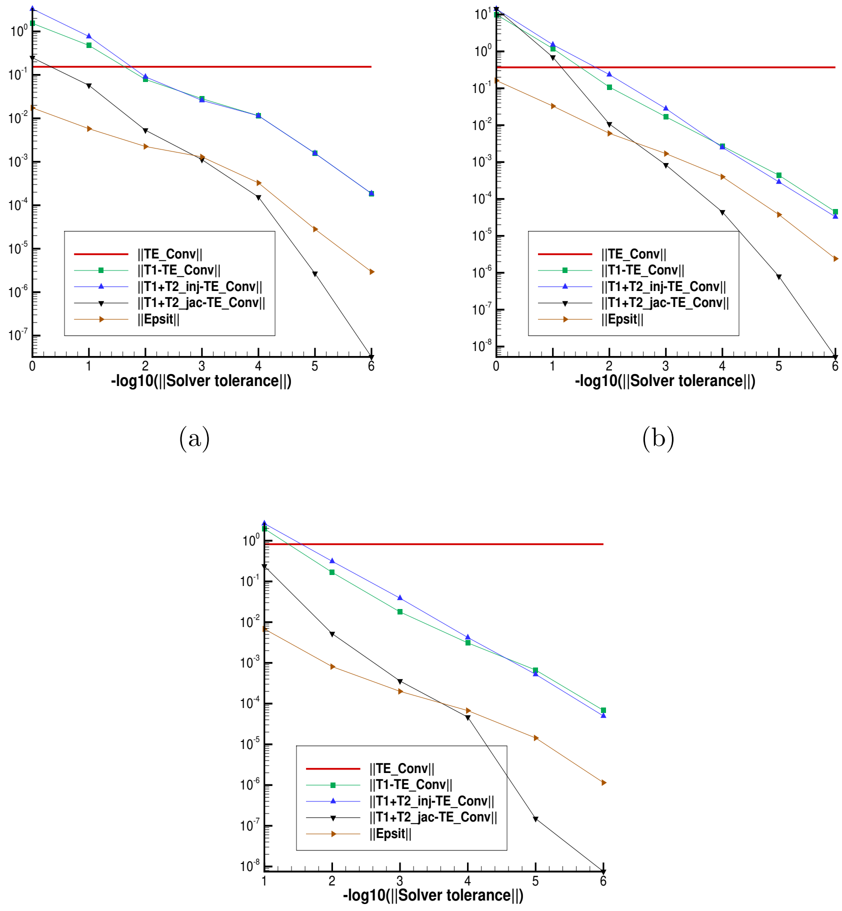
Figure 2: Two-dimensional NACA0012 adapted meshes from converged solutions. Left column: feature-based and right column: TE-based adaptation. First to third row: subsonic, transonic and supersonic test cases respectively.

4.1. Quasi-a priori τ-estimation

In this part, our aim is to verify Equation 5, in particular how the treatment of the second term T_2 of Equation 4 affects the convergence speed towards the final truncation error. In order to do that, we run the Euler solver from DLR TAU-Code to various tolerances (scaled L_2 residuals), from 1 to 10^{-6} to get a non-converged solution. A converged solution is obtained as well by setting the solver tolerance to 10^{-10}. Once these solutions are obtained, the truncation error estimation of Equation 4 is applied following three paths:

- T_1 **only:** in this case only the first term T_1 is computed, note that this is correct when the solution is converged ($T_2 = 0$),
- $T_1 + T_2$ **Injection:** T_2 is computed as well using simple injection to restrict it on the coarse mesh, then added to T_1,
- $T_1 + T_2$ **Jacobian:** T_2 is computed following **(A4)**, then is added to T_1.

For each flow solver tolerance, the L_2 norm (for the continuity equation) of the difference between the quasi-*a priori* truncation error estimation and its value computed from a converged solution, is obtained and reported in Figure 3(a)-(c) for the three test cases. The L_2 norms of the iteration error and the converged truncation error estimation can be found as well. These numerical experiments are in perfect agreement with our theoretical expectations. If the second term T_2 is omitted from Equation 4, then the difference between non-converged and converged truncation errors decreases at the same rate of the iteration error, so no particular gain is obtained using this methodology. Remarkably, adding the second term T_2 by injecting it onto the coarse mesh Ω^H does not improve the rate of convergence at all. On the other hand, computing T_2 following **(A4)** drastically decreases the minimum solver tolerance to get a reasonably accurate estimation. Typically, more than one order of magnitude for the difference between truncation error estimation computed from non-converged and converged solution can be obtained for a solver tolerance of 10^{-2}, for all test cases.



(a)

(b)

(c)

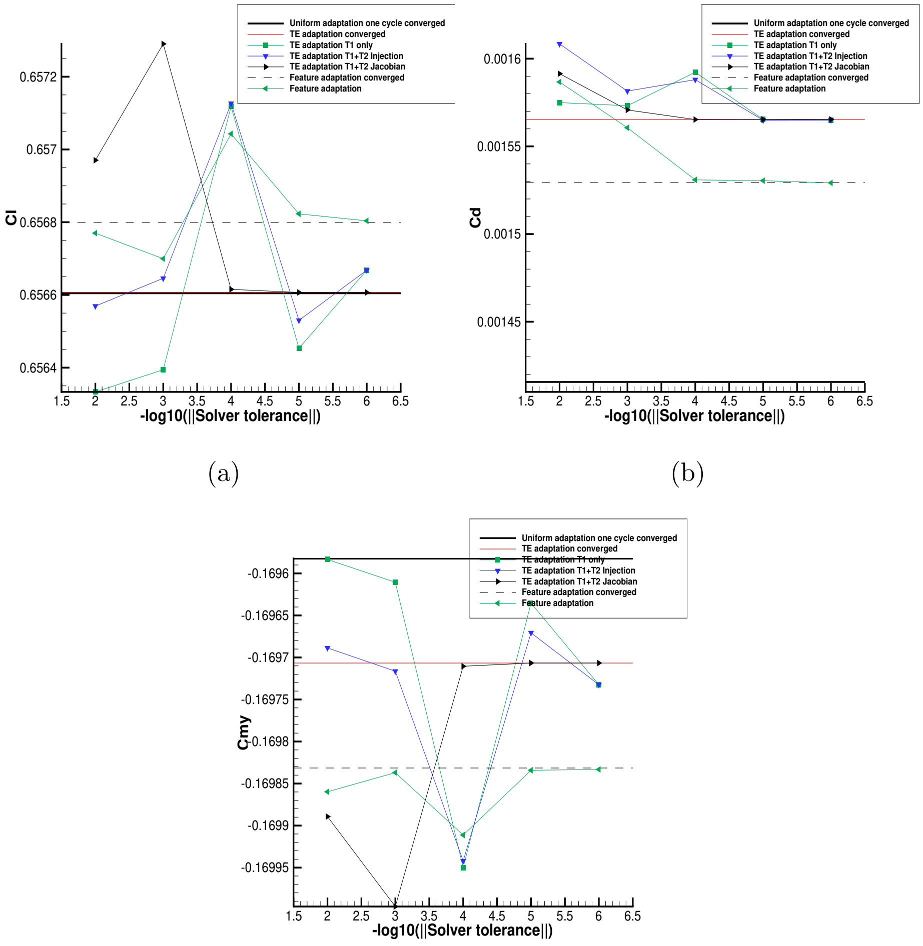
Figure 3: Quasi-*a priori* truncation error estimation. L_2-norm of the TE for the continuity equation. (a): Subsonic case, (b): transonic case and (c): supersonic case.

4.2. Quasi-a priori adaptation

Based on the quasi-*a priori* τ-estimation experiments obtained in Section 4.1, a local adaptation methodology is derived. One local adaptation is applied (40% new nodes) from each quasi-*a priori* truncation error estimation and compared to classical feature-based adaptation as well as a globally refined mesh (obtained by bisecting all the edges of the initial mesh Ω^h). Results obtained from the converged solution (10^{-10}) are computed as well, both for truncation error-based and feature-based sensor. Once the adapted mesh is obtained, a converged solution is computed. Quantitative analysis is obtained by computing and comparing the force coefficients, lift (C_l), drag (C_d) and pitching moment (C_{m_y}) coefficients and is reported in Figure 4(a)-(c) for the subsonic, Figure 5(a)-(c) for the transonic and Figure 6(a)-(c) for the supersonic test case.

As far as adaptations from the converged solution is concerned, it can be noticed that the truncation error indicator obtains more accurate results than the feature-based sensor for the prediction of all force coefficients. Indeed, taking as reference values the force coefficients obtained on a globally refined mesh (obtained by bisecting all the edges of the initial mesh), it can be seen that local mesh adaptation from τ-estimation yield closer predictions for all test cases.

In the case of locally adapting the mesh by computing the sensors from non-converged solutions, the results follow the analysis from Section 4.1. It is clear on these graphs that adaptations based on non-converged feature-based or truncation error without particular treatment for the second term T_2 last to provide stabilized values. On the other hand, adapting the mesh from a truncation error sensor which satisfies **(A4)** provides very acceptable results from lower solver tolerances. Typically, stabilized values can be obtained from a solver tolerance of 10^{-4} for the subsonic test case and 10^{-3} for the transonic and supersonic test cases.

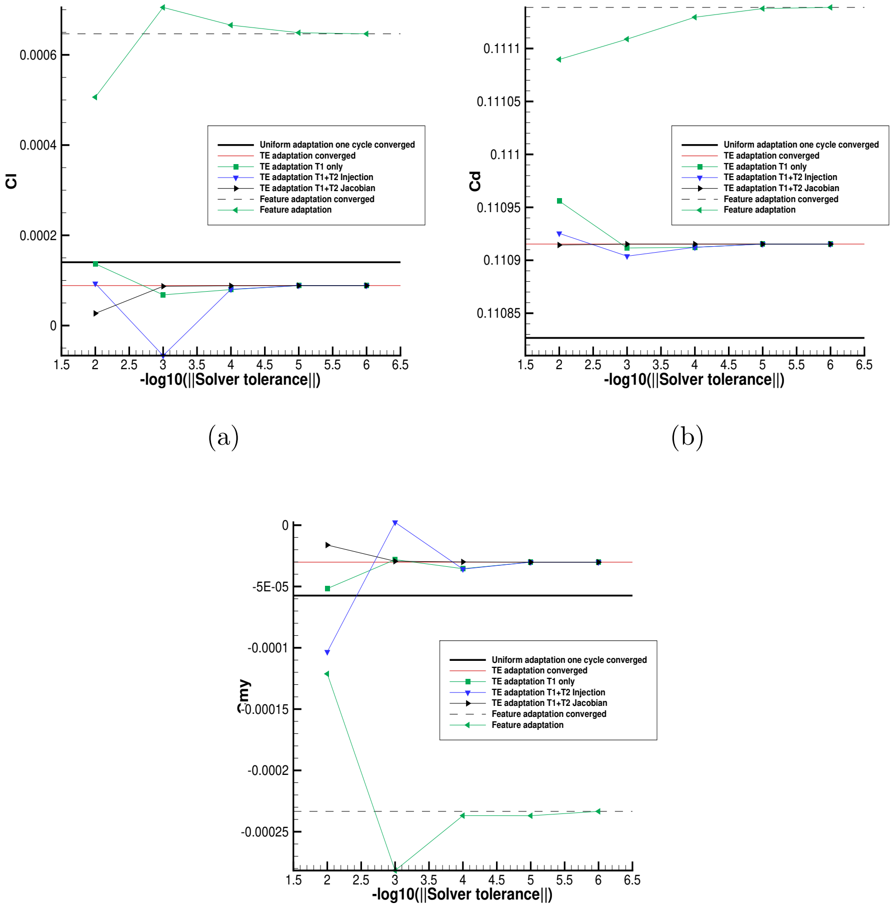


(a)

(b)

(c)

Figure 4: Effect on force coefficients of quasi-*a priori* adaptation, subsonic case. (a): C_l (b): C_d, (c): C_{m_y}.

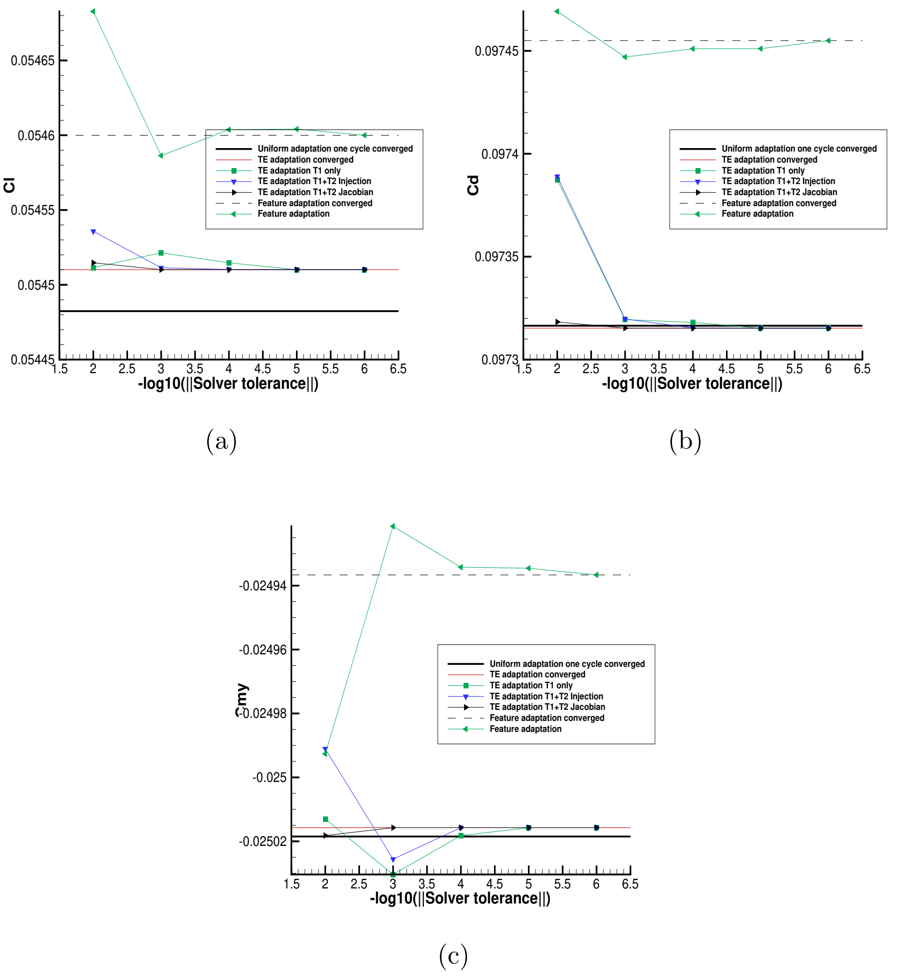


(a)

(b)

(c)

Figure 5: Effect on force coefficients of quasi-*a priori* adaptation, transonic case. (a): C_l (b): C_d, (c): C_{m_y}.



(a)

(b)

(c)

Figure 6: Effect on force coefficients of quasi-*a priori* adaptation, supersonic case. (a): C_l (b): C_d, (c): C_{m_y}.

4.3. Quasi-a priori τ-extrapolation

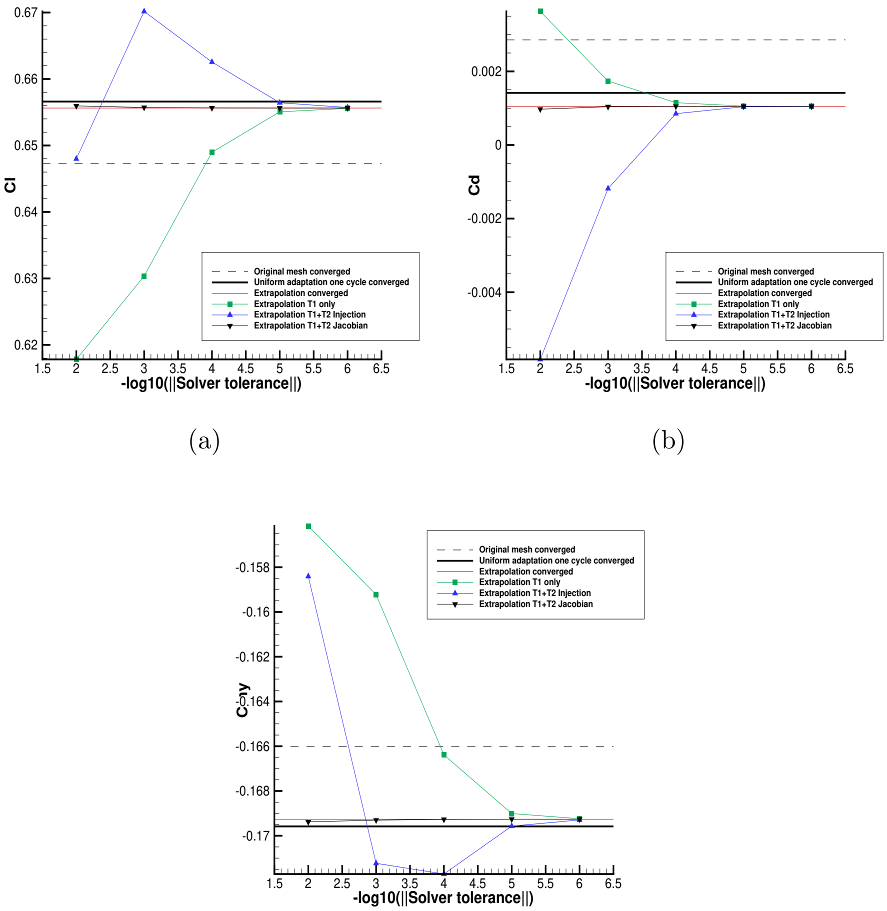
With a truncation error estimation in hand, a natural next step is to use it in order to increase the accuracy of the numerical scheme. This goal is achieved by solving Equation 9, which consists of the original PDE forced by the truncation error estimate. While the usual τ-extrapolation procedure involves the solution of the problem on the coarse mesh Ω^H, as described in Equation 9, a further step is considered here by preliminary interpolating the truncation error estimate back to the initial mesh Ω^h. The extrapolated problem reads:

$$\mathcal{R}^h(\bar{\boldsymbol{U}}^h) = \frac{\rho^p}{1-\rho^p}\mathcal{I}_H^h \tau_h^H \tag{12}$$

The procedure closely follows the quasi-*a priori* adaptation method. Once the truncation error estimations are obtained from the different solutions of the Euler equations (from the three paths described earlier and different solver tolerances), they are interpolated back to the original mesh Ω^h and Equation 12 is solved until convergence is reached (10^{-10}). The three truncation error estimation approaches are checked and the force coefficients obtained from the solution of Equation 12 are reported in Figure. 7 for the subsonic, Figure 8 for the transonic and Figure 9 for the supersonic test cases.

τ-extrapolation from a converged solution provides a substantial gain in the accuracy of all the force coefficients, reaching almost the results obtained on an overly refined mesh for all force coefficients, and in some cases even better predictions are obtained (by observing that all force coefficient converge monotonically). This clearly shows the efficiency of the approach in order to increase the accuracy of the numerical scheme.

Results obtained from non-converged solutions show very interesting results. For the subsonic test case, while the T_1 **only** and the T_1+T_2 **Injection** need a solution obtained with a solver tolerance of at least 10^{-5} to get stabilized, it is remarkable that if the truncation error estimation satisfies **(A4)**, then the force coefficients almost reach their final values for a solver tolerance of 10^{-2}. In the transonic case, the T_1 **only** and the $T_1 + T_2$ **Injection** require a solution from a solver tolerance of at least 10^{-5} to give steady results, whereas if the truncation error estimation satisfies **(A4)**, then the force coefficients almost reach their steady values for a solver tolerance of 10^{-3}. The supersonic test case follows this trend: if the T_1 **only** and the $T_1 + T_2$ **Injection** need a solution obtained with a solver tolerance of at least 10^{-4} to get stabilized, it can be noticed that if the truncation error estimation satisfies **(A4)**, then the force coefficients almost reach their final values for a very low solver tolerance of 10^{-2}.

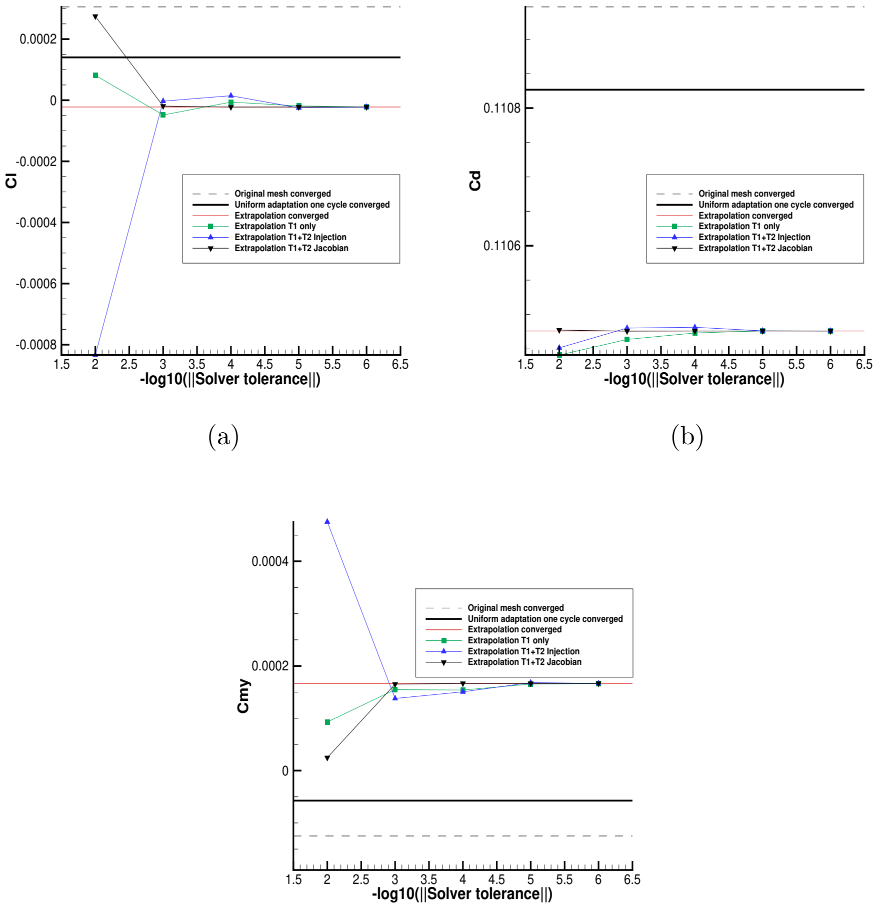


(a)

(b)

(c)

Figure 7: Effect on force coefficients of quasi-*a priori* τ-extrapolation, subsonic case. (a): C_l (b): C_d, (c): C_{m_y}.

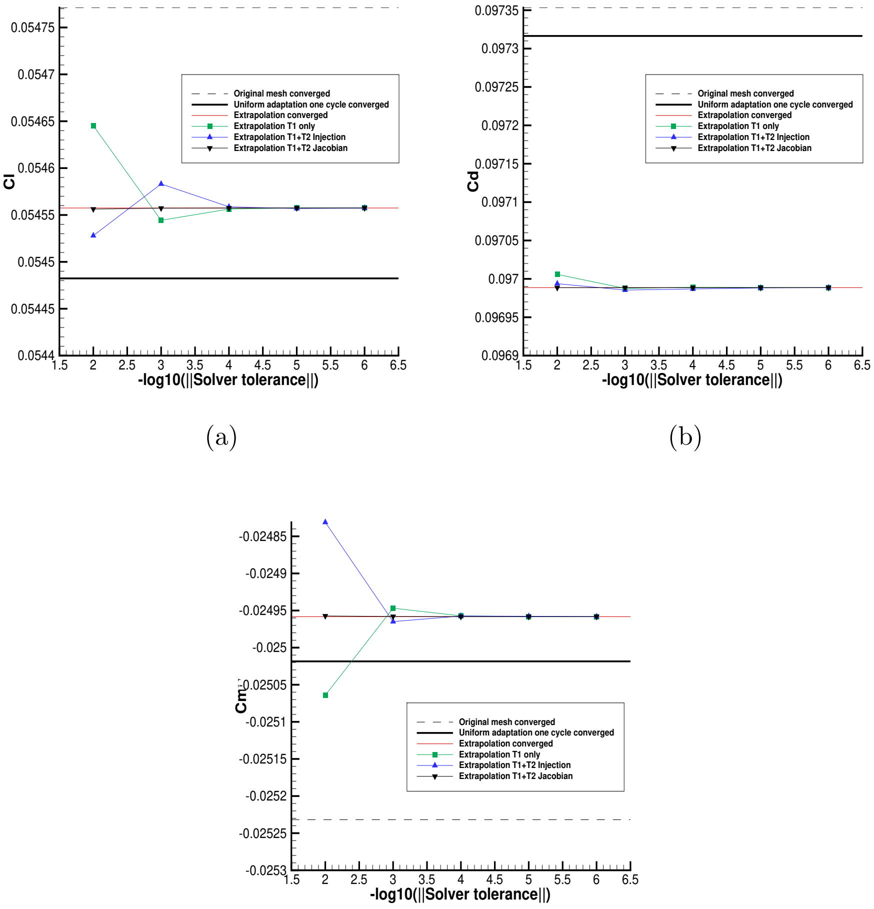


(a)

(b)

(c)

Figure 8: Effect on force coefficients of quasi-*a priori* τ-extrapolation, transonic case. (a): C_l (b): C_d, (c): C_{m_y}.



(a)

(b)

(c)

Figure 9: Effect on force coefficients of quasi-*a priori* τ-extrapolation, supersonic case. (a): C_l (b): C_d, (c): C_{m_y}.

4.4. Computational costs

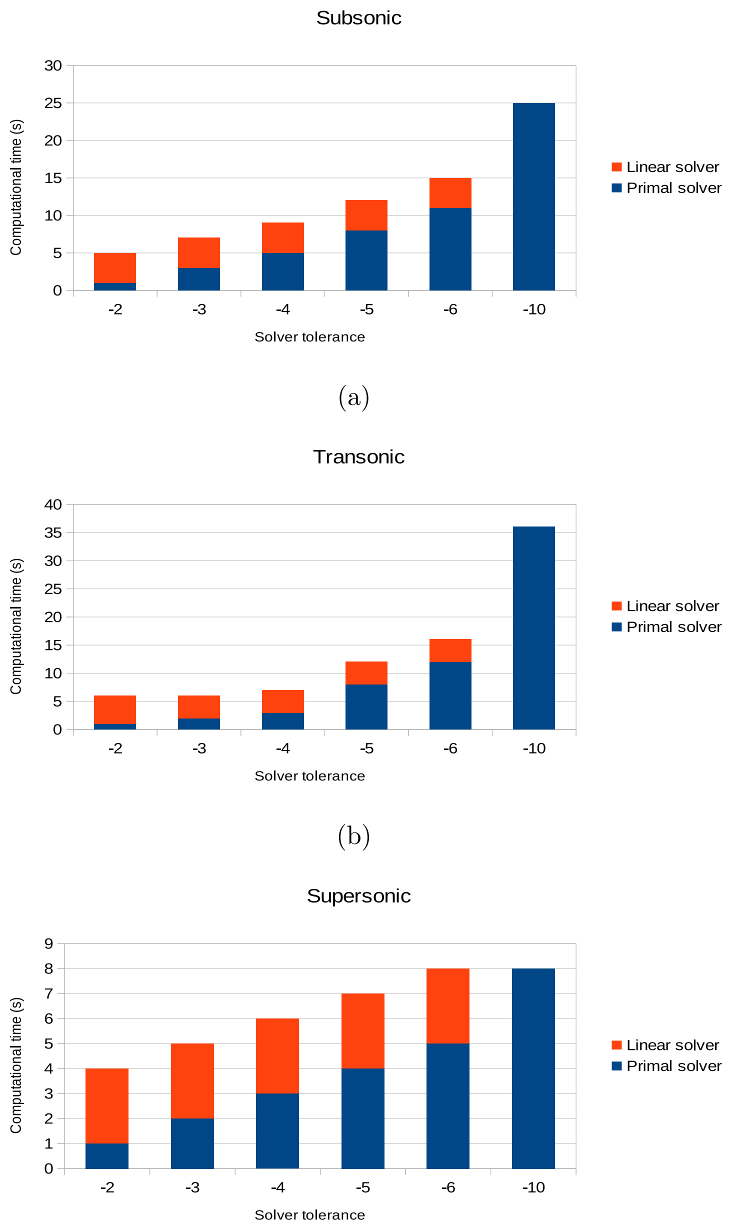
4.4.1. τ-estimation

In this subsection our goal is to have a closer look at the computational costs associated to the calculation of the truncation error estimate. To this aim, for each solver tolerance, we reported in Figure 10(a)-(c) for the three test cases, the computational times for the truncation error estimation based on assumption **(A4)**.

It is important to notice that, only computational times associated to the flow solver and the linearized problem are taken into account. Computational times associated to the preprocessing, residual evaluations for the truncation error estimation and local refinement are negligible with respect to solving the Euler equations or the linear problem.

The discretization parameters used in the computation of solutions for the set of Euler equations are described now. The pseudo-time integration consists of a three-steps Runge-Kutta with four levels of W-Cycle multigrid and a constant CFL number of 1.8. This temporal integration is the default choice left by the DLR TAU-Code. The linear system associated to the truncation error estimation is solved using the external package PETSc and it employs a preconditioned ILU(0)-GMRES solver.

For each test case, by comparing Figure 10(a) and Figure 3(a) (Figure 10(b) and Figure 3(b), Figure 10(c) and Figure 3(c) respectively) it can be seen that for an error lower than 10% in the quasi-*a priori* truncation error estimation with respect to its converged value, the new proposed estimator (following **(A4)**) requires a solver tolerance of 10^{-2} and therefore yields to a total normalized time of 0.2 (respectively 0.17 and 0.5 for the transonic and supersonic test cases). This means it needs 20% (respectively 17% and 50%) of the computational time required to evaluate the truncation error from a converged solution. In this particular case the flow solver requires approximately only one second to reach this tolerance and most computational time is spent in the linear system (four seconds). It is remarkable that the time required to solve the linear system is independent of the solution. It is interesting to notice that the performance is lower for the supersonic test case. The reason is that the linear solver seems to be insensitive to the highly hyperbolic flow, while the flow solver converges must faster.



(a)

(b)

(c)

Figure 10: Computational times, τ-estimation from non-converged solutions using **T1+T2 Jacobian** strategy. (a): Subsonic (b): transonic, (c): supersonic test cases.

4.4.2. Adaptation and τ-extrapolation

The computational time analysis of Section 4.4.1 is extended here to take into account the adaptation and the extrapolation procedures. In addition to the times required to compute the solution on the original mesh and to solve the linear system, the time required to solve until the convergence on the adapted mesh or for the extrapolated problem are added. Here, the CPU time required for preprocessing of the meshes and the adaptation are not taken into account as they are negligible. The solutions on the locally adapted meshes as well as the solutions of the extrapolated problem are initialized from free-stream values.

In this section, the reference time is chosen to be the computational time required to solve the Euler equations on an overly refined mesh, as is described in Table 2 for the subsonic case and in Table 3 and Table 4 for the transonic and supersonic test cases respectively. It is interesting to remark that the solution on the original mesh only requires 12% (13% and 15% for the transonic and supersonic cases respectively) of the computational effort associated to solving the problem on the overly refined mesh, showing how improving the accuracy by global refinement for industrial problems is unpractical.

As far as adaptation results from non-converged solutions is concerned, and in particular the one obtained using the new proposed methodology (following **(A4)**), it can be seen in Figure 11(a)-(c) that some computational time can be gained. For instance, considering a solver tolerance of 10^{-4} (10^{-3} for the transonic and supersonic test cases), for which stable results were obtained, the computational time decreases to 26% (24% and 43% respectively) of the overall time necessary to get a converged solution on the globally refined mesh.

In the case of τ-extrapolation (see Figure 12(a-c)), the gain in CPU cost is even better. Performing τ-extrapolation gives results similar or more accurate than the overly refined mesh and needs only 24% (27% and 28% respectively) of time. In the case of quasi-*a priori* τ-extrapolation, using the new proposed methodology we verified that an accurate solution could be obtained from a truncation error estimation evaluated from a solver tolerance of 10^{-2} for the subsonic and transonic test case and 10^{-3} for the supersonic case, thus in this case the total solving time drops down to 14% (15% and 24% respectively).

Table 2: Computational times, subsonic case, converged solutions on the original and globally refined mesh.

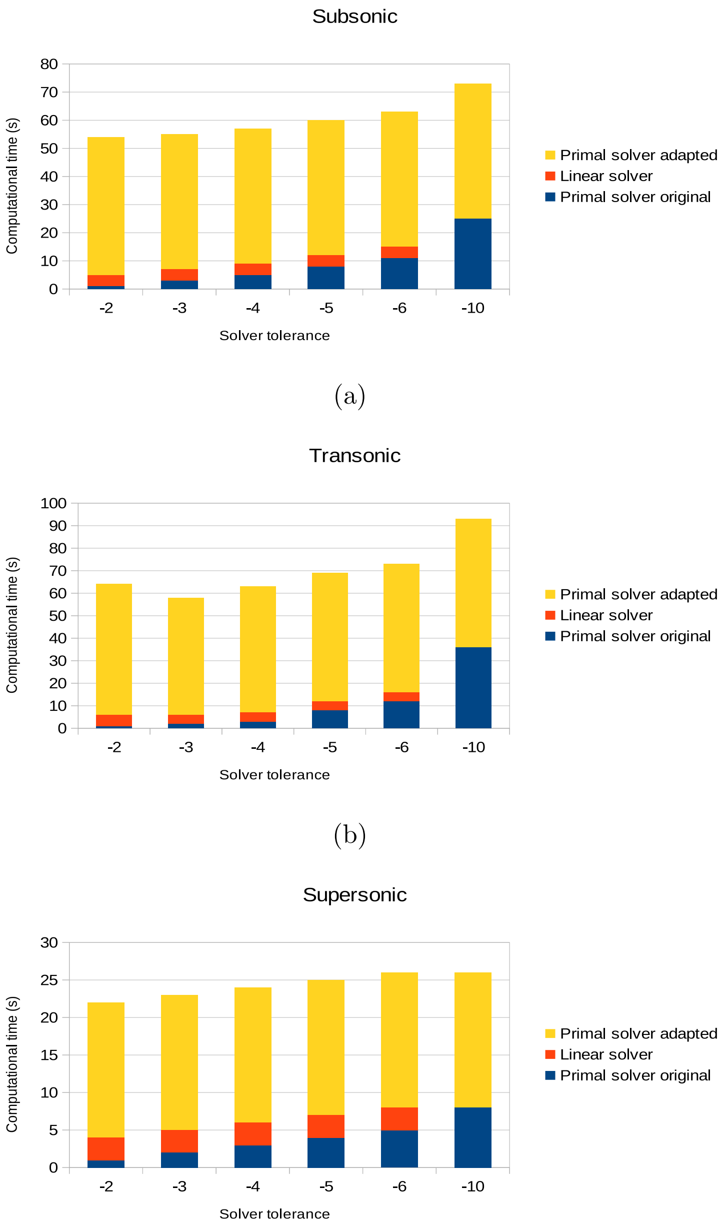
	Original	Uniform Adaptation
Solver time (s)	25	216
Normalized time	0.12	1

Table 3: Computational times, transonic case, converged solutions on the original and globally refined mesh.

	Original	Uniform Adaptation
Solver time (s)	36	268
Normalized time	0.13	1

Table 4: Computational times, supersonic case, converged solutions on the original and globally refined mesh.

	Original	Uniform Adaptation
Solver time (s)	8	54
Normalized time	0.15	1

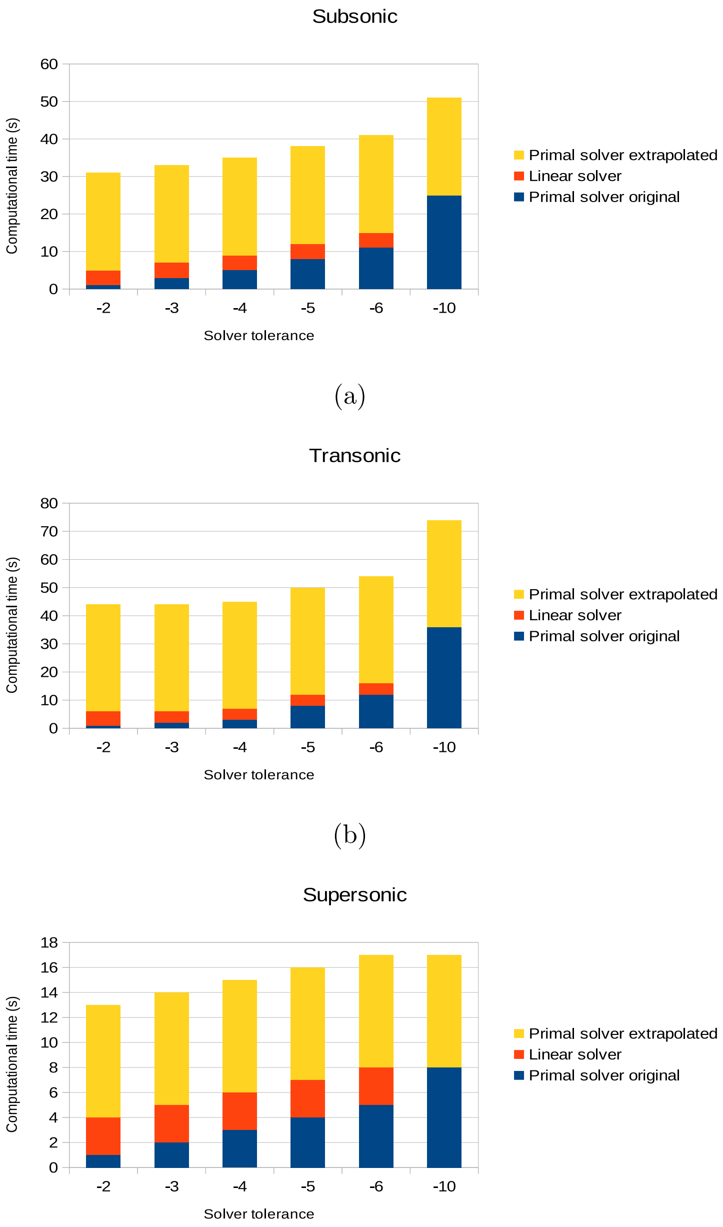


(a)

(b)

(c)

Figure 11: Computational times, adaptation from non-converged solutions using **T1+T2 Jacobian** strategy. (a): Subsonic (b): transonic, (c): supersonic test cases.



(a)

(b)

(c)

Figure 12: Computational times, extrapolation from non-converged solutions using **T1+T2 Jacobian** strategy. (a): Subsonic (b): transonic, (c): supersonic test cases.

5. Conclusion

In this paper, we applied a new methodology to estimate the local truncation error of the steady-state Euler equations. This estimator is based on the so-called τ-estimation which consists on evaluating the spatial discretization on two embedded meshes. The approach described in this article does not assume a converged solution, but requires the solution of an additional linear system for the discrete Jacobian.

Based on these quasi-*a priori* evaluations, the influence of the level of convergence of the flow solver is studied on a local adaptation procedure and a higher order extrapolation methodology. Test cases consisting of the two-dimensional inviscid flow past a NACA0012 airfoil at different free-stream conditions have been used to assess the performance of this approach, and comparisons with a classical feature-based mesh adaptation indicator have been carried out.

It is demonstrated that this approach, which lies between *a priori* and *a posteriori*, allows for an interesting reduction in computational time with respect to the *a posteriori* τ-estimation approach, whereas the precision of the estimation as well as the local adaptation and higher order extrapolation remains accurate.

Acknowledgments

The research described in this paper has been supported by Airbus-Spain under project CORESIMULAERO/FUSIM-E. The TAU Code is property of the Deutsches Zentrum für Luft- und Raumfahrt (DLR), developed at the Institute of Aerodynamics and Flow Technology at Göttingen and Braunschweig, and has been licensed to UPM through a research and development cooperation agreement.

[1] Dwyer, H. A., and Kee, R. J., and Sanders, B. R., *Adaptive grid method for problems in fluid mechanics and heat transfer*, AIAA Journal, Vol. 18, pp. 1205-1212, 1980

[2] Palmério, C., and Olivier, C., and Dervieux, A., *2-D and 3-D unstructured mesh adaption relying on physical analogy*, Proc. 2nd Int. Conj. on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Miami, FL, 1988

[3] Hecht, F. and Mohammadi, B., *Mesh adaptation by metric control for multi-scale phenomena and turbulence*, AIAA Paper 97-0859, 1997

[4] Frey, P. J. and Alauzet, F., *Anisotropic Mesh Adaptation for CFD Computations*, Computer Methods in Applied Mechanics and Engineering, Vol. 194, pp. 5068-5082, 2005

[5] Mavriplis, D. J., *Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes*, AIAA Journal, Vol. 28, pp. 213-221, 1990

[6] Parthasarathy, V. and Kallinderis, Y., *New Multigrid Approach for Three-Dimensional Unstructured, Adaptive Grids*, AIAA Journal, Vol. 32, pp. 956-963, 1994

[7] Kallinderis, Y. and Vijayan, P., *Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes*, AIAA Journal, Vol. 31, pp. 1440-1447, 1993

[8] Parthasarathy, V. and Kallinderis, Y., *Directional Viscous Multigrid Using Adaptive Prismatic Meshes*, AIAA Journal, Vol. 33, pp. 69-78, 1995

[9] Braaten, M. E. and Connell, S. D., *Three-Dimensional Unstructured Adaptive Scheme for the Navier-Stokes Equations*, AIAA Journal, Vol. 34, pp. 281-290, 1996

[10] Dannenhoffer, J.F. and Baron, J.R., *Adaptation procedures for steady state solution of hyperbolic equations*, AIAA Paper 84-0005, 1984

[11] Warren, G. P., Anderson, W. K., Thomas, J. P. and Krist, S. L., *Grid Convergence for Adaptive Methods*,AIAA-91-1592,1991

[12] Fidkowski, K. J., *Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results*, AIAA-2009-1303

[13] Fidkowski, K. J. and Roe, P. L., *An entropy adjoint approach to mesh refinement*, SIAM Journal on Scientific Computing, Vol. 32, 2010

[14] Aftosmis, M.J., *Upwind method for simulation of viscous flow on adaptively refined meshes*, AIAA Journal, Vol. 32, pp. 268-277, 1994

[15] Brandt, A., *Multigrid Techniques: 1984 Guide with applications to fluid dynamics*

[16] Fraysse, F., and De Vicente, J. and Valero, E., *The estimation of truncation error by τ-estimation revisited*, Journal of Computational Physics, Vol. 231, pp. 3457-3482, 2012

[17] Fraysse, F., Valero, E. and Ponsín, J., *Comparison of Mesh Adaptation Using the Adjoint Methodology and Truncation Error Estimates*, AIAA Journal, Vol. 50, No. 9, pp. 1920-1932, 2012

[18] Fraysse, F., de Vicente, J. and Valero, E., *Mesh adaptation and higher order extrapolation of the Reynolds Averaged Navier-Stokes equations using τ-estimation*, Proceedings of the Institution of Mechanical Engineers, Part G, Journal of Aerospace Engineering, DOI: 10.1177/0954410012462521, 2012

[19] Gerhold, T., Hannemann, V. and Schwamborn, D., *On the Validation of the DLR-TAU Code*, New Results in Numerical and Experimental Fluid Mechanics, Notes on Numerical Fluid Mechanics, 72, pp. 426-433, 1999

[20] Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., Curfman McInnes, L., Smith, B.F. and Zhang, H., *PETSc Users Manual*, ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012

[21] Venditti, D. A., *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA,2002