Comparison of Mesh Adaptation Using the Adjoint Methodology and Truncation Error Estimates

François Fraysse* and Eusebio Valero†
Universidad Politécnica de Madrid, 28040 Madrid, Spain
and
Jorge Ponsín‡
Instituto Nacional de Técnica Aeroespacial, 28850 Madrid, Spain

Mesh adaptation based on error estimation has become a key technique to improve the accuracy of computational-fluid-dynamics computations. The adjoint-based approach for error estimation is one of the most promising techniques for computational-fluid-dynamics applications. Nevertheless, the level of implementation of this technique in the aeronautical industrial environment is still low because it is a computationally expensive method. In the present investigation, a new mesh refinement method based on estimation of truncation error is presented in the context of finite-volume discretization. The τ estimation method uses auxiliary coarser meshes to estimate the local truncation error, which can be used for driving an adaptation algorithm. The method is demonstrated in the context of two-dimensional NACA0012 and three-dimensional ONERA M6 wing inviscid flows, and the results are compared against the adjoint-based approach and physical sensors based on features of the flow field.

Nomenclature

h	=	local mesh size associated with the current grid of interest
h_c	=	local mesh size associated with the coarsest grid, $h_c > h$
h_f	=	local mesh size associated with the finest grid, $h_f < h < h_c$
$\boldsymbol{I}^h$	=	linear continuum-to-grid restriction operator
$\hat{\boldsymbol{I}}_h^{h_c}$, $\boldsymbol{I}_h^{h_c}$	=	fine-to-coarse restriction operators
$\hat{\boldsymbol{I}}_h^{h_f}$, $\boldsymbol{I}_h^{h_f}$	=	coarse-to-fine interpolation operators
J, J_h	=	exact and discrete functional output
p	=	formal order of accuracy of the spatial scheme
$\boldsymbol{R}$	=	symbolic operator representing the partial differential equation
$\boldsymbol{R}_h$	=	symbolic operator representing the discrete partial differential equation, discrete residual
$\boldsymbol{U}$, $\boldsymbol{U}_h$, $\tilde{\boldsymbol{U}}_h$	=	exact solution vector of the partial differential equation, exact, and approximate solution vector (respectively) of the discretized partial differential equation
$\boldsymbol{\epsilon}^h$, $\boldsymbol{\epsilon}_{it}^h$	=	exact discretization and iteration error
$\boldsymbol{\tau}^h$, $\boldsymbol{\tau}_h^{h_c}$	=	exact and relative truncation error
$\boldsymbol{\Psi}_h$	=	discrete adjoint solution

I. Introduction

NOWADAYS, improving the accuracy of the computational fluid dynamics (CFD) computations has become a major goal in the aeronautical industry. The use of CFD for aerodynamic design has increased within the last years due to the associated product costs and time-to-market reduction. Because of the requirement of fuel reduction, to be competitive airplanes must be optimized in the range of few drag counts. Therefore, in the last years, the required error bands in global aerodynamical coefficients have been drastically reduced.

The accuracy of a CFD computation depends on two kinds of error sources: modeling errors and numerical or discretization errors. It has been shown that, for a standard airplane at cruise conditions, the CFD accuracy mainly depends on discretization errors [1]. It is well known that, in the asymptotic range, discretization errors are proportional to the local size of the mesh as $\mathcal{O}(h^p)$, where h represents the local size of the mesh elements and p is the formal discretization order of the numerical method. The reduction of the numerical error may be accomplished by a twofold strategy: to increase the order of the discretization scheme and/or to increase the number of nodes. These two techniques are known as p and h adaptation, respectively; p or even hp adaptation [2] is usually considered within the finite-element community, where the order p can be prescribed in a flexible way. In the finite-volume community, however, h refinement is usually preferred because of the very nature of this discretization approach.

The selection of a reliable adaptation parameter is a key aspect to reduce the errors in the computation [3]. One classical approach for adaptation indicators in the literature is gradient and undivided difference-based feature detectors [4–7]. This approach, known as featured-based indicator, can detect flow phenomena, when taking for instance the primitive variables into account (e.g., shocks are detected because of large gradients). Although these methods have been very successful for a wide range of problems [4–12], their lack of mathematical foundation makes them difficult to apply in a user-independent way to problems far from established experience. Indicators based on estimates of interpolation error [10–12] somehow remedy to the lack of formalism. However, because these lead to an indication of the local curvature of a variable or a combination of variables, they are not clearly different from regular feature-based indicators [8].

Consequently, adaptation parameters based on physical features work generally well when one aims at solving the details of the flow or improving the accuracy in regions where the physical scales must be resolved (shear layers, shock waves, etc.). However, it has been shown that increasing the resolution in these regions does not necessarily improve the accuracy of engineering outputs of interest as, for example, global forces [13]. Therefore, in the last years, special attention has been paid to indicators based on numerical errors, the so-called family of residual-based adaptation indicators. We will describe in this paper two residual-based indicators. The first

Received 14 July 2011; revision received 24 January 2012; accepted for publication 24 February 2012.

*Ph.D. Student, Applied Mathematics Department, Escuela Técnica Superior de Ingenieros Aeronáuticos; francois.fraysse@upm.es.

†Professor, Applied Mathematics Department, Escuela Técnica Superior de Ingenieros Aeronáuticos; eusebio.valero@upm.es.

‡Research Engineer, Fluid Dynamics Branch; ponsinj@inta.es.

one relates the local residual to an output functional using the solution of a dual problem, this is the well known adjoint methodology. The second one is based on truncation error (TE) estimates, known as τ estimation [14].

The adjoint-based error-correction method [15] uses an adjoint formulation to relate the error in the functional to the local residual of the primal solution. Adjoint-based methods provide tools to improve the accuracy of output functionals by either computing correction terms or through grid adaptation, for which an output-oriented adaptation sensor is obtained, thus avoiding the use of overly refined meshes. However, in finite-volume methods, most of the adjoint h-adaptive techniques require the use of an embedded or uniformly refined mesh to compute the local residual. The explicit generation and storage of this mesh demands high computational and memory resources, which can be prohibitive for three-dimensional computations. Furthermore, the adjoint equations generally yield the same level of stiffness as the primal equations. Thus, for turbulent viscous flows, these equations are in general very expensive and require robust and efficient iterative methods. Therefore, it would be desirable to have a simpler and computationally cheaper adaptation method for improving the functional outputs accuracy. This high computational cost has been considered by other authors, not only in the finite-volume community (for instance, in the context of hp adaptation [2]) yielding some cheaper but nevertheless efficient indicators like the entropy adjoint [16] or, as we will present in this paper, truncation error estimation.

Because the discretized equations represent approximations to the differential equation, the exact solution of the latter does not satisfy the difference equation. The imbalance, which is due to truncation of the Taylor series, is called truncation error. Analysis of the truncation error can be done by deriving analytically the Taylor series expansions for a given numerical scheme [17–19]. However, the primary issue with this approach is the complexity of the related expressions, especially for multidimensional problems on arbitrary grids. The second drawback is the lack of generality, as the truncation error differs from one numerical scheme to another. Another family of methods employed to study the discretization or truncation error is based on Richardson extrapolation [14,20,21]. The estimation of the discretization error by Richardson extrapolation [22–24] is based on the existence of a Taylor series expansion of the solution, assuming a smooth solution of the partial differential equation. The major advantage of this approach is that it is independent of the numerical scheme and is then easily extendable to any numerical solver. However, it requires the computation of an approximated solution on at least two meshes (three if the order of accuracy of the numerical scheme is considered as an unknown) of different spacings, making it hardly suitable for three-dimensional industrial applications. On the other hand, the estimation of the truncation error by means of τ estimation [14] is an interesting alternative as it does not require the solution on a secondary grid but only the computation of the residual. Furthermore, it is closely linked to the forcing term in the full-approximation storage (FAS) approach of the multigrid technique [21], making it easy to compute in a solver where a multigrid strategy is implemented.

An extensive analysis can be found in Bernert [25] and Fulton [26]. The former derives stringent conditions on the restriction operators to transfer information from the different grid levels. The latter introduces a modified formulation that allows for an accurate truncation error estimation using injection/full-weighting. However, both authors focused on finite-difference methods applied on Cartesian uniform grids. Further analysis can be found in Fraysse et al. [27], where the truncation error estimator is derived and analyzed in the context of general finite-volume schemes on structured and unstructured meshes as well as nonconverged solutions and nonlinear equations.

Once a good estimation of the error is available, either associated with a given functional output from the adjoint methodology or by means of truncation error estimate, a robust mesh adaptation strategy can be derived.

We propose in this paper to develop a mesh adaptation methodology based on τ estimation and to compare the results with an adjoint-based and a classical feature-based indicator. The outline of this paper is as follows. The mathematical foundations of the adjoint and τ estimation methods are described in Sec. II. The details of the implementation procedure are explained in Sec. III, where a short description of the feature-based adaptation methodology is also included. The results of the two residual-based methods and the comparison with the feature-based indicator are discussed in Sec. IV. Conclusions and future works are described in Sec. V.

II. Mathematical Background

A. Error Estimation Using Adjoint Methodology

The adjoint-based approach estimates the discretization error of scalar quantities derived (obtained by postprocessing) from the flow solution. The motivation of this method comes from the fact that, in many applications, engineers are more interested in accurately computing some global outputs of industrial interest (i.e., total forces, heat fluxes, etc.) than knowing the flow details in some areas of the physical domain. Typical global outputs in aerodynamic simulations are the lift, drag, or moment coefficients, which can be obtained from a surface integral of the form

$$J(\boldsymbol{U}) = \int_{\partial\Omega} f(\boldsymbol{U})\,\mathrm{d}\sigma$$

where $\boldsymbol{U}$ is the solution of the partial differential equations (PDEs), for example, Euler equations, which model the flow motion on a given domain Ω. These equations are denoted symbolically as

$$\boldsymbol{R}(\boldsymbol{U}) = 0 \tag{1}$$

The solution of the governing flow equations is obtained by discretizing Eq. (1) in the computational domain Ω_h. The resulting discrete equations on the mesh Ω_h with characteristic size h may be written as

$$\boldsymbol{R}_h(\boldsymbol{U}_h) = 0 \tag{2}$$

where $\boldsymbol{R}_h$ is also known as the discrete nonlinear residual operator, and $\boldsymbol{U}_h$ is the numerical approximation of $\boldsymbol{U}$ on mesh h. Equivalently, a numerical approximation of the functional $J_h(\boldsymbol{U}_h)$ is obtained.

The aim of the adjoint error estimation is twofold: first, to obtain an estimation of the discretization error in the functional $J(\boldsymbol{U}) - J_h(\boldsymbol{U}_h)$. As it will be shown later, this can be accomplished naturally by using adjoint equations. Second, to provide a sensor for a grid adaptation process which eventually improves the functional accuracy.

The adjoint method for error estimation used here follows the work of Venditti and Damorfal [28], who developed a duality-based error estimation and adaptive procedure in the framework of the discrete adjoint equations. The mathematical formulation is reproduced here for convenience.

Let us first consider the estimation of the relative functional discretization error between two meshes: an initial mesh Ω_h, on which it is affordable to obtain the solution, and a finer embedded mesh Ω_{h_f}, obtained by uniform refinement of the initial mesh. The aim is to obtain an estimation of the relative functional output discretization error $J_{h_f}(\boldsymbol{U}_{h_f}) - J_h(\boldsymbol{U}_h)$ without solving the discrete equations on the fine mesh. Assume now that an approximate solution $\tilde{\boldsymbol{U}}_{h_f}$ is available on the fine mesh. As it will be explained later, this approximate solution is obtained by interpolation of the current mesh solution $\boldsymbol{U}_h$, and it can be viewed as a small perturbation of the exact discrete solution $\boldsymbol{U}_{h_f}$, which is unknown. Performing first-order Taylor expansions of the functional with respect to this perturbed solution yields

$$J_{h_f}(\boldsymbol{U}_{h_f}) \approx J_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) + \left.\frac{\partial J_{h_f}}{\partial \boldsymbol{U}_{h_f}}\right|_{\tilde{\boldsymbol{U}}_{h_f}} \delta\boldsymbol{U}_{h_f} \tag{3}$$

where $\delta \boldsymbol{U}_{h_f} = \boldsymbol{U}_{h_f} - \tilde{\boldsymbol{U}}_{h_f}$ is defined as the solution error vector. Consider now the linearization of the discrete residual, Eq. (2), about the perturbed solution $\tilde{\boldsymbol{U}}_{h_f}$:

$$\boldsymbol{R}_{h_f}(\boldsymbol{U}_{h_f}) \approx \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) + \left.\frac{\partial \boldsymbol{R}_{h_f}}{\partial \boldsymbol{U}_{h_f}}\right|_{\tilde{\boldsymbol{U}}_{h_f}} \delta \boldsymbol{U}_{h_f} \tag{4}$$

where $\partial \boldsymbol{R}_{h_f}/\partial \boldsymbol{U}_{h_f}|_{\tilde{\boldsymbol{U}}_{h_f}}$ is the discrete Jacobian matrix of the discrete residual, evaluated with $\tilde{\boldsymbol{U}}_{h_f}$.

Because $\boldsymbol{R}_{h_f}(\boldsymbol{U}_{h_f}) = 0$, an estimation of the error vector is obtained by symbolically inverting Eq. (4):

$$\delta \boldsymbol{U}_{h_f} \approx -\left[\left.\frac{\partial \boldsymbol{R}_{h_f}}{\partial \boldsymbol{U}_{h_f}}\right|_{\tilde{\boldsymbol{U}}_{h_f}}\right]^{-1} \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) \tag{5}$$

Substituting the error vector into Eq. (3), the following estimation for the functional is obtained:

$$J_{h_f}(\boldsymbol{U}_{h_f}) \approx J_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) - \boldsymbol{\Psi}_{h_f}^T \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) \tag{6}$$

where the adjoint solution $\boldsymbol{\Psi}_{h_f}$ has been introduced, and T stands for transpose. This adjoint solution satisfies the following discrete linear system:

$$\left[\left.\frac{\partial \boldsymbol{R}_{h_f}}{\partial \boldsymbol{U}_{h_f}}\right|_{\tilde{\boldsymbol{U}}_{h_f}}\right]^T \boldsymbol{\Psi}_{h_f} = \left[\left.\frac{\partial J_{h_f}}{\partial \boldsymbol{U}_{h_f}}\right|_{\tilde{\boldsymbol{U}}_{h_f}}\right]^T \tag{7}$$

Note that the relative discretization error can be obtained from expression Eq. (6) simply by subtracting the functional output $J_h(\boldsymbol{U}_h)$ from both sides:

$$J_{h_f}(\boldsymbol{U}_{h_f}) - J_h(\boldsymbol{U}_h) \approx (J_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) - J_h(\boldsymbol{U}_h)) - \boldsymbol{\Psi}_{h_f}^T \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) \tag{8}$$

The relative discretization error estimated by Eq. (8) would require the solution of the adjoint system Eq. (7) on the fine mesh Ω_{h_f}, which is computationally expensive. In fact, if no special iterative method is used to solve it, the required CPU time is of the same order as required for the flow solution (known as the primal problem). For this reason, an approximate adjoint solution $\boldsymbol{\Psi}_h^{h_f}$ is introduced in Eq. (6), yielding the following expression for the error estimation:

$$J_{h_f}(\boldsymbol{U}_{h_f}) - J_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) \approx -(\boldsymbol{\Psi}_h^{h_f})^T \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) + (\delta \boldsymbol{\Psi}_h^{h_f})^T \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f}) \tag{9}$$

where $\delta \boldsymbol{\Psi}_h^{h_f} = \boldsymbol{\Psi}_h^{h_f} - \boldsymbol{\Psi}_{h_f}$. In practice, the approximated adjoint solution $\boldsymbol{\Psi}_h^{h_f}$ is obtained by interpolation of the solution of the adjoint equations obtained on the current mesh Ω_h:

$$\left[\frac{\partial \boldsymbol{R}_h}{\partial \boldsymbol{U}_h}\right]^T \qquad \boldsymbol{\Psi}_h = \left[\frac{\partial J_h}{\partial \boldsymbol{U}_h}\right]^T \tag{10}$$

Equation (9) has two terms on the right-hand side: The first one is referred to as the computable correction because an improved value of the functional can be obtained by adding this term to the approximated functional evaluated with $\tilde{\boldsymbol{U}}_{h_f}$. Moreover, this term is computed with solutions coming from the current mesh, which are supposed to be known. The second one is known as the remaining error term, $(\boldsymbol{\Psi}_h^{h_f} - \boldsymbol{\Psi}_{h_f})^T \boldsymbol{R}_{h_f}(\tilde{\boldsymbol{U}}_{h_f})$. It takes into account the error produced by approximating the fine mesh adjoint solution by a reconstruction from the coarse mesh. This term is unknown because the adjoint solution on the fine mesh is never computed. Nevertheless, retaining this term in the error representation formula may be useful because, as we will show later, it can be used to define a suitable adjoint-based adaptation parameter.

For the rest of the paper, the discrete interpolation operators from the coarse mesh flow and adjoint solutions onto the fine mesh are denoted as $\hat{\boldsymbol{I}}_h^{h_f}$ and $\boldsymbol{I}_h^{h_f}$, respectively; for example,

$$\tilde{\boldsymbol{U}}_{h_f} = \hat{\boldsymbol{I}}_h^{h_f} \boldsymbol{U}_h, \qquad \boldsymbol{\Psi}_h^{h_f} = \boldsymbol{I}_h^{h_f} \boldsymbol{\Psi}_h$$

Thus, from Eq. (6), a corrected value of the functional can be computed from the primal and adjoint coarse mesh solution by

$$J_{h_f}(\boldsymbol{U}_{h_f}) \approx \tilde{J}_{h_f}(\boldsymbol{U}_h, \boldsymbol{\Psi}_h) = J_{h_f}(\hat{\boldsymbol{I}}_h^{h_f} \boldsymbol{U}_h) - (\boldsymbol{I}_h^{h_f} \boldsymbol{\Psi}_h)^T \boldsymbol{R}_{h_f}(\hat{\boldsymbol{I}}_h^{h_f} \boldsymbol{U}_h) \tag{11}$$

The coarse-to-fine prolongation operators $\hat{\boldsymbol{I}}_h^{h_f}$ and $\boldsymbol{I}_h^{h_f}$ used in Eq. (11) can be based on linear or high-order interpolation. In the present work, a linear projection has been used to estimate the computable correction. It has been determined in a previous work [29] that a higher-order interpolation does not offer significant gain in the accuracy of the corrected functional compared to a simpler linear interpolation.

Observe that the adjoint approach estimates the functional on a refined mesh with the additional cost of solving the adjoint equations on the original mesh and one residual evaluation on the fine mesh. Although this is quite attractive, there are three drawbacks that make this technique computationally expensive. The first one is that the fine mesh has to be generated and stored explicitly to avoid significative modifications of the base flow solver. The second one concerns the adjoint linear system, which possesses the same stiffness as the primal equations (same eigenvalues). Therefore, robust and efficient iterative methods are required for solving the discrete adjoint equations, specifically in the case of turbulent flows. The third drawback is that the adjoint solutions are not independent of the functional to be analyzed, and therefore it must be recomputed any time a new functional is studied unless a multitarget error estimation [30] is considered.

B. Error Indicator by τ Estimation Method

Let us consider again the discretization of a partial differential equation on a grid Ω_h indexed by a mesh size parameter h [Eq. (2)]:

$$\boldsymbol{R}_h(\boldsymbol{U}_h) = 0$$

The corresponding local truncation error is defined as

$$\boldsymbol{R}_h(\boldsymbol{I}^h \boldsymbol{U}) = \boldsymbol{\tau}^h \tag{12}$$

where $\boldsymbol{I}^h$ represents a linear continuum-to-grid Ω_h transfer for the specified exact solution $\boldsymbol{U}$. Therefore, the truncation error can be seen as the residual left by the exact solution when applied onto the discretized PDE.

In addition to the discrete equation Eq. (12), and considering an FAS multigrid algorithm [14], the coarse-grid equation may be written as

$$\begin{aligned} \boldsymbol{R}_{h_c}(\hat{\boldsymbol{U}}_{h_c}) &= \boldsymbol{R}_{h_c}(\hat{\boldsymbol{I}}_h^{h_c} \tilde{\boldsymbol{U}}_h) - \boldsymbol{I}_h^{h_c}(\boldsymbol{R}_h(\tilde{\boldsymbol{U}}_h)) \\ \hat{\boldsymbol{U}}_{h_c} &= \hat{\boldsymbol{I}}_h^{h_c}(\boldsymbol{\epsilon}_{it}^h + \tilde{\boldsymbol{U}}_h) \end{aligned} \tag{13}$$

corresponding to the discrete equation on a coarser mesh Ω_{h_c}, with mesh ratio $\rho = h/h_c < 1$. In Eq. (13), $\tilde{\boldsymbol{U}}_h$ is the current approximation of the solution (relaxed on the fine grid and not necessarily converged); $\boldsymbol{\epsilon}_{it}^h = \boldsymbol{U}_h - \tilde{\boldsymbol{U}}_h$ is the fine grid iteration error whose high frequencies have to be smoothed; and $\hat{\boldsymbol{I}}_h^{h_c}$ represents the fine-to-coarse transfer operator of the solution, whereas $\boldsymbol{I}_h^{h_c}$ represents the fine-to-coarse transfer operator of the residual. Similarly, introducing the relative local truncation error $\boldsymbol{\tau}_h^{h_c}$, Eq. (13) may be written as

$$\boldsymbol{R}_{h_c}(\hat{\boldsymbol{U}}_{h_c}) = \boldsymbol{\tau}_h^{h_c} \tag{14}$$

with

$$\boldsymbol{\tau}_h^{h_c} = \boldsymbol{R}_{h_c}(\hat{\boldsymbol{I}}_h^{h_c} \tilde{\boldsymbol{U}}_h) - \boldsymbol{I}_h^{h_c}(\boldsymbol{R}_h(\tilde{\boldsymbol{U}}_h)) \tag{15}$$

Our goal is to use $\boldsymbol{\tau}_h^{h_c}$ to estimate $\boldsymbol{\tau}^{h_c}$. If this can be done with sufficient accuracy, then one can use this local error as a mesh

adaptation indicator or an uncertainty estimator or to increase the order of accuracy of the spatial scheme.

In this study, we consider that the fine grid solution is converged. In this case, $\tilde{\boldsymbol{U}}_h = \boldsymbol{U}_h, \boldsymbol{\epsilon}_{it}^h = 0$, and the second right-hand-side term of Eq. (15) can be neglected.

The following theorem provides the relation between the accuracy of $\boldsymbol{\tau}_h^{h_c}$ toward $\boldsymbol{\tau}^{h_c}$ and the order of the restriction operators acting in Eq. (15).

Theorem 1 (Truncation Error Estimate): Assume that there exists $p \geq 1$ and q, $s \geq 1$ such that, if $\boldsymbol{U} \in \mathcal{C}^{n+p+q}(\Omega)$ and the following assumptions hold:

1) The local truncation error is of order p: $\boldsymbol{\tau}^h = h^p \boldsymbol{I}^h \boldsymbol{V} + \mathcal{O}(h^{p+q}), \boldsymbol{V} \in \mathcal{C}^q(\Omega)$.

2) The local discretization error is of order p: $\boldsymbol{\epsilon}^h = h^p \boldsymbol{I}^h \boldsymbol{W} + \mathcal{O}(h^{p+q}), \boldsymbol{W} \in \mathcal{C}^q(\Omega)$.

3) The fine-to-coarse transfer operator of the solution is of order s: $\hat{\boldsymbol{I}}_h^{h_c} \boldsymbol{I}^h \boldsymbol{U} = \boldsymbol{I}^{h_c} \boldsymbol{U} + \mathcal{O}(h_c^s)$; then

$$\boldsymbol{\tau}_h^{h_c} = (1 - \rho^p)\boldsymbol{\tau}^{h_c} + \mathcal{O}(h_c^{\min(s,p+q,2p)}), \qquad \rho = h/h_c \tag{16}$$

The main conclusion of Eq. (16) is related with the order of the restriction operator. Looking at the exponent of the order of magnitude of the remaining term in Eq. (16), it can be deduced that it is necessary to use higher-order interpolation $s > p$ to transfer the solution from fine to coarse mesh. If $s \leq p$, then the truncation error estimation will be dominated by a term $\mathcal{O}(h_c^s)$, spoiling the general results of the formula. Proofs to this theorem, together with a detailed analysis of the truncation error in general geometries and schemes, can be found in Fraysse et al. [27].

Our aim in this work is to test the capabilities of a mesh-adaptation algorithm, driven by truncation error distribution, to improve the accuracy of engineering functional outputs. This study will focus on a posteriori adaptation of steady flows. Furthermore, as it will be detailed later, the coarse grid employed for the computation of the truncation error estimate, on which the fine grid solution is restricted, is built in such a way that each fine grid node coincides to a coarse grid node. In this way, no complex restriction operator has to be considered to recover full accuracy of the estimator, and direct injection (equivalent to $s = \infty$) can be performed like in a finite-differences formulation.

III. Mesh Adaptation Methodology

The methods described here have been implemented and checked in the industrial triangular adaptive upwinding (TAU) code of DLR, German Aerospace Center [31]. In TAU, adaptation is performed by bisecting the edges of an element according to some specific sensor. In this section, we will describe three different methodologies/sensors used to perform the mesh adaptation.

A. Feature-Based Adaptation Methodology

A widely used family of mesh adaptation methods is based on local physical sensors. These methods are usually known as the feature-based sensors. In the current work, the feature-based sensor is constructed from the differences of the flow variables on each side of the edges.

To select zones to be refined, it is first determined which edges of the primary grid have to be adapted depending on the desired dimensions for the resulting grid.

Thus, for all edges, an adaptation parameter I_f is defined. Only the edges with values of I_f above some specific threshold are bisected. I_f is defined as

$$I_f = \Delta V_e \|\boldsymbol{x}_e\|_2^{\alpha}$$

where $\|\cdot\|_2$ is the usual Euclidean norm, $\boldsymbol{x}_e = \boldsymbol{x}_{p_1} - \boldsymbol{x}_{p_2}$ is the vector edge, α is an edge length scaling factor (typically set to $\alpha = 0.5$), and

$$\Delta V_e = \max\left(c_{\phi_i} \frac{\Delta\phi_i}{(\Delta\phi_i)_{\max}}\right)$$

with

$$0 \leq i < N$$

and N being the number of different flow variables considered. In this study, ϕ_i is an equal combination of density, velocity module, total enthalpy, and total pressure. This particular choice proved to be suitable to the widest range of test cases and is the DLR TAU code default. Thus,

$$\Delta\phi_i = |\phi_i(\boldsymbol{x}_{p_1}) - \phi_i(\boldsymbol{x}_{p_2})|$$

The weights c_{ϕ_i} are parameters enabling to choice of different combinations of the single parts of the indicator, fixed in this work to $c_{\phi_i} = 1$.

Finally, for an equilibrated scaling of each part, the calculated values must be distributed in [0, 1]. Therefore, the maximum of all values is determined by

$$(\Delta\phi_i)_{\max} = \max((\Delta\phi_i)_e)$$

for all edges e.

B. Adjoint-Based Adaptation Methodology

Within the adjoint error-estimation framework, the adaptation indicator may be based on the first or second terms, respectively, on the right-hand side of Eq. (9): the local value of the computable correction [32] or the relative error of the corrected functional, which is somehow represented by the remaining error term [33]. In the former, the idea behind the adaptive process is to obtain adapted meshes in which the functional is computed accurately. In the latter, the goal is to adapt the meshes to minimize the error after the adjoint correction is done, so that the method is a combination between error estimation and mesh adaptation, but the computable error must not necessarily go to zero as the adaptation process progresses. A comparison between these two adaptation parameters has been performed previously [29], and it has been checked that adaptive parameters based on the remaining error lead to more economical meshes (in terms of final number of nodes). Therefore, this magnitude has been selected as a driving adaptation parameter for this investigation.

Thus, for each element i, the following indicator is defined:

$$\boldsymbol{\eta}_i^{h_f} = |(\delta\boldsymbol{\Psi}_{h_f}^h)_i^T \boldsymbol{R}_{h_{fi}}(\tilde{\boldsymbol{U}}_{h_f})| \tag{17}$$

Because the fine mesh adjoint solution is never computed, it must be approximated in some way. The usual approach [33] is to compute this term as the difference between two different interpolations (from $\boldsymbol{\Psi}_h$): a linear one and a higher-order one based on either compact edge cubic Hermite polynomials [34,35] or a patch least-square reconstruction using quadratic finite elements [36]. In the present investigation, we have used an edgewise Hermite interpolation because we found it sufficiently accurate [29] and computationally cheap. In this way, the local adaptation parameter for an element i of the fine mesh is defined by

$$\boldsymbol{\eta}_i^{h_f} = |(\boldsymbol{Q}_h^{h_f}\boldsymbol{\Psi}_h - \boldsymbol{L}_h^{h_f}\boldsymbol{\Psi}_h)_i \boldsymbol{R}_{h_{fi}}(\boldsymbol{L}_h^{h_f}\boldsymbol{U}_h)| \tag{18}$$

where $\boldsymbol{Q}_h^{h_f}$ and $\boldsymbol{L}_h^{h_f}$ are the high-order and linear interpolant operators, respectively. Once each local component of the error bound is computed on the fine grid, it is transferred back to the coarse mesh in such a way that the global error bound is conserved; that is,

$$E_h = \sum_{i \in \Omega_h} \boldsymbol{\eta}_i^h = \sum_{j \in \Omega_{h_f}} \boldsymbol{\eta}_j^{h_f} = E_{h_f} \tag{19}$$

A fine-to-coarse grid operator, $\boldsymbol{P}_{h_f}^h$, which takes into account the old and new nodes surrounding each coarse node, has been used to fulfill the previous condition.

The DLR TAU code adaptation module has been modified to include the interpolation operators described before. This module is

used twice within each adaptation cycle: first to create explicitly the true embedded fine mesh of the actual coarse one and perform the interpolations of the coarse mesh solutions, and second to refine the coarse mesh from the selected points for adaptation, once the local error bound has been computed and transferred back to the coarse mesh.

Simple equidistribution of error has been proven to be a very efficient strategy to be used in conjunction with the adjoint methodology [33,35]. It can be shown that, under some simple assumptions, this method provides final optimal meshes with regard to the global error [37]. The goal of this strategy is to distribute equally the local error on the adapted grid with the constraint that the estimated error bound (given either by the computable correction or the remaining error) is below some user prescribed tolerance. If N_h is the current number of nodes of the coarse mesh, the admissible error is then given by

$$\epsilon = \frac{\mathrm{TOL}}{N_h}$$

A node (cell) i is marked for refinement if the local error is above the error threshold, that is,

$$I_{\mathrm{ad}} = \frac{\eta_i^h}{\epsilon} \geq 1 \tag{20}$$

C. Truncation-Error-Based Adaptation Methodology

To compute an accurate estimation of the local truncation error, it is necessary to dispose of two grids, which are topologically consistent [27]. For that purpose, we denote the coarse grid by Ω_{h_c} and the fine grid by Ω_h (which is the actual grid where the flow is solved), obtained by uniform refinement of grid Ω_{h_c}. Constructing the fine grid Ω_h from the coarse grid Ω_{h_c} gives two major advantages in the scope of truncation error estimation. On one hand, the conservation of grid characteristics (nonuniformities, distortion, etc.) ensuring that the computed error on Ω_{h_c} is truly representative of the error on Ω_h (in the asymptotic range). On the other hand, each fine grid point coincides with a coarse grid point, which avoids implementing a high-order restriction of the fine grid solution onto the coarse grid (see the discussion at the end of Sec. III.D).

Thus, following Eq. (15) and neglecting the second right-hand-side term (converged solution), the truncation-error-based sensor developed in this work is defined as:

$$I_{te_i} = \frac{|\Omega_{h_{ci}}|^{\alpha}}{1-\rho^p}\left|\tau_{h_i}^{h_c}\right|, \qquad i \text{ node index} \tag{21}$$

where

1) $|\Omega_{h_{ci}}|$ is the volume of the element associated with node i. This scaling prevents performing infinite refinements in regions where the error might decrease at a very low rate, or even diverge (typically, shock waves where $\|\boldsymbol{\tau}_h^{h_c}\|_{L_2} = \mathcal{O}(h_c^{-1/2})$, see Sonar [38]) without substantially contributing to the prescribed functional.

2) α is a volume-scaling factor $\alpha = 0.5$.

3) $\rho = h/h_c$ is a characteristic grid-length ratio from grid Ω_h to grid Ω_{h_c}, set here to $\rho = 0.5$.

4) p is the formal order of accuracy set to $p = 2$ using the Jameson–Schmidt–Turkel (JST) spatial scheme.

5) $\tau_{h_i}^{h_c}$ is the truncation error at node i, obtained by applying Eq. (15):

$$\boldsymbol{\tau}_{h_i}^{h_c} = \boldsymbol{R}_{h_{ci}}^{q}(\hat{\boldsymbol{I}}_h^{h_c}\boldsymbol{U}_{h_i})$$

where $\boldsymbol{R}_{h_{ci}}^{q}$ is the residual associated with the equation q (density, momentum, or total energy). In these computations, the truncation error associated with the density $q = 1$ has been considered in all the test cases. No significant differences in the adapted meshes have been observed if the momentum or energy equations are considered in the computation of the sensor.

1) $\hat{\boldsymbol{I}}_h^{h_c}$ is a restriction operator from grid Ω_h to grid Ω_{h_c}. As explained earlier, the way we construct Ω_h and Ω_{h_c} allows the use of direct injection ($s = \infty$).

2) $\boldsymbol{U}_h$ is the converged solution obtained on grid Ω_h.

A point i on the grid Ω_{h_c} is then flagged for refinement if

$$I_{te_i} \geq \lambda$$

where λ is a user-given threshold that controls the tradeoff between increased accuracy and increased work. Typical values of this parameter are $\mathcal{O}(h_c^3)$.

Based on this indicator strategy, one cycle of adaptation is performed in this way:

1) The solution $\boldsymbol{U}_h$ of the primal equations is computed on grid Ω_h.

2) A pointwise restriction is applied over the solution $\boldsymbol{U}_h$ from grid Ω_h to grid Ω_{h_c}.

3) The truncation-error-based sensor of Eq. (21) is computed on grid Ω_{h_c}.

4) The truncation error is smoothed by a few Laplacian iterations.

5) A file containing the indices of the sensor nodes of grid Ω_{h_c}, where the indicator is above the user-given threshold λ, is written.

6) A full refinement of all elements sharing a marked node is applied over the grid Ω_{h_c}.

7) A uniform refinement of grid $\Omega_{h_{\mathrm{cad}}}$ to grid $\Omega_{h_{\mathrm{ad}}}$ is performed.

A key point of this procedure is that the adaptation is performed on the coarse grid Ω_{h_c}. Then, the final adapted grid $\Omega_{h_{\mathrm{ad}}}$ is obtained by uniform refinement of grid $\Omega_{h_{\mathrm{cad}}}$. This choice ensures that the two grid levels generated are topologically consistent, which allows computing again the truncation error for a further adaptation cycle, thus avoiding an extra source of error due to inconsistency between meshes.

D. Implementation Issues

The adjoint methodology involves the use of three components: the (primal) flow solver, the discrete adjoint solver, and the adaptation module. The primal solver used for the present investigation is DLR TAU code [31], which is used as a standard code within the European aeronautical industry. The DLR TAU code solves the Reynolds-averaged Navier–Stokes equations on unstructured hybrid grids employing a second-order finite-volume discretization. Convective terms are discretized with central fluxes and a scalar JST numerical dissipation model. Other discretization models are available for the convective fluxes, but the JST model is still the most common option for industrial applications for moderate transonic flows. A discrete adjoint equation solver is available for the DLR TAU code [39,40] and JST scheme. The adjoint equations are solved iteratively either by an adjoint consistent Runge–Kutta/LU–SGS scheme or by employing Krylov iterative methods. In each adaptation cycle, the adaptation module is employed twice: first to generate explicitly a uniformly refined mesh, and second to adapt the elements whose edges have been selected for adaptation after the adjoint error estimation has been done. An edge of the original grid is marked for adaptation when both end nodes have been previously marked, i.e, if both end nodes fulfills the condition of Eq. (20). In the present study, the adaptation process is performed for a prescribed number of adaptation cycles.

As far as the adaptation based on truncation error estimation is concerned, the adaptation module is used three times. Once the primal solver has computed a steady solution, it is restricted pointwise to the coarse grid by the derefinement procedure of DLR TAU code adaptation module. Once the error indicator has been computed on that coarse grid, it is adapted. The refinement procedure used in the frame of this work is based on full refinement of all elements sharing a marked node. This procedure yields smoother adapted grids but significantly increases the number of nodes at each adaptation step with respect to edge selection, which is used in the adjoint and feature-based methodologies. It is also not fully optimized with respect to the edge-based construction of the DLR TAU code; thus the latter extension is in current progress. After computing the coarse adapted grid, a last call to the adaptation

module is done in uniform refinement mode to compute the final fine adapted grid. This procedure naturally provides two consistent grids (in the sense of element orientation and volume ratios between fine and coarse grid), which is necessary for an accurate estimation of the truncation error but requires an initial coarse grid from which the fine initial grid is obtained by uniform refinement.

An alternative and cheaper approach is to use the multigrid agglomeration algorithm, already implemented in most of the industrial solvers and in particular in the DLR TAU code, which naturally generates coarse grids in the multigrid cycle. In this case, the agglomeration algorithm must preserve the topology and the uniformity between fine and coarse meshes, which avoids the introduction of new sources of error due to the agglomeration method and not present in the original meshes. This is usually normal for structured meshes but more complex to fulfill for unstructured ones, which is the case of the DLR TAU code. The quantification of the additional error committed, in the estimation of the truncation error, and its impact on the adaptation, is not a trivial task and is on current development.

Finally, special care has to be taken into account with the wall boundaries treatment. In our case, a cubic spline reconstruction of new nodes inserted on wall boundaries has been used for all approaches.

IV. Numerical Results

To test and compare the adaptation techniques presented in the last sections, we will present in the following a set of two- and three-dimensional test cases representative of different flow regimes.

In this work, the assessment of the numerical indicators will be limited to Euler computations. Our aim is to demonstrate the effectiveness of the error indicators in simple configurations where numerical errors are more delimited from other sources of errors. Navier–Stokes computations involve some complex issues concerning grid adaptation techniques (grid anisotropy, node projection, turbulence models, associated stiffness of the primal/adjoint systems, etc.), which must be treated carefully. These issues will be considered in a future work.

The two-dimensional test cases considered here will be computed on a NACA0012 airfoil. To cover most of the phenomena existing in inviscid compressible flows, three different cases will be analyzed: at subsonic, transonic, and high-transonic regimes. Finally, a three dimensional ONERA M6 wing at transonic regime will be investigated.

For the present investigation, we specified error tolerances: 1) lift coefficient accuracy ($\mathrm{TOL}(C_L) = 0.01$), 2) drag coefficient accuracy ($\mathrm{TOL}(C_D) = 0.001$), and 3) pitching moment coefficient accuracy ($\mathrm{TOL}(C_{M_y}) = 0.001$. These tolerances have been specified within the European Union project ADIGMA as representative of industrial requirements [41].

Table 1 Far-field conditions

Test case	Mach number	Angle of attack α, deg
Subsonic	0.4	5
Transonic	0.8	1.25
High transonic	0.95	0

The results of adjoint-based and truncation-error-based adaptation will be compared against the common feature-based adaptation technique. We will present a study of the evolution of the force coefficients with respect to the number of nodes for the three adaptation techniques discussed earlier.

The force coefficients C_{F_i} and moment coefficients C_{M_i} are defined as

$$C_{F_i} = \frac{F_i}{\frac{1}{2}\rho v^2 A}, \qquad C_{M_i} = \frac{M_i}{\frac{1}{2}\rho v^2 A l} \tag{22}$$

where F_i and M_i are the components of the integral on the wall force and momentum, respectively, and ρ, v, A, and l are the reference density, velocity, representative area, and length, respectively. The standard aerodynamic force coefficients C_L, C_D, and C_{M_y} are always given in the aerodynamic coordinate system.

A. Inviscid Two-Dimensional Test Cases

In this section, the performance of the different error sensors described previously is demonstrated for the case of inviscid flow over a NACA 0012 profile. Euler computations are done in three flow regimes: subsonic, transonic, and transonic in near sonic conditions. The far-field flow conditions are summarized in Table 1. Each of these flow regimes presents particular flow features that yield different answers from the described error sensors. Results on these test cases have been reported previously in the context of feature-based [4] and adjoint-based [36] grid adaptation. Here, we extend the study to the new proposed truncation-error-based adaptation sensor, and comparisons are done with the adjoint-based and the feature based approaches.

These three test cases share the same initial grid (see Fig. 1), which is composed of 11,750 nodes, 23,040 triangles, and 459 points on the airfoil. The outer boundary is located at a distance of 50 chords away from the airfoil profile. This initial grid is obtained by isotropic uniform refinement of an initial coarse mesh. The existence of both meshes is necessary to start the truncation-error based adaptive process described in Sec. III.D.

1. *Subsonic Flow Conditions*

First, a subsonic Euler flow is considered. The far-field conditions imposed are $M_\infty = 0.4$ and $\alpha = 5$ deg, which yield a fully subsonic steady solution (see Fig. 2).

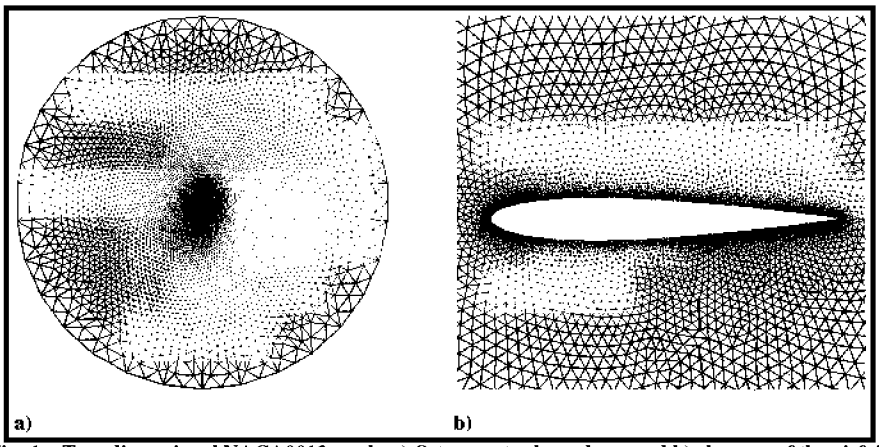

Fig. 1 Two-dimensional NACA0012 mesh. a) O-type outer boundary, and b) close-up of the airfoil.

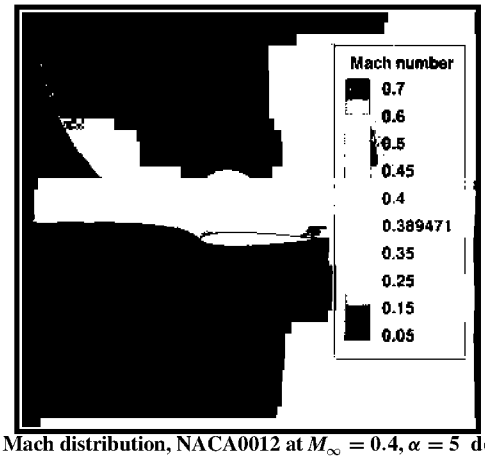

Fig. 2 Mach distribution, NACA0012 at $M_\infty = 0.4, \alpha = 5$ deg, initial mesh.

Two different functionals are analyzed in this case: the lift C_L and y moment C_{M_y}. For the truncation error indicator, the sensor for both functionals is based on density, whereas for the feature-based indicator the sensors for both functionals are based on an equally balanced combination of density, velocity module, total pressure, and enthalpy. As far as the adjoint based sensor is concerned, two different adjoint solutions must be computed. The lift tolerance for the adjoint-based sensor is prescribed to 0.01, which is representative of the required accuracy in the aerospace industry. For this test, the truncation error sensor λ is fixed to 0.0001.

The adapted meshes are presented in Fig. 3 after four cycles of adaptation. The three methods clearly create different meshes. The adjoint C_L-based adaptation concentrates nodes on regions where the discretization errors may have influence on the computation of the lift. These regions are the airfoil trailing edge and especially the upstream region of the airfoil. This behavior is typical of the adjoint-based approach, where the adjoint solution has a singularity at the incoming stagnation streamline, and hence this area is usually detected for refinement by the adjoint-weighted methods. The truncation-error-based indicator creates a mesh that is highly refined at the leading edge and at the trailing edge. Finally, the feature-based adaptation procedure refines more uniformly around the profile and especially in the region where the flow accelerates (on the suction side close to the leading edge). Note that the differences in the extension of the adapted areas are significant, especially for the feature-based method, for which a large area of the suction side is adapted.

It is remarkable that some new zones are refined away from the profile. The explanation resides in the fact that both adjoint and truncation error sensors are a measure of the numerical error (functional discretization and truncation error, respectively), which is strongly related to mesh distortion. This phenomena is caused by the local nature of the refinement procedure, which locally creates nonuniformities that might increase the error up to $\mathcal{O}(1)$ [42]. A remedy would be to impose some postadaptation smoothing iterations.

Finally, in Fig. 4, the C_L and C_{M_y} coefficients have been computed at each adaptation cycle for the three methodologies and compared against uniform refinement and the value obtained by Richardson extrapolation. In the case of adjoint-based adaptation, the indicator must be determined for each functional separately. This implies the solution of the corresponding adjoint system, unless special treatment for simultaneous multiple targets is used [30,43]. The prescribed tolerance for the pitching moment coefficient is TOL $= 0.001$. Two curves are presented for the adjoint-based sensor, with and without correction term; see Sec. III.A, Eq. (11). The three adaptation indicators yield an improved prediction of C_L and C_{M_y} with respect to the initial mesh. The three methods allow a rapid convergence of C_L and C_{M_y}, but after some adaptation cycles they start to stall. In the case of the adjoint-based indicator, the computable correction does not improve in the last two adaptation cycles. As a

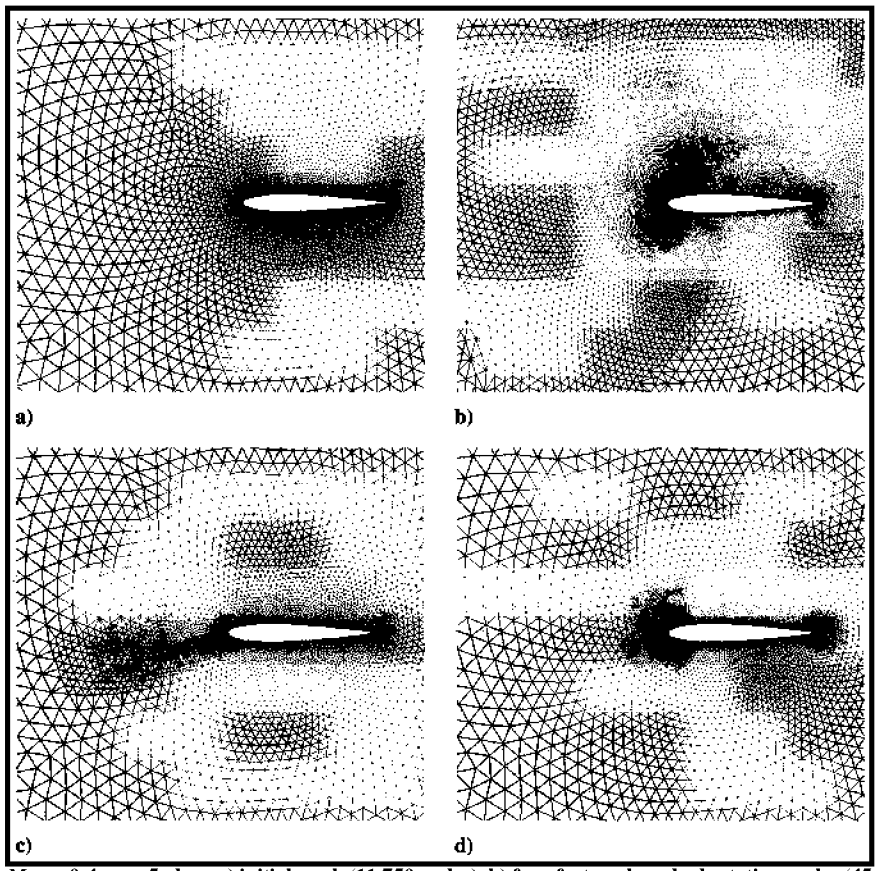

Fig. 3 NACA0012 grids, $M_\infty = 0.4, \alpha = 5$ deg: a) initial mesh (11,750 nodes), b) four feature-based adaptation cycles (45,214 nodes), c) four adjoint C_L-based adaptation cycle (16,521 nodes), and d) four TE-based adaptation cycles (40,211 nodes).

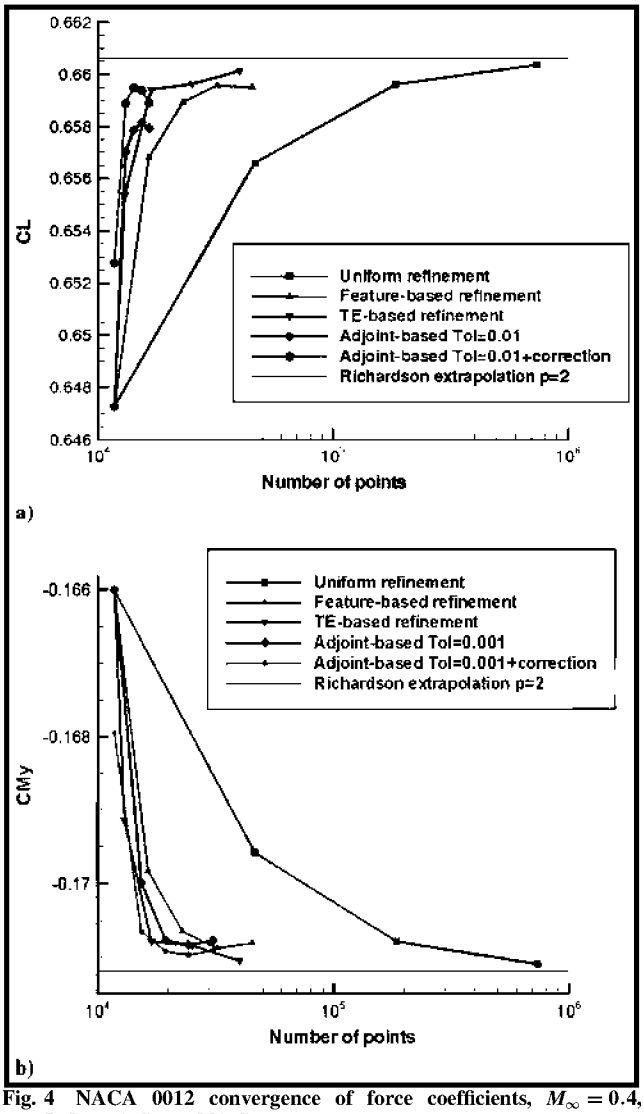

Fig. 4 NACA 0012 convergence of force coefficients, $M_\infty = 0.4$, $\alpha = 5$ deg: a) C_L, and b) CM_y.

result, the number of added points is small in these adaptation cycles, and consequently the adaptation process saturates. If the required lift tolerance is reduced (see Fig. 5), the convergence of the lift with the adjoint-based method improves reaching almost the asymptotic level. However, this comes at the cost of increasing the number of added points. Concerning the truncation error sensor, the last two

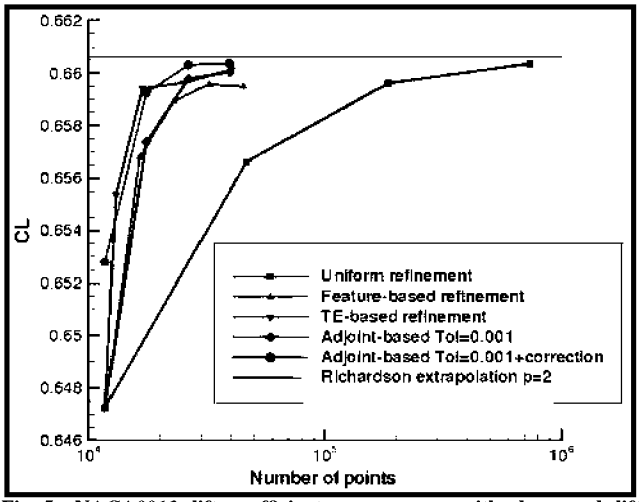

Fig. 5 NACA0012 lift coefficient convergence with decreased lift tolerance, $M_\infty = 0.4$, $\alpha = 5$ deg.

adaptation cycles give almost no improvement of the functionals with a high increase of the number of nodes. The adjoint-based and the truncation-error-based methods permit a gain of more than an order of magnitude in the number of nodes with respect to uniform refinement, and both yield comparable levels of functional outputs as the feature-based approach but with fewer points.

Regarding computational efficiency, taking as a reference the time required to solve the flow problem in the last uniform grid, the adjoint-based method produces the same level of accuracy employing about 20 times less computing time when a krylov iterative method [44] is used for solving the adjoint equation. In the case where the linear adjoint system is solved with the same iterative method as the flow solver (LU–SGS-smoothed four-level multigrid W cycle), the efficiency decreases to 10%. Instead, the truncation-error-based method is computationally more efficient because there is no need to solve any additional equations or to use an auxiliary embedded mesh to estimate the numerical error.

2. *Transonic Flow Conditions*

The second test is a transonic Euler flow. The far-field conditions imposed are $M_\infty = 0.8$ and $\alpha = 1.25$ deg, which yield a strong shock wave on the suction side and a weaker one on the pressure side (see Fig. 6).

In this case, the functionals studied are the lift C_L and the drag C_D. The appearance of drag in the compressible regime is directly related to the existence of shock waves as well as the consequent total pressure losses and entropy creation. In fact, drag and lift values are very sensitive to the upper shock wave location, which depends on the correct flow resolution at the airfoil trailing edge and upstream shock areas.

The adapted meshes are presented in Fig. 7 after four adaptation cycles. The three methods clearly create different meshes. As in the previous case, the adjoint C_L-based adaptation concentrates nodes on regions where the functional discretization error is high and consequently may have influence on the accuracy of the computation of the lift (e.g., upstream and the trailing edge regions) and a bit less around the shocks. The truncation-error-based indicator creates a mesh that is mainly refined around the upper side shock due to unbounded derivatives and less at the leading and trailing edges. Finally, the feature-based adaptation procedure strongly refines both shocks as well as the wake region up to the far-field boundary due to the contact discontinuity (strongly refined by the total pressure sensor). Similar to the subsonic test case, the feature-based method tends to adapt much larger regions around the airfoil, whereas the numerical error-based methods tend to limit the adaptation to very localized areas. As a result, the numerical error indicators produce more economical meshes in terms of number of points of the final adapted meshes.

After four adaptation cycles, we observe, like the subsonic test case, the appearance of new regions flagged for refinement that were

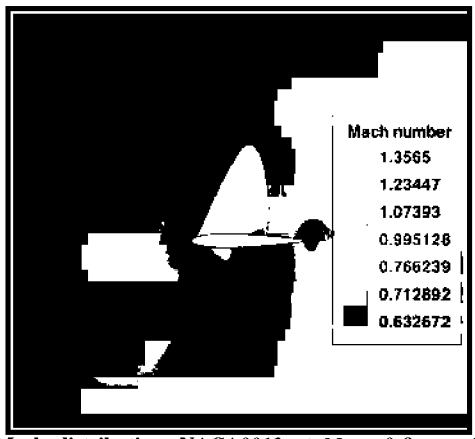

Fig. 6 Mach distribution, NACA0012 at $M_\infty = 0.8$, $\alpha = 1.25$ deg, initial mesh.

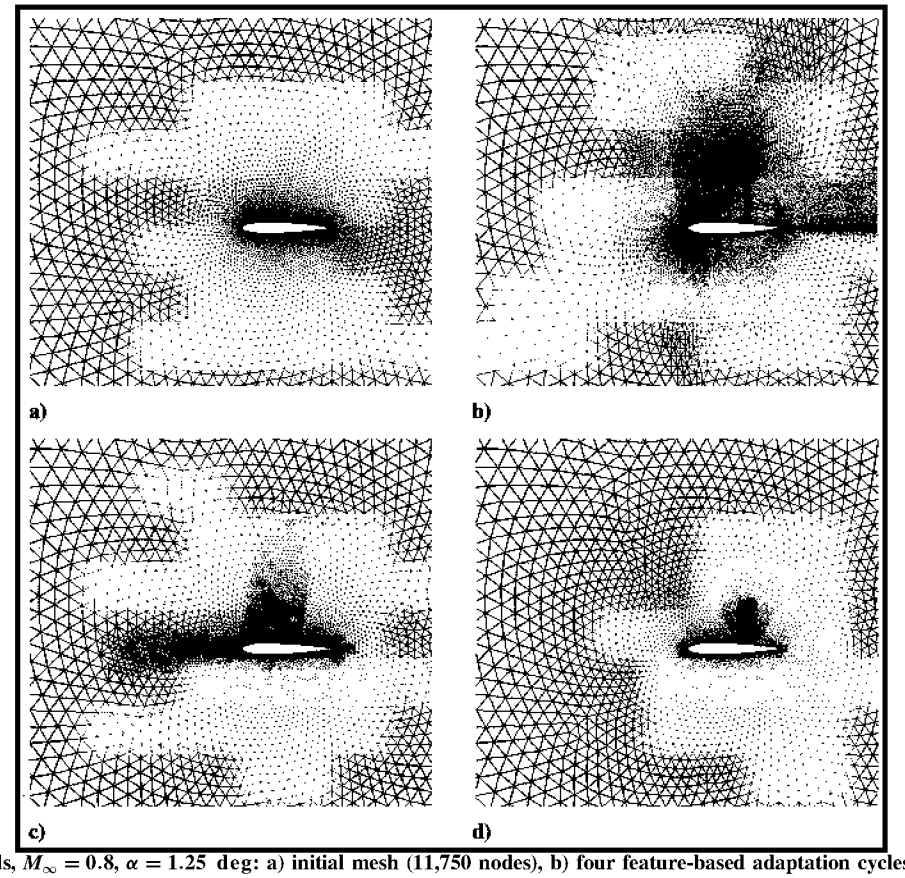

Fig. 7 NACA0012 grids, $M_\infty = 0.8$, $\alpha = 1.25$ deg: a) initial mesh (11,750 nodes), b) four feature-based adaptation cycles (48,466 nodes), c) four adjoint C_L-based adaptation cycle (21,320 nodes), and d) four TE-based adaptation cycles (29,387 nodes).

not present in the initial grid. As previously mentioned, this can be related with the local nonuniformities generated by the refinement process. However, this effect is much less accused for the truncation error approach. In fact, the highest errors are located at the shock making the source of error generated by the grid invisible for this particular test case.

Finally, in Fig. 8, the C_L and C_D coefficients have been computed at each adaptation cycle for the three methodologies and compared against the uniform refinement and Richardson extrapolation. Two curves are presented for the adjoint-based sensor, with and without correction term. The three adaptation indicators yield an improved prediction of C_L and C_D with respect to the initial mesh. Here, the feature-based technique over-refines regions (mainly the wake), which do not strongly influence the accuracy of the force coefficients, thus increasing unnecessarily the number of nodes for a given C_L or C_D with respect to the numerical-error-based approaches. The adjoint-based and the truncation-error-based perform very well and require almost two orders of magnitude fewer points than the globally refined grids for the last adaptation cycle. Here, the truncation-error-based adaptation method is almost competitive with the adjoint-based approach with correction and still much cheaper. It is worthwhile to note again that the truncation error indicator gives correct results for both functionals using only the truncation error distribution without the need of solving additional equations.

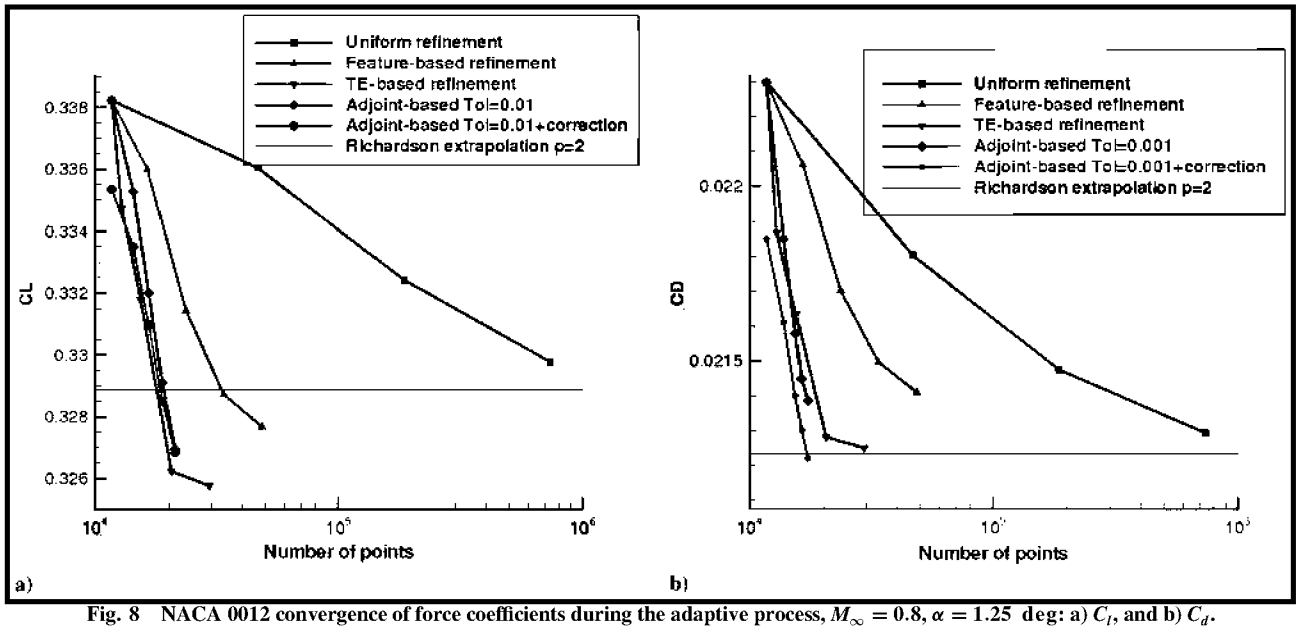

Fig. 8 NACA 0012 convergence of force coefficients during the adaptive process, $M_\infty = 0.8$, $\alpha = 1.25$ deg: a) C_l, and b) C_d.

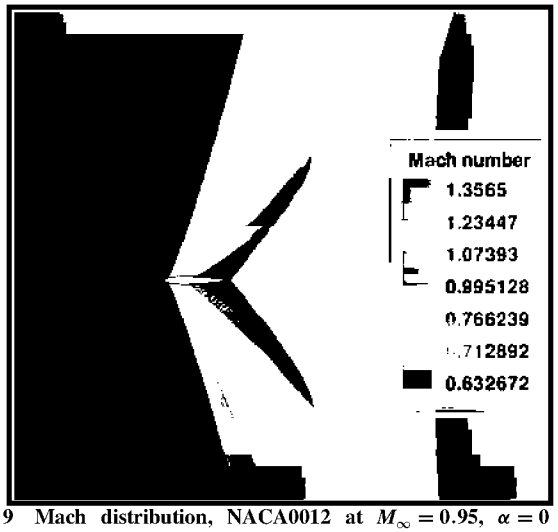

Fig. 9 Mach distribution, NACA0012 at $M_\infty = 0.95$, $\alpha = 0$ deg, initial mesh.

3. *High Transonic Flow Conditions*

Feature-based sensors are widely used in unstructured CFD codes for mesh adaptation for their capability of detecting regions of high gradients (shocks, separation, boundary layers, vortices, etc.) with the objective of refining these regions to improve the accuracy there. The h adaptation based on this idea is commonly implemented in unstructured CFD codes. When a special flow feature (for instance a shock) appears at a surface where one wants to compute a force coefficient, it is natural to think that refining the shock region will improve the functional outputs. However, if a shock occurs far from the profile, physical sensors will refine regions, which might not influence the accuracy of functional outputs.

To prove this fact, a transonic Euler flow near sonic conditions ($M_\infty = 0.95$) is considered. The flow is characterized by the presence of a fishtail shock wave structure emanating from the airfoil's trailing edge (two strong oblique shock waves starting at the trailing edge and a weak vertical one joining both in the wake; see Fig. 9). This test case was studied by Warren et al. [4] in the context of grid-adaptation methods concerning the prediction of the downstream normal shock location. Here, we focus on the effectiveness of grid adaptation for drag coefficient prediction. The flow pattern in this case is quite challenging for grid-adaptation sensors. The main flow features occur far away from the profile, and the refinement of this region does not necessarily have an impact on the functional output. On the contrary, there are some areas where the flow is smooth that can have a large influence on the functionals. These areas are located along the expansion waves emanating from the airfoil on the leading edge region as well as on the airfoil trailing edge zone just upstream of the location of the oblique shock waves. This test case has proven difficult for adaptation methods because the possible numerical errors produced at upstream locations are transported downstream; thus local error-based methods (which assume implicitly that transported error components are small) might fail at detecting the regions that affect global functional outputs (such as lift and drag) the most.

In this test, only the C_D has been considered. As expected, the feature-based sensor highly refines the downstream oblique shocks (see Fig. 10b). The truncation-error-based sensor is also sensitive to the shocks (see Fig. 10d) but also refines the leading edge. Note that the error distribution is asymmetric, whereas the solution is symmetric. This shows the high dependence of the truncation error with respect to the grid, as it can be verified that using a perfectly symmetric grid yields a symmetric error distribution. The adjoint-based methodology predicts a very different error distribution in which the oblique shock waves are not refined at all, indicating that these regions do not affect the computation of the drag.

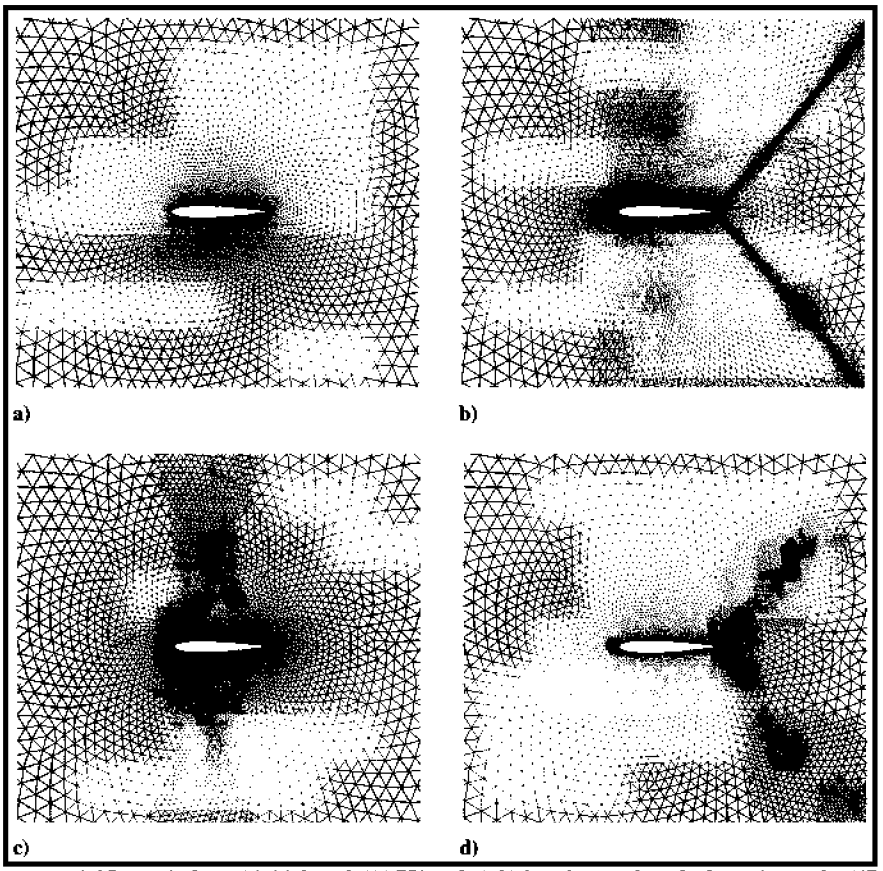

Fig. 10 NACA0012 grids, $M_\infty = 0.95, \alpha = 0$ deg: a) initial mesh (11,750 nodes), b) four feature-based adaptation cycles (47,897 nodes), c) four adjoint C_D-based adaptation cycle (19,426 nodes), d) four TE-based adaptation cycles (40,614 nodes).

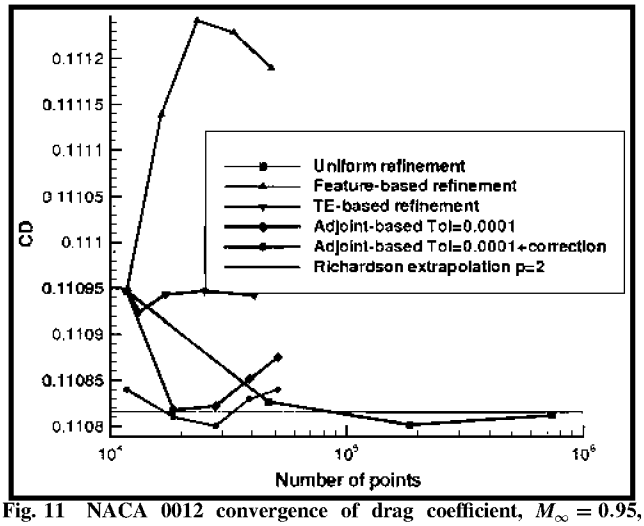

Fig. 11 NACA 0012 convergence of drag coefficient, $M_\infty = 0.95$, $\alpha = 0$ deg.

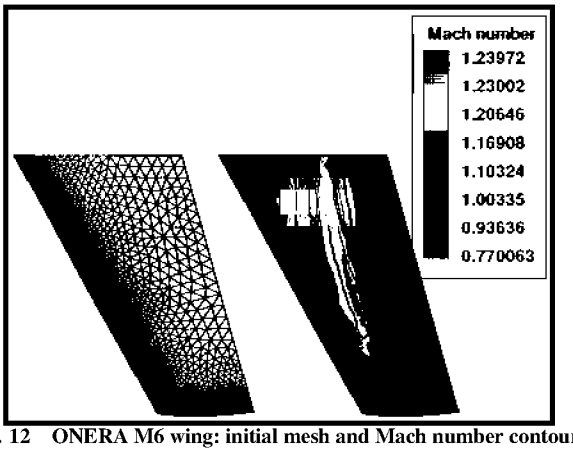

Fig. 12 ONERA M6 wing: initial mesh and Mach number contours.

Figure 11 shows the C_D coefficient computed at each adaptation cycle for the three methodologies and compared against the uniform refinement and Richardson extrapolation results. Note that, in this case, the adjoint C_D tolerance was lowered to 0.0001 because the initial mesh already yields a drag value clearly below industrial tolerance [41]. Assuming that the correct value is obtained by uniform refinement, the featured based method gives a totally wrong result. The refinement procedure gives worse results than the one obtained with the original mesh, although, in any case, only four drag counts of maximum difference are observed. The truncation-error-based method gives almost no significative differences between the different meshes, less than half a drag count. It is obvious that a better resolution of the shocks does not have any influence in the drag computation accuracy. Finally, the adjoint based method shows a better performance in this test case, in that the error tendency follows that corresponding to uniform refinement, although it seems to deteriorate after two adaptation cycles, whereas the differences with the other estimators are still very low, below one drag count. Besides the quantitative irrelevance of the values, the conclusion is that this test shows the main differences between adjoint and truncation error methods. The former is clearly dedicated to a functional and can detect the regions of influence more efficiently although at a higher computational cost, whereas the latter is more general and can be applied to a broad spectrum of problems.

B. Inviscid Three-Dimensional Transonic Case

To assess the performance of these algorithms in three-dimensional problems, a final calculation has been performed on the ONERA M6 wing at a freestream Mach number of $M_\infty = 0.84$ and an angle of attack of $\alpha = 3.06$ deg. This case is an AGARD standard case [45] for the assessment of inviscid flowfield methods. This test case is often used by researchers for code validation; hence, a lot of computations [35,46] have been published, and experimental data

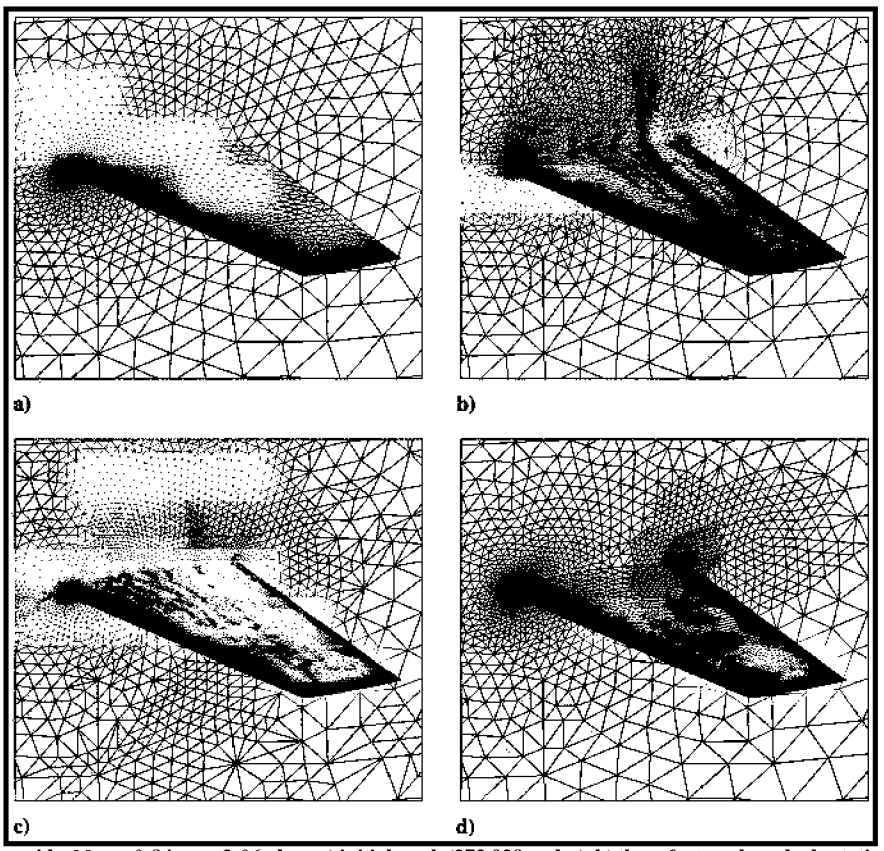

Fig. 13 ONERA M6 wing grids, $M_\infty = 0.84, \alpha = 3.06$ deg: a) initial mesh (373,938 nodes), b) three feature-based adaptation cycles (1,443,848 nodes), c) three adjoint-based adaptation cycle (615,465 nodes), and d) three TE-based adaptation cycles (459,346 nodes).

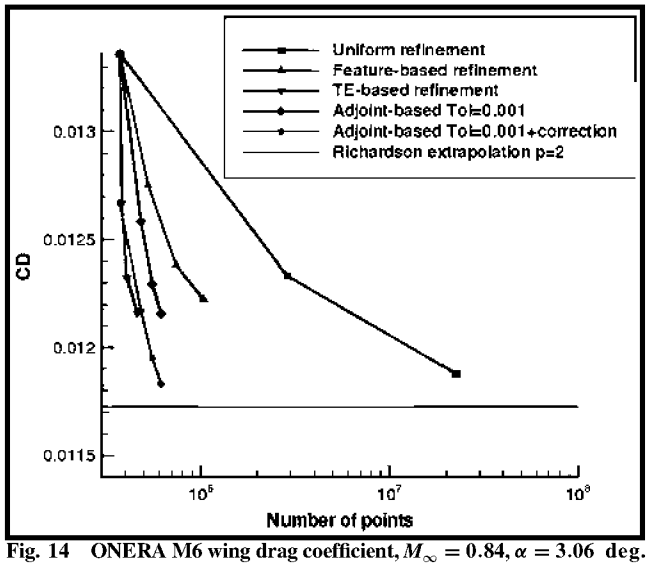

Fig. 14 ONERA M6 wing drag coefficient, $M_\infty = 0.84, \alpha = 3.06$ deg.

are available [45]. At these transonic flow conditions, a lambda-shaped shock wave is present on the upper side of the wing (see Fig. 12).

The initial grid is composed of 373,938 grid nodes, 2,078,216 tetrahedra, and 103,168 surface triangles (Fig. 12). The mesh is highly clustered at the leading edge and has a low resolution at the trailing edge and midpart of the wing; thus all indicators tend to refine these regions. The C_D functional has been checked with the three methods. In the case of the adjoint-based method, the adaptation is driven by specifying a 0.001 tolerance on the drag coefficient C_D, whereas λ is defined as 0.001 for the truncation error sensor.

The adapted meshes are presented in Fig. 13 after three cycles of adaptation. The three indicators generate meshes of very different shape. The adjoint C_D-based adaptation tends to uniformly add nodes on the wing, whereas the feature-based adaptation, as in the transonic two-dimensional NACA0012 test case of Sec. IV.A.2, strongly refines the lambda shock. The truncation-error-based indicator somehow lies in between, with nodes being clustered at the trailing edge and in the midpart of the wing.

Here, we do not observe spurious errors on the adapted meshes as it occurs with the two-dimensional NACA0012 subsonic test case of the Sec. IV.A.1. The explanation of this fact lies with the gradients of the present three-dimensional test case; the magnitude of the truncation error is governed by two components, one associated with the mesh quality (nonuniformity, skewness, etc.) and the other by the smoothness of the flow. The ONERA M6 wing flow at transonic regime is highly hyperbolic and exhibits strong shocks, thus making invisible the source of error generated by the mesh nonuniformities.

The comparison of the drag coefficient convergence, for all mesh adaptation algorithms presented in this paper, with respect to uniform refinement and Richardson extrapolation, is presented in Fig. 14. The worst result is attributed to the feature-based adaptation method, which predicts the same level of drag with approximately four times fewer nodes than the global refinement strategy with three adaptation cycles. The adjoint-based and truncation-error-based methods are highly superior here. Clearly, the adjoint C_D-based method with correction gives the best results, the computed drag obtained after three level of refinement is even more accurate than after two levels of global adaptation (note that the corresponding grid has over 22 million points) while requiring almost 40 times fewer nodes.

V. Conclusions

Decreasing computational work by mesh adaptation has been proven an elegant and efficient technique. However, current formulations employing indicators for mesh (de-)refinement based on flow features do not take into account the hyperbolic properties of Euler equations, and no guarantee can be given on the convergence of the adaptation procedure to the uniform refinement procedure. Furthermore, adapting the grid based on flow variables will not necessarily improve the accuracy of global quantities of interest in industry, such the lift or drag values.

This leads to the development of adaptation algorithms based on numerical error. In this work, a truncation-error-based indicator has been successfully developed and showed better convergence properties of force coefficients on two- and three-dimensional Euler test cases than feature based sensors and similar performances to the adjoint sensor. For the ONERA M6 wing, up to 10 times fewer points are found to be required for a given C_D level with respect to global refinement, using a truncation-error sensor.

The adaptation method based on the adjoint methodology has given the best results in terms of convergence, especially when the adaptation is coupled with the correction term. However, the adaptation based on truncation error is competitive, and it gives very encouraging results and is clearly computationally cheaper as it does not need an embedded mesh or the solution of a dual problem.

In light of the results obtained for two different functional outputs, it is seen that the truncation-error-based indicator performs quite well compared to the adjoint-based one. The reduction of the global error allows a significant improvement in both functional outputs at the same time in a very computationally efficient way.

Acknowledgments

The research described in this paper has been supported by the Ministry of Defense National Institute for Aerospace Technology under the activity Termofluidodinámica (IGB99001) and by Airbus-Spain under project DOVRES/FUSIM-E. The TAU Code is property of the DLR, German Aerospace Center, developed at the Institute of Aerodynamics and Flow Technology at Göttingen and Braunschweig, and has been licensed to the National Institute for Aerospace Technology and the Universidad Politécnica de Madrid through a research and development cooperation agreement.

References

[1] Mavriplis, D. J., "Grid Resolution Study of a Drag Prediction Workshop Configuration using NSU3D Unstructured Mesh Solver," 23rd AIAA Applied Aerodynamics Conference, Toronto, AIAA Paper 2005-4729, June 2005.

[2] Gao, H., and Wang, Z. J., "A Residual-Based Procedure for *hp* Adaptation on 2D Hybrid Meshes," 49th AIAA Aerospace Sciences Meeting, Orlando, FL, AIAA Paper 2011-492, Jan. 2011.

[3] Fidkowski, K. J., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. doi:10.2514/1.J050073

[4] Warren, G. P., Anderson, W. K., Thomas, J. P., and Krist, S. L., "Grid Convergence for Adaptive Methods," 10th AIAA Computational Fluid Dynamics Conference, Honolulu, AIAA Paper 1991-1592, June 1991.

[5] Dannenhoffer, J. F., and Baron, J. R., "Adaptation Procedure for Steady State Solution of Hyperbolic Equations," 22nd AIAA Aerospace Sciences Meeting, Reno, NV, AIAA Paper 1984-5, Jan. 1984.

[6] Kallinderis, Y., and Baron, J., "Adaptation Methods for a New Navier–Stokes Algorithm," *AIAA Journal*, Vol. 27, No. 1, 1989, pp. 37–43. doi:10.2514/3.10091

[7] Aftosmis, M., "Upwind Method for Simulation of Viscous Flow on Adaptively Refined Meshes," *AIAA Journal*, Vol. 32, No. 2, 1994, pp. 268–277. doi:10.2514/3.11981

[8] Nithiarasu, P., and Zienkiewicz, O., "Adaptive Mesh Generation for Fluid Mechanics Problems," *International Journal for Numerical Methods in Engineering*, Vol. 47, Nos. 1–3. doi:10.1002/(SICI)1097-0207(20000110/30)47:1/3<1::AID-NME810>3.0.CO;2-Q, 2000, pp. 629–662.

[9] Aftosmis, M., Melton, J., and Berger, M., "Adaptation and Surface Modeling for Cartesian Mesh Methods," 12th AIAA Computational Fluid Dynamics Conference, and Open Forum, San Diego, CA, AIAA Paper 1995-1725, June 1995.

[10] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466.

doi:10.1016/0021-9991(87)90093-3

[11] Hecht, F., and Mohammadi, B., "Mesh Adaptation by Metric Control for Multi-scale Phenomena and Turbulence," 35th Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 1997-859, Jan. 1997.

[12] Fortin, M., Vallet, M.-G., Poirier, D., and Habashi, W., "Error Estimation and Directionally Adaptive Meshing," 25th Fluid Dynamics Conference, Colorado Springs, CO, AIAA Paper 1994-2211, June 1994.

[13] Balasubramanian, R., and Newman, J. C., "Comparison of Adjoint-Based and Feature-based Grid Adaptation for Functional Outputs," 24th AIAA Applied Aerodynamics Conference, San Francisco, AIAA Paper 2006-3314, June 2006.

[14] Brandt, A., and Livne, O. E., *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, Cambridge Univ. Press, New York, 2011.

[15] Fidkowski, K. J., "Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results," 47th AIAA Aerospace Sciences Meeting, Orlando, FL, AIAA Paper 2009-1303, Jan. 2009.

[16] Fidkowski, K. J., and Roe, P. L., "An Entropy Adjoint Approach to Mesh Refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287.
doi:10.1137/090759057

[17] Leonard, B. P., "Comparison of Truncation Error of Finite-Difference and Finite-Volume Formulations of Convection Terms," NASA TM 105861, 1992.

[18] Jeng, Y. N., and Chen, J. L., "Truncation Error Analysis of the Finite Volume Method for Steady Convective Equation," *Journal of Computational Physics*, Vol. 100, 1992, pp. 64–76.
doi:10.1016/0021-9991(92)90310-U

[19] Hagen, S. F., "Estimation of the Truncation Error for the Linearized, Shallow Water Momentum Equations," *Engineering with Computers*, Vol. 17, No. 4, 2001, pp. 354–362.
doi:10.1007/s366-001-8301-z

[20] Briggs, W., Henson, V. E., and McCormick, S., *A Multigrid Tutorial*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[21] Trottenberg, U., Oosterleeand, C., and Schuler, A., *Multigrid*, Academic Press, New York, 2001.

[22] Ilinca, C., Zhang, X. D., Trapanier, J. Y., and Camarero, R., "A Comparison of Three Error Estimation Techniques for Finite-Volume Solutions of Compressible Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, No. 4, 2000, pp. 1277–1294.
doi:10.1016/S0045-7825(99)00377-1

[23] Garbey, M., and Shyy, W., "Error Estimation, Multilevel Method and Robust Extrapolation in the Numerical Solution of PDEs," *14th International Conference on Domain Decomposition Methods*, 2003.

[24] Phillips, T. S., and Roy, C., "Evaluation of Extrapolation-Based Discretization Error and Uncertainty Estimators," 49th AIAA Aerospace Sciences Meeting, Orlando, FL, AIAA Paper 2011-215, Jan. 2011.

[25] Bernert, K., "τ-Extrapolation-Theoretical Foundation, Numerical Experiment, and Application to Navier–Stokes Equations," *SIAM Journal on Scientific Computing*, Vol. 18, No. 2, 1997, pp. 460–478.
doi:10.1137/S1064827594276266

[26] Fulton, S. R., "On the Accuracy of Multigrid Truncation Error Estimates," *Electronic Transactions on Numerical Analysis*, Vol. 15, 2003, pp. 29–37.

[27] Fraysse, F., de Vicente, J., and Valero, E., "The Estimation of Truncation Error by τ-Estimation Revisited," *Journal of Computational Physics*, Vol. 231, No. 9, 2012, pp. 3457–3482.
doi:10.1016/j.jcp.2011.09.031

[28] Venditti, D. A., and Darmofal, L. D., "Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow," *Journal of Computational Physics*, Vol. 164, No. 1, 2000, pp. 204–227.
doi:10.1006/jcph.2000.6600

[29] Ponsn, J., "Grid Adaptation for Compressible Flows using Adjoint Equations," M.S. Thesis, Technical University of Madrid, Madrid, 2010.

[30] Hartmann, R., "Multitarget Error Estimation and Adaptivity in Aerodynamic Flow Simulations," *SIAM Journal on Scientific Computing*, Vol. 31, No. 1, 2008, pp. 708–731.
doi:10.1137/070710962

[31] Schwamborn, D., Gerhold, T., and Heinrich, R., "The DLR Tau-Code: Recent Applications in Research and Industry," *ECCOMAS CFD 2006—European Conference on Computational Fluid Dynamics*, edited by P. Wesseling, E. Oñate, and J. Périaux, Delft University of Technology, Delft, The Netherlands, 2006.

[32] Kim, H. J., Tokano, Y., and Nakahashi, K., "Error Estimation and Grid Adaptation Using Euler Adjoint Method," 17th AIAA Computational Fluid Dynamics Conference, Toronto, AIAA Paper 2005-5336, June 2005.

[33] Venditti, D. A., and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, 2003, pp. 22–46.
doi:10.1016/S0021-9991(03)00074-3

[34] Hämmerlin, G., and Hoffman, K. H., *Numerical Mathematics*, Springer–Verlag, Berlin, 1991.

[35] Park, M., "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, Vol. 42, No. 9, 2004, pp. 1854–1862.
doi:10.2514/1.10051

[36] Venditti, D. A., "Grid Adaptation for Functional Outputs of Compressible Flow Simulations," Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2002.

[37] Becker, R., and Rachnnacher, R., "Weighted a Posteriori Error Control in Finite Element Methods," *Proceedings of ENUMATH-97*, Heidelberg, 1998, pp. 621–637.

[38] Sonar, T., "Strong and Weak Norm Refinement Indicators Based on the Finite Element Residual for Compressible Flow Computation," *Impact of Computing in Science and Engineering*, Vol. 5, No. 2, 1993, pp. 111–127.
doi:10.1006/icse.1993.1005

[39] Dwight, R. P., and Brezillon, J., "Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization," *AIAA Journal*, Vol. 44, No. 12, 2006, pp. 3022–3031.
doi:10.2514/1.21744

[40] Brezillon, J., and Dwight, R. P., "Discrete Adjoint of the Navier–Stokes Equations for Aerodynamic Shape Optimization," *EUROGEN 2005—6th Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*, Munich, 2005.

[41] Hartman, R., Held, J. L. T., and Leicht, T., and Prill, F., "Error Estimation and Adaptive Mesh Refinement for Aerodynamic Flows," *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Vol. 113, 2010, pp. 339–353.
doi:10.1007/978-3-642-03707-8_24

[42] Syrakos, A., and Goulas, A., "Finite Volume Adaptive Solutions Using SIMPLE as Smoother," *International Journal for Numerical Methods in Fluids*, Vol. 52, No. 11, 2006, pp. 1215–1245.
doi:10.1002/fld.1228

[43] Carpentieri, G., Koren, B., and Tooren, M. J., "Development of the Discrete Adjoint for a Three-Dimensional Unstructured Euler Solver," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 237–245.
doi:10.2514/1.32871

[44] Dwight, R. P., "Efficiency Improvements of RANS-based Analysis and Optimization Using Implicit and Adjoint Methods on Unstructured Grids," Ph.D. Thesis, School of Mathematics, Univ. of Manchester, Manchester, England, U.K., 2006.

[45] Schmitt, V., and Charpin, F., "Pressure Distribution on the ONERA M6 wing at Transonic Mach Numbers," in *Experimental Data Base for Computer Program Assessment*, AGARD Rept. AR-138, 1979.

[46] Batina, J. T., "Accuracy of an Unstructured-Grid Upwind-Euler Algorithm for the ONERA M6 Wing," *Journal of Aircraft*, Vol. 28, No. 6, 1991, pp. 397–402.
doi:10.2514/3.46040

Z. Wang
Associate Editor

This article has been cited by:

1. F. Fraysse, G. Rubio, J. de Vicente, E. Valero. 2014. Quasi-a priori mesh adaptation and extrapolation to higher order using τ-estimation. *Aerospace Science and Technology* **38**, 76-87. [CrossRef]
2. F. Fraysse, E. Valero, G. Rubio. 2013. Quasi-a priori truncation error estimation and higher order extrapolation for non-linear partial differential equations. *Journal of Computational Physics* **253**, 389-404. [CrossRef]