POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea in Ingegneria Aeronautica
Ingegneria Aeronautica

MESH QUALIFICATION AND ADAPTATION BASED ON THE TOTAL DERIVATIVE OF AERODYNAMIC FUNCTIONS W.R.T MESH COORDINATES

Tutor: Prof. Franco AUTERI

Author:
Dario MILANI Matr. 680063

Anno Accademico 2012-2013

Acknowledgements

I would like to take this opportunity to thank the ”Method and Tools” department in Airbus St.Martin, Toulouse. All the components of this department contributed to the daily creation of a positive and productive atmosphere, which constitutes the main condition for any kind of evolution, both personal and technical. I would like to thank in particular my tutor in Airbus, Renaud Sauvage, who chose me for this important project and always gave me the best baselines in order to achieve my tasks efficiently. A particular attention should be directed to Maxime Nguyen-Dinh, my colleague and point of reference for the whole project. He transmitted to me all of his experience, without restriction, and contributed to the progression of my personal knowledge. Beside, these kind of contributions are often neglected, and that is why i want to highlight to all of you who are reading this report, that without the guide and the precious help of Maxime Nguyen-Dinh these results would not have been possible.

Contents

Nomenclature

W	flow conservative variables
W_b	flow conservative variables extrapolated on the surface
α	vector of geometry parametrization variables
X	mesh volume
F	goal function as function of W and X
$\mathcal{F}$	goal function as function of α
F	goal function as function of X
R	residual equations as function of W and X
Λ	adjoint vector
n_β	dimension of a generic vector β
f	analytic flux
Ω	domain
$\partial\Omega$	contour of Ω
ρ	fluid density
u	fluid velocity along X
v	fluid velocity along Y
w	fluid velocity along Z
E	fluid internal energy
$\wp$	projection operator
$D_{G,L}$	circle in G with radius L
$\theta i, j$	local sensor
θ	qualification criterion
$\theta[F]$	qualification criterion computed for the function F
$g^{i,j}$	contra-variant metric tensor
$x_{\xi_i \xi_j}$	covariant metric tensor
P_k	control function
θ_{gb}	qualification criterion for gradient based adaptation
C	computational space
P	parameter space
D	physical space
$\boldsymbol{\xi}$	coordinates in the computational space
$\boldsymbol{s}$	coordinates in the parameter space
$\boldsymbol{x}$	coordinates in the physical space

Introduction

This internship took place in AIRBUS OPERATIONS SAS whose location is in Saint Martin du Touch, Toulouse. The head of Methods and Tools for Aerodynamics department is Pascal Larrieu and the objectives of this department are framed in the development of methods and tools for CFD simulations. My team is the team EGAMT2 head by M.J Esteve and constitutes, with two other teams, the whole department. The various tools that are now in process of being developed by the team are targeted for shape optimization and improvement of the quality of meshes for the estimation of an objective function.

Nowadays, an huge number of simulations with an increasing accuracy are required in order to evaluate the aerodynamic performances. These performances (such as the drag, the lift and aerodynamic moments) are directly linked with the specific fuel consumption, which is one of the stones leading the competitiveness. It appears clear that these performances need to be evaluated with an increasing accuracy and at a decreasing cost. That is why, beside the reference constituted by the wind tunnel tests, the Computational Fluid Dynamics has reached an important position into the industrial framework.

One of the sources of the CFD simulations inaccuracy is the discretization error which comes from the fact that we dispose of a finite number of points and often we are not able to find the best way to distribute them in the domain. Furthermore, a successful answer to the need reducing the discretization error on estimation of the performance is represented by the goal mesh adaptation strategies which lead to the adaptation of the mesh in order to improve the accuracy within the estimation of one performance which is named in this case goal function.

With this study we mean to propose a new goal-oriented mesh adaptation strategy based on the total derivative of the aerodynamic function w.r.t mesh coordinates. The expensive in the calculation of this quantity is reduced thanks to the use of an adjoint method. We will present in a first step a theory review of the most important works present in literature and that are linked with our method. In a second step we will provide with a theoretical formulation of our step adaptation strategy and in particular to the definition of the adaptation sensor and the qualification criterion. Those two last quantities are studied in the fourth step. In a fifth step a practical mesh adaptation example is performed on a 2D test case and it will be compared with a test case using a feature based grid adaptation strategy in a sixth step. At the end we will provide with the extension of our sensor to a 3D test case.

Chapter 1

Theory review

Nowadays a huge number of CFD simulations is steadily carried out in order to evaluate the aerodynamic performances and to improve them by shape optimization. A major cause of the error associated with these simulations is the discretization of the computational domain from which comes the interest in the research of error reduction methods by mesh adaptation. However it is not always easy to define a process and an "indicator" to point out where and how a mesh has to be adapted in order to improve its quality. This can be intuitively implemented using an approach based on the flow features. For example to use the static pressure gradient as an indicator for mesh adaptation. This would lead to a refinement in the regions of interest capturing shocks and vortices otherwise dumped. While these methods capture the flow features as described, it does not necessarily reduce the error on the desired aerodynamic functions. Then more useful techniques for mesh adaptation are directly based on the function of interest for example using error estimations. Moreover these "goal-oriented" methods often use the adjoint vector of the function of interest which allows to compute gradients of the functions of interest with respect to a set of shape parameters (for shape optimization) and also with respect to volume mesh coordinates (for mesh adaptation).

This chapter is organized as follows. The first paragraph is a short presentation of the fluid mechanics equations, the second paragraph is an introduction to the gradient computation in shape optimization framework, the third paragraph is a review of the state of the art in mesh adaptation based on adjoint vector and the fourth paragraph is the presentation of the proposed methodology.

1.1 The equations of fluid mechanics

This paragraph is dedicated to the presentation of the equations that describe compressible, viscous and turbulent flows. The equations are written in the conservative form being the most adapted form within the finite volume context.

The Euler equations

Inviscid flow are described by the Euler equations. The set of this five equations forms a non-linear partial differential system which can be re-written in the following Cartesian form:

$$\frac{\partial w}{\partial t} + div(f) = 0, \tag{1.1}$$

where w is the continuous flow field and f is the Euler flux density:

$$f = \begin{pmatrix} \rho\bar{U} \\ \rho\bar{U} \bigotimes \bar{U} + p\bar{\bar{I}} \\ (\rho E + p)\bar{U} \end{pmatrix},$$

where ρ is the density, $\bar{U}$ is the velocity, E is the total energy and p is the static pressure which has to be expressed as

$$p = (\gamma - 1)\rho(E - \frac{||\bar{U}||^2}{2})$$

where γ is the specific heat ratio.

Navier-Stokes equations

Viscous flows are described by the Navier-Stokes equations. The conservative form of these equations is:

$$\frac{\partial w}{\partial t} + div(f - f^V) = 0, \tag{1.2}$$

where

$$f^V = \begin{pmatrix} 0 \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}}\bar{V} - \bar{q} \end{pmatrix},$$

where $\bar{\bar{\tau}}$ and $\bar{q}$ are the viscous shear stress and the heat flux respectively.

The RANS equations

At a certain Reynolds number it is observed the transition of the flow from laminar to turbulent. This transition give rise to chaotic and random processes which take place at different length and time scales. When there is a big separation between the different scales, the resolution of all of them become very expensive and the cost increases with the Reynolds number. That is why an average of the steady Navier-Stokes equations are most often preferred.This approach consist into the decomposition of the flow variables into a mean part and a fluctuating one. The application of the mean properties to the Navier-Stokes equations with this composition leads to the RANS equations (Reynolds Averaged Navier-Stokes). However the turbulence have to be modelled. Several turbulence model are present in the literature and the chosen one in this framework is the one of Spalart-Allmaras [11].

1.2 Gradient computation in the framework of shape optimization

This section is devoted to introduce and to provide a brief review of several methods for the computation of gradients that are at the basis of the developed methods in the present work.

We denote by α a vector of size n_α containing the design parameters of a shape (as illustrated in Figure 1.1 where blue and red meshes correspond to two different values of parameter α).

The volume mesh coordinates are denoted by X, W is a vector of size n_W containing the CFD solution of the problem (conservative variables at the centre of the cells) associated with the mathematical model used and W_b is the vector containing the extrapolation of W on the surface (Figure 1.2):

$$W = W(\rho, \rho u, \rho v, \rho w, \rho E)$$
$$W_b = W_b(\rho, \rho u, \rho v, \rho w, \rho E)$$

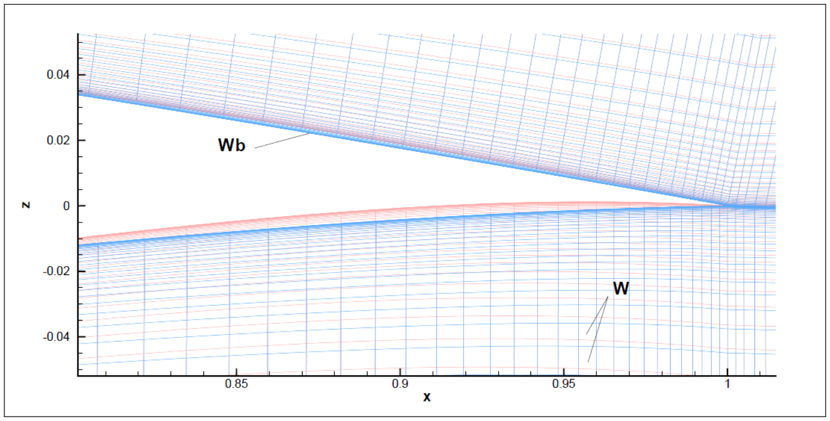


Figure 1.1: Airfoil shape and volume mesh plotted for two α values

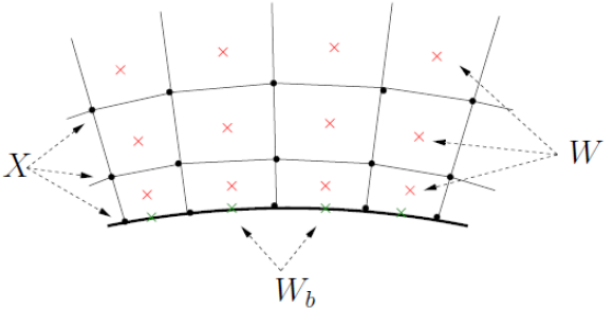


Figure 1.2: Notations for the mesh X, the flow field W and the flow field on the walls W_b

We denote by R the discretized equations of fluid dynamics (e.g. discrete form of Euler or RANS equations written in section 1.1) which link the fields W and the mesh X by the relation:

$$R(W, X) = 0,$$

which is a set of n_W nonlinear equations.

We consider a scalar function of interest, or goal, which is denoted by F. This function is basically the lift, the drag or others quantities of interest whose computation depends on the volume mesh X and the flow field W. So that we can write

$$F = F(W, X).$$

This equation can also be written by highlighting the dependence on the parameter α

$$\mathcal{F}(\alpha) = \mathcal{F}(W(\alpha), X(\alpha))).$$

Most often shape optimization methods need to evaluate the derivative of the function $\mathcal{F}$ with respect to parameters α:

$$\frac{d\mathcal{F}(\alpha)}{d\alpha}$$

The gradient can be evaluated following one of many already existing approaches which will be presented in the following sections (namely by finite differences, by the Discrete direct method or by the Discrete adjoint method).

1.2.1 Finite differences

Gradient evaluation by finite differences requires no modification of the solver itself but has the drawback to require a flow evaluation for every perturbed state $\alpha + \delta\alpha$ plus one (for a total of $n_\alpha + 1$ flow computations).

$$\frac{d\mathcal{F}(\alpha)}{d\alpha_i}\delta\alpha_i \simeq [\mathcal{F}(\alpha + \delta\alpha_i) - \mathcal{F}(\alpha)] = [f(W(\alpha + \delta\alpha_i), X(\alpha + \delta\alpha_i)) - f(W(\alpha), X(\alpha))]$$

In the second order finite differences formula the flow state variables need to be evaluate for every perturbed state $\alpha + \delta\alpha$ and $\alpha - \delta\alpha$ ($2n_\alpha$ times) according to the formula:

$$\frac{d\mathcal{F}(\alpha)}{d\alpha_i}\delta\alpha_i \simeq \frac{[\mathcal{F}(\alpha + \delta\alpha_i) - \mathcal{F}(\alpha - \delta\alpha_i)]}{2} = \frac{[\mathcal{F}(W(\alpha + \delta\alpha_i), X(\alpha + \delta\alpha_i)) - \mathcal{F}(W(\alpha - \delta\alpha_i), X(\alpha - \delta\alpha_i))]}{2}$$

The computational cost is dominated by the number of design parameters ($n_{\alpha+1}$ flow computations for the first order formula and $2n_\alpha$ flow computations for the second order formula).

1.2.2 Discrete direct method

A different method can be obtained if we consider the dependency of the function $\mathcal{F}$ with respect to W and X so that the differentiation leads to:

$$\frac{d\mathcal{F}(\alpha)}{d\alpha} = \frac{\partial F}{\partial W}\frac{dW}{d\alpha} + \frac{\partial F}{\partial X}\frac{dX}{d\alpha}. \tag{1.3}$$

We assume that the governing equations can be written in the form $R(W(\alpha)), X(\alpha)) = 0$, and that the discrete residual R is continuously differentiable with respect to the flow field and the mesh nodes coordinates. These hypothesis make that we can differentiate the fluid mechanics equations expressed as function of the parameter α as $(R(W(\alpha)), X(\alpha)) = 0)$ with respect to α leading to:

$$\frac{\partial R}{\partial W}\frac{dW}{d\alpha} = -\frac{\partial R}{\partial X}\frac{dX}{d\alpha}. \tag{1.4}$$

We consider the two last equations, the second one can be regarded as a linear system in the unknowns $dW/d\alpha$ (one for every parameter). After the calculation of $dW/d\alpha$ it is possible to substitute it into equation 1.3 in order to compute the gradient. This requires to solve n_α linear systems (of dimension n_W) in order to compute the term $dW/d\alpha$ The computational cost depends on the size of $\partial R/\partial W$ and the number of parameters α. This new procedure allows to calculate the required gradient with one non-linear solution plus n_α linear solutions instead of the at least $n_\alpha + 1$ non-linear solutions with the finite difference method.

1.2.3 Discrete adjoint method

The equation of residuals (1.4) can be multiplied by an arbitrary vector Λ^T of dimension n_W. This leads to:

$$\Lambda^T\frac{\partial R}{\partial W}\frac{dW}{d\alpha} + \Lambda^T\frac{\partial R}{\partial X}\frac{dX}{d\alpha} = 0,$$
$$\forall\Lambda \in \Re^{n_w}.$$

Adding this expression to the gradient formula (1.3):

$$\frac{d\mathcal{F}(\alpha)}{d\alpha} = \frac{\partial F}{\partial W}\frac{dW}{d\alpha} + \frac{\partial F}{\partial X}\frac{dX}{d\alpha} + \Lambda^T\frac{\partial R}{\partial W}\frac{dW}{d\alpha} + \Lambda^T\frac{\partial R}{\partial X}\frac{dX}{d\alpha},$$
$$\forall\Lambda \in \Re^{n_W}.$$

Carrying out the term $dW/d\alpha$ we obtain the following relation:

$$\frac{dF(\alpha)}{d\alpha} = (\frac{\partial F}{\partial W} + \Lambda^T \frac{\partial R}{\partial W})\frac{dW}{d\alpha} + \frac{\partial F}{\partial X}\frac{dX}{d\alpha} + \Lambda^T(\frac{\partial R}{\partial X}\frac{dX}{d\alpha}),$$
$$\forall \Lambda \in \Re^{nw}.$$

The vector Λ^T (called adjoint vector) is chosen in order to annihilate the first term of the last equation (this fact allows to avoid the computation of the n_α linear systems required for the discrete direct method) This lead to:

$$\frac{\partial F}{\partial W} + \Lambda^T(\frac{\partial R}{\partial W}) = 0. \tag{1.5}$$

The gradient is then given by:

$$\frac{dF(\alpha)}{d\alpha} = \frac{\partial F}{\partial X}\frac{dX}{d\alpha} + \Lambda^T(\frac{\partial R}{\partial X}\frac{dX}{d\alpha}). \tag{1.6}$$

The computation of Λ (1.5) does not depend on the number of parameters n_α but only on the number of functions of interest (n_F).

The crucial point we have to point out is the significant difference in terms of cost between the different approaches. Indeed, in typical aerodynamic applications, the number of parameters is greater than number of functions ($n_\alpha \gg n_F$) and then the adjoint method is most often preferred to others.

1.3 State of the art of mesh adaptation using the discrete adjoint method

Our purpose is to build a functional output grid adaptation strategy based on the adjoint vector. In literature different strategies have been described with the same purpose. The reference on adjoint-based grid adaptation for functional output is the method proposed by: David A. Venditti and David L. Darmofal [15]. We can also cite the work of Richard P. Dwight [8].

1.3.1 The method of Venditti and Darmofal

Venditti and Darmofal proposed to use an adjoint formulation to relate the local residual error in the flow solution to the global error in the chosen output thus obtaining a mesh adaptation strategy.

This method needs two mesh levels. A coarse mesh on which CFD calculations are affordable, and a fine mesh used to generate a correction of the error of discretization (it should be noted that CFD calculations on the fine mesh are supposed to be extremely expensive). This correction is made up because it is possible to evaluate a part of the error. The part of the error which is evaluable is called computable error while the non computable part is called remaining error. The computable error can be directly used to correct the functional estimation. Finally, the adaptation method is based on an estimation of the remaining error. Within the following passages we try to exhibit the formulation of this remaining error which is used to adapt the mesh. The flow solution on a very fine grid is denoted by W_h(where we suppose that the solution is very close to the exact one) and δW_h the error perturbation vector of the solution on the fine grid, the expression for the perturbed direct flow solution $\tilde{W_h}$ is:

$$\tilde{W_h} = W_h + \delta W_h. \tag{1.7}$$

The functional calculated over the fine grid is denoted by F_h and is evaluated on the W_h and $\tilde{W_h}$. This lead to the following relation:

$$\delta F_h = F_h(\tilde{W_h}) - F_h(W_h). \tag{1.8}$$

As well as for the already defined residuals calculated over the fine grid:

$$\delta R_h = R_h(\tilde{W_h}) - R_h(W_h) = R_h(\tilde{W_h}), \tag{1.9}$$

where $R_h(W_h)$ is zero by construction. Writing down the equations of the small variations of F_h and R yields:

$$\delta F_h \approx \frac{\partial F_h}{\partial W_h}\delta W_h, \tag{1.10}$$

$$R_h(\tilde{W_h}) = \delta R_h(\tilde{W_h}) \approx \frac{\partial R_h}{\partial W_h}\delta W_h. \tag{1.11}$$

Multiplying the last equation by a vector Λ_h^T leads to:

$$\Lambda_h^T R_h(\tilde{W_h}) \approx \Lambda_h^T \frac{\partial R_h}{\partial W_h}\delta W_h. \tag{1.12}$$

Summing member to member the equations (1.10) and (1.12):

$$\delta F_h + R_h(\tilde{W_h}) \approx \frac{\partial F_h}{\partial W_h}\delta W_h + \Lambda_h^T \frac{\partial R_h}{\partial W_h}\delta W_h. \tag{1.13}$$

Which can be factorised by δW_h :

$$\delta F_h + \Lambda_h^T R(\tilde{W_h}) \approx (\frac{\partial F_h}{\partial W_h} + \Lambda_h^T \frac{\partial R}{\partial W_h})\delta W_h. \tag{1.14}$$

The vector Λ^T is chosen such that term on the right hand side of the equation vanishes:

$$\frac{\partial F_h}{\partial W_h} + \Lambda_h^T \frac{\partial R}{\partial W_h} = 0. \tag{1.15}$$

With this choice of Λ, equation (1.14) leads to:

$$\delta F_h - \Lambda_h^T R(\tilde{W_h}) \approx 0. \tag{1.16}$$

We can notice that this relation correspond to (1.5) on the fine mesh. After its computation it is possible to correct the generated error in the functional $F_h(\tilde{W_h})$ by a perturbation of the flow solution $\tilde{W_h}$ using the term:

$$\delta F_h \approx -\Lambda_h^T R(\tilde{W_h}),$$

which is an exact expression for linear functional and residuals. The computation of the adjoint vector on the fine grid is too expensive. Therefore an estimation of Λ_h is considered and it is expressed as we did for the residuals and the functional as the sum of the adjoint vector calculated on the fine grid and a perturbation $\delta\Lambda_h$

$$\tilde{\Lambda_h} \equiv \Lambda_h + \delta\Lambda_h. \tag{1.17}$$

The perturbation on the functional of interest can be rewritten taking into account this last relation:

$$\delta F_h \approx \Lambda_h^T R_h(\tilde{W_h}) = \tilde{\Lambda}_h^T R_h(\tilde{W_h}) - \delta\Lambda_h^T R_h(\tilde{W_h}). \tag{1.18}$$

We are able now to highlight the computable correction term which is:

$$\tilde{\Lambda}_h^T R_h(\tilde{W_h}),$$

and the remaining error which is:

$$-\delta\Lambda_h^T R_h(\tilde{W_h}).$$

The expression of the remaining error will be:

$$\delta F_h - \tilde{\Lambda}_h^T R_h(\tilde{W}_h) \approx -\delta\Lambda_h^T R_h(\tilde{W}_h). \tag{1.19}$$

The strategy used by the authors consists in a first step to correct the functional with the term $\tilde{\Lambda}_h^T R_h(\tilde{W}_h)$ and in a second step to use an adaptive process to further enhance the functional estimation accuracy by reducing the remaining error which is expressed in the right-side of equation (1.19). The fields Λ_h and W_h represent the values of Λ and W calculated on the fine mesh, as the fields Λ_H and W_H represent the values of Λ and W calculated on the coarse mesh. The computation of Λ_h and W_h is considered too expensive and approximation $\tilde{\Lambda}_h$ and $\tilde{W}_h$ are computed by interpolation of the variables Λ_H and W_H over the fine grid.

The advantage of this method is that they build this correction which can be used in any context (scheme) but the bigger drawback is that a second very fine mesh is required with all the complication brought about data storage and interpolation problems.

1.3.2 The method of Dwight

The method proposed by Dwight [2] is a very different adjoint-based method which considers the sensitivity of the functional of interest to the level of dissipation introduced by the numerical flux (artificial viscosity scheme with two parameters $k^{(2)}$ $k^{(4)}$ which control the dissipation). The study was effectuated on the numerical flux of the Jameson *et.al.* scheme ([3]). A study on several mesh sizes shows that more than 90% of the discretization error is due to the dissipation terms which are setted by the parameters $k^{(2)}$ and $k^{(4)}$. The expression of the discretization error due to dissipation is:

$$\eta = k^{(2)} \frac{dF}{dk^{(2)}} + k^{(4)} \frac{dF}{dk^{(4)}}. \tag{1.20}$$

The derivatives that appear in this relation can only be computed using the adjoint method which leads to:

$$\frac{dF}{dK^2} = \lambda^T \frac{\partial R}{\partial K^2}$$

and

$$\frac{dF}{dK^4} = \lambda^T \frac{\partial R}{\partial K^4}$$

Thus the error measure can be rewritten:

$$\eta = \Lambda(k^2 \frac{\partial R}{\partial k^2} + k^4 \frac{\partial R}{\partial k^4}), \tag{1.21}$$

where it appears clearly the role of the adjoint vector. The method consist in defining the dissipation parameters for each cell:

$$\eta_i = \Lambda(k_i^2 \frac{\partial R}{\partial k_i^2} + k_i^4 \frac{\partial R}{\partial k_i^4}). \tag{1.22}$$

Intuitively it is possible to refine locally the mesh in order to minimize the dissipation error Moreover $F - k^{(2)} dF/dk^{(2)} - k^{(4)} dF/dk^{(4)}$ is considered as a corrected output value.

The positive aspect of this method with respect to the method implemented by Venditti and Dermofal is that this method does not require a second mesh, but the most important drawback is that it is restricted to the Jameson *et. al.* scheme.

1.4 The proposed approach

1.4.1 The total derivative of aerodynamic w.r.t mesh nodes coordinates

It has been shown in the paragraph (1.2.3) that the expression of $dF/d\alpha$ can be evaluated using an adjoint approach with the following relation:

$$\frac{d\mathcal{F}(\alpha)}{d\alpha} = \frac{\partial F}{\partial X}\frac{dX}{d\alpha} + \Lambda^T(\frac{\partial R}{\partial X}\frac{dX}{d\alpha}). \tag{1.23}$$

Factorising this last equation with respect to $dX/d\alpha$ we obtain:

$$\frac{d\mathcal{F}(\alpha)}{d\alpha} = (\frac{\partial F}{\partial X} + \Lambda^T\frac{\partial R}{\partial X})\frac{dX}{d\alpha}. \tag{1.24}$$

This allows to identify the expression of the total derivative of F w.r.t to mesh nodes coordinates:

$$\frac{d\mathcal{F}(\alpha)}{d\alpha} = (\frac{d\mathrm{F}}{dX})\frac{dX}{d\alpha}. \tag{1.25}$$

It can also be proved that the term into brackets of the expression (1.24) is the term (dF/dX) which compares in the expression (1.25). We notice that the totally derivative of our functional compares naturally in the formulation of the adjoint method, and its expression is given by:

$$\frac{d\mathrm{F}}{dX} = \frac{\partial F}{\partial X} + \Lambda^T\frac{\partial R}{\partial X}. \tag{1.26}$$

This derivative is the starting point of our work.

1.4.2 Meanings of $d\mathbf{F}/dX$

The derivative $d\mathrm{F}/dX$ is a link between the functional F and the volume mesh X so it gives us essential information for the mesh adaptation. As it indicates the sensitivity of the functional output with respect to the volume mesh (Figure 1.3), this variable is calculated for every point of the grid and it defines the influence of these points position on the goal estimation.

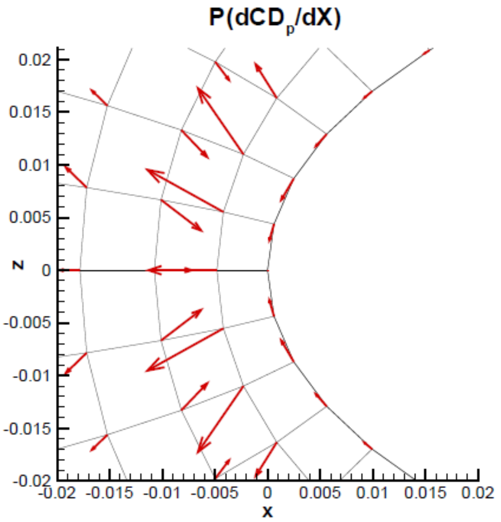


Figure 1.3: Example $(dCd_p)/(dX)$

The equation (1.26) is composed by two terms:

$$\frac{d\mathrm{F}}{dX} = \underbrace{\frac{\partial F}{\partial X}}_{\text{geometrical derivative}} + \underbrace{\Lambda^T\frac{\partial R}{\partial X}}_{\text{aerodynamic derivative}}. \tag{1.27}$$

The first term in the right hand side of the equation is the goal sensitivity with respect to the mesh coordinates of the nodes that belong to the support of F. The second term contains the goal sensitivity w.r.t the mesh nodes through the sensitivity of the flow field W w.r.t the nodes locations.

In the following example we provide additional explanations of these two terms. We consider the wave drag (caused by the shock) derivative w.r.t the volume mesh coordinates. It is calculated on the contour underlined on the Figure (1.4).

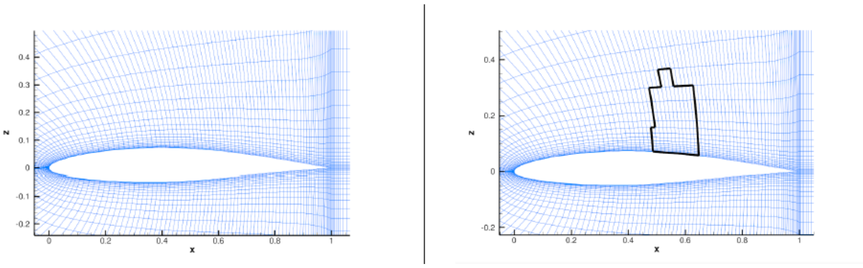


Figure 1.4: Mesh around the RAE2822 airfoil (left); Contour of integration (right)

As expected the high values of the first term $\partial F/\partial X$ are placed on the integration contour, because a displacement of the mesh coordinates X generates an important variation of F even if it is purely geometric (Figure 1.5).

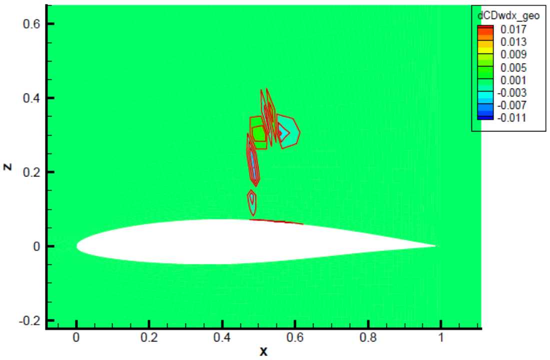


Figure 1.5: Geometrical derivative (x-component)

As the second term $\Lambda^T(\partial R/\partial X)$ keep into account the aerodynamic it is expected that the calculation of the goal function is influenced by the entire field. In fact, we exhibit the behaviour of this second term and we notice that it has high values on the trailing edge and on the leading edge. This is shown in Figure 1.6 and it is expected as the aerodynamic couples the entire domain, and in particular it is influenced by the trailing edge and the leading edge. The sum of this two terms is the functional derivative with respect to mesh coordinates (Figure 1.7). In [5] Nielsen was the first to use this quantity. His purpose was to avoid a prohibitive storage of information related to the singles terms in (1.26).

1.4.3 Mesh adaptation criteria built up from dF/dX

The term $d\mathrm{F}/dX$ represents a vector which is defined on every node of the mesh. Moreover there are nodes of the mesh which need to be treated differently. We consider the nodes on the walls contour. On these nodes, the components of $d\mathrm{F}/dX$ normal to the walls correspond to a

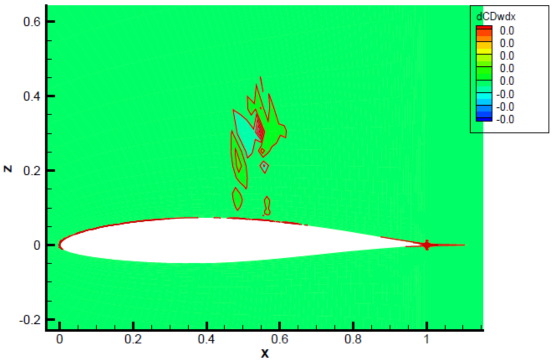


Figure 1.6: Aerodynamic derivative (x-component)

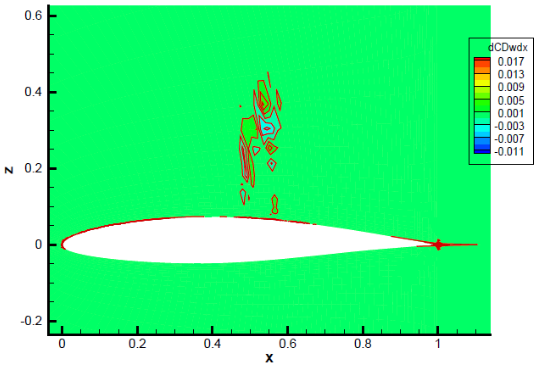


Figure 1.7: Complete derivative (x-component)

change of the shape. Than we introduce a projection (which we will call as $\wp$) of $d\mathrm{F}/dX$ which does not mean to change the shape of the airfoil and is defined by:

$$\begin{array}{ll} \wp(d\mathrm{F}/dX) = d\mathrm{F}/dX & \text{In the volume} \\ \wp(d\mathrm{F}/dX) = d\mathrm{F}/dX - (d\mathrm{F}/dX \cdot \vec{n})\vec{n} & \text{Along the walls contour} \\ \wp(d\mathrm{F}/dX) = 0 & \text{At the corner of the walls} \end{array}$$

This projection is introduced in order to maintain only the components of $d\mathrm{F}/dX$ (Figure 1.8) that are actually usable for mesh adaptation. A Taylor series at the first order gives:

$$\mathrm{F}(X + dX) \simeq \mathrm{F}(X) + \frac{d\mathrm{F}}{dX} dX$$

By applying the projection we also get:

$$\mathrm{F}(X + dX) - \mathrm{F}(X) \simeq \wp(\frac{d\mathrm{F}}{dX}) dX$$

A majoration of the right hand side of this expression can be computed. This is obtained by considering that the admissible mesh displacement dX is such that each node is to stay in a circle of radius half the distance to its closest neighbour. Our criterion is now able to take into account the already existing distance between those nodes so that we will displace them in agreement with the "room" at disposition.

In addition to this local criterion we introduced a global one, which allows to define the global quality of the mesh. All the explanations and developments will be given in the following

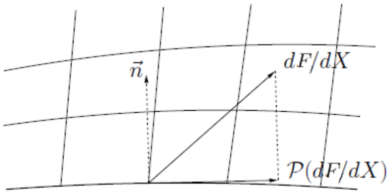


Figure 1.8: dF/dX projection

chapter. We notice that the first use of $d\mathrm{F}/dX$ for mesh adaptation have been done by Peter *et. al.* in [7] for 2D and 3D Euler flow computations.

1.5 Conclusions

This chapter gave a review about the different strategies which can be found in literature which use the adjoint methods to reduce the error associate with a computation of a functional F. After that we introduced the basic idea of our method which is close to the Venditti and Dermofal [15] one but is quite different in respect to the method implemented by Dwight [2]. The important advantage of our method is that it does not depend on the scheme used as for Dwight method, and it does not require two levels of mesh as for the method implemented by Venditti and Dermofal. The adjoint is calculated because it serves to compute the sensitivity $d\mathrm{F}/dX$. Our purpose is to use this sensitivity as indicator for our mesh adaptation strategy as we explain in the following chapter.

Furthermore our method uses the variable Λ which is also often used in shape optimization framework. This fact lead undoubtedly to computation time saving because Λ would be already given and it would not require further computations.

Chapter 2

The proposed goal-oriented mesh adaptation methodology based on dF/dX

This chapter is devoted to the theoretical presentation of the sensor built up from $d\mathrm{F}/dX$ and the mesh adaptation strategy used for practical applications. First of all the local sensor (which will be used to induce the adaptation of the mesh) and the global one (which is used as indicator of the global mesh quality) are presented. In a second step the mesh adaptation strategy based on an elliptic PDE is presented.

2.1 Definition of an adaptation sensor and a qualification criterion

The previous chapter introduced the total derivative of the function of interest w.r.t mesh coordinates (1.26). As this derivative indicates the sensitivity of the goal function F w.r.t the volume mesh X, we can intuitively state that it tends towards zero when the mesh is ideally adapted for the calculation of F. This condition can be achieved only in the case of infinity grid refinement, so it is clear that we have to use the sensitivity $d\mathrm{F}/dX$ in a different way. We explain within this section the considerations which bring us to the theoretical definition of the adaptation sensor and the qualification criterion.

The first consideration which has to be taken into account is about stability. We take a volume mesh denoted by X and a second mesh volume which is a perturbed state of the first one, which is denoted by $X + dX$. The flow state variables computation (W) allows the computation by integration of a goal function F. This goal function can be the lift coefficient, the drag coefficient or others quantities of interest for the both initial mesh and perturbed mesh:

$$\begin{aligned} X &\rightarrowtail \mathrm{F}(X) \\ X + dX &\rightarrowtail \mathrm{F}(X + dX) \end{aligned}$$

The mesh X is considered to be at a stable position if $\mathrm{F}(X + dX)$ - $\mathrm{F}(X)$ is small. This fact means that for a perturbation of the mesh nodes position, the functional calculated does not change significantly. In fact, a condition which is necessary for a good mesh quality is that its perturbation does not play an important role in the calculation of the solution, because it would not have physical reasons as the boundary conditions would not change.

It is necessary to consider that on the solid contours the influence of the perturbation dX along the normal direction of the solid contours has physical reasons to change the evaluation of the functional $\mathrm{F}(X + dX)$ as it represents a change of the solid shape. This consideration leads to the introduction of the projection operator $\wp$ in paragraph 2.1.1, and the correct formulation of the stability condition has to delete the components of $d\mathrm{F}/dX$ that change the shape. We state as consequence that for a volume mesh denoted by X and a second volume mesh which is a perturbed state of the first one, which is denoted by $X + dX$. The mesh X is considered a stable position if $\wp(\mathrm{F}(X + dX))$ - $\wp(\mathrm{F}(X))$ is small.

The attempt towards the stability condition is made by trying to pull $d\mathrm{F}/dX$ towards zero, but in a discretized context where a redistribution of point it is necessary to improve the accuracy of the goal function estimation without increasing the number of nodes and the calculation time, it is preferable to redistribute the points trying to reach a situation in which the sensitivity $d\mathrm{F}/dX$ is the same along the mesh.

Moreover, in a structured mesh context, where the regularity is one of the crucial points, it is preferable to weight the sensitivity by a characteristic length $r_{i,j}$. Thanks to $r_{i,j}$ it is possible to take into account that on the zones where the mesh is already fine, an important sensitivity of $d\mathrm{F}/dX$ would induce a strong irregularity, so a smaller mesh points movement is allowed.

In the following we mean to explain the practical aspect which stands with the previous considerations, especially on the projection operator and the characteristic length which lead to the definition of the sensor and the qualification criteria.

Roughly speaking, the multiplication by a characteristic length assures that the sensor will be able to take into account the importance of the mesh displacement on the calculation of F weighted by the local size of the mesh. If we achieve a reduction of this criterion we also achieve the best distribution of points to calculate F.

2.1.1 Projection of $d\mathrm{F}/dX$

As we explained in section 1.4.3, close to the wall a displacement of the grid would lead to a deformation of the wall geometry. A projection operator has therefore been devised to avoid any deformation of the body shape and it will be denoted by $\wp(d\mathrm{F}/dX)dX$ (Figure 1.8) in what follows. This operator should preserve the displacement of the internal points while avoiding any movement of the boundary points that would deform the body shape, additional details will be given in the following. The projection is represented in Figure 2.1 where the effect of the projection on the displacement of a point lying on a solid surface is illustrated. It is clear that the points on the skin can move only along the solid surface. The projection is defined through the operator on which the goal sensitivity acts as:

$$\wp(\frac{d\mathrm{F}}{dX})dX. \tag{2.1}$$

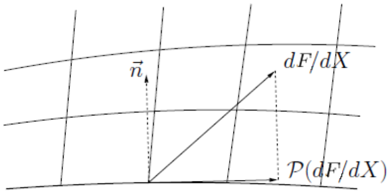


Figure 2.1: dF/dX projection

Since a given grid does not, in general, allow a complete resolution of the physic of the problem which is necessary for an accurate computation of a given output, $\wp(d\mathrm{F}/dX)dX$ will be highly discontinuous especially at the first step of adaptation as shown in Figure 2.3.

In order to partially supply at this fact it is possible to introduce a spatial-mean which will be furthermore studied in section 2.1.2.

Definition of the global and the local criteria of mesh quality

We denote by $r_{i,j}$ half the distance between the grid point $X_{i,j}$ and the closest one. This leads to the formal definition of our qualification criterion θ as follows:

$$\theta(i,j) = \|\wp(\frac{d\mathrm{F}}{dX})_{i,j}\| \; r_{i,j}, \tag{2.2}$$

An example of the $\theta_{(}i,j)$ field is reported in Figure 2.2

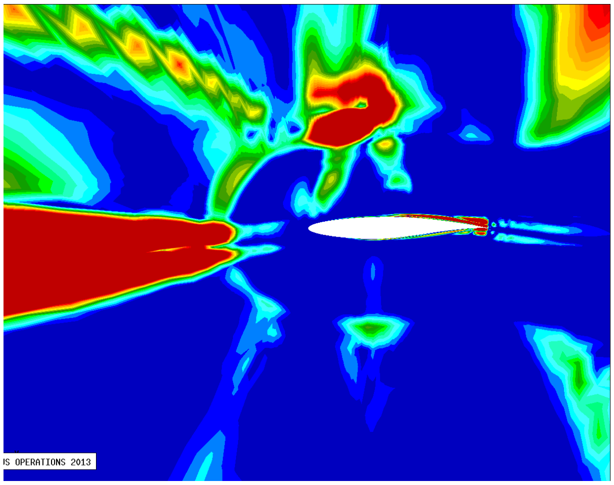


Figure 2.2: $\theta[Cl_p]$ around the RAE2822 airfoil

A global quality criterion is defined as:

$$\theta = \frac{1}{N_i N_j} \sum_{i;j} \theta(i,j). \tag{2.3}$$

The local criterion, equation (2.2) gives information about the local quality of the grid and it has high values on the nodes whose relocation would cause a large variation of the output. The global criterion, equation (2.3) is built as the average of the local criterion on the whole domain and it is useful to evaluate the quality of the entire mesh. High values of θ mean that the point relocation has a high impact on the function and therefore the mesh should be adapted.

Additional explanation about the development of this sensor will be given in the following section.

2.1.2 Spatial mean

In this paragraph we will discuss about the tool we used to smooth the field $\wp(d\mathrm{F}/dX)$. This point is very important because it allows to obtain a more uniform field which is more useful to capture the regions where a grid modification would have an important impact on the computation of F. Otherwise it would not be possible to distinguish them clearly as $\wp(d\mathrm{F}/dX)$ is highly irregular (Figure 2.4, left). Furthermore a smoothing is necessary because there might be nodes close to each others with vectors pointing in opposite directions as illustrated in Figure 2.3. This fact would cause a local compensation effect in and the nodes would be moved without any advantage. The smoothing is obtained by a filtering of the $\wp(d\mathrm{F}/dX)$ function.

The filter is in fact a weighted average. The filtered value on a grid point is computed by weighted average of the values on the points lying in its neighbourhood. It is possible to consider neighbourhoods of different radius so that the filtered value of $\wp(dF/dX)$ over a node is obtained taking into account a variable number of nodes around him. The neighbourhood

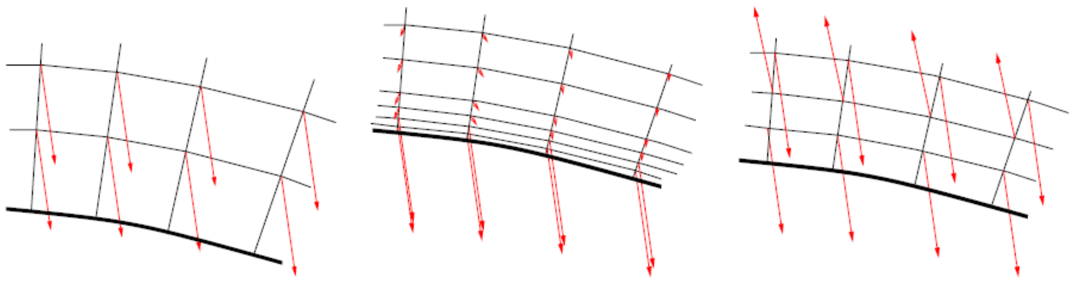


Figure 2.3: (Left) large regular $\wp(d\mathrm{F}/dX)$ with a large node displacement possibility; (Center) large regular $\wp(d\mathrm{F}/dX)$ without large possible displacement of the nodes; (Right) large irregular $\wp(d\mathrm{F}/dX)$ with large possible displacement of nodes

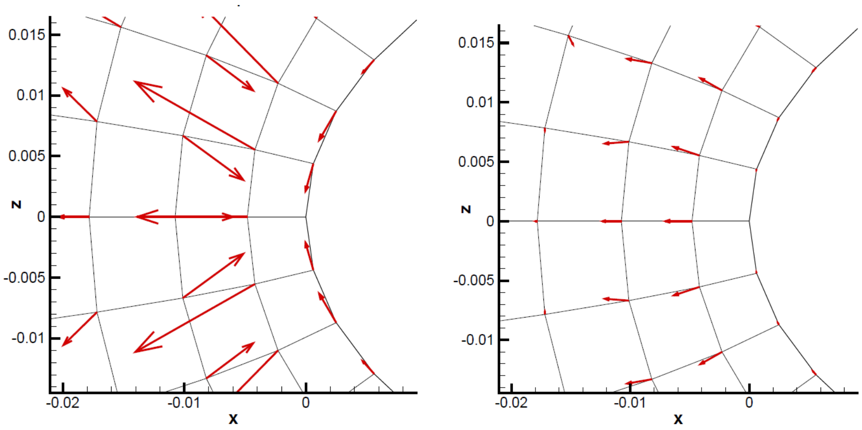


Figure 2.4: Example of discontinuities of $d(Cd_p)/dX$ (Left); Regularization of $d(Cd_p)/dX$ obtained by filtering (Right)

radius (L) is defined in percentage of characteristic length such as the cord of the airfoil. The smoothing effects obtained by this operation are shown on Figure 2.4 (Right).

The choice of L has to be made as a compromise among different requirements such as the degree of regularity required, the complexity of the problem geometry, the computational cost, and so on. A direct comparison of computations made with study with different radius has been performed in order to highlight issues and to evaluate the impact of the radius on the computed qualification criteria.

In the following example $\theta_1 = \theta_1[Cl_p]$ and $\theta_2 = \theta_2[Cd_p]$ are the two goal functions chosen. The calculation is performed on the RAE2822 airfoil, the Reynolds number is Re=6.5 10^6, the Mach number is $M_\infty = 0.725$, the angle of attack is $\alpha = 2.466$ and a Spalart-Allmaras turbulence model has been employed. The calculation was performed on a structured grid (Figure 2.5) using an upwind-scheme (Roe scheme).

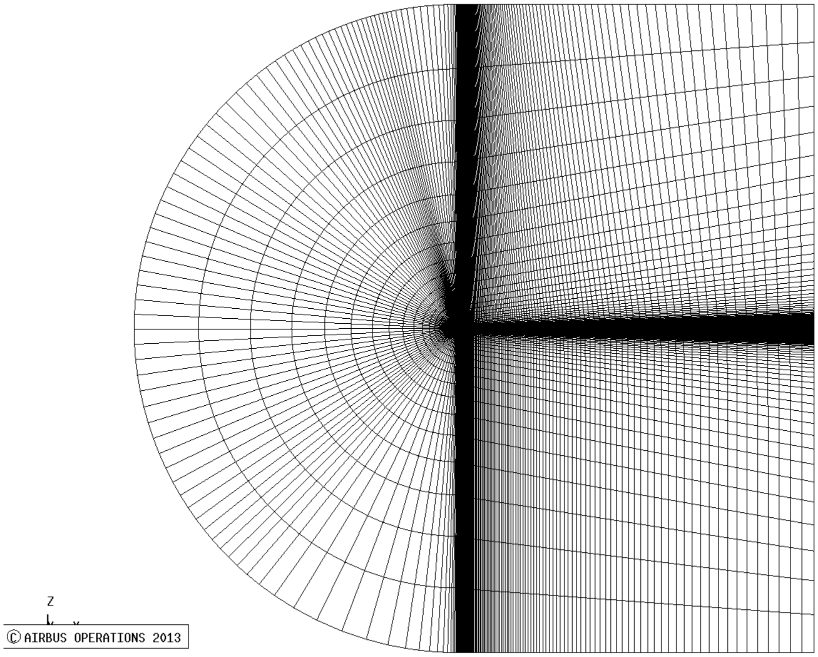


Figure 2.5: Mesh around the RAE2822 airfoil

The flow solution is plotted for the Mach number variable in Figure 2.6. The θ_1 and θ_2 built for different values of the radii L are plotted in the Figure 2.7 and 2.8 respectively.

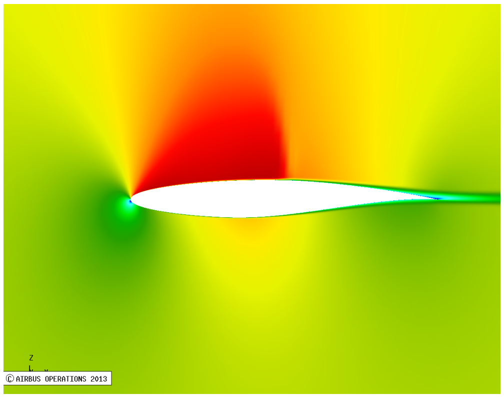


Figure 2.6: Mach around the RAE2822 airfoil

It can be observed that the criterion θ calculated for Cl_p and Cd_p detects the zones which have to be adapted in order to reduce the influence of the mesh on the calculation of these two objective functions. The criterion is plotted in Figure 2.7 for Cl_p and in Figure 2.8 for Cd_p for increasing radii. When $L = 0$ and the filter is not applied the figure shows that the zones detected are the shock wave, the trailing edge and the leading edge as others methods for mesh adaptation would detect. As usual in mesh adaptation literature a comparison with the feature based adaptation method is performed. This comparative highlights that our method detects zones of the domain which would not be detected from a feature-based method. An important example consists in the detection of the upstream zone. That region is detected because the

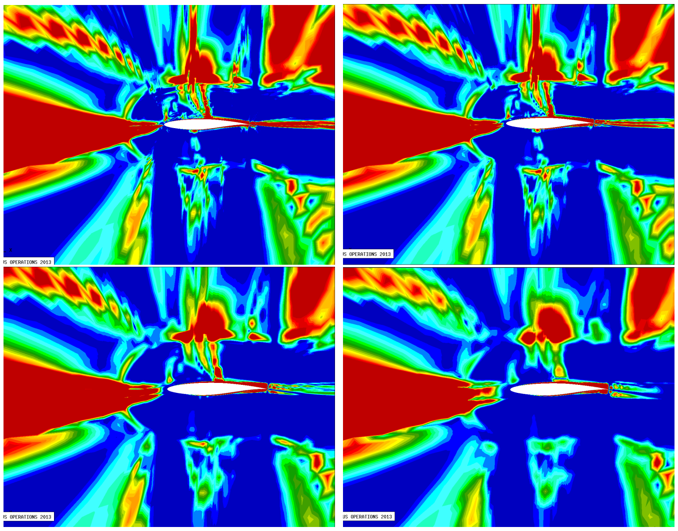


Figure 2.7: Quality criteria for Cl_p, θ_1: without average, $L = 0$,$L = 0.2$,$L = 0.4$,$L = 0.8$

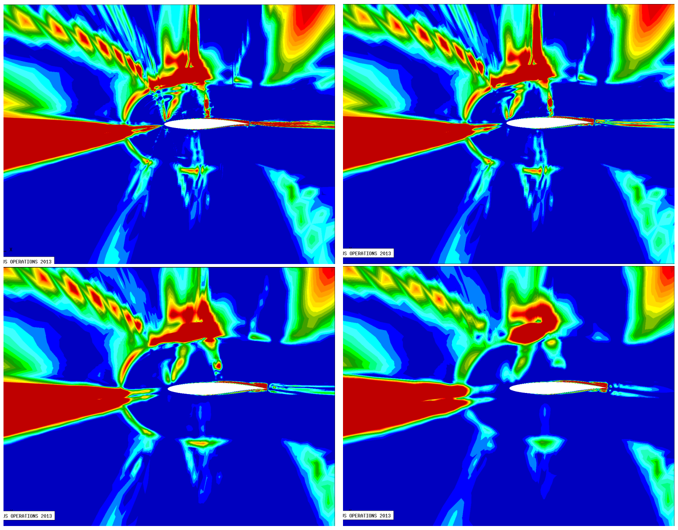


Figure 2.8: Quality criteria for Cd_p, θ_2: without average, $L = 0$,$L = 0.2$,$L = 0.4$,$L = 0.8$

proposed method because this criterion is able to predict that the error associated with the flow computation are transported along the flow field by advection, so that the errors introduced along the inlet boundary are transported up to the airfoil generating an inaccurate estimate of the goal F.

Furthermore the importance of a good accuracy on the upstream is underlined by looking at the figures generated for increasing values of L. The criterion is plotted for $L = 0.2$, $L = 0.4$ and $L = 0.8$. It is clear that the refinement of the mesh in the upstream region becomes more important with respect to the region around the shock. Evidences of this fact will be given in the Section 4.4 where the numerical solution is compared with experimental results.

It can be observed that θ decreases while the radius increases. Furthermore, the criterion predicts a better global mesh quality when the average is taken. This fact is due to the chosen filtering operator which has the effect of smoothing the zones containing high values of $d\mathrm{F}/dX$. This is a positive aspect because as it allows to deal with a regular field which is a crucial point in the adaptation process, in particular for structured meshes. The unfiltered field θ contains all the information linked to the sensitivity of the function of interest with respect to the volume mesh. Of this information we choose to retain just the smooth part in order to adapt the mesh.

2.2 Local mesh generation and adaptation

The tools presented in this chapter are general. In fact they can be employed for any type of grid. A crucial point in the adaptation process is the smoothness and regularity of the mesh. Elliptic equations are often used for mesh generation [12] because of their high regularity.

2.2.1 Elliptic equations for mesh generation

We will refer to three different vector spaces with their respective coordinate systems: the first one is the computational space with coordinates $\boldsymbol{\xi}$. The second one is the parameter space with coordinates $\boldsymbol{s}$. The third one is the physical domain with coordinates $\boldsymbol{x}$. The coordinates in the physical space can be considered as a function of the coordinates $\boldsymbol{\xi}$. This lead towards the definition of the following form of PDEs:

$$\sum_{i,j=1}^{3} g^{ij} x_{\xi^i \xi^j} + \sum_{k=1}^{3} g^{kk} P_k x_{\xi^k} = 0. \tag{2.4}$$

Where x is the position vector (in the Cartesian reference), $x = (x_1, x_2, x_3)$, $\xi^i = (\xi^1, \xi^2, \xi^3)$ are the curvilinear coordinates and $g^{ij}(i,j = 1,3)$ is the contra-variant metric tensor and finally P_k are the control functions. There is a unique link between the generated mesh and the control functions. For example once we generate the grid it is possible to associate to it the vector of control functions $(P_k^{initial})$ and vice versa (it is possible to show that we can always calculate it, even if the starting mesh does not come from a PDEs system resolution). In fact it is also possible to associate a modification of the control functions of a mesh, keeping the regularity of the initial mesh, and to adapt it.

A good method to adapt a mesh is to suitably modify the control functions and then recompute the mesh coordinates corresponding to the new control functions. The success of this method relies on the way this modification is preformed. In this work, we used the information related to the local criterion $\theta_{i,j}$. The variation of control function can be expressed in general as follows:

$$P_k = P_k^{initial} + \epsilon P_k^{adapt}. \tag{2.5}$$

Where P_k^{adapt} are the modified control functions that will be presented in 2.2.2, $P_k^{initial}$ are the initial control functions used to generate the initial mesh, and ϵ is a parameter which controls the magnitude of the modification to the original mesh. More details about these quantities will be given shortly. For the moment we underline the regularity of the generated grid which makes these equations interesting to us. Elliptic equations present in general a high regularity which leads to the fact that a small perturbation of the control functions induces a perturbation on all the nodes of the mesh. In Figure 4.1 the value of the function has been perturbed in a single grid node and it can be observed how the modification affects all the field.

2.2.2 Computation of $P_k^{initial}$

The adaptation process developed in this project requires an initial mesh. For every given mesh, the computation of $P_k^{initial}$ coefficients is possible according to the 3x3 system in (2.4) in unknowns P_k. This system has one real solution, and its solution is inexpensive but it must be carried out for every mesh node. Once $P_k^{initial}$ has been computed it is possible to adapt by adding a perturbation to the initial control functions by P_k^{adapt} which takes into account the local information about the need of refinement. The correspondent mesh coordinates X of the adapted grid are now the non-linear unknowns of the equation 2.4, as will be made clearer in the next section.

2.2.3 Computation of P_k^{adapt} and mesh adaptation

Let us focus now on the calculation of P_k^{adapt}. This could be done for instance according to the method proposed by Soni et. al. ([10]) which is based on the state variables in order to capture flow features as shocks. Instead, the computation of P_k^{adapt} for our purposes is performed exploiting the information provided by the local $\theta_{i,j}$ sensor. With this section we mean to explain how the local $\theta_{i,j}$ sensor can be used to compute P_k^{adapt}.

We start by considering $s^i : X \rightarrow [0,1](i = 1,2)$ fields. Each field s^i is associated to a geometric direction and it assumes large values in areas where the mesh needs to be refined. The definition of s^i depends on the criteria selected by the author. For Soni et al.[10] the field s^i was the component of the gradient of the Mach number in the direction i. We instead compute the field s^i from the component of $\wp(dF/dX)r$ in the i direction. So, the component of $\wp(dF/dX)r$ in every i direction is filtered and it is divided by the maximum value of $\wp(dF/dX)r$. These operations of filtering and division lead to the definition of a field for every topological direction whose values stay between -1 and 1. We define this field as s^i.

The field s^i is then used to compute a field s which compresses in one field the information connected to the different directions (present in each field s^i). Soni et al. built up s in [10] as follows:

$$s = 1 + s^1 \oplus s^2 \oplus s^3. \tag{2.6}$$

where the symbol $\oplus$ represents the Boolean sum ($a_1 \oplus a_2 = a_1 + a_2 - a_1 a_2$) and it is useful in this context because it has the propriety to exhibit high values if one of its arguments has a high value.

The field s is computed starting from s^i fields and it is shown in (Figure 2.9). We highlight that the difference with respect to the method proposed by Soni. et. al. is that s^i in the present case is directly linked to the adaptation sensor.

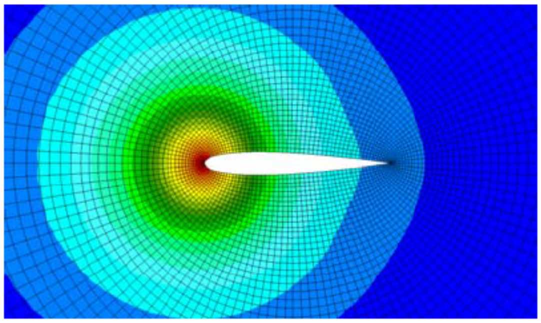


Figure 2.9: The field s is plotted for NACA0012, Re=∞ and Mach$<$ 0.7 during the adaptation based on the Cd

Finally the control functions P_k^{adapt} are defined as the derivative of this scalar field s respect to each topological direction:

$$P_k^{adapt} = \frac{s_{\xi_k}}{s}. \tag{2.7}$$

The field s is built in order to achieve values between 1 and 2 and so that the topological derivatives stay between 0 and 1. It follows that the variable P_k^{adapt} is always well defined. In our framework the s^i defined above corresponds to the norm of $\wp(d\mathrm{F}/dX)$ times the characteristic length in the i^{th} geometrical direction.

It has to be noticed that in practice, the highest values of P_k^{adapt} are significantly above the average. This is due to two main aspects. In fact, for coarse grid the regularity of the solution is not guaranteed while it is as the mesh size increases and the cell length tends towards zero (a demonstration is given in the [7] for Euler flows).Consequently it is necessary to smooth the

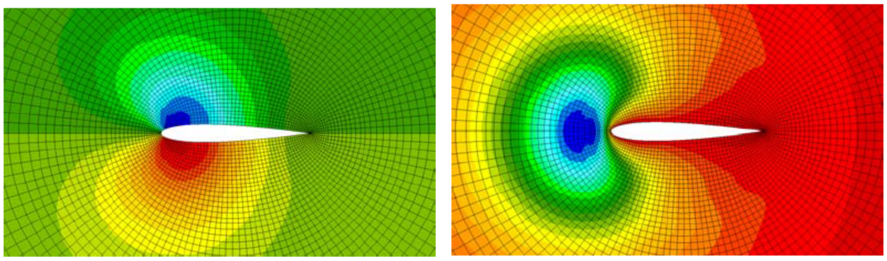


Figure 2.10: (Left) derivative P^{adapt} along i topological direction; (Right) derivative P^{adapt} along the j topological direction

sensor in order to avoid those strong discontinuities and the necessity to introduce a cut-off in order to partially supply at the high dispersion of the function around its average (details will be presented in the paragraph 4.2.6). The figure 2.10 shows P_i^{adapt} P_j^{adapt} which are respectively the criteria along the topological i and j directions built for a simple Euler-compressible case.

After these operations it will be possible to adapt the mesh according to the directions identified as 'important' by the sensor (Figure 2.11).

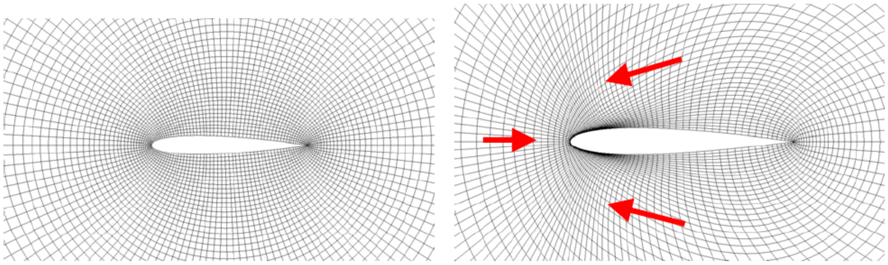


Figure 2.11: (Left) initial mesh; (Right) adapted mesh towards the directions calculated (red vectors)

2.2.4 Computation and role of ϵ

In agreement with equation (2.5), the proposed method employs the functions P_k^{adapt} in order compute the modified control function P_k used to generate the mesh according to equation (2.4).

The contribution of P_k^{adapt} to P_k is tuned by the scalar parameter ϵ. Different tests have been performed to calibrate this parameter. It is trivial to observe that:

- if this quantity tends towards zero, mesh will not be modified.
- if this quantity increases, a big modification is obtained.

Therefore ϵ controls the impact of the sensor at every adaptation step. The choice of this parameter is crucial. In fact, it has to be computed as a compromise between convergence speed (the smaller values of this parameter the larger number of iterations are required) and the confidence on the sensor θ Furthermore, important values of ϵ have to be avoid also because important modifications are linked with the possibility to create folding near critical mesh points.

2.3 Conclusions

In this chapter we gave some details about the proposed adaptation strategy.

In the first section we explained how we built the adaptation sensor and the qualification criterion starting from dF/dX. It was explain in this section that the adaptation criterion is able to take into account when there are regions of the mesh which are already very refined with respect to others. It has also been explained how the average operator allows to improve the performance of our sensor.

The second section meant to explain how the adaptation sensor is used to compute the adapted the mesh. With this purpose it was explained that the adaptation sensor is used to build the control functions P_i^{adapt} which, as well as the adaptation sensor has high values in the regions which have to be refined. $P_i^{initial}$ are instead the control functions associated to the initial mesh. Then, $P_i^{initial}$ and P_i^{adapt} are used to compute the new control function in accord to the equation (2.5). After the computation of the new control functions we showed how to compute the adapted mesh in accord with elliptic equations.

Chapter 3

Hierarchy of five C-type grids around the RAE2822 airfoil

The goal-oriented mesh adaptation based on dF/dX was already applied to the solution of Euler equations in [7]. In this paper the authors showed important properties of the present goal-oriented method which are now extended to the RANS equations in this work. Before showing a complete adaptation loop, which will be presented in chapter 4, it is interesting to analyse the behaviour of θ on a hierarchy of five grids. First, we will discuss the θ-sensor proprieties (paragraph 3.1), afterwards we will focus on the properties of the adaptation process (paragraph 3.2).

3.1 Behaviour of the mesh sensor and mesh quality criterion

With this paragraph we give a practical example. The definition of the sensor has been done in section 2.1. There the formulation of the local $\theta_{i,j}$ criterion and of the global θ criterion. The local criterion $\theta_{i,j}$ is a sensor which is formulated to detect the mesh zones whose sensitivity on the calculation of the goal F is important and the global criterion θ is useful to evaluate the quality of the entire mesh and his attitude for computing the goal function.

3.1.1 The mesh quality criterion and the introduction of the average operation

It is interesting to study the behaviour of the sensor on a hierarchy of grids in which we know *a priori* what is the level of qualification. We know *a priori* that among the five meshes with the same topology the quality of the grid increase with the level because for each level the number of nodes in each topological direction is multiplied by two. It is known that the error associated with the computation of the solution decreases with the decreasing of the dimensions of the cells. This lead to the fact that if the geometry is not modified the error associated with the estimation of the function F chosen will decrease as we move towards grid more refined.

As it can be observed in table 3.1, when the grid quality increases, θ decreases by about one order of magnitude so the correlation between the sensor θ and the quality of the grid is verified. The functions chosen in this example are the drag coefficient $F = Cd$ whose sensor is θ_1 and the lift coefficient $F = Cl_p$ whose sensor is θ_2.

It can be observed from the table 3.1 that the criterion implemented has a good correlation with the accuracy of the computation of F. The reduction of θ is due to the reduction of the both terms dF/dX and $r_{i,j}$. This last variable is reduced on every point of a factor two along the hierarchy while θ decreases of about one order of magnitude. This is enough to confirm that globally the sensitivity of F to the mesh displacement dX is smaller for the fine grids.

Table 3.1: Validation of the θ criterion on a mesh hierarchy (The symbol * means that the computation is retained to be too expensive)

Level	number of nodes	Cd	Cl_p	$\theta_1 * 10^{-7}$	$\theta_2 * 10^{-7}$	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
1	34 848	142.61	0.71837	41.265	61.380	28.659	40.369
2	135 200	123.93	0.73950	5.0184	7.7841	3.3888	5.2250
3	532 512	118.77	0.74988	0.8271	1.0175	0.5411	0.6693
4	2 113 568	118.33	0.75445	0.1861	0.1553	0.1344	0.0957
5	8 421 408	118.29	0.75583	0.0465	0.0203	*	*

That show the efficiency of this global sensor to detect if a mesh is more reliable or not for the calculation of an objective function.

It has to be said that all the nonlinear effects are not taken into account. The criterion is in fact based on a first order Taylor expansion of the functional F, so the lost of a perfect correlation is due to the fact that the functional F is a non linear function of dX. Furthermore, as we took a first order approximation, there are second order effects which we are not able to take into account.

About the local sensor, it has to be verified that the local criterion is able to detect the relative sensitivity of the mesh zones in every grids and that this sensitivity. We also expect that it assumes important values in the same zones for all the grids of a given hierarchy. Once again, the lost of a perfect correlation between the accuracy of F and θ is attributable to the non linearity of the functional F.

We mean now to investigate the behaviour of the local criteria into the details.

The mesh adaptation sensor and the introduction of the average operator

In section 2.1 we introduced both global and local criterion. Furthermore the criterion detects a better estimation of the second goal function with the increasing number of points and their values along the mesh are less important and the estimation of the functions actually reach his limiting values.

We will plot now the $\theta_{i,j}$-criterion field within the hierarchy of meshes we have used for the validation. The mesh hierarchy is shown on Figure 3.1, and the θ criterion is shown in the Figures 3.2 and 3.3 respectively for $F_1 = Cd$ and $F_2 = Clp$.

The colour-map used is based on the global criterion for the two functions along the iteration. It follows that it is possible to visualize the zones with an important sensitivity with respect to the global quality of every single case in order to point out for every grids which are the relative important zones detected.

It can be checked that for every function the zones detected are the same (for all the meshes). This fact is verified because the topology is the same so even if by adding points we achieve a better global quality, the geometry has to be adapted "in the same manner" in order to achieve a better performances on the estimation of the goal without increasing the number of points, so the calculation time. The natural consequence of this fact is the attempt to adapt the mesh. An important remark is that, roughly speaking, the redistribution of the points accordingly with this sensor will not pretend to get better the prediction of the global flow solution but it will certainly allow the best use of CPU and memory in order to compute the goal F.

With the knowledge of the behaviour within a mesh refinement we can now spend a few words to explain why those zones are detected. So we focus for instance on the third hierarchy mesh level. As it can be noticed in Figure 3.4 the main zones detected for the both function Cl and Cd are: The upstream, the shock, the trailing edge,the boundary layer and a zone over the shock. Naturally the magnitude of the local $\theta_{i,j}$ is different for the two functions but it has the same magnitude order.

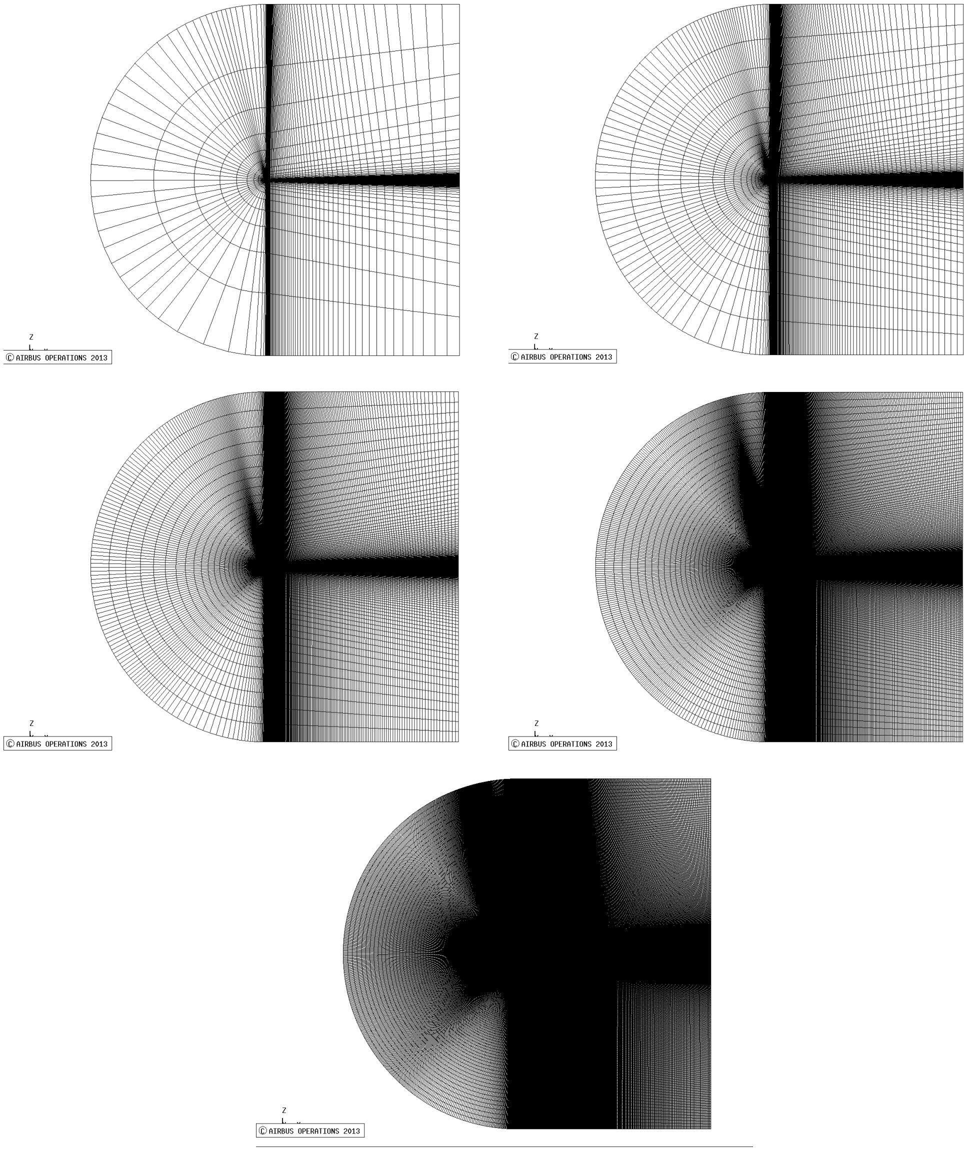
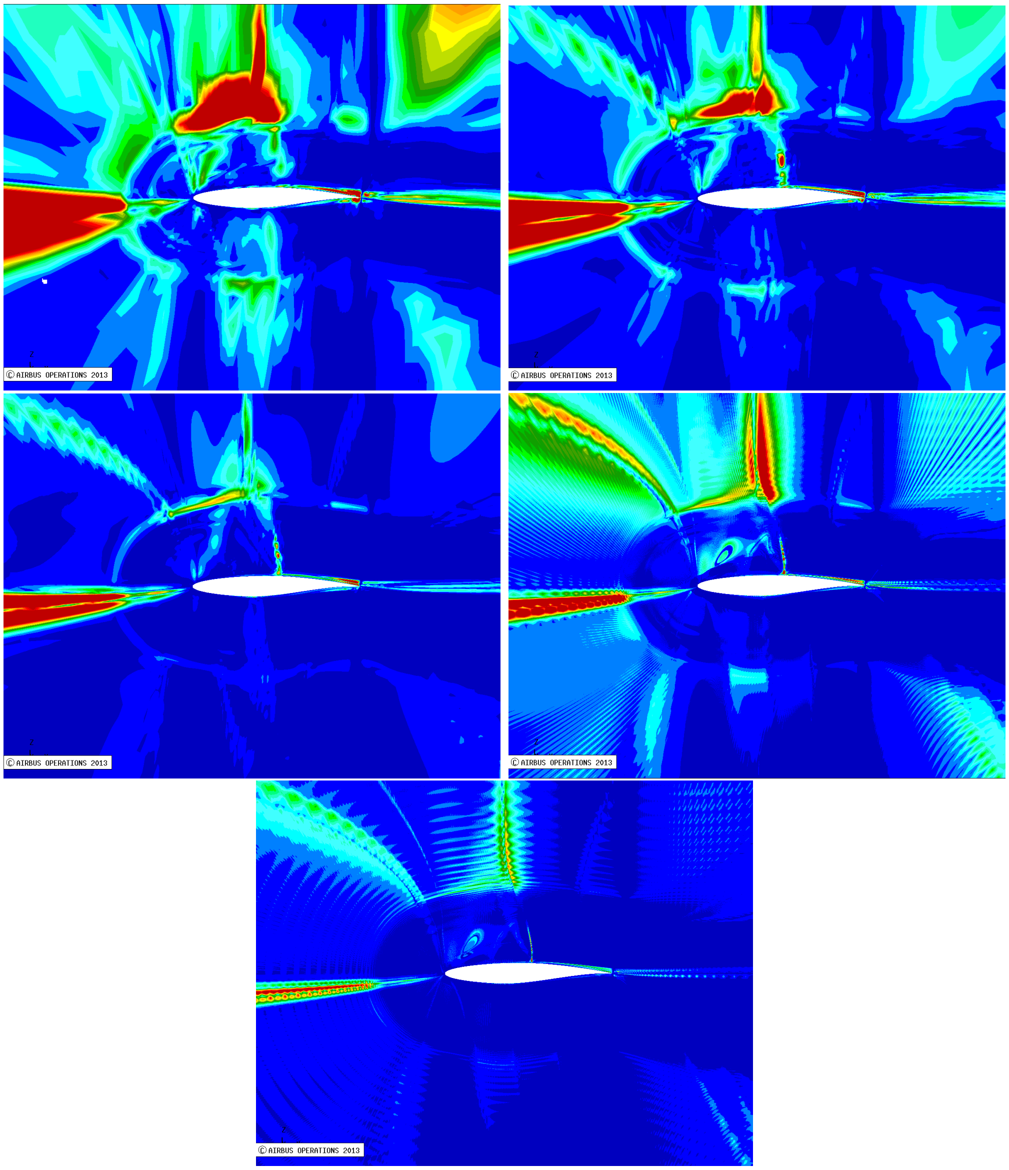


Figure 3.1: Mesh hierarchy



©AIRBUS OPERATIONS 2013

©AIRBUS OPERATIONS 2013

©AIRBUS OPERATIONS 2013

©AIRBUS OPERATIONS 2013

©AIRBUS OPERATIONS 2013

Figure 3.2: Criterion θ_1 plotted on the mesh hierarchy

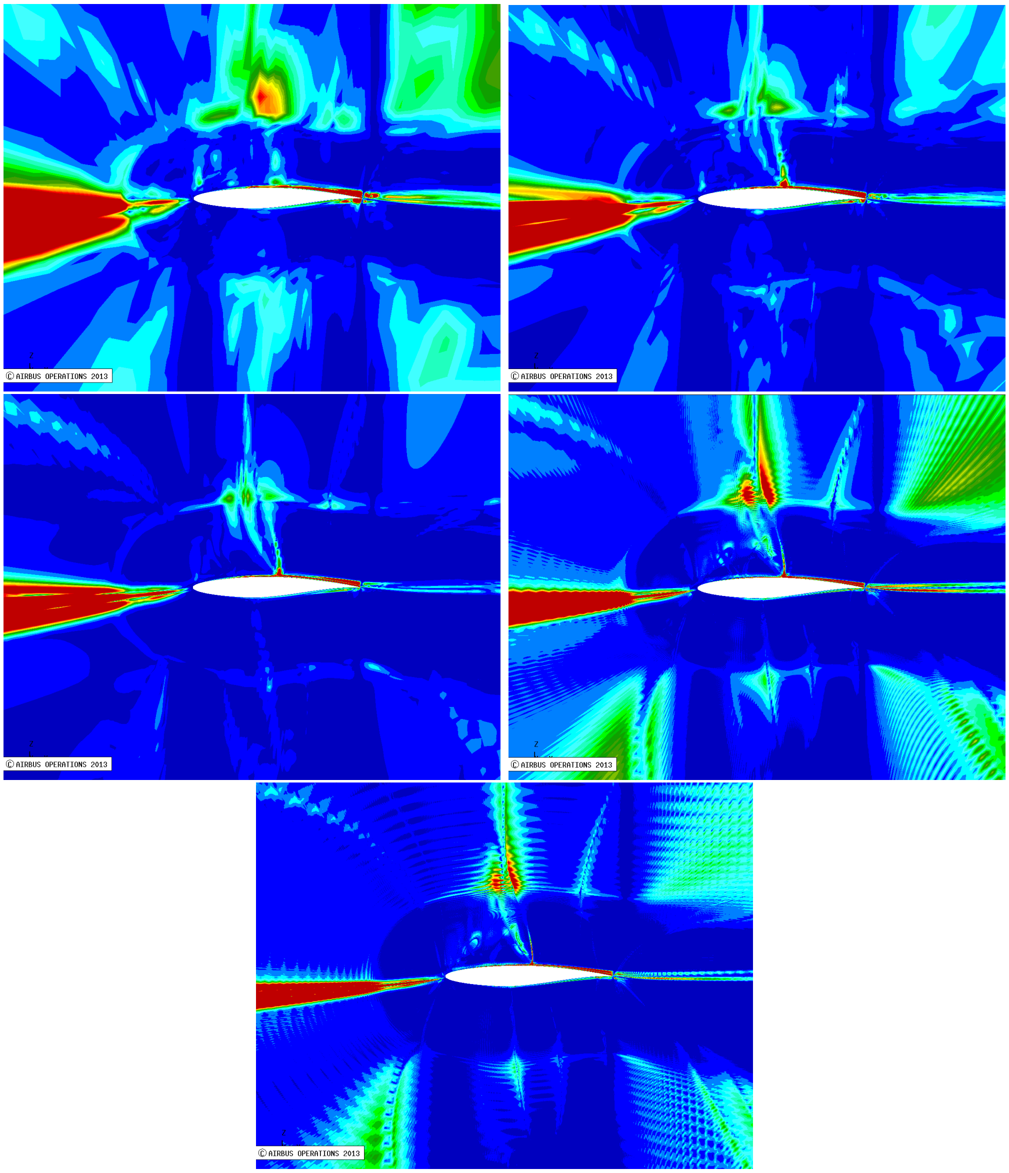


Figure 3.3: Criterion θ_2 plotted on the mesh hierarchy

The mesh is a structured multi-block mesh and is not uniform (see Figure 3.1). The computation of our sensor on this mesh allows the visualization of the following main zones:

Upstream

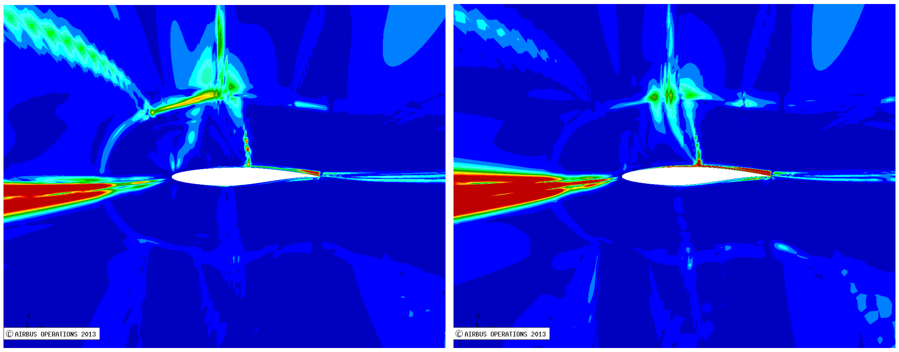


Figure 3.4: Criterion θ_1 near field region

The upstream is detected as the transport of numerical errors by advection would affect all the flow field. The upstream zone should be more refined in order to solve all the rest downstream. It is known that a criterion based on the features of the flow would not consider this fact as we will show in the section dedicate to the feature based adaptation.

Shock

The shock is detected (Figure 3.5) as it contains strong gradients and a coarse grid in this zone would provoke an increasing of the dissipation so a rough resolution in term of estimation of both θ_1 and θ_2. The criterion suggests (qualitatively in accordance with the feature based) a better refinement of this zone.

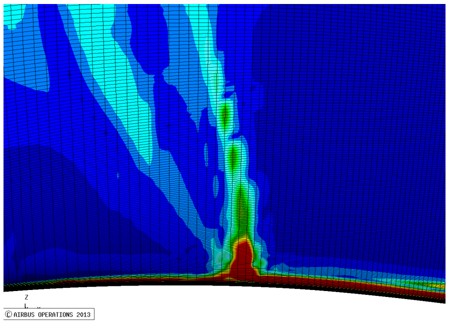


Figure 3.5: Criterion θ_1 around the shock

Zone over the shock

The detection zone over the shock which is shown in Figure 3.6 is due to the fact that in this zone the gradients are still important for the computation of the functions but the mesh given is coarse in this zone by construction. The criterion exhibit that the discontinuity in this zone should be avoided so a refinement in this zone is necessary in order to attenuate the numerical error dues to the rough resolution of the gradients.

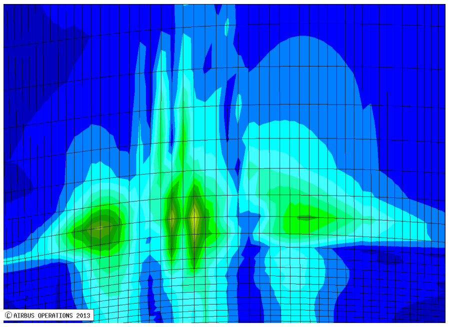


Figure 3.6: Zone over shock, θ_1 criterion plotted

Boundary layer

The boundary layer is a question a little bit more delicate. The boundary layer is turbulent and the method suffers of a lost in precision of the gradients calculated in this zone. This lost in precision is due to the fact that for the resolution of the adjoint equation, and in particular for the derivation of the term $\partial R/\partial W$ the thin layer and fixed turbulent viscosity hypothesis are taken into account (recent works at Onera are trying to provide the method with a complete linearisation of $\partial R/\partial W$ but for instance no work is concluded). The hypothesis play a marginal role on our adaptation process but they would be very important within the attempt to adapt very refined meshes.

Trailing edge

The trailing edge is an important zone within this context because the function θ exhibits high values. This is common to all adaptation process as well as the feature based. Numerical aspects and relative importance of the both θ_1 and θ_2 have been considered and treated in the chapter 4.

3.2 Influence of the number of points on the adaptation

With this paragraph we mean to study the effect of the adaptation method on an hierarchy of grids. The method just implemented was designed with the goal of generality. In fact it has to be proved that this method works with every grid and in particular that it is robust. We obtained a prove of this propriety thanks to an adaptation on three mesh refinement levels. This fact serves to show that in all the cases our method is reliable and robust so that it will tend to adapt the mesh with respect to the goal function for every given mesh.

Cd adaptation

Table 3.2: 1^{st} level Cd adaptation

Iteration	$Cd * 10^{-4}$	Cl_p	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
0	142.61	0.71837	28.659	40.369
1	133.90	0.73475	22.552	35.428
2	133.75	0.73478	22.480	36.799
3	133.49	0.73288	21.861	35.713

Table 3.3: 2^{nd} level Cd adaptation

Iteration	$Cd * 10^{-4}$	Cl_p	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
0	123.93	0.73950	3.388	5.225
1	120.02	0.74438	2.431	4.287
2	119.45	0.74189	2.418	4.275
3	119.41	0.74194	2.437	4.203

Table 3.4: 3^{th} level Cd adaptation

Iteration	$Cd * 10^{-4}$	Cl_p	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
0	118.77	0.74988	0.541	0.669
1	118.19	0.75133	0.521	0.570
2	118.07	0.75100	0.528	0.570

Cl_p adaptation

Table 3.5: 1^{st} level Cl_p adaptation

Iteration	$Cd * 10^{-4}$	Cl_p	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
0	142.61	0.71837	28.659	40.369
1	136.84	0.73421	25.344	34.827
2	135.69	0.74230	24.140	31.687

Table 3.6: 2^{nd} level Cl_p adaptation

Iteration	$Cd * 10^{-4}$	Cl_p	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
0	123.93	0.73950	3.388	5.225
1	120.81	0.74559	2.760	4.006
2	120.52	0.74613	2.689	3.748
3	120.31	0.74660	2.700	3.671

Table 3.7: 3^{th} level Cl_p adaptation

Iteration	$Cd * 10^{-4}$	Cl_p	$\bar{\theta_1} * 10^{-7}$	$\bar{\theta_2} * 10^{-7}$
0	118.77	0.74988	0.541	0.669
1	118.32	0.75028	0.510	0.600
2	118.30	0.75068	0.511	0.574
3	118.19	0.75061	0.510	0.563

3.3 Conclusions

With this chapter we have been able to give numerical evidences about the local sensor and the global qualification criteria. Furthermore we showed in the first paragraph that the global qualification criteria is able to detect when a mesh is more or less adapted for the calculation of the chosen functional as decreasing values of θ are calculated for grids of increasing performances. With this paragraph we also exhibited that the local criterion $\theta_{i,j}$ detects the same zones of the grids which have to be adapted. As the grids have the same topology and different number of points, this last results confirm that an improvement of the accuracy can be achieved by a redistribution of the points in accordance with the zones detected by the local sensor. With the second paragraph it can be noticed that an improvement of the accuracy is obtained by the redistribution of the points thanks to the local sensor on different grids. Furthermore, the adaptations for the different grid points number showed that this method is robust as it allows an improvement of the solution accuracy even when a mesh is coarse (first mesh level). The synthetic conclusion of this chapter is that our strategy is reliable and it can be actually used to adapt a given mesh for the computation of a goal function F as we show in the following chapter.

Chapter 4

Application for RANS flows around the RAE2822 airfoil

The goal of this chapter is the presentation of an application of our proposed goal-oriented mesh adaptation methodology described into the chapter 2. The chapter will be organised as follows:

- Mesh adaptation process and software
- Presentation of the steps involved in the adaptation process
- 2D goal function grid adaptation examples
- Feature based method adaptation and comparison with θ method adaptation

In the previous chapter we described the principles of this method and we deal now with all the computational aspects and considerations necessary to apply our method. In the first section we will provide the reader with a brief description of the software used. In the second section we will go into the details of every step adaptation in order to underline the different considerations, and parameters used within the computation. Finally we will give numerical evidences about the mesh adaptation process. Before to start it has to be noticed that the present method has be designed with the only requirement of structured grids, so other choices as the scheme, the turbulent model used with the specification annexes are necessary for the comprehension of this particular example and its analysis.

4.1 Mesh adaptation process and software

With this section we mean to illustrate the software used in our framework.

The proposed goal-oriented mesh adaptation illustrated in the chapter 2 has been implemented as independent from the framework and it has been showed that it is clearly preferable in respect to the feature based methods in terms of accuracy presented in chapter 4.4. Furthermore the application of a method based on the discrete adjoint in a shape optimization context which also is based on the adjoint gives others additional advantages.

The software involved in our adaptation loop are:

- The CFD solver we used to calculate the direct solution W and the adjoint vector Λ is *elsA*, which is a finite volume code developed by ONERA [1].
- The computation of the mesh sensor has been performed with an in-house code which uses the method developed and presented in chapter 2 in order to compute the quality criteria both local and global ($\theta_{i,j}$ and θ) and generate the weight functions (P_k^{final}) which will induce the re-meshing

- The last step of our mesh adaptation loop consists in the generation of the new grid (X^{n+1}). This operation is implemented in the code Lama 3D using the methods previously proposed in paragraph 2.2.3.

4.2 Presentation of the steps involved in the adaptation process

4.2.1 CFD solver

In our application the numerical scheme used by *elsA* for the evaluation of the primal solution is based on Roe flux [9]. The Roe method is a finite volume scheme based on solving a localized Riemann problem to calculate the flux at a given face of the domain. The basic premise of this problem is that changes in a flow can be transmitted only through entropy waves and acoustic waves, and only at some given speeds, which represent the eigenvalues of the governing non-linear equation system. In one-dimension, there are three wave speeds, corresponding to the entropy wave at the speed the fluid is travelling, and acoustic waves representing the speed of sound relative to the fluid speed in the upstream and downstream directions (note that these waves may not actually be in the upstream and downstream directions respectively, but this is the sense in which they are defined). Since the solution of the equation set changes only across one of these waves, the solution of the Euler equations at any point in space and time can be represented by a summation of the state to the extreme left or right of the space, plus (or minus) one or more of the state changes across these waves. Since the Euler and the RANS equations are non-linear, the corresponding Riemann problem is non-linear as well. This could be too expensive to calculate in some cases, and Roe found out that a properly selected approximate problem does the job just as well in most cases and saves on calculation complexity [14].

The schema used is a second order in space scheme. This characteristic is obtained thanks to the application of a MUSCL (monotone upstream-centred schemes for conservation laws) scheme for the reconstruction of the states (details about this method can be found in [14]). The flux limiter used is the one of van Albada [13].

In our context additional details about the scheme are not essentials. The main issue related to our process is related to the consistency of the flow solution within the anisotropy of the mesh in fact the second order convergence according to MUSCL scheme is guaranteed for regular grids only. Beside the attempt on generating smoothed and regular grids, the adaptation process is made to generate a change of geometry which would lead in some cases at the lost of a perfect isotropy. The main consequence of this fact is that beside the corrections to take into account the anisotropy implemented by the developers of this software [1], as far as we go from this condition the mesh solver will converge slower to the solution so that additional errors attributed to this anisotropy will occur in the mesh adaptation process. Another numerical issue that arise within the next steps is the difficulty in the inversion of the matrix $\partial R \partial W$ when it is weakly conditioned. It is known that a discretization of the convective fluxes with a decentred formula (i.e upwind) assure a better conditioning of this matrix with clear advantages for the computation of his inverse [6]

4.2.2 Post processing

The solution in terms of state variables W given by the CFD solver are then elaborated by the post processing by an Airbus in-house tool named Zapp. This elaboration mainly consist on the computation of the goal function and the partial derivatives needed for the computation of the adjoint vector in equation (1.2.2).

4.2.3 Adjoint vector

The adjoint vector has been introduced in paragraph 1.2.3. The formulation of the adjoint problem as illustrates in paragraph 1.2.3 is the following: Compute Λ^T t.c

$$\frac{\partial F}{\partial W} + \Lambda^T(\frac{\partial R}{\partial W}) = 0. \tag{4.1}$$

The last equation can be transposed obtaining

$$\frac{\partial F}{\partial W}^T + (\frac{\partial R}{\partial W})^T\Lambda = 0. \tag{4.2}$$

If we add and subtract an approximation of the second member of equation 4.2 we have:

$$\frac{\partial F}{\partial W}^T + (\frac{\partial R}{\partial W})^T\Lambda + (\frac{\partial R}{\partial W})^{T^{APP}}\Lambda - (\frac{\partial R}{\partial W})^{T^{APP}}\Lambda = 0, \tag{4.3}$$

where APP stands for approximation. We notice that until now no approximations have been performed. Rearranging this expression leads to:

$$(\frac{\partial R}{\partial W})^{T^{APP}}(\Lambda^{n+1} - \Lambda^n) = -(\frac{\partial F}{\partial W}^T + (\frac{\partial R}{\partial W})^T\Lambda^n), \tag{4.4}$$

which is a classical Newton iterative system which converge to the exact solution in fact if Λ^{n+1} tends towards Λ^n the term on the left of the equal tends to 0 and on the right we recognise the exact adjoint equation.

It is clear that now the term $(\partial R/\partial W)^{T^{APP}}$ has to be inverted in order to calculate the solution, instead of $(\partial R/\partial W)^T$. This term has to be chosen by providing a good conditioning and an easy inversion. We do not enter into the details of the computation but it has to be noticed that this term is the same that was used for the direct flow solution. The consequence is that his inversion can use the same benefit as the inversion that was calculated in the direct more.

Artificial dissipation

The matrix $\partial R/\partial W^T$ is badly conditioned. It follows that in order to compute the adjoint an additional term of dissipation has to be introduced into this equation as:

$$(\frac{\partial R}{\partial W})^{T^{APP}}(\Lambda^{n+1} - \Lambda^n) = -(\frac{\partial F}{\partial W}^T + (\frac{\partial R}{\partial W})^T\Lambda^n) + AD, \tag{4.5}$$

where AD is the Additional Dissipation term which has to be chosen as a compromise between the accuracy of the solution and the robustness required. Our choices are however piloted from the experience.

4.2.4 Mesh Sensor

With this section we mean to give details about the tool Mesh Sensor. This program is basically made in order to calculate the control functions P_k used for the remeshing (as explained in the following section) and the mesh quality indicator starting from the information which lie on the quantity dF/dX. It has to be noticed that the input dF/dX is a very irregular field while the field of control functions has to be as regular as possible in order to avoid to have an irregular. That is why the information treated for the generation of the control functions are divided into the following steps:

- Projection of dF/dX
  This projection means to eliminate the component of dF/dX which is normal at the skin of the solid. This component would affect the geometry and in the present context this has absolutely to be avoided. The operation which allows to achieve this projection as it has been introduced in 2.1 is denoted as $\wp(dF/dX)$.

- Spatial mean of $\wp(dF/dX)$
  In this second step the operation of averaging is performed averaging the $\wp(\frac{dF}{dX})$. In fact following the details already given in 2.1.2 and the study of radius L influence it is possible to chose a value for this radius and the operation of average will be implemented giving as output an average field. All the consideration we have underlined in the section 2.1.2 are valid and serve to orientate the choice. In general the introduction of an operation of average allows to have more coherent field because every point is affected by a quantity proportional with the sensitivity but inversely proportional at the distance from the current node. The compensation effects are taken into account. On the example we will give of mesh adaptation, we chose a radius of about $L = 20$ which correspond a 0.02 of the length cord. Furthermore this operation is tested to be useful for many reasons even if the choice of a radius L should be made by knowledge of the resources available. The computation time of this step will certainly increase with the number of node X_j whose distance form the node X_i is smaller than $L|X_j - X_i|$. For this reason a study on the radius influence has been made whose results can be found. After this operation it is necessary to introduce another average step. We remark that the first one was necessary in order to avoid the consideration of the possibility of normal displacement so that even during the average operation this "information" would not spread to the other nodes near the skin, and in the second step it is necessary because the information of dF/dX is spread from the neighbour to the skin nodes and it allows to definitely eliminate the skin-normal components.

- Computation of θ
  The next step is the calculation of the θ criterion. After the computation of the half of the distance between every single node and the nearest one ($r_{i,j}$), it is performed the multiplication between $|dF/dX|_{i,j}$ and $r_{i,j}$ for the X, Y, Z directions defining respectively $\bar{\theta_x}$, $\bar{\theta_y}$, $\bar{\theta_y}$ This operation allows to define the $\theta_{i,j}$ criterion, which gives information about the local quality of the mesh, as the square root of the different components in every node (i,j). The associate global criterion, as it has been defined in the previous chapter is $\sum_{i,j} \frac{1}{N_i,N_j}\theta_{i,j}$. This number allows to define the global quality of the current mesh for the calculation of the required output F and it can be compared with other grids.

- Computation of θ_x^1, θ_y^1
  The local criterion allows to detect the zones with an high influence on the goal. It is important to consider that the field just created presents strong discontinuities. In order to perform a successful mesh adaptation based on dF/dX we retain that it is necessary reduce the leakage of the fields θ_x^1 and θ_y^1. In fact those fields present peaks with a magnitude that can achieve 1000 times the average and an adaptation induced directly from this field would not be efficient as it would finally detect the relative importance of some points on the grid. In order to reduce that dispersion we retain to introduce a cut-off on these fields. The reduction of the dispersion allows to detect simultaneously a more consistent number of points so that in every iteration step the global quality is better reached. This cut-off is built with two values, the cut-off max and the cut-off min (we also performed a study concerning the influence of the different values chosen on the mesh adaptation process). Furthermore all the values of θ bigger than cut-off max are turned to cut-off max so as the values of θ smaller than cut-off min are turned to cut-off min. This process will allow to define relative important weight function on more zones simultaneously (otherwise neglected because of the high relative importance of θ with respect to the other zones. The function affected by the cut-off is defined θ^1 (cut-off effects according to the signal theory) and it is originally but on the different topology directions.

- Construction of s
  As it has been defined in the section 2.2.3, the field s is given thanks to the boolean operator as:

$$s = 1 + s^1 \oplus s^2 \oplus s^3$$

where the different components s^i are defined as:

$$s^i = \frac{\bar{\theta}_i}{max(\bar{\theta}_i, \bar{\theta}_j)}. \tag{4.6}$$

It appears clear now that the cut-off introduced above has an important role, in fact $max(\bar{\theta}_i, \bar{\theta}_j, \bar{\theta}_k)$ can be one thousand times the average in a few points so that after the computation of this ratio all the others terms would be considered negligible.

The variable s as well as s^x, s^y and s^z present now a reduced dispersion with all the benefit of the case in order to induce the adaptation.

- Second cut-off applied to the field s
  The purpose of this operation has to be regarded once more in the sense of the following steps. The importance of the construction of regular control function require the introduction of this cut-off in order to reduce the highest values associated with this field.

- Smoothing
  The weight functions are now considered to have acceptable values to induce the remeshing. The only remaining issue is the absence of regularity. A big amount of tests has been performed in order to understand the best way to smooth the fields of weight function to get them ready for a remeshing. The proposed method is a good compromise between the analysis of the field on the practical example that we will show in the following. It is shown that a successful method to induce a good quality weight field is made up from: One step isotropic smoothing, one mesh of anisotropic smoothing plus another step of isotropic smoothing. It is important to understand that this method allows to reach a good compromise between the orientation of the sensor (which is save by the anisotropic smoothing) and the homogeneity of the field obtained in particular thanks to the isotropic iterations.

4.2.5 Grid generation and adaptation (Lama 3D)

As it has already been announced the elliptic equations system used for the grid generation and adaptation is used in order to guarantee the regularity of the grid. One can state in other words that according with the Laplace solution, a perturbation of the function in a point of the domain has a propagation along the entire domain. In our framework the function is the volume mesh position of every node of the grid and the perturbation is the variation of this function (Figure 4.1) with respect to the initial grid coordinates (Figure 4.1 in blue). In effect the perturbation of a single node position generates a different mesh where every single node has been displaces (Figure 4.1 in red).

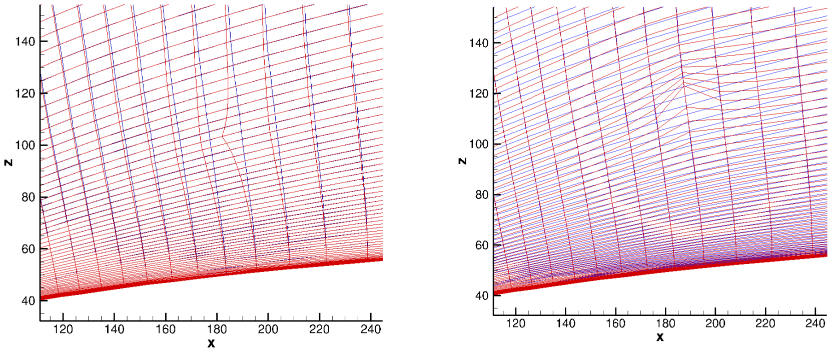


Figure 4.1: Effect of single node displacement in direction i (Left); and in direction j (Right)

This behaviour allows to generate a regular grid despite the discontinuities that beside the smoothing iterations introduced are present within the control functions P. From the practical point of view the equation (2.4) needs to be re-written in order to be computed, because we dispose naturally of the contra-variant metric tensor, so we avoid their direct calculation as they depend on the covariant metric tensor through the Jacobian. The general formulation of the problem, as it has been presented in the paragraph 2.2.1 is:

$$\sum_{i,j=1}^{3} g^{ij} x_{\xi^i \xi^j} + \sum_{k=1}^{3} g^{kk} P_k x_{\xi^k} = 0. \tag{4.7}$$

Where the expressions of the contra-variant metric tensor components are related to the co-variant metric tensor components through the Jacobian ($J^2 = det(a_{i,j})$). Multiplying the last equation by J^2 we obtain:

$$\sum_{i,j=1}^{3} G^{ij} x_{\xi^i \xi^j} + \sum_{k=1}^{3} G^{kk} P_k x_{\xi^k} = 0. \tag{4.8}$$

where $G^{ij} = J^2 g^{ij}$ and in particular:

$$G^{1,1} = (x_\xi, x_\xi) \; G^{1,2} = (x_\xi, x_\eta) \; G^{2,2} = (x_\eta, x_\eta)$$

The discretization of the PDEs equations given in a 2D case (the extension to 3D is straightforward). We approximate the second order derivatives $x_{\xi^i \xi^i}$ as:

$$\begin{aligned} x_{\xi\xi} &= x_{i-1,j} - 2x_{i,j} + x_{i+1,j} \\ &= x_{i-1,j} + x_{i+1,j} - 2x_{i,j} \\ &= x_{i-1,j} + x_{i+1,j} - 2x_{i,j} \end{aligned}$$

so that

$$\tilde{x} = x_{i-1,j} + x_{i+1,j}$$

and the same kind of expression can be achieved for every set of $x_{\xi^i \xi^i}$ coordinates.

$$G^{11}(x_{\xi\xi} + P_{x_\xi}) + G^{22}(x_{\eta\eta} + Q_{x_\eta}) + 2G^{12} x_{\xi\eta} \approx 0. \tag{4.9}$$

Now, substituting in the PDEs equations the approximate expression for $x_{\xi^i \xi^i}$ that we just wrote above we obtain:

$$G^{11}(x_{\xi\xi} - 2x_{i,j} + P_{x_\xi}) + G^{22}(x_{\eta\eta} - 2x_{i,j} + Q_{x_\eta}) + 2G^{12} x_{\xi\eta} \approx 0 \tag{4.10}$$

The approximated value is obtained solving (4.10) when it is equal to zero thanks to an approximation of x Rearranging the last equation we have:

$$x_{ij} = \frac{1}{2}(\frac{(G^{11}(\tilde{x}_{\xi\xi} + Px_\xi) + G^{22}(\tilde{x}_{\eta\eta} + Qx_\eta))}{G^{11} + G^{22}} + \frac{2G^{12} n_{\xi\eta}}{G^{11} + G^{22}}). \tag{4.11}$$

and defining:

$$\hat{x}_{ij} = \frac{1}{2}\tilde{x_{ij}}$$

$$x_{i,j} = \frac{G^{11}(\hat{x}_{\xi\xi} + \frac{1}{2}Px_\xi) + G^{22}(\hat{x}_{\eta\eta} + \frac{1}{2}Qx_\eta)}{G^{11} + G^{22}} + \frac{G^{12} x_{\xi\eta}}{G^{11} + G^{22}}. \tag{4.12}$$

with control parameter P and Q to be chosen (P, Q and R in the 3D case). The equation written in this form can lead to an iterative method to approximate the mesh volume X within the iterations:

$$x_{i,j}^{n+1} = A(x_{i,j}^{n}). \tag{4.13}$$

$$A = \frac{G^{11}(\hat{x}_{\xi\xi} + \frac{1}{2}Px_{\xi}) + G^{22}(\hat{x}_{\eta\eta} + \frac{1}{2}Qx_{\eta})}{G^{11} + G^{22}} + \frac{G^{12}x_{\xi\eta}}{G^{11} + G^{22}}$$

We define $P_k^{initial}$ the control functions computed at the initial step thanks to the PDEs elliptic equations. We notice that at the initial step (for a given grid to be adapted)the unknown is P_k functions which are computable for every grid. It is in fact verified that for a generic mesh X well defined a set of P_k functions exists and it is unique. In fact, during the iterations, the mesh is adapted modifying these control functions, and a mesh X is determinate for every set of control function P_k given. The computation of P_k becomes of primary importance for a mesh adaptation method. The basic idea is that the function P_k contain the information given by our sensor in order to adapt the grid coherently. Exhaustive explanations will be given in the following paragraph. For the following iteration steps the unknowns of the elliptic equations will be the points x_i generated thanks to a set of P_k functions.

4.2.6 P_k effects

Here we will present the induced choice for the control parameter which are defined for a given mesh and have to be chose in order to modify it. One can intuitively think that a good method in order to compute the functions P_k at every step of the method is to consider the initial value of P_k corresponding at a given mesh to be adapted and add at this function within each adaptation step a function which is able to take into account the information given by the sensor $d\mathrm{F}/dX$. If we define:

P_k^n	control function
P_k^{init}	control function on direction k at step 0
$\tilde{P}_k^n$	control function on direction k at step n computed by the sensor $d\mathrm{F}/dX$
P_k^n	control function on direction k at step n used for the mesh generation $d\mathrm{F}/dX$
s	relaxation term coefficient

we are able to compute the control functions for every n adaptation iteration step as:

$$P_k^1 = (1-s)P_k^{init} + s\tilde{P}_k^1. \tag{4.14}$$

$$P_k^n = (1-s)P_k^{n-1} + s\tilde{P}_k^n. \tag{4.15}$$

Where s is the relaxation coefficient which has to be chosen as a compromise between the convergence velocity (obtained ideally s=1) and the required robustness. The P_k in the step n has also a role within the computation of P_k at the step $n+1$ in order to induce a 'changing' of the grid weighted by the changing at the precedent step in order to avoid the oscillations.

Another important issue of initial grids is that they can be generated in a way that doesn't keep into account the physic of the problem. In general these grids show very high refinements in zones with a relative low influence on the functional of interest and the physic of the problem. This fact lead in general to slow convergences of these grids and in some cases to a determination of such an anisotropy of the grid in the attend reach the adaptation which will cause an augmentation of the error especially when schemes which do not use a correction for anisotropy.

Furthermore, in order to treat general initial grids it is possible to define a "cut-off" which allow to distinguish zones with important values of the function P_k (with an important influence on the computation of F) and other zones which are given "too much refined" and present low values of P_k functions. In practice we chose to turn to 0 the values of P_k as they are under the cut-off value. In this way we allow our method to use those zones which are too much refined (having a very low values on the criteria) to be de-refined in order to follow the refinement induced by high values of the criteria in different zones.

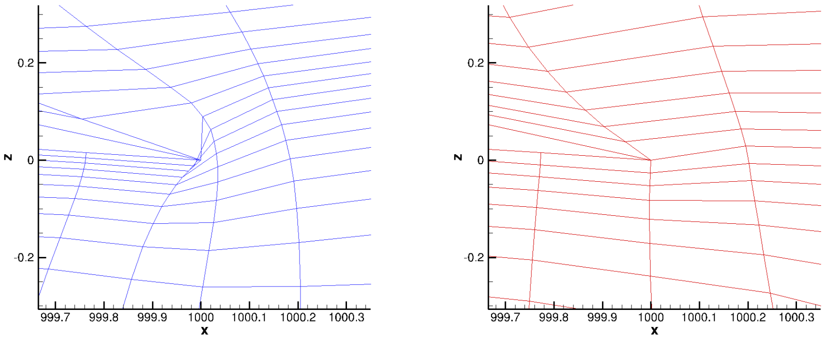


Figure 4.2: Effect of distortion of the grid due to an attempt of adaptation in the trailing edge where the last point cannot be displaced i (Left); j (Right)

4.2.7 Post processing FFd72

The solution in terms of state variables W given by the CFD solver are then elaborated in order to calculate the variables presented in this paragraph. This quantities are the more generic goal function which is used to estimate in CFD. These quantities are calculated following different approaches in order to highlight the quality of the numerical solution.

The different approaches are the near-field computation and the far-field computation. In an ideal case these two approaches should lead to the same results but in practice there is a difference dictated by the numerical errors.

Near-field analysis

If we take the sum of the forces on the skin perpendicular to the infinite velocity direction we obtain the lift and his associated dimensionless quantity (Cl) if we divide it by ($f1/2\rho V_\infty^2 S$)

The lift coefficient (Cl) can be decomposed in two different contributions. The contribution of the pressure (Cl_p) and the contribution due to the friction forces (Cl_f) as:

$$Cl = Cl_p + Cl_f$$

At the same manner we can define the drag coefficient (Cd) by taking the sum of the forces on the skin in the direction of the infinity velocity.

This quantity is also decomposed into the contribution of pressure and the one due to the friction forces as:

$$Cd = Cd_p + Cd_f$$

$$Cl_p = \frac{\int\int_{S_S}(p - p_\infty) n_z dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.16}$$

$$Cl_f = -\frac{\int\int_{S_S}(\vec{\tau}_z \vec{n}) dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.17}$$

$$Cd_p = \frac{\int\int_{S_S}(p - p_\infty) n_x dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.18}$$

$$Cd_f = -\frac{\int\int_{S_S}(\vec{\tau}_x \vec{n}) dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.19}$$

Far-field analysis

If we now move far from the surface of the skin we can perform another type of analysis. Theoretically the results taken with this second method should be coincident with the ones calculated with the near field analysis approach but in computational fluid dynamics more or less important differences are present **??**. The differences found in practice are caused by the numerical error. This analysis allow to predict the drag due to different factors as the shock, the induced drag, and the viscous drag considering separately the integration along the different surfaces associated with these different features.

$$Cd = Cd_{wave} + Cd_v + Cd_i. \tag{4.20}$$

The component associated to the shock is obtained as integration along a surface S_W which surrounds the shock as:

$$Cd_{wave} = -\frac{\int\int_{S_W} \vec{f_i}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.21}$$

The component associated to the viscosity is calculated as the sum of the following components:

$$Cd_v = Cd_{vp} + Cd_f$$

where

$$Cd_{vp} = -\frac{\int\int_{S_V} \vec{f_i}\vec{n}dS + D_p}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.22}$$

and

$$Cd_f = -\frac{\int\int_{S_S} \vec{\tau_x}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.23}$$

while the component due to the 3D finite span-wise effects is:

$$Cd_i = -\frac{\int\int_{S_I} \vec{f_i}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.24}$$

It is also defined the Oswald factor otherwise called aerodynamic efficiency factor:

$$OF = \frac{C_L^2}{\pi ARC_{Di}}. \tag{4.25}$$

where

$$AR = \frac{b^2}{S_{ref}}. \tag{4.26}$$

so that we can define the physical irreversible drag as

$$Cd_{irr} = Cd_v + Cd_w$$

and the spurious component of the drag as

$$Cd_{sp} = Cd_{OSW} - Cd_{irr}$$

with

$$Cd_{OSW} = \frac{\int\int_{S_I} \vec{f_{vw}}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.27}$$

and

$$Cd_{irr} = \frac{\int\int_{S_V+S_W} \overrightarrow{f_{vw}}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.28}$$

$$Cd_{rev} = \frac{\int\int_{S_I} \overrightarrow{f_v w}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}} - \frac{\int\int_{S_v+S_W} \overrightarrow{f_v w}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.29}$$

$$Cd_{irr} = Cd_v + Cd_w. \tag{4.30}$$

In two dimensional case the definitions assume an aspect partially different, in particular:

$$Cd_{sp,irr} = \frac{\int\int_{S_I} \overrightarrow{f_{vw}}\vec{n}dS - \int\int_{S_v+S_W} \overrightarrow{f_{vw}}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.31}$$

$$Cd_{sp,rev} = \frac{\int\int_{S_I} \vec{f_i}\vec{n}dS}{\frac{1}{2}\rho V_\infty^2 S_{ref}}. \tag{4.32}$$

The quantities $\vec{f}_{vw}$ and f_i are calculated as from the dynalpy-like vector observing that it is possible to chose a decomposition as:

$$\vec{f} = -\rho(u - u_\infty)\vec{q} - (p - p_\infty)\vec{i} + \vec{\tau}_x. \tag{4.33}$$

as

$$\vec{f} = \vec{f_i} + \vec{f}_{vw}$$

$$\vec{f_i} = -\rho(u - u_\infty\vec{q} - \Delta\bar{u})\vec{q} - (p - p_\infty)\vec{i}$$

$$\vec{f}_{vw} = -p\Delta\bar{u}\vec{q} + \vec{\tau}_x$$

and

$$\Delta\vec{u} = u_\infty\sqrt{1 + 2\frac{\Delta H}{u_{\infty^2}} - \frac{2}{(\gamma-1)M_\infty^2}[(e^{\frac{\Delta S}{r}})^{\frac{\gamma-1}{\gamma}} - 1]} - u_\infty. \tag{4.34}$$

and ΔH Δs are respectively the variations of enthalpy and entropy relative to their freestream values ad $\vec{q}$ is the velocity vector (Van der Vooren formulation).

We observe that for our case the goal functions are defined as integration on the skin surfaces so that we attempt to adapt for the near fields quantity. The reduction of the spurious component of the drag is in our case a criterion sufficient but it is not necessary to achieve a good prediction of the drag by near field integration

4.3 2D goal function grid adaptation examples

Numerical evidences about an adaptation is given in this section. The values we retained to exhibit are the force coefficients Cd and Cl_p with the relatives θ criteria ($\theta[Cl_p]$ and $\theta[Cd]$). The criterion we used to stop the iterations is based on the quality criteria. When it does not change significantly we stop the process. Another criterion used is the changing on the function estimated. We adopted the both criteria because as every step is computationally expensive, we can not afford to compute the convergence especially because the iteration start to give rise to an oscillating behaviour, and we retain to stop the iteration before as during the oscillation the quality of the solution will not change significantly.

4.3.1 Test case presentation

The 2D profile used for the adaptation example is the RAE2822. This is a trans-sonic profile which is often used in literature in order to compare results. This adaptation example is performed on a multi-blocks grid of about 135 200 nodes. The domains in two dimensions are usually very big compared to the solid shape of the body, and for this case the domain length is considered to be sixty times the cord length. The detail about the flow and adjoint computation are given in section 4.2, $\bar{\theta}_1 = \bar{\theta}_1[Cd]$ and $\bar{\theta}_2 = \bar{\theta}_2[Cl_p]$ are the two goals chosen. The calculation is performed on a airfoil RAE2822, the Reynolds number is Re=6.5 10^6, the Mach number is $M_\infty = 0.725$, the angle of attack is $\alpha = 2.466$ using the Spalart-Allmaras model. The flow solution in terms of Mach is exposed in Figure 4.3

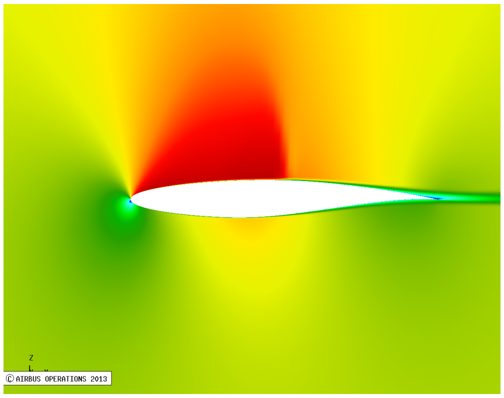


Figure 4.3: Mach around the RAE2822 airfoil

4.3.2 Mesh adaptation results

The adaptation for $\bar{\theta}_1 = \bar{\theta}_1[Cd]$ is:

Table 4.1: Cd adaptation example $\bar{\theta}_1$

Iteration	Cd	Clp	$\bar{\theta}_1$	$\bar{\theta}_2$	$Cd_{Sp,rev}$	$Cd_{Sp,irr}$
0	123.93	0.73950	3.3875	5.2251	3.32	2.95
1	120.01	0.74423	2.7638	4.1711	3.42	2.71
2	119.97	0.73999	2.7542	4.1759	3.43	2.78
3	119.46	0.74152	2.8010	4.1881	3.49	2.93

and the adaptation for $\bar{\theta}_2 = \bar{\theta}_2[Cl_p]$

Table 4.2: Clp adaptation example $\bar{\theta}_2$

Iteration	Cd	Clp	$\bar{\theta}_1$	$\bar{\theta}_2$	$Cd_{Sp,rev}$	$Cd_{Sp,irr}$
0	123.93	0.73950	3.3875	5.2251	3.32	2.95
1	120.87	0.74724	2.8044	3.9498	3.44	2.44
2	120.25	0.74708	2.6995	3.6656	3.49	2.34
3	119.92	0.74756	2.7211	3.6982	3.54	2.37

It is clear that the method exposed allows to adapt the mesh for the calculation of the interest function of interest. The figures 4.4 and 4.5 shows the adapted mesh for the calculation of Cl_p and Cd respectively.

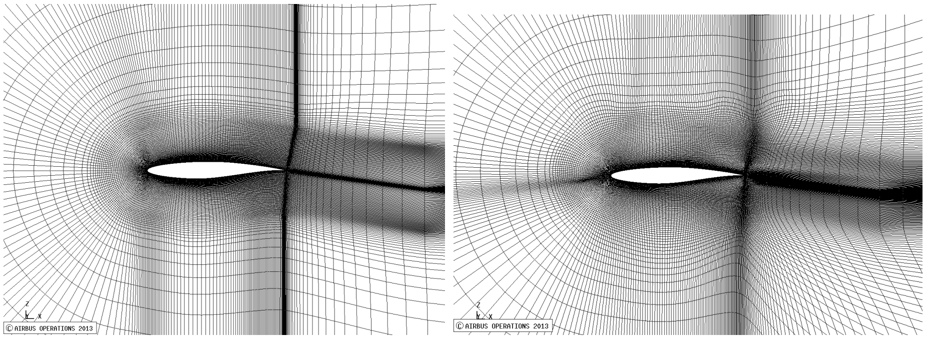


Figure 4.4: Initial mesh (Left); Adapted mesh for Cl_p (right)

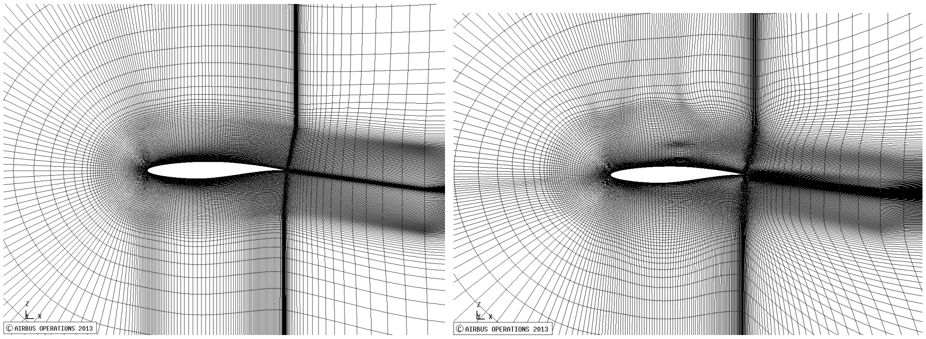


Figure 4.5: Initial mesh (Left); Adapted mesh for Cd (right)

4.4 Feature based grid adaptation

In this section we mean to summarize briefly the main characteristics and to give an example of a feature based grid adaptation. This study is made in parallel with the example given for the objective function grid adaptation illustrated in the section 4.3 in order to get a comparative between our formulation and a classic feature based adaptation one. This comparison will be made by regarding especially the accuracy of the two methods and their consistency.

4.4.1 Feature based adaptation theory

Feature based grid adaptation is a technique used to adapt, to refine or to coarse a mesh following the features present on the flow. Furthermore, by looking a flow computation on a coarse structured grid and uniform, for example on an airfoil at Mach ≈ 1 it can be observed that the shock expected is not well computed or completely dumped while we intuitively expect it to be present. This is due to the fact that within the flow field regions characterized by abrupt variables discontinuities are present. This aimed the improvement of the so called feature based adaptation methods which consist in a local refinement of the grid where shocks, vortices, trailing edges and so on.

One possible choice to detect these feature could be to moving nodes accordingly to a sensor based on the gradient of flows variables as pressure, velocity and density. This sensor would allow to detect the zones of the flow field characterized by the formation of big discontinuities and in a following step to refine them in order to avoid to dump the computation of the flow variables inside these zones of the current grid. These methods are named gradient based grid adaptation methods.

Among the different possible choices of gradient flow variables as sensor, it can be chosen the pressure gradient, the total pressure gradient, the density gradient and so on. We have chosen the Mach gradient because it allows to capture the discontinuity that influence directly our adaptation.

The criterion θ_{gb} used for this adaptation is given by the equations 4.35

$$\theta_{gb} = ||\nabla M|| dX. \tag{4.35}$$

Where ∇M is computed for all the nodes (see equation 4.36) and it has been multiplied by a scalar quantity dX (which is defined as dX used in the θ criterion in section 1.4.3) allows to take into account the possibility of every node displacement.

We take into account a volume Ω on which we integrate the Mach gradient:

$$\int_{\Omega} \nabla M d\Omega = \int_{\partial\Omega} M \overrightarrow{n} d\Gamma. \tag{4.36}$$

And we execute this operation on every cells where M_i is the Mach on the center of every cells and $M_{i,j}$ is the value interpolated on the cell contours with normal $\overrightarrow{n}_{i,j}$ and surface Ω_i.

$$\Omega_i \nabla M_i = \sum_j M_{i,j} \overrightarrow{n}_{i,j}. \tag{4.37}$$

High values of θ_{gb} will underline nodes where the Mach gradient multiplied by the half distance of the closest nodes is high so that the mesh will be moved in order to avoid it.

Additional details about the implementation of this method will be given in the following section.

4.4.2 Feature based adaptation example

We performed an adaptation on the same initial grid as for the θ-sensor method. The model used, the scheme (upwind) and their parameters are the same as for the θ-method. The step adaptation also uses the same method (elliptic grid generation) where the weight function this time are those which have been introduced in the section 4.4.1 so that the only difference consists in the definition of the "sensor" used for the remeshing. The remeshing step is performed by displacing the mesh nodes accordingly to θ_{gb} The gradient used in this section is the Mach gradient. We compute his module on every mesh point (Figure 4.6):

$$||\nabla M||$$

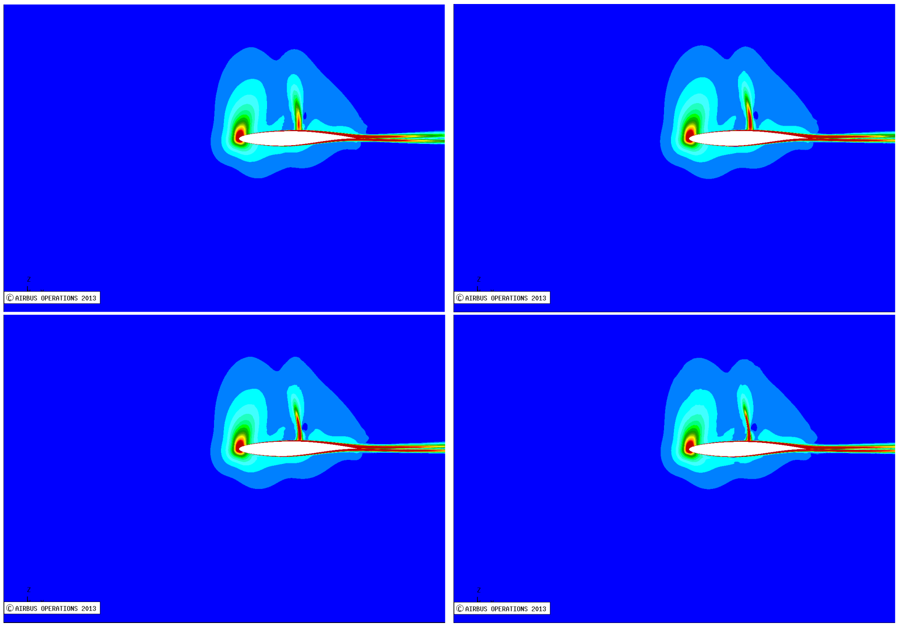


Figure 4.6: (Top Left) initial Mach gradient module; (Top Right) second iteration Mach gradient module; (Bottom Left) third iteration Mach gradient module; (Bottom right) fourth iteration Mach gradient module

It can be observed that the method allows a better definition of the shock and the vortex on the trailing edge which are partially dumped on the first flow evaluation. As we already explained the criterion built form ∇M multiplies this quantity by dX. We expect that the zones underlined will tend to be extended (with respect to ∇M) at zones where the nodes are more spaced, while they will tend to be reduced where the nodes are already very close. The criterion built is plotted in Figure 4.7 with confirmation of this fact. Indeed we notice that a bigger upstream of nodes is involved in the definition of the current criterion when is weighted by dX.

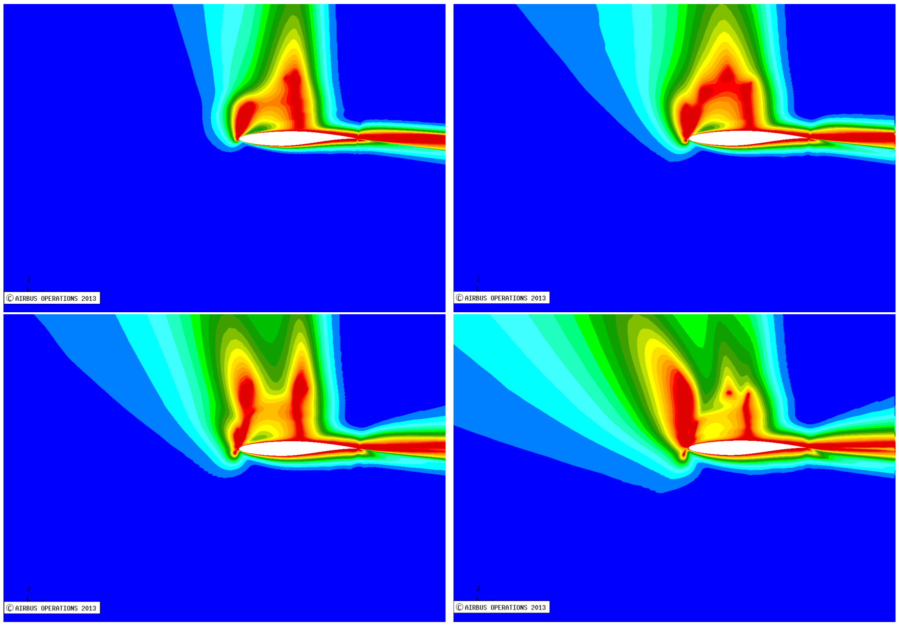


Figure 4.7: (Top Left) θ_{gb} on the initial mesh; (Top Right) θ_{gb} on the second iteration mesh; (Bottom Left) θ_{gb} on the third mesh; (Bottom right) θ_{gb} on the fourth iteration mesh

This criterion induce the nodes displacement in order to adapt the mesh within the iteration as it is exposed in Figure 4.8.

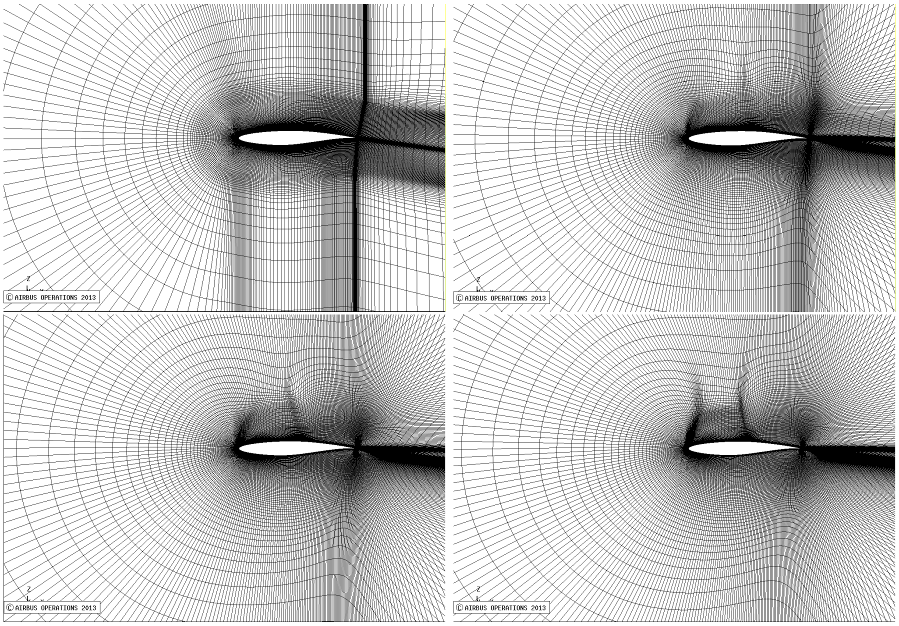


Figure 4.8: (Top Left) initial mesh; (Top Right) second mesh; (Bottom Left) third iteration mesh; (Bottom right) fourth iteration mesh

The mesh displacement is clearly made in towards the shock the vortices and the leading edge with the result of a better resolution of the flow features characterized by important gradients.

This is not directly linked with a good estimation of the functional variables as can be Cl_p or Cd_p. The results will be exposed in the following table in order to be compared with the results given through the θ method. We notice that the term W represent the global criterion quality with respect to θ_{gb} and it is calculated in order to be compared with $\bar{\theta}$.

Table 4.3: Gradient Based Adaptation Results

Iteration	Cd	Cl_p	W
0	123.69	0.73993	2.24255
1	120.268	0.743502	2.13495
2	118.856	0.741779	2.30950
3	117.953	0.738592	2.38660

4.4.3 Feature based (θ_{gb}) Vs Output based (θ)

The most important evidence rising from the comparison between the methods is that there is not a monotone improvement of the estimation function on the feature based grid adaptation. This fact is natural as the adaptation induced by the presence of the features as strong gradients is made to lead to a better resolution of those features. We take the classical example of the shock. We imagine that we start with a uniform grid in which a CFD computation has been performed. The evaluation of the gradients (of pressure for instance) would lead to a refinement around the shock wave. This would imply a refinement of other zones with respect to the initial grid as the upstream zone for which the errors increase. The increase of the errors on the computation of the upstream zone would be carried away by advection to the down flow, until the shock. Beside a better resolution of the shock the result rising from the incertitude of the results upstream would certainly lead to a better convergence of the shock feature but in a false position. Now, the feature based adaptation would certainly induce to a better definition of the shock but in many case this will have as consequences a worst estimation of other quantities as his position that are more important for the correct estimation of the functional of interest F.

Certainly our method, being goal-oriented based would not occur into this error, but as much as the discretized equations which stands behind $R = 0$ are exact (without approximations) our method would adapt the mesh establishing the relative importance of the different features (for example establishing the relative wight between gradients resolution and upstream resolution) in order to achieve a good calculation of the objective function F.

4.5 Conclusions

We have been able to show that the method just built leads towards an improvement of the accuracy on the computation of the function of interest F using the sensor $\theta_{i,j}$ The study has been performed on a 2D test case using RANS equations and the particular points redistribution is naturally in accordance with this particular test case and model used. The adaptation loop improved the estimation of both Cd and Cl_p. We highlight that there are still present some limits whose resolution would generate an additional improvement of the accuracy. These limits are both geometricals and physicals:
-The geometric limit consists in the way the singularities on the body are treated (as the trailing edge of this profile). For those points no actual displacement is allowed, so that they constitute a node redistribution limit.
-A physical limit consist in the way the partial derivative $\partial R/\partial W$ is computed. In fact it has been neglected the sensitivity of the turbulence with respect to the mesh displacement (because of hypothesis: ν_T= const) , and the effects generated by the inter-components of the Laplacian of the velocity (as a thin boundary layer hp has been done). This fact makes up a lost in the accuracy of dF/dX.
Furthermore we noticed that the grid generation and adaptation by Laplacian equation generates the most smoothed possible grid. This is important to ensure the regularity but it can reduce sometimes the effects of the adaptation. In spite of these limits, the numerical results and the comparison with a gradient-based method show that the local sensor allows a grid adaptation improving the accuracy of the function chosen and the global sensor is well correlated to the effective quality of the grid for the computation of the function chosen. A comparison with a feature based adaptation strategy showed the benefits of our method in terms of accuracy.

Chapter 5

3D grid local and global criteria

5.1 Introduction

With this chapter we mean to extend our research work to a 3D industrial case. For our test we used in particular the wing-fuselage configuration XRF1. This chapter is necessary as in development phase the extension itself is not straightforward because there are many issues which are not taken into account within a 2D case and the physic of the problem is more complicated so we devoted this chapter in order to ensure the validation of the theoretical method exposed in chapter 2. The present study will be divided into different steps as we did for the two-dimensional case as we did for in the chapter 4.

- The first step will consist into an introduction of the test case with his related direct flow computation in the paragraph 5.2

- The second step will be devoted to the study of the qualification criteria for the both goal chosen Cl_p and Cd in the paragraph 5.3

5.2 Direct and adjoint computation

This study is performed on the wing-fuselage XRF1. For the sake of clarity we call as 'Direct' computation the traditional flow-solution and with adjoint computation, the computation of the vector Λ. The parameters used for the direct computation of the flow variables are the followings: The Mach considered is $M = 0.83$ and the Reynolds number is $Re = 49.92 * 10^6$. Furthermore, the domain has been discretized with a mesh of about 21 millions point and it is shown on Figure 5.1. We take as is often done in 3D cases the half part of the geometry as it is shown in the following picture. The goals are CL_p and Cd.

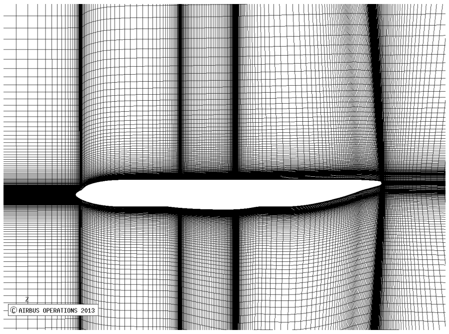


Figure 5.1: Mesh on the symmetry plane

We start with the observation that the number of points is sufficient for a good prediction of the entire flow features of this conditions. The conservative variables computed by the flow solver are combined in order to show up the behaviour variables in Figure 5.2. These variables are important for the following considerations as a validation needs at least a good comprehension of the physic.

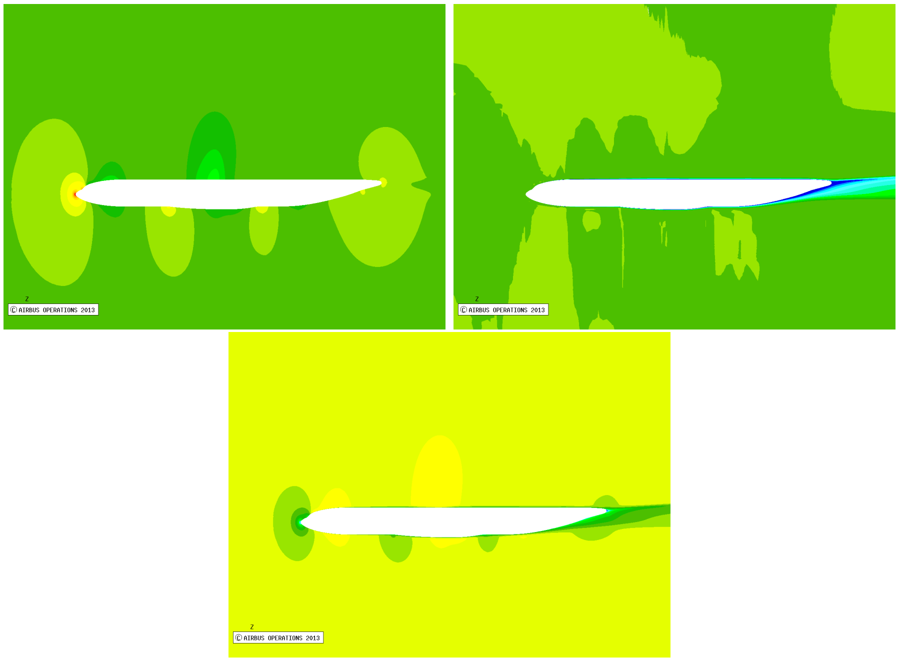


Figure 5.2: pressure distribution, total pressure distribution, Mach distribution on $O_{z,y}$

It can be noticed that the behaviour of the criterion is to highlight the zone where the gradients are more important and that for every function of interest chosen. It is also noticed that the upstream is detected as a sensible zone on the evaluation of the function for all the functions while the wake seems to be neglected. The importance of the upstream is expected as the sensor is able to 'understand' that the error on the upstream are transported by advection towards the configuration XRF1. We have to remember that the criterion is built up by multiplication of dF/dX and a characteristic length. This fact explains why the wake seems to be neglected, in fact we have an high refinement on the zone corresponding to the wake which lead to a less important detection. This also explain why the zone upstream has a smaller importance. A complete presentation of these facts can be found in the following section.

5.3 Behaviour of the criterion θ

In order to understand the behaviour of the sensor θ it is necessary to look at it with respect to the mesh. On the mesh, are present zones with an high refinement. In these zones the criterion θ has not important values because of the very small length $r_{i,j}$

We will focus on the calculation of the objective functions: Clp , Cd and for this last one we will also study his decomposition into Cd_p and Cd_f, so that we define for this particular test case: $\theta_1 = \theta_1(Clp)$ $\theta_2 = \theta_2(Cd_p)$ $\theta_3 = \theta_3(Cd_f)$ $\theta_4 = \theta_4(Cd) = \theta_2 + \theta_3$

We remark that in the case we want to analyse the behaviour of $\bar{\theta}$ it is not true that $\bar{\theta_4} = \bar{\theta_2} + \bar{\theta_3}$, because the operator average that we defined is not linear.

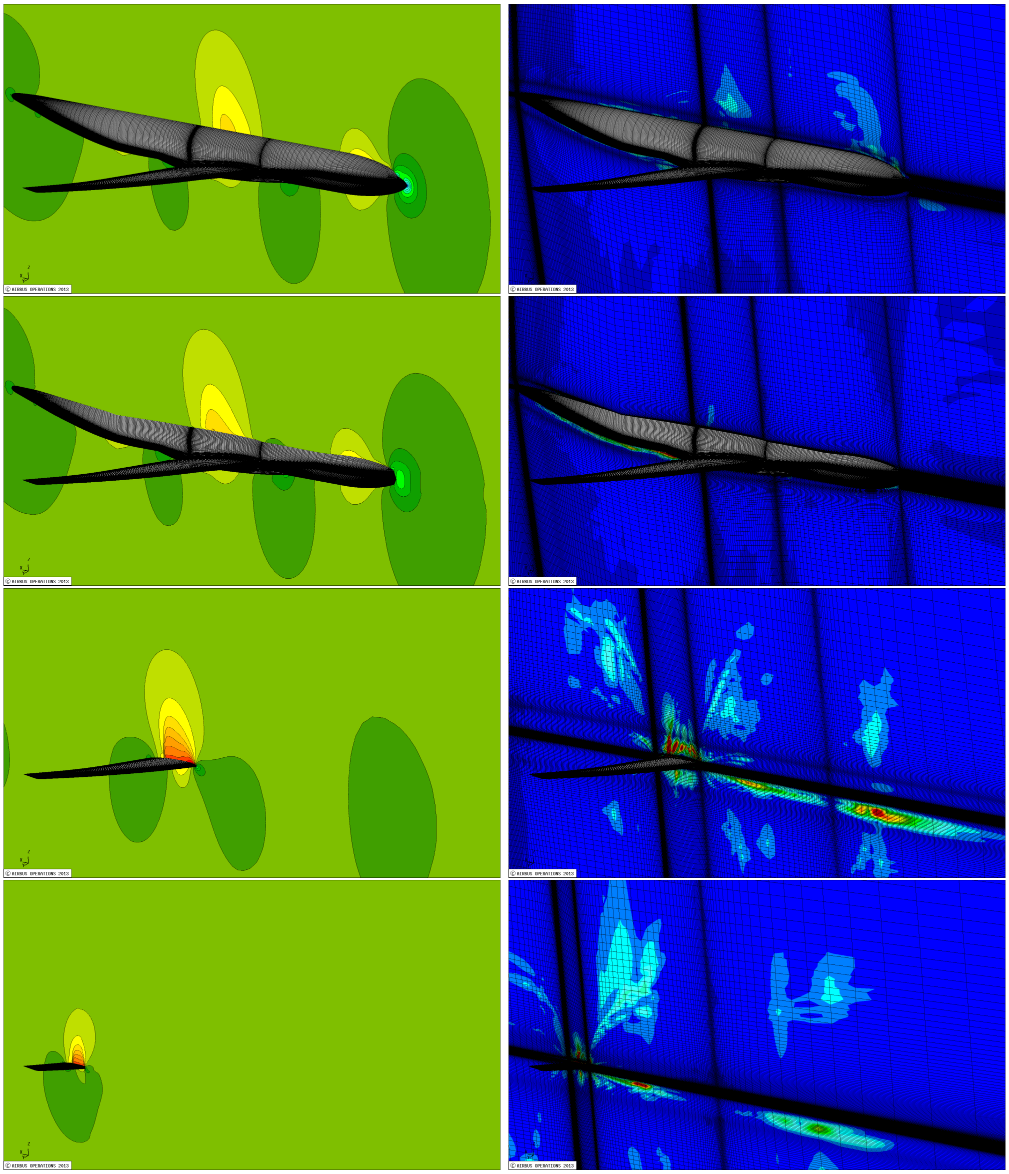


Figure 5.3: Cp (Left); θ_1 (Right)

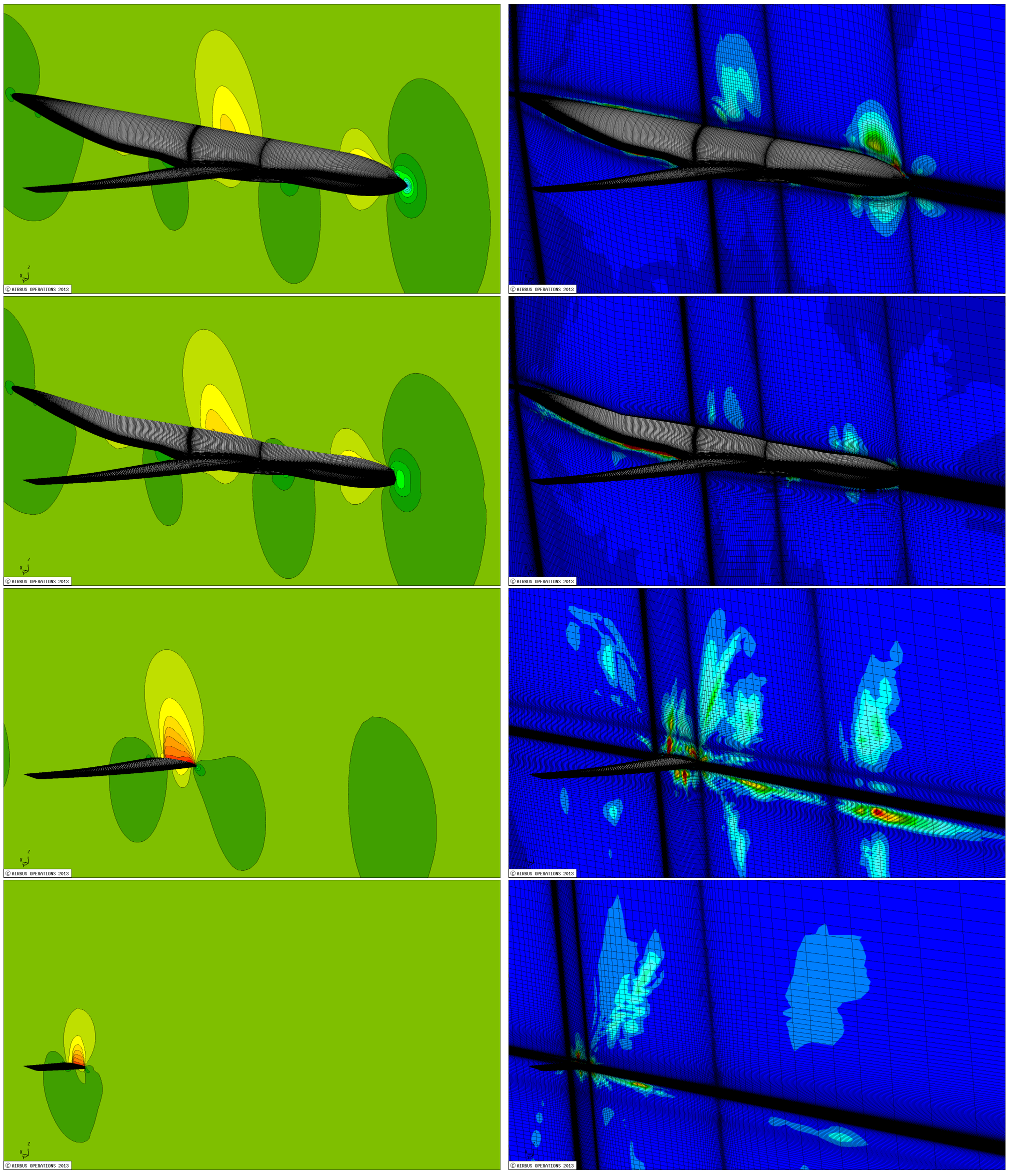


Figure 5.4: Cp (Left); θ_4 (Right)

5.4 Conclusions

After the mathematical construction of the different steps involved in the adaptation process, a study of the different steps has been performed. The strategy used to adapt the grids for the calculation of the function has been proved in the previous chapters (for a test case 2D) to be efficient and beside the approximations made, the local sensors has allowed the individuation of the zones with an high sensitivity with respect to the functions chosen. The grid adaptation induced has allowed the improvement on the function estimation accuracy and the global sensor built has been shown to indicate when a mesh is more or less adapted for the calculation of the functional chosen. With this chapter an extension to an industrial case 3D of the sensor allowed its the validation for an industrial case in which the computational cost is more important, the physic is more complex and the geometry (i.e multi blocks and non coincident grids) is more sophisticated.

Conclusions

The goal of this study was to build a strategy in order to answer to the need of accuracy on the evaluation of aerodynamic performances (i.e lift and drag) which are closely connected with the fuel consumption. Furthermore it was required to build a strategy which allows this improvement without to increase the computational cost of the CFD solution, leading to a redistribution of points rather than an addition. The construction of such a method has been performed starting by theoretical considerations on the stability of a mesh which are described at the beginning of the second chapter. In the following of the chapter we described step by step the theoretical aspects leading to the definition of a sensor $\theta_{i,j}$ used to induce the adaptation, and another sensor or criterion θ used to qualify the mesh. The study and the coherence of these sensors w.r.t our purposes is presented within the chapter three. In this chapter it was concluded that the sensor and the criterion are able to be used for our scopes. Then a mesh adaptation process was exhibited within the fourth chapter in order to achieve an improvement of the accuracy. Here we chose Cd and Cl_p as goal functions and we compared the estimation of these variables for adapted meshes with respect to limiting values obtained from a calculation of Cd and Cl_p on a very fine mesh and we proved the increasing of the goals estimation accuracy. In chapter fourth we also performed a comparison between our method and the gradient based one, showing the benefits of our strategy. Then an extension in a 3D industrial case of the is performed showing his reliability in complex cases. This extension has been showed up in chapter five where we also retained to point out from a practical point of view that the sensor is enough mature to be used for an industrial application (i.e A320neo A380 and A350).

Beside we manage with some limits, the present work showed that the total derivative of an aerodynamic function with respect to the volume mesh can be successfully used to build a strategy to adapt the mesh. Furthermore the strategy built from dF/dX presents other additional aspect of improvement. The $\nu_T = const$ and the thin boundary layer hypothesis made within the linearization of the term $\partial R/\partial W$ cause a deterioration of the accuracy of dF/dX. The limit constituted by these hypothesis can be removed in the future in order to enhance an improvement of $\theta_{i,j}$. The geometric limit constituted by the fixed points can be eliminated in order to obtain a points redistribution totally coherent with the zones individuated by $\theta_{i,j}$.

Appendix A

Construction of a mesh generation elliptic system of PDEs

The generation of a structured mesh is following. We define a computational space C as a unit square with coordinates $\boldsymbol{\xi}$, a parameter space as unite square P with coordinates $\boldsymbol{s}$ and the physical domain D with physic coordinates $\boldsymbol{x}$.

The mesh used for CFD is here defined as a map which associates every element of the computational space to the elements of the physical domain. Furthermore we define it as:

$$\boldsymbol{x}\colon\, C \rightarrow D,$$

such that every single node on the physical space is the map of a node in the computational domain. The construction of the map $C \rightarrow D$ can be written as the combination of the maps $C \rightarrow P$ and $P \rightarrow D$.

In practice $C \rightarrow P$ and $D \rightarrow P$ are first computed. In fact this last one is imposed as harmonic function:

$$\Delta \boldsymbol{s} = \sum_{k=1}^{3} \frac{\partial^2 \boldsymbol{s}}{\partial x_k^2} = 0. \tag{A.1}$$

After the setting of the map $D \rightarrow P$, and after the observation that the solution of these equations depend on the boundary conditions, the only variable is the choice of the map $C \rightarrow P$.

We define now certain quantities which will be useful in the following:

$\boldsymbol{g_i} = \frac{\partial \boldsymbol{x}}{\partial \xi_i}$	covariant base vectors
$\boldsymbol{g^i} = \frac{\partial \xi}{\partial \boldsymbol{x_i}}$	contravariant base vectors
$A_{ij} = g_{ij} = (\boldsymbol{g_i}, \boldsymbol{g_j})$	covariant metric tensor
$B^{ij} = g^{ij} = (\boldsymbol{g^i}, \boldsymbol{g^j})$	cotravariant metric tensor

Where the covariant and contra-variant metric tensor are linked by:

$$B = A^{-1}$$

Observing that the vector $\boldsymbol{s}$ can be expressed in respect to both the variables $\boldsymbol{x}$ and $\boldsymbol{\xi}$ as:

$$\boldsymbol{s}(\boldsymbol{x}) = \overline{\boldsymbol{s}}(\boldsymbol{\xi}(\boldsymbol{x}))$$

his Laplacian assumes the form:

$$\Delta \boldsymbol{s} = \sum_{i,j=1}^{3} g^{ij} \overline{\boldsymbol{s}}_{\xi_i \xi_j} + \sum_{k=1}^{3} \Delta \xi_k \overline{\boldsymbol{s}}_{\xi_k}. \tag{A.2}$$

Now we impose the condition of regularity ($\Delta \boldsymbol{s} = 0$) and we observe that the last equation can be expressed as:

$$\Delta \boldsymbol{\xi} = \sum_{i,j=1}^{3} g^{ij} P_{ij}. \tag{A.3}$$

If we recall the equation in variable $\boldsymbol{s}$ and we substitute the variable $\boldsymbol{x}$ we can write it down as:

$$\Delta \boldsymbol{x} = \sum_{i,j=1}^{3} g^{ij} \overline{\boldsymbol{x}}_{\xi_i \xi_j} + \sum_{k=1}^{3} \Delta \xi_k \overline{\boldsymbol{x}}_{\xi_k}. \tag{A.4}$$

Now we observe that $\Delta \boldsymbol{x} = 0$ and we substitute the expression in $\Delta \boldsymbol{\xi}$ to obtain:

$$\sum_{i,j=1}^{3} g^{ij} x_{\xi_i \xi_j} + \sum_{k=1}^{3} \sum_{i,j=1}^{3} g^{ij} \overline{P}_{ij}^{k} x \xi_k = 0. \tag{A.5}$$

with control functions:

$$P_k = \sum_{i,j=1}^{3} \frac{g^{ij}}{g^{kk}} \overline{P}_{ij}^{k}. \tag{A.6}$$

In order to conclude we observe that this expression is written in the form which can be used to solve the problem because it directly linked with the discretization being the covariant and contra-variant variables expressible as finite differences as we wrote down in the article.

Bibliography

[1] L Cambier and JP Veuillot. Status of the elsa cfd software for flow simulation and multidisciplinary applications. *AIAA paper*, 2008.

[2] Richard P Dwight. Heuristic *a posteriori* estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *Journal of Computational Physics*, 2008.

[3] A Jameson, W Schmidt, and E Turkel. Numerical solution of the euler equations by finite volume method using runge–kutta time stepping schemes. *AIAA Paper*.

[4] R. Sauvage M. Meaux J.A. Desideri M. Nguyen-Dinh, J. Peter. *Mesh qualification and local adaptation based on total derivatives of aerodynamic functions w.r.t. mesh coordinates. Application to Eulerian flow computation*. 2012.

[5] Eric J Nielsen and Michael A Park. Using an adjoint approach to eliminate mesh sensitivities in computational design. *AIAA journal*, 2006.

[6] Jacques Peter. Introduction à l'optimisation de forme en aérodynamique et quelques exemples d'application. *Office national d'études et de recherches aerospatiales*, 2006.

[7] Jacques Peter, Maxime Nguyen-Dinh, and Pierre Trontin. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates. with applications to euler flows. *Computers & fluids*, 2012.

[8] Jacques EV Peter and Richard P Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 2010.

[9] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 1981.

[10] Bharat K Soni, Roy Koomullil, David S Thompson, and Hugh Thornburg. Solution adaptive grid strategies based on point redistribution. *Computer Methods in Applied Mechanics and Engineering*, 2000.

[11] Allmaras Spalart. A one equation turbulence model for aerodinamic flows. *AIAA journal*.

[12] Stephanus Petrus Spekreijse. Elliptic grid generation based on laplace equations and algebraic transformations. *Journal of Computational Physics*, 1995.

[13] GD Van Albada, B Van Leer, and WW Roberts Jr. *A comparative study of computational methods in cosmic gas dynamics*. 1997.

[14] Bram Van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of computational Physics*, 1979.

[15] David A Venditti and David L Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 2003.