Shape optimization in computational fluid dynamics

Quang Vinh Dinh — Gilbert Rogé — Cyrille Sevin — Bruno Stoufflet

Dassault Aviation
78, quai Marcel Dassault, 92214 Saint-Cloud

ABSTRACT. *This paper is devoted to shape optimization for Partial Differential Equations (PDE) systems related to Computational Fluid Dynamics (CFD). Numerical approximation of the PDE's relies on schemes satisfying discrete maximum principles and using unstructured meshes generated from the shape parameters. The theory of control is applied to the discrete design problem with the resulting constrained optimization problem solved by gradient based algorithms. An automatic differentiation procedure for Fortran codes is extensively used to carry out the CFD sensitivity analysis.*

RÉSUMÉ. *Cet article est consacré à l'optimisation de formes aérodynamique pour les systèmes d'Equations aux Dérivées Partielles (EDP) appliquées à la mécanique des fluides numériques. L'approximation numérique des EDP repose sur des schémas satisfaisant les principes des maximum discrets, et considérant des maillages non structurés issus des formes paramétrées. La théorie du contrôle est appliquée au problème discret, le problème d'optimisation avec contraintes qui en est issu étant résolu par des algorithmes utilisant le calcul du gradient. Un programme de différentiation automatique de codes Fortran est utilisé dans l'étape de calcul des gradients.*

KEY WORDS : *shape optimization, Computational Fluid dynamics, optimal control theory, automatic differentiation.*
MOTS-CLÉS : *optimisation de formes, mécanique des fluides numérique, théorie du contrôle optimal, différentiation automatique.*

Revue européenne des éléments finis. Volume 5 - n° 5-6/1996, pages 569 à 594

1 Introduction

Numerical optimization based on CFD tools demands specific developments to derive efficient strategies based on complex flow models. Among alternative methods, we have chosen to treat design problems using the theory of control [10],[14] of systems governed by partial differential equations. Flow equations are treated as constraints which necessitate the solution of the linear system called the adjoint equation, involving the transposed Jacobian matrix of the system of equations. These methods avoid the limitations of classical optimization algorithms where the gradient is computed by finite differences meaning that the computational effort is directly proportional to the number of design variables (optimization parameters) considered. In the approach selected here, the gradient is evaluated without additional flowfield computations ; it only requires a computational effort related to the solution of the adjoint linear system which is considerably limited compared to a flowfield evaluation. We have selected the discrete sensitivity analysis where the control theory is applied on the discrete system itself. Specific developements on Euler equations will be addressed ; Finite Volume Galerkin methods, upwind, centered. Implicit methods will use the system derived from the first-order accurate approximation or directly the second-order linearized system computed with automated differentiation tools. As the boundary conditions are included in the formulation, the matrix involved in the implicit time integration leads directly to the adjoint system. An other possible ingredient introduced by A. Dervieux [7] is the multilevel parametrization for hierarchical optimization, which is envisaged in a near future.
Some illustrations will be given on examples of optimization of 2D airfoils using unstructured meshes.

2 Optimization strategies

The general formulation of a shape optimization problem has been introduced by many authors [4], [14], [7]. It has been applied in many situations and recently by Jameson [8]. He advocates the application of the control theory directly to the system of partial differential equations. The adjoint equations are formed as a system of differential equations. These differential equations are then discretized and solved in the same manner as the original flow equations. This approach can be called as continuous sensitivity analysis. Here, a different approach, called discrete sensitivity analysis is applied consisting in applying the control theory to the discrete equations itselves. Two advantages make it attractive: one manipulates exact gradient and the adjoint operator is easily derived from the implicit matrix.

The general formulation may be written as :

Find a shape $\gamma \in \mathcal{O}$ (a family of shapes) such that $\gamma^* = \arg \min_{\gamma \in \mathcal{O}} j(\gamma)$ where $j(\gamma)$ denotes a cost function given by

$$j(\gamma) = J(\gamma, \mathbf{W}(\gamma)) = \int_\gamma C(\mathbf{W}(\gamma)) d\sigma$$

$$J : \mathcal{O} \times V^{d+2} \rightarrow IR$$

- C is a function incorporating the global aerodynamic coefficients to be optimized and some penalized constraints.
- d denotes the space dimension (2 or 3) for the flow domain $\Omega \in IR^d$ around shape γ.
- $\mathbf{W}(\gamma)$ is the state-vector (e.g. density, momentum, energy) solution of the state equation :

$$\mathbf{E}(\gamma, \mathbf{W}(\gamma)) = 0 \text{ in } \Omega$$

- V is a subspace of $L^2(\Omega)$.

The shape γ^* satisfies the optimality condition $\frac{dj(\gamma)}{d\gamma} = 0$, where the derivative of cost function $j(\gamma)$ is given by :

$$\frac{dj(\gamma)}{d\gamma}\delta\gamma = \frac{\partial J}{\partial \gamma}\delta\gamma + < \frac{\partial J}{\partial \mathbf{W}}, \frac{d\mathbf{W}}{d\gamma}\delta\gamma > .$$

The dependance of $\mathbf{W}$ and γ to the flow domain Ω is expressed by :

$$\frac{\partial \mathbf{E}}{\partial \gamma}\delta\gamma + \frac{\partial \mathbf{E}}{\partial \mathbf{W}}\frac{d\mathbf{W}}{d\gamma}\delta\gamma = 0.$$

Introducing an adjoint state (or Lagrange multiplier) Ψ satisfying the adjoint equation

$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}}^* \Psi = \frac{\partial J}{\partial \mathbf{W}},$$

the gradient of the cost function can be reformulated into the following expression :

$$\frac{dj(\gamma)}{d\gamma}\delta\gamma = \frac{\partial J}{\partial \gamma}\delta\gamma - < \Psi, \frac{\partial \mathbf{E}}{\partial \gamma}\delta\gamma >$$

We will treat the case of two-dimensional shape optimization for (multi) - airfoil profiles including the capacity of multi-point optimization. The objective is to select relevant strategies before going to 3D cases. As a first step, we will limit ourselves to inviscid flow as a model for compressible flow. In a

first study, we applied the theoretical approach to treat full potential equation in [19]. In this present paper, we will detail the construction of the optimization algorithm when using numerical approximations of the Euler equations described in the first section.

As our numerical methods are based on unstructured meshes in order to treat general geometries, a key point of the method is the relation between optimization parameters and geometrical characteristics of the mesh used in numerical approximations. This geometric modeller, which will be described in the next section, involves two mesh representations :

- a *surface* mesh representation, usually defined by a CAD software.
- a *volume* mesh representation, prescribed by an unstructured mesh generator/deformer.

Once, the whole chain of operations from shape γ to state vector $\mathbf{W}$ is obtained,we have sufficient information to apply the control theory which sets up the following system of equations :

$$\begin{cases} \mathbf{E}(\gamma, \mathbf{W}(\gamma)) = 0 & \text{(state)} \\ \dfrac{\partial \mathbf{E}}{\partial \mathbf{W}}(\gamma, \mathbf{W}(\gamma))^* \cdot \Psi(\gamma) = \dfrac{\partial J}{\partial \mathbf{W}}(\gamma, \mathbf{W}(\gamma)) & \text{(adjoint)} \\ j'(\gamma) = \dfrac{\partial J}{\partial \gamma}(\gamma, \mathbf{W}(\gamma)) - < \Psi(\gamma), \dfrac{\partial \mathbf{E}}{\partial \gamma}(\gamma, \mathbf{W}(\gamma)) > & \text{(optimality)} \end{cases} \quad (1)$$

The introduction of the adjoint problem resulted from the application of the control theory on the continuous system where the flow equations are viewed as a constraint. This analysis can be directly performed on the discretized system. This discrete sensitivity analysis is detailed hereafter.

Once the derivative of cost function j is obtained, it can be fed into a numerical optimization algorithm to obtain an improved design. A first class of problems addressed are the inverse problems aiming to control a prescribed boundary pressure distribution. The cost function can be formulated as

$$j(\gamma) = \int_\gamma (p - p^{obj})^2 d\sigma \quad \text{or in discrete form} \quad j(\gamma) = \sum_{l \in \mathcal{L}} \alpha_l \, (p_l - p_l^{obj})^2$$

where $card(\mathcal{L})$ is the number of pressure points given on the profile.

A second class of problem is related to the minimization of a general cost function. Among them, problem of minimizing the drag coefficient will be addressed in defining the cost function as the pressure drag coefficient (augmented with viscous drag if a viscous correction is considered) submitted to constraints as given target pressure distribution or given lift.

For each of these problems, we will use a conjugate gradient based [15] numerical optimizer, where constraints are incorporated into the cost function through penalty terms.

3 Discretization schemes

3.1 The CFD solver

3.1.1 General formulation

From these last years, design principles of numerical schemes for compressible flows tend to emerge in a frame shared by the whole CFD community based on schemes satisfying discrete maximum principles for scalar equations. This frame dictates the way followed by various teams to construct high-order non-oscillatory schemes.

For conservation laws, a convenient approach to design discretization schemes is based on two steps:

(1) Monotonicity principle for non-oscillatory low-order scheme
(2) High-order construction preserving property (1)

On this basis (to satisfy (2)) are constructed a large number of family of schemes (at the top, stand MUSCL type reconstruction or introduction of anti-diffusive term). Successful attemps avoiding step (2) are reported comprising Finite Element approaches based on Petrov-Galerkin formulations ([17]), distributive schemes balancing fluxes on the element ([6]) and recently mixed Finite Volume approximations considering the gradient as a degree of freedom [2]. In this paper, we rely on a classical MUSCL reconstruction or diffusive term addition.

Concerning the point (1), for a large number of years, total variation diminushing (TVD) schemes concept has dominated. This notion is difficult to extend in more than one-dimension and specially when dealing with unstructured meshes. Moreover, this is confirmed by recent theoretical investigations on the convergence of schemes on unstructured meshes ([5], [13]) where the functional space BV is not the appropriate one for weak convergence. As mentionned before, a more incisive frame is to search for a discrete maximum principle. This has motivated to construct discretizations with positive coefficients ([3], [8], [1]).

We consider the following hyperbolic system of $d + 2$ conservation laws in

the flow domain $\Omega \in I\!R^d$:

$$\mathbf{W}_t + \vec{\nabla} \cdot \vec{\mathbf{F}}(\mathbf{W}) = 0 \tag{2}$$

where

- The vector of unknowns $\mathbf{W}$ belongs to V^{d+2}, V being a subspace of $L^2(\Omega)$.
- $\mathbf{W}_t$ denotes the time-derivative of $\mathbf{W}$.
- $\vec{\mathbf{F}}(\mathbf{W}) = (\mathbf{F}_\alpha(\mathbf{W}))_{1 \leq \alpha \leq d}$ is the vector of flux functions $\mathbf{F}_\alpha \in V^{d+2}$.

Next, we assume that Ω is a polyedral bounded domain of $I\!R^d$ and let V_h be the set of continuous, piecewise linear polynomial functions defined on a standard "triangulation" of Ω. A Lagrange-Galerkin formulation of equation (2) can be written as :

$$\int_\Omega \mathbf{W}_t \, \phi \, d\Omega + \int_\Omega \phi \, \vec{\nabla} \cdot \vec{\mathbf{F}}(\mathbf{W}) \, d\Omega = 0 \quad , \quad \forall \phi \in V_h \tag{3}$$

For the space V_h, we define a set of basis functions $\{\phi_i\}_{1 \leq i \leq N}$, where N is the number of vertices of the mesh and ϕ_i the basis function associated to vertex i, with local compact support $supp(i)$. In that case, if $\vec{\mathbf{F}}(\mathbf{W}) = \sum_{j=1}^{N} \vec{\mathbf{F}}_j(\mathbf{W}) \phi_j$, and after some straight-forward integration by parts, equations (3) can be recast into :

$$\forall i \quad , \quad 1 \leq i \leq N$$

$$\int_\Omega \phi_i \, \mathbf{W}_t \, d\Omega - \sum_{j=1}^{N} \int_\Omega \phi_j \vec{\nabla} \phi_i \cdot \vec{\mathbf{F}}_j(\mathbf{W}) \, d\Omega + \int_\Gamma \phi_i \vec{\mathbf{F}}(\mathbf{W}) \cdot \vec{\nu} \, d\sigma = 0 \tag{4}$$

where $\Gamma = \partial\Omega$ and $\vec{\nu}$ is the outward unit normal to Γ.

Next, we construct a dual mesh consisting of cells C_i (associated to vertex i) which form a partition of domain Ω. The following vector is introduced for the edge between vertices i and j :

$$\vec{\eta}_{ij} = \int_{supp(i) \bigcap supp(j)} (\phi_i \vec{\nabla} \phi_j - \phi_j \vec{\nabla} \phi_i) d\Omega = \int_{\partial C_i \cap \partial C_j} \vec{\eta} \, d\sigma$$

where $\vec{\eta}$ is the normal to *surface* $\partial C_i \cap \partial C_j$, the direction of which is defined to be from vertex i to vertex j. The next identity holds for a close contour around vertex i :

$$\sum_{j \in K(i)} \vec{\eta}_{ij} . \vec{\mathbf{F}}_i(\mathbf{W}) + \int_{\partial C_i} \phi_i \vec{\mathbf{F}}_i(\mathbf{W}) \cdot \vec{\nu} \, d\sigma = 0 \tag{5}$$

where $K(i)$ is the set of neighbouring vertices to vertex i. With these definitions, a centered approximation of the Finite Volume Galerkin formulation for (4) would be :

$$\int_\Omega \phi_i \, \mathbf{W}_t \, d\Omega + \sum_{j \in K(i)} \int_{\Omega_h} \vec{\eta}_{ij} \cdot \frac{\vec{\mathbf{F}}_i(\mathbf{W}) + \vec{\mathbf{F}}_j(\mathbf{W})}{2} + \int_{\partial C_i \cap \Gamma} \phi_i \frac{\vec{\mathbf{F}}_i(\bar{\mathbf{W}}) + \vec{\mathbf{F}}(\bar{\mathbf{W}})}{2} \cdot \vec{\nu} \, d\sigma = 0 \tag{6}$$

where $\bar{\mathbf{W}}$ is a boundary state vector, which is a function of $\mathbf{W}_i$ and some prescribed boundary values.

Finally, introducing a numerical flux function $\Phi_{\mathbf{F}}$ and using mass-lumping for the identity operator - which is a valid approximation for steady-state calculations - the previous equation reduces to :

$$S_i \frac{d\mathbf{W}_i}{dt} + \sum_{j \in K(i)} \Phi_{\mathbf{F}}(\mathbf{W}_i, \mathbf{W}_j, \vec{\eta}_{ij}) + \int_{\partial C_i \cap \Gamma} \phi_i \frac{\vec{\mathbf{F}}_i(\bar{\mathbf{W}}) + \vec{\mathbf{F}}(\bar{\mathbf{W}})}{2} \cdot \vec{\nu} \, d\sigma = 0 \tag{7}$$

where $S_i = \int_{C_i} d\Omega$.

In the sequel, we will be solely interested in the Euler equations of gas dynamics.

As an example, for the Euler equations in two dimensions, i.e. $d = 2$:

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix} \quad \mathbf{F}_1(\mathbf{W}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{pmatrix} \quad \mathbf{F}_2(\mathbf{W}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{pmatrix}$$

where ρ is the density, (u,v) is the velocity-vector, e is the total specific energy and p is the pressure, with the following equation of state for a perfect gas :

$p = p(\rho, e)$.

3.1.2 Upwind scheme

A first example of a numerical flux $\Phi_{\mathbf{F}}$ is that given by Van Leer [18]. This approximation gives good results for high Mach number and is hereafter described in a one-dimensional framework :

$$F = F_{vl}^{+} + F_{vl}^{-} \tag{8}$$

Assuming $f_1^+ = \frac{1}{4}\rho c(\frac{u}{c}+1)^2$, the numerical flux is given by :

$$F_{vl}^+(\mathbf{W}) = \begin{cases} F & \text{if } u \geq c, \\ \begin{pmatrix} f_1^+ \\ \frac{f_1^+}{\gamma}((\gamma-1)u+2c) \\ \frac{f_1^+}{2(\gamma^2-1)}((\gamma-1)u+2c)^2 \end{pmatrix} & \text{if } -c < u < c, \\ 0 & \text{if } u \leq -c \end{cases} \tag{9}$$

Here, u is velocity, ρ is density, c is local speed of sound, and γ is the ratio of heat coefficients.

For $d > 1$, one just have to define u, c, ρ along edges. We are using this scheme to derive easy-to-use first-order Jacobian for our implicit scheme (see section 3.1.4).

3.1.3 Centered scheme

A somewhat more elaborated scheme is constructed adding an artificial dissipation term modulated by an adequate sensor to derive a monotone scheme (see Jameson in [9]). The level of artificial dissipation in smooth region is low while it is increased to a value in the neighborhood of a shock wave to recover a first-order scheme. Following Peraire and al [12], let us construct a fourth-order derivative of the solution, evaluated in the following manner, along the edge going from vertex N_i to vertex N_j :

$$D_{ij}^{(4)}\tilde{\mathbf{W}} = \tilde{\mathbf{W}}_j - \tilde{\mathbf{W}}_i - \nabla\tilde{\mathbf{W}}_{ij} \cdot \overrightarrow{N_iN_j} \tag{10}$$

where $\tilde{\mathbf{W}}$ is taken equal to $\mathbf{W}$ with the specific enthalpy replacing energy in the last equation. The gradient is constructed as follows:

$$\nabla\tilde{\mathbf{W}}_{ij} \cdot \overrightarrow{N_iN_j} = \frac{1}{2}(\nabla\tilde{\mathbf{W}}_i \cdot \overrightarrow{N_iN_j} + \nabla\tilde{\mathbf{W}}_j \cdot \overrightarrow{N_iN_j}) \tag{11}$$

Then the numerical flux is written as the sum of the centered scheme with a sensor-driven dissipation made of a second order term active for shock capturing and the fourth order term active in smooth regions

$$\Phi_{\mathbf{F}}(\mathbf{W}_i, \mathbf{W}_j, \vec{\eta}_{ij}) = \vec{\eta}_{ij} \cdot \frac{\vec{\mathbf{F}}_i(\mathbf{W}) + \vec{\mathbf{F}}_j(\mathbf{W})}{2} - |\lambda_{ij}| \left(\epsilon_{ij}^{(2)}(\tilde{\mathbf{W}}_i - \tilde{\mathbf{W}}_j) + \epsilon_{ij}^{(4)} D_{ij}^{(4)}\tilde{\mathbf{W}}\right) \tag{12}$$

The scalar viscosity considered here is governed by the maximum eigenvalue $|\lambda_{ij}|$ which is computed as

$$\lambda_{ij} = \frac{1}{2} \left(\vec{u}_i \cdot \vec{\eta}_{ij} + c_i \|\vec{\eta}_{ij}\| + \vec{u}_j \cdot \vec{\eta}_{ij} + c_j \|\vec{\eta}_{ij}\| \right) \tag{13}$$

We introduce a pressure sensor, based on second derivatives of pressure variable p.

$$ps_{ij} = \min(ps_i, ps_j) \quad \text{with} \quad ps_i = \frac{|p_j - p_i - \nabla p_i \cdot \overrightarrow{N_i N_j}|}{|p_j - p_i + \nabla p_j \cdot \overrightarrow{N_i N_j}|} \tag{14}$$

Then the sensors can be defined as

$$\epsilon^{(2)} = \min(1, 2\, ps_{ij}) \quad \text{and} \quad \epsilon^{(4)} = -2 \max(0, k_4 - \epsilon^{(2)}) \tag{15}$$

where k_4 is some threshold value.

3.1.4 Solution strategy

The discretized state equations - solving the two-dimensional Euler equations - are given in formula (7) described above and can be summed up as follows :

$$S_i \frac{d\mathbf{W}_i}{dt} + \mathbf{E}_i = 0 \tag{16}$$

where $\mathbf{E}_i$ is the steady-state residual vector given by :

$$\mathbf{E}_i = \sum_{j \in K(i)} \Phi_F(\mathbf{W}_i, \mathbf{W}_j, \vec{\eta}_{ij}) + \int_{\partial C_i \cap \Gamma} \Phi_i \frac{\vec{\mathbf{F}}_i(\bar{\mathbf{W}}) + \vec{\mathbf{F}}(\bar{\mathbf{W}})}{2} \cdot \vec{\nu}\, d\sigma \tag{17}$$

A linearized implicit solution of this system usually involves a representation of the Jacobian $\mathbf{A}_{ij} = \frac{\partial \mathbf{E}_i}{\partial \mathbf{W}_j}$ through the following linear system :

$$\frac{S_i}{\Delta t} \delta\mathbf{W}_i + \sum_{j=1}^{N} \mathbf{A}_{ij}\, \delta\mathbf{W}_j = -\mathbf{E}_i$$

where $\delta\mathbf{W}$ is the time increment for state-vector $\mathbf{W}$.

The order of accuracy of the aerodynamic analysis is determined from the evaluation of the residual vector $\mathbf{E}_i$, which can be first or second-order in space. However, using first-order Jacobian and Block-Jacobi relaxation for the linear system, is sufficient to converge the implicit scheme. In that case, only block-diagonal matrices need to be stored, saving considerable memory, especially for 3D cases. To recover quadratic convergence, one can consider second-order Jacobian, obtained from automatic differentiation tools (see section 3.3.3).

3.2 The geometric modeller

As pointed out before, a key point of our method is the relation between the optimization parameters, subsequently defined to be $z = \{z_m\}_{1 \leq m \leq M}$, and the geometrical characteristics of the numerical meshes.

In a standard CFD analysis, the user will start by specifying some values for z (e.g. airfoil length, thickness, twist, camber etc...). These values will be fed into a CAD software and will produce a *surface* mesh. Finally, a *volume* mesh is built on top of this *surface* mesh with the help of an unstructured mesh generator.

Thus, the relation between z and the final numerical meshes can be decomposed into elementary functions ; the overall gradient will be the product of these elementary gradients for each of which an appropriate evaluation can be used. For example, one can anticipate to use finite differences restricted to some specific elementary operations to calculate the necessary information. In the present work, this manner has been avoided by using analytical dependance between intervening variables. The strategy chosen here can be easily described.

For the *surface* mesh, an initial collection of boundary points - i.e. a *surface* topology and a initial position $(x_i^{(b)}, y_i^{(b)})$ for these points - is defined by the CAD software from an initial set of values for z . Subsequent perturbations arising from variations of z are handled - with the *surface* topology fixed - by automatic differentiation via the ODYSSEE tool (cf INRIA [11]), of the relevant subroutines in the CAD software. In the results presented hereafter, the optimization parameters z chosen are explicit functions of the ordinates v_i of the control points (of coordinates (u_i, v_i)) of the cubic splines defining the boundary and the angle of attack α of the profile.

For the *volume* mesh, an initial grid is constructed using an unstructured mesh generator starting from the prescribed initial *surface* mesh. Subsequent perturbed meshes in the optimization process are determined from variations of the position of these boundary mesh points by solving an elasticity equation which will provide a new position for the interior mesh points (of coordinates $(x_i^{(m)}, y_i^{(m)})$). As the topology of the *volume* mesh is preserved by this approach, the relation between boundary and interior mesh points is differentiable and the corresponding gradient is easily computed.

Thus, by working with fixed topologies for the *surface* and *volume* meshes, the relation between the optimizattion parameters and the numerical meshes is totally differentiable. However, it should be noted that our mesh adaption capability is somewhat limited : only mesh movement is permitted and mesh refinement is not allowed.

3.3 Sensitivity analysis

3.3.1 Gradient of cost function

As stated above, let us now apply control theory directly on the discretized system with optimization parameters $\{z_m\}_{1\leq m\leq M}$. To fix ideas, let's take a cost function of the form

$$j(z) \;=\; J(z,\mathbf{W}) \;=\; \frac{1}{2}\sum_{l\in\mathcal{L}} \alpha_l\ (p_l - p_l^{obj})^2$$

designed for an inverse problem with a given target pressure distribution $\{p_l^{obj}\}_{l\in\mathcal{L}}$. Of course, apart from this particular definition of j, what follows is general and is not restricted to a specific problem. The partial derivatives of $J(z,\mathbf{W})$ are $\dfrac{\partial J}{\partial z}$ and

$$\frac{\partial J}{\partial \mathbf{W}_k} = \sum_{l\in\mathcal{L}} \alpha_l\ (p_l - p_l^{obj})\ \frac{\partial p_l}{\partial \mathbf{W}_k}\quad ,1\leq k\leq N$$

The state equation is now :

$$\mathbf{E}_i(z,\mathbf{W}) \;=\; 0\ \ ,\ \ 1\leq i\leq N \tag{18}$$

where $\mathbf{E}_i$ is the steady-state residual defined in (17).

Its derivative with respect to parameters z is given by

$$\frac{\partial \mathbf{E}_i}{\partial z_m} + \sum_{j=1}^{N} \mathbf{A}_{ij}\frac{\partial \mathbf{W}_j}{\partial z_m} = 0\ \ ,\ \ 1\leq i\leq N\ \ ,\ \ 1\leq m\leq M$$

Some notations are required. We denote $<<.,.>>$ the duality product in IR^{d+2}. For any operator B of $\mathcal{L}(IR^{d+2},IR^{d+2})$, we define its adjoint operator to be B^T such that :

$$<<B^T\ U,V>>=<<U,B\ V>>\ \ ,\ \ \forall\ U\in IR^{d+2}\ \ ,\ \ \forall\ V\in IR^{d+2}.$$

Introducing an adjoint state $\mathbf{\Psi}$ solution of the following system :

$$\sum_{j=1}^{N} \mathbf{A}_{ji}^T\ \mathbf{\Psi}_j = \frac{\partial J}{\partial \mathbf{W}_i}\ \ ,\ \ 1\leq i\leq N \tag{19}$$

we get the following for the derivative of $j(z)$:

for $1 \leq m \leq M$

$$\begin{aligned}
\frac{dj}{dz_m} &= \frac{\partial J}{\partial z_m} + \sum_{k=1}^{N} << \frac{\partial J}{\partial \mathbf{W}_k} , \frac{\partial \mathbf{W}_k}{\partial z_m} >> \\
&= \frac{\partial J}{\partial z_m} + \sum_{k=1}^{N} << \sum_{j=1}^{N} \mathbf{A}_{jk}^T \mathbf{\Psi}_j , \frac{\partial \mathbf{W}_k}{\partial z_m} >> \\
&= \frac{\partial J}{\partial z_m} + \sum_{j=1}^{N} << \mathbf{\Psi}_j , \sum_{k=1}^{N} \mathbf{A}_{jk} \frac{\partial \mathbf{W}_k}{\partial z_m} >> \\
&= \frac{\partial J}{\partial z_m} - \sum_{j=1}^{N} << \mathbf{\Psi}_j , \frac{\partial \mathbf{E}_j}{\partial z_m} >>
\end{aligned}$$

3.3.2 Solution strategy

The computation of the gradient of the cost function requires the solution of the non-linear state equation (18) and the linear adjoint equation (19). Different strategies have been adopted which are taylored to the requirements of each problem.

- solution of the state equation.

Adopting the notations :

$\mathbf{A} = [\mathbf{A}_{ij}]_{1 \leq i,j \leq N}$ with $\mathbf{A}_{ij} = \frac{\partial \mathbf{E}_i}{\partial \mathbf{W}_j}$, $\mathbf{D} = diag(S_i)_{1 \leq i \leq N}$

we recall the results of section 3.1.4 and rewrite equation (18) as :

$$\left[\frac{\mathbf{D}}{\Delta t} + \mathbf{A} \right] \delta \mathbf{W} = - \mathbf{E}(\mathbf{W}) \tag{20}$$

which is a linearized implicit Newton-like iterative procedure to solve the non-linear state equation(18). Since we are looking for steady-state solutions, time accuracy is not mandatory and, as noted in section 3.1.4, using an approximate operator $\tilde{\mathbf{A}}$ and block-Jacobi relaxation for the linear system, is sufficient to converge the Newton procedure.

For $\tilde{\mathbf{A}}$, we have used the following approximations :

$$\mathbf{A}_{ij} \approx \tilde{\mathbf{A}}_{ij} = \frac{\partial \mathbf{E}_i^{VL1}}{\partial \mathbf{W}_j} \tag{21}$$

where $\mathbf{E}_i^{VL1}$ is the steady-state residual with a first-order upwind numerical flux from Van-Leer (see section 3.1.2). The block matrices $\{\tilde{\mathbf{A}}_{ij}\}_{1 \leq i,j \leq N}$ are stored in a block-sparse matrix fashion to be used hereafter.

- solution of the adjoint equation.

We recall the adjoint equation (19) with the above notations :

$$\mathbf{A}^T \, \mathbf{\Psi} \; = \; \frac{\partial J}{\partial \mathbf{W}} \tag{22}$$

An accurate evaluation of the gradient $\frac{dj}{dz}$ requires the "exact" Jacobian $\mathbf{A}_{ij} \; = \; \frac{\partial \mathbf{E}_i}{\partial \mathbf{W}_j}$, which, for a second-order approximation involving a larger numerical stencil, can be time-consuming to construct and will need a large storage space.

A first simple approach is to use, as for the state equation, the approximate operator $\tilde{\mathbf{A}}^T$ for the adjoint equation. This leads to an approximate gradient $\frac{\tilde{dj}}{dz}$ which is fairly easy to construct.

For more complex cases where accuracy is a must, we will use $\tilde{\mathbf{A}}^T$ as a preconditionner in an iterative procedure to compute $\mathbf{\Psi}$:

$$\text{for } k \; = \; 0 \text{ until convergence} \left\{ \begin{array}{lcl} \tilde{\mathbf{A}}^T \Delta \mathbf{\Psi} & = & \frac{\partial J}{\partial \mathbf{W}} \; - \; \mathbf{A}^T \mathbf{\Psi}_k \\ \mathbf{\Psi}_{k+1} & = & \mathbf{\Psi}_k \; - \; \Delta \mathbf{\Psi} \end{array} \right. \tag{23}$$

Such a procedure is interesting because :

- A more diagonally dominant matrix may be used to drive the solution of the linear systems (as opposed to the sometimes ill-conditionned high-order Jacobian).
- The higher-order Jacobian resides on the right-hand side and may be dealt with in an explicit manner.
- We don't really need the high-order approximation of the Jacobian, but only its product with a vector, which has a really lower cost.

It can be noticed that building all of the required Jacobian and derivatives by hand, should be very complex. So, to get the second-orders corrective terms, we used automatic differentiation. The software ODYSSEE [11] has been utilized to obtain complex derivatives such as the numerical flux derived in (12).

3.3.3 Automatic differentiation by ODYSSEE

From the previous results, we see that to have the gradient of the cost function, we need to get the following derivatives :

- $\left\{\dfrac{\partial J}{\partial z_m}\right\}_{1\leq m\leq M}$: this computation involves the CAD software subroutines defining the *surface* mesh from parameters z. In our present case, cubic splines are used and the application of an automatic differentiator like ODYSSEE is straightforward.

- $\left\{\dfrac{\partial \mathbf{E}_i}{\partial z_m}\right\}_{1\leq i\leq N\ ,\ 1\leq m\leq M}$: in addition to the CAD software subroutines this computation involves the mesh deformation step to move the interior mesh points. Since this is done via a linear elasticity problem, the resulting Jacobian is easily computed. However, since the amount of storage for $\left\{\dfrac{\partial \mathbf{E}_i}{\partial z_m}\right\}$ is of the order of $N \times M$, which can be very large, we have to set up a procedure to compute the matrix vector product $\left\{\displaystyle\sum_{j=1}^{N} << \Psi_j\ ,\ \frac{\partial \mathbf{E}_j}{\partial z_m} >>\right\}_{1\leq m\leq M}$ which requires considerably less storage.

- $\left\{\dfrac{\partial J}{\partial \mathbf{W}_i}\right\}_{1\leq i\leq N}$: application of ODYSSEE gives us a straightforward computation for this term.

- $\left\{\mathbf{A}^T_{ji} = \left(\dfrac{\partial \mathbf{E}_j}{\partial \mathbf{W}_i}\right)^T\right\}_{1\leq i,j\leq N}$: this is the most time-consuming part of the gradient computations. It involves only the steady-state residual vector $\mathbf{E}_i$ which can be evaluated as a first-order or a second-order scheme.

For this last term, application of the automatic differentiator ODYSSEE in its adjoint mode has been done in a careful step-by-step manner and a thorough validation of the results has been done. Once again, in order to save storage space, only matrix-vector product subroutines are set-up, requiring that the adjoint equation be solved with an iterative algorithm. Verification of these subroutines is done in the following way :

Let F be a real valued function :

$$\begin{aligned} F:\ IR^p &\rightarrow IR^q \\ x &\rightarrow F(x) \end{aligned}$$

and x_0 be a vector in IR^p. We assume that we have access to the following operation $\dfrac{dF}{dx}(x_0).\delta x$, for a given δx in IR^p.

Using Taylor expansion, a centered finite-difference approximation for the derivative will give the following error estimation :

$$\frac{F(x_0 + \varepsilon\delta x) - F(x_0 - \varepsilon\delta x)}{2\varepsilon} - \frac{dF}{dx}(x_0)\delta x \approx O(\varepsilon^2)\| \delta x \|^3 \ , \ \forall \varepsilon > 0 \quad (24)$$

We have done these verifications for a two-dimensional flow around a NACA0012 profile at free-stream Mach number of 0.7 and an angle of attack of 1.49 degree .The mesh used has 1814 vertices and 3425 triangles. Using the second-order centered scheme described in section 3.1.3, we have converged to the steady-state $\{x_i^0\}_{1\leq i\leq N}$ and compute the matrix-vector product $\{\mathbf{A}_{ij}(x^0)\}_{1\leq i,j\leq N}\{\delta x_j\}_{1\leq j\leq N}$ for a random vector $\{\delta x_j\}_{1\leq j\leq N}$ of unit norm. Figure (7)shows the error estimation of formula (24) for different values of ε. We can see the variation in ε^2 up to a certain value of ε, under which the round-off error becomes overwhelming.

3.4 An optimization algorithm

Finally, we can sum up our results and propose the following optimization algorithm :

1 Mesh construction around the initial profile with initial parameters $\{z_m\}^0_{1\leq m\leq M}$

2 Solution of the state equation (second-order approximation) by a pseudo-unsteady implicit method using the first-order Jacobian

3 Computation of derivatives $\left\{\frac{\partial J}{\partial \mathbf{W}_j}\right\}_{1\leq j\leq N}$

4 Solution of the adjoint equation, either with

4a the first-order Jacobian transposed of step 2

or

4b the second-order Jacobian, using the first-order Jacobian as preconditionner

5 Computation of the gradient $\left\{\frac{dj}{\partial z_m}\right\}_{1\leq m\leq M}$

6 Conjugate gradient minimization of j with gradient $\left\{\frac{dj}{\partial z_m}\right\}_{1\leq m\leq M}$

7 Upgrade of optimization variables $\{z_i\}^n_{1\leq m\leq M}$

8 Construction of a new mesh

8a *surface* mesh perturbation by $\{z_i\}^n_{1\leq m\leq M}$

8a *volume* mesh deformation by solution of an elasticity equation

Go back to step 2

4 Numerical experiments

4.1 Design with a small number of parameters

4.1.1 Description of the physical problem

We study a very simple modelization, of the pseudo-balancing problem of a civil airplane ($\longrightarrow$ naturally stable) in landing approach (cf. Fig.1).

Figure 1: Civil aircraft

In the part of the flight domain, the speed must be reduced. To maintain a sufficient lift, the lift coefficient(C_L) must be increased by augmenting the angle of attack (α). However, the angle of attack must lie of course in the flight domain ($\alpha_{min} \leq \alpha \leq \alpha_{max}$).

The deploying of the highlift devices (slat and flap) will assume the increase of C_L, and if the couple (C_L,v) is reachable in the admissible flight domain, the desired C_L allowing to balance the plane.

The presence of a stabilizer allows to obtain a null moment ($C_M = 0$) around the centroïd of the plane. In our simple model (wing + forebody + flap), the C_L can be estimated, supposing the distance between the aerodynamic

center of the wing and the one of the stabilizer equal to n times the chord (of the wing). We have :

$$C_L^{\text{airplane}} \simeq C_L^{\text{wing}} + \frac{1}{n} C_M^{\text{wing}}$$

obtained by writing that we are at the airplane equilibrium ($C_M^{\text{airplane}} = 0$). For the example presented below, n is taken equal to 5 (a typical value for civil business jets).

4.1.2 Equilibrium with forebody, flap and incidence

The methodology described herebefore is used to compute the configuration corresponding to the prescribed value of C_L. Three parameters (α, δ^b and δ^v) are taken as design variables.

The objective is to obtain a C_L^{airplane} equal to 4.481 .

One can see on Fig.2 the evolution of C_L^{airplane}, on Fig.3 the evolution of the incidence angle α, which stabilizes itself at $-0.36°$, starting from $0°$, on Fig.4 the evolution of the forebody's incline angle δ^b, on Fig.5 the evolution of the flap's incline angle δ^v, both of them standing around a convergence value of $28.95°$(starting from$29.00°$) and $-19.04°$(starting from $-19.00°$) respectively. On Fig.6, one can see the evolution of the cost function's decimal logarithm. The desired solution is obtained in a few numbers of step. Concerning this study, more details could be found in [16].

4.2 Multipoint optimization

4.2.1 Testing the adjoint on a centered-scheme

The above algorithm has been applied to airfoil design problems using the conjugate gradient algorithm as optimizer. State equations are solved by the centered scheme described in the section 3.1.3.

The first computation deals with a reconstruction problem of a NACA63215 airfoil starting from a NACA0012 profile and aims to evaluate on this academic case the ability of the method. We have 8 control points on the profile, 550 nodes in the mesh. It is a subsonic test case ($M_\infty = 0.40$). We compare the results of using an adjoint operator of the first-order and of the second-order.

Despite the approximate discrete sensitivity approach chosen here, the NACA63215 airfoil is recovered in about 22 optimization steps if you choose a second-order adjoint. We observe that in the first-order adjoint case, we are blocked at the 15th iteration, and couldn't converge better as for the profile (Fig.8), as for the pressure distribution (Fig.9).

4.2.2 Test case TE4

A multipoint problem has been treated by the above method. The objective of the problem is to find a profile combining desired properties in subsonic abd transonic regimes. Two profiles presented in Fig.10 corresponding respectively to a highlift airfoil designed for low speed regime and a low drag one for transonic regime have been selected. Pressure distributions obtained for each airfoil (at $M_\infty = 0.2$, $\alpha = 9°$ and $M_\infty = 0.77$, $\alpha = 1°$ respectively), have been taken as target values to form a cost function blending the two individual inverse cost functions. The obtained profile is compared to the initial one (taken as a NACA4412 airfoil) on Fig.11. Pressure distributions for each regime on initial and final shapes are compared with the target one on Fig.12 and Fig.13 respectively.

The number of control points defining the optimization parameters is 35. The behaviour of the cost function with respect to the number of flow evaluation is presented and shows the difficult convergence of the method. An "optimized" profile is nevertheless obtained after 30 optimization steps. This profile is presented and some oscillations visible near the trailing edge indicates that the control is not sufficiently regular.

5 Conclusion

Control theory applied to the discrete system using an approximate discrete sensitivity analysis provides an effective approach to treat shape optimization problems using even complex unstructured mesh simulation tools.

In this paper, this approach has been illustrated in a pre-industrial environment, resulting in a general two-dimensional aerodynamic shape optimization tool which combines :

- a geometric surface representation by cubic splines;
- an unstructured 2D mesh generator/deformer;
- a CFD solver based on Euler equations.

This tool is an useful test-bed for different numerical strategies to be adapted in more complex 3D cases. Future work will deal with multipoint optimization and constrained optimization based on interior point methods.

The authors thanks A. Dervieux and his team from INRIA Sophia Antipolis and E. Laporte for his help in implementing the methods.

References

[1] T.J. BARTH and S.W. LINTON.- An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation, AIAA paper 95-0221, 33rd Aerospace Sciences Meeting and Exhibit, Reno, 1995.

[2] B. BERDE and M. BOREL.- Moment Approach for the Navier-Stokes Equations, AIAA paper 95-1663, 12th Computational Fluid Dynamics, June 19-22, 1995.

[3] A. DERVIEUX, L. FEZOUI, M.P. LECLERCQ and B. STOUFFLET.- A general upwind formulation for compressible flows on multi-element meshes, in Numerical Methods for Fluid Dynamics 4, M.J. Baines and K.W. Morton eds., Oxford Science Publications, 1993.

[4] J. CEA.- Numerical Methods of Shape Optimal Design. in Optimization of Distributed Parameter Structures (eds. E.J. Haug and J. Cea) Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1981.

[5] S. CHAMPIER., T. GALLOUET and R. HERBIN.- Convergence of an upstream finite volume scheme on a triangular mesh for a nonlinear hyperbolic equation, preprint.

[6] R. STRUIJS, P.L. ROE and H. DECONINCK.- Fluctuation splitting schemes for the 2D Euler Equations, Von Karman Institute LS 1991-01 in CFD, 1991.

[7] A. DERVIEUX, J.M. MALE, N. MARCO, J. PERIAUX, B. STOUFFLET and H.Q. CHEN.- Some Recent Advances in Optimal Shape Design for Aeronautical Flows, in Computational Fluid Dynamics '94, J. Wiley, 1994.

[8] J. REUTHER anf A. JAMESON.- Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory, AIAA paper 95-0123, 33rd Aerospace Sciences Meeting and Exhibit, Reno, 1995.

[9] A. JAMESON.- Numerical Solution of the Euler Equations for Compressible Inviscid Fluids. in Numerical Methods for the Euler equations in Fluid Dynamics, ed. by F. Angrand, A. Dervieux, J.A. Desideri and R. Glowinski (SIAM, Philadelphia), 1985.

[10] J-L. LIONS.- Contrôle optimal des systèmes gouvernés par des équations aux dérivées partielles, Dunod, Gauthier-Villars, Paris 1968.

[11] N. ROSTAING, S. DALMAS and A. GALLIGO.- Automatic Differentiation in Odyssée. Tellus, (45A):358-368, 1993.

[12] J. PERAIRE, J. PEIRO, L. FORMAGGIA, K. MORGAN and O.C. ZIENKIEWICZ.- Finite Element Euler Computations in Three Dimensions, Int. J. Numer. Methods Eng. 26, 2135-2159, 1988.

[13] B. PERTHAME, Y. QIU and B. STOUFFLET.- Sur la convergence des schémas fluctuation-splitting pour l'advection et leur utilisation en dynamique des gaz, C.R. Académie des Sciences Paris, t.319, Série I, 283-288, 1994.

[14] O. PIRONNEAU.- Optimal Shape Design for Elliptic Systems, Springer-Verlag, New-York, 1984.

[15] E. POLAK.- Computational Methods in Optimization : a unified approach. Mathematics in Science and Engineering. Vol. 77, 1971.

[16] Q.V. DINH, G. ROGE, C. SEVIN and B. STOUFFLET.- Contrôle à petit nombre de paramètres en aérodynamique, 32ème colloque d'aérodynamique appliquée, Ecole Centrale de Lyon, 25-27 Mars 1996.

[17] F. CHALOT, M. MALLET and M. RAVACHOL.- A Comprehensive Finite Element Navier-Stokes Solver for Low and High-Speed Aircraft Design, AIAA paper 94-0814, 32nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, 1994.

[18] B. VAN LEER.- Flux Vector Splitting for the Euler Equations, Lecture Notes in Physics, vol. 170. page 405-512, 1982.

[19] Q.V. DINH, B. STOUFFLET and A. VOSSINIS.- ICIAM Conference, Philadelphia, 1993.

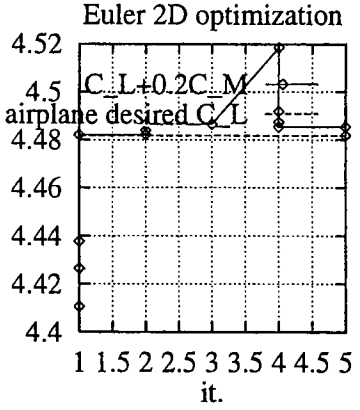

Figure 2: Evolution of C_L^{airplane}

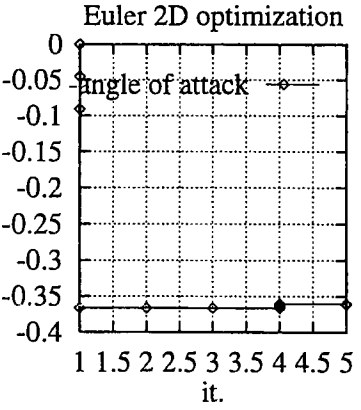

Figure 3: Evolution of $-\alpha$

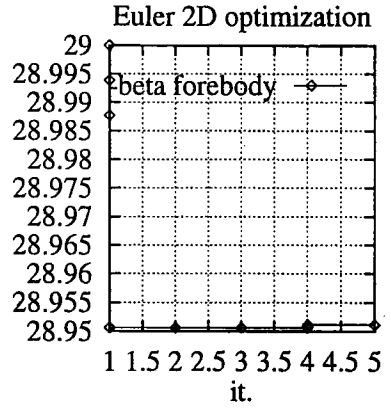

Figure 4: Evolution of $-\delta^b$

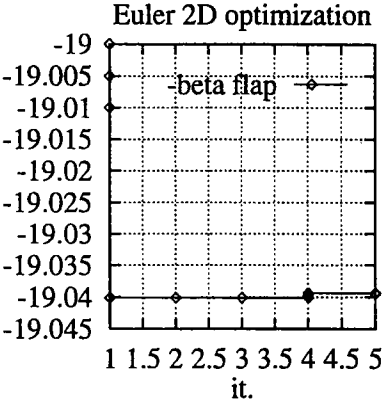

Figure 5: Evolution of $-\delta^v$

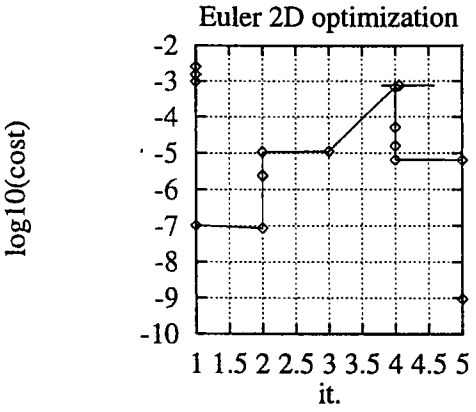

Figure 6: Evolution of the cost function's decimal logarithm

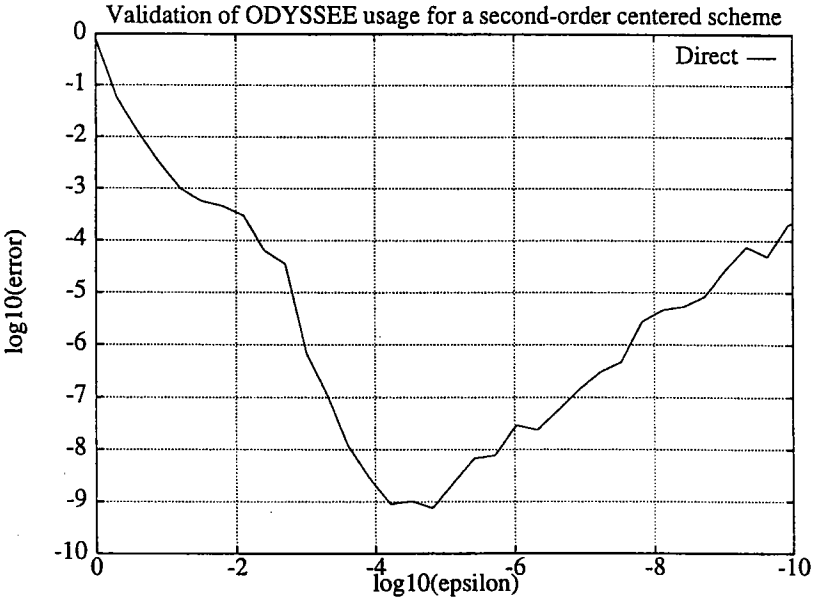

Figure 7: Odyssee validation

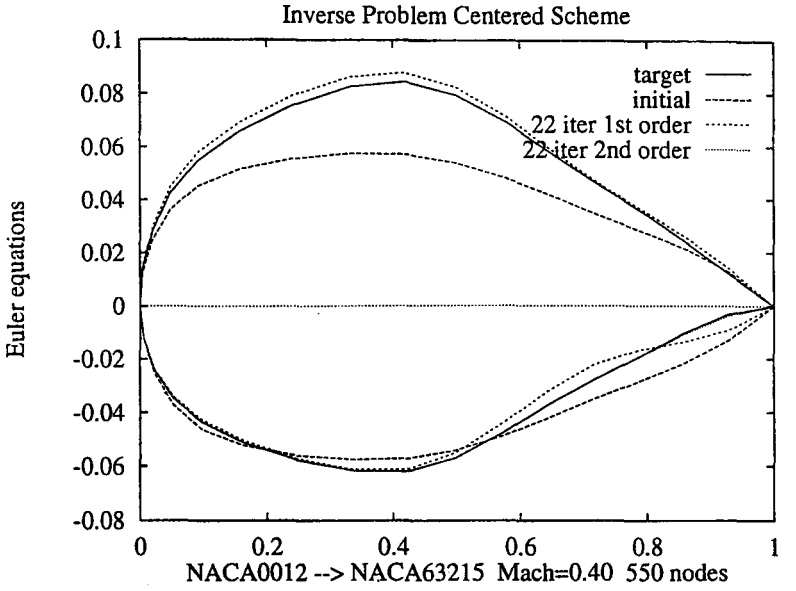

Figure 8: Profile

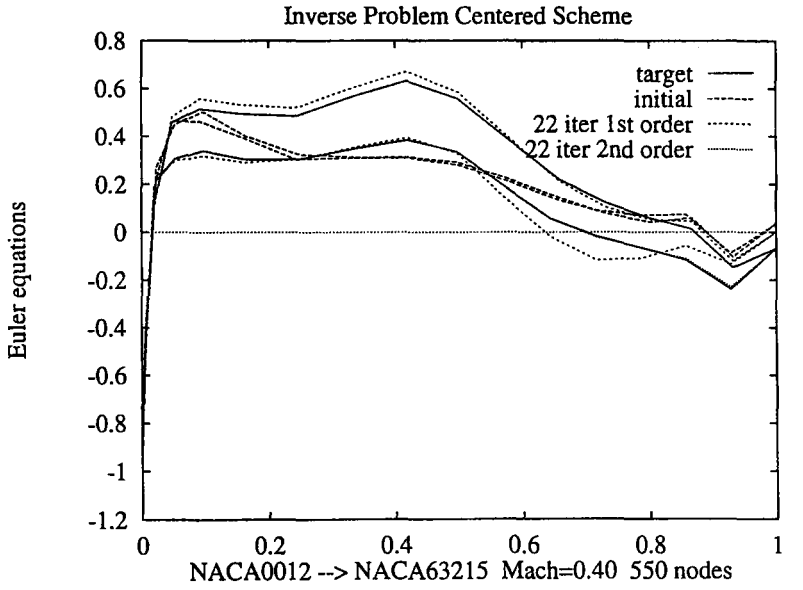

Figure 9: Pressure distribution

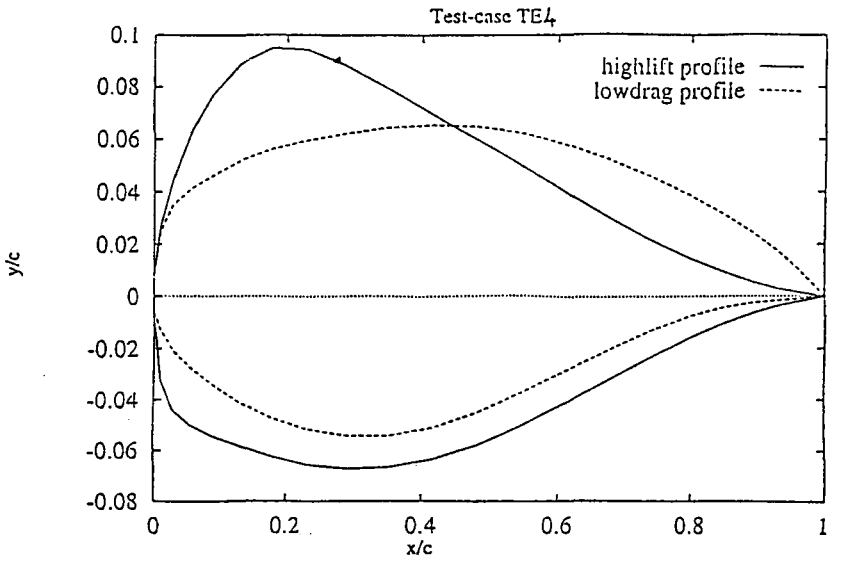

Figure 10: Profiles defining objective function

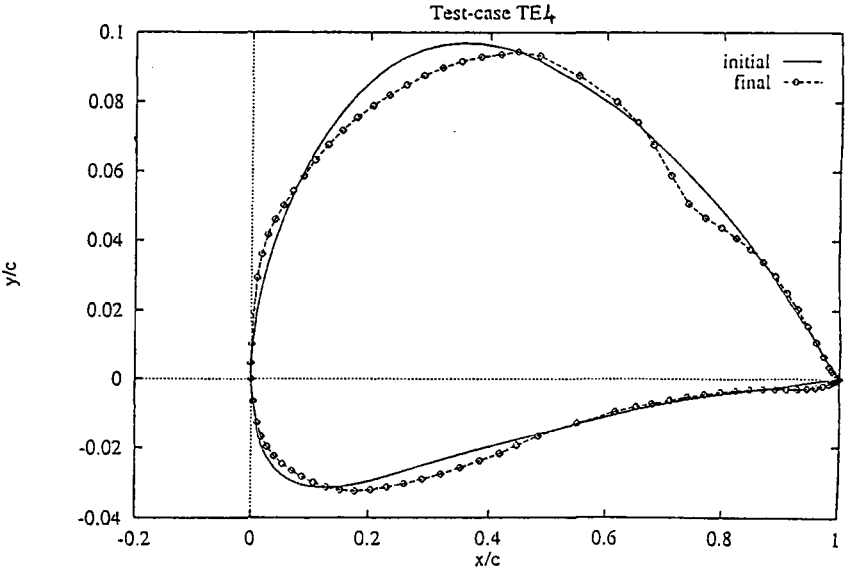

Figure 11: Initial and final profile

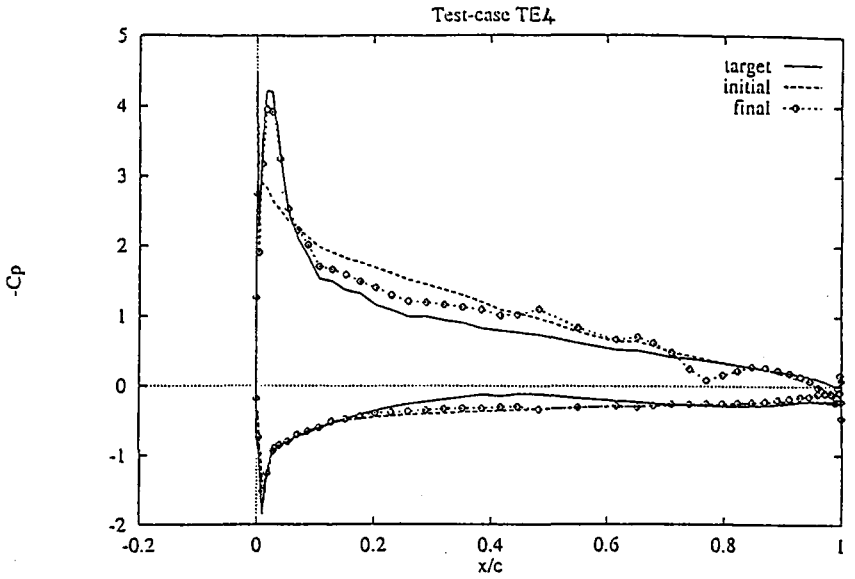

Figure 12: Pressure distribution for highlift condition

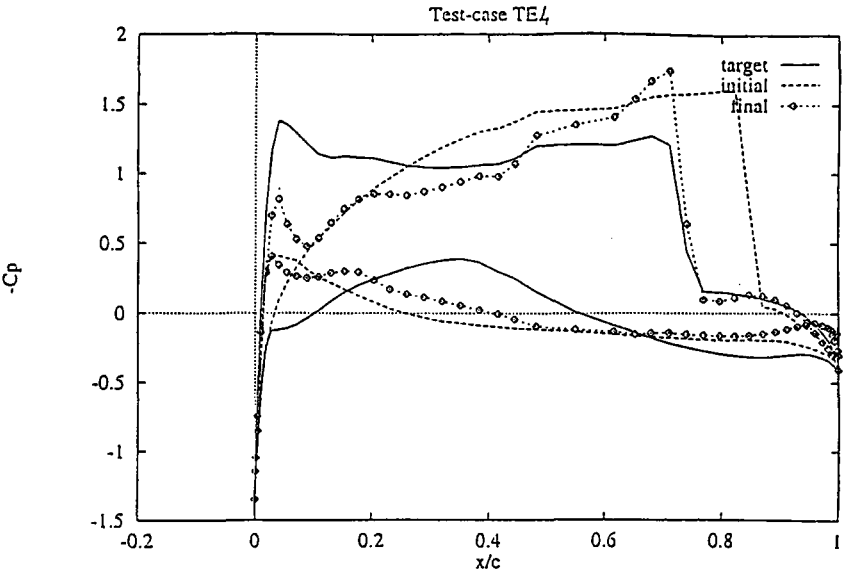

Figure 13: Pressure distribution for lowdrag condition