A quasi-Newton interior point algorithm applied to constrained optimum design in computational fluid dynamics

José Herskovits* — Emmanuel Laporte — Patrick Le Tallec****
Gines Santos*

** COPPE, Federal University of Rio de Janeiro*
Caixa Postal 68503, 21945-970 Rio de Janeiro, Brésil

*** INRIA Rocquencourt, BP 105, 78153 Le Chesnay cedex*

ABSTRACT. *This paper introduces and describes a second order interior point method well adapted to constrained shape optimal design in engineering. The theoritical background is presented and detailed implementation procedures are given in the case of nonlinear inequality constraints. The algorithm is then applied to two significative shape optimum problems in Computational Fluid dynamics.*

RÉSUMÉ. *Cet article propose et décrit un algorithme de point intérieur pour l'optimisation de formes en ingénierie. Il rappelle les principes de base de la méthode et décrit en détails les procédures de calcul nécessaires à son utilisation sur ordinateur, dans le cas de problèmes avec contraintes de type inégalités non linéaires. L'algorithme est enfin illustré par deux exemples d'application à des problèmes d'optimisation de formes en Mécanique des Fluides.*

KEY WORDS : *interior points, feasible sets, quasi-Newton, deflexion, line search, adjoint state, CFD.*
MOTS-CLÉS : *points intérieurs, quasi-Newton, déflexion, recherche linéaire, état adjoint, mécanique des fluides numérique.*

1. Introduction

This paper is concerned with the local numerical minimization of a function submitted to a set of nonlinear smooth equality and inequality constraints. Engineering design is a natural application for this problem, since designers want to find the best design that minimizes a given cost and satisfies all the requirements of feasibility. Calling $[x_1, x_2, ..., x_n]$ the design variables, $f(x)$ the objective function, $g_i(x)$; $i = 1, 2, .., m$ the inequality constraints and $h_i(x)$; $i = 1, 2, .., p$ the equality constraints, the optimization problem can be written in all generality

$$\left.\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ \text{and} & h(x) = 0, \end{array}\right\} \qquad (1)$$

where $g \in R^m$ and $h \in R^p$ are functions in R^n.

In optimal design, $x \in R^n$ usually denote certain space coordinates of a set of control points from which the shape of the object under optimisation is deduced by spline interpolation. Moreover, in such problems, the cost function is usually a fuction of two variables $f(x) = \tilde{f}(x, \tilde{u}(x))$, the state variable $\tilde{u}(x) \in U$ being implicitely defined as a function of x by solving a state equation $E(x, \tilde{u}(x)) = 0 \in U$ which characterises the state of the system under study as a function of its shape. Such optimum design problems have two main characteristics that make them difficult to solve. First, the cost function is very difficult to compute because each evaluation requires the solution of a the state equation. Second, the resulting optimisation problem is often poorly conditioned and higly constrained.

Feasible direction algorithms are an important class of methods for solving such constrained optimization problems. At each iteration, the search direction is a feasible direction of the inequality constraints and, at the same time, a descent direction of the objective or an other appropriate function. A constrained line search is then performed to obtain a satisfactory reduction of the function, without loosing the feasibility.

The fact of giving feasible points makes feasible direction algorithms very efficient in engineering design, where functions evaluation is in general very expensive. Since any intermediate design can be employed, the iterations can be stopped when the cost reduction per iteration becomes small. Moreover, there are also several examples that deal with an objective function, or constraints, that are not defined at infeasible points. This is very frequent with size or shape constraints in structural optimization: structures or flows become unstable or impossible to compute for certain shapes. In another direction, feasible points are very useful for real time problems. When applying feasible direction algorithms to real time problems, as feasibility is maintained and cost reduced, the controls can be activated at each iteration. Last, second order feasible direction algorithms have the additional advantage of being rather insensitive to the condition number of the problem, which is a big advantage in engineering design.

In this paper we discuss the numerical implementation of a quasi - Newton feasible direction algorithms that uses fixed point iterations to solve the nonlinear equalities included in the Karush-Kuhn-Tucker optimality conditions. With the object of ensuring convergence to admissible points, the system is solved in such a way as to have the inequalities in Karush-Kuhn-Tucker conditions satisfied at each iteration. Based on the present general technique, first order, quasi Newton or Newton algorithms can also be obtained. In particular, the algorithms described in [6, 8] and the quasi Newton one presented in [7] can be recovered.

each iteration, followed by an inexact line search. In practical applications, advantage of the structure of the problem and particularities of the functions in it, can be taken to improve calculus efficienty. This is the case of the Newton algorithm for linear and quadratic programming presented in [17], the Newton algorithm for limit analysis of solids, [19] and the Newton algorithm for stress analysis of linear elastic solids in contact, [1, 18].

Several problems in Engineering Optimization were solved using the present method. We can mention applications in structural optimization, [10, 16], fluid mechanics, [2, 4, 5] and multidisciplinary optimization with aerodynamics and electromagnetism, [2, 3]. Two applications are presented at the end of the paper in order to illustrate the performances of the proposed algorithms in aerodynamic optimum design.

2. The feasible direction interior point method

To describe the basic ideas of these technique, we consider the inequality constrained optimization problem

$$\left.\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0. \end{array}\right\} \tag{2}$$

We denote $\nabla g(x) \in R^{n\times m}$ the matrix of derivatives of g and call $\lambda \in R^m$ the vector of dual variables, $L(x,\lambda) = f(x) + \lambda^t g(x)$ the Lagrangian and $H(x,\lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 g_i(x)$ its Hessian. Finally, $G(x)$ denotes a diagonal matrix such that $G_{ii}(x) = g_i(x)$.

The corresponding Karush Kuhn Tucker first order optimality conditions can be expressed as

$$\nabla f(x) + \nabla g(x)\lambda = 0 \tag{3}$$

$$G(x)\lambda = 0 \tag{4}$$

$$\lambda \geq 0 \tag{5}$$

$$g(x) \leq 0. \tag{6}$$

Modern interior point techniques try to solve the quadratic optimality conditions (3)-(4) by a Newton's algorithm, which is modified in order to verify the admissibility inequalities (5)-(6) at each step. In this framework, a typical Newton's iteration to solve the nonlinear system of equations (3)-(4) in (x,λ) is given by

$$\begin{bmatrix} B & \nabla g(x^k) \\ \Lambda^k \nabla g^t(x^k) & G(x^k) \end{bmatrix} \begin{bmatrix} x_0^{k+1} - x^k \\ \lambda_0^{k+1} - \lambda^k \end{bmatrix} = -\begin{bmatrix} \nabla f(x^k) + \nabla g(x^k)\lambda^k \\ G(x^k)\lambda^k \end{bmatrix} \tag{7}$$

where (x^k,λ^k) is the starting point of the iteration and $(x_0^{k+1},\lambda_0^{k+1})$ is the new estimate, $B = H(x^k,\lambda^k)$ and Λ^k a diagonal matrix with $\Lambda_{ii}^k \equiv \lambda_i^k$.

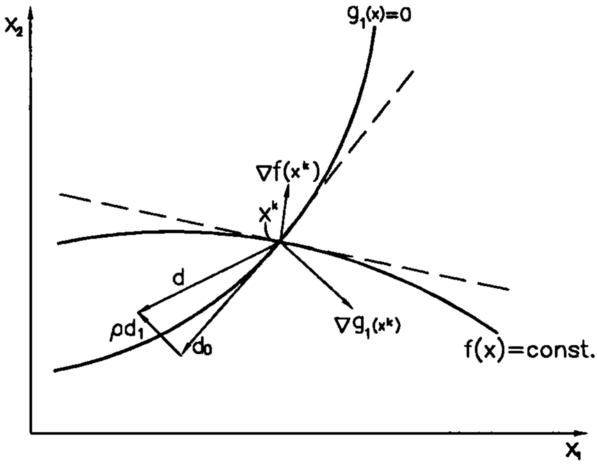

Figure 1: Search direction

In this paper, we can take B equal to a quasi-Newton estimate of $H(x, \lambda)$ or to the identity matrix. To ensure global convergence, the updating rule for B must generate positive definite matrices, (see [11]).

Iterations (7) are modified in a way to get, for a given interior (admissible) pair (x^k, λ^k), a new interior estimate with a better objective. With this purpose, a direction $d_0^k = x_0^{k+1} - x^k$ is first defined in the primal space by the Newton update (7):

$$B^k d_0^k + \nabla g(x^k) \lambda_0^{k+1} = -\nabla f(x^k) \tag{8}$$

$$\Lambda^k \nabla g^t(x^k) d_0^k + G(x^k) \lambda_0^{k+1} = 0, \tag{9}$$

It can be proved that d_0^k is a descent direction of f. However, d_0^k is not useful as a search direction since it is not necessarily feasible. This is due to the fact that if any constraint goes to zero, (9) forces d_0^k to tend to a direction tangent to the feasible set, (see [11], and thus the search line $x^k + td_0^k$ will not in general be included in this feasible set.

To obtain a feasible direction, we define the new linear system in d^k and $\bar{\lambda}^{k+1}$

$$B^k d^k + \nabla g(x^k) \bar{\lambda}^{k+1} = -\nabla f(x^k) \tag{10}$$

$$\Lambda^k \nabla g^t(x^k) d^k + G(x^k) \bar{\lambda}^{k+1} = -\rho^k \lambda^k, \tag{11}$$

obtained by adding a negative vector to the right side of (9), where $\rho^k > 0$, d^k is the new direction and $\bar{\lambda}^{k+1}$ is the new estimate of λ. We have now that d^k is a feasible direction, since $\nabla g_i^t(x^k) d^k = -\rho^k < 0$ for the active constraints.

The inclusion of a negative number in the right hand side of (9) produces a deflection of d_0^k, proportional to ρ^k, towards the interior of the feasible region. To ensure that d is a descent direction also, we establish an upper bound on ρ^k by imposing

$$(d^k)^t \nabla f(x^k) \le \alpha (d_0^k)^t \nabla f(x^k); \alpha \in (0,1), \tag{12}$$

which implies $(d^k)^t \nabla f(x^k) < 0$. Thus, d^k is a feasible descent direction.

To obtain the upper bound on ρ^k, we solve the linear system in (d_1^k, λ_1^{k+1})

$$B d_1^k + \nabla g(x^k) \lambda_1^{k+1} = 0 \tag{13}$$

$$\Lambda^k \nabla g^t(x^k) d_1^k + G(x^k) \lambda_1^{k+1} = -\lambda^k. \tag{14}$$

Since

$$d^k = d_0^k + \rho^k d_1,$$

we have that, if

$$\rho^k < \frac{(\alpha - 1)(d_0^k)^t \nabla f(x^k)}{(d_1^k)^t \nabla f(x^k)}, \tag{15}$$

then (12) holds.

A new primal point is finally obtained by an inexact line search along d^k, looking for a new interior point $(x^k + t^k d^k)$ with a satisfactory decrease of the function. Different rules can be employed to define new positive dual variables λ^{k+1}.

In general, the rate of descent of f along the new direction d^k will be smaller than along d_0^k. This is a price that we pay for obtaining a feasible descent direction. To keep this price small in the neighborhood of the solution (and thus to maintain superlinear convergence), we will also bound ρ^k by an adequate function of the residual.

In Figure 1, the search direction of an optimization problem with two design variables and one constraint is illustrated. At x^k on the boundary, the descent direction d_0 is tangent to the constraint. Even if in this exemple we can prove that d_1 is orthogonal to d_0, in general we can only say that d_1 is in the subspace orthogonal to the active constraints at x^k. Since d_1 points to the interior of the feasible domain, it improves feasibility. Observe finally that for linear constraints, the tangent direction d_0 is admissible, and that therefore no deflexion along d_1 is needed for such constraints.

3. Line Search Procedures for Interior Point Algorithms

Interior Point algorithms need an efficient constrained line search. In the present method, once a search direction is obtained, the first idea would consist in finding t that minimizes $f(x^k + td^k)$ subject to $g(x^k + td^k) \le 0$. Instead of making such an exact minimization on t, it is in fact much more efficient to employ inexact line search techniques. For that, we have to state a criterium to know wether a proposed step length is good or not and if not to define an iterative algorithm to obtain a good step length. We recall below the inexact line search criteria proposed in [12], extending to interior point algorithms the Armijo's and Wolfe's criteria originally developed for unconstrained optimization.

3.1. Armijo's Constrained Line Search Scheme

Define the step length t as the first number of the sequence $\{1, \nu, \nu^2, \nu^3, ...\}$ satisfying

$$f(x + td) \leq f(x) + t\eta_1 \nabla f^t(x)d \tag{16}$$

and

$$g(x + td) \leq 0, \tag{17}$$

where both η_1 and ν belong to the open interval $(0, 1)$. □

3.2. Wolfe's Constrained Line Search Criterion

Wolfe's criterion adds to Armijo's constraints (16) and (17) the requirement that either the gradient direction has a significative change between two iterates or one constraint has a significative growth. It writes :

accept a step length t if (16) and (17) are true and at least one of the followings $m + 1$ conditions hold:

$$\nabla f^t(x + td)d \geq \eta_2 \nabla f^t(x)d \tag{18}$$

and

$$g_i(x + td) \geq \gamma g_i(x); i = 1, 2, ..., m \tag{19}$$

where now $\eta_1 \in (0, 1/2), \eta_2 \in (\eta_1, 1)$ and $\gamma \in (0, 1)$. □

Conditions (16) and (17) define upper bounds on the step length in both criteria and, in Wolfe's criterion, a lower bound is given by one of the conditions (18) and (19). The iterative procedure to find a step length satisfying Wolfe's criterion will be presented later.

4. The Algorithms

Based on the ideas presented above, we describe in this section the main algorithm for inequality constrained optimisation, including the quasi - Newton updating rule and the line search algorithm.

4.1. The Quasi - Newton Interior Point Algorithm

Parameters. $\alpha \in (0, 1)$ and positive φ, ϵ, β and λ^I.

Data. Initialize x such that $g(x) < 0$, $\lambda > 0$ and $B \in R^{n \times n}$ symmetric and positive definite.

Step 1. Computation of the search direction d.

i) Solve the linear system for (d_0, λ_0)

$$Bd_0 + \nabla g(x)\lambda_0 = -\nabla f(x), \tag{20}$$

$$\Lambda \nabla g^t(x)d_0 + G(x)\lambda_0 = 0. \tag{21}$$

If $d_0 = 0$, stop.

ii) Solve the linear system for (d_1, λ_1)

$$Bd_1 + \nabla g(x)\lambda_1 = 0, \tag{22}$$

$$\Lambda \nabla g^t(x)d_1 + G(x)\lambda_1 = -\lambda. \tag{23}$$

iii) If $d_1^t \nabla f(x) > 0$, set

$$\rho = inf[\varphi \parallel d_0 \parallel^2 ; \ (\alpha - 1)d_0^t \nabla f(x)/d_1^t \nabla f(x)]. \tag{24}$$

Otherwise, set

$$\rho = \varphi \parallel d_0 \parallel^2 . \tag{25}$$

iv) Compute the search direction

$$d = d_0 + \rho d_1. \tag{26}$$

Step 2. Line search.

Find a step length t satisfying a given constrained line search criterion on the objective function f and such that $g(x + td) < 0$.

Step 3. Updates.

i) Define a new x:

$$x = x + td$$

ii) Define a new λ:
Set, for $i = 1, m$,

$$\lambda_i := \sup\, [\lambda_{0i}; \epsilon \parallel d_0 \parallel^2]. \tag{27}$$

If $g_i(x) \geq -\beta$ and $\lambda_i < \lambda^I$, set $\lambda_i = \lambda^I$.

iii) Update B symmetric and positive definite.

iv) Go back to *Step 1.* □

We assume that the updating rule of B satisfies:
Assumption There are positive σ_1 and σ_2 such that

$$\sigma_1 \parallel d \parallel^2 \leq d^t B d \leq \sigma_2 \parallel d \parallel^2 \ \text{ for any } d \in R^n.$$

□

In *Step 1* ρ is bounded as in (15) and, in addition, it is not allowed to grow faster than d_0^2. The new components of λ are a second order perturbation of the components of λ_0, given by Newton's iteration (8). If β and λ^I are taken sufficiently small, then after a finite number of iterations, λ_i becomes equal to λ_{0i} for all active constraints. Observe also that λ_i can be eliminated from systems (20)-(21) or (22)-(23), yielding an equation with unknown d and matrix

$$M = B + \sum_i \frac{\lambda_i}{-g_i} \nabla g_i \nabla g_i^t \approx D^2 f + \sum_i \lambda_i g_i \left(\frac{1}{g_i} D^2 g_i - \frac{1}{g_i^2} \nabla g_i \nabla g_i^t \right).$$

The method can therefore be viewed locally as a penalty method associated to the penalty (barrier) function $\ln(-g_i)$, and to a dynamically updated penalty coefficient $\lambda_i^{k-1} g_i^{k-1}$. The second order corrections on λ_i control in fact this penalty coefficient from below.

The theoretical analysis in [11] includes first a proof that the solutions of the linear systems (20)-(21) and (22)-(23) are unique, provided that the vectors $\nabla h_i(x)$, for $i = 1, 2, ..., p$, and $\nabla g_i(x)$ for $i \in I(x)$ are linearly independent. Then, it is shown that under the previous assumption any sequence $\{x^k\}$ generated by the algorithm converges to a Karush-Kuhn-Tucker point of the problem, whatever updating strategy is used for λ and B. We also have that (x^k, λ_0^k) converges to a Karush-Kuhn-Tucker pair (x^*, λ^*) and, then global convergence in the dual space is also obtained.

In [11] it was also proved that the convergence of the present quasi - Newton algorithm is two-step superlinear, provided that a unit step length is obtained after a finite number of iterations.

In constrained optimization, B is a quasi - Newton approximation of the Hessian of the Lagrangian $H(x, \lambda)$. We can use the exact Hessian (exact Newton) or we could obtain B by using the same updating rules as in unconstrained optimization, but taking $\nabla_x L(x, \lambda)$ instead of $\nabla f(x)$.

However, since $H(x, \lambda)$ is not necessarily positive definite at a K-K-T point, it is not always possible to get B positive definite, as required by the present method. To overcome this difficulty, we employ BFGS updating rule as modified by Powell, [15]:

4.2. The Quasi - Newton Matrix Updating

Take

$$\delta = x^{k+1} - x^k$$

and

$$\gamma = \nabla_x L(x^{k+1}, \lambda_0^k) - \nabla_x L(x^k, \lambda_0^k).$$

If

$$\delta^t \gamma < 0.2 \delta^t B^k \delta,$$

then compute

$$\phi = \frac{0.8 \delta^t B^k \delta}{\delta^t B^k \delta - \delta^t \gamma}$$

and take

$$\gamma = \phi\gamma + (1-\phi)B^k\delta.$$

Set

$$B^{k+1} := B^k + \frac{\gamma\gamma^t}{\delta^t\gamma} - \frac{B^k\delta\delta^t B^k}{\delta^t B^k\delta}$$

□

4.3. The Line Search Algorithm

A step length satisfying Wolfe's criterion for interior point algorithms can be obtained iteratively in a similar way as in [13]. Given an initial t, if it is too short, extrapolations are done until a good or a too long step is obtained. If a too long step was already obtained, interpolations based on the longest short step and the shortest long step are done, until the criterion is satisfied. Since the function and the directional derivative are evaluated for each new t, cubic interpolations of f can be done. As the criterion of acceptance is not severe, the process generally requires very few iterations. Altogether, this leads to the following algorithm :

Parameters. $\eta_1 \in (0, 0.5)$, $\eta_2 \in (\eta_1, 1)$ and $\gamma \in (0,1)$.

Data. Define an intial estimate of the step length, $t > 0$. Set the upper estimate t_R and the lower estimate t_L to zero.

Step 1. Test for the upper bound on t.

If,

$$f(x+td) \leq f(x) + t\eta_1 \nabla f^t(x)d$$

and

$$g(x+td) \leq 0,$$

Go to *Step 2.* Else, the upper bound is too large and therefore go to *Step 4.*

Step 2. Test for the lower bound on t.

If

$$\nabla f^t(x+td)d \geq \eta_2 \nabla f^t(x)d,$$

or any

$$g_i(x+td) \geq \gamma g_i(x) \text{ for } i = 1, 2, ..., m,$$

then t **verifies Wolfe's Criterium. STOP.**

Else, go to *Step 3.*

Step 3. Get a longer t.

Set $t_L = t$

i) If $t_R = 0$, find a new t by extrapolation based on $(0, t)$.

ii) If $t_R > 0$, find a new t by interpolation in (t, t_R).

Step 4. Get a shorter t.

Set $t_R = t$.

Find a new t by interpolation in (t_L, t) Return to *Step 1.* □

5. Optimization of Riblets

5.1. Formulation of the Problem

The goal of the present application is the drag reduction of riblets. The problem to solve is to minimise the drag of the riblet with respect to its shape ω, while keeping a constant flow area. More precisely, we want to minimize the drag of a **steady laminar incompressible** flow inside a channel delimited by two vertical planes and two horizontal walls with riblets (the floor and the ceiling (Figure (2). The riblets are usually small and shaped like a saw tooth.

In theory, the flow is governed by the three-dimensional Navier-Stokes equations. Shape optimisation with such a state equation is out of reach. Therefore, we begin by neglecting all diffusion effects along the axial direction $z = x_3$. The resulting Navier-Stokes equations reduce then to the system

$$\begin{aligned} \nabla v + u_{3,3} &= 0 \\ u_3 v_{,3} + v.\nabla v - \nu \Delta v + \nabla p_t &= f \quad \text{in} \quad \Omega = \omega \times (0, z_{max}) \\ u_3 u_{3,3} + v.\nabla u_3 - \nu \Delta u_3 + p_{t,3} &= f_3. \end{aligned}$$

Above, $(v, u_3) = (u_1, u_2, u_3)$ denote the three dimensional velocity field, and the differential operators ∇, Δ denote the gradient and Laplacian with respect to the two-dimensional variables (x_1, x_2). Differentiation of a given function f with respect to the longitudinal direction $x_3 = z$ is denoted by $f_{,3}$. In addition, we also suppose that the pressure gradient along the x_3 direction is quasi independent of the two-dimensional variables (x_1, x_2). This amounts to assume that the total pressure field is of the form

$$p_t = \Lambda(x_3) + p(x_1, x_2, x_3)$$

with

$$\Lambda'(x_3) >> \frac{\partial p}{\partial x_3}.$$

The equations reduce then to the Parabolised Navier-Stokes equations (PNS)

$$\begin{aligned} \nabla v + u_{3,3} &= 0 \\ u_3 v_{,3} + v.\nabla v - \nu \Delta v + \nabla p &= f \quad \text{in} \quad \Omega = \omega \times (0, z_{max}) \qquad (28) \\ u_3 u_{3,3} + v.\nabla u_3 - \nu \Delta u_3 + \Lambda_{,3}(x_3) &= f_3. \end{aligned}$$

In view of the PNS equations (28), it seems perfectly natural to treat the axial variable $z = x_3$ as the standard time variable in a two-dimensional time evolution

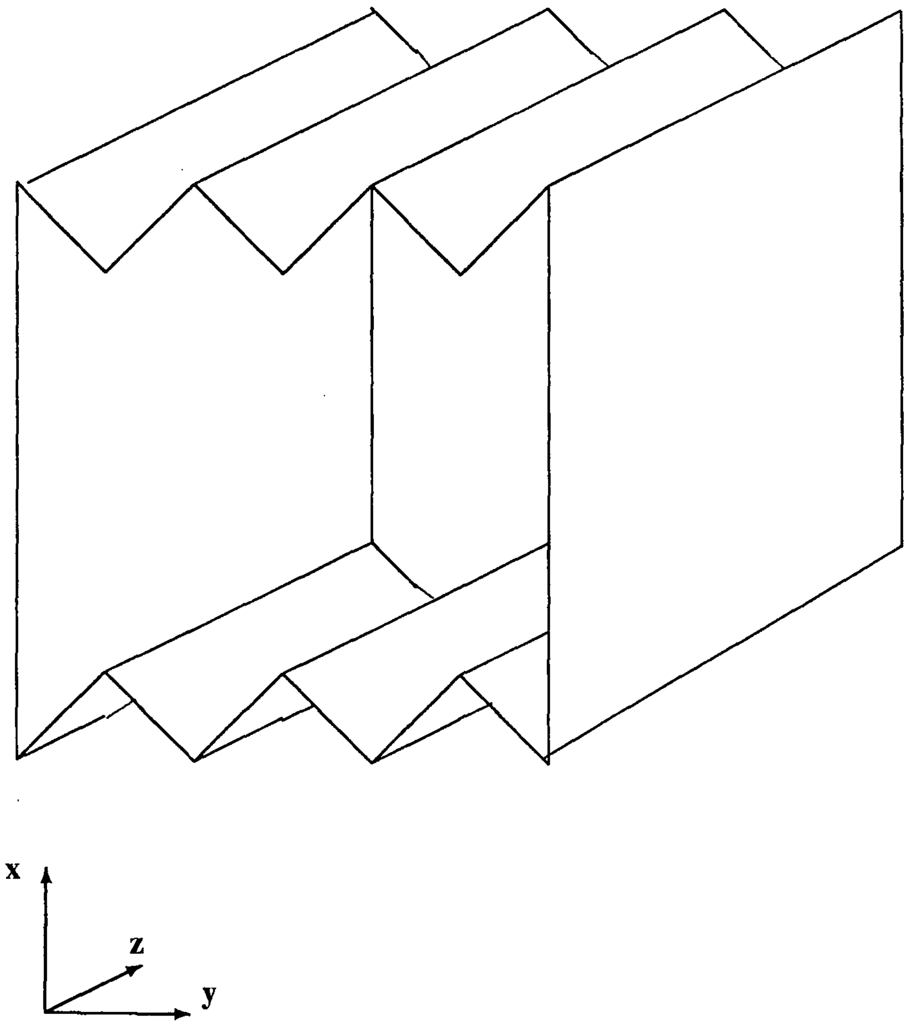

Figure 2: General geometry

problem, to treat all derivatives in $z = x_3$ by a first order backward Euler discretisation scheme, and to treat all convection terms in v by the method of characteristics.

For this purpose, for any given point $x \in \omega$, for any given cross section z, we first compute the characteristic curve χ^z arriving at point x in section z by solving the first order differential equation

$$\frac{\partial \chi^z(x;\tau)}{\partial \tau} = \frac{v(\chi^z,\tau)}{u_3(\chi^z,\tau)}, \tag{29}$$

$$\chi^z(x,0) = x. \tag{30}$$

We then approximate all convection terms on the cross section $n+1$ by the first order upwind formula

$$(u_3 f_{,3} + v\nabla f)(x, z^{n+1}) = u_3^n \Big(\frac{f^{n+1} - f^n o \chi^n(x, -\Delta z)}{\Delta z}\Big). \tag{31}$$

With this choice, the PNS equations written on a given section $n+1$ take the form

$$u_3^n \Big(\frac{u_3^{n+1} - u_3^n o\chi^n(-\Delta z)}{\Delta z}\Big) - \nu\Delta u_3^{n+1} + \lambda^{n+1} = f_3^{n+1}, \tag{32}$$

$$\int_\omega u_3^{n+1} = \int_\omega u_3^0, \tag{33}$$

$$\nabla . v^{n+1} = -\frac{u_3^{n+1} - u_3^n}{\Delta z}, \tag{34}$$

$$u_3^n \Big(\frac{v^{n+1} - v^n o\chi^n(-\Delta z)}{\Delta z}\Big) - \nu\Delta v^{n+1} + \nabla p^{n+1} = f^{n+1}. \tag{35}$$

Problem (32)-(35) is now a sequence of perturbed two-dimensional Stokes type problems which can be solved by standard finite element methods.

The associated optimal design problem to solve is then to minimize the drag

$$Min_{\omega \in S} J(\omega) \qquad (P_m)$$

with function cost

$$\begin{aligned} J(\omega) &= \int_{z=0}^{z_{max}} C_f dz \\ &= \frac{\nu}{2}\int_{z=0}^{z_{max}} \int_\omega |\nabla v|^2 + |\nabla u_3|^2 d\omega dz \\ &\approx \frac{\nu \Delta z}{2} \sum_{n=1}^{N-1} \{\int_\omega (|\nabla v^n|^2 + |\nabla u_3^n|^2\})d\omega \\ &\quad + \frac{\nu\Delta z}{4}\int_\omega (|\nabla v^N|^2 + |\nabla u_3^N|^2) d\omega + \frac{\nu \Delta z}{4}\int_\omega (|\nabla v^o|^2 + |\nabla u_3^o|^2) d\omega. \end{aligned}$$

Here v^o et u_3^o are the invariant initial data on the inflow cross section and the state variables $\vec{v} = (u_3, v, p_t)$ are solution of the above parabolic system (32)-(35) that we we write under the variational form

$$a(\vec{v}, \vec{\varphi}) = 0, \forall \vec{\varphi}.$$

Finally, S is the set of admissible controls defining the two-dimensional shape ω of the riblet.

We want to solve the above constrained optimisation problem by the quasi-Newton algorithm described in the preceeding section. This requires an analytic calculation of the gradient of J. For this purpose, we first transform the function to minimize and obtain the resulting gradient by introducing and computing an adjoint state.

5.2. Reduction to a transpiration problem

At first order, changing the shape of the flow domain while conserving its volume can be identified to a change of Dirichlet boundary conditions. To be more specific, let $\vec{v}$ be the solution of the flow equation

$$a(\vec{v},\vec{w}) = 0 \quad \text{on} \quad \Omega^i,$$

set on the initial domain Ω^i with boundary condition

$$\vec{v} = 0 \quad \text{on} \quad \Gamma_1^i.$$

Let any point $\vec{x}$ of the initial boundary Γ_1^i move of a given small displacement $d\vec{x}$. Let $\vec{v} + d\vec{v}$ be the solution of our initial equation on the same initial domain, but with the new boundary condition

$$\vec{v} + d\vec{v} = -\nabla\vec{v}.d\vec{x} \quad \text{on} \quad \Gamma_1^i.$$

At first order, we have on the new boundary

$$\begin{aligned}(\vec{v}+d\vec{v})(\vec{x}+d\vec{x}) &= (\vec{v}+d\vec{v})(\vec{x}) + \nabla(\vec{v}+d\vec{v}).d\vec{x}\\ &= -\nabla\vec{v}.d\vec{x} + \nabla\vec{v}.d\vec{x} + 0(|d\vec{x}|^2)\\ &= 0.\end{aligned}$$

Hence $\vec{v} + d\vec{v}$ is solution of the original flow problem set on the new domain $\Omega = \Omega^i + d\Omega$ with noslip boundary condition

$$\vec{v} + d\vec{v} = 0 \quad \text{on} \quad \Gamma_1^i + d\Gamma_1.$$

In other words, updating the shape of the domain Ω amounts to update the boundary conditions of the original problem by the quantity $-\nabla\vec{v}.d\vec{x}$.

In variational form, this means that we can locally replace the state equation for any shape configuration close to the initial shape Ω^i by the new equation

$$\begin{aligned}&a(\vec{v}+d\vec{v} - Tr^{-1}(\nabla\vec{v}.d\vec{x}), \vec{w}^i) = 0,\\ &\forall\vec{w}^i \in V^i, \vec{v}+d\vec{v} \in V^i,\end{aligned}$$

where the variational form a and the functional space V^i are associated to the fixed domain Ω^i and are considered as independent of the shape variables.

5.3. Calculation of the gradient

Introducing the Lagrange multiplier (the adjoint state) of this new state equation, we can reformulate our shape optimum problem (P_m) locally around the present shape Ω as the problem of finding the critical points of the lagrangian :

$$\mathcal{L}(d\vec{x}, d\vec{v}, \vec{w}) = J(\vec{x} + d\vec{x}, \vec{v} + d\vec{v}) - a(\vec{v} + d\vec{v} - Tr^{-1}(\nabla\vec{v}.d\vec{x})d\vec{v}, \vec{w}).$$

The differentiation with respect to $\vec{v}$ at the present configuration $d\vec{x} = 0$ yields the adjoint state equation $\frac{\partial \mathcal{L}}{\partial v} = 0$, that is

$$a'_u(d\vec{v}, \vec{w}) = \frac{\partial J}{\partial \vec{v}}(\vec{x}, \vec{v}).d\vec{v}, \forall d\vec{v}.$$

Then, if $\vec{v}$ and $\vec{\omega}$ are solutions of the state equation and adjoint state equations respectively, the total gradient of the cost function with respect to the control variable r is classically given by the simple formula

$$\begin{aligned} \frac{dJ}{d\vec{x}}(\vec{x}, \vec{v}(\vec{x})).d\vec{x} &= \frac{\partial \mathcal{L}}{\partial \vec{x}}(\vec{x}, \vec{v}).d\vec{x} \\ &= \frac{\partial J}{\partial \vec{x}}(\vec{x}, \vec{v}).d\vec{x} \\ &\quad -a'_u(-Tr^{-1}(\nabla\vec{v}.d\vec{x}), \vec{w}) \\ &= \frac{\nu}{2}\int_{\Gamma_1}\int_0^1 |\nabla\vec{v}|^2(\vec{x} + sd\vec{x})\frac{dads}{\vec{x}.\vec{n}} \\ &\quad -a'_u(-Tr^{-1}(\nabla\vec{v}.d\vec{x}), \vec{w}). \end{aligned}$$

5.4. One-dimensional Validation

The depth $\Delta = 2\delta$ of a tooth is the design parameter, which is to be optimized in the process. The Reynolds number is based on the horizontal velocity in the centre of the channel w_o, at the inflow cross section. The Reynolds number and riblet spacing are given by $Re = w_o D/\nu = 4200$ and $d/D = 0.1135$ respectively. In order to keep a constant cross section during the design, the height 2D of the riblet is the average height of the channel (measured at half slope) and not the maximum height.

We have observed in our direct simulations that all major effects on friction were governed by the first part of the riblet. Therefore, in our optimisation process, and in order to reduce simulation costs, we have reduced the computational domain to the strip

$$0 \leq z \leq 10D$$

and run all tests with $\Delta z = 0.1$.

The test optimisation problem tries to find the maximum value of δ for which the drag is less than 1.1 times the drag of a flat plate :

$$\max\delta, \text{ with } J(\delta) = \int_z C_f dz \leq 1.1 * J(0).$$

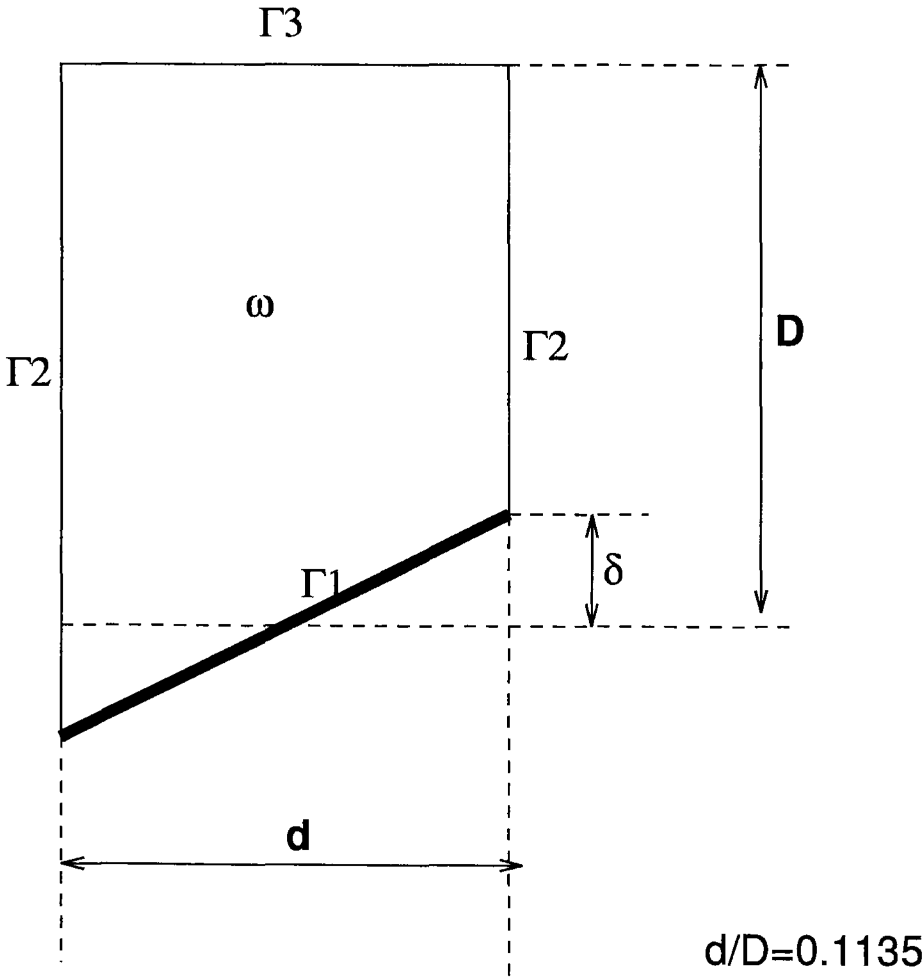

Figure 3: Cross section for a half riblet

iter	δ	val.max constraint
1.0	2.0335981e-02	-1.0507274e-04
2.0	7.4844836e-02	-4.2366251e-06
3.0	7.5881480e-02	-1.2365828e-06
4.0	7.6243493e-02	-1.8012456e-07
5.0	7.6303915e-02	-3.1610285e-09
6.0	7.6304996e-02	-5.7011147e-10

Table 1: Convergence of the Herskovits interior point algorithm for the inverse problem (optimisation of δ).

Using our interior point algorithm, which amounts here to use a succession of line searches, we obtain a solution $\delta = 7.6305\ 10^{-2}$ in six iterations. For this value, the drag constraint is saturated within a 10^{-10} accuracy (Table 2).

6. Airfoil Optimization for Unsteady Flows

6.1. Formulation of the Problem

The problem to solve is to minimize the drag of the airfoil with respect to its shape γ, while keeping a minimum area for the airfoil. More precisely, our goal is to minimize the drag of an airfoil γ in a **viscous transonic** flow in an unbounded domain. (cf. fig. 4)

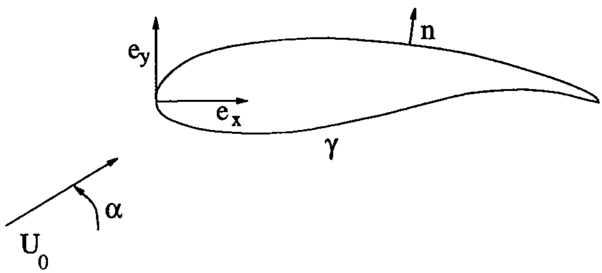

Figure 4: Airfoil in an unbounded domain (U_0 = velocity in the free stream).

When the angle of attack α is 0°, the drag is mostly due to pressure efforts and can therefore be approximated by

$$C_d\left(p_\gamma\left(t\right), \gamma\right) = \left(\int_\gamma p_\gamma\left(t\right) \vec{n}\, d\gamma\right) . \vec{e}_x,$$

where $p_\gamma\left(t\right)$ is the pressure distribution on the airfoil γ at the instant t. The pressure depends on the state variables $W_\gamma(t, \vec{x})$ which statisfy the state equation

$$E(\gamma, W_\gamma) = 0, \tag{36}$$

where E can be the Euler or the Navier-Stokes equations.

In our case, the initial state W_0 and the integration (pseudo)-time T are given data of the problem and we suppose that a converge steady state is reached at time T. Then for a given airfoil γ, we will solve the state equation (36) on $[0, T]$, then we will compute the following cost function

$$j(\gamma) = C_d(p_\gamma(T), \gamma),$$

which can be re-written as

$$j(\gamma) = J(\gamma, W_\gamma).$$

We define a set S_0 of admissible airfoils with the help of a "minimum airfoil" and a "maximum airfoil" (cf. fig. 5): if, for an airfoil $\gamma \in S_0$, we call ω_γ the open set of $\mathbb{R}^2$ which is enclosed by the curve γ, then we must have

$$\omega_{min} \subset \omega_\gamma \subset \omega_{max}$$

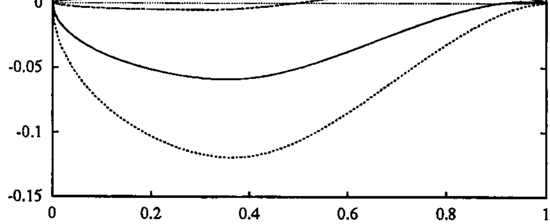

Figure 5: Definition of the set of admissible airfoils.

In the constraints, we also want the area of the airfoil to be always greater than a given area. This constraint can be expressed under the inequality

$$g(\gamma) \leq 0,$$

where

$$g(\gamma) = Cte - coef \times \int_{\omega_\gamma} dx.$$

Usually, the constant will be the initial area $g(\gamma_0)$ where γ_0 is the initial airfoil at the beginning of the optimization process, and the coefficient will be 1.01 .

The optimal design constrained problem to solve is then:

$$\min_{\gamma \in S_0} j(\gamma),$$

under the constraint

$$g(\gamma) \leq 0.$$

6.2. Discretization and Computation of the Gradient

In the present study, the state equations governing the external flow are the Navier-Stokes equations which can be written under the following form:

$$\frac{\partial W_\gamma}{\partial t} + \nabla.(F(W_\gamma) - N(W_\gamma)) = 0,$$

where F and N represent respectively the advective and viscous operators. To compute the state W_γ, we use the flow solver NSC2KE [14], using a kinetic scheme of 2^{nd} order, Van Albada delimitors and local time stepping. Using an explicit time integration, the discretized formulation of this equation writes

$$W_h^{k+1} - W_h^k = \mathcal{F}(X, W_h^k),$$

where X is the vector of the mesh nodal coordinates and W_h^k the discretized state variables at the time-step k. Then the discrete cost function j_h can be written

$$j_h(X) = J_h\left(X, W_h^{k_T}\right),$$

where k_T is the last time-step and J_h the discretization of J.

As it would not be reasonnable to compute the matrix $\frac{\partial W}{\partial X}$, we compute the gradient with the following algorithm using a discrete adjoint variable λ :

$$\begin{cases} \lambda_{k_T} = \frac{\partial J_h}{\partial W}\left(X, W_h^{k_T}\right), \\ \lambda_k = W_h^k + \left\langle \left(\frac{\partial \mathcal{F}}{\partial W}(X, W_h^k)\right)^*, \lambda_{k+1}\right\rangle \quad \text{for } k = k_T - 1, \ldots, 1, \\ \frac{dj_h}{dX}(X) = \frac{\partial J_h}{\partial X}\left(X, W_h^{k_T}\right) + \sum_{k=0}^{k_T-1} \left\langle \left(\frac{\partial \mathcal{F}}{\partial X}(X, W_h^k)\right)^*, \lambda_{k+1}\right\rangle. \end{cases}$$

The routines which compute the products $\langle ., . \rangle$ are obtained by the automatic differentiator Odyssée [6]. As explained in a companion paper in this journal issue, automatic Differentiation (**AD**) is a tool which takes the Fortran subroutine computing a function and gives a Fortran subroutine computing the derivatives of this function (cf. following table).

math. func.	$y = f(x)$	$dydx = \frac{\partial f}{\partial x}(x)$	$xad = \left\langle \left(\frac{\partial f}{\partial x}(x)\right)^{*}, yad \right\rangle$
Routines	`f(x,y)`	`fd(x,y,dydx)`	`fad(x,y,xad,yad)`
In. var.	`x`	`x`	`x,yad`
Out. var.	`y`	`y,dydx`	`xad`
	`Odyssée` in.	`Odyssée` output	

We show respectively on the next figures 6 and 7 a partial view of a typical mesh (2001 nodes) and the vector field of the gradient of j_h with respect to X:

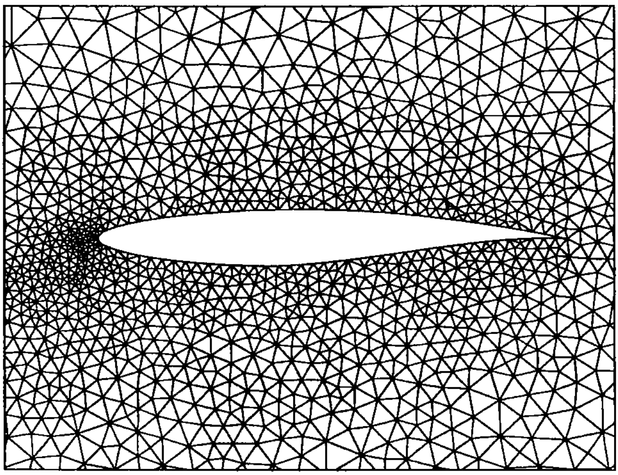

Figure 6: Mesh around an airfoil.

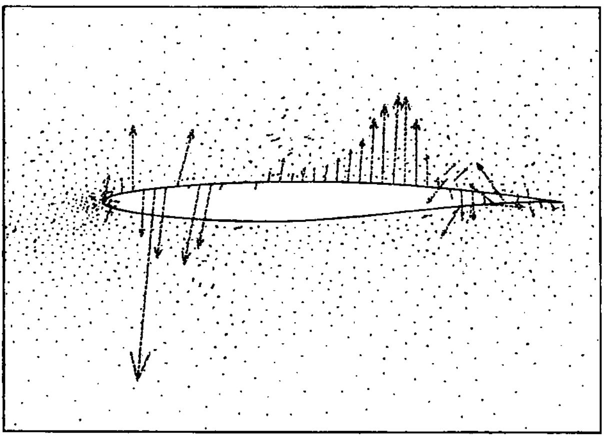

Figure 7: Vector field of the gradient around the airfoil.

Having obtained the whole routine which computes the gradient of the function j_h with respect to X, we only need to add a function transforming the discrete control variable ξ_h into node coordinates X. This uses a smooth mesh adapter $X(\xi_h)$. Then the complete function writes

$$j_h(X(\xi_h))$$

and the gradient of the cost function takes the final form :

$$\left\langle \left(\frac{dX}{d\xi_h}(\xi_h) \right)^*, \frac{dj_h}{dX}(X(\xi_h)) \right\rangle .$$

6.3. *Numerical Experiment*

The numerical experiment considers the minimization of the previously defined cost function, initialising the optimisation algorithm with the RAE airfoil design. The Mach number is 0.85 .

The mesh of the initial airfoil has 2001 nodes with 86 on the airfoil. The discrete control variables are the ordinates of the nodes on the airfoil, but the nodes at the leading edge and at the trailing edge are fixed. We have then 84 control parameters.

To be sure that our airfoil, defined by ξ_h, remains in S_0, we use box constraints: if ξ_{min} (resp. ξ_{max}) is the vector of the control parameters which defines the "minimum" (resp. "maximum") airfoil, then the box constraints write

$$\xi_{min} \leq \xi_h \leq \xi_{max}.$$

With the area constraint, the total number of constraints is then $1 + 2 \times 84 = 169$.

The CFL number is 1.5 and we use 200 time steps.

On the following figure 8 is the convergence of the cost function. We have a 34.8% reduction of the cost function after 31 optimization iterations.

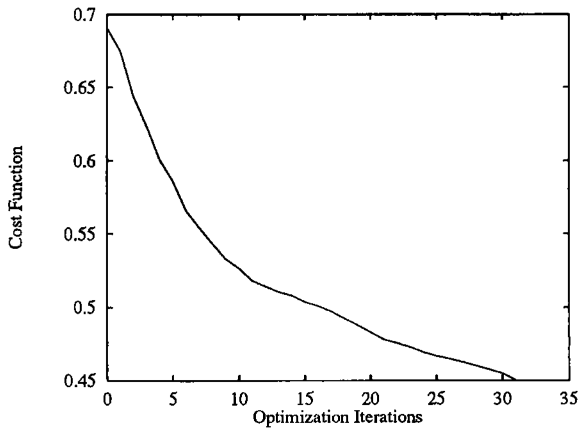

Figure 8: Cost function during optimization.

On the next figure are the intial and optimized airfoils. We can see that due of our choice of parametrisation variables, the optimized airfoil has spurions oscillations. We smooth these oscillations to have a better shape. After smoothing, the cost reduction is of 35.8%.

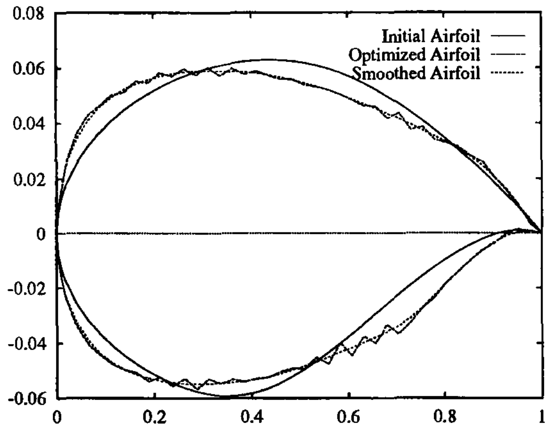

Figure 9: Initial and Optimized Airfoils.

7. Conclusion

This paper has described a second order interior point method well adapted to constrained shape optimal design in fluid dynamics. The theoritical background has been explained and detailed implementation procedures have been given in the case without equality constraints. The algorithm has been successfully applied to many different engineering problems. Here, we have concentrated on two typical CFD problems. The first application to riblets design had very few control parameters, but the calculation of the gradient required a rather detailed variational rewriting of the state equation. The second application used a more straightforward code automatic differentiation procedure to compute gradients, but was more sensitive to the choice of a correct stopping criterion in solving the state and adjoint state equation, and to the choice of an adequate shape parametrisation of the airfoil.

These applications illustrate the need of additional research developments in two main directions :

- parametrisation strategies leading to well posed problems and efficient design,
- use of approximate solutions of the state equation in the optimisation algorithm.

This last point is related to the socalled oneshot methods which nolonger treat the state equation as an implicit definition of the state variables which are then eliminated from the optimisation problem, but consider these equations as equality constraints in the original minimisation problem. It therefore relates to the ongoing effort on the development of interior point algorithms for problems with severe equality constraints.

References

[1] AUATT, S. S., *Nonlinear Programming Algorithms for Elastic Contact Problems*, (in Portuguese), COPPE, Federal University of Rio de Janeiro, Brazil, MS Thesis, 1993.

[2] BARON, F. J. AND PIRONNEAU, O., *Multidisciplinary Optimal Design of a Wing Profile*, Proceedings of STRUCTURAL OPTIMIZATION 93, Rio de Janeiro, Edited by J. Herskovits, 8/1993.

[3] BARON, F. J. *Constrained Shape Optimization of Coupled Problems with Electromagnetic Waves and Fluid Mechanics*, (in Spanish), University of Malaga, Spain, PhD Thesis, 1994.

[4] BARON, F. J., DUFFA, G., CARRERE, F. AND LE TALLEC, P., *Optimisation de forme en aerodynamique*, (in French), CHOCS, Revue scientifique et technique de la Direction des Applications Militaires du CEA, France, 1994.

[5] GILBERT J.-C., LE VEY G. AND MASSE J., *La différentiation automatique de fonctions représentées par des programmes*, INRIA Research Report RR1557, 1991.

[6] HERSKOVITS, J., *A Two-Stage Feasible Directions Algorithm for Non-Linear Constrained Optimization*, Research Report No. 103, INRIA, BP 105, 78153 Le Chesnay CEDEX, France, 1982.

[7] HERSKOVITS, J., *A Two-Stage Feasible Directions Algorithm Including Variable Metric Techniques for Nonlinear Optimization*, Research Report n0 118, INRIA, BP 105, 78153 Le Chesnay CEDEX, France, 1982.

pour

[8] HERSKOVITS, J., *A Two-Stage Feasible Directions Algorithm for Nonlinear Constrained Optimization*, Mathematical Programming, Vol. 36, pp. 19-38, 1986.

[9] HERSKOVITS, J. AND COELHO, C.A.B. *An Interior Point Algorithm for Structural Optimization Problems*, in Computer Aided Optimum Design of Structures: Recent Advances, edited by C.A. Brevia and S.Hernandez, Computational Mechanics Publications, Springer-Verlag, June 1989.

[10] HERSKOVITS, J.,*An Interior Point Technique for Nonlinear Optimization*, Research Report No. 1808, INRIA, BP 105, 78153 Le Chesnay CEDEX, France, 1992.

[11] HERSKOVITS, J.,*An Interior Point Technique for Nonlinear Optimization*, Research Report No. 1808, INRIA, BP 105, 78153 Le Chesnay CEDEX, France, 1992.

[12] HERSKOVITS, J., *A View on Nonlinear Optimization*, Advances in Structural Optimization, Edited by J. Herskovits, Kluwer Academic Publishers, Dordrecht, pp 71-117, 1995.

[13] HIRIART-URRUTY, J. B. AND LEMARÉCHAL, C. *Convex analysis and Minimization Algorithms*, Springer - Verlag, Berlin, Heildelberg, 1993.

[14] MOHAMMADI B., *Fluid dynamics computation with NSC2KE: an user-guide: release 1.0.*, INRIA Technical Report RT0164, 1994.

[15] POWELL M. J. D., *Variable Metric Methods for Constrained Optimization* , in Mathematical Programming - The State of the Art, Edited by A. Bachem, M. Grotschet and B. Korte, Springer-Verlag, Berlin, 1983.

[16] SANTOS, G., *Feasible Directions Interior Point Algorithms for Engineering Optimization* , (in Portuguese), DSc. Dissertation, COPPE - Federal University of Rio de Janeiro, Mechanical Engineering Program, Caixa Postal 68503, 21945-970, Rio de Janeiro, Brazil, 1996.

[17] TITS, A.L., ZHOU, J.L. *A Simple, Quadratically Convergent Interior Point Algorithm for Linear Programming and Convex Quadratic Programming*, Large Scale Optimization: State of the Art, W.W. Hager, D.W. Hearn and P.M. Pardalos Eds., Kluwer Academic Publishers B.V., 1993.

[18] VAUTIER, I., SALAUN, M. AND HERSKOVITS, J., *Application of an Interior Point Algorithm to the Modeling of Unilateral Contact Between Spot-Welded Shells*, Proceedings of STRUCTURAL OPTIMIZATION 93, Rio de Janeiro, Edited by J. Herskovits, 8/1993.

[19] ZOUAIN, N.A., HERSKOVITS J., BORGES, L.A. AND FEIJOO, R. *An Iterative Algorithm for Limit Analysis with Nonlinear Yield Functions*, International Journal on Solids and Structures, Vol 30, No 10, pp 1397-1417, Gt. Britain, 1993.