Comparison of some optimization algorithms for optimum shape design in aerodynamics

Olivier Pironneau, Andreas Vossinis

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1392

Programme 6
Calcul scientifique, Modélisation et
Logiciels numériques

COMPARISON OF SOME OPTIMIZATION ALGORITHMS FOR OPTIMUM SHAPE DESIGN IN AERODYNAMICS

Olivier PIRONNEAU
Andreas VOSSINIS

Février 1991

RR_1392

COMPARISON OF SOME OPTIMIZATION ALGORITHMS FOR OPTIMUM SHAPE DESIGN IN AERODYNAMICS

COMPARAISON DE QUELQUES ALGORITHMES POUR L' OPTIMISATION DE FORMES EN AERODYNAMIQUE

Olivier PIRONNEAU[1]
Andreas VOSSINIS[1,2]

1 INRIA - Rocquencourt - B.P. 105 - 78153 Le Chesnay - FRANCE

2 National Technical University of Athens - 42, Patission Str.
10682 Athens - GREECE

Abstract

Optimum Shape Design Problems in Fluid Mechanics have important industrial applications. Evidently, it is desired that optimization methods are quick and with low memory requirement. The numerical methods used for the solution of optimization problems frequently need the computation of the gradient of the cost function and usually require a linear search method. These computations include the solution of a fluid mechanics problem several times, so they may be time consuming.

Optimization methods which don't need the gradient computation of the cost functional have been developed, but often they are not general. We compare in this work certain Optimization Methods which use the gradient computation and some others that don't need the gradient. We are interested in the behaviour of every method till its convergence and in its convergence speed.

Keywords: Optimization, Fluid Mechanics, Finite Element Method, Optimum Design.

Résumé

Les problèmes d' Optimization de Forme en Mécanique de Fluides ont d' importantes applications industrielles. Evidemment, on veut que les méthodes d' optimisation soient rapides et peu encombrantes en mémoire. Les Méthodes Numériques utilisées pour la solution des problèmes d' optimization ont souvent besoin du calcul du gradient de la fonction–coût et d' habitude demandent une minimisation monodimensionnelle. Ces calculs comprennent la solution d' un problème de la mécanique de fluides de nombreuses fois, donc ils prennent beaucoup de temps.

Des Méthodes d' Optimization qui ne nécessitent pas le calcul du gradient de la fonction–coût ont été développées, mais elles ne sont pas toujours générales.

On compare dans ce travail certaines méthodes d' optimization qui utilisent le calcul du gradient et quelques méthodes qui ne le nécessitent pas. On s' intéresse au comportement de chaque méthode jusqu' à la convergence et à sa vitesse de convergence.

Mots–clefs: Optimization, Mécanique de Fluides, Méthode des Eléments Finis, Design Optimum.

CONTENTS

0. INTRODUCTION

Optimum shape design problems in fluid mechanics require the solution of the state equations many times and this is expensive.

Several methods have been developped and have been found to be efficient. Some of them use the gradient of the cost functional in order to find the direction on which optimization must be done. In this case attention is needed when computing the gradient, because this is usally costly.

We use in this work gradient methods and methods which don't need the gradient computation in order to solve a model problem, namely to find the nozzle which has a certain velocity distribution on its walls under the assumption of a two dimensional, irrotational and incompressible flow and with a given flux. The functional to minimize is a velocity difference norm integral on the unknown boundary. We use the finite element method in order to solve the state equations of the fluid mechanics problem. On the other hand, the finite element method is well adapted to the optimum design problems, since the mesh can well fit the boundary of the domain to optimize. We describe the optimization problem and the solution of the state equation in Chapter I.

The gradient optimization methods used are the steepest descent, which seems to be efficient (see [1], [2] and [7]) in a large category of optimum design problems and GMRES in order to solve the system of equations deduced from the optimality condition: gradient of cost functional = 0. Gradient methods are described in paragraph II.2 of Chapter II. The adjoint state equation is used in order to compute the gradient of the cost function in the application of these methods.

The optimization methods with no gradient computation are Powell's method and least squares method, since the functional to minimize is a sum of squares. This gives the idea to use also GMRES as a direct minimization procedure, since this method solves a system of equations by minimizing the residual norm over a Krylov subspace. The optimization methods with no gradient computation are described in paragraph II.3.

The purpose of this work is to compare these methods at their convergence behaviour and speed. We find the convergence curve and count the CPU time of every method. These are the comparison criteria and we can have an indication for the application of a certain method in more complicated problems. Numerical experiments and results are described in Chapter III.

Finally, we apply the method which behaves better in order to solve a lifting airfoil optimization problem. The problem is described in Chapter IV. The formulation and solution of the state equations and of the optimum design are given. Numerical results and comments are also presented.

I. AN OPTIMIZATION PROBLEM

I.1 DESCRIPTION OF THE PROBLEM

The optimization problem we treat here is the following:

Find the nozzle which realizes a certain velocity distribution on its walls for an incompressile, two-dimensional, irrotational, steady flow

More precisely, if we call:

- $\vec{v} = (v_1, v_2)$ a given velocity distribution we want to realize on the nozzle walls,
- $\vec{u} = (u_1, u_2)$ the velocity distribution of the flow,
- Γ the boundary of the domain,
- $\mid \vec{x} \mid$ the Euclidian norm of $\vec{x}$,

the Optimum Shape Design Problem is:

Find a boundary Γ such that the functional

$$E(\Gamma) = \int_\Gamma \mid \vec{u} - \vec{v} \mid^2 d\Gamma \tag{I.1.1}$$

is minimized.

I.2 THE STATE EQUATION

Consider a two-dimensional flow, which in addition is incompressible and irrotational. We introduce the stream function Ψ such that the velocity at every point of the flow field is:

$$\vec{u} = rot\Psi = (\frac{\partial \Psi}{\partial x_2}, -\frac{\partial \Psi}{\partial x_1}) \tag{I.2.1}$$

so the mass continuity equation $div\vec{u} = 0$ is identically satisfied.
Since $rot\vec{u} = 0$, (irrotational flow) the equation for Ψ is the Laplace equation

$$-\Delta \Psi = 0 \tag{I.2.2}$$

The boundary conditions for Ψ will be: a) the flow velocity at the entrance and the exit of the nozzle is parallel to the nozzle axis (taken as axis x_1, see Figure I.2.1) and uniform

$$\Psi = u_0.x_2 + constant \tag{I.2.3}$$

b) the profile and the axis of the nozzle are streamlines.

Therefore, Ψ is the solution of the following equations:

$$\begin{cases} -\Delta\Psi = 0 \text{ in } \Omega \\ \dfrac{\partial\Psi}{\partial\vec{n}} \mid_{\Gamma_1 \cup \Gamma_3} = 0 \\ \Psi \mid_{\Gamma_2} = 1 \\ \Psi \mid_{\Gamma_4} = 0 \end{cases} \tag{I.2.4}$$

where we suppose that $u_0.L = 1$ (figure I.2.1).

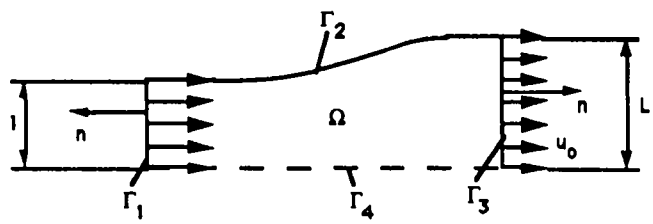

Figure I.2.1

I.3 THE OPTIMIZATION PROBLEM

The function to minimize is given by equation (I.1.1), where Γ is replaced by Γ_2. If we use (I.2.1), we obtain the formula:

$$E(\Gamma_2) = \int_{\Gamma_2} \mid rot\Psi - \vec{v} \mid^2 d\Gamma. \tag{I.3.1}$$

We remark that:

$$\mid rot\Psi - \vec{v} \mid^2 = \mid grad\Psi - \vec{c} \mid^2 \tag{I.3.2}$$

with $\vec{c} = (-v_2, v_1)$.

Using (I.3.1) and (I.3.2), the functional to minimize becomes:

$$(I.3.3) \qquad E(\Gamma_2) = \int_{\Gamma_2} |\, grad\Psi - \vec{c}\,|^2 \, d\Gamma.$$

So, the minimization problem is:

Find the boundary Γ_2 which minimizes (I.3.3), with Ψ solution of (I.2.4).

I.4 VARIATIONAL FORMULATION AND DISCRETIZATION

Multiplying (I.2.3) by a test function $w \in H^1(\Omega)$, integrating on Ω (Fig. I.2.1) and using Green's formula and (I.2.8), we obtain:

$$(I.4.1) \qquad -\int_\Omega \Delta\Psi.wdx = \int_\Omega grad\Psi.gradwdx - \int_{\Gamma_2 \cup \Gamma_4} \frac{\partial\Psi}{\partial\vec{n}} wds = 0.$$

We conclude from (I.4.1) that Ψ is computed solving:
Find Ψ such that

$$(I.4.2) \qquad \int_\Omega grad\Psi.gradwdx = 0, \forall w \in V_0(\Omega)$$

$$\Psi - \Psi_0 \in V_0(\Omega)$$

where

$$V_0(\Omega) = \{\Psi \in H^1(\Omega) : \Psi = 0 \text{ on } \Gamma_2 \cup \Gamma_4\}$$

and Ψ_0 is an extension inside Ω of the boundary condition

$$(I.4.3) \qquad \Psi_0 \,|_{\Gamma_2} = 1$$

We use the Finite Element Method (FEM) to solve (I.4.2). We discretize this equation using P^1 triangular Finite Elements and we approach the solution Ψ by Ψ_h solution of the problem:
Find $\Psi_h \in V_h(\Omega_h)$ that satisfies

$$(I.4.4) \qquad \int_{\Omega_h} grad\Psi_h.gradw_h dx = 0, \forall w_h \in V_{0h}(\Omega_h)$$

where

$$V_h(\Omega_h) = \{w_h \in C^0(\Omega_h) : w_h \,|_{T_j} \in P^1\},$$

$$V_{0h}(\Omega_h) = \{w_h \in C^0(\Omega_h) : w_h \,|_{T_j} \in P^1, w_h = 0 \text{ on } \Gamma_{2h} \cup \Gamma_{4h}\}$$

and $\Psi_h = \Psi_{0h}$ onΓ_{2h}. Ψ_{0h} is a discretized approximation of Ψ_0. We denote by Γ_{2h} and Γ_{4h} the discretized boundaries Γ_2 and Γ_4, respectively and by T_j a triangle of the discretization Ω_h of Ω.

Then, we approach $E(\Gamma_2)$ by $E_h(\Gamma_{2h})$:

$$E_h(\Gamma_{2h}) = \int_{A_h} | grad\Psi_h - \vec{c} |^2 \, dx \tag{I.4.5}$$

where

$$A_h = \{ \text{ Triangles having one edge on } \Gamma_{2h} \}.$$

It is remarkable that we approach (I.3.3) by an integral on the elements adjacent to the discretized boundary, because the discrete gradient of Ψ ($grad\Psi_h$) is piecewise constant (since we use P^1 finite elements). If we call $grad\Psi_{hi}$ the gradient of the solution on the i-th element and $\vec{c}_i = \vec{c}\,|_{\text{i-th element}}$, (I.4.5) gives:

$$E_h(\Gamma_{2h}) = \sum_{T_i \in A_h} | grad\Psi_{hi} - \vec{c}_i |^2 \, . \, | T_i | \tag{I.4.6}$$

I.5 THE DISCRETE FORMULATION OF THE PROBLEM AND ITS CONSTRAINTS

The boundary Γ_{2h} is defined by the line through the points q_i that discretize the boundary Γ_2. The discrete optimization problem is:

(*P.I.5*) Find q_i such that (I.4.6) is minimum with Ψ_h solution of (I.4.4).

The constraints for this problem are the following:

a) q_{i1} is fixed
b) the $q_i \in \Gamma_{2h}$ are such that $0 < q_{i2} < \infty$
c) q_{i2} is constant at the exit.

We remark that constraint a) means that the unknowns of the problem are the ordinates of $q_i \in \Gamma_{2h}$.

II. SOLUTION OF THE PROBLEM

The optimization problem described in I will be solved using gradient methods (Steepest Descent Algorithm and solution of the optimality condition $gradE_h(\Gamma_{2h}) = 0$ with GMRES) and with methods with no gradient computation, namely GMRES, Powell's Method and Least Squares Method.

The linear system (I.4.4) is solved using the Finite Element Method (FEM) and this requires the construction of the Finite Element Mesh of Ω (nozzle longitudinal section).

We describe in the following the construction of the mesh and the optimization methods.

II.1 CONSTRUCTION OF THE MESH

The mesh of the nozzle is constructed by mapping the mesh of a rectangle onto Ω_h.

The nozzle mesh is obtained by dilating the rectangle mesh on the x_2 direction by a ratio $l_i = \frac{q_{i2}-y_1}{y_2-y_1}$ on every line with $x_1 = q_{i1}$ (see Fig. II.1.1). So, the discretization points of the nozzle mesh have the same abscissa as the rectangle mesh nodes and ordinates $q_{k2} = y_1 + l_i \cdot (r_{k2} - y_1)$ where q_k corresponds to r_k on the rectangle. The element and node numbering is the same for the two meshes, only the nodes ordinates change.

The mesh construction is shown on the Figure II.1.1.

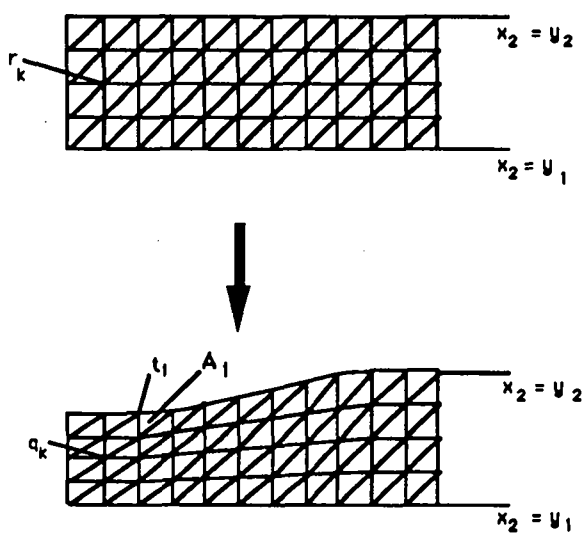

Figure II.1.1

II.2 GRADIENT METHODS

It is known that for a continuously differentiable function $E(z)$ with $z \in R^N$ a necessary condition for having a minimum is that $grad_z E(z) = 0$. Gradient methods try to satisfy this condition.

Such methods can be used if we consider the functional given by (I.4.6) as a function of $z \in R^N$ where

$$(II.2.1) \qquad z = \begin{pmatrix} q_{12} \\ q_{22} \\ \vdots \\ q_{N2} \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

and $q_i \in \Gamma_{2h}$.

II.2.1 COMPUTATION OF THE DISCRETE GRADIENT OF E_h

The computation of $gradE_h$ must be done cheaply, that is by computing the function Ψ_h the least number of times.

We perform this computation in its discrete form. We need for this the two following propositions:

Proposition II.2.1.1. *Assume that f is a continuously differentiable function on T_j. If T_j' is obtained from T_j by moving a vertex q_k into $q_k + \delta q_k$, then*

$$\int_{T_j' \setminus T_j} f dx - \int_{T_j \setminus T_j'} f dx = \int_{T_j} \delta q_k . \nabla (f w_k) dx + o(\mid \delta q_k \mid),$$

where w_k is the basis function associated with the vertex q_k.

Proposition II.2.1.2. *Let T_j' be the element obtained from T_j by moving a vertex $q_k \in T_j$ into $q_k + \delta q_k$. If w_i is the basis function associated with the vertex $q_i \in T_j$ and w_i' is the same basis function when q_k is replaced by $q_k + \delta q_k$, then*

$$w_i'(x) - w_i(x) = -w_k(x) \nabla w_i(q_k) . \delta q_k + o(\mid \delta q_k \mid), \ \forall x \in T_j \cap T_j', \forall q_i, q_k \text{ vertices of } T_j$$

where w_k is the basis function associated with the vertex q_k.

These propositions are found in [1].

We prove in Appendix A that if we denote by $q_{kl}, l = 1, 2$ the x_1 and x_2 coordinates of q_k, we have from (A.24)

$$(II.2.1.1) \qquad \begin{aligned} \frac{\partial E_h}{\partial q_{kl}} = & \int_{\Omega_h} [(\nabla w_k . \nabla P_h) \frac{\partial \Psi_h}{\partial x_l}(q_k) \\ & + (\nabla \Psi_h . \nabla w_k) \frac{\partial P_h}{\partial x_l}(q_k) - (\nabla \Psi_h . \nabla P_h) \frac{\partial w_k}{\partial x_l}] dx \\ & - 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \nabla w_k \frac{\partial \Psi_h}{\partial x_l}(q_k) dx \\ & + \int_{A_h} \mid \nabla \Psi_h - \vec{c} \mid^2 \frac{\partial w_k}{\partial x_l} dx, \end{aligned}$$

where P_h is the solution of the adjoint equation (A.16).

We have found so far the $gradE_h$ with respect to the coordinates of the nodes of Ω_h as if these nodes moved independently. In fact, only the nodes on Γ_{2h} move independently and the other nodes follow this motion. So, we have to find the $gradE_h$ with respect to the coordinates of these boundary nodes. In addition, nodes on Γ_{2h} move only on the x_2 direction and the optimization parameters are the $\{t_i, i = 1, 2, \ldots, N\}$ of eq. (II.2.1). Therefore, we are interested in

$$(II.2.1.2) \qquad \frac{\partial E_h}{\partial t_i} = \sum_{k,l} \frac{\partial E_h}{\partial q_{kl}} \cdot \frac{\partial q_{kl}}{\partial t_i} = \sum_{k} \frac{\partial E_h}{\partial q_{k2}} \cdot \frac{\partial q_{k2}}{\partial t_i}$$

since

$$\frac{\partial E_h}{\partial q_{k1}} = 0.$$

In order to compute $\frac{\partial q_{k2}}{\partial t_i}$ we use the fact that the nozzle mesh is constructed by dilating the mesh of a rectangle, as described in II.1. Supposing that the edges of the rectangle parallel to x_1 direction are $x_2 = y_1$ and $x_2 = y_2$ (see Fig. II.1.1) and denoting by $r_{kl}, l = 1, 2$ the coordinates of the node in the rectangle mesh which corresponds to q_k we have

$$(II.2.1.3) \qquad q_{k2} = y_1 + l_i.(r_{k2} - y_1), \forall q_k \text{ such that } q_{k1} = q_{i1}$$

where $l_i = \frac{t_i - y_1}{y_2 - y_1}$ the dilation ratio on the line $x_1 = q_{i1}$. From eq. (II.2.1.3) we see that

$$(II.2.1.4) \qquad \frac{\partial q_{k2}}{\partial t_i} = \begin{cases} \frac{r_{k2} - y1}{y_2 - y_1} & \text{if } q_{k1} = q_{i1} \\ 0 & \text{otherwise.} \end{cases}$$

So, we have proved the

Proposition II.2.1.3. *The gradient of E_h is given by eq. (II.2.1.1), (II.2.1.2) and (II.2.1.4).*

We have to note that the computation of $gradE_h$ by the use of the adjoint state equation is rather complicated. An alternative is to compute $gradE_h$ by the finite difference method. More precisely, we may consider a δt_i for every t_i and to compute the solution Ψ'_h in the new domain Ω'_h, where t_i is replaced by $t_i + \delta t_i$. Then, we compute

$$\frac{\partial E_h}{\partial t_i} = \frac{E'_h - E_h}{\delta t_i},$$

where E_h, E'_h are given by eq. (I.4.6) for Ψ_h, A_h and Ψ'_h, A'_h, respectively. The computation is simpler, but much more expensive since we must solve the state equation N times and not only one as we do by using the adjoint state equation.

II.2.2 STEEPEST DESCENT METHOD

Suppose we want to find

$$\min_{z \in R^N} E(z).$$

From the relation

$$(II.2.2.1) \qquad E(z+\delta z) = E(z) + gradE(z).\delta z + o(\mid \delta z \mid) \ , \forall \delta z \in R^N,$$

we obtain for $\delta z = -\rho gradE(z)$ with ρ a small positive number:

$$(II.2.2.2) \qquad E(z+\delta z) = E(z) - \rho \mid gradE(z) \mid^2 + o(\rho \mid gradE(z) \mid).$$

Since $\rho \mid gradE(z) \mid^2 > 0$ and $o(\rho \mid gradE(z) \mid)$ can become very small, we see that using (II.2.2.2) we can make $E(z+\delta z) < E(z)$.

So, we conclude the following algorithm for minimizing $E(z)$:

Algorithm II.2.2.1

- Choose $z^0 \in R^N$, $\rho > 0$ and $\varepsilon \in R$
- For $k = 0, 1, \ldots$ (till satisfied) do
  Compute $h^k = -gradE(z^k)$
  Compute $\rho^k = argminE(z^k + \rho h^k)$ with $\rho > 0$
  Set $z^{k+1} = z^k + \rho^k h^k$
- If $\mid gradE(z^{k+1}) \mid \leq \varepsilon$
  stop
- else $k = k + 1$.

Theorem II.2.2.1. *If E is continuously differentiable and bounded from below, then all accumulation points $\hat{z}$ of the sequence z^k constructed by algorithm II.2.2.1 satisfy $gradE(\hat{z}) = 0$.*

A proof of this theorem can be found in [1].

II.2.3 GENERALIZED MINIMAL RESIDUAL ALGORITHM (GMRES)

GMRES algorithm has initially been developed for solving a linear system $Az = f$ by minimizing the residual norm over a Krylov subspace when the matrix A is not positive definite. We remind that when A is a square matrix and v a nonzero vector, the $m - th$ Krylov subspace associated with A and v is defined by $K_m \equiv span\{v, Av, A^2v, \ldots, A^{m-1}v\}$. A more generalized form of GMRES is for solving a non-linear system of equations $F(z) = 0$, with $z, F \in R^N$. GMRES algorithm is the following

Algorithm II.2.3.1

- Choose $z^0 \in R^N$
- For $k = 0, 1, \ldots$ do

  Construct an orthonormal basis $\{p_j\}_{j=1}^m$ of a Krylov subspace

  with $p_1 = -F(z^k)/ \mid F(z^k) \mid$ and whose dimension is $m << N$

  Form the new solution

  $$z^{k+1} = z^k + \sum_{j=1}^{m} \alpha_j p_j$$

  where the $\alpha_j \in R$ are chosen to minimize $\mid F(z^{k+1}) \mid^2$
- If the solution is satisfactory

  stop
- else $k = k + 1$.

The construction of the Krylov subspace orthonormal basis is done according to Arnoldi's orthogonalization method, as follows

Algorithm II.2.3.2

- Set $\hat{p}_1 = -F(z^k)$ and $p_1 = \hat{p}_1 / \mid \hat{p}_1 \mid$
- For $j = 1, 2, \ldots, m-1$ do

  compute

  $$\hat{p}_{j+1} = \bar{F}(z^k; p_j) - \sum_{i=1}^{j} b_{ij} p_i$$

  where $b_{ij} = \bar{F}(z^k; p_j).p_i$

  and

(*II*.2.3.1) $$\bar{F}(z^k;p_j)=\lim_{\varepsilon\to 0}\frac{F(z^k+\varepsilon p_j)-F(z^k)}{\varepsilon}$$

- normalize $p_{j+1}=\hat{p}_{j+1}/\mid \hat{p}_{j+1}\mid$
- end.

We remark that $\bar{F}(z^k;p_j)$ is the action of the $gradF(z_k)$ to p_j and the vectors p_j are such that $p_j.p_i=0,\forall\, i=1,2,\ldots,j-1$. In Algorithm II.2.3.1 the minimization of $\mid F(z^{k+1})\mid^2$ is done by solving a least squares problem obtained writing approximatively

(*II*.2.3.2) $$\mid F(z^{k+1})\mid^2 \simeq \mid F(z^k)+\sum_{j=1}^{m}\alpha_j\bar{F}(z^k;p_j)\mid^2 .$$

If this minimization is not satisfactory, an "exact" minimization of $\mid F(z^{k+1})\mid^2$ is performed, without making the approximation (II.2.3.2). The criteria according to which a solution is satisfactory are the following

a) $\mid F(z^k)\mid < \varepsilon_1 \mid F(z^0)\mid$
or
b)the norm of the projected $gradF$ in the Krylov subspace is less than ε_2 i.e.

$$\mid F(z^k)\mid .\sqrt{\sum_{j=1}^{m}(\bar{F}(z^k;p_j).F(z^k))} < \varepsilon_2$$

or
c) the relative difference norm $\mid z^{k+1}-z^k\mid /\mid z^k\mid$ between two successive solutions is less than ε_3
where $\varepsilon_1,\varepsilon_2,\varepsilon_3$ are three tolerances given to the algorithm.
GMRES has certain important properties which make it fast and robust. We summarize them in the following Propositions:

Proposition II.2.3.1. *The solution z^k of $F(z)=0$ found by Algorithm II.2.3.1 in the step k is exact if and only if the following equivalent conditions hold:*

- *The algorithm breaks down at step k*
- $\hat{p}_{k+1}=0$
- *The degree of the minimal polynomial of the initial residual vector $-F(z^0)$ is equal to k.*

Proposition II.2.3.2. *The residual norm of the approximate solution z^k is equal to the $(k+1)$st component of the right-hand side of the linear system corresponding to the Least Squares Problem obtained by the minimization of eq. (II.2.3.2).*

The prooves of these propositions are in [5].

One more advantage of GMRES is that only the action of $gradF$ to a vector is needed, not the computation of $gradF$ and this can be done easily using finite differences according to eq. (II.2.3.1).

We remind that for the minimization problem (P.I.5) we use GMRES to solve the system derived from the optimality condition $gradE_h$ that is we put $F = gradE_h$.

II.3 METHODS WITH NO GRADIENT COMPUTATION

The computation of $gradE_h$ and of the optimal step size in gradient methods are usually difficult to program. Therefore, we propose certain optimization methods which do not need the gradient of the functional to minimize.

These methods are described in the following.

II.3.1 GMRES AS A MINIMIZATION TOOL

Assume that we have a function

$$F: R^N \longmapsto R^N \qquad (II.3.1.1)$$

whose norm is to be minimized, that is we have the optimization problem

$$\text{Find } \min_{z \in R^N} \mid F(z) \mid^2 . \qquad (P.II.3.1.1)$$

In order to solve this problem, we can use GMRES as described in Algorithm II.2.3.1 substituting the vector F of the non-linear system of equations by the function given of eq. (II.3.1.1). Then, $\mid F(z) \mid^2$ is minimized over the Krylov subspace constructed in every step of GMRES algorithm considered as a function of the scalars α_j that project the solution on the subspace.

We are able to use GMRES in this way if we construct a vector $F \in R^N$ whose i-th component is

$$F_i = \sqrt{\mid \nabla \Psi_{hi} - \vec{c}_i \mid^2 . \mid T_i \mid} \qquad (II.3.1.2)$$

so $\mid F \mid^2 = E_h$.

Evidently, using GMRES with no gradient computation has the same advantages mentioned in II.2.3. We would like however to remark that the linearized solution obtained by minimizing the approximated $\mid F(z) \mid^2$ given by eq. (II.2.3.2) is usually sufficient when $\min \mid F(z) \mid^2 = 0$ but this minimization isn't sufficient otherwise – that is $\mid F(z) \mid^2$ must be minimized exactly and it is usually costly. This is due to the fact that GMRES solves a system $F(z) = 0$, by minimizing $\mid F(z) \mid^2$ on a Krylov subspace even if $\mid F(z) \mid^2 \neq 0$. In this case we cannot write that $F(z) = 0$.

II.3.2 POWELL' S METHOD

We consider a function $f: R^N \longmapsto R$ and we want to find

$$\min_{z \in R^N} f(z). \tag{P.II.3.2.1}$$

The outline of Powell's method for solving (P.II.3.2.1) is the following:

- Choose a point $z \in R^N$ and a direction $x \in R^N$
- Find $\min_{\lambda \in R} f(z + \lambda x)$ and set $z = z + \lambda x$
- Choose a new direction $x \in R^N$ and solve the previous minimization problem.

The idea of the method looks simple, since it deals with one dimensional minimization. All methods using the above described principle of one dimensional minimization choose the directions according to one of the following criteria:

a) the used directions must be taken parallel to the eigenvector corresponding to the largest eigenvalue of f

or

b) they must be such that minimization along one of them must not affect minimization along the other directions. Most methods satisfy condition b), which implies

$$\vec{u}.H.\vec{v} = 0 \tag{II.3.2.1}$$

for two different directions $\vec{u}$ and $\vec{v}$ passing through point z. H denotes the Hessian matrix of f at point z, that is the matrix whose elements are

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(z). \tag{II.3.2.2}$$

We can prove eq. (II.3.2.1) taking the Taylor's expansion of $f(z)$ considering z as the origin of a coordinates system $\{x_i, i = 1, \ldots, N\}$

$$\begin{aligned} f(x) &= f(z) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + o(\mid \delta x \mid^2) \\ &\simeq c - g.x + \frac{1}{2} x.H.x \end{aligned} \tag{II.3.2.3}$$

where $c = f(z), g = -\nabla f(z)$. We see from eq. (II.3.2.3) that $\nabla f(x) = H.x - g$ and

$$\delta(\nabla f) = H.\delta x. \tag{II.3.2.4}$$

Condition b) means that $\delta(\nabla f)$ must stay perpendicular to the direction $\vec{u}$ when minimization along direction $\vec{v}$ is performed. According to eq. (II.3.2.4) this changement equals to $H.\vec{v}$ and since it must be perpendicular to $\vec{u}$, eq. (II.3.2.1) must be satisifed. Directions satisfying eq. (II.3.2.1) are called conjugate directions.

Powell's method constructs a set of mutually conjugate directions (directions that satisfy eq. (II.3.2.1) pairwise) and minimizes f along them:

Algorithm II.3.2.1

- Initialize: $\{\vec{u}_i = \vec{e}_i, i = 1, \ldots, N\}$

  where $\vec{e}_i$ are the unit vectors on the N directions

(1) • Choose the initial point z^0

- For $i = 1, \ldots, N$ do

  Find

  $\min_{\lambda \in R} f(z^{i-1} + \lambda \vec{u}_i)$

  Set $z_i = z_{i-1} + \lambda \vec{u}_i$ and $\vec{u}_i = \lambda \vec{u}_i$

- end
- For $i = 1, \ldots, N-1$ do

  Set $\vec{u}_i = \vec{u}_{i+1}$

- end
- Set $\vec{u}_N = z^N - z^0$
- Find $\min_{\lambda \in R} f(z^N + \lambda \vec{u}_N)$
- Set $z^0 = z^N + \lambda \vec{u}_N$ and $\vec{u}_N = \lambda \vec{u}_N$
- If $\min_z f(z)$ is found

  stop

- else go to (1)

Powell proved (cf [8]) the following

Proposition II.3.2.1. *Assume that $f: R^N \longmapsto R$ is a quadratic function. Then, m iterations of Algorithm II.3.2.1 produce a set of directions $\vec{u}_i$ whose last m members are mutually conjugate.*

We deduce the

Corollary II.3.2.1 *N iterations of Algorithm II.3.2.1 (that is $N.(N+1)$ one dimensional minimizations) take exactly to the minimum of a quadratic form.*

Powell's method has the advantage to converge quadratically. Algorithm II.3.2.1 however has the disadvantage to tend producing linearly dependent vectors because $\vec{u}_N$ is replaced by $\vec{u}_N = z^N - z^0$, so the directions used are not mutually conjugate any more. In this case the minimum of $f(z)$ is found over a subspace of R^N. This problem is handled by trying to find some "good" directions of descent instead of N mutually conjugate directions, but the quadratic convergence is spoiled. Effectively, the fundamental procedure is not changed, but we replace by $z^N - z^0$ the old direction on which f decreased the most. It is done because the largest decrease direction is likely to be a major component of the direction we add. So, replacing it avoids a linear dependence.

However, there are some exceptions at this rule. The previous set of directions is kept if one of the following conditions is satisfied (denoting by $f_0 = f(z^0), f_N = f(z^N), f_E = f(2z^N - z^0), \delta f$ = absolute value of the largest decrease of f)

a) $f_E \geq f_0$ because $z^N - z^0$ is not satisfactory

b) $2(f_0 - 2f_N + f_E)[(f_0 - f_N) - \delta f]^2 \geq (f_0 - f_E)^2 \delta f$ because the decrease on $z^N - z^0$ is not due to decrease on a direction or we have the indication - by the second derivatives - that we are at the minimum on the direction $z^N - z^0$.

The modified algorithm is used to solve (P.I.5) when f is given by eq. (I.4.6). The one dimensional minimization on every direction is performed using Brent's method. Brent's method minimizes a single variable function using the golden section search (see [3]) when the function is not smooth or a parabolic extrapolation otherwise (observing that a sufficiently smooth function is almost parabolic around its minimum). A detailed description of Brent's method and of golden section search is given in [3].

II.3.3 LEAST SQUARES METHOD

We observe that the functional E_h to minimize, given by eq. (I.4.6), is a sum of squares. Therefore, it can be minimized by solving directly a Least Squares Problem. We describe the Least Squares Problem in its general form:

(P.II.3.3) given a matrix A of dimensions $M \times N (M > N)$ and a vector b with M components, find a vector z of N components which minimizes $\mid Az - b \mid^2$.

If the chosen norm is the euclidian norm, the solution of problem (P.II.3.3) verifies a linear system of equations according to the following theorem:

Theorem II.3.3.1 *The vector $z \in R^N$ solves (P.II.3.3) if and only if it is solution of the linear system:*

$$A^t A z = A^t b. \qquad (II.3.3.1)$$

This system has always at least one solution and if $A^t A$ is non-singular the solution is unique.

We refer to [9] for a proof of this theorem.

The system (II.3.3.1) is solved by applying a $Q \cdot R$ factorization of the matrix A where Q is orthogonal with dimensions $M \times N$ and R superior triangular of dimension N. If we

write the equality $A = Q \cdot R$ by columns, we have for the $i-th$ column of A:

$$(II.3.3.2) \qquad a_i = \sum_{j=1}^{N} q_j r_{j,i} = \sum_{j=1}^{i} r_{j,i} q_j \Longrightarrow r_{i,i} q_i = a_i - \sum_{j=1}^{i-1} r_{j,i} q_j,$$

where q_j are the columns of Q. We verify from eq. (II.3.3.2) that the vector q_j is a linear combination of $a_1, a_2, \ldots, a_j$, therefore the vectors q_j may be constructed by the Gram-Schmidt orthonormalization procedure (see Algorithm II.2.3.2 for a more general case of a non-linear system).

Thus, the matrix $A^t A$ is written:

$$(II.3.3.3) \qquad A^t A = R^t Q^t Q R = R^t R = \bar{R}^t \bar{R}$$

where $\bar{R}$ is the non-zero part of R. This factorization of $A^t A$ is similar to a Cholesky factorization. We can obtain a factorization of A by the applying Givens' transformation, that is by multiplying succesively A by matrices of elementary plane rotations. We refer to [9] for a detailed description of Givens' transformation and for the programming of the factorization. We note that the factorization of A gives immediately the solution of the Least Squares Problem, because:

$$(II.3.3.4) \qquad \mid Ax - b \mid^2 = \mid QRx - b \mid^2 = \mid Rx - Q^t b \mid^2 = \mid \bar{R}x - c_1 \mid^2 + \mid c_2 \mid^2,$$

so any vector x verifying $\bar{R}x = c_1$ realizes the minimum of $\mid Ax - b \mid^2$. We remark that the solution is unique when $deg(A) = N$ and is easily obtained since $\bar{R}$ is upper triangular. If $deg(A) < N$ there is an infinity of solutions.In this case, the solution of minimal norm is proved to be again the solution of $\bar{R}x = c_1$.

The optimization problem (P.I.5) has been solved using an appropriate subroutine containing the Least Squares Method of Harwell Computer Library, considering E_h as the euclidian norm of a function $F: R^N \longmapsto R^N$, given by (II.3.1.2) which is sum of a squares.

III. NUMERICAL EXPERIMENTS

III.1 PROBLEMS TREATED

The optimization problem (P.I.5) has been solved in two cases:

a) Inverse Problem

The velocity distribution $\vec{c}$ on the boundary Γ_{2h} is produced by a known nozzle, i.e. having a given nozzle we solve the state equations (I.2.4) on the correspoding Ω and we derive the velocity distribution on Γ_{2h}. In this case $\min E_h = 0$.

The different optimization methods are tested to find the known boundary and we are interested in the convergence speed of every method.

b) Real Problem

The velocity distribution $\vec{c}$ is arbitrary. The nozzle which realizes it is not known. In this case we cannot assert that $\min E_h = 0$.

The methods are tested to find a nozzle that has this velocity distirbution on its profile and we are also interested in the convergence speed of every method.

III.2 RESULTS

For every method used we have counted the number of solutions of the state equations (solution of $-\Delta\Psi = 0$), which is the main computing cost, and the corresponding functional value. The CPU time is counted too.

For all tests, the nozzle section (Ω) has been discretized by 250 elements and 156 nodes and the unknown boundary Γ_{2h} has been discretized by 26 points. For Powell's method only 13 optimization parameters have been used corresponding to the odd numbered vertical coordinates of the discretization points on Γ_{2h}, because with 25 optimization parameters this method needs an excessively big CPU time.

In order to see how the convergence and the CPU time of every method varies when the number of the optimization points varies, we have done the same tests with two more values of optimization parameters: 51 discretization points on Γ_{2h} (which implies 50 optimization points), 900 elements and 510 mesh nodes; 15 discretization points on Γ_{2h}, that is 14 optimization parameters, 90 elements and 140 mesh nodes. For Powell's method only the latter test has been done, as it would be useless to perform the test with 50 optimization parameters.

The following figures show the results.

III.3 Comments and Conclusion

Diagrams III.3.1, III.3.6 and III.3.9 show the convergence curves for an Inverse Problem and that GMRES has a fast convergence in both cases of use: without gradient computation and used for the solution of the optimality condition. This is due to the fact that GMRES converges quickly when it is used to solve a system $F(z) = 0$. Effectively, this is done in both cases of use and is expected to work well since $\min E_h = 0$.

The Steepest Descent Method has also worked quite well, especially in the beginning of the optimization procedure. This result was expected too, as it is known from numerical experiments (cf [1] and [7]) that this method is efficient when the Adjoint State Equation is used to compute the gradient of the cost functional.

The Least Squares Method has a rather slow convergence, mainly in the beginning of the procedure and it converges abruptly after a certain number of iterations. We see the expected form of the convergence of this method.

Powell's method is slow even for a small number of optimization paramaters and this can be forseen by the application of Corollary II.3.2.1. Diagram III.3.2 shows the convergence of Powell's method for an Inverse Problem and a Real Problem when 25 optimization parameters are used. Clearly, the method is too expensive, that's why it has not been tested in the two other cases of discrtization.

The profile convergence shows that the prescribed boundary is well found by all methods. Certain methods, however, give oscillating profiles during the optimization procedure and mainly at the beginning (for example Steepest Descent Powell's method). We remark that GMRES and Least Squares Method give practically not oscillating profiles. We show in Diagram III.3.3 the boundary convergence for GMRES.

For the Real Problem, we see at Diagrams III.3.4, III.3.7 and III.3.10 that the gradient methods (Steepest Descent and Gradient-GMRES) find practically the same minimal value of the cost functional. Powell's method converges to the same value too. GMRES with no gradient computation and Least Squares Method converge rather slowly to a higher value (for GMRES is almost twice the value found by the gradient methods).

We remark the expected convergence of Steepest Descent and a good convergence of Gradient-GMRES. We note the fast convergence of Gradient-GMRES, especially when the number of optimization points increases. This result is expected, because of the efficiency of GMRES in solving fast systems of equations. GMRES with no gradient computation has a rather slow convergence, because the algorithm still solves a system $F(z) = 0$ (we remind that F has the components given by eq. (II.3.1.2)), so $\mid F(z) \mid$ and $\min \mid F(z) \mid$ are forced to be zero at the solution z, while $\min E_h = \min \mid F(z) \mid^2 \neq 0$.

The Least Squares Method is slow again while Powell's method is better than in the inverse problem but still slow compared with the Steepest Descent and GMRES in its both cases of use.

The converged boundaries obtained by every method (Diagarams III.3.5, III.3.8 and III.3.11) show that all methods, except the Least Squares Method, converge to the same profile, that is they find practically the same minimum of the cost functional.

The Least Squares Method gives an oscillating final boundary and this is rather bad, because it implies that we have to use constraints and smoothing of the boundary (not use all the boundary discretization points as optimization points). We remind that boundary smoothing has been used for Powell's method.

Finally, we can observe the efficiency of every method, regarding its computing cost, from Diagrams III.3.12 and III.3.13. Diagram III.3.12 shows the number of solutions of the State Equation as a function of the optimization points for an Inverse Problem. We deduce that for GMRES in both cases of use , the number of solutions of the State Equation increases a little when the number of optimization parameters increases. More precisely, when the number of optimization points is rather little, GMRES needs almost the same number of Laplacian solutions till the convergence. If the number of optimization points is high the computing cost increases remarkably when it is used with no gradient computation: when we double the optimization parameters the cost is more than double. We don't have the same effect however when Gradient-GMRES is used: doubling the optimization points increases the cost less than the double. This is also due to the efficiency of GMRES when the number of unknowns is big.

We can see the same effect in the Steepest Descent when the number of parameters is high, but we can also observe its sensitivity when the number of optimization parameters is low: the computing cost is more than double when we don't even double the optimization points. We can explain this fact with the observation that the solution of the State Equation and the Gradient of the Cost Function are better approximated when we use a better discretization.

Least Squares Method has almost the same behaviour as GMRES with no gradient computation: the computing cost doesn't increase proportionally to the number of the optimization points if this is low but it increases proportionally when we increase the optimization points to higher numbers.

In the case of a Real Problem, we see that Steepest Descent and Gradient-GMRES have the same behaviour as for the Inverse Problem. This is due to the fact that these methods work in the same way in both cases.

GMRES with no gradient computation changes its effects, especially when the number of optimization parameters is high. GMRES doesn't increase proportionally the computing cost. Least Squares Method increases it rather much.

We can conclude that the method of Gradient-GMRES is an efficient one in both kinds of problems. If, however, we have to treat a problem where the minimum is equal to zero, we can use GMRES with no gradient computation. In addition, in this case we have the advantage of not computing the gradient, which is complicated as we have seen.

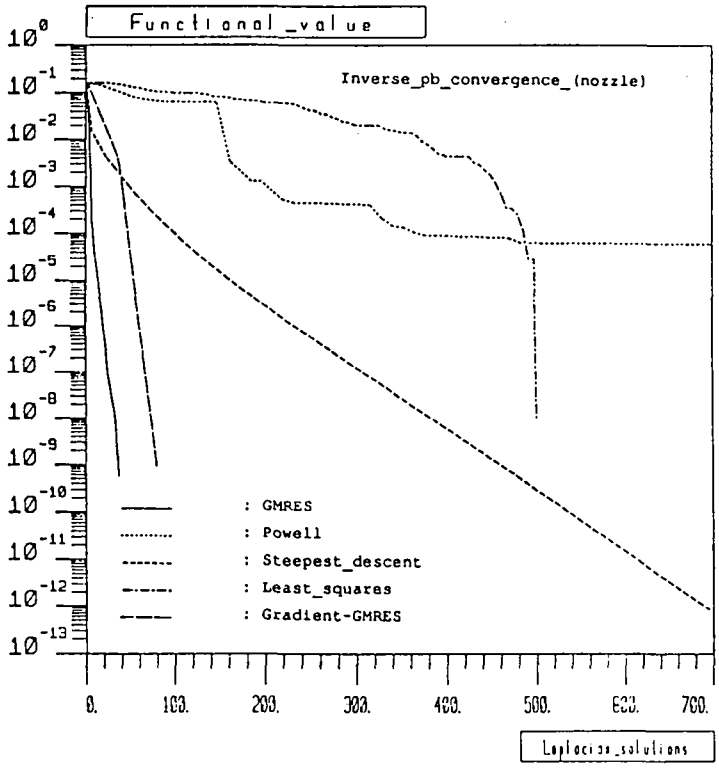

Diagram III.3.1: Nozzle optimization with 25 optimization parameters. Convergence comparison for an inverse problem.

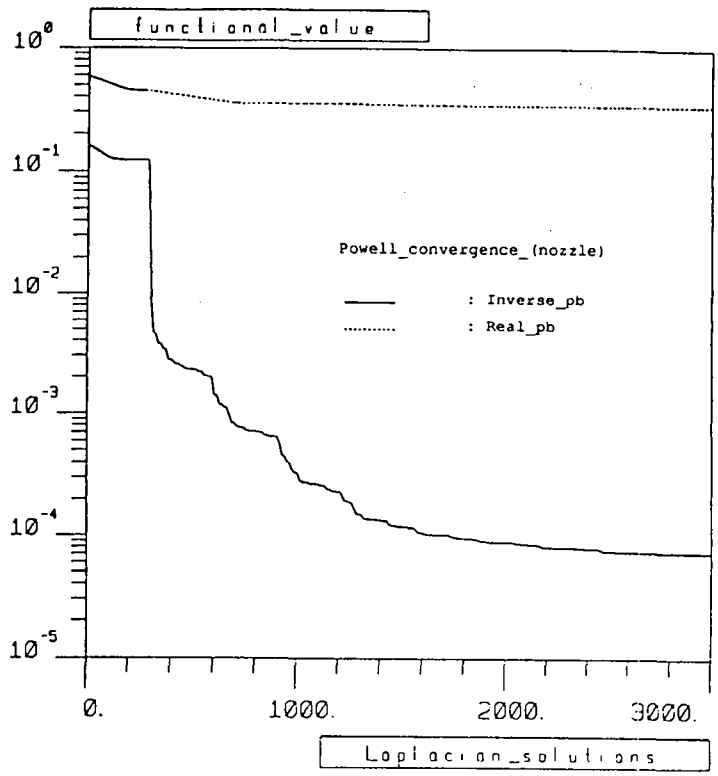

Diagram III.3.2: Nozzle optimization with 25 optimization parameters using Powell's method for an inverse and a real problem.

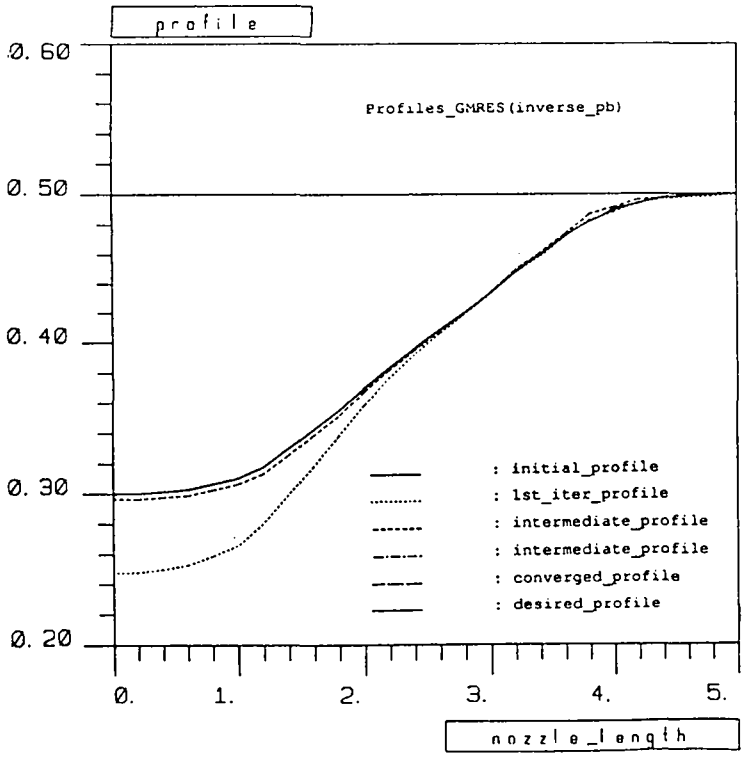

Diagram III.3.3: Nozzle optimization with 25 optimization parameters. Boundary convergence for GMRES.

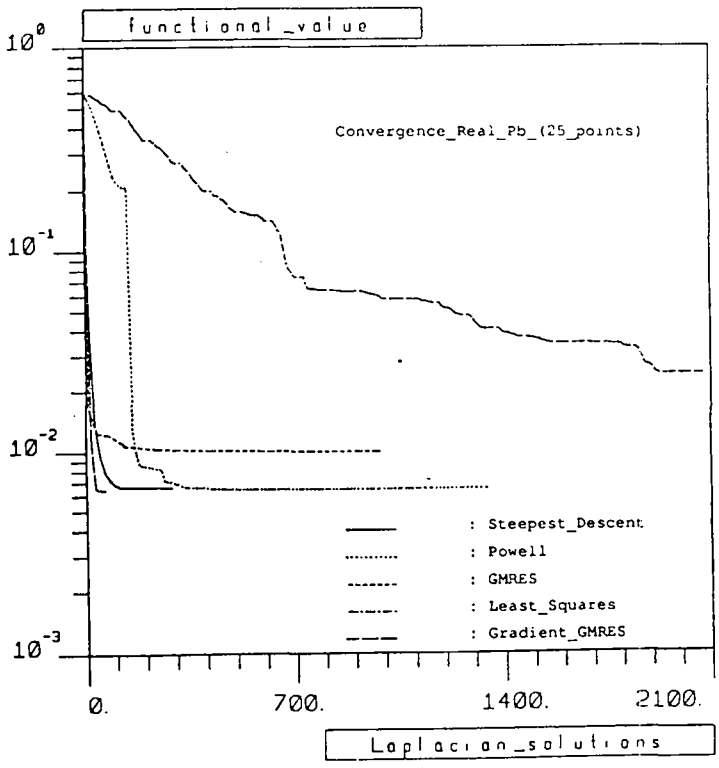

Diagram III.3.4: Nozzle optimization with 25 optimization parameters. Convergence comparison for a real problem.

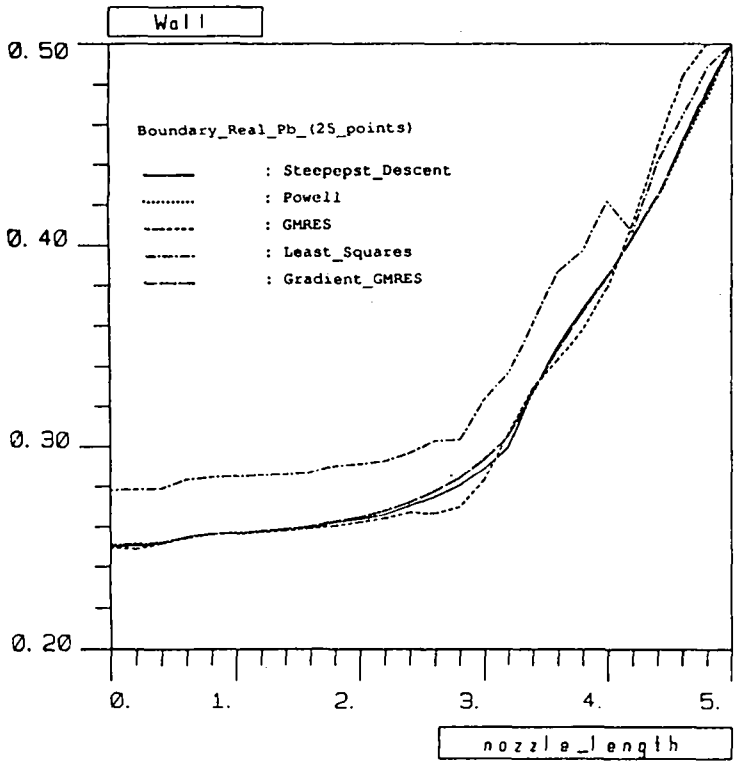

Diagram III.3.5: Nozzle optimization with 25 optimization parameters. Converged boundaries for a real problem.

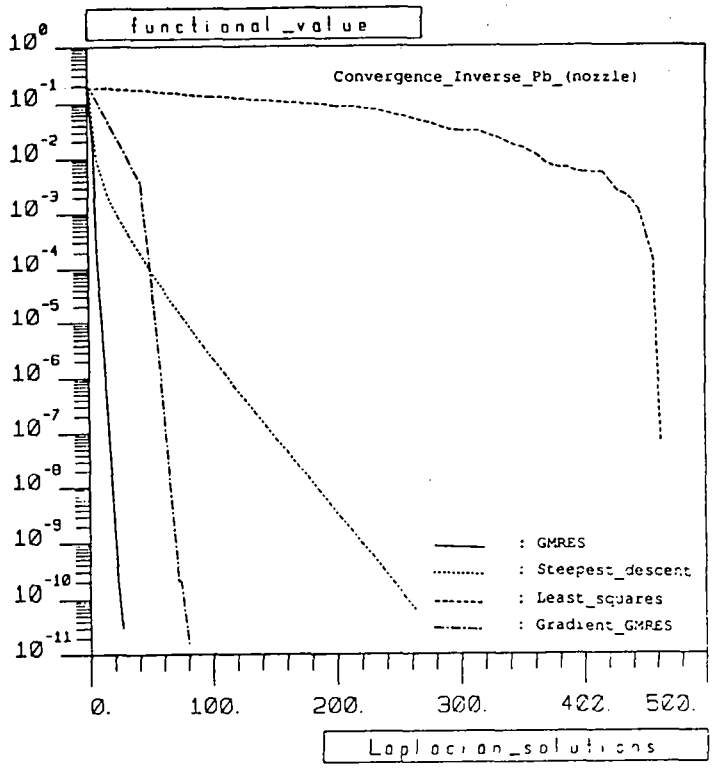

Diagram III.3.6: Nozzle optimization with 14 optimization parameters. Convergence comparison for an inverse problem.

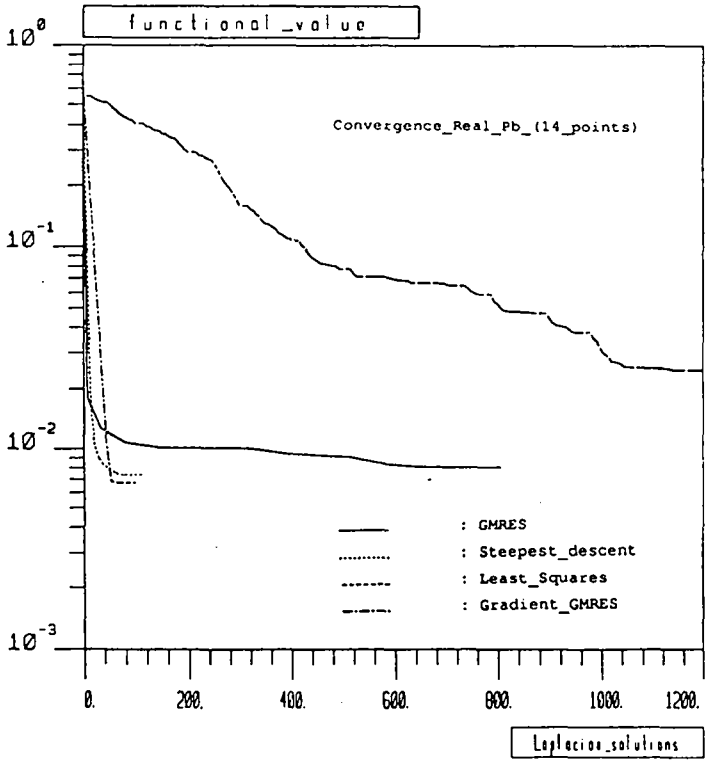

Diagram III.3.7: Nozzle optimization with 14 optimization parameters. Convergence comparison for a real problem.

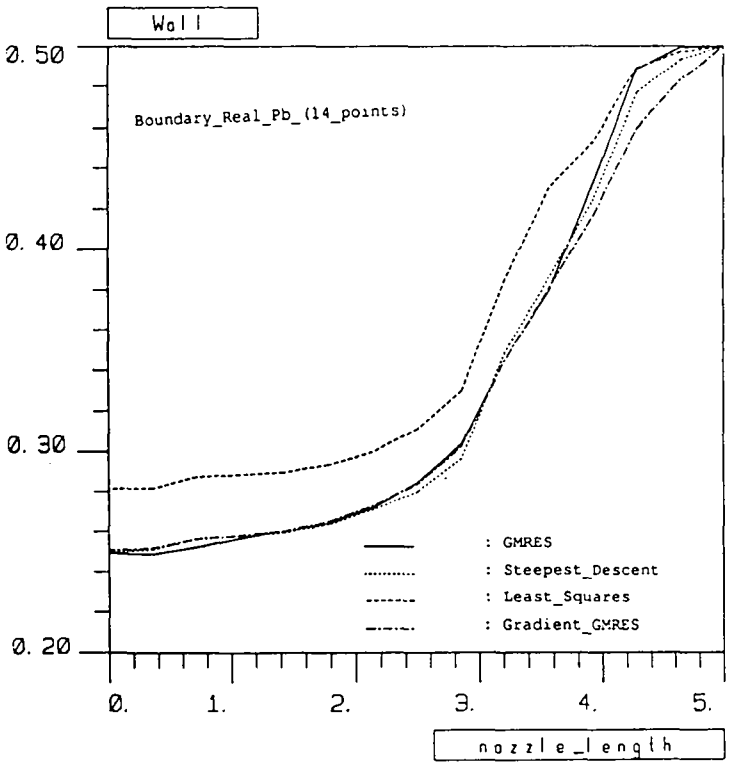

Diagram III.3.8: Nozzle optimization with 14 optimization parameters. Converged boundaries for a real problem.

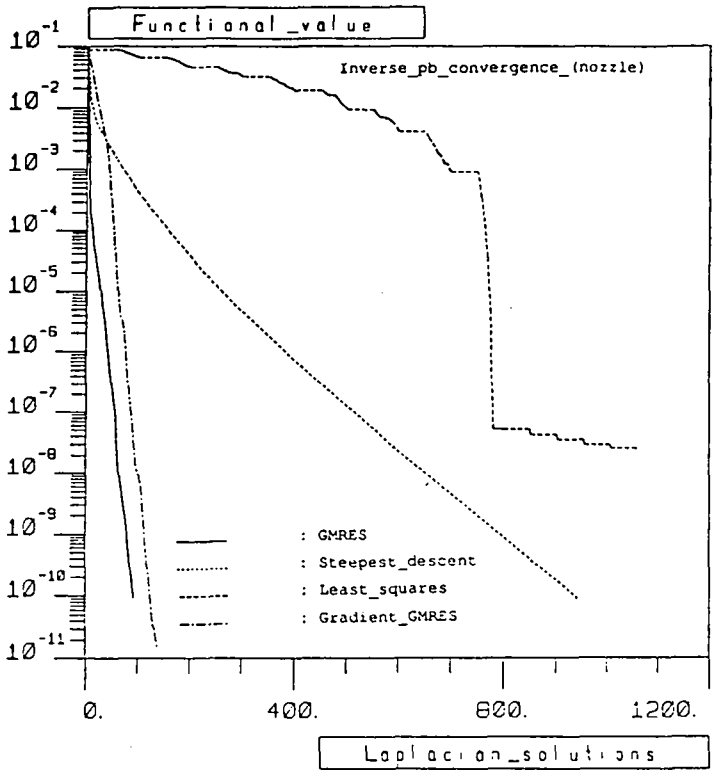

Diagram III.3.9: Nozzle optimization with 50 optimization parameters. Convergence comparison for an inverse problem.

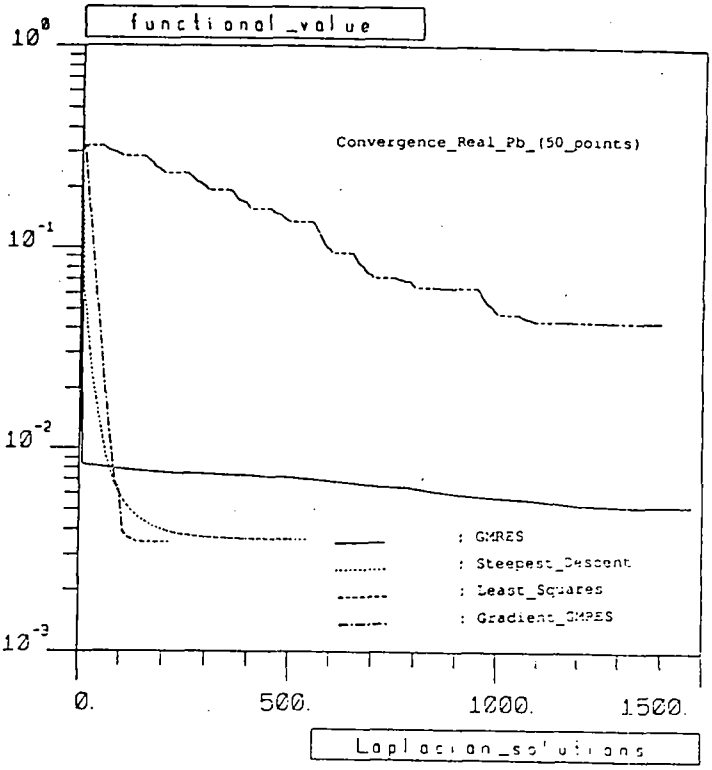

Diagram III.3.10: Nozzle optimization with 50 optimization parameters. Convergence comparison for a real problem.

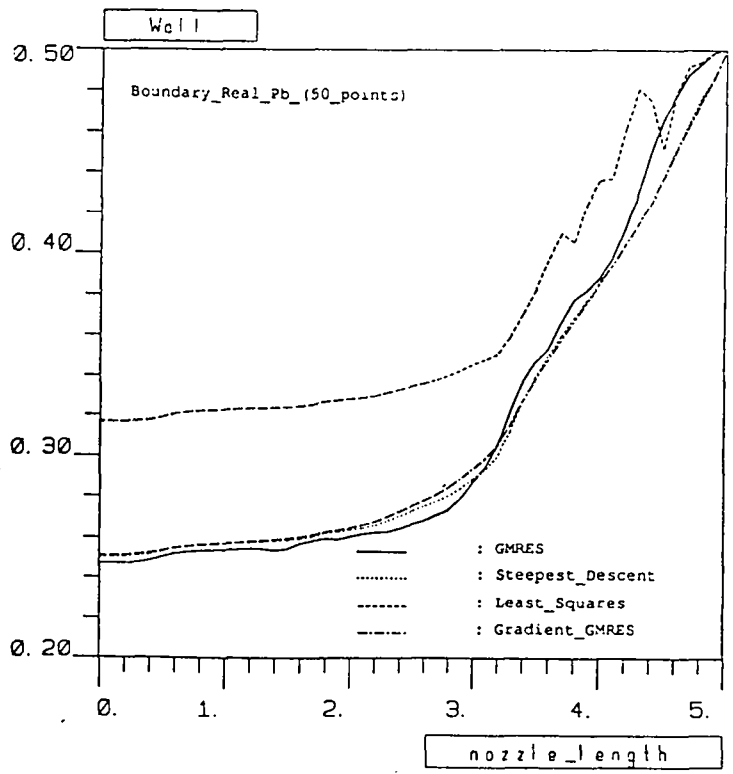

Diagram III.3.11: Nozzle optimization with 50 optimization parameters. Converged boundaries for a real problem.

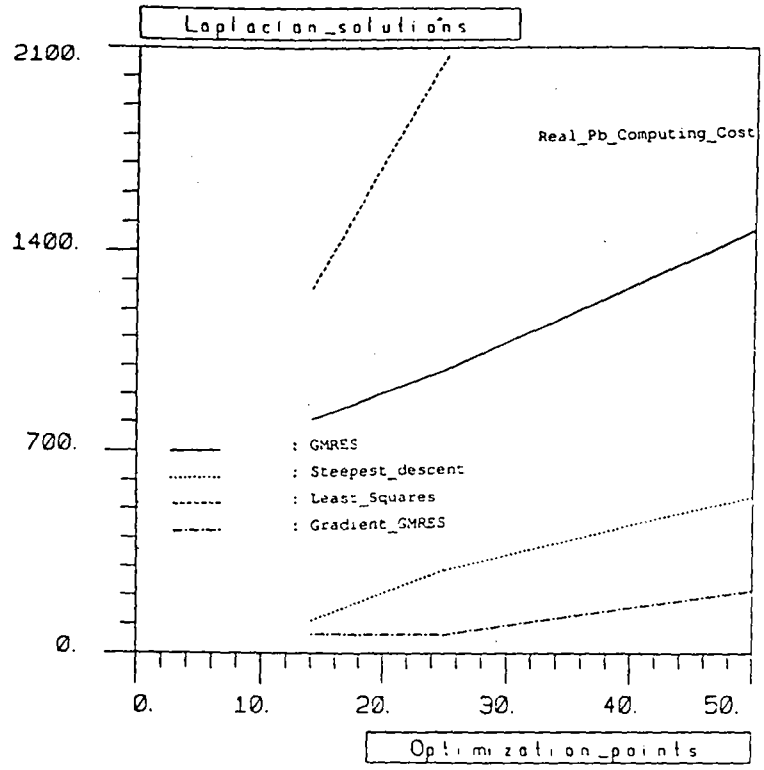

Diagram III.3.13: Computing cost vs the number of optimization parameters for a real problem.

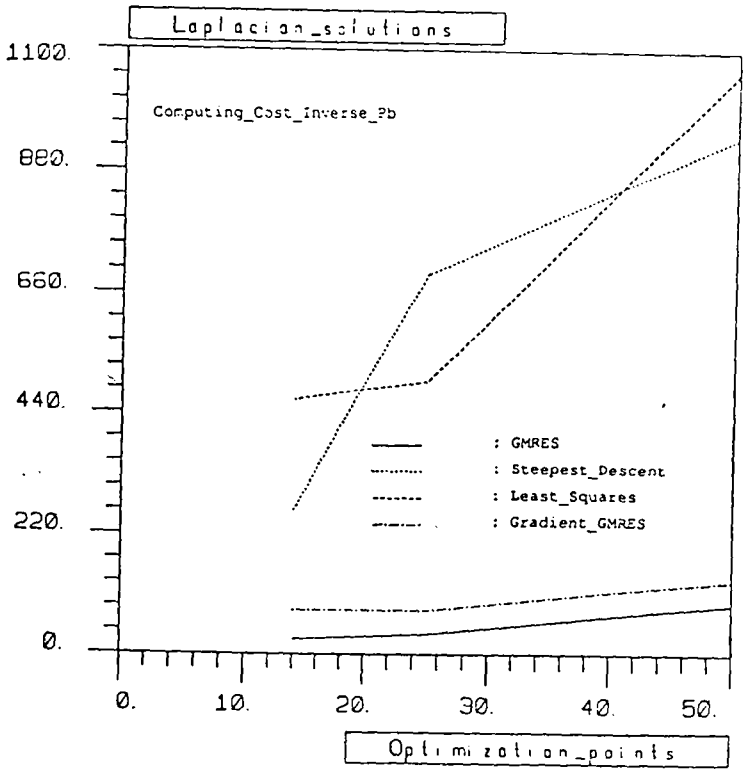

Diagram III.3.12: Computing cost vs the number of optimization parameters for an inverse problem.

IV. APPLICATION IN THE OPTIMIZATION OF LIFTING BODIES

IV.1 DESCRIPTION OF THE PROBLEM AND OF THE STATE EQUATIONS

The optimization problem has the same formulation as in I.1:

Find the lifting airfoil that achieves a certain velocity distribution on its boundary in an incompressible, two-dimensional, irrotational and steady flow.

We remind that this problem consists to find a boundary Γ which minimizes the functional

$$(IV.1.1) \qquad E(\Gamma) = \int_{\Gamma} \mid \vec{u} - \vec{v} \mid^2 d\Gamma$$

with the notation of I.1.

The flow is modelled in terms of the potential ϕ, that is we write

$$\vec{u} = grad\phi$$

at every point (since $rot\vec{u} = 0$) and by the continuity equation ($div\vec{u} = 0$) we deduce that

$$(IV.1.2) \qquad \Delta\phi = 0 \text{ in } \Omega.$$

Now, the flow domain is infinite, but we approximate it by a finite domain Ω (see Figure IV.1.1), whose "infinite" boundary Γ_∞ is sufficiently big compared to the airfoil boundary Γ_c. The speed is considered uniform at Γ_∞ with a certain incidence θ and equal to $\vec{u}_\infty$, so the boundary condition on it is

$$(IV.1.3) \qquad \frac{\partial\phi}{\partial n} \mid_{\Gamma_\infty} = \vec{u}_\infty \cdot \vec{n}.$$

The flow slips on the airfoil, therefore we have

$$(IV.1.4) \qquad \frac{\partial\phi}{\partial n} \mid_{\Gamma_c} = 0.$$

In addition, as the trailing edge P is a singularity of Γ_c, $\mid \nabla\phi(x) \mid$ becomes infinite when $x \to P$ and therefore

$$(IV.1.5) \qquad rot\vec{u} = \omega\delta_\Sigma,$$

where ω is constant and δ_Σ is Dirac's function on the streamline Σ from P. The discretization of (IV.1.5) is difficult, so we formulate the state equations on $\Omega - \Sigma$ and we replace it by the condition

$$(IV.1.6) \qquad \phi \mid_{\Sigma+} - \phi \mid_{\Sigma-} = \alpha.$$

We derive (IV.1.6) by integrating the equation $\frac{\partial \phi}{\partial \sigma} |_{\Sigma^+} = \frac{\partial \phi}{\partial \sigma} |_{\Sigma^-}$ which expresses the continuity of $\vec{u}$ through Σ. It is proved that (IV.1.6) is independent from the line Σ (which is not known) in the two dimensional case. The constant α is computed by solving Joukowski's condition

$$(IV.1.7) \qquad | \nabla \phi(P^+) |^2 = | \nabla \phi(P^-) |^2$$

(see Fig. IV.1.1 for P^+ and P^-). The discontinuity of ϕ on Σ implies a circulation around the airfoil, which causes a lift equal to $F_L = \alpha \rho \, | \vec{u}_\infty |^2$. Finally, in order to have a unique solution for ϕ, we must determine its value on a point. So, we add the condition

$$(IV.1.8) \qquad \phi(P) = 0, P = \text{ trailing edge.}$$

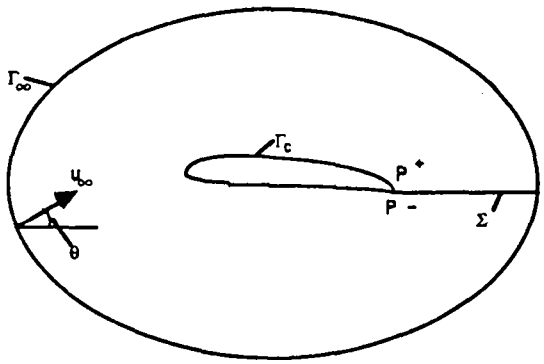

Figure IV.1.1

Therefore, the state equations for ϕ are

$$(IV.1.9) \qquad \begin{cases} -\Delta \phi = 0 \text{ in } \Omega - \Sigma \\ \dfrac{\partial \phi}{\partial n} |_{\Gamma_\infty} = \vec{u}_\infty \cdot \vec{n} \\ \dfrac{\partial \phi}{\partial n} |_{\Gamma_c} = 0 \\ \phi |_{\Sigma^+} - \phi |_{\Sigma^-} = \alpha \\ | \nabla \phi(P^+) |^2 = | \nabla \phi(P^-) |^2 \\ \phi(P) = 0 \end{cases}$$

In order to solve (IV.1.9), we decompose ϕ: $\phi = \phi_0 + \alpha \phi_1$, where the non-lifting potential ϕ_0 is computed by solving the following system

$$(IV.1.10) \qquad \begin{cases} -\Delta \phi_0 = 0 \text{ in } \Omega \\ \dfrac{\partial \phi_0}{\partial n} |_{\Gamma_\infty} = \vec{u}_\infty \cdot \vec{n} \\ \dfrac{\partial \phi_0}{\partial n} |_{\Gamma_c} = 0 \\ \phi_0(P) = 0 \end{cases}$$

and the discontinuous potential ϕ_1 is the solution of

$$(IV.1.11) \qquad \begin{cases} -\Delta\phi_1 = 0 \text{ in } \Omega - \Sigma \\ \dfrac{\partial\phi_1}{\partial n} \mid_{\Gamma_c \cup \Gamma_\infty} = 0 \\ \phi_1 \mid_{\Sigma^+} - \phi_1 \mid_{\Sigma^-} = 1 \\ \phi_1(P) = 0 \end{cases}$$

The constant α is then computed by solving the equation obtained by Joukowski's condition.

IV.2 VARIATIONAL FORMULATION AND DISCRETIZATION

The variational formulation of (IV.1.10), obtained by applying Green's formula and taking into account the boundary conditions, is

find $\phi_0 \in V_0(\Omega)$ such that

$$(IV.2.1) \qquad \int_\Omega grad\phi_0 \cdot gradw dx = \int_{\Gamma_\infty} \vec{u}_\infty \cdot \vec{n} d\gamma, \forall w \in V_0(\Omega)$$

where $V_0(\Omega) = \{w \in H^1(\Omega), w(P) = 0\}$.

In the same way, we obtain the variational formulation for (IV.1.11)

find $\phi_1 \in W$ such that

$$(IV.2.2) \qquad \int_{\dot{\Omega}} grad\phi_1 \cdot gradw dx = 0, \forall w \in V'$$

where

$$\dot{\Omega} = interior(\bar{\Omega} - \Sigma), V' = \{w \in H^1(\dot{\Omega}), w(P) = 0, w \mid_{\Sigma^+} - w \mid_{\Sigma^-} = \beta, \beta \text{ any constant}\},$$

$$W = \{w \in V', w \mid_{\Sigma^+} - w \mid_{\Sigma^-} = 1\}.$$

The discretization of (IV.2.1) and (IV.2.2) is done by P^1 triangular finite elements. Denoting by the subscript h the discretized approximations of domains, boundaries (the construction of the mesh is discussed later) and discrete solutions, we have the discretized form of (IV.2.1)

find $\phi_{0h} \in V_{0h}$ such that

$$(IV.2.3) \qquad \int_{\Omega_h} grad\phi_{0h} \cdot gradw_h dx = \int_{\Gamma_{\infty h}} \vec{u}_{\infty h} \cdot \vec{n} d\gamma, \forall w_h \in V_{0h}$$

where $V_{0h} = \{w_h \in C^0(\Omega_h) : w_h \mid_{T_k} \in P^1 \text{ and } w_h(P) = 0\}$.

The discretized form of (IV.2.2) is

find $\phi_{1h} \in W_h$ such that

$$(IV.2.4) \qquad \int_{\Omega_h} grad\phi_{1h} \cdot gradw_h dx = 0, \forall w_h \in V_{0h}$$

where

$$W_h = \{w_h \in C^0(\bar{\Omega}_h) : w_h \mid_{T_k} \in P^1, w_h(P) = 0, w_h \mid_{\Sigma^+} - w_h \mid_{\Sigma^-} = 1\}.$$

In order to solve Joukowski's condition in the discrete case, we consider the gradient of the solution restrained in the triangles which have one edge on the airfoil boundary and contain P (see Fig. IV.2.1), since $grad\phi$ is piecewise constant. We use Newton's method for solving the non-linear equation obtained from (IV.1.7).

The airfoil boundary is constructed using a cubic spline which passes through certain control points and has given derivatives at the leading and at the trailing edge. The domain Ω_h is automatically meshed using Voronoi's method if $\Gamma_{ch}, \Gamma_{\infty h}$ and Σ_h are discretized.

The functional to minimize is approximated, as in the case of the nozzle optimization, by an integral on the triangles having one edge on the boundary to optimize (see eq. (I.4.5)) with Γ_{2h} replaced by Γ_{ch}, Ψ_h by ϕ_h and $\vec{c}$ by $\vec{v}_h$.

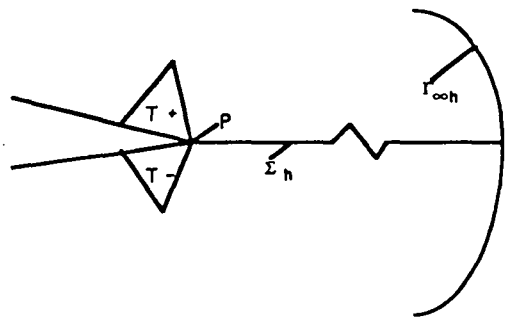

Figure IV.2.1

IV.3 SOLUTION OF THE OPTIMIZATION PROBLEM

The initial choice of the optimization parameters is the ordinates of the spline control points and the angles of the airfoil at the trailing edge P. We note that the leading and trailing edge are control points for the spline but they are fixed, that is their ordinates are not optimization parameters. In order to prevent the optimization points to cross, we must add bound constraints in the optimization procedure. Such a treatment would give complex programming, so it is desired to transform the constrained optimization problem to unconstrained. This is done in the following way: suppose that the parameter $v_i \in [a_i, b_i]$. We map the interval $[a_i, b_i]$ to $[-1, 1]$, so v_i is mapped to $p_i = 2\frac{(v_i - a_i)}{(b_i - a_i)} - 1$. Then we optimize with respect to $t_i = arccos(p_i)$ with no constraints as $p_i = cos(t_i) \in [-1, 1]$.

The optimization method used is GMRES for solving the optimality condition: $gradE_h = 0$. We have also used GMRES with no gradient computation for comparison. We give the computation of the gradient.

Since the functional is the same as in the case of the nozzle optimization, eq. (A.3) is still valid if we replace the stream function Ψ_h by the potential ϕ_h and the distribution $\vec{c}$ by the desired velocity distribution $\vec{v}_h$. In order to introduce again an adjoint equation, we have to differentiate the state equations (IV.1.9). We need for that a discrete variational formulation of (IV.1.9). According to Angrand (cf [7]) we have that the solution of (IV.1.9) is approximated by the solution of the system

find $\phi_h \in V_h'$ such that

$$(IV.3.1) \qquad \int_{\Omega_h} \nabla\phi_h \cdot \nabla w_h dx = \int_{\Gamma_{\infty h}} \vec{u}_{\infty_h} \cdot \vec{n} w_h d\gamma, \ \forall w_h \in V_{0h}$$

where

$$V_h' = \{w_h \in C^0(\dot{\Omega}_h) : w_h \mid_{T_k} \in P^1, w_h(P) = 0, w_h \mid_{\Sigma_h^+} - w_h \mid_{\Sigma_h^-} = \alpha,$$
$$\mid \nabla\phi_h(P^+) \mid^2 = \mid \nabla\phi_h(P^-) \mid^2\}.$$

Considering two different configurations Γ_c and Γ_c' of the airfoil boundary we have on the two corresponding domains Ω_h and Ω_h'

$$(IV.3.2) \qquad \int_{\Omega_h'} \nabla\phi_h' \cdot \nabla w_h' dx - \int_{\Omega_h} \nabla\phi_h \cdot \nabla w_h dx = 0$$

since the boundary $\Gamma_{\infty h}$ is fixed. So, if we note again

$$(IV.3.3) \qquad \delta J_{T_j} = \int_{T_j'} \nabla\phi_h' \cdot \nabla w_h' dx - \int_{T_j} \nabla\phi_h \cdot \nabla w_h dx$$

we obtain

$$(IV.3.4) \qquad \sum_j \delta J_{T_j} = \sum_j \int_{T_j'} \nabla\phi_h' \cdot \nabla w_h' dx - \sum_j \int_{T_j} \nabla\phi_h \cdot \nabla w_h dx = 0.$$

Therefore, eq. (A.14), (A.15) and (A.16) are still valid with the appropriate substitutions and if we take into account the discontinuity of ϕ_h on Σ_h in the computation of $grad\phi_h$ on the triangles on both sides of Σ_h. We can again perform the same gradient computation as in Appendix A.

Thus, the gradient of the cost functional with respect to the coordinates of the nodes is

$$(IV.3.5) \qquad \frac{\partial E_h}{\partial q_{kl}} = \int_{\Omega_h} [(\nabla w_k . \nabla P_h)\frac{\partial\phi_h}{\partial x_l}(q_k)$$
$$+ (\nabla\phi_h . \nabla w_k)\frac{\partial P_h}{\partial x_l}(q_k) - (\nabla\phi_h . \nabla P_h)\frac{\partial w_k}{\partial x_l}]dx$$
$$- 2\int_{A_h} (\nabla\phi_h - \vec{v}_h).\nabla w_k \frac{\partial\phi_h}{\partial x_l}(q_k)dx$$
$$+ \int_{A_h} \mid \nabla\phi_h - \vec{v}_h \mid^2 \frac{\partial w_k}{\partial x_l} dx \text{ with } l = 1, 2$$

and P_h the solution of the adjoint equation
find $P_h \in V_{0h}$ such that

$$(IV.3.6) \qquad \int_{\Omega_h} \nabla P_h . \nabla w_h dx = 2 \int_{A_h} (\nabla \phi_h - \vec{v}_h) . \nabla w_h dx, \forall w_h \in V_{0h}.$$

Then, the gradient with respect to z_i (ordinates of the optimization points) is

$$(IV.3.7) \qquad \frac{\partial E_h}{\partial z_i} = \sum_{k,l} \frac{\partial E_h}{\partial q_{kl}} \cdot \frac{\partial q_{kl}}{\partial z_i}.$$

In this case the derivative $\frac{\partial q_{kl}}{\partial z_i}$ is not analytically known, but is computed numerically using the finite difference method: we suppose that the optimization points i=1,2,$\cdots$,N move on the x_2 direction by δz_i and we compute the displacements of all the mesh nodes according to the expression

$$(IV.3.8) \qquad \delta q_k = \frac{1}{s_k} \sum_{i=1}^{N} w_i s_{ik} \delta z_i$$

where w_i is a weight function (for example in the two dimensional case we can consider the sum of the lengths of the edges on both sides of point i), $s_{ik} = \frac{1}{d_{ik}^{\beta}}$ with d_{ik}=distance of points i and k and β constant ≥ 2 and s_k normalization factor naturally taken equal to $\sum_{i=1}^{N} w_i s_{ik}$. So, $\frac{\partial q_{kl}}{\partial z_i} = \frac{\delta q_{kl}}{\delta z_i}$. The gradient with respect to the optimization parameters t_i is

$$(IV.3.9) \qquad \frac{\partial E_h}{\partial t_i} = \frac{\partial E_h}{\partial z_i} \cdot \frac{\partial z_i}{\partial p_i} \cdot \frac{\partial p_i}{\partial t_i}$$

with $\frac{\partial z_i}{\partial p_i} = .5(b_i - a_i)$ and $\frac{\partial p_i}{\partial t_i} = -sin(t_i)$, when $t_i \in [a_i, b_i]$. We have proved the

Proposition IV.3.1 *The gradient of the cost functional E_h is given by the equations (IV.3.5), (IV.3.6), (IV.3.7) ,(IV.3.8) and (IV.3.9).*

IV.4 AIRFOIL OPTIMIZATION WITH PRESSURE RECOVERY CRITERION

We consider now the following problem:

Find a lifting airfoil that achieves a certain pressure distribution on its boundary in an incompressible, two-dimensional, irrotational and steady flow.

The flow is modelled again in terms of the potential ϕ at every point, so the state equation, variational formulation and discretization are the same as in IV.1 and IV.2. The functional to minimize is (since the pressure is equal to $p = b - .5\vec{u}^2$ with b= constant)

$$(IV.4.1) \qquad E(\Gamma_c) = \int_{\Gamma_c} | \ \ | \nabla \phi |^2 - p_t \ |^2 \, d\Gamma_c$$

where Γ_c is the airfoil boundary and p_t is the desired pressure distribution on the airfoil. This functional is again approximated by an integral on the elements having one edge on the airfoil boundary, thus given by the sum

$$(IV.4.2) \qquad E_h(\Gamma_{ch}) = \sum_{T_i \in A_h} | \; | \nabla \phi_{hi} |^2 - p_{thi} \; |^2 \cdot | T_i |$$

with A_h the set of the above mentioned elements.

The method chosen to solve this problem is GMRES for solving the optimality condition, so we have to compute the gradient of the cost functional with respect to the optimization parameters. The optimization parameters are the ones given in IV.3 and we proceed in the same way: we first compute the $gradE_h$ with respect to the coordinates q_{kl}, $k = 1, \ldots, N$ and $l = 1, 2$ of all mesh points and then with respect to the optimization parameters t_i using the expression (IV.3.9). We perform a more general calculation of $gradE_h$.

The functional E_h depends formally on the optimization parameters t_i but it can also be considered (cf [12]) as a function of q_{kl}, $k = 1, \ldots, N$ and $l = 1, 2$ and the solution ϕ_{hj}, $j = 1, \ldots, N-1$ on the nodes without Dirichlet boundary condition (that is the trailing edge P is excepted). Therefore we write

$$(IV.4.3) \qquad E_h(t_i) = J_h(q_{kl}) = F_h(q_{kl}, \phi_{hj})$$

and supposing that a node k moves by δq_k, we have that F_h changes by

$$(IV.4.4). \qquad \delta F_h = \frac{\partial F_h}{\partial q_k} \cdot \delta q_k + \sum_j \frac{\partial F_h}{\partial \phi_{hj}} \cdot \delta \phi_{hj}$$

We introduce the adjoint state equation

find $P_h \in V_{0h}$ such that

$$(IV.4.5) \qquad \int_{\Omega_h} \nabla P_h \cdot \nabla w_j dx = \frac{\partial F_h}{\partial \phi_{hj}}, \; \forall j = 1, \ldots, N-1.$$

So, we obtain:

$$(IV.4.6) \quad \sum_j \frac{\partial F_h}{\partial \phi_{hj}} \cdot \delta \phi_{hj} = \sum_j \int_{\Omega_h} \nabla P_h \cdot \nabla (w_j \delta \phi_{hj}) dx$$
$$= \int_{\Omega_h} \nabla P_h \cdot \nabla (\sum_j w_j \delta \phi_{hj}) dx = \int_{\Omega_h} \nabla P_h \cdot \nabla \delta \tilde{\phi}_h dx.$$

Using eq. (A.15) with $w_h = P_h$, (IV.4.6) gives

$$(IV.4.7) \sum_j \frac{\partial F_h}{\partial \phi_{hj}} \cdot \delta \phi_{hj} = \int_{\Omega_h} [(\nabla w_k \cdot \nabla P_h) \nabla \phi_h(q_k) \cdot \delta q_k$$
$$+ (\nabla \phi_h \cdot \nabla w_k) \nabla P_h(q_k) \cdot \delta q_k - (\nabla \phi_h \cdot \nabla P_h) \nabla w_k \cdot \delta q_k] dx.$$

From eq. (IV.4.4), (IV.4.5), (IV.4.6) and (IV.4.7) we compute

$$(IV.4.8) \qquad \frac{\partial E_h}{\partial q_{kl}} = \frac{\partial F_h}{\partial q_{kl}} + \int_{\Omega_h} [(\nabla w_k . \nabla P_h)\frac{\partial \phi_h}{\partial x_l}(q_k) + (\nabla \phi_h . \nabla w_k)\frac{\partial P_h}{\partial x_l}(q_k) - (\nabla \phi_h . \nabla P_h)\frac{\partial w_k}{\partial x_l}]dx.$$

We have proved the following

Propostion IV.4.1 *The gradient of the cost functional E_h is given by the equations (IV.4.5), (IV.4.8) and (IV.3.9) where the partial derivatives $\frac{\partial z_i}{\partial p_i}$ and $\frac{\partial p_i}{\partial t_i}$ are the ones of IV.3.*

We note that $\frac{\partial F_h}{\partial q_{kl}}$ and $\frac{\partial F_h}{\partial \phi_{hj}}$ can be approximated by finite difference as follows:

$$\frac{\partial F_h}{\partial q_{kl}} \simeq \frac{F(q_1, \ldots, q_k + e_l \delta q_k, \ldots, q_N, \phi_{h1}, \ldots, \phi_{hN-1}) - F(q_1, \ldots, q_N, \phi_{h1}, \ldots, \phi_{hN-1})}{\delta q_k}$$

$$\frac{\partial F_h}{\partial \phi_{hj}} \simeq \frac{F(q_1, \ldots, q_N, \phi_{h1}, \ldots, \phi_{hj} + \delta \phi_{hj}, \ldots, \phi_{hN-1}) - F(q_1, \ldots, q_N, \phi_{h1}, \ldots, \phi_{hN-1})}{\delta \phi_{hj}},$$

where e_l is the l basis vector of the (x_1, x_2) frame. This finite difference approach allows to change the optimization criterion easily.

IV.5 NUMERICAL EXPERIMENTS

We use for the numerical tests on both criteria a discretization of the airfoil boundary with 56 points, while 20 control points are used for the spline. The mesh of Ω_h has 1000 nodes and 1896 elements. The optimization parameters correspond to the spline control points and to the angles of the airfoil at the trailing edge, so we have 20 optimization parameters because the leading and trailing edges are fixed. This discretization is used for the inverse problems.

For the real problem the number of optimization parameters is again 20 and the airfoil is discretized using 42 points. Ω is discretized with 1000 nodes and 1910 elements.

IV.5.1 TESTS FOR THE VELOCITY RECOVERY CRITERION

The test cases are an inverse problem where $\min E_h = 0$ and a real problem with $\min E_h \neq 0$.

For the inverse problem, we try to recover the velocity distribution on the boundary of a NACA0012 airfoil starting from a NACA0020. The results are shown in Diagrams IV.5.1 and IV.5.2 when GMRES is used for the solution of the optimality conditions.

For the real problem, the velocity distribution to recover is produced by deforming non-symmetrically the velocity distribution of a NACA0012. The starting airfoil is again

the NACA0020. Diagram IV.5.3 shows the convergence of Gradient-GMRES and of GMRES with no gradient computation. Diagram IV.5.4 shows the profiles till convergence of Gradient-GMRES.

We have also tested a more delicate inverse problem, that is to recover the velocity distribution on a Korn's airfoil starting from a NACA64A410 profile, using again GMRES for solving the optimality conditions. Results are shown on Diagrams IV.5.5 and IV.5.6.

IV.5.2 TESTS FOR THE PRESSURE RECOVERY CRITERION

The test case is an inverse problem, more precisely to recover the pressure distribution corresponding to Korn's airfoil having as initial configuration the NACA64A410 profile. The method used is GMRES for solving the optimality condition. Results are shown on Diagrams IV.5.7 (functional value convergence) and IV.5.8 (profile convergence).

IV.6 COMMENTS AND CONCLUSION

We observe that GMRES is a rather powerful tool in the optimization of lifting bodies when used for solving the optimality condition. We can decrease the functional value by a factor of 100 while the number of Laplacian solutions is reasonable in both kinds of problems tested when velocity distribution is recovered. However, the method becomes more expensive when the criterion is pressure distribution achievement (since this kind of problem is more sensitive than the velocity recovery) but is still satisfactory. In addition, we can see that the boundary convergence is satisfactory in all test cases. We note that the minimization over the Krylov subspace is done by the Least Squares Method (inspite the oscillations we have at the convergence) and not by an exact minimization because the latter would be costly as the optimization of airfoils is more sensitive than the nozzle optimization.

GMRES with no gradient computation (tested only for the velocity recovery problem) behaves as in the case of the nozzle optimization in a real problem, so the obtained minimum of the functional and the corresponding profile are unsatisfactory. The minimization over the Krylov subspace is exact, otherwise important oscillations occur at the convergence. The same effect is observed in the inverse problem, so use of the exact minimization is necessary and the convergence becomes slow. For this reason the result is not presented.

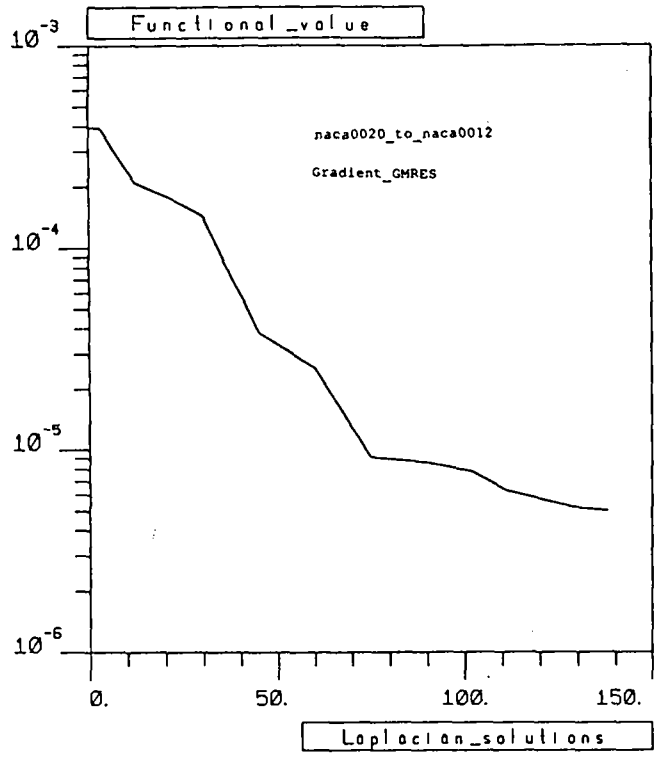

Diagram IV.5.1: Airfoil optimization with 20 optimization parameters. Convergence of Gradient-GMRES for an inverse problem - target velocity (NACA0020 to NACA0012).

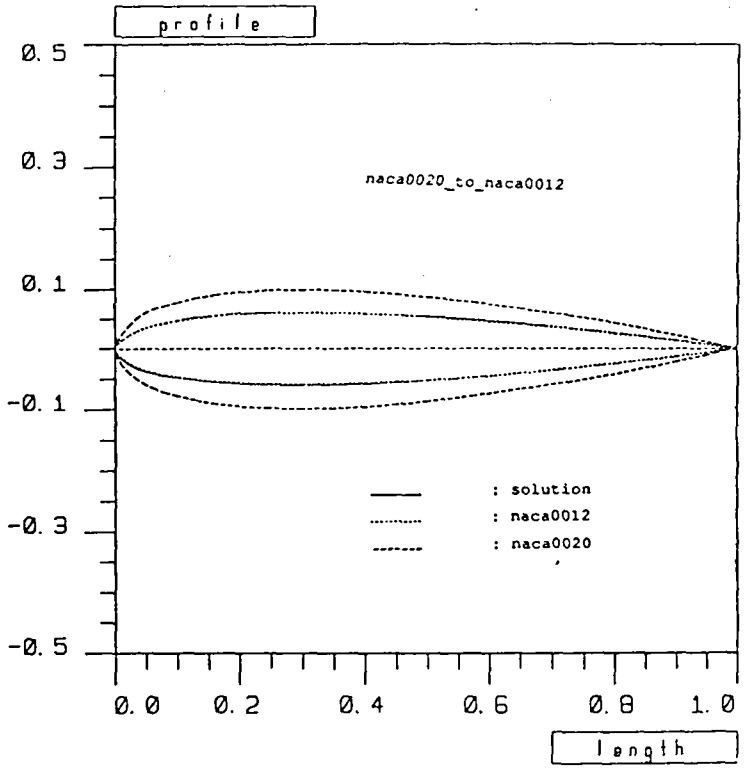

Diagram IV.5.2: Airfoil optimization with 20 optimization parameters. Profile convergence using Gradient-GMRES in an inverse problem - target velocity (NACA0020 to NACA0012).

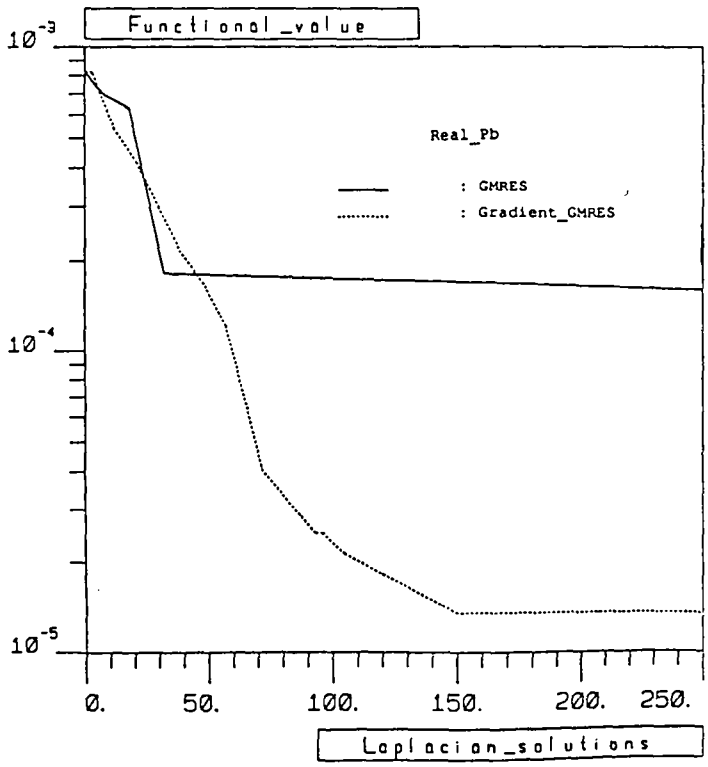

Diagram IV.5.3: Airfoil optimization with 20 optimization parameters. Convergence comparison for a real problem - target velocity.

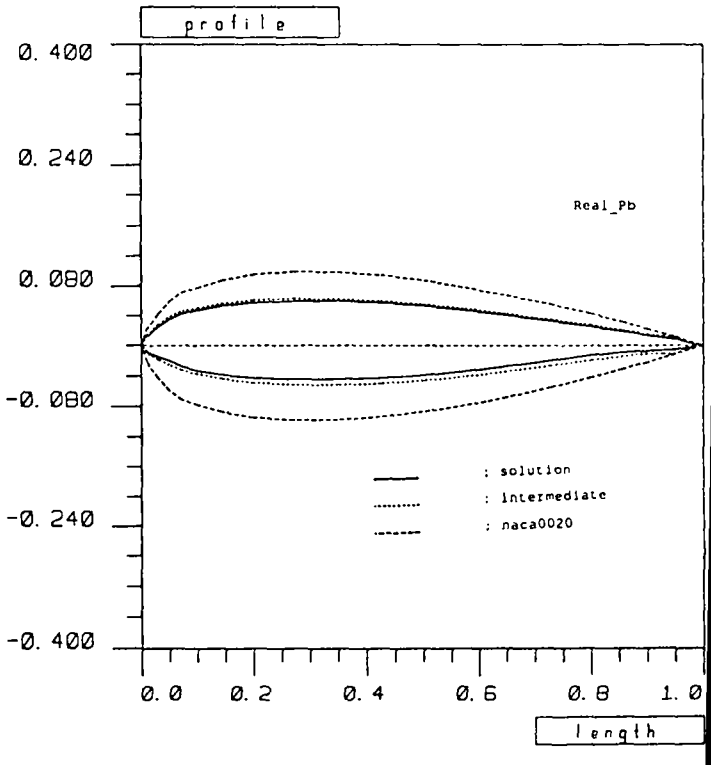

Diagram IV.5.4: Airfoil optimization with 20 optimization parameters. Profile convergence using Gradient-GMRES in a real problem - target velocity.

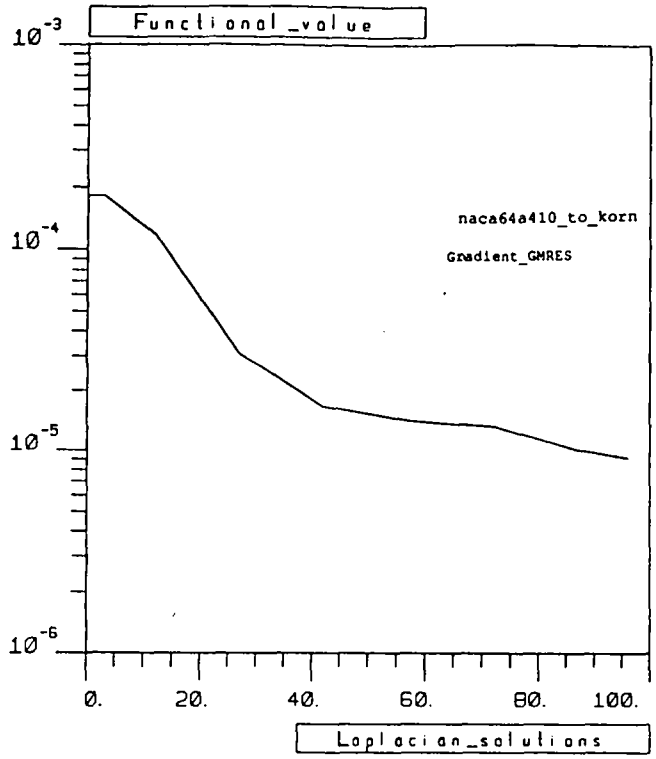

Diagram IV.5.5: Airfoil optimization with 20 optimization parameters. Convergence of Gradient-GMRES for an inverse problem - target velocity (NACA64A410 to Korn's profile).

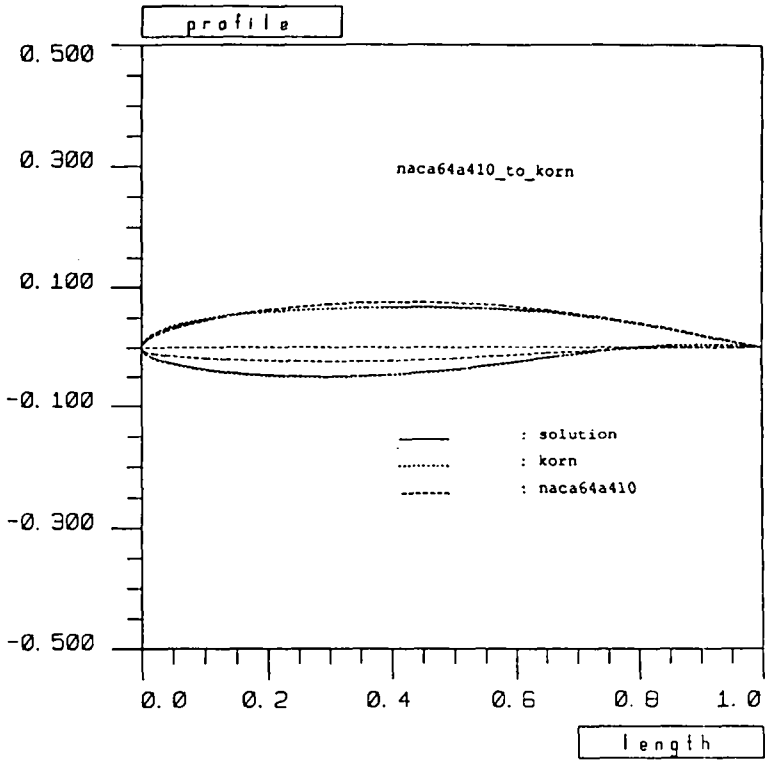

Diagram IV.5.6: Airfoil optimization with 20 optimization parameters. Profile convergence using Gradient-GMRES in an inverse problem - target velocity (NACA64A410 to Korn's profile).

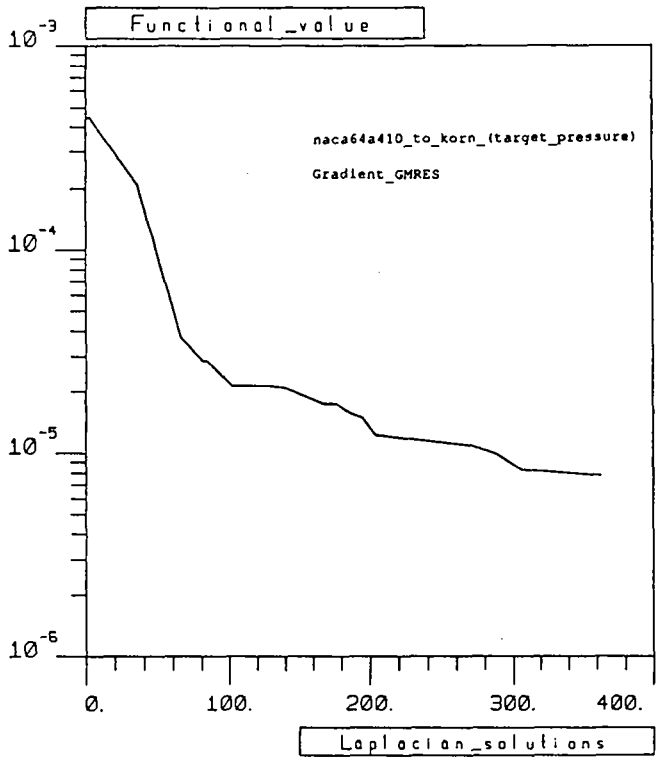

Diagram IV.5.7: Airfoil optimization with 20 optimization parameters. Convergence of Gradient-GMRES for an inverse problem - target pressure (NACA64A410 to Korn's profile).

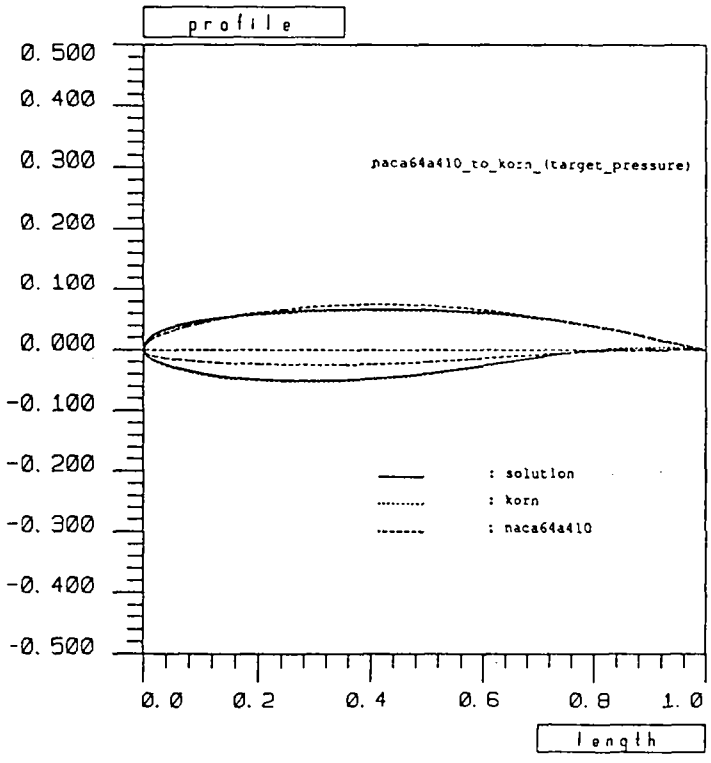

Diagram IV.5.8: Airfoil optimization with 20 optimization parameters. Profile convergence using Gradient-GMRES in an inverse problem - target pressure (NACA64A410 to Korn's profile).

APPENDIX A

We compute the discrete gradient of the cost functional with respect to the coordinates of the mesh nodes. We suppose that a vertex q_k of the triangle $T_j \in \Omega_h$ moves to $q_k + \delta q_k$ and the new triangle is T_j'. Then, since the mesh changes from Ω_h to Ω_h', the solution also changes from Ψ_h to Ψ_h' and from equation (I.4.5) we obtain

$$(A.1) \qquad \delta E_h = E_h' - E_h = \int_{A_h'} |\nabla\Psi_h' - \vec{c}|^2\, dx - \int_{A_h} |\nabla\Psi_h - \vec{c}|^2\, dx =$$

$$\int_{A_h' \cap A_h} (|\nabla\Psi_h' - \vec{c}|^2 - |\nabla\Psi_h - \vec{c}|^2)dx + \int_{A_h' \setminus A_h} |\nabla\Psi_h' - \vec{c}|^2\, dx - \int_{A_h \setminus A_h'} |\nabla\Psi_h - \vec{c}|^2\, dx$$

where A_h changes by a "small" amount to A_h' and denoting by $A_h \setminus A_h' = A_h - A_h \cap A_h'$ and $\nabla f = gradf$. Since we also have

$$(A.2) \qquad \int_{A_h' \cap A_h} (|\nabla\Psi_h' - \vec{c}|^2 - |\nabla\Psi_h - \vec{c}|^2)dx =$$

$$2\int_{A_h' \cap A_h} (\nabla\Psi_h - \vec{c}).\nabla\delta\Psi_h dx + o(\delta_{\Psi h}) =$$

$$2\int_{A_h} (\nabla\Psi_h - \vec{c}).\nabla\delta\Psi_h dx + o(\delta\Psi_h) + o(|\delta q_k|).$$

Thus, from (A.1) we have

$$(A.3) \qquad \delta E_h = 2\int_{A_h} (\nabla\Psi_h - \vec{c}).\nabla\delta\Psi_h dx + \int_{A_h' \setminus A_h} |\nabla\Psi_h' - \vec{c}|^2\, dx$$

$$- \int_{A_h \setminus A_h'} |\nabla\Psi_h - \vec{c}|^2\, dx.$$

The first integral of (A.3) will be computed using the adjoint state equation.

In order to derive the adjoint state equation we consider again the discretized variational form (I.4.4) of the state equations (I.2.4). Since

$$\Omega_h = \bigcup_j T_j\ ,$$

we have that

$$(A.4) \qquad \int_{\Omega_h} \nabla\Psi_h.\nabla w_h dx = \sum_j \int_{T_j} \nabla\Psi_h.\nabla w_h dx = \sum_j J_{T_j} = 0, \forall w_h \in V_{0h}.$$

Considering in Ω'_h the corresponding triangle T'_j to T_j, we write:

$$(A.5) \qquad \delta J_{T_j} = \int_{T'_j} \nabla\Psi'_h.\nabla w'_h dx - \int_{T_j} \nabla\Psi_h.\nabla w_h dx.$$

Since $T_j \backslash T'_j = T_j - T_j \cap T'_j$ and $T'_j \backslash T_j = T'_j - T_j \cap T'_j$, we have

$$(A.6) \qquad \int_{T'_j} \nabla\Psi'_h.\nabla w'_h dx = \int_{T'_j \backslash T_j} \nabla\Psi'_h.\nabla w'_h dx + \int_{T_j \cap T'_j} \nabla\Psi'_h.\nabla w'_h dx$$

$$(A.7) \qquad \int_{T_j} \nabla\Psi_h.\nabla w_h dx = \int_{T_j \backslash T'_j} \nabla\Psi_h.\nabla w_h dx + \int_{T_j \cap T'_j} \nabla\Psi_h.\nabla w_h dx.$$

Further, we see that

$$(A.8) \qquad \int_{T_j \cap T'_j} (\nabla\Psi'_h.\nabla w'_h - \nabla\Psi_h.\nabla w_h) dx =$$

$$\int_{T_j \cap T'_j} (\nabla\delta\Psi_h.\nabla w_h + \nabla\Psi_h.\nabla\delta w_h) dx + o(\delta\Psi_h) + o(\delta w_h)$$

$$\simeq \int_{T_j} (\nabla\delta\Psi_h.\nabla w_h + \nabla\Psi_h.\nabla\delta w_h) dx + o(\delta\Psi_h) + o(\delta w_h).$$

According to the Proposition II.2.1.1 we can write

$$(A.9) \qquad \int_{T'_j \backslash T_j} \nabla\Psi'_h.\nabla w'_h dx - \int_{T_j \backslash T'_j} \nabla\Psi_h.\nabla w_h dx = \int_{T_j} \delta q_k.\nabla w_k(\nabla\Psi_h.\nabla w_h) dx.$$

So, from (A.5), (A.6), (A.7), (A.8) and (A.9) we have

$$(A.10) \qquad \delta J_{T_j} = \int_{T_j} [\nabla\delta\Psi_h.\nabla w_h + \nabla\Psi_h.\nabla\delta w_h + \delta q_k.\nabla w_k(\nabla\Psi_h.\nabla w_h)] dx.$$

Since

$$w_h = \sum_j \alpha_j w_j, \forall w_h \in V_{0h} \ ,$$

we have acording to the Proposition II.2.1.2

$$(A.11) \qquad \delta w_h = -w_k \nabla w_h(q_k).\delta q_k, \forall \ w_h \in V_{0h}.$$

In addition, as the solution

$$\Psi_h = \sum_i \Psi_i w_i$$

and denoting by

$$\delta\tilde{\Psi}_h = \sum_i \delta\Psi_i w_i$$

we write

$$(A.12) \qquad \begin{aligned} \delta\Psi_h &= \sum_i \delta\Psi_i w_i + \sum_i \Psi_i \delta w_i + o(|\ \delta q_k\ |) \\ &\simeq \delta\tilde{\Psi}_h + \sum_i \Psi_i(-w_k \nabla w_i(q_k).\delta q_k) \\ &= \delta\tilde{\Psi}_h - w_k \nabla\Psi_h(q_k).\delta q_k. \end{aligned}$$

Using equations (A.11) and (A.12) in (A.10), we obtain

$$(A.13) \qquad \begin{aligned} \delta J_{T_j} &= \int_{T_j} [\nabla(\delta\tilde{\Psi}_h - w_k \nabla\Psi_h(q_k).\delta q_k).\nabla w_h \\ &+ \nabla\Psi_h.\nabla(-w_k \nabla w_h(q_k).\delta q_k) + \delta q_k.\nabla((\nabla\Psi_h.\nabla w_h)w_k)]dx \\ &= \int_{T_j} [\nabla\delta\tilde{\Psi}_h.\nabla w_h - \nabla(w_k \nabla\Psi_h(q_k).\delta q_k).\nabla w_h \\ &- (\nabla\Psi_h.\nabla w_k)(\nabla w_h(q_k).\delta q_k) + (\delta q_k.\nabla w_k)(\nabla\Psi_h.\nabla w_h)]dx \\ &= \int_{T_j} [\nabla\delta\tilde{\Psi}_h.\nabla w_h - (\nabla w_k.\nabla w_h)(\nabla\Psi_h(q_k).\delta q_k)]dx \\ &- \int_{T_j} [(w_k \nabla(\nabla\Psi_h(q_k).\delta q_k).\nabla w_h + (\nabla\Psi_h.\nabla w_k)(\nabla w_h(q_k).\delta q_k)]dx \\ &+ \int_{T_j} (\nabla\Psi_h.\nabla w_h)(\nabla w_k.\delta q_k)dx + o(\delta\Psi_h) + o(\delta w_h) + o(|\ \delta q_k\ |). \end{aligned}$$

So we obtain from (A.13)

$$(A.14) \qquad \begin{aligned} \sum_j \delta J_{T_j} &= \int_{\Omega_h} [\nabla\delta\tilde{\Psi}_h.\nabla w_h - (\nabla w_k.\nabla w_h)(\nabla\Psi_h(q_k).\delta q_k) \\ &- (\nabla\Psi_h.\nabla w_k)(\nabla w_h(q_k).\delta q_k) + (\nabla\Psi_h.\nabla w_h)(\nabla w_k.\delta q_k)]dx = 0 \end{aligned}$$

where we neglected the higher order terms of eq. (A.13) and we used the eq. (A.4). Equation (A.14) implies that

$$(A.15) \qquad \begin{aligned} \int_{\Omega_h} \nabla\delta\tilde{\Psi}_h.\nabla w_h dx &= \int_{\Omega_h} [(\nabla w_k.\nabla w_h)(\nabla\Psi_h(q_k).\delta q_k) \\ &+ (\nabla\Psi_h.\nabla w_k)(\nabla w_h(q_k).\delta q_k) - (\nabla\Psi_h.\nabla w_h)(\nabla w_k.\delta q_k)]dx. \end{aligned}$$

We introduce the adjoint state equation

Find $P_h \in V_{0h}$ such that

$$(A.16) \qquad \int_{\Omega_h} \nabla P_h.\nabla w_h dx = 2\int_{A_h} (\nabla\Psi_h - \vec{c}).\nabla w_h dx, \forall w_h \in V_{0h}.$$

If we take $w_h = \delta\tilde{\Psi}_h$ in (A.16) we have the equation

$$(A.17) \qquad \int_{\Omega_h} \nabla P_h . \nabla \delta\tilde{\Psi}_h dx = 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \delta\tilde{\Psi}_h dx$$

and taking $w_h = P_h$ in (A.15) we find

$$(A.18) \qquad \int_{\Omega_h} \nabla P_h . \nabla \delta\tilde{\Psi}_h dx = \int_{\Omega_h} [(\nabla w_k . \nabla P_h)(\nabla \Psi_h(q_k) . \delta q_k)$$
$$+ (\nabla \Psi_h . \nabla w_k)(\nabla P_h(q_k) . \delta q_k) - (\nabla \Psi_h . \nabla P_h)(\nabla w_k . \delta q_k)] dx.$$

Using eq. (A.12), the first integral of (A.3) becomes

$$(A.19) \qquad \begin{aligned} G_h &= 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \nabla \delta \Psi_h dx \\ &= 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \nabla \delta\tilde{\Psi}_h dx \\ &- 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \nabla (w_k \nabla \Psi_h(q_k) . \delta q_k) dx \\ &= \tilde{G}_h - \hat{G}_h . \end{aligned}$$

From eq. (A.17) and (A.18) we deduce

$$(A.20) \qquad \tilde{G}_h = \int_{\Omega_h} [(w_k . \nabla P_h)(\nabla \Psi_h(q_k) . \delta q_k)$$
$$+ (\nabla \Psi_h . \nabla w_k)(\nabla P_h(q_k) . \delta q_k) - (\nabla \Psi_h . \nabla P_h)(\nabla w_k . \delta q_k)] dx$$

and using this equation in (A.19) we find

$$(A.21) \qquad \begin{aligned} G_h &= \tilde{G}_h - 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \nabla w_k (\nabla \Psi_h(q_k) . \delta q_k) dx \\ &- 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) w_k . \nabla (\nabla \Psi_h(q_k) . \delta q_k) dx \\ &= \int_{\Omega_h} [(\nabla w_k . \nabla P_h)(\nabla \Psi_h(q_k) . \delta q_k) + (\nabla \Psi_h . \nabla w_k)(\nabla P_h(q_k) . \delta q_k) \\ &- (\nabla \Psi_h . \nabla P_h)(\nabla w_k . \delta q_k)] dx \\ &- 2 \int_{A_h} (\nabla \Psi_h - \vec{c}) . \nabla w_k (\nabla \Psi_h(q_k) . \delta q_k) dx. \end{aligned}$$

The last two integrals of (A.3) are (according to Proposition II.2.1.1)

$$(A.22) \qquad \int_{A'_h \setminus A_h} | \nabla \Psi'_h - \vec{c} |^2 \, dx - \int_{A_h \setminus A'_h} | \nabla \Psi_h - \vec{c} |^2 \, dx =$$
$$\int_{A_h} \delta q_k . \nabla (| \nabla \Psi_h - \vec{c} |^2 \, w_k) dx = \int_{A_h} \delta q_k . \nabla w_k \, | \nabla \Psi_h - \vec{c} |^2 \, dx.$$

because $\mid \nabla\Psi_h - \vec{c} \mid^2$ is piecewise constant. So, by (A.3), (A.21) and (A.22) we obtain

$$
\begin{aligned}
(A.23) \qquad \delta E_h = & \int_{\Omega_h} [(\nabla w_k.\nabla P_h)(\nabla\Psi_h(q_k).\delta q_k) \\
& + (\nabla\Psi_h.\nabla w_k)(\nabla P_h(q_k).\delta q_k) - (\nabla\Psi_h.\nabla P_h)(\nabla w_k.\delta q_k)]dx \\
& - 2\int_{A_h} (\nabla\Psi_h - \vec{c}).\nabla w_k(\nabla\Psi_h(q_k).\delta q_k)dx \\
& + \int_{A_h} \delta q_k.\nabla w_k \mid \nabla\Psi_h - \vec{c} \mid^2 dx.
\end{aligned}
$$

Therefore, the gradient of the cost functional with respect to the coordinates of the mesh nodes is

$$
\begin{aligned}
(A.24) \qquad \frac{\partial E_h}{\partial q_{kl}} = & \int_{\Omega_h} [(\nabla w_k.\nabla P_h)\frac{\partial\Psi_h}{\partial x_l}(q_k) \\
& + (\nabla\Psi_h.\nabla w_k)\frac{\partial P_h}{\partial x_l}(q_k) - (\nabla\Psi_h.\nabla P_h)\frac{\partial w_k}{\partial x_l}]dx \\
& - 2\int_{A_h} (\nabla\Psi_h - \vec{c}).\nabla w_k\frac{\partial\Psi_h}{\partial x_l}(q_k)dx \\
& + \int_{A_h} \mid \nabla\Psi_h - \vec{c} \mid^2 \frac{\partial w_k}{\partial x_l}dx.
\end{aligned}
$$

BIBLIOGRAPHY

[1] O. Pironneau
Optimal Shape Design for Elliptic Systems
Springer Series in Computational Physics, 1984

[2] E. Polak
Computational Methods in Optimization
Academic Press, New York 1971

[3] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling
Numerical Recipes – The Art of Scientific Computing
Oxford University Press, 1987

[4] O. Pironneau
Finite Element Methods for Fluids, 1989
John Wiley

[5] Y. Saad, M.H. Schultz
GMRES: a generalized minimal residual algorithm for solving
nonsymmetric linear systems
SIAM Vol.7, No 3, July 1986

[6] L.B. Wington, N.J. Yu, D.P. Young
GMRES Acceleration of Computational Fluid Dynamics Codes
American Institute of Aeronautics and Astronautics, 1984

[7] F. Angrand
Méthodes Numériques pour des Problèmes de Conception
Optimale en Aérodynamique
Thesis (in French), Université Paris VI, 1980

[8] R.P. Brent
Algorithms for Minimization without Derivatives
Englewood Cliffs, N.J. Prentice Hall, 1973

[9] P. Lascaux – R. Théodor
Analyse Numérique Matricielle Appliquée à l' Art de l' Ingénieur
Masson, Paris, 1986

[10] Y. Saad
Krylov Subspace Methods: theory, algorithms and applications

p. 477 – 496, Proceedings, Ninth International Conference on Computing Methods in
Applied Science and Engineering, 29 January – 2 February 1990

[11] M.O. Bristeau
Application of a Finite Element Method to Transonic Flow Problems using an Optimal Control Approach
Von Karman Institute for Fluid Dynamics
Lecture Series 1978 - 4

[12] G. Arumugam
Optimization de Forme pour quelques problèmes de Mécanique de Fluides
Thesis (in French), Université Paris VI, 1987

Imprimé en France
par
. l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399