GROWTH Project GRD1-1999-10752 "AEROSHAPE"

ADAPTATION OF OPTIMIZATION METHODS TO OPTIMAL DESIGN IN AERODYNAMICS

Contribution of partner INRIA To Task 3.4

Contribution to Deliverable D3-18

François Courty(*), Alain Dervieux(*), Jean-Charles Gilbert(**)
(*) Inria Sophia-Antipolis BP 93
06902 Valbonne Cédex, France
(**) Inria, Domaine de Voluceau BP 105
78153 Le Chesnay, France

June 4th, 2001

Summary

This report gives a synthesis the reflections of partner INRIA concerning some useful adaptation of the modern optimization algorithm to the context of optimal design in aerodynamics.

The role of INRIA in Aeroshape Task 3.4 is to study the specific needs in optimization tools related to Optimal Aerodynamic design, and to propose some new algorithms for dealing with them.

Partner INRIA has first studied which optimization tool could be used by an engineer passing from an optimization loop involving divided differences to an new one involving an adjoint evaluation.
1. It is in fact reasonable to just replace the divided differences by the analytical gradient. Then usual gradient or quasi Newton optimizers can be applied. An important gain in efficiency is already obtained for a number of parameters larger than 10-20 and if the number of state constraints is not larger than 5-10.
2. The case of equality constraints such as state equations is addressed by Sequential Quadratic Programming. Then at each main optimization step, the state equation is not solved, but only a linearised system ("Newton step"). This option has been already tested in the literature for Optimum Aerodynamics Design with some success and it merits some attention. One particular mechanism used for SQP robustness is the method of trust regions, used for controlling the stability and efficiency of the SQP iteration; it can be useful for industrial optimization computations.
3. Partner INRIA also proposes to go further in simplifying the state equation solution step towards "one-shot" type loops, by trying to reformulate it in the SQP +trust-region formalism. A first attempt takes the form of a new optimisation algorithm, but many theoretical investigations are yet necessary before first trials.
4. In the case of inequality state constraints, some recent interior points algorithms are suggested to partners.
5. Partner INRIA also proposes a status of its investigations concerning the treatment of multiple state constraints.

1 Introduction

The role of partner INRIA in Task 3.4 of Work Package 3 in Aeroshape project is the "Study of new gradient-based optimization strategies to efficiently solve constrained optimization problems".

Let us try to define in short the specificities of optimal design in aerodynamics

The main characteristics of Optimal Design is, according to Optimal Control formulation, its setting through a *state variable*. From which depend an *objective cost functional* and several *control and state dependent constraint functionals*. The state variable is *implicitly* defined from the control variable through the state equation.

The state variable can be in principle eliminated from the initial formulation, since it is a function of the control variable.
Then a new control problem is formulated with implicit composed mappings.

In a different approach, the state variable is combined with the control variable into a new optimization variable, for which the state equation is a set of *equality constraints*.
In that case, we are in the context of Sequential Quadratic Programming (SQP).

The CPU cost of each state and adjoint state solution is preponderant in Optimal Aerodynamic Design, and investigation of algorithms involving only partial solution of them can contribute to important CPU reductions.

In the case of several state constraints, usual approaches will rely on the same number of adjoint state systems. We give an example of a modern interior points algorithms for dealing with inequality constraints without computing several adjoints.

Some possible strategies for lowering the number of adjoints is then explored.

2 The Optimal Aerodynamic Design problem

We choose a prototype problem in order to introduce the important common features in Optimal Aerodynamic Design and we recall the basic gradient methodology.

2.1 A model problem

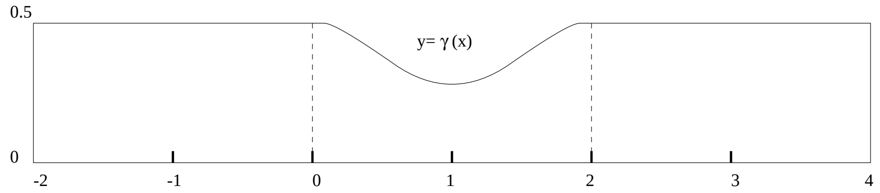

FIG. 1 – *The nozzle model problem*

We start with a model problem. It consists in solving an shape inverse/optimisation problem for a family of 2D nozzles in which the flow is modelled by the Euler equations. By inverse/optimisation we mean that the formulation is that of an optimisation problem, but the precise description will be, for assessment purpose, that of an inverse problem. The flow model is moderately complex and mesh will be a variable one. The geometry is depicted in Fig.1, inlet and outlet are of constant section, and the median part can vary through the ordinates of the upper boundary. Variations around this formulation have been studied in an European BRITE-ECARP project ([13][14]).

Taking the notations of optimal control theory, the control variable is a function γ of the abscissa which defines the domain shape (Fig.1), and to which corresponds a value of the objective or cost function j to be minimised with respect to γ. More precisely, the set of admissible controls is a subset Γ_{ad} of $\mathcal{C}^1([0,1],I\!R^2)$. To each γ in Γ_{ad} corresponds a smooth geometrical domain Ω_γ of the plan $I\!R^2$, the boundary of which is also denoted by γ.

Let us introduce:

$$\begin{aligned} J : \Gamma_{ad} \times H^1(I\!R^2, I\!R^4) &\longrightarrow I\!R \\ (\gamma, W) &\longmapsto J(\gamma, W) = \int_\gamma (P(W) - P^{target})^2 d\sigma \end{aligned} \tag{1}$$

where $P(W)$ is the pressure of W, (that is $0.4(W_4 - 0.5(W_2^2 + W_3^2)/W_1)$), W being a function of the independent variables x and y, and P^{target} a prescribed pressure generally computed from a prescribed shape.

The optimisation problem to be solved is written:

$$\text{Find } \gamma_0 \text{ in } \Gamma_{ad} \text{ such that } j(\gamma_0) = \min_{\gamma \in \Gamma_{ad}} j(\gamma) \tag{2}$$

with

$$j(\gamma) = J(\gamma, W(\gamma)) \tag{3}$$

and where $W(\gamma)$ is the solution of the state equation related to the control γ as follows:

$$\Psi(\gamma, W(\gamma)) \;=\; 0 \tag{4}$$

where

$$\Psi(\gamma, W) \;=\; \frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial y} + \text{Boundary conditions} \tag{5}$$

in which:

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad F(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u \frac{e+p}{\rho} \end{pmatrix}, \quad G(W) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v \frac{e+p}{\rho} \end{pmatrix} \tag{6}$$

where ρ is the density, u and v the velocity components following x and y, e the total energy, p the pressure given by:

$$p \;=\; P(W) \;=\; (\kappa - 1)[e - \frac{1}{2}\rho(u^2 + v^2)] \tag{7}$$

where κ is 1.4.

We observe that, at least formally, we can apply a chain rule for the differentiation of j, which leads to introduce an adjoint state, and allows to write the *optimality system* as follows:

$$\left\{ \begin{array}{rcl} \Psi(\gamma, W) & = & 0 \\ & & \text{(state)} \\ (\frac{\partial \Psi}{\partial W}(\gamma, W))^* \Pi & = & \frac{\partial J}{\partial W}(\gamma, W) \\ & & \text{(adjoint state)} \\ j'(\gamma) = \frac{\partial J}{\partial \gamma}(\gamma, W) - < \Pi, \frac{\partial \Psi}{\partial \gamma}(\gamma, W) > & = & 0 \\ & & \text{(optimality)} \end{array} \right. \tag{8}$$

2.2 Discrete standpoint

The mesh is made of triangles. It starts from an orthogonal triangulation with coordinates (x^1_{ij}, y^1_{ij}) of the constant-section nozzle geometry i.e. defined by $\gamma = 1$.
For any given control $(\gamma_1, i = 1, m)$, we derive a mesh on a accordion-like mode (Fig.2), that is by sliding vertices along vertical lines:

$$\forall i \in \{1, \ldots, m\},\; y^\gamma_{im'} = \gamma_i \;\forall j \in \{1, \ldots, m\},\; y^\gamma_{ij} = y^\gamma_{im'}\; y^1_{ij} \tag{9}$$

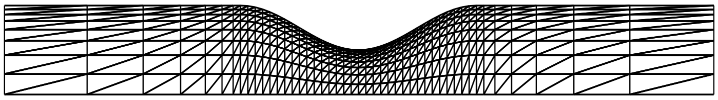

FIG. 2 – *Nozzle mesh with 423 vertices*

m is the number of points in x direction in the interval $[0,2]$ and m' is the number of points in the y direction.
In the sequel, the coordinates of the mesh will be shortly represented by the vector X of dimension M $(M = mm')$.
The mesh coordinates X are thus defined by (9) as an explicit function of the control parameter γ.

For discretizing the Euler equations, and more precisely the flux balance Ψ, we apply a vertex centered upwind finite-volume formulation (relying on median dual cells) ([**?**]) in which first order upwinding is applied for simplicity.

$$\Psi(\gamma, W_j^n) \quad = \quad \sum_{k \in V(j)} \Phi(W_j^n, W_k^n, \overrightarrow{\eta}_{jk}(\gamma)) + \text{ boundary conditions} \tag{10}$$

where $V(j)$ is the set of nodes (=vertices) that are neighbors of j, ∂C_j the cell boundary associated with node j, $\partial C_{jk} = \partial C_j \bigcap \partial C_k$, $\overrightarrow{\eta}_{jk} = \displaystyle\int_{\partial C_{jk}} \overrightarrow{n}\, d\sigma$ and $\overrightarrow{n}$ the outward pointing normal vector to ∂C_j.
The function Ψ depends on state variable W through the two first variables of the Riemann solver Φ and depends on the bounday parameter γ through mesh metrics and elementary fluxes $\overrightarrow{\eta}_{jk}(\gamma)$.
The upwinding in elementary flux Φ is the van Leer flux splitting (which is differentiable). It is useful to know that state solution will be iteratively obtained from a linearised implicit time advancing:

$$\left(\frac{M}{\Delta t^n} + \Psi'(W^n)\right) \delta W^{n+1} \quad = \quad -\Psi(W^n) \tag{11}$$

M is the diagonal mass matrix, and when the time step Δt^n grows infinitely, this algorithm (11) becomes a Newton iteration for solving $\Psi(W) = 0$.

We are now considering an optimisation with m real parameters:

$$\begin{aligned} j : I\!R^m &\longrightarrow I\!R \\ \gamma &\longmapsto j(\gamma) \end{aligned}$$

$$\begin{aligned} J : I\!R^m \times I\!R^N &\longrightarrow I\!R \\ (\gamma, w) &\longmapsto J(\gamma, w) \end{aligned}$$

where the dimension N of the state variable is four times the number of nodes (=vertices) $N = 4 \times mm'$.

The discrete variant of (2) writes:

$$\text{Find } \gamma_0 \text{ in } I\!R^m \text{ such that } j(\gamma_0) = \min_{\gamma \in I\!R^m} j(\gamma). \tag{12}$$

We obtain an adjoint state and optimality conditions by a way similar to that of the previous section.

2.3 Optimisation with gradient

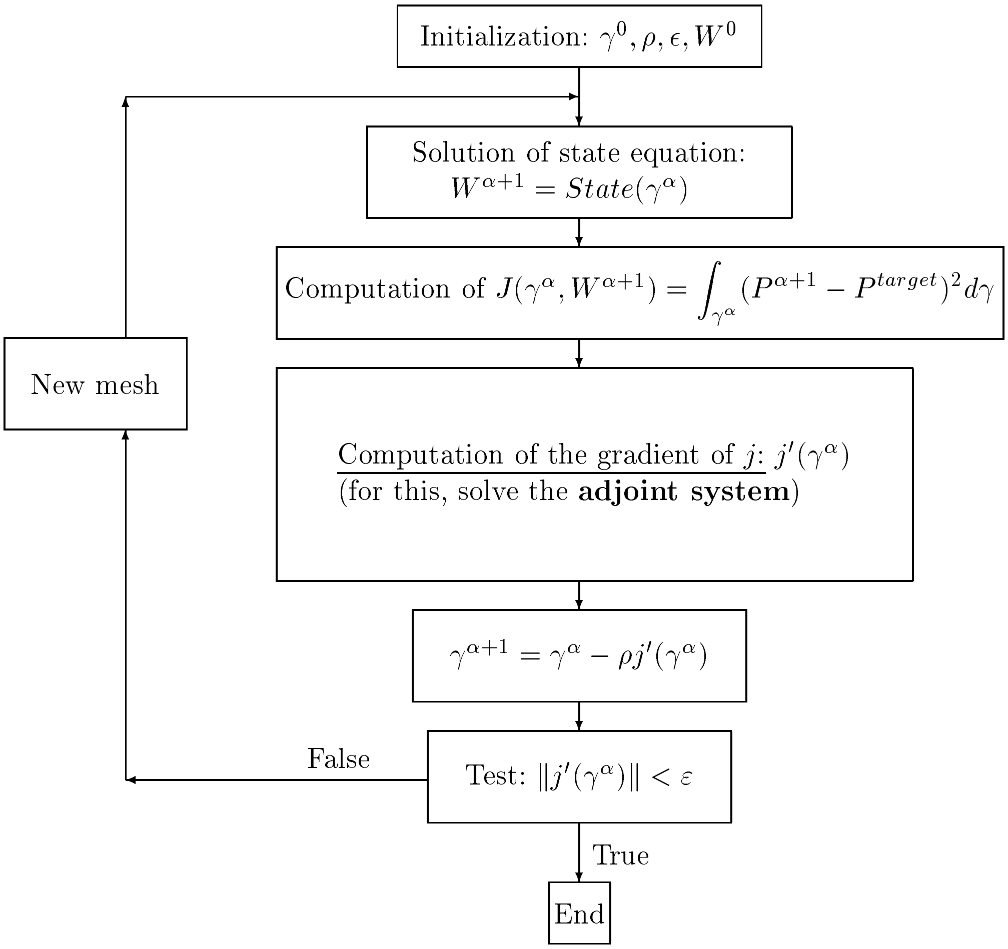

FIG. 3 – *Description of the "gradient/adjoint algorithm"*

In Fig. 3 is depicted the constant step gradient method with adjoint, as a reference algorithm. The rest of the paper examines different ways to improve or extend the gradient optimization algorithm.

2.4 Common features in Optimal aerodynamic Design

Let us assume that we have the software to compute the whole adjoint-based sensitivity. In a adjoint-based optimization loop, the computing effort is essentially invested into the solution of the state and adjoint equations:
- The state equation is, in new optimisation approaches, an Euler or Reynolds-Averaged Navier-Stokes discrete system. The number of unknown is very large (several millions, to compare to the number of parameters, several hundreds), and it is a stiff nonlinear system, involving sparse but large and non-symmetric matrix.
- The adjoint state equation is as large, linear, nonsymmetrical, and in practise not much easier to solve than the state equation.

It is then important:

- either to have a gradient algorithm with state and adjoint systems solved at each iteration, with a very fast convergence, in less than ten iterations,

- or to have an algorithm that relaxes the needs in complete solution of state and adjoint systems at each iteration.

A second point is that in standard optimization algorithms, the gradient of each state constraint is necessary. This means that an adjoint system has to be solved for each of these constraints. Either a large number of constraints should be avoided, or a new approach with only one adjoint system should be found.

3 Local SQP methods

Minimizing a functional with a Newton-Raphson iteration consits in approximating the functional by a quadratic (tangent) one. The Sequential Quadratic Programming method extends this idea to minimum problems with equality constraints:

$$\min \quad \phi(z) \tag{13}$$

$$\text{subject to} \quad c(z) = 0. \tag{14}$$

At an iterate z, we generate a displacement

$$d = \begin{pmatrix} d_x \\ d_s \end{pmatrix} \tag{15}$$

by solving the quadratic program

$$\min \quad \nabla\phi(z)^T d + \frac{1}{2} d^T \mathcal{W} d \tag{16}$$

$$\text{subject to } A(z)^T d + c(z) = 0, \tag{17}$$

where $\mathcal{W}$ is the Hessian of the Lagrangian of problem (13) [1] with respect to z, and where A^T is the Jacobian of c.

In this section we consider the discrete problem of Section 2. We try to keep the same notations and at the same time some coherence with rather usual notations in Optimization theory, as, for example, in [4],p135-.

3.1 Formulation with respect to a Lagrangian

The gradient of functional J is denoted by $g = \begin{pmatrix} \frac{\partial J}{\partial W} & \frac{\partial J}{\partial \gamma} \end{pmatrix}$ and the Jacobian of constraints is denoted by A.

According to the optimality system (8), the constraints Jacobian A at $x = (W, \gamma)$ is given by $A = \begin{pmatrix} \frac{\partial \Psi}{\partial W} & \frac{\partial \Psi}{\partial \gamma} \end{pmatrix}$. Here A is a $m \times n$ matrix. It is partitioned into $(B \quad N)$ where $B = \frac{\partial \Psi}{\partial W}$ and $N = \frac{\partial \Psi}{\partial \gamma}$.
Matrix B is a square one, of order m and N is a $m \times (n-m)$ rectangular matrix.
The Lagrangian of the optimization problem is
$L(x, \lambda) = J(x) - \lambda^T \Psi(x)$,
where λ is the Lagrange multiplier. The notation $\| \, . \, \|$ holds for the norm l_2.

The constrained discrete problem writes now:

$$\text{Find } x_{opt} = (W_{opt}, \gamma_{opt}) \text{ in } I\!R^{m_e} \times I\!R^{n-m_e} \tag{18}$$
$$\text{such that } J(x_{opt}) = \min_{\Psi(x)=0} J(x). \tag{19}$$

According to the Karush-Kuhn-Tucker theory (**KKT**, see [2] 12.2), that we apply to Problem (18), there exists a Lagrange multiplier λ_{opt}, belonging to $I\!R^m$ such that the following equalities are satisfied at optimum $x_{opt} = (\gamma_{opt}, W_{opt})$.

$$\begin{cases} \nabla_x L(x_{opt}, \lambda_{opt}) = 0 \\ \Psi(x_{opt}) = 0 \end{cases} \tag{20}$$

From the definition of the initial optimization problem, we know that $x = (W, \gamma)$ is solution of the state equation, so that the constraint $\Psi(x_{opt}) = 0$ is active. Returning to the notations of the initial problem, we get:

1. Please do not confuse with the state variable W in Section 2

$$
\begin{cases} g(x_{opt}) - A(x_{opt})^T \lambda_{opt} = 0 \\ \Psi(x_{opt}) = 0 \end{cases} \tag{21}
$$

We then replace A by its expression:

$$
\begin{cases} \dfrac{\partial J}{\partial W}(\gamma_{opt}, W_{opt}) - < \dfrac{\partial \Psi}{\partial W}(\gamma_{opt}, W_{opt}), \lambda_{opt} >= 0 \\ \dfrac{\partial J}{\partial \gamma}(\gamma_{opt}, W_{opt}) - < \dfrac{\partial \Psi}{\partial \gamma}(\gamma_{opt}, W_{opt}), \lambda_{opt} >= 0 \\ \Psi(x_{opt}) = 0 \end{cases} \tag{22}
$$

Let us compare the optimality system equations (8) with equations of system (22). We observe that the adjoint state Π_{opt} is nothing else but the Lagrange multiplier λ_{opt}, and to solve (8) is to solve the system (21) given but the KKT conditions. In the sequel, we shall denote by Π the Lagrange multiplier arising in KKT conditions.

3.2 Newton method

This section recalls some known Newton formulation with the notations of [2], 18.1.

A possible approach for solving (21), is the Newton iteration for the simultaneous solution in x and Π, of the nonlinear system of $n+m$ equations and $n+m$ unknowns:

$$
F(x, \Pi) = \begin{bmatrix} \nabla J - A(x)^T \Pi \\ \Psi(x) \end{bmatrix} = 0 \tag{23}
$$

The Jacobian of (23) is given by:

$$
\begin{bmatrix} \mathcal{W}(x, \Pi) & -A(x)^T \\ A(x) & 0 \end{bmatrix} \tag{24}
$$

Where $\mathcal{W}$ is the Hessian of the Lagrangian :

$$
\mathcal{W}(x, \Pi) = \nabla^2_{xx} L(x, \Pi) i.e. \mathcal{W}(x, \Pi) = \nabla^2_{xx}(J(x) - \Pi^T \Psi(x)).
$$

Starting from iterate (x_k, Π_k), the Newton step gives:

$$\begin{bmatrix} x_{k+1} \\ \Pi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \Pi_k \end{bmatrix} + \begin{bmatrix} d_k \\ p_k \end{bmatrix} \tag{25}$$

where d_k and p_k are solutions of the KKT system:

$$\begin{bmatrix} \mathcal{W}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ p_k \end{bmatrix} = \begin{bmatrix} -\nabla J_k + A_k^T \Pi_k \\ -\Psi_k \end{bmatrix} \tag{26}$$

This produces a primal-dual method in the sense that when linearizing the optimality conditions, we have generated a sequence (x_k, Π_k), where x_k approximates a primal solution x_{opt} and where Π_k approximates a dual solution Π_{opt} (cf [4] 10.2).
We can express the Newton system (26) as the optimality system of a tangent quadratic problem:

$$\begin{cases} \min\limits_{d \in \mathbb{R}^n} \; d^T g_k + \dfrac{1}{2} d^T \nabla_x^2 L(x_k, \Pi_k) d \\ \qquad A_k^T d + \Psi_k = 0 \end{cases} \tag{27}$$

3.3 Quasi-Newton Reduced Hessian method

We follow [4], 10.3. The set

$$\nu_x := \{y \in \mathbb{R}^n : \Psi(y) = \Psi(x)\}$$

is a submanifold of $\mathbb{R}^n$ with dimension $n-m$. The tangent space to it a point x is the null space $KerA_x$ of the Jacobian $A_x = A(x)$. Subspace $KerA_x$ and a supplementary subspace allow a decomposition $\mathbb{R}^n$ making easier the description an interpretation of algorithms.
We assume that there exists a smooth mapping $Z^- : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n - m_e}$, such that
$\forall x \in \mathbb{R}^n, Z_x^- = Z^-(x)$ is a basis of the tangent space, that is:

$$\forall x \in \mathbb{R}^n,\; Z_x^- \textit{ is injective and } AZ_x^- = 0. \tag{28}$$

Then the columns of Z_x^- form a basis of the tangent space $KerA_x$, see details in [4] 10.1.

On the other side, since A_x is surjective, it has a right inverse A_x^-, a $n \times m$ matrix that we assume to be given by the value at x of a smooth mapping $A_x^- : I\!R^n \to I\!R^{n\times m}$ thus satisfying:

$$A_x A_x^- = I_m, \ \forall x \in I\!R^n. \tag{29}$$

This implies in particular that A_x^- is injective. The image of A_x^- is a supplementary space of $KerA_x$ since $Im(A_x^-)\bigcap Ker(A_x) = \{0\}$ and
$dimIm(A_x^-) + dimKer(A_x) = m_e + (n - m_e) = n$.
We can thus write $I\!R^n$ as the direct sum of the subspace generated by the columns of Z_x^- and of the subspace generated by the columns of A_x^-, in other words, for any $x \in I\!R^n$,

$$I\!R^n = Im(Z_x^-)\bigoplus Im(A_x^-).$$

Let $Z^- : I\!R^n \to I\!R^{n\times n-m_e}$ and $A^- : I\!R^n \to I\!R^{n\times m}$ two mappings respectively satisfying (28) and (29). then there exists a unique mapping:

$$Z : I\!R^n \to I\!R^{n-m_e\times n}$$

satisfying for any $x \in I\!R^n$:

$$Z_x A_x^- = 0_{n-m_e\times m} \tag{30}$$
$$Z_x Z_x^- = I_{n-m_e} \tag{31}$$

where $Z_x := Z(x)$. This mapping is completely determined by the following identity, valid for any $x \in I\!R^n$:

$$I_n = A_x^- A_x + Z_x^- Z_x. \tag{32}$$

Using statement (32), we get, for any $v \in I\!R^n$,

$$v = A_x^- A_x v + Z_x^- Z_x v, \tag{33}$$

a relation which decompose v into a horizontal component $Z_x^- Z_x v$ and a vertical one $A_x^- A_x v$. This decomposition is determined as soon as we have matrices Z_x^- and A_x^-.
We call *reduced gradient* of J at x for the basis Z^-, the vector of $I\!R^{n-m}$ defined by

$$\tilde{g}(x) := \left(Z^-(x)\right)^T \nabla J(x) \ = \ \left(Z^-(x)\right)^T g(x) \tag{34}$$

If we multiply the first equation of (21) by Z_x^{-T}, we get thanks to (28), the reduced optimality system:

$$\left\{ \begin{array}{l} \tilde{g}(x_{opt}) = 0 \\ \Psi(x_{opt}) = 0 \end{array} \right. \tag{35}$$

3.3.1 Restoration phase

In order to restore the constraint, we use a Newton step, starting from an iterate x_k. We define the vertical step by:

$$v_k = A_k^- \Psi_k = Y_k \Psi_k. \tag{36}$$

Then, replacing matrix A_k^-, we get:

$$v_k = \begin{pmatrix} B_k^{-1} \\ \\ 0_{n-m \times m} \end{pmatrix} \Psi_k = \begin{pmatrix} (\dfrac{\partial \Psi_k}{\partial \mathcal{W}})^{-1} \Psi_k \\ \\ 0_{n-m \times 1} \end{pmatrix}. \tag{37}$$

3.3.2 Minimisation phase

We define the tangent step by $h_k = Z^-(v_k) G_k^{-1} \tilde{g}(x_k)$, where G_k is an approximation of the reduced Hessian of the Lagrangian: $Z^-(x_k)^T L(x_k, \Pi_k) Z^-(x_k)$.

4 Trust region methods

In modern optimization, sensitivity computations are used for reducing the initial problem to simplified ones.
- When a descent direction is available, a linear search will find the minimum in the direction.
- When a quadratic model is available, as in (16), its minimization is (relatively) easy, but does it give a good update of the initial problem unknown? We know that Newton-type iterations (as SQP iteration may be) do not converge in an arbitrary neighborhood of the solution. It is then necessary to evaluate in which local region minimizing the quadratic model is also a good step for minimizing the initial problem (13).

Then the approximate quadratic problem (16) is replaced by

$$\begin{aligned} \min \quad & \nabla\phi(z)^T d + \frac{1}{2} d^T \mathcal{W} d \\ \text{subject to } & A(z)^T d + c(z) = 0 \\ & \|d\| \quad \leq \quad \Delta \end{aligned} \tag{38}$$

This is the idea of trust region methods. We explain now these ideas in the context of our Optimum Design problem.

4.1 Solving for $x = (W, \gamma)$.

The algorithm proposed by Byrd [7] and Omojokun[8] can be understood as a SQP method with a trust region. This algorithm is a sequel of those proposed by Vardi [5], and Celis, Dennis and Tapia [6]. The algorithm transforms the constrained SQP problem in an unconstrained problem with a trust region, that is more easy to solve. The Byrd-Omojokun algorithm in particular is especially interesting for large scale problems.

Starting from the iterate $x_k = (W_k, \gamma_k)$, we choose a trust radius Δ_k and Lagrange multipliers Π_k. The new step d_k is obtained by solving:

$$\begin{aligned} \min_{d \in I\!R^n} \quad & d^T g_k + \frac{1}{2} d^T \nabla_x^2 L(x_k, \Pi_k) d \\ \text{under the linearized constraint} \quad & A_k d + \Psi_k = 0 \\ & \|d\| \leq \Delta_k \end{aligned} \tag{39}$$

where subscript k means that a function is evaluated at x_k. However, the third line in (39) can induce a too small step d so that the linearized constraint may not be satisfied after this step.

4.1.1 State restoration phase

Byrd and Omojokun propose to compute first a step that is into the trust region, and satisfies the linearized constraint as much as possible. This is obtained by introducing a relaxation parameter $\xi \in (0,1)$ and by computing a step v_k which solves the vertical or normal subproblem:

$$\begin{aligned} & \min_{v \in I\!R^n} \ \|A_k v + \Psi_k\| \\ \text{under condition} \quad & \|v\| \leq \xi \Delta_k \end{aligned} \tag{40}$$

This vertical problem may have many solutions, but Lalee, Nocedal and Plantenga have shown in [3] that is has always a unique solution in the image space of A_k. This allows a complete decoupling between this subproblem and the second subproblem.

We remark that, in the case of optimal design with an invertible state equation (the partial Jacobian $\dfrac{\partial \Psi}{\partial W}$ is invertible), it is enough and clever to restore W (and not γ).

4.1.2 Minimisation phase

The Byrd-Omojokun algorithm is built in such a way that the step d does not get closer to the admissible manifold than v_k does. Then we reformulate the definition of d:

$$\begin{aligned} & \min_{d \in I\!R^n} \ d^T g_k + \frac{1}{2} d^T \nabla_x^2 L(x_k, \Pi_k) d \\ \text{under the constraint} \quad & A_k d = A_k v_k \\ & \|d\| \leq \Delta_k \end{aligned} \tag{41}$$

An efficient technique for solving this step relies on the reduced-Hessian strategy. Problem (42) has a non-empty admissible region: for example $d = v_k$ satisfies all constraints. Then it is solved by adding an extra step to the vertical one v_k. For doing this, we use matrix Z_k^- above, the column vector of which form a basis of the null space of A_k, so that $A_k Z_k^- = 0$ (cf (28)). We then define the total step of the algorithm as the sum $d_k = v_k + Z_k^- u_k$ where $u \in I\!R^{n-m_e}$ is defined later. We replace the new expression of d in (42). Remembering that v_k and $Z_k^- u_k$ are orthogonal, and skipping constant terms, we get:

$$\min_{u \in I\!R^{n-m_e}} \quad (g_k + \nabla_x^2 L_k v_k)^T Z_k^- u + \frac{1}{2} u^T \left(Z_k^-\right)^T \nabla_x^2 L_k Z_k^- u \tag{42}$$

$$\textit{under the contraint} \quad \|Z_k^- u\| \leq \sqrt{\Delta_k^2 - \|v_k\|^2}. \tag{43}$$

We denote by u_k the solution of this horizontal (or tangent) problem and we define the total step as the sum of vertical step v_k and tangent step $h_k = Z_k^- u_k$.

$$d_k = v_k + Z_k^- u_k. \tag{44}$$

We would then set:

$$x_{k+1} = x_k + d_k, \tag{45}$$

under the condition that γ_{k+1} actually decreases the *merit function* (to be defined in the sequel); then the minimization variable is incremented by d_k. Otherwise the diameter Δ_k of the trust region is lowered and the minimization variable is not updated.

The Byrd-Omojokun method has replaced the initial problem by two quadratic programming problems with trust region.

4.2 Byrd-Omojokun algorithm overview

Initialization of constants $\varepsilon > 0$ and $\eta \in (0, 1)$
Initialization of x_0 and $\delta_0 > 0$.
For k=0,...
Compute J_k, Ψ_k, g_k, A_k and Z_k^-.
Compute multipliers Π_k.
If $\|g_k - A_k \Pi_k\|_\infty < \varepsilon$ **et** $\|\Psi_k\|_\infty < \varepsilon$ **Then stop.** (Problem is solved)
Restoration phase
Compute v_k by solving the vertical subproblem
Minimisation phase
Compute $\nabla_x^2 L(x_k, \Pi_k)$ or use a l-BFGS approximation [2]
Compute u_k by solving the horizontal subproblem.
Let $d_k = v_k + Z_k^- u_k$.
Compute the actual reduction a-red of the merit function
and the reduction p-red of the quadratic model
If $\frac{a-red}{p-red} \geq \eta$

2. limited memory BFGS, see [11].

Then let $\gamma_{k+1} = \gamma_k + d_k$, $\Delta_{k+1} \geq \Delta_k$
else let $\gamma_{k+1} = \gamma_k$, $\Delta_k < \|d_k\|$
End of loop : next k**.**

4.3 Trust region algorithm extended overview

The details of the trust region mechanism and merit function definition can be found in [2], 18.9.
We recall here some useful informations for understanding this report. Details on dogleg or Steihaug search methods should be found in [2].

Constants initialization: $\varepsilon > 0$ and $\eta, \xi \in (0,1)$.

Initialization of x_0, Δ_0 and μ_0.

For $k = 0, ...$
Compute J_k, Ψ_k, g_k, A_k.
Compute Z_k.

Compute multipliers Π_k by solving $(A_k^T A_k)\Pi_k = A_k^T g_k$. See [2], (18.16).
Conjugate gradient method for $A_k^T A_k$.
If $\|g_k - A_k \Pi_k\|_\infty < \varepsilon$ **and** $\|\Psi_k\|_\infty < \varepsilon$ **then stop.**

Restoration phase.
Compute vertical step v_k.
Option 1: Dogleg method of Powell.
Compute the Cauchy step v_k^c.
Compute the Newton step v_k^n.
Solve $A_k v + \Psi_k = 0$ by applying a right inverse of A_k,
$A_k^- = \begin{pmatrix} B_k^{-1} \\ 0 \end{pmatrix}$.
Option 2: Steihaug method.

Minimisation phase.
Compute W_k by using a l-BFGS approximation.
Compute the horizontal step u_k by the unscaled Steihaug iteration.
Let $d_k = v_k + Z_k u_k$. We use the merit function of Han: $\Phi(x) = J(x) + \mu\|\Psi(x)\|_2$.
Compute parameter μ^+.
Compute the actual reduction $a - red = J(x_k) - J(x_k + d_k) + \mu^+(\|\Psi(x_k)\| - \|\Psi(x_k + d_k)\|)$.
Compute the predicted reduction $p - red = -d_k^T g_k - \frac{1}{2} d_k^T W_k d_k + \mu(\|\Psi(x_k)\| - \|A_k^T v_k + \Psi(x_k)\|)$.

If $\frac{a-red}{p-red} \geq \eta$
Then $x_{k+1} = x_k + d_k$, $\mu_{k+1} = \mu^+$, and the trust region is enlarged.

Else if $\|v_k\| \leq 0.8\xi\Delta_k$ **et** $\|v_k\| \leq 0.1\|Z_k u_k\|$

Then Compute the second-order correction $d_{SOC} = d_k - A_k(A_k^T A_k)^{-1}\Psi(x_k + d_k)$.

Recompute $a - red$ for d_{SOC}.

If $\frac{a-red}{p-red} \geq \eta$

then $x_{k+1} = x_k + d_{SOC}$, $\mu_{k+1} = \mu^+$, and enlarge the trust region.

Else $x_{k+1} = x_k$, $\mu_{k+1} = \mu_k$, and reduce the trust region.

Else $x_{k+1} = x_k$, $\mu_{k+1} = \mu_k$, and reduce the trust region.

End of loop : next k**.**

4.4 Reduced-Hessian SQP algorithm for Optimum Design

The optimal design problem without constraints, when adjoint is available, allows the reduction of unknown by state equation solution. The reduced Hessian is not available and should be replaced by some BFGS device. State resolution can be performed with a Newton step, so that a *globally quasi-Newton* algorithm can be applied.
In [10], an optimum design problem is solved with a SQP due to Gill *et al.* [9]. In this section we recommend a standard algorithm under the formulation given in the monograph of Nocedal and Wright ([2], 18.7). The notations of the Aerodynamics optimum design problem described in Section 2 are systematically used.

Initialization of constants $\eta \in (0, 0.5), \tau \in (0, 1)$.

Initialization of x_0 and Π_0.

Initialization of an approximation of the reduced Hessian M_0, a symmetric positive definite $(n-m) \times (n-m)$ matrix. We can for example choose $M_0 = I$
Compute de J_0, Ψ_0, g_0, A_0.

For k=0,...

If $\|g_k - A_k \Pi_k\|_\infty < \varepsilon$ and $\|\Psi_k\|_\infty < \varepsilon$ then stop.

Restoration phase.
Null Space-Method.
Compute the vertical Newton step p_Y by solving: $(A_k A_k^-) p_Y = \Psi_k$.
or compute it by: $v_k = A_k^- p_Y = \begin{pmatrix} \left(\dfrac{\partial \Psi_k}{\partial W}\right)^{-1} \Psi_k \\ 0 \end{pmatrix}$ (36).

Minimisation phase.
Compute the tangent step p_Z by solving: $M_k p_Z = -Z_k^T g_k$, i.e.:

We look for p_Z such that $M_k p_Z = \left(\dfrac{\partial \Psi_k}{\partial \gamma}\right)^T \left(\left(\dfrac{\partial \Psi_k}{\partial W}\right)^{-1}\right)^T \dfrac{\partial J_k}{\partial W} - \dfrac{\partial J_k}{\partial \gamma} = \left(\dfrac{\partial \Psi_k}{\partial \gamma}\right)^T \Pi_k - \dfrac{\partial J_k}{\partial \gamma}$.

We need to invert M_k and $\dfrac{\partial \Psi_k}{\partial W}$.

Compute

$$h_k = Z_k p_k = \begin{pmatrix} -\left(\left(\dfrac{\partial\Psi_k}{\partial W}\right)^{-1}\right)\left(\dfrac{\partial\Psi_k}{\partial\gamma}\right) {M_k}^{-1}\left[\left(\dfrac{\partial\Psi_k}{\partial\gamma}\right)^T\left(\left(\dfrac{\partial\Psi_k}{\partial W}\right)^{-1}\right)^T\dfrac{\partial J_k}{\partial W}-\dfrac{\partial J_k}{\partial\gamma}\right] \\ {M_k}^{-1}\left[\left(\dfrac{\partial\Psi_k}{\partial\gamma}\right)^T\left(\left(\dfrac{\partial\Psi_k}{\partial W}\right)^{-1}\right)^T\dfrac{\partial J_k}{\partial W}-\dfrac{\partial J_k}{\partial\gamma}\right] \end{pmatrix}$$

$$\text{Let: } h_k = \begin{pmatrix} -\left(\left(\dfrac{\partial\Psi_k}{\partial W}\right)^{-1}\right)\left(\dfrac{\partial\Psi_k}{\partial\gamma}\right) {M_k}^{-1}\left[\left(\dfrac{\partial\Psi_k}{\partial\gamma}\right)^T\Pi_k-\dfrac{\partial J_k}{\partial\gamma}\right] \\ {M_k}^{-1}\left[\left(\dfrac{\partial\Psi_k}{\partial\gamma}\right)^T\Pi_k-\dfrac{\partial J_k}{\partial\gamma}\right] \end{pmatrix}$$

Let $p_k = A^-{}_k p_Y + Z_k p_Z = v_k + h_k$ (see [2] p.430, 450; [4] p.134 (10.6))

Choose parameter μ_k in such a way that p_k is a descent direction for the merit function Φ at x_k. It is more or less necessary that Φ be differentiable, and several solutions are proposed in [2] (18.33), (18.36), Th 18.3 and [3] 4.6.

Let $\alpha_k = 1$.
Linear search of Wolfe.[2] p.37
While $\Phi(x_k + \alpha_k p_k; \mu_k) > \Phi(x_k; \mu_k) + \eta\alpha_k D\Phi(x_k; p_k)$
Do $\alpha_k = \tau_\alpha \alpha_k$ pour un $\tau_\alpha \in (0, \tau)$.
End do while.

Let $x_{k+1} = x_k + \alpha_k p_k$.
Compute $J_{k+1}, \Psi_{k+1}, g_{k+1}, A_{k+1}$.

Compute adjoint state Π_{k+1}: $\Pi_{k+1} = \left(\left(\dfrac{\partial\Psi_k}{\partial W}\right)^{-1}\right)^T \dfrac{\partial J_k}{\partial W}$.
Let $s_k = x_{k+1} - x_k = \alpha_k p_k, y_k = Z_k^T(\nabla_x L_k(x_{k+1}, \Pi_{k+1}) - \nabla_x L_k(x_k, \Pi_{k+1}))$.
If the criterion is satisfied **then**
Standard BFGS :
Update M_{k+1} with the Sherman-Morrison-Woodburry formula ([4]p.185,[2] 8.1, [3] (4.10))
$M_{k+1} = M_k - \frac{M_k s_k s_k^T M_k}{s_k^T M_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$.
Limited memory BFGS: [4]p.72,[2] 9.1, [3] 4.4

Else $M_{k+1} = M_k$.
End if

End of loop : next k.

In the previous algorithms, the state variable is updated by only a Newton step, of, very frequently, by the Cauchy step, which is a kind of gradient step for the minimization of the norm of equation residual. This means that the effort devoted to the state equation resolution can be rather small at each restoration phase.
At the contrary, the adjoint state equation is assumed to be exactly solved. This may then represent the dominant effort in one iteration.
The next section is devoted to the reduction of state and adjoint solution effort in each iteration.

5 Towards one-shot

5.1 Motivations

In aerodynamics optimum design using RANS models, reducing computing costs and related latence are still an important issue. This is due to the important computing effort necessary to solve the state equation or the adjoint state equation.

One attack of this problem is, as shown above, to build a fast converging quasi-Newton algorithm. But this is done by applying an algebraic quasi-Newton process, such as BFGS, which by construction is more efficient for a rather small number of optimization variables.

A second standpoint relies on the reduction of the cost of one optimization iteration, trying to perform it with a computational cost smaller that a state or adjoint exact solution.
The one-shot principle ([12]) consists in replacing the solution phase for state of adjoint by a *partial* or *uncomplete* resolution of these systems.

A Newton step for the state equation:

$$v = -(\frac{\partial \Psi}{\partial W})^{-1} \Psi(x_k)$$

will be replaced by a *pseudo-Newton* one:

$$v = -S\ \Psi(x_k)$$

where S is a low cost operator approximating the inverse of the Jacobian $\frac{\partial \Psi}{\partial W}$. Typically S can be a few iteration of a artificial time advancing scheme, or a few SSOR iterations applied to a simplified (first-order spatial) Jacobian.
This simplification is also to be applied in the state update in the minimization phase.

The solution of adjoint system:

$$\Pi = (\frac{\partial \Psi}{\partial W})^{-T} \frac{\partial J}{\partial W}$$

will be replaced by:

$$p = S^T \left((\frac{\partial \Psi}{\partial W})^T \Pi - \frac{\partial J}{\partial W} \right).$$

This modification cannot be accounted by the previous SQP algorithm. We propose a new algorithm in which the adjoint system is considered as an extra (equality) constraint. The state and adjoint constraints will be restored in two restoration phases prior to the minimization phase. Robustness of the global algorithm will be managed by a trust region method.

5.2 Algorithm

N.B.: *The new algorithm presented now is based on heuristics and its use should be tried only after the necessary a priori theoretical analysis, that is in progress, has been completed.*

Initialization of constants $\varepsilon > 0$ and $\eta, \xi \in (0,1)$.
Initialization of x_0, Δ_0 and μ_0.

For k=0,...
If $\|g_k - A_k \Pi_k\|_\infty < \varepsilon$ **et** $\|\Psi_k\|_\infty < \varepsilon$ **then stop.**

State restoration phase.
For minimizing $\frac{1}{2}\|\Psi(x)\|^2$.
we consider the quadratic model $\frac{1}{2}\|\frac{\partial \Psi}{\partial W} v + \Psi(x_k)\|^2$.
Compute Cauchy step $v_k^c = \beta \left(\frac{\partial \Psi}{\partial W}\right)^T \Psi(x_k)$.
Compute pseudo Newton step $v_k^n = -\bar{S}\Psi(x_k)$.
Note that v is of dimension n and S of dimension $m \times m$.
Let $\bar{S} = \begin{pmatrix} S \\ 0 \end{pmatrix}$.
Compute vertical step v_k with the dogleg method.

Adjoint restoration phase.
For minimizing $\frac{1}{2}\|R\|^2 = \frac{1}{2}\|\left(\frac{\partial \Psi}{\partial W}\right)^T \Pi - \frac{\partial J}{\partial W}\|^2$ with respect to Π.
we consider the quadratic model:

$$\frac{1}{2}\|\left(\frac{\partial \Psi}{\partial W}\right)^T p + R_k\|^2 = \frac{1}{2}\|\left(\frac{\partial \Psi}{\partial W}\right)^T \Pi - \left(\frac{\partial \Psi}{\partial W}\right)^T \Pi_k + R_k\|^2$$

$$= \frac{1}{2}\|\left(\frac{\partial \Psi}{\partial W}\right)^T \Pi - \frac{\partial J}{\partial W}(x_k)\|^2$$

.

This functional is quadratic in Π.

Compute Cauchy step $p_k^c = \beta \left(\frac{\partial \Psi}{\partial W}\right) (-\left(\frac{\partial \Psi}{\partial W}\right)^T \Pi_k + R_k) = -\beta \left(\frac{\partial \Psi}{\partial W}\right) \frac{\partial J}{\partial W}(x_k)$.

Compute pseudo Newton step $p_k^n = S^T \left(\left(\frac{\partial \Psi}{\partial W}\right)^T \Pi - \frac{\partial J}{\partial W}\right)$.
Compute vertical step p_k with the dogleg method.

Phase de Minimisation.

Compute $\tilde{W}_k$ with an l-BFGS approximation.

Compute Cauchy step $u_k^c = -\beta^c \tilde{g} = -\beta^c \left(-\left(\frac{\partial \Psi}{\partial \gamma} \right)^T \Pi_k + \frac{\partial J}{\partial \gamma} \right)$.

Compute pseudo Newton step u_k^n.

Choose a partial resolution algorithm for $\tilde{W}_k u + \tilde{g}_k = 0$.

Compute horizontal step u_k with the dogleg method.

Let $d_k = v_k + Z_k^- u_k$.

Validation phase.

Updating the trust region size.

End of loop: next k.

5.3 The validation phase in details

For simplifying the merit function definition, state and adjoint equations are considered as a unique constraint.

We adopt the Han merit function formulation:
$\Phi(x,\Pi) = J(x) + \frac{\mu}{2}\left(\|\Psi(x)\| + \| \left(\frac{\partial \Psi}{\partial W}\right)^T (x)\Pi - \frac{\partial J}{\partial W} \| \right).$
where $\|.\|$ holds for L^2 norm.

The quadratic associated model is:

$$m_k(d,p) = d^T g_k + \tfrac{1}{2} d^T W_k d + \tfrac{\mu}{2}\left(\| \left(\frac{\partial \Psi}{\partial W}\right) v + \Psi(x_k)\| + \| \left(\frac{\partial \Psi}{\partial W}\right)^T p - R_k \| \right).$$

For the computation of parameter μ^+, we follow [3] p.694

$$\mu^+ = max\left(\mu_k, 0.1 + \frac{d_k^T g_k + \frac{1}{2} d_k^T W_k d_k}{\frac{1}{2}(\|\Psi_2(x)\| + \|R_k\| - \| \left(\frac{\partial \Psi}{\partial W}\right) v_k + \Psi(x)\| - \| \left(\frac{\partial \Psi}{\partial W}\right)^T p_k - R_k\|)} \right) \quad (46)$$

The validation phase writes then:

Compute actual reduction $a - red = \Phi(x_k, \Pi_k) - \Phi(x_k + d_k, \Pi_k + p_k)$.

$$\begin{aligned} a - red \quad = \quad & J(x_k) - J(x_k + d_k) + \\ \frac{\mu^+}{2}(\|\Psi(x_k)\| + \| \left(\frac{\partial \Psi}{\partial W}\right)^T (x_k)\Pi_k - \frac{\partial J}{\partial W}(x_k)\| - & \\ \|\Psi(x_k + d_k)\| - \| \left(\frac{\partial \Psi}{\partial W}\right)^T (x_k + d_k)(\Pi_k + p_k) - \frac{\partial J}{\partial W}(x_k + d_k)\|). & \quad (47) \end{aligned}$$

Compute predicted reduction $p - red = m_k(0,0) - m_k(d_k, p_k)$.

$$\begin{aligned} p - red \quad = \quad & -d_k^T g_k - \frac{1}{2} d_k^T W_k d_k + \\ \frac{\mu^+}{2}\left(\|\Psi(x)\| + \|R_k\| - \| \left(\frac{\partial \Psi}{\partial W}\right) v_k + \Psi(x)\| - \| \left(\frac{\partial \Psi}{\partial W}\right)^T p_k - R_k\| \right). & \quad (48) \end{aligned}$$

If $\frac{a-red}{p-red} \geq \eta$
then $x_{k+1} = x_k + d_k$, $\mu_{k+1} = \mu^+$, and the trust region is enlarged.

$\Pi_{k+1} = \Pi_k + p_k$.

else if $\|v_k\| \leq 0.8\xi\Delta_k$ **and** $\|v_k\| \leq 0.1\|Z_k u_k\|$

then Compute the second-order correction

Recompute $a - red$ for d_{SOC}.

If $\frac{a-red}{p-red} \geq \eta$

then $x_{k+1} = x_k + d_{SOC}$, $\mu_{k+1} = \mu^+$, and the trust region is enlarged.

$\Pi_{k+1} = \Pi_k + p_k$.

else $x_{k+1} = x_k$, $\mu_{k+1} = \mu_k$, and trust region is reduced.

$\Pi_{k+1} = \Pi_k$.

else $x_{k+1} = x_k$, $\mu_{k+1} = \mu_k$, and trust region is reduced.

$\Pi_{k+1} = \Pi_k$.

6 Inequality constraints

In the mid-80's and in 90's a new family of optimization methods for *inequality constraints* has arisen. These methods are refered as *interior point methods*, in contrast to the *simplex algorithm* for linear programming of Dantzig, which iterates on the boundary of admissible parameters and dates back to the 60's.

Let us consider the solution of the following nonlinear programming problem

$$\begin{aligned} \min \quad & \phi(z) && (49)\\ \text{subject to} \quad & c_E(z) = 0 && (50)\\ & c_I(z) \leq 0. && (51) \end{aligned}$$

By introducing *slack variables* we obtain the following problem in the variables z and σ,

$$\begin{aligned} \min \quad & \phi(z) && (52)\\ \text{subject to} \quad & c_E(z) = 0 && (53)\\ & c_I(z) + \sigma = 0 && (54)\\ & \sigma \geq 0 && (55)\\ & && (56) \end{aligned}$$

This transformation does not discard the main difficulty, which is the identification of active constraints. To avoid this problem, interior points essentially work by keeping the iterate out of the set where constraints could be active. In [1], SQP, trust region and interior points are combined by choosing for the interior points a *barrier function* approach:

$$\begin{aligned} \min \quad & \phi(z) - \mu \sum_{i=1}^{m_1} ln\sigma^{(i)} && (57)\\ \text{subject to} \quad & c_E(z) = 0 && (58)\\ & c_I(z) + \sigma = 0 && (59) \end{aligned}$$

where μ is a penalty parameter and where components of σ are assumed to be strictly positive. The way to make μ go to zero is defined and analysed in [1].

Problem (57) has only equality constraints and can be solved with the methods of the type defined in the previous sections of this report. The main novelty is that only a subset of equality constraints constitutes an invertible "state equation".
Let us exhibit the corresponding KKT condition. We denote by e the column vector of dimension m_1 with 1 as all entries, and S the diagonal $m_1 \times m_1$

with $\sigma^{(k)}$ as k-th diagonal coefficient. The KKT conditions for Problem (57) gives rise to the following system of nonlinear equations in $z, \sigma, \lambda_E, \lambda_I$, where λ_E and λ_I are the Lagrange multipliers for the two equality constraints:

$$\begin{pmatrix} \nabla\phi(z) + A_E(z)\lambda_E + A_I(z)\lambda_I \\ -\mu S^{-1}e + \lambda_I \\ c_E(z) \\ c_I(z) + \sigma \end{pmatrix} = 0. \tag{60}$$

Let us restrict us to the aerodymamics Optimal Design case where:

$$z = (\gamma, W), \quad c_E(z) = \Psi(\gamma, W), \quad \phi(z) = J(\gamma, W). \tag{61}$$

The first line of (60) splits into one line for γ derivative ($\Pi = \lambda_I$):

$$\frac{\partial J}{\partial \gamma}\delta\gamma + \Pi\frac{\partial \Psi}{\partial \gamma}\delta\gamma + \lambda_I\frac{\partial c_E}{\partial \gamma}\delta\gamma = 0,\ \forall\delta\gamma \tag{62}$$

and one line for the W derivative:

$$\frac{\partial J}{\partial W}\delta W + \Pi\frac{\partial \Psi}{\partial W}\delta W + \lambda_I\frac{\partial c_E}{\partial W}\delta W = 0,\ \forall\delta W. \tag{63}$$

We see that with this formulation, we do not need to compute the sensitivity of the inequality state constraints as in methods where γ is the only one variable. Instead, the adjoint equation (63) for Π takes the inequality constraints into account in its left-hand side, and the algorithm can run with only one adjoint equation, of same dimension as the state equation.

Newton's method applied to the KKT system gives:

$$\begin{pmatrix} \nabla^2_{xx}L & 0 & A_E & A_I \\ 0 & \mu S^{-2} & 0 & I \\ A_E^T & 0 & 0 & 0 \\ A_I^T & I & 0 & 0 \end{pmatrix}\begin{pmatrix} d_x \\ d_\sigma \\ d_{\lambda_E} \\ d_{\lambda_I} \end{pmatrix} = \begin{pmatrix} \nabla\phi + A_E\lambda_E + A_I\lambda_I \\ -\mu S^{-1}e + \lambda_I \\ c_E(z) \\ c_I(z) + \sigma \end{pmatrix} \tag{64}$$

In practise we would replace the Hessian $\nabla^2_{xx}L$ by an l-BFGS matrix. We do not have anxiety concerning the invertibility of (64). Indeed, the system is more or less triangular. We can use the third equation for updating the state variable W, then the fourth equation for updating σ, then the second equation for updating λ_I. The adjoint state Π is then updated with (63). The design parameter γ is updated by (62)

7 Conclusions

This report is a very preliminary one. We have proposed an overview of SQP methods with the standpoint of applying them to a generical Aerodynamics Optimum Design problem.

SQP methods are rather well adapted to this context, under their reduced-Hessian version.

Their robustness can be much improved by introducing the trust region method in the (reduced-Hessian or not) SQP algorithm.

But even in these conditions, the adjoint system has to be completely solved at each optimization iteration. In order to relax this point without loosing robustness, we have started to design a trust-region-one-shot version in which both state and adjoint systems can be only partially solved. This is still at reflexion and analysis stage.

The case of inequality constraints is also considered. We recommend for treating them an extended SQP method that involves interior points and trust region. This method answers to the question of the possible extra cost induced by extra adjoint states to compute. Indeed, with the barrier treatment of interior points, we need to compute only one adjoint state.
This carries some argument against keeping the old optimization algorithms, which worked well in combination with divided differences, but which need derivatives for each functional (cost and state inequality constraints) and therefore need computing adjoints for each of them.

This report contains all the references describing in details the optimization algorithms refered in the text. Implementations are also available for most of the algorithms (not the new one under study).

All the above points will be further examined in the sequel of our study but we expect technical exchanges partners on the basis of this report, i.e. as soon as possible after its delivery.

Références

[1] R.H. Byrd, J.-C. Gilbert and J. Nocedal, A trust region method based in interior point techniques for nonlinear programming, *Mathematical Programming*, 89 (2000) 149-185. also version available as INRIA RR 2896, http://www.inria.fr/rrrt/rr-2896.html,1996

[2] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer (1999).

[3] M. Lalee, J. Nocedal; T. Plantenga, On the implementation of an algorithm for large-scale equality constrained optimization, *SIAM J. Optim.* 8 (1998), no. 3, 682–706.

[4] J.-F. Bonnans, J.-C. Gilbert, C. Lemaréchal and C. Sagastizabal, *Optimisation numérique. (in French) [Numerical optimization] Aspects théoriques et pratiques. [Theoretical and applied aspects]* Mathématiques et

Applications [Mathematics et Applications], 27. Springer-Verlag, Berlin, 1997.

[5] A. Vardi, A trust region algorithm for equality constrained minimization: convergence properties and implementation. *SIAM J. Nume Anal.* 22 (1985), no. 3, 575–591.

[6] M.R. Celis, J.E. Dennis and R.A. Tapia, A trust region strategy for non-linear equality constrained optimization. Numerical optimization, 1984 (Boulder, Colo., 1984), 71–82, SIAM, Philadelphia, Pa., 1985.

[7] R.H. Byrd, Robust trust region methods for constrained optimization, Houston, TX, May 1987.

[8] E.O. Omojokun, Trust Region Algorithms for Optimization with Non-linear Equality and Inequality Constraints, *Ph.D. thesis*, Dept. of Computer Science, University of Colorado, Boulder, 1989.

[9] P. Gil, M. Saunders and W. Murray, SNOPT: an SQP algorithm for large scale constrained optimization. Report 97-2m, Dep. of Math., University of California, San Diego, 1997

[10] K. Maute, N. Nikbay and C. Farhat, Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. CU-CAS-00-23, Center for Aerospace structures, University of Colorado, Boulder, november 2000

[11] D.C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Programming* Ser.B, 54, 503-528 (1989)

[12] S. Ta'asan, One shot methods for optimal control of distributed parameter systems I: finite dimensional control *ICASE Report*, 91-2 (1991)

[13] F. Beux, Shape optimization of an Euler flow in a nozzle. in *Notes on numerical fluid mechanics*, vol. 55, Periaux et al Eds, EUROPT - A European Initiative on Optimum Design Methods in Aerodynamics, pp. 115-131, Vieweg, Braunsweig Wiesbaden, 1994

[14] N. Marco and A. Dervieux, Numerical optimizers for aerodynamic design using transonic finite-element solvers. – BRITE-ECARP Final report, July 1995.