1

1

Numerische Mathematik manuscript No.
(will be inserted by the editor)

MULTILEVEL FUNCTIONAL PRECONDITIONING FOR SHAPE OPTIMISATION

F. Courty[1], **A. Dervieux**[1]

INRIA, Centre de Sophia-Antipolis,
BP 93, 2004 Route des Lucioles,
06902 Sophia-Antipolis Cedex, France
Email: Francois.Courty@inria.fr Email: Alain.Dervieux@Inria.fr
web page: http://www-sop.inria.fr/tropics/Francois.Courty/index.html

Received: date / Revised version: date

Summary The additive multilevel method is extended to a general preconditioner, that applies on unstructured meshes, and addresses differential and pseudo-differential optimality systems. Application to the optimal design of 3D aerodynamic shapes is demonstrated.

Key words: multilevel, optimal shape design, partial differential equations, finite element, unstructured mesh.

1 Introduction

During the last two decades, researchers have looked for efficient solvers for CFD problems. Now that efficient CFD solvers are available, many researchers are focusing on solving Optimum Shape design problems. The vast majority use gradient-based methods, which have gained efficiency thanks to the use of adjoint states. Adjoint states are easier to develop thanks to the progress of Automated Differentiation tools, see for example [13]. But gradient-based methods generally re-

⋆ *Present address:* INRIA, Centre de Sophia-Antipolis,
BP 93, 2004 Route des Lucioles,
06902 Sophia-Antipolis Cedex, France
Send offprint requests to:

quire preconditioners. To illustrate this, consider the application of gradient iteration in a space of functions. The functional analysis tells us that the continuous iterate $u^{n+1} = u^n - \rho g^n$ is less regular than u^n when the gradient g^n involves k-th derivatives of u^n. In other words the iteration operator is unbounded, with a *regularity loss* equal to k. The iteration then diverges. In the discrete case, this results in the amplification of high frequency error components, except for very small, mesh dependant, step lengths ρ.

Experiences show that this problem is very frequent in Optimal Shape Design. In this paper, we propose an explanation of this, related to the lack of smoothness of the Hadamard derivative of a function with respect to the domain shape. As a practical consequence, the discrete gradients to be used for updating shapes are oscillatory. To cure this, Jameson has proposed in [21] a method of correction smoothing that solves a Laplace-Beltrami system set on the boundary. This method is also used in [16]. In [1], Arian and Ta'asan analyse the Fourier symbol of the Hadamard-type functional Hessian. According to the degree of this Hessian, these authors propose to use either a Laplace-Beltrami operator when the regularity loss is 2 or a Neumann-Dirichlet pseudo-differential operator when regularity loss is 1. This latter method can be more expensive, since 3D systems are then solved.

In this paper, we propose to push further the idea of Arian and Ta'asan by using a family of preconditioners for which the regularity gain is a parameter that can be specified according to the problem to solve. For this, we adapt the Bramble-Pasciak-Xu (BPX) additive multilevel preconditioner (see [3,25,5]). In order to connect its design with the regularity loss, we present it directly under a functional standpoint. Some interesting properties of the functional version are easily derived from existing results.
The discrete variant is adapted to unstructured meshes by applying an agglomeration principle [15].
It remains to investigate how this preconditioner is used for solving optimisation problems. Starting with 1D examples, we first examine the introduction of this kind of preconditioner in a popular class of quasi Newton optimisation algorithm, namely SQP methods relying on the BFGS technique.
Then we consider the introduction of our method in a pre-industrial shape optimisation loop introduced in [22] and addressing the aerodynamical and sonic boom optimization of supersonic aircraft geometries.

Contents

2 Functional preconditioning

Large scale problems coming from Partial Differential Equations generally result in a poor conditioning that further degrades when the number of degrees of freedom is increased. To explain and solve this problem, we can either analyse directly the behavior of discrete eigenvalues when the number of unknowns is increased or analyse the continuous -functional- problem and the continuous version of the algorithm to apply. We concentrate on the second way.

2.1 Optimization model problem

Let us consider the minimization of the following functional:

$$j \ : \ H_0^1(0,1) \rightarrow I\!R \tag{1}$$

$$j(u) = \frac{1}{2} \int_0^1 \ (\ k|grad \ u|^2 \ - \ f \ u) \ dx \tag{2}$$

where k is a positive smooth function on $[0,1]$. We first observe that the *formal gradient* iteration

$$u^{n+1} \ = \ u^n \ - \ \rho(div \ k \ grad \ u^n \ - \ f) \tag{3}$$

would produce a new iterate u^{n+1} that is not smooth enough to compute Δu^{n+1} in the next iteration since Δu^n is generally not in $H_0^1(0,1)$.
This is bad news for the discrete version:

$$u_h^{n+1} \ = \ u_h^n \ - \ \rho((div \ k \ grad)_h \ u_h^n \ - \ f_h) \ . \tag{4}$$

Iteration(4) is essentially the **Jacobi iteration**. Its convergence strongly depends on mesh size.
Conversely, we can introduce the canonical isomorphism associated to the usual H^1 norm:

$$\Lambda \ : \ (H_0^1(0,1))' \longrightarrow H_0^1(0,1)) \tag{5}$$

defined as follows:

$$(g,u)_{(H_0^1(0,1)))'\times H_0^1(0,1))} \ = \ <\Lambda g, u>_{H_0^1} \ = \ \int (grad \ \Lambda g).(grad \ u) \ dx \tag{6}$$

and explicitly computed by solving the *Poisson equation*:

$$div\ grad\ \Lambda g \ = \ g \ \ on \ \ \Omega, \ \ \Lambda g \ = \ 0 \ \ on \ \ \partial\Omega. \tag{7}$$

Then the *functional gradient* method writes:

$$u^{n+1} \ = \ u^n \ - \ \rho \Lambda j'(u^n), \tag{8}$$

which gives:

$$u^{n+1} \ = \ u^n \ - \ \rho \Lambda (div\ k\ grad\ u \ - \ f). \tag{9}$$

Iteration (9) is a Functional Least Square algorithm according to [4].

The main motivation for doing this is that the continuous algorithm has a convergence rate that does not depend on the mesh size. Then we can try to build some consistent discretization that will hopefully converge with a rate not so different from the continuous rate. This means that *essentially mesh-independent rates* might be obtained. Another way to understand that point is to remember that in the linear periodic case, Fourier's analysis shows that operators involving spatial derivatives will have large eigenvalues of the order of the inverse of mesh step at power the degree of the spatial differentiation. The smaller this degree, the better the condition number. It is clear that the two degrees of spatial differentiation lost by (3) are compensated in (9) by the introduction of Λ.
In other words, we shall modify the iteration in such a way that **the resulting order of spatial differentiation after preconditioning be equal to zero**.

2.2 An example : Hadamard formula and regularity issues

We want to minimize a functional $j(\gamma)$ with respect to a "shape parameter" γ. More precisely, in an optimal shape design problem, an initial geometry Ω_0 is equipped with a direction V normal to its boundary. A family of domains Ω_γ of R^d is parameterized by a displacement $\gamma \in \mathcal{C}^{l+\alpha}(\partial\Omega_0)$ of the boundary in the normal direction V.
The state equation is the Poisson problem:

$$-div\ grad\ z(\gamma) \ = \ f \ \ ; \ \ z(\gamma) \ = \ 0 \quad on \quad \partial\Omega_\gamma \ .$$

Let D be a subdomain of Ω_γ (inside Ω_γ for any γ admissible). The functional j to minimize is defined by:

$$j(\gamma) \ = \ \frac{1}{2}||z(\gamma) - z_{target}||^2_D \ .$$

A classical difficulty for computing the gradient of j comes from the variable domain in the state equation. We choose a family of diffeomorphisms $(T_\gamma)_\gamma$ such that:

$$T_\gamma \ \ maps \ \ \Omega_0 \ \ on \ \ \Omega_\gamma ,$$
$$T_\gamma \ \ is\ identity \ \ on \ \ D \ .$$

Then it is possible to adapt some well known results related to the Garabedian "interior variation" method (see [10], [17], [20], [8], [9]) and to show, under regularity assumptions on f and $\partial\Omega_0$, that the mapping:

$$\gamma \rightarrow z(\gamma)|_D$$

is continuously differentiable from $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$ in $\mathcal{C}^{l-1+\alpha}(D)$.
In fact the gradient of this mapping is the solution of a Dirichlet problem with a non-homogeneous Dirichlet condition expressed as a function of the normal derivative of $z(\gamma)$ and of the boundary perturbation. We recognize in the mapping:

$$\gamma \rightarrow \frac{\partial z}{\partial n}(\gamma)|_{\partial\Omega_\gamma}$$

a pseudo-differential operator close to the Dirichlet-Neumann classical one. By the chain rule j is also differentiable and its gradient is expressed as follows:

$$j'(\gamma).\delta\gamma \ \ = \ \ \int_{\partial\Omega_\gamma} \frac{\partial z(\gamma)}{\partial n_\gamma} \ \frac{\partial p(\gamma)}{\partial n_\gamma} \ \ < n_\gamma, n_0 > \delta\gamma \ \ d\partial\Omega_\gamma$$

where n_γ is the normal to $\partial\Omega_\gamma$, and $p(\gamma)$ the following adjoint state:

$$-div\ grad\ p(\gamma) \ = \ z_{target} - z(\gamma) \ \ ; \ \ p(\gamma) \ = \ 0 \ \ \ on \ \ \ \partial\Omega_\gamma \ .$$

Taking the L^2 space as pivot space for our gradient method would produce the following gradient:

$$g_{L^2}(\gamma) \ \ = \ \ \frac{\partial z(\gamma)}{\partial n_\gamma} \ \frac{\partial p(\gamma)}{\partial n_\gamma} \ \ < n_\gamma, n_0 > \ \ .$$

We observe that this correction to the parameter γ, belonging to $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$, is only of regularity $\mathcal{C}^{l-1+\alpha}(\partial\Omega_0)$. This is not regular enough if we want to iterate in $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$. For this choice of functional Banach spaces, our analysis shows that the loss of regularity is 1.

Remark 1: For applying a gradient iteration, we should prefer a Hilbert space context, although this might be not strictly necessary (see the discussion of [11] for this point). Let us however work with Sobolev spaces. In that case, we have not an optimal analysis. From the Sobolev injection theorem, we have a differentiable application :

$$j : H^{l+\frac{d-1}{2}+\varepsilon+\alpha}(\partial\Omega_0) \to I\!R$$

where d is the dimension of space and ε a small positive number, and:

$$g_{L^2} \ \in \ H^{l-1+\alpha}(\partial\Omega_0).$$

Therefore, in this Hilbert context, the loss of regularity between γ and its correction $g_{L^2}(\gamma)$ is at most $1 + \frac{d-1}{2} + \varepsilon.\Box$

The example of this section shows that in complex optimization problems, and similarly to the above elliptic example, the direct application without canonic isomorphism of a gradient method may result in non-regular continuous iterations, and then may induce in the discrete version some behaviors analogous to Jacobi or unpreconditioned Jacobi-type elliptic solvers. Like in Sec. 1, we need a preconditioner for compensating the loss of regularity.

3 Additive multilevel preconditioners

Additive multilevel preconditioners have been initially derived in a discrete context for solving elliptic Partial Differential systems which are typically of second order or of even order. An extremely rich litterature exists on this topic. The hierarchical basis method was first analysed by Yserentant in his pioneering paper [26]. The work of Yserentant was apparently motivated by the famous unpublished technical report of Bank and Dupont [2]. A more complete theory was proposed by Bramble, Pasciak and Xu [3], [24]. See also the wavelet extensions, for example in [5]. An extended theory can be found in [14,25].
The purpose of this section is to recall basic notations of [25] and to reformulate in a more adapted statement the main result which we want to use in the sequel. For simplicity, we state it in the case of Dirichlet boundary conditions.
Let $V = H_0^1(\Omega)$ included in $H = L^2(\Omega)$, the second space being equipped with the scalar product and the norm:

$$(u,v) \ = \ \int_\Omega u \ v \ \ dx; \ \ \|u\| = \ (u,u)^{1/2}$$

Let $(V_k)_{k=0,1,...}$ be a sequence of subspaces of V :

$$V_0 \subset \cdots \subset V_k \subset \cdots \subset V$$

This sequence of spaces is assumed to satisfy some adequate properties expressing that they are good approximation spaces for V, with mesh size ratio of 2 between V_k and V_{k-1}, see [25]. The uniform mesh size in V_k is assumed to be:

$$h_k \;\; = \;\; 2^{-k}$$

For any k we introduce the operators $Q_k : V \longrightarrow V_k$ defined for all $u \in V, v \in V_k$ by

$$(Q_k u, v) = (u, v) \;;\; Q_{-1} = 0 \;.$$

For convenience, following [25], $x \;\approxeq\; y$ means that $cx \;\leq\; y \;\leq\; Cx$ for some constants C and c that are independent of level index k.

Theorem [25]:

Let $s \;\in\; I\!R, \;\; -3/2 \;\leq\; s \;\leq\; 3/2$.
For all $v \;\in\; H_0^s(\Omega)$, we have

$$\|v\|^2_{H^s(\Omega)} \;\approxeq\; \sum_{k=0}^{+\infty} \|(Q_k - Q_{k-1})v\|^2_{H^s(\Omega)} \;\approxeq\; \sum_{k=0}^{+\infty} h_k^{-2s} \|(Q_k - Q_{k-1})v\|^2.$$

□

Let $a \in [0, 3/2]$ and B defined by:

$$B \;=\; \sum_{k=0}^{+\infty} \left(\frac{1}{2^a}\right)^k \;(Q_k - Q_{k-1}) \tag{10}$$

Let u be an element of $H_0^s(\Omega)$ and s such that $-3/2 \;\leq\; s \;\leq\; 3/2$.
Using the Theorem above, we deduce

$$\|u\|^2_{H^s(\Omega)} \;\approxeq\; \sum_{k=0}^{+\infty} \|(Q_k - Q_{k-1})u\|^2_{H^s(\Omega)} \;\approxeq\; \sum_{k=0}^{+\infty} h_k^{-2s} \|(Q_k - Q_{k-1})u\|^2,$$

$$\|u\|^2_{H^s(\Omega)} \;\approxeq\; \sum_{k=0}^{+\infty} h_k^{-2(s+a)} \,(h_k^a)^2 \, \|(Q_k - Q_{k-1})u\|^2,$$

$$\|u\|^2_{H^s(\Omega)} \;\approxeq\; \sum_{k=0}^{+\infty} h_k^{-2(s+a)} \|(Q_k - Q_{k-1})Bu\|^2.$$

Applying again the Theorem above, we derive the following result:

Corollary [6] :

Let s be such that $-3/2 \ \leq \ s \ \leq \ s+a \ \leq \ 3/2$.
Then the operator B is bounded from $H_0^s(\Omega)$ into $H_0^{s+a}(\Omega)$ and for all $u \ \in \ H_0^s(\Omega)$, we have

$$\|Bu\|_{H^{s+a}(\Omega)} \ \approxeq \ \|u\|_{H^s(\Omega)} \ .$$

□

This statement shows that the operator B has smoothing properties that can be quantified as a gain in regularity. This gain is exactly a. It can be prescribed by the user in order to precondition an operator having a as regularity loss. Thanks to this continuous standpoint, the only adaptation we shall apply to a multilevel preconditioner for making it adequate to a given operator will be to choose an exponent a that compensate the loss of derivative in that operator. We have not yet precisely defined the V_k's. We introduce in the next sections a non classical construction of these spaces.

4 1D Preconditioner by node agglomeration

The agglomeration method for building additive multi-level preconditioners applying to equations was introduced in [18]. The purpose of this section is to consider its adaptation in 1D to optimization problems.

4.1 Operator construction

Let us consider a sequence of discretizations of $[0,1]$ obtained by dividing the interval in 2^k subintervals, with $1 \leq k \leq N$.
Let $h = \frac{1}{2^k}$ and $\mathcal{M}_h$. We concentrate on building the transfer operator between $\mathcal{M}_h$ and $\mathcal{M}_{2h}$.

In a first step, we define a sequence of **non-smooth spaces**. Each function on $[0,1]$ that we consider is defined by its values in *fine nodes* of level N and a continuous linear interpolation on fine mesh of level N. Let j be a coarse cell, it contains two fine cells, $2j$ and $2j-1$.

We define $\overline{P}$ the linear prolongation operator from $E_{2h} = V_{k+1}$ to $E_h = V_k$ as follows:

$$\forall u_{2h} \in E_{2h} \quad (\overline{P} u_{2h})_{2j} \;=\; (\overline{P} u_{2h})_{2j-1} \;=\; (u_{2h})_j \tag{11}$$

The above transfer defines non-smooth spaces generated by step functions. The adjoint operator, $\overline{P}^*$, is the restriction operator from E_h to E_{2h}. If two fine cells, $2j$ and $2j-1$, are included in the coarse cell j, we put:

$$(\overline{P}^* u_h)_j = \frac{1}{2}\left[(u_h)_{2j-1} \;+\; (u_h)_{2j}\right] \;. \tag{12}$$

We restrict to the case of differential (or pseudo-differential) operators with degree from zero to two. In contrast with the usual Galerkin nested finite element sequence, the previous sequence of spaces generated with the tranfers $\bar{P}$ and $\bar{P}^*$ does not enjoy the smoothness properties necessary to build an efficient multi-level preconditioner for second-order operators. This point is discussed in detail in [18]. It is related to the fact that the orders of accuracy of transfers have a sum equal to $1+1=2$, not strictly larger than the order 2 of the operator, see [12].
Let us now define **smoother transfers**. It is proposed in [18], to recover the necessary smoothness by introducing an "averaging operator" L and its adjoint L^* with respect to the discrete L^2 scalar product.

$$(Lu)_j = \;\frac{1}{4}\; u_{j-1} \;+\; \frac{1}{2}\; u_j \;+\; \frac{1}{4}\; u_{j+1} \tag{13}$$

Then we put :

$$V_k \;=\; L_k\; \bar{P}_k\; (V_{k+1}) \tag{14}$$

or equivalently

$$Q_m \;=\; \Pi_{k=0}^{m-1} L_k\; \bar{P}_k\; \Pi_{k=m-1}^{k=0}\; \bar{P}_k^*\; L_k^*. \tag{15}$$

Then we define the preconditioner by:

$$B \;=\; \sum_{k=0}^{N} \left(\left(\frac{1}{2^a}\right)^k \;-\; \left(\frac{1}{2^a}\right)^{k-1} \right) Q_k \;. \tag{16}$$

4.2 Some experiments

We want to emphasize that this kind of preconditioner can be used for solving a symmetric linear system inside a conjugate gradient as well as for finding a minimum in combination with a quasi-Newton minimization algorithm.
In the first case, we can use only once per iteration the **second-order preconditioner** ($a = 2.$) inside a Polak-Ribiere conjugate gradient for solving the Laplace equation. Figure 1 show that the optimal value of parameter a is indeed 2, and that for this optimal value we get a mesh-independant convergence rate.
In the second -optimization- case, we apply a quasi-Newton SQP-BFGS optimizer (see for example in [19]) on the same problem. The novelty is that, for updating both approximate Hessian and inverse in the BFGS formula, we need to apply the variable changes. They consist of **first-order preconditioner** and its inverse, built respectively with a equal to 1 and -1. In Fig. 2 four experiments are depicted. The number of unknowns is first $m = 32$. As expected the standard SQP-BFGS solves the optimum in about $5 + m/2$ iterations, that is 21 iterations. The preconditioned version has a convergence which is more progressive, but finally not faster. But for a larger number of unknowns, $m = 256$, the unpreconditioned SQP-BGFS has a very slow convergence. In contrast, the preconditioned SQP-BFGS convergence is about the same as for the coarse-mesh one.

5 Unstructured meshes

We give now the definition of the agglomeration-based additive multilevel preconditioner on unstructured meshes. The first section describes the case of a 2D or 3D mesh. The second section adresses the particular features of an unknown defined on a non-plane surface discretised by triangles in 3D.

5.1 Multidimensional Agglomeration

Let us recall briefly the definitions of the transfer operators $\overline{P}$ and $\overline{P}^*$ between two levels, and the definitions of L and L^*. We start from a fine triangulation or tetrahedrization. Nodes are located at vertices. The discrete fine level is the subspace of linear combinations of $P1$ shape functions: $E_h = span\{\varphi_i,\ i = 1 \cdots n_h\}$.
E_h is equipped by the following mass lumped scalar product, i.e. that

is weighted by the dual cell measures $\mathcal{M}eas\ (i)$:

$$\forall\ u_h\ and\ v_h \in E_h \quad (u_h, v_h)_h = \sum_i^{n_h} (u_h)_i\ (v_h)_i\ \mathcal{M}eas\ (i) \tag{17}$$

The agglomeration process relies on a partition of the set $I^f = \{1, \cdots, i, \cdots, n_h\}$ of fine indices i.

$$I^f = I_1 \cup \cdots \cup I_J \cup \cdots \cup I_{n_{2h}} \quad (n_{2h} \ll n_h) \tag{18}$$

where any I_J involves the indices of a few neighbouring nodes. An algorithm for building such a partition can be found in [15]. For any I_J, a coarser basis function is defined by: $\Phi_J = \sum_{i \in I_J} \varphi_i$ and the coarser space is given by:

$$E_{2h} = Span\{\Phi_J,\ J = 1 \cdots n_{2h}\}$$

The linear prolongation operator, $\overline{P}$, from E_{2h} to E_h is defined by

$$\forall u_{2h} \in E_{2h} \quad \overline{P}u_{2h} = u_{2h}\ \in\ E_h\ . \tag{19}$$

Its adjoint $\overline{P}^*$ is the restriction operator from E_h to E_{2h}, and it is defined via $\overline{P}$ as its adjoint from the previous scalar product by :

$$(\overline{P}^* u_h)_J = \frac{\sum_{j_m \subset J}^{n_h} (u_h)_{j_m} \mathcal{M}eas(j_m)}{\mathcal{M}eas(J)} \tag{20}$$

where J is a coarse cell, j_m are fine cells included in J. $\mathcal{M}eas(J)$ represents the measure of coarse cell J . Similarly to the 1D case we need to introduce a smoothing operator. This “smoothing” operator is an average between a node and its neighbors. It writes:

$$(Lu)_i = (1-\theta)u_i + \theta\ \frac{\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathcal{M}eas(j)\ u_j}{\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathcal{M}eas(j)} \tag{21}$$

where $\mathcal{N}(i)$ represents the set of neighbors of cell i and θ is the smoothing parameter. The adjoint L^* of L is defined by:

$$(L^* u)_i = (1-\theta)u_i + \theta \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{u_j\ \mathcal{M}eas(j)}{\sum_{k \in \mathcal{N}(j) \cup \{j\}} \mathcal{M}eas(k)}\ . \tag{22}$$

The rest of the construction of the preconditioner follows the rules in (15), (16).

5.2 Agglomeration for a surface in 3D

We return to the notations of Sec. 2.2.

Let Σ_0 be the initial 3D surface, made of triangles. The generic discrete surface Σ_γ, is defined by the translation of length γ, of the vertices of Σ_0 along an approximate unit normal vector $\mathbf{n}$ to Σ_0 defined at vertices.

$$\mathbf{x}_i^\gamma \text{ is a vertex of } \Sigma_\gamma \quad \Leftrightarrow \quad \mathbf{x}_i^\gamma = \mathbf{x}_i^o + \gamma(i)\ \mathbf{n} \tag{23}$$

where i is the index of the vertex, $\mathbf{x}_i^o$ is the physical position of the vertex of Σ_0 with same index i. In order to precondition a correction on γ, we can construct a sequence of spaces and operators following the same process as in (18, 19, 20), but restricted to the surface and with the area of surfacic cells $\mathcal{A}rea(j)$ instead of cell measures. In order to adapt our operators to irregular surfaces, the smoothing operator L, is now weighted by a scalar product of normals to the surface:

$$(L\ \mathbf{x})_i = (1-\theta)\mathbf{x}_i + \theta\ \frac{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}\mathbf{x}_j}{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}} \tag{24}$$

where w_{ij} are the weights defined by :

$$w_{ij} = \max\ (\mathcal{A}rea(i)\mathbf{n}_i \cdot \mathcal{A}rea(j)\mathbf{n}_j\ ,\ 0) \quad \|\mathbf{n}_i\| = 1\ \ \forall\ i\ . \tag{25}$$

With this formula, the smoothing is cancelled on geometry diedra (e.g. on wing trailing edges). The rest of preconditioner definition is the same as in previous section. The smoothing parameter θ will be set in practice equal to $\frac{1}{2}$. Since the parameterized shapes to be considered are closed curves, we shall not need any boundary condition.

6 Application to an optimal shape design problem

We now describe the adaptation of the proposed multi-level preconditioner to an optimal shape design loop. This loop is presented and motivated from the CFD standpoint in [22]. We recall this in short.

6.1 The aerodynamical shape design problem

In [22], the authors propose to measure the "sonic boom downwards emission" (SBDE) by evaluating the volume integral of the squared pressure gradient in an "observation box" Ω^B (as shown in Figure 3) below the object. The cost functional is therefore the following:

$$j(\gamma) = \alpha_1(C_D - C_D^t)^2 + \alpha_2(C_L - C_L^t)^2 + \alpha_3 \int_{\Omega^B} |\nabla p|^2 dV \quad (26)$$

where α_1, α_2 and α_3 are constants that prescribe the weights of three sub-criteria, related to lift, drag and sonic boom emission. They are evaluated by solving the 3D Euler equations around the geometry specified by the shape parameter. Practically, the observation box Ω^B is a part of the computational domain placed below the airplane. Its upper boundary is a plane close below the aircraft. Of course, in presence of shocks, the value of the SBDE term is infinite in the continuous case, but finite (and mesh dependant) in the discrete case.

As explained in Sec. 2.2, gradients methods in optimal shape design need functional preconditioning, due to a loss of regularity in the gradient. In the case of the Euler equations, the loss of derivative, can be derived in a formal way from Hadamard-type formulas, see [23]. The loss of regularity is 1 as in Sec. 2.2.

6.2 Discretized problem

We do not go into the details of the discrete CFD model, described in [23]. In short, it uses an upwind Euler solver applying to unstructured tetrahedrizations. The shape is changed by moving the nodes on the boundary of the mesh along normals to that boundary. Their displacement is taken into account by a transpiration condition in order to avoid costly remeshings. A gradient of the discrete functional is computed with the help of an adjoint system, built using the Automatic Differentiation tool `TAPENADE`, [13]. Since this gradient is defined as a scalar field defined on the nodes of the boundary, we can apply to it the surfacic multilevel preconditioner introduced in Sec. 5.2.

6.3 Numerical results

The above preconditioner is applied to the shape optimisation starting from an ONERA M6 wing (Fig. 4).
It is the occasion to verify how the best parameter a is well predicted

by the theory. We first study this with a coarse geometry, involving 2203 nodes for the 3D mesh, but yet 780 shape parameters.
Two optimisation algorithms are applied, a gradient (Fig.5) and a conjugate gradient (Fig.6). Convergence of the gradient iteration can be evaluated with the evolution of the gradient norm. The case without preconditioning is the case where $a = 0$. Then convergence is the slowest of the different options tested. Values like $a = 0.5$, $a = 1.5$, $a = 1.$, $a = 2.$ provide good speedups, in particular for the 10 first iterations. This point is important for the shape design loops which, in practice, are too computer comsuming to allow for more optimisation iterations. But the theoretical value $a = 1.$ appears as numerically optimal, for both cases of a pure gradient optimiser as well as a conjugate gradient one.
Let us verify that the efficiency of the preconditioner is good for finer discretisations. We consider a second mesh with 15463 nodes. The number of shape parameters is 3222. See Fig.7 and Fig.8.
The value $a = 1.$ appears again as numerically optimal. The convergence speedup is again good (CPU ratio is 4) and in the case of the conjugate gradient, a good shape is obtained in about 12 iterations.
The case of $a = 2.$ desserves some comments. Indeed, it is equivalent to a Laplace-Beltrami smoothing. All the above results show that this option is not bad, but is clearly not as good as the optimal one.
In [23] the proposed preconditioner is applied to more complex geometries.

7 Conclusion

This paper studies a new preconditioning strategy for optimal shape design. An additive multilevel preconditioner is built from (a) the classical Bramble-Pasciak-Xu principle, and (b) the agglomeration principle. Functional analysis considerations show the central role of the loss of regularity in the gradient iteration. This should be compensated by the regularity gain of the preconditioner. In the proposed method, this regularity gain is easily prescribed by the user at any real value.
We show that this helps to combine the preconditioner with a BFGS acceleration.
In the case of shape design, we exhibit a simplified example for which the loss of derivative can be rigorously evaluated and is equal to 1.
The final application which is depicted is a pre-industrial one already addressed by different methods [7]. The results (i) confirm the *a priori*

analysis of the regularity loss and (ii) demonstrate that the proposed method improves notably the efficiency of the shape design tool.

References

1. E. Arian and S. Ta'asan. Analysis of the Hessian for aerodynamic optimization: Inviscid flow. *Comput. Fluids*, 28(7):853–877, 1999.
2. R.E. Bank and T.Dupont. Analysis of a two-level scheme for solving finite element equations. Technical report, Technical Report CNA-159,Center for Numerical Analysis,University of Texas at Austin, 1980.
3. J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1–22, 1990.
4. M.-O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, and O. Pironneau. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finit element methods (I). least squares formulation and conjugate gradient solutions of the continuous problems. *Comput. Meths. Appl. Mech. Engrg.*, (17/18):619–657, 1979.
5. A. Cohen. Wavelet methods in numerical analysis. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume VII, pages 417–713. Elsevier Science, 2000.
6. F. Courty. *Optimisation Différentiable en Mécanique des Fluides Numérique.* Thesis, University of Paris XI, 2003. to appear.
7. F. Courty and A. Dervieux. A SQP-like one-shot algorithm to optimal control problems. *submitted to Mathematical Programming.*
8. A. Dervieux. Résolution de problèmes à frontière libre. *Thesis, university of Paris VI*, 1981. (in French).
9. A. Dervieux and B. Palmerio. Une formule de Hadamard dans des problèmes d'identification de domaines; exemples. *C. R. Acad. Sc. Paris*, 280, Serie A:1761–1764, 1975. (in French).
10. P.R. Garabedian. Partial differential equations. *Wiley*, New-York, 1964.
11. A. Griewank. The local convergence of Broyden-like methods on Lipschitzian problems in Hilbert spaces. *SIAM J. Numer. Anal.*, 24:684–705, 1987.
12. W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin-Heidelberg-New-York, 1985.
13. INRIA Tropics team. On-line documentation of the Tapenade AD tool. Technical report. http://www.inria.fr/tropics.
14. A. Kunoth. *Multilevel Preconditioning.* PhD thesis, University of Berlin, 1994.
15. M.-H. Lallemand, H. Steve, and A. Dervieux. Unstructured multiggridding by volume agglomeration : current status. *Computer Fluids*, 21(3):397–433, 1992.
16. B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids.* Clarendon Press - Oxford, 2001.
17. F. Murat and J. Simon. Quelques résultats sur le contrôle par un domaine géométrique. *Publications du Laboratoire d'Analyse Numérique, university of Paris VI*, 1974.
18. B. Koobus N. Marco and A. Dervieux. An additive multilevel preconditioning method. *Journal of Scientific Computing*, 12(3):233–251, 1997.
19. J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, New York, 1999.
20. O. Pironneau. *Optimal shape design for elliptic systems.* Springer-Verlag, 1984.
21. J. Reuther and A. Jameson. Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory. AIAA Paper 95-0123, 1995. 33rd Aerospace Sciences Meeting and Exhibit.

22. M. Vazquez, A. Dervieux, and B. Koobus. Aerodynamical and sonic boom optimization of a supersonic aircraft. INRIA Report 4520, 2002.
23. M. Vazquez, A. Dervieux, and B. Koobus. Multilevel optimization of a supersonic aircraft. to appear.
24. J. Xu. *Theory of multilevel methods*. PhD thesis, Cornell university, 1989.
25. J. Xu. An introduction to multilevel methods. In M. (ed.) et al. Ainsworth, editor, *Wavelets, multilevel methods and elliptic PDEs. 7th EPSRC numerical analysis summer school, University of Leicester, Leicester, GB, July 8–19, 1996.*, pages 213–302. Oxford: Clarendon Press. Numerical Mathematics and Scientific Computation., 1997.
26. H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

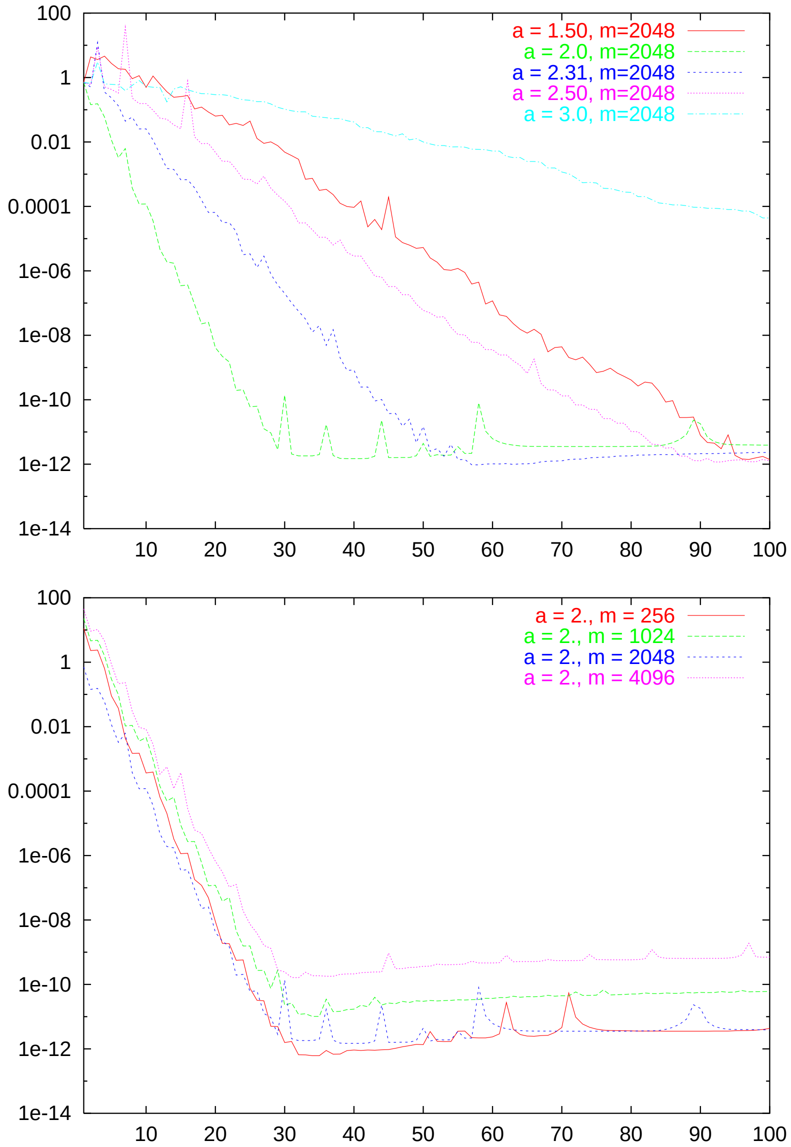

Fig. 1. Resolution of a Poisson problem with a preconditioned Polak-Ribiere CG: The optimality of $a = 2.$ is put in evidence (top) together with mesh-size independant convergency for this parameter (bottom)

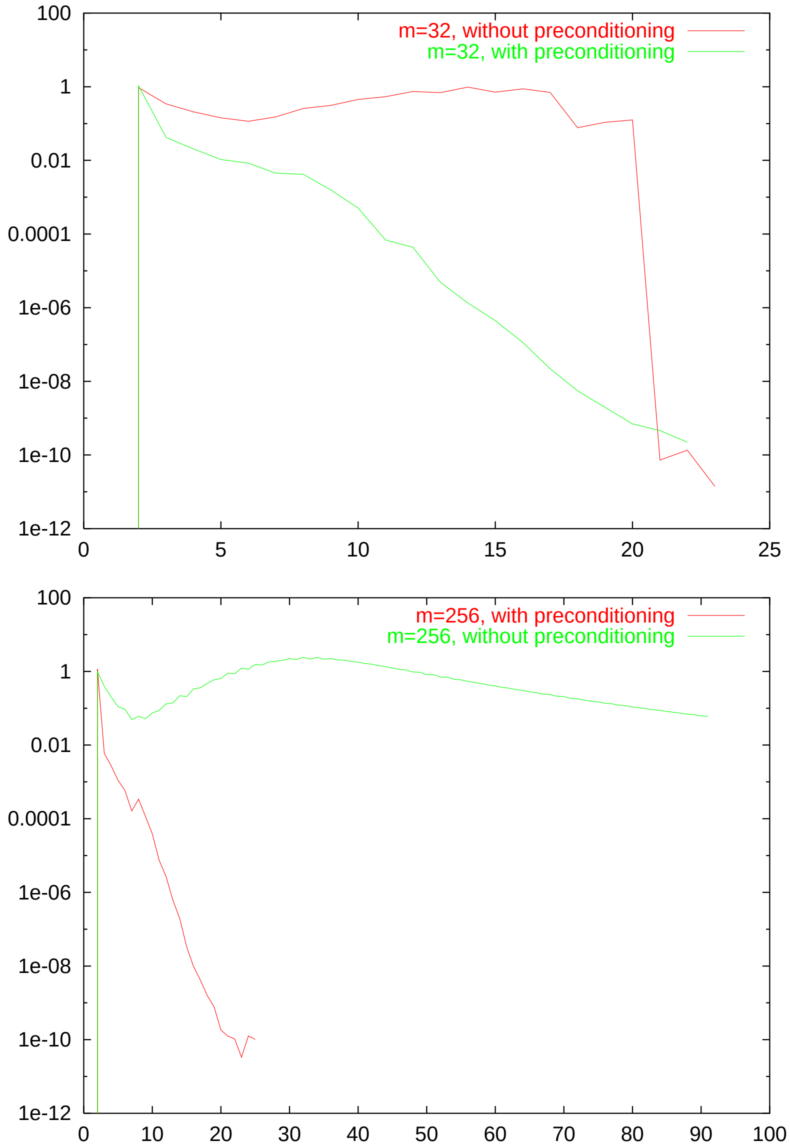

Fig. 2. Resolution of a second order optimization problem with a preconditioned SQP-BFGS, behavior of the different residuals as functions of iteration number, for problem sizes m = 32 and m = 256.

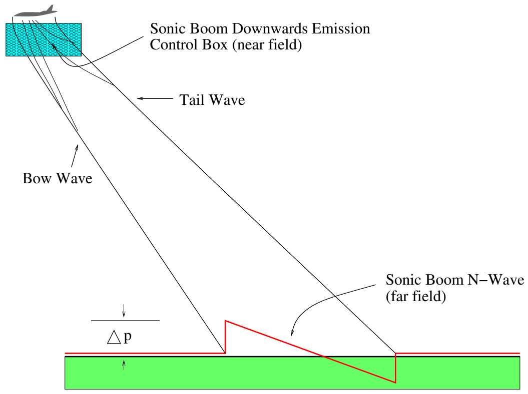

Fig. 3. The sonic boom. Sketch of near and far field shock wave patterns of a supersonic aircraft. SBDE's control box Ω^B.

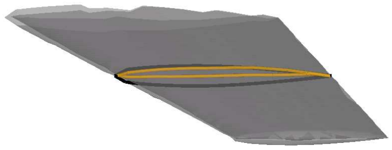

Fig. 4. ONERA M6 shape optimisation: sketch of geometry, of initial and final shape.

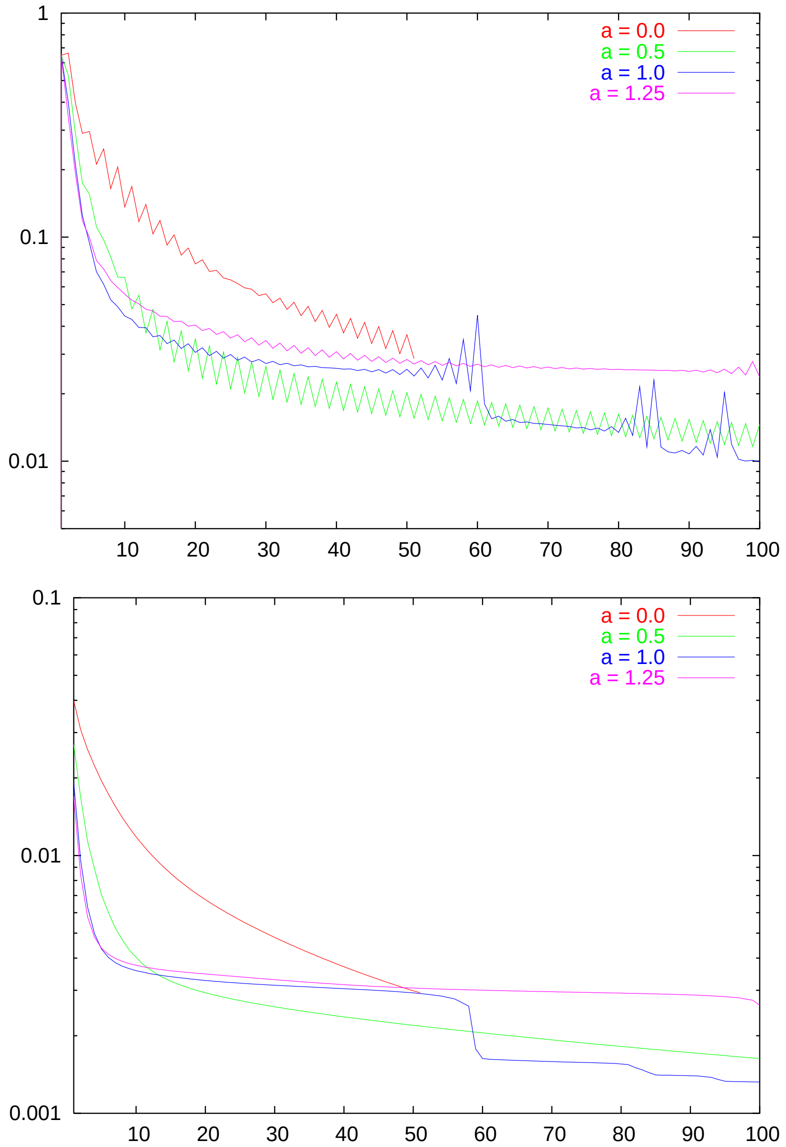

Fig. 5. ONERA M6 shape optimization with a gradient method: Upper Gradient convergence, Lower Cost function, ns = 2203

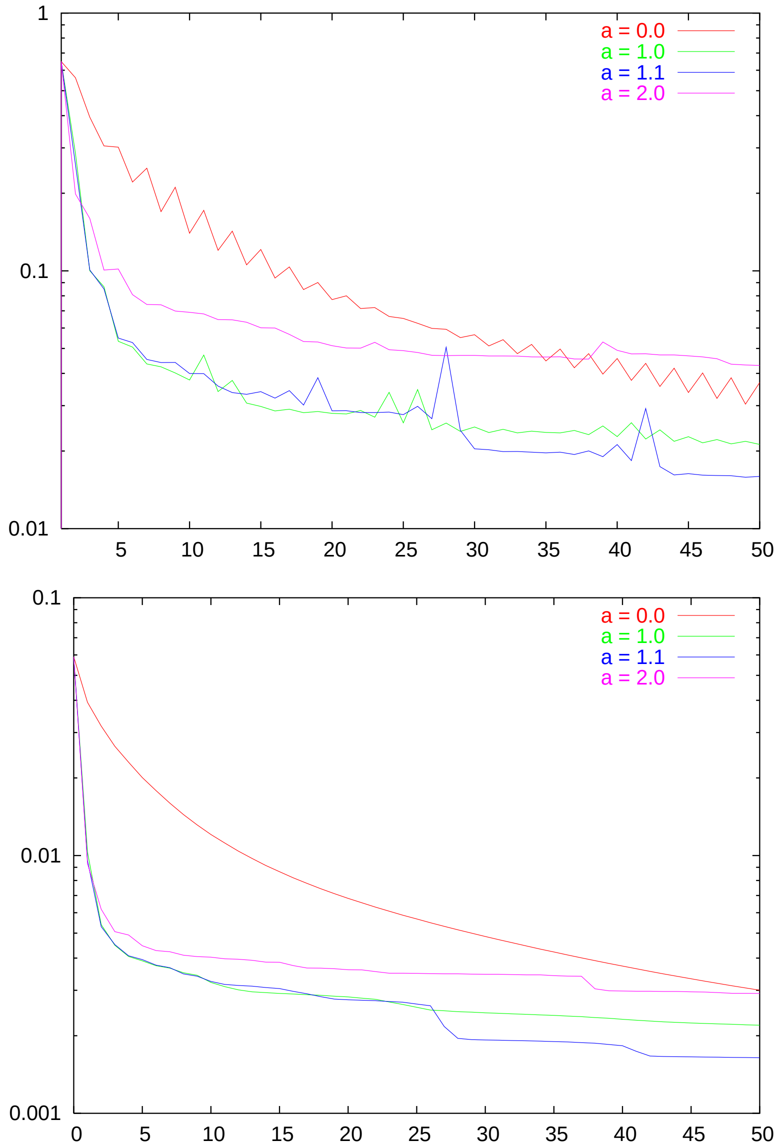

Fig. 6. ONERA M6 shape optimization with a conjugate gradient method: Upper Gradient convergence, Lower Cost function, ns = 2203

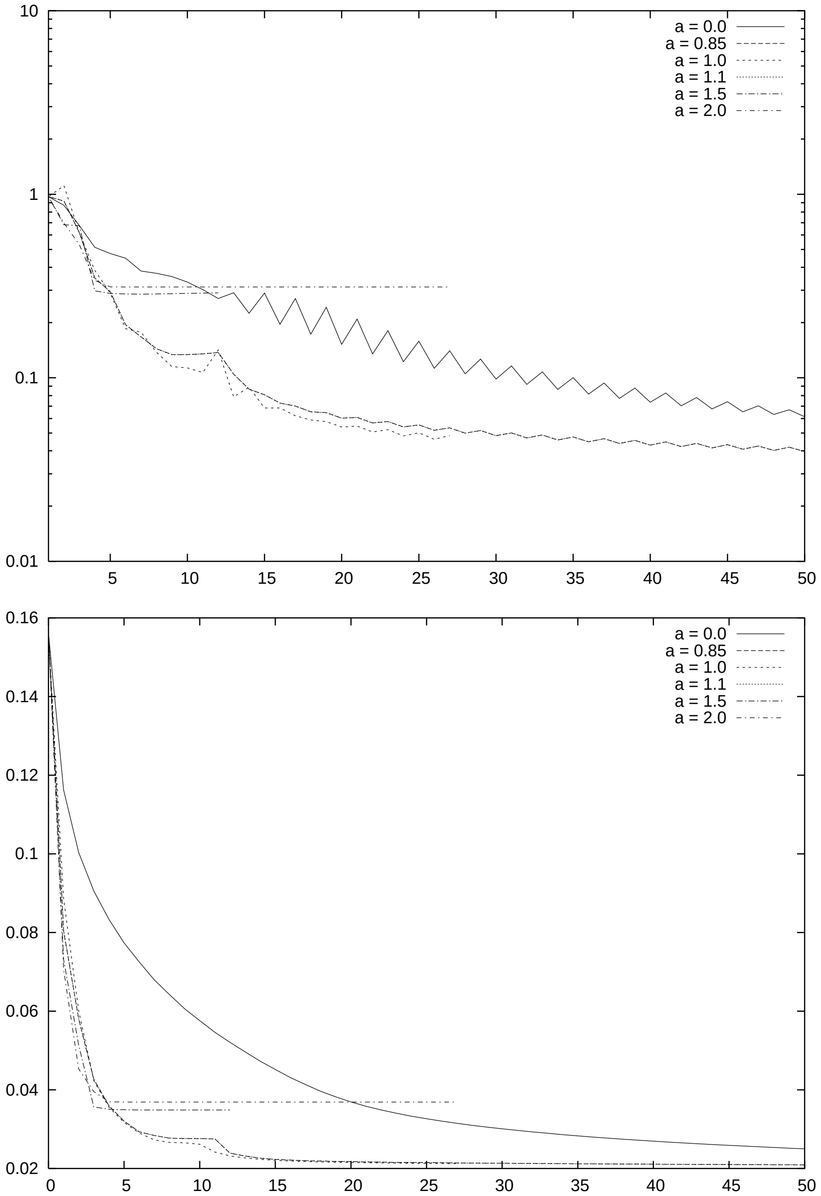

Fig. 7. ONERA M6 shape optimization with a gradient method: Upper Gradient convergence, Lower Cost function, ns = 77315

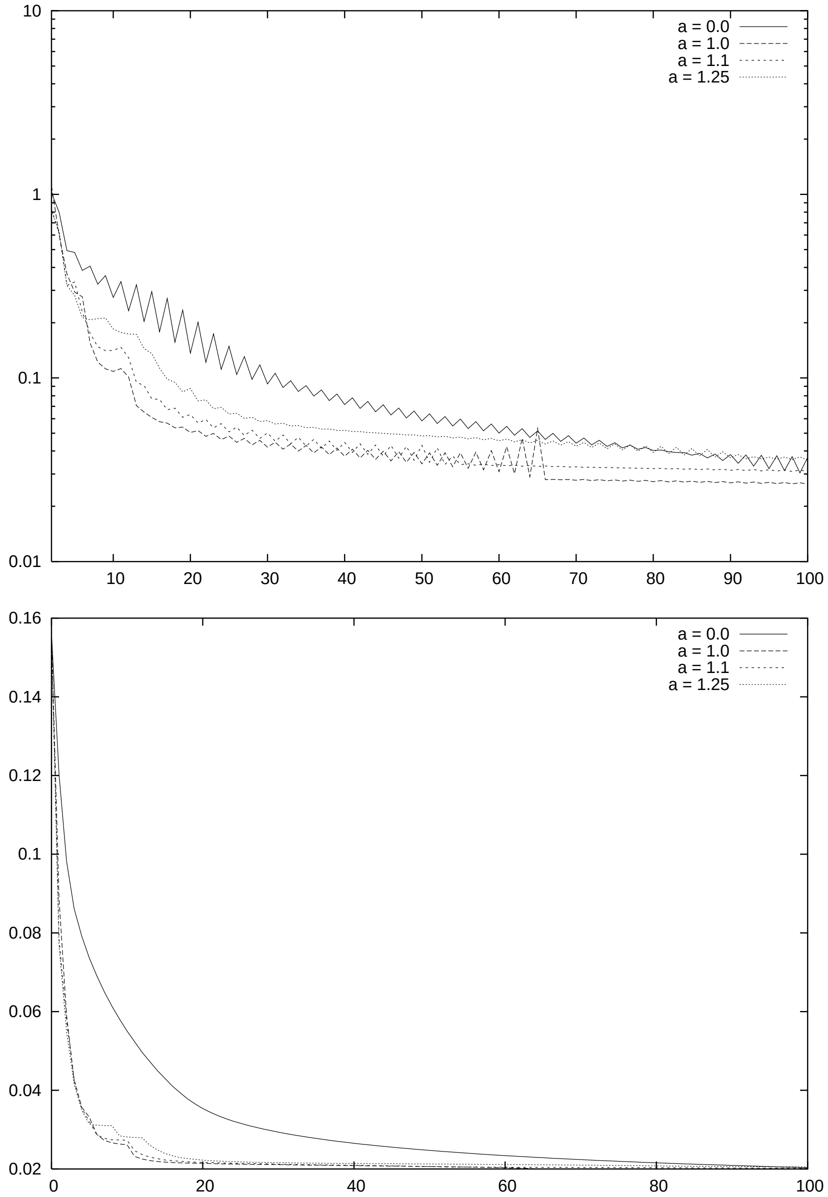

Fig. 8. ONERA M6 shape optimization with a conjugate gradient method: Upper Gradient convergence, Lower Cost function, ns = 77315