Thirteenth International Conference on Domain Decomposition Methods
Editors: N. Debit, M.Garbey, R. Hoppe, J. Périaux, D. Keyes, Y. Kuznetsov

43 A Mesh Refinement Method for Optimization with DDM

Géraldine Lemarchand, Olivier Pironneau[1] and Elijah Polak[2]

Approximate Gradient

We apply here an idea developed in [PP02] whereby mesh refinement can be mixed with approximate gradients within an optimization loop. This is particularly useful for problems where the exact gradient is difficult to compute, which is the case of DDM problems[BW86]

Consider a generic optimization problem and its finite dimensional approximation

$$\min_{z\in Z} J(z) \qquad \min_{z\in Z_h} J_h(z). \tag{1}$$

The following is the method of Steepest descent with a Goldstein/Armijo rule for the step size:

Algorithm 1 *:*

while $\|\text{grad}_z J_h(z^m)\| > \epsilon$ ***do***
{
$z^{m+1} = z^m - \rho \text{grad}_z J_h(z^m)$ *where* ρ *is such that*

$$-\beta\rho\|w\|^2 < J_h(z^m - \rho w) - J_h(z^m) < -\alpha\rho\|w\|^2 \tag{2}$$

with $w = \text{grad}_z J_h(z^m)$ *Set* $m := m + 1$*;*
}

Now consider the same algorithm with parameter refinement

Algorithm 2 *:*

while $h > h_{min}$ ***do***
{ ***while*** $\|\text{grad}_z J_h(z^m)\| > \epsilon h^\gamma$ ***do***
{
$z^{m+1} = z^m - \rho \text{grad}_z J_h(z^m)$ *where* ρ *such that,*

$$-\beta\rho\|w\|^2 < J_h(z^m - \rho w) - J_h(z^m) < -\alpha\rho\|w\|^2 \tag{3}$$

with $w = \text{grad}_z J_h(z^m)$. *Set* $m := m + 1$*;*
}
$h := h/2$*;*
}

Convergence is straightforward to establish as it is either Steepest Descent or $\text{grad} J_h \to 0$ by the fact that $h \to h/2$.

[1]Université Paris VI and IUF (pironneau@ann.jussieu.fr)
[2]EEC, University of California, Berkeley

Approximate Gradients

Another possible gain in speed arises from the observation that we may not need to compute the exact gradient $\mathrm{grad}_z J_h$!

Assume that N is an iteration parameter and that $J_{h,N}$ and $\mathrm{grad}_z J_{h,N}$ denote approximations of J_h and $\mathrm{grad}_z J_h$ in the sense that

$$\lim_{N\to\infty} J_{h,N}(z) = J_h(z) \quad \lim_{N\to\infty} \mathrm{grad}_{z\,N} J_{h,N}(z) = \mathrm{grad}_z J_h(z). \tag{4}$$

Now consider the following algorithm with additional parameter K and $N(h)$ with $N(h) \to \infty$ when $h \to 0$:
The following is Steepest descent with Goldstein/Armijo rule, mesh refinement and approximate gradients:

Algorithm 3 *:*

while $h > h_{min}$
{
while $|\mathrm{grad}_{z\,N} J^m| > \epsilon h^\gamma$
{
try *to find a step size* ρ *with* $w = \mathrm{grad}_{z\,N} J(z^m)$

$$-\beta\rho\|w\|^2 < J(z^m - \rho w) - J(z^m) < -\alpha\rho\|w\|^2 \tag{5}$$

if *success* ***then***
$\{z^{m+1} = z^m - \rho\mathrm{grad}_{z\,N} J^m; m := m+1;\}$
else $N := N + K$;
}
$h := h/2; N := N(h)$;
}

The convergence is established by observing that Goldstein's rule gives a bound on the step size:

$$-\beta\rho\mathrm{grad}_z J\cdot h < J(z+\rho h) - J(z) = \rho\mathrm{grad}_z J\cdot h + \frac{\rho^2}{2}J''hh \tag{6}$$

$$\Rightarrow \quad \rho > 2(\beta-1)\frac{\mathrm{grad}_z J\cdot h}{J''(\xi)hh} \tag{7}$$

so that

$$J^{m+1} - J^m < -2\frac{\alpha(1-\beta)}{\|J''\|}|\mathrm{grad}_z J|^2 \tag{8}$$

Thus at each grid level the number of gradient iterations is bounded by $O(h^{-2\gamma})$. Therefore the algorithm does not jam and as before the norm of the gradient decreases with h.

Applications

Distributed control and DDM

Let $S \subset \Gamma = \partial\Omega$

$$\min_{v \in L^2(S)} J(v) = \int_\Omega [(u-u_d)^2 + |\nabla(u-u_d)|^2] \tag{9}$$

subject to

$$u - \Delta u = 0 \text{ in } \Omega, \frac{\partial u}{\partial n}|_S = \xi v \quad u_{\Gamma - S} = u_d\} \tag{10}$$

Then the optimality conditions are

$$\delta J = \int_S \xi (u - u_d) \delta v \tag{11}$$

Let $\Omega = \Omega_1 \cup \Omega_2$, let $\Gamma = \partial\Omega$ and $\Gamma_{ij} = \partial\Omega_i \cap \Omega_j$. The multiplicative Schwarz algorithm for the Laplace equation starts from a guess u_1^0, u_2^0 and computes the solution of

$$u - \Delta u = f \text{ in } \Omega, \quad u|_\Gamma = u_\Gamma \tag{12}$$

as the limit in n of $u_i^n, i = 1, 2$ defined by

$$\begin{aligned}
&u_1^{n+1} - \Delta u_1^{n+1} = f \text{ in } \Omega_1, \\
&u^{n+1}|_{\Gamma \cap \overline{\Omega}_1 - S} = u_\Gamma \quad u_1^{n+1}|_{\Gamma_{12}} = u_2^n \quad \frac{\partial u_1^{n+1}}{\partial n}|_S = \xi v \\
&u_2^{n+1} - \Delta u_2^{n+1} = f \text{ in } \Omega_2, \\
&u^{n+1}|_{\Gamma \cap \overline{\Omega}_2 - S} = u_\Gamma \quad u^{n+1}|_{\Gamma_{21}} = u_1^n \quad \frac{\partial u_2^{n+1}}{\partial n}|_S = \xi v
\end{aligned}$$

The discretized problem is

$$\begin{aligned}
&\min_{v \in V_h} J_h^N(v) = \|u^N - u_d\|_\Omega^2 : \quad u_j^0 = 0, \quad n = 1..N \quad \forall w \in V_h \\
&u_j^n|_{\partial\Omega_{ij}} = u_j^{n-1}, \quad \int_{\Omega_j} [u_j^n w + \nabla u_j^n \nabla w] = \int_S \xi v w
\end{aligned}$$

where N is the number of Schwarz iterations. The exact discrete optimality conditions are difficult to implement because we may need to store all intermediate functions generated by the Schwarz algorithm (at least for the nonlinear cases) and integrate the system for the adjoint vectors in the reverse order. So here we will try to use the approximate gradient

$$\theta_{h,N} = \|u_{h,N} - u_d\|_S \tag{13}$$

where u_h is computed by N iterations of the Schwarz algorithm.

while $h > hmin$ {
while $\theta_{h,N} > \epsilon(h)$ {
if $(J_{h,N}^{m+1} - J_{h,N}^m < -\alpha\rho^m\theta_{h,N}^2)$
{ do a gradient iteration of step size ρ^m and m:=m+1 }
else N:=N+K
}
h:= h/2 }

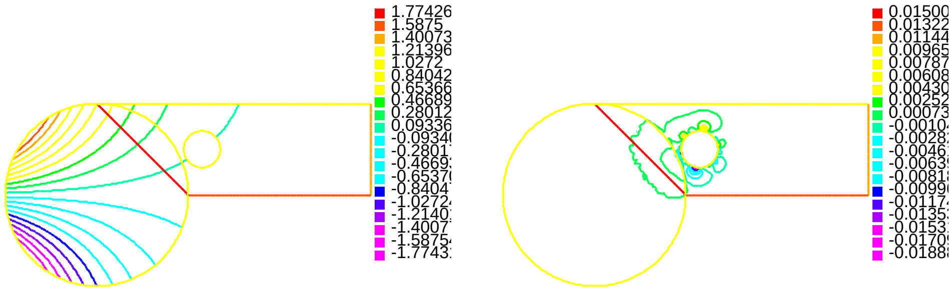

Figure 1: *The computed solution u (left) and the error $u - u_d$ (right).*

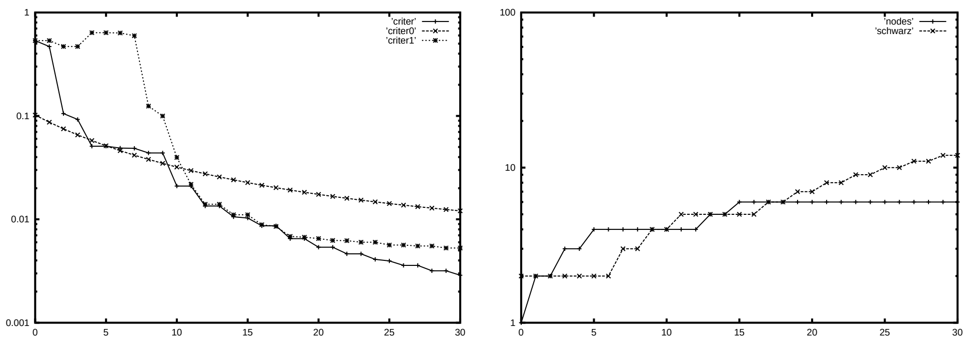

Figure 2: *After 30 iterations the gradient is 10^{-6} times its initial value, while without mesh refinment it has been divided by 100 only (embedded grid effect). On the* **left***, is shown the cost function versus iteration number with and without mesh adaptation for Problem P_1. The smooth curve ($- + -$) corresponds to standard steepest descent on the finest mesh with 500 Gauss-Seidel iterations for the linear systems. The broken curve ($- \times -$) shows cost function decrease with Algorithm 1. Although the two curves are similar, there is an order of magnitude decrease in computing time using Algorithm 1. On the* **right** *is shown the history of the parameters in the algorithm, N and h.*

Numerical results

$u_d = e^{-x\sqrt{2}} sin(y)$. $\xi = \sin(30 * (x - 1.15)) + \sin(30 * (y - 0.5))$. The number of Schwarz iterations is initialized at 1. Results are shown in Figures 1 and 2.

Control in the coefficients

An absorbant coating of thickness α on an airfoil S is optimized to cancel the reflected accoustic wave in a sector σ. The Leontowitch conditions models the thin coating:

$$\min_{\alpha} \int_{\Sigma} |u|^2 \quad \text{subject to}$$
$$\omega^2 u + \Delta u = 0 \qquad \frac{\partial u}{\partial n} - i\omega u = 0 \text{ on } \Gamma_\infty, \quad \frac{\partial u}{\partial n} + \alpha\omega u = 0 \text{ on } S$$

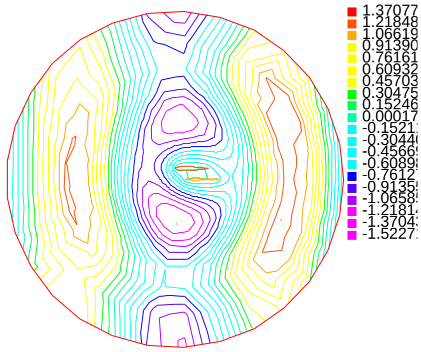

Figure 3: *Real part of the solution of Helmholtz equation*

The problem is discretized by the finite element method of degree 1 [Cia78] on triangles. The linear systems are solved with a Gauss factorization. The same gradient method with inexact gradients is applied (i.e. the gradient of the continuous problem discretized) with domain decomposition where one domain surrounds one of the airfoil. Figures 3 and 4 show the solution and Figure 5 shows the history of the convergence compared with a straight steepest descent method and a steepest descent with mesh refinement only and no DDM. The FEM software [BHOP99] has been used.

Optimal Shape Design

A transonic flow is computed by solving the Euler system of partial differential equation with NSC2KE[MP01] and the profile is optimized so as to minimize the pressure drag. The state equation is non-linear and the acceleration by approximate gradient is on the number of Newton iterations in the flow solver. There is no DDM here. The results are shown in Figures 6 and 7.

References

[BHOP99]Dominique Bernardi, Frederic Hecht, Kohji Otsuka, and Olivier Pironneau. freefem+, a finite element software to handle several meshes. *http://www.freefem.org*, 1999.

[BW86]Petter E. Bjørstad and Olof B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1093–1120, 1986.

[Cia78]Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.

[MP01]Bijan Mohammadi and Olivier Pironneau. *Applied OPtimal Shape Design*. Oxford University Press, Oxford, 2001.

[PP02]Olivier Pironneau and Elijah Polak. Consistent approximations and approximate functions and gradients in optimal control. *SIAM J. Numer. Anal.*, page to appear, 2002.

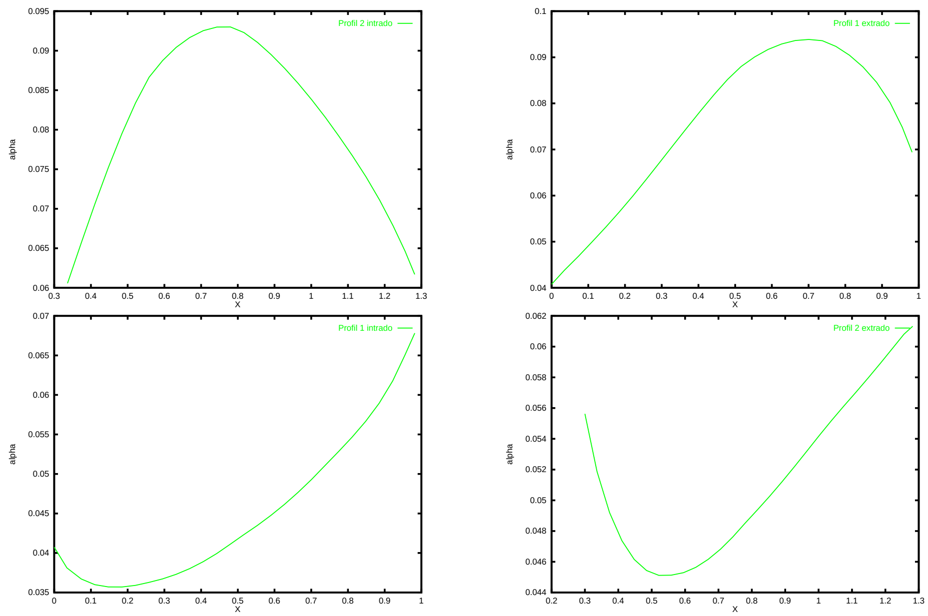

Figure 4: *α versus distance to the leading edge on the two sides of each airfoil.*

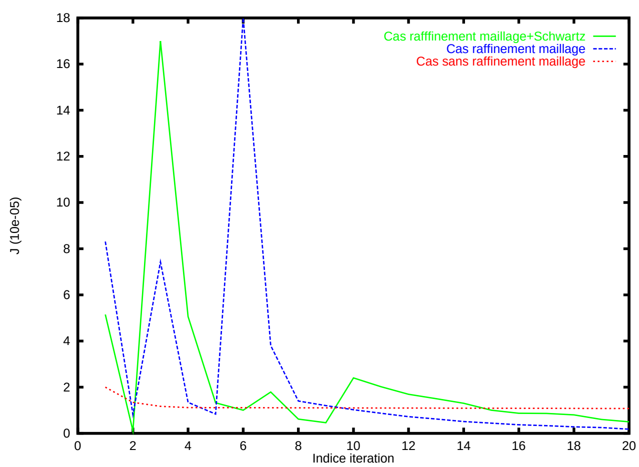

Figure 5: *History of the convergence of the cost function for the coating problem. The method with mesh refinement and adapted Schwarz iteration number (green curve) is compared with a straight steepest descent method (red curve) and a steepest descent with mesh refinement only and no DDM (blue curve).*

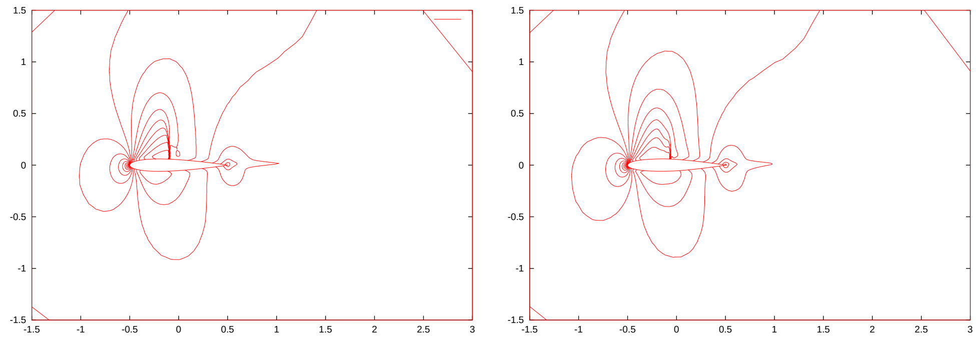

Figure 6: *Mach lines for the flow around the airfoil before shape optimization (left) and after. Notice that the shock tends to disappear, an expected result since the drag is a pressure drag.*

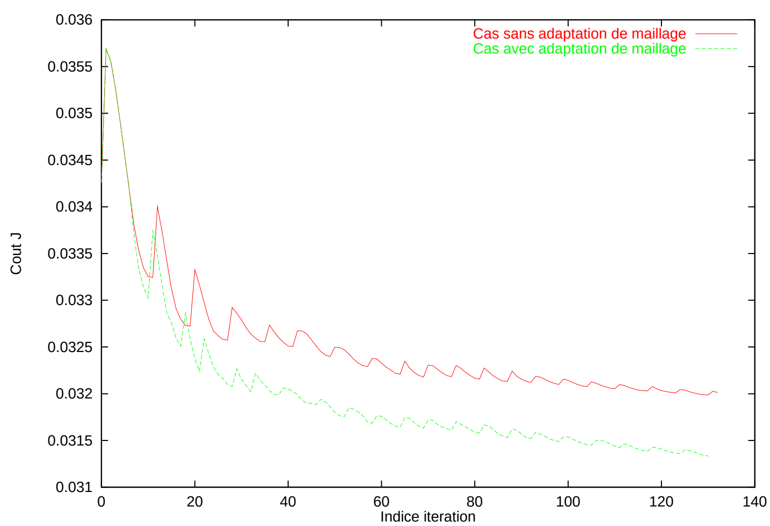

Figure 7: *History of the decrease of the cost function with and without mesh refinement and approximate gradient based on non converged flow solvers. The curve in red (top curve) is without mesh refinement but with control over the iteration number for the flow solver and the green curve is the same with mesh refinement.*