Mesh adaptation for time dependent simulation, optimization and control

Bijan Mohammadi [*] — **Frédéric Hecht** [**]

[*] *Université Montpellier II and INRIA - M3N*
Math. CC51
34090 Montpellier
Bijan.Mohammadi@math.univ-montp2.fr

[**] *Université Pierre et Marie Curie and INRIA - Gamma*
4 place Jussieu
75005 Paris, France
Frederic.Hecht@inria.fr

ABSTRACT. *The paper describes our adaptative design and control platform for aerodynamic configurations. The aim is to analyze the impact of mesh discretization and connectivity changes on the convergence of minimization algorithms applied to shape design and state control. Hence, we are interested in analyzing the impact of these changes for unsteady configuration simulations. We propose a transient fixed point approach for the set mesh-solution-gradient limiting these effects.*

RÉSUMÉ. *Nous décrivons un algorithme d'optimisation et de contrôle adaptatif sur des maillages non structurés à connectivité variable. On considère des applications aérodynamiques, mais l'approche est générique. Le but est d'étudier l'impact du changement de discrétisation et de connectivité du maillage sur la convergence des méthodes d'optimisation et de contrôle. Ainsi, on s'intéresse aussi à l'impact de ces changements sur les configurations instationnaires. On propose un algorithme de remaillage utilisant la solution d'un problème de point fixe instationnaire pour l'ensemble maillage-solution-gradient, tentant de limiter ces perturbations.*

KEYWORDS: *Optimization, Unstructured anisotropic mesh adaptation, unsteadiness.*

MOTS-CLÉS : *Optimisation, Génération de maillages non structurés anisotropes, instationnarités.*

Revue européenne des éléments finis. Volume 10 – n° 5/2001, pages 575 à 593

1. Introduction

Mesh adaptation by metric control is a natural way for mesh generation and optimization [GEO 98, HEC 97, BOR 96, CAS 00, BOR 96a, HAB 00]. In the past, we have used these techniques for steady inviscid and viscous laminar and turbulent configurations. This paper is to extend the application range to unsteadiness as well as to optimization problems and in particular, to see the impact of this additional non-differentiability in the simulation loop on the optimization. The needs for adaptivity in unsteady configurations is even more crucial than for steady cases as often the unsteadiness occupies the whole computational domain implying an uniformly fine mesh everywhere.

For steady configurations, the adaptive algorithm consists in searching for a fixed point for the couple formed by the mesh and the corresponding state. To get the fixed point, we usually apply an iterative approach similar to multigrids with no restriction operator (only keeping coarse to fine transfer). We show here how, for unsteady configurations, to solve for the mesh and corresponding state couple a transient fixed point problem. This is performed for both fixed and moving domains, where in case of modified mesh points distribution and connectivities, a classical ALE [DON 82, NKO 94] approach cannot be used anymore.

To reduce the impact of mesh nodes and connectivity changes during adaptation, we add to the Delaunay generation procedure a new constraint, keeping the new mesh close to the background mesh, if the metric predicts negligible changes for the mesh compared to the background one. From the metric definition point of view, we introduce, in addition to the metric intersection performed for systems [HEC 97], a metric intersection procedure in time for the mesh to be optimal for all intermediate states between two adaptations.

As we use a dynamic formulation for optimization and control problems, these problems can be seen as time dependent simulations and therefore what developed for mesh adaptation for unsteady cases can be applied. In addition, this is done together with an approximate gradient approach for optimization problems. We used this incomplete evaluation of gradients for shape design and state control for 2 and 3D, steady and unsteady configurations for fixed connectivities. The key idea in our incomplete gradient evaluation is that the sensitivity with respect to the state can be neglected for cost functions defined over the shape. This being often the case in applications, as cost functions are often based on boundary integrals [MOH 97, MOH 97b, MOH 99, MED 99, MOH 99b]. One feature of incomplete sensitivity is that, due to their definition over the shape, they are not affected by the non-differentiability introduced by the connectivity changes.

2. Dynamic shape optimization

Consider the following optimization or control problem:

$$\left.\begin{array}{l} \min\limits_{x(t)} J(x(t), q(x), U(q)), \quad E(x(t), q(x), U(q)) = 0, \\ \\ g_1(x(t)) \leq 0, g_2(q) \leq 0, \quad g_3(q, U(q)) \leq 0, \end{array}\right\} \qquad [1]$$

where $x(t) \in R^n$ describes a time dependent parameterization. This can be for optimization problems a geometrical CAD-based model or a CAD-free model [MOH 97] and for control problems, the amount of the injection/suction velocity or a flap deflection [MOH 99, MOH 00]. For steady configurations, the time is fictitious and solving this minimization problems is equivalent to capture fixed point for the dynamic system below describing the evolution of $x(t)$. g_1 defines the constraints on the parameterization, g_2 those on geometrical quantities and g_3 state constraints. $E(x, q, U)$ designs the state equation. In this work, we consider as state equations, either the Euler or the Navier-Stokes equations and the $k - \varepsilon$ model [MOH 94]. Details for this problem formulation and parameterization can be found in [MOPI 00]. We only recall the dynamic minimization formulation and the incomplete sensitivity concept and its validity domain. We need mesh adaptation in both the optimization and simulation levels.

2.1. *Closure equation for* $x(t)$

In our approach, minimization algorithms are seen as closure equations for the parameterization. In other words, we introduce a new time dependent problem for $x(t)$. This can also be seen as an equation for the structure when using the elastic CAD-Free parameterization [MOH 00]. We can show that most linear or quadratic gradient based minimization algorithms can be written under the following form :

$$\dot{x} + \epsilon \ \ddot{x} = -F(\mathcal{P}, M^{-1}, (\nabla_{xx} J)^{-1}, \nabla_x J), \qquad [2]$$

where F is a function of the exact or incomplete gradient and of the inverse of the Hessian of the cost function. It also takes into account the projection $\mathcal{P}$ over the admissible space and the smoothing operator we use when using the CAD-Free parameterization [MOH 97, MOPI 00, MOH 00, MOH 99]. To advance in time [2], we use a central difference scheme for the second order operator and a forward difference scheme for the first order one. Denotes by δx^p the shape deformation at step p, the discretized heavy ball method [ATT 99, ATT 96] reads :

$$(\frac{\epsilon}{\lambda^2} + \frac{1}{\lambda})\delta x^{p+1} = \frac{\epsilon}{\lambda^2}\delta x^p - F(\mathcal{P}^p, (M^p)^{-1}, (\nabla_{xx} J^p)^{-1}, \nabla_{x^p} J^p). \qquad [3]$$

Usually, $\mathcal{P}$ does not depend on p except when using mesh adaptation [HEC 97, BOR 96a, CAS 00, MOH 97, MOPI 00].

2.2. *Incomplete sensitivities*

Consider the general simulation loop, involved in [1], leading from shape parameterization to the cost function :

$$J(x) : x \rightarrow q(x) \rightarrow U(q(x)) \rightarrow J(x, q(x), U(q(x))).$$

The gradient of J involves different contributions:

$$\frac{dJ}{dx} = \frac{\partial J}{\partial x} + \frac{\partial J}{\partial q}\frac{\partial q}{\partial x} + \frac{\partial J}{\partial U}\frac{\partial U}{\partial q}\frac{\partial q}{\partial x}. \qquad [4]$$

In most applications, the cost function is, or can be reformulated, in such a way to have the following characteristics:

– The cost function J and the parameterization x are defined on the shape (or a same part of it),

– J is of the form

$$J(x) = \int_{\Gamma_S} f(x, q) g(u) d\gamma, \qquad [5]$$

where Γ_S is the shape or a part of the shape. J means that it involves a product of geometrical and state based functions,

We showed that for such cost functions, the sensitivity with respect to the state can be neglected. But this only in regions where the curvature of the shape is not too large [MOH 97, MOH 99, MOPI 00]. This means for instance away from leading and trailing edge points.

The incomplete sensitivity approach means that we can drop the last term in [4]:

$$\frac{dJ}{dx} \sim \frac{\partial J}{\partial x} + \frac{\partial J}{\partial q}\frac{\partial q}{\partial x}. \qquad [6]$$

We can illustrate this idea on the following simple example. Consider as cost function $J = a\, u_x(a)$ and as state equation the following advection-diffusion equation:

$$u_x - Pe^{-1}\ u_{xx} = 0, \ \text{ on }]a, 1[, \ u(a) = 0, \ u(1) = 1.$$

We are in the domain of application of the incomplete sensitivities, where the cost function is product of state and geometrical quantities and is defined at the boundary.

$$J_a(a) = u_x\ (a) \left(1 + a \frac{Pe^{-1}\ \exp(Pe^{-1}\ a)}{\exp(Pe^{-1}\ a) - \exp(Pe^{-1})}\right). \qquad [7]$$

The second term in the parenthesis is the state linearization contribution which is negligible for large Peclet number. In all case, the sign of the sensitivity is always correct.

This approximation has other interesting features. In particular, when using mesh adaptation, only the shape discretization impacts the sensitivity evaluation. Therefore, the impact of the non-differentiability of unstructured mesh generations is reduced. Indeed, incomplete sensitivities only require for the evaluation of geometrical sensitivities around the shape. However, these sensitivities require an accurate evaluation of the state in space for optimization problems, but also in time when targeting control problems by control laws based on instantaneous incomplete gradients. Finally, incomplete sensitivities are used together with CAD-Free parameterization [MOH 97, MOPI 00], with interfaces with CAD tools in both ends of the optimization loop (i.e. CAD to initial surface mesh and final surface mesh to CAD). This point enables the optimization platform to be generic and not to depend on a given CAD parameterization, which is often application-dependent. Another interesting feature in this parameterization is the possibility for an easy change of the scalar product used to define the Hilbert space in which optimization is performed (space of deformations). This is done introducing appropriate smoothers which also have preconditioning effects [MOPI 00].

3. Metric definition

Details on the ingredients used in the metric definition for inviscid and viscous laminar and turbulent flows involving shocks and boundary layers can be found in [BOR 96, CAS 00, BOR 96a, HEC 97, FRE 99, GEO 91, MOPI 00]. We only describe the metric intersection procedure which is useful for systems and also for time dependent phenomena, where the mesh is adapted to be optimal for the capture of the various solution states in time between two adaptations.

The key idea in mesh adaptation by metric control is to modify the scalar product used in the mesh generator for distance and volume evaluations. Therefore, the aim, using an automatic mesh generation method is to construct equilateral elements according to a new adequate metric. The scalar product is based on the evaluation of the Hessian $\mathcal{H}$ of the variables of the problem. Indeed, for a P^1 Lagrange discretization of a variable u, the interpolation error is bounded by:

$$\mathcal{E} = |u - \Pi_h u|_0 \leq c \sup_{T \in \mathcal{T}_h} \sup_{x,y,z \in T} |\mathcal{H}(x)|(y-z).(y-z) \qquad [8]$$

where $\Pi_h u$ is the P^1 interpolate of u, $|\mathcal{H}(x)|$ is the Hessian of u at point x after being made positive definite, and where $.$ is the dot product. Now, if we generate, using a Delaunay procedure as a mesh generation method a mesh with edges close to unit length in the metric $\mathcal{M} = \frac{|\mathcal{H}|}{(c\mathcal{E})}$, the interpolation error $\mathcal{E}$ is equi-distributed over the edges a_i of the mesh. More precisely, we have

$$\frac{1}{c\mathcal{E}} a_i^T \mathcal{M} a_i \leq 1. \qquad [9]$$

For systems, the previous approach leads to a metric for each variable. For two metrics $\mathcal{M}_1$ and $\mathcal{M}_2$, we define an metric intersection $\mathcal{M} = \mathcal{M}_1 \cap \mathcal{M}_2$, such that the

unit ball of $\mathcal{M}$ is include in the intersection of the two unit balls of metric $\mathcal{M}_2$ and $\mathcal{M}_1$ with the procedure defined in [FRE 99, chap 10.3.3] for example.

For time dependent problems, this metric intersection is applied in time, for each variable, between the metrics defined by this variable at two successive iterations:

$$\hat{\mathcal{M}}^{n+1} = \hat{\mathcal{M}}^{n} \cap \mathcal{M}^{n+1}, \qquad [10]$$

where n denotes the time integration iteration. Once, the intersection in time performed, the procedure above is applied for the intersection between the metrics of the different variables.

4. Mesh generation and adaptation

We use the mesh generation tools developed at INRIA-Gamma project [FRE 99, GEO 91]. One novelty introduced however in this work in the Delaunay mesh generation part is to introduce an extra criteria keeping the new mesh nodes and connectivities unchanged as much as possible compared to the previous mesh where the prescribed mesh by the metric is similar to the previous mesh. This is suitable of course for time dependent simulations and reduces the perturbation introduced by remeshing and solution interpolation from the background over the new mesh.

The algorithm of adapted mesh generation is a Delaunay-like method ([FRE 99, Sec. 7.3.1]) where the insert point procedure is fully describe in [FRE 99, Sec. 7.6.1]. This algorithm can be resume as:

1. Create a bounding box mesh.

2. Discretize the boundary with respect of metric, we call $\mathcal{S}_B$ the set of boundary points.

3. Insert one by one the points of $\mathcal{S}_B$.

4. Enforce the boundary mesh.

5. Remove outside triangles.

6. DO: far field point creation $\mathcal{S} = \emptyset$.

– Split all the too long edges such that the lengths of sub-edges 'in the metric' are close to one. Add the new splitting points in $\mathcal{S}$.

– Insert all the points of $\mathcal{S}$,if they are not too close 'in the metric' to an existing point.

7. Until $\mathcal{S}$ is empty

8. Do some small regularization by moving vertices and swapping edges to improve the mesh quality.

To introduce the idea above, we need to use the points of the background mesh. The algorithm becomes:

1. Create a bounding box mesh.

2. Discretize the boundary with respect to the metric, Let be $\mathcal{P}$ a set equal $\mathcal{S}_B$ the set of boundary point.

3. Sequentially add the internal points of the background mesh to $\mathcal{P}$, if a point is not too close (distance in the metric less than $\frac{1}{\sqrt{2}}$) to a point of $\mathcal{P}$.

4. Insert one by one the points of $\mathcal{P}$.

5. In-force the boundary mesh.

6. Remove outside triangles.

7. DO: far field point creation $\mathcal{S} = \emptyset$.

– Split all the too long edges such that the lengths of sub-edges 'in the metric' are close to one. Add the new splitting points in $\mathcal{S}$.

– Insert all the points of $\mathcal{S}$,if they are not too close 'in the metric' to an existing point.

8. if $\mathcal{S}$ is not empty, repeat 7.

9. Do some small regularization.

, see above.

5. Mesh adaptation for unsteady simulations

The following algorithm shows the classical fixed point algorithm applied to the couple of mesh and corresponding state for an unsteady simulation.

At step i of adaptation, we denote the mesh, the state and the metric by $\mathcal{H}_i$, $\mathcal{S}_i$, $\mathcal{M}_i$. $\Delta t = \frac{\tau}{N_{adap}}$, where τ is the smallest time scale of the phenomena we want to capture. Hence, performing two adaptive simulations for N_{adap} and $2N_{adap}$ permits to evaluate the mesh independence of unsteadiness. The adaptive loop reads:

initialisation : $i = 0, \quad t = 0$	$\mathcal{H}_0, \mathcal{S}_0, \Delta t$ given,
While $(t < t_{max})$**, Do**	
compute the metric :	$(\mathcal{H}_i, \mathcal{S}_i) \rightarrow \mathcal{M}_i,$
generate the new mesh using this metric :	$(\mathcal{H}_i, \mathcal{M}_i) \rightarrow \mathcal{H}_{i+1},$
interpolate the previous solution over the new mesh :	$(\mathcal{H}_i, \mathcal{S}_i, \mathcal{H}_{i+1}) \rightarrow \overline{\mathcal{S}}_{i+1},$
advance in time by Δt the solution over this mesh :	$(\mathcal{H}_{i+1}, \overline{\mathcal{S}}_{i+1}) \rightarrow \mathcal{S}_{i+1},$
$i++$ **Done**	

This algorithm introduces perturbations due to non-consistent interpolation of the solution between meshes for both steady and unsteady computations. This leads in particular to smoothing effects when applied to unsteady configurations [CAS 00]. We propose the following corrections to the algorithm:

– to replace the metric by the intersection in time of the metrics as defined above,

– to look for a transient fixed point for the mesh-state couple making $NFIX$ internal fixed point iterations.

$\mathcal{H}_0, \mathcal{S}_0, \Delta t, NFIX$ given, $i = 0, \quad t = 0$

While $(t < t_{max})$**, Do** $j = 0$

While $(j < NFIX \quad \text{or} \quad ||(\frac{\partial \mathcal{S}}{\partial t})_i^{j+1})| - |(\frac{\partial \mathcal{S}}{\partial t})_i^j)| > TOL)$, **Do**

If$(j = 0)$**Then**

initialize the intersected metric over time with an isotropic coarse metric: $\hat{\mathcal{M}}_i^j = (\mathcal{I}(\frac{1}{h_{max}^2}))$,

Else If $(j > 0)$ **Then**

generate the new mesh using the time intersected metric: $(\mathcal{H}_i^j, \hat{\mathcal{M}}_i^j) \rightarrow \mathcal{H}_i^{j+1}$,

interpolate the previous solution and its time derivative over the new mesh : $(\mathcal{H}_i^{j+1}, \mathcal{S}_i^0, (\frac{\partial \mathcal{S}}{\partial t})_i^0, \mathcal{H}_i^j) \rightarrow (\overline{\mathcal{S}}_i^{j+1}, \overline{(\frac{\partial \mathcal{S}}{\partial t})}_i^{j+1})$,

End If

advance in time by Δt the solution over this mesh and compute the intersected metric over time: $(\mathcal{H}_i^{j+1}, \overline{\mathcal{S}}_i^{j+1}, \overline{(\frac{\partial \mathcal{S}}{\partial t})}_i^{j+1}) \rightarrow (\mathcal{S}_i^{j+1}, (\frac{\partial \mathcal{S}}{\partial t})_i^{j+1}, \hat{\mathcal{M}}_i^j)$,

perform metric intersection.

$j++$ **Done**.

$(\mathcal{H}_{i+1}^0, \mathcal{S}_{i+1}^0, (\frac{\partial \mathcal{S}}{\partial t})_i^0) \leftarrow (\mathcal{H}_i^j, \mathcal{S}_i^j, (\frac{\partial \mathcal{S}}{\partial t})_i^j)$,

$i++$ **Done**

6. Mesh Adaptation in optimization and control

We show how to couple shape optimization or control tools and mesh adaptation by metric control. When mesh adaptation is used, not only the mesh in the computational domain, but also the number and positions of the discretization points over the shape will change during optimization. This means that the shape and unstructured mesh deformation tools should take into account these transformations. In addition, in the

CAD-Free framework, a change in shape discretization implies a change in the design space which we should avoid so as to keep solving the initial optimization problem.

6.1. *Adaptive optimization algorithm*

The following algorithm shows how to couple shape optimization based on incomplete sensitivities and mesh adaptation. we define an exhaustive control space X suitable for a good description of any shape deformations. This can be defined by a fine surface mesh giving a good representation of the shape.

At step i of the adaptive optimization algorithm, we denote the mesh, the solution, the metric and the cost $J(x_i, U(x_i))$ by $\mathcal{H}_i$, $\mathcal{S}_i$, $\mathcal{M}_i$ and $J(x_i)$. The adaptative optimization loop reads:

$\mathcal{H}_0, \mathcal{S}_0, X$ given,

Adaptation loop : Do	$i = 0, .., iadapt$
1. identify the wall nodes :	$\mathcal{H}_i \rightarrow x_i,$
2. compute the incomplete gradient :	$(x_i, \partial\mathcal{H}_i, \mathcal{S}_i\vert_{\partial\mathcal{H}_i}) \rightarrow \frac{dJ(x_i)}{dx_i},$
3. interpolate the gradient :	$\frac{dJ}{dX} = \Pi_i(\frac{dJ(x_i)}{dx_i}),$
4. define the deformation using $\mathcal{P}$:	$\tilde{X},$
5. interpolate back the deformation:	$\tilde{x}_i = \Pi_i^{-1}(\tilde{X}),$
6. deforme the mesh :	$(\tilde{x}_i, \mathcal{H}_i) \rightarrow \tilde{\mathcal{H}}_i,$
7. update the solution over the deformed mesh :	$(\tilde{\mathcal{H}}_i, \mathcal{S}_i) \rightarrow \tilde{\mathcal{S}}_i,$
8. compute the metric :	$(\tilde{\mathcal{H}}_i, \tilde{\mathcal{S}}_i) \rightarrow \mathcal{M}_i,$
9. generate the new mesh using this metric :	$(\tilde{\mathcal{H}}_i, \mathcal{M}_i) \rightarrow \mathcal{H}_{i+1},$
10. interpolate the previous solution over the new mesh :	$(\tilde{\mathcal{H}}_i, \tilde{\mathcal{S}}_i, \mathcal{H}_{i+1}) \rightarrow \overline{\mathcal{S}}_{i+1},$
11. advance in pseudo-time the solution over this mesh:	$(\mathcal{H}_{i+1}, \overline{\mathcal{S}}_{i+1}) \rightarrow \mathcal{S}_{i+1},$
End Do.	

Here the projection $\mathcal{P}$ is defined once for all, but the interpolation operator Π_i is changing during adaptation and is available as the trace of the interpolation operator defined between the new and the background mesh, where the local metric is defined.

6.2. *Adaptive control algorithm*

In our approach, control is seen as unsteady shape optimization and the time in the dynamic minimization approach becomes real. To provide mass injection velocity, we use the Hadamard equivalent boundary condition which enables to account for small deformations on the non deformed shape and therefore permits to use a dynamic shape optimization algorithm to define control laws [MOPI 00, MOH 99b, MOH 99] for injection devices. Once the locations of control devices (mass injection or piezo-electric devices) on the exhaustive parameterization defined, we use the previous adaptative optimization algorithm after changing step 6 by the two following steps defining the equivalent injection velocity:

6a. Define the normals to the deformed shape and
the shape speed in the fixed reference: (n_m^p, V_m^p),
6b.define the new injection velocity: $u^p.n_f = u^p(n_f - n_m^p) + V_m^p.n_m^p$,

where subscript m denotes the deformed and f the non deformed geometry. The projection operator for control configurations accounts for the maximum injection/suction velocity or equivalent deformation allowed by the control device. Indeed, for a control law to be efficiently realizable by a piezo-electric or an injection/suction device, the amount of the required deformation or injection/suction has to be small. For instance, it should not exceed one percent of the flow speed.

Finally, to improve the coupling between mesh adaptation and control, we combine the transient adaptative fixed point algorithm for unsteady simulation presented above with the previous adaptative control algorithm.

7. Numerical Experiences

We consider the impact of mesh adaptation on different types of applications:

– Time dependent compressible flow simulations,

– Constrained drag reduction for steady transonic aerodynamic flows,

– Flow control for unsteady transonic aerodynamic flows.

7.1. *Unsteady simulation*

We present the application of the algorithms presented above to three configurations:

1. Sod shock tube, treated as a full 2D case and where the adaptive solutions are compared to a reference solution computed on a fixed fine mesh. The aim is to show through this simple example how the mesh can be late over the solution in an unsteady simulation leading to a phase error for the solution. This is improved by the transient fixed point approach.

2. Transonic inviscid flow over a Naca airfoil at Mach 0.8 with a limit steady state solution. We show the impact of the transient fixed point approach on the evolution of aerodynamic lift and drag coefficients.

3. Oscillating airfoil in a transonic flow. The aim is to show the impact of a suitable time dependent adaptation on the evolution of aerodynamic coefficients. This case is presented bellow with and without our control algorithm applied.

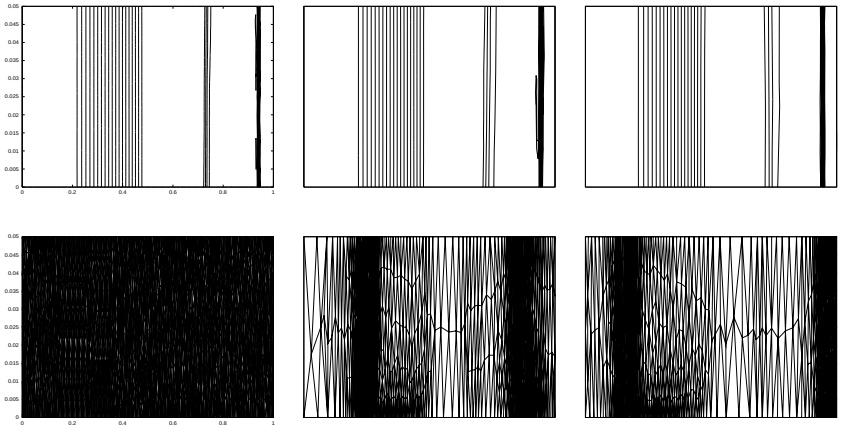

Figure 1. *Sod shock tube: impact of the transient fixed point algorithm with four internal iterations for adaptation compared to the classical algorithm. Mesh and corresponding iso-Mach contours at* $t = 0.25$*. Two successive adaptations are distant of* $\Delta t = 0.025$*. Left: reference mesh and solution, middle: classical fixed point, right: transient fixed point with 4 inner fixed point iterations. The minimum mesh size is identical for all three meshes.*

7.2. *Adaptation and optimization*

This is a drag reduction problem for a transonic inviscid and viscous turbulent flow with constraints on the lift and volume treated by penalty. The cost function is given by:

$$J(x) = C_d + \alpha|C_l - C_l^0| + \beta|Vol - Vol_0|,$$

where α and β are penalty parameters, C_d is the drag coefficient, C_l and C_l^0 are the actual and initial lift coefficients and Vol and Vol_0 the actual and initial volumes. The

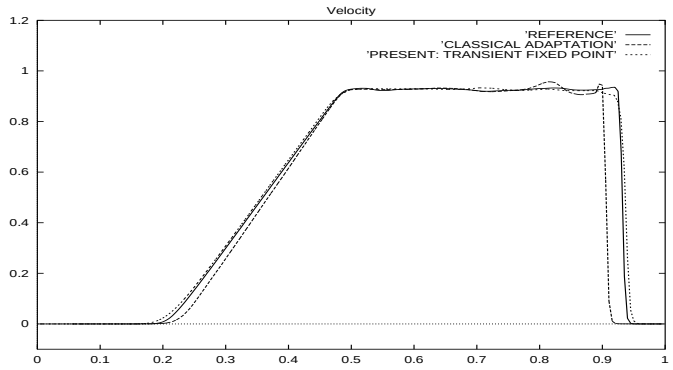

Figure 2. *Sod shock tube: Velocity distribution on the lower wall. The classical approach leads to a phase error for all variables. In addition, there are less spurious oscillations when mesh generation is consistent with the time integration scheme.*

initial airfoil is either the NACA0012 or the RAE2822. The state equations are either the Euler system in conservation variables or the Navier-Stokes and $k - \varepsilon$ equations with wall functions [MOH 94]. The aim is to show the impact of mesh adaptation on optimization seen as time dependent simulation. For the first case, mesh adaptation introduces perturbations in the control space (CAD-Free parameterization changes at each adaptation) and the convergence is not monotonic, while for the second case, despite a more complex physic for the problem and finer meshes near the wall to capture the boundary layer, the convergence history is almost not perturbed compared to an optimization for a fixed fine mesh.

Notice that, for the first case, from one mesh to the next there is a jump in the histories of the cost function and gradient norm due to the fact that this optimization has been done with the control changing at each iteration of adaptation. This is to show the importance for the adaptation not to pollute the control space structure.

7.3. *Adaptation and control*

The previous example was the first step toward control, as we consider control problems as generalized optimization problems for time dependent cost functions. This is illustrated by the following problem of control of the unsteadiness created by the oscillating airfoil presented above. We would like also to see the impact of the transient fixed point algorithm for the ensemble given by: simulation, sensitivity analysis, closed loop control law definition and adaptive mesh generation. The lift and drag histories shows that the efficiency of the control increases with the accuracy of the time dependent adaptive simulations. Indeed, the control is based on incomplete instantaneous sensitivities with their accuracy depending on the accuracy

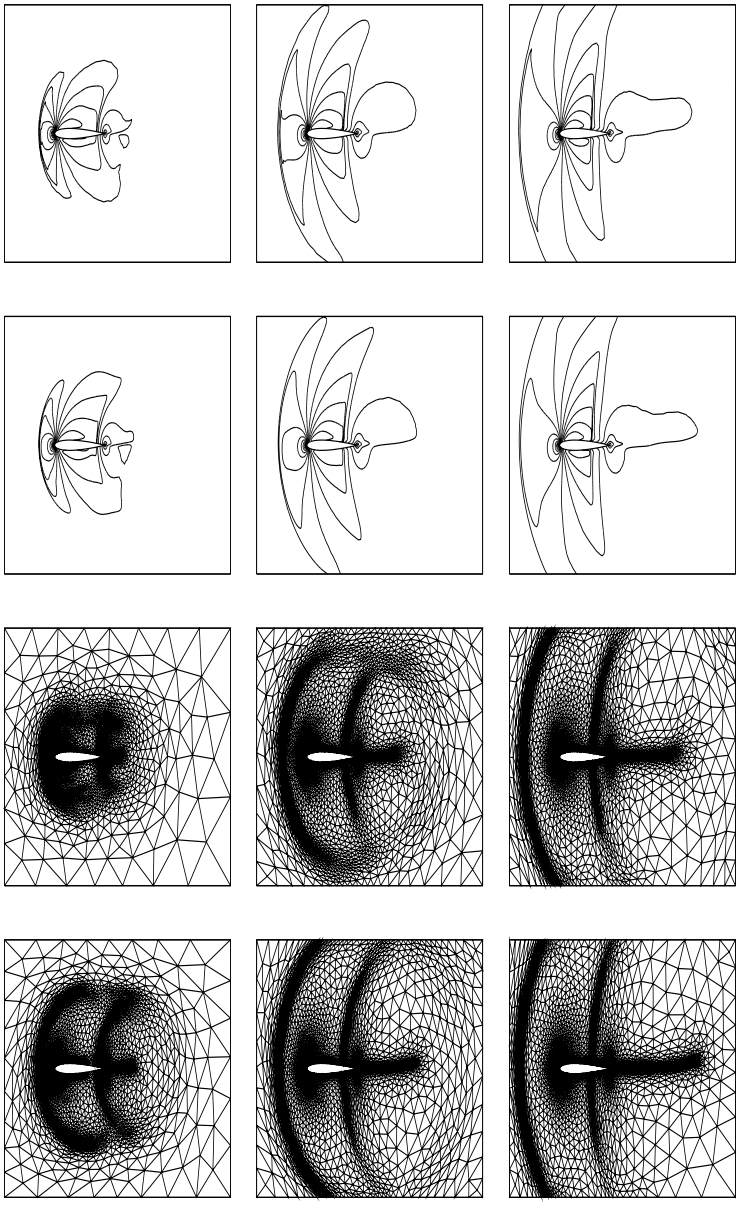

Figure 3. *Impact of the transient fixed point algorithm with four internal iterations (lower pictures) for adaptation compared to the classical algorithm. Snapshots of the mesh and the corresponding iso-Mach contours at* $tU - \infty/Chord = 0.3, 0.6, 0.9$. *Two successive adaptations are distant of* $\Delta tU_\infty/Chord = 0.05$.

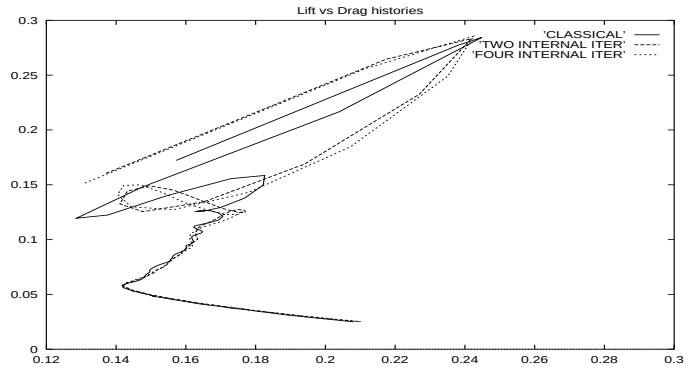

Figure 4. *Lift vs. drag coefficients histories with the classical and using the transient fixed point algorithm. The final solution being steady, all simulations recover the same values. This justifies the use of the classical approach for steady configurations. But the history obtained with the classical approach is quite different.*

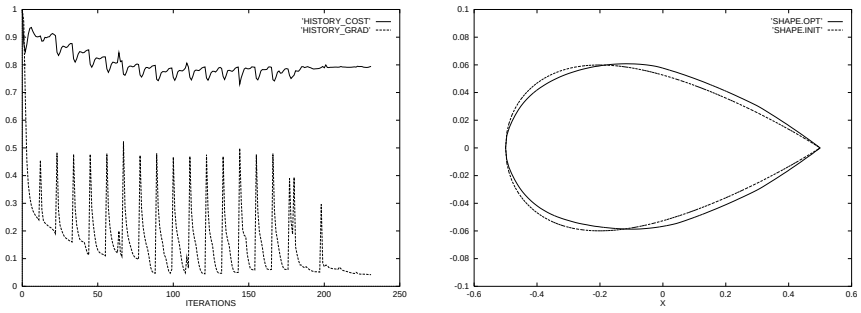

Figure 5. *Cost function and norm of gradients for the airfoil optimization (left) and the initial and final forms (right). In this case, the control space changes at each adaptation.*

of the unsteady state. The airfoil oscillates, with a reduced frequency of 10, in a range of 0.1 degree, around its equilibrium position at 2 degrees of incidence. The inflow Mach number is 0.75. This is a high frequency case which might be interesting for noise reduction due to induced aerodynamic fluctuations. The aim is to reduce the instantaneous aerodynamic drag coefficient. The instantaneous cost function to be minimized is $J(t, x(t)) = |C_d(t) - C_d^{target}|$, where target denotes a target value (here $C_d^{target} = 0.013$).

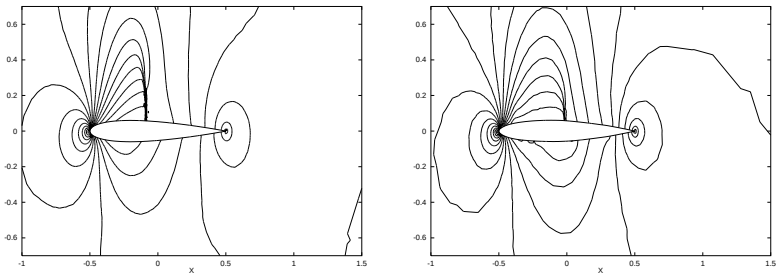

Figure 6. *Pressure contours before and after optimization. Notice the disappearance of the shock.*

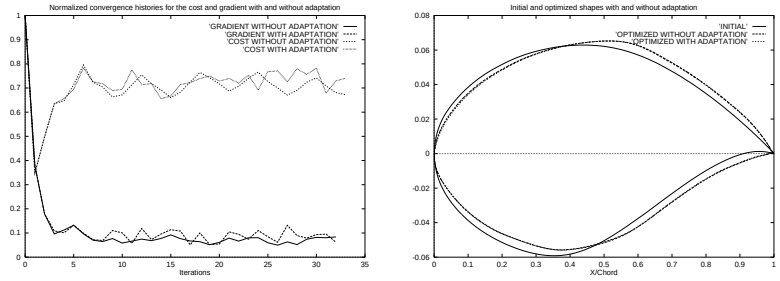

Figure 7. *Shape optimization at transonic viscous turbulent regime: cost function and norm of gradients for the airfoil optimization for the optimization alone and when combined with adaptation (left) and the initial and final shapes for each approaches (right). With adaptation, we can be confident that the mesh is always adequate for intermediate states. The jumps in the convergence histories are much less important than for the previous case.*

8. Concluding Remarks

The application of our control and optimization platform combining instantaneous and incomplete sensitivities with dynamic minimization algorithms has been presented. The platform has been coupled with mesh adaptation techniques designed to capture unsteadiness. This has been made possible, over both fixed or moving domains, introducing a transient fixed point approach. In this technique we look for solutions of a fixed point problem for the ensemble {mesh, solution, sensitivity} linked together through the state and sensitivity equations, the minimization dynamic system and a local metric based Delaunay mesh generator.

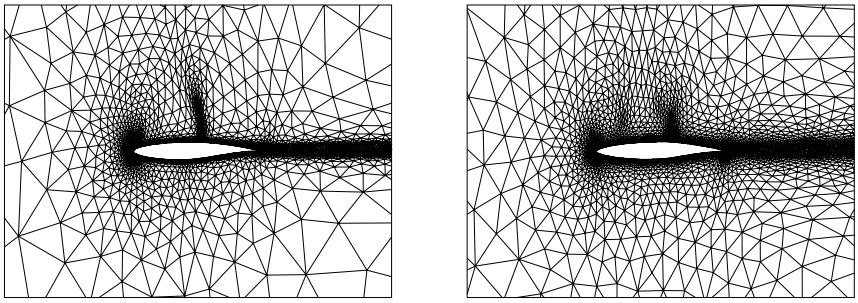

Figure 8. *Shape optimization at transonic viscous turbulent regime: Initial and final meshes when using adaptation during optimization.*

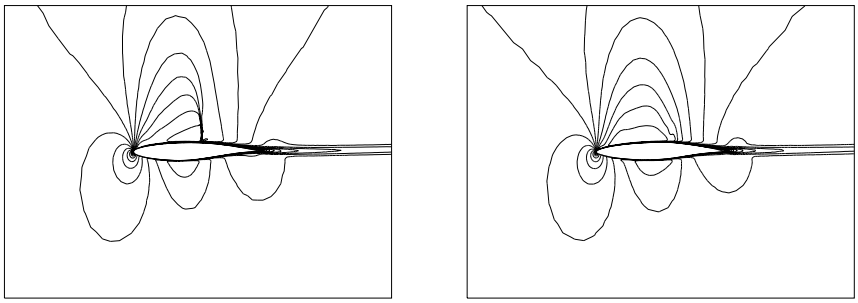

Figure 9. *Shape optimization at transonic viscous turbulent regime: Initial and final iso-Mach contours for the previous meshes.*

Our main goal was to make the cost of control and simulation problems similar, and this without a loss of accuracy due to coarser meshes, usually used for sensitivity analysis for instance. This is particularly true for unsteady configurations. Indeed, when unsteadiness occupies a large part of the computational domain, its capture would imply a fine mesh everywhere. Something which is out of reach. On the other hand, classical adaptation methods are not suitable for unsteadiness capture as they introduce uncertainties due to adaptation and interpolation between meshes.

This global approach enables to propose adaptive real time control tools based on sensitivities as an alternative to black-box control tools using filtering approach.

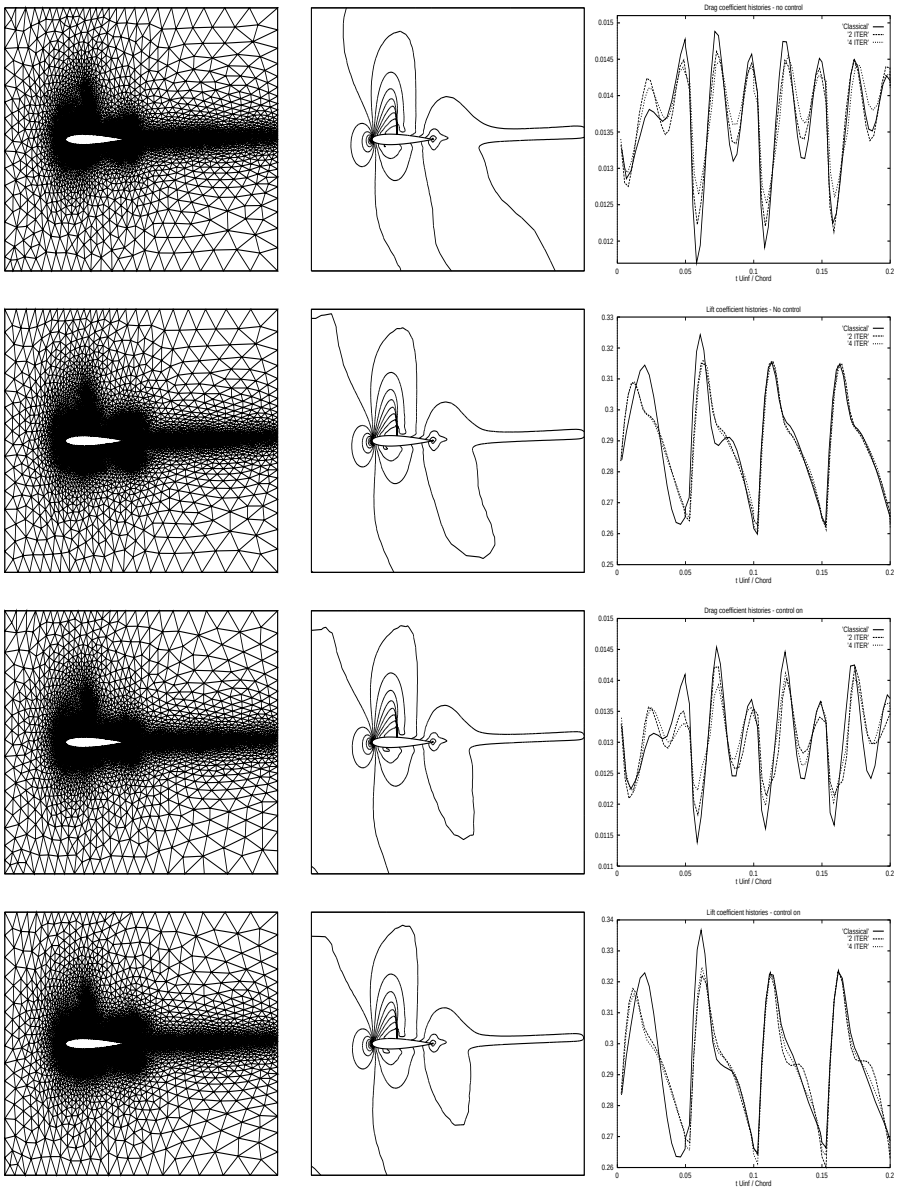

Figure 10. *Oscillating airfoil (upper pictures:control off - lower pictures: on): impact of the transient fixed point algorithm with four internal iterations for adaptation compared to the classical algorithm with the control on and off. Snapshots of the mesh and corresponding solution at $tU_\infty/Chord = 0.2$. Two successive adaptations are distant of $\Delta tU_\infty/Chord = 0.04$. The lift and drag histories show that only two internal iterations quite improves the prediction. In addition, they show that the efficiency of the control increases with the accuracy of the time dependent adaptive simulations. The transient fixed point algorithm together with time metric intersection procedure makes the mesh to be finer over the region the state varies in the time interval between two adaptations.*

Acknowledgments

The author would like to thank Professors J. L. Lions and O. Pironneau for the interest they brought to this work. Many thanks to C. Faure for making `Odyssee` available to us. Particular thanks to Dasssault Aviation and in particular Q. Dinh, M. Mallet, G. Rogé, Ph. Rostand and B. Stoufflet for their support during this work realization.

9. References

[ATT 99] ATTOUCH H., REDONT P., GOUDOU X. «The Heavy Ball with Friction Method » *Advances in Contemprary Mathematics*, 1999.

[ATT 96] ATTOUCH H., COMINETTI R. «A Dynamical Approach to Convex Minimization Coupling Approximation with the Steepest Descent Method» *J. Differential Equations*, 128 (2) pp. 519-540, 1996.

[BOR 96] BOROUCHAKI H., CASTRO-DIAZ M.J., GEORGE P.L., HECHT F. AND MOHAMMADI B., «Anisotropic adaptive mesh generation in two dimensions for CFD» *5th Inter. Conf. on Numerical Grid Generation in Computational Field Simulations*, Mississipi State Univ., 1996.

[BOR 96a] BOROUCHAKI H.,GEORGE P.L. AND MOHAMMADI B. «Delaunay Mesh Generation Governed by Metric Specifications. Part II : Applications» *Finite Element in Analysis and Design, special issue on mesh adaption*, 1996.

[CAS 00] CASTRO-DIAZ M.J., HECHT F. AND MOHAMMADI B. «Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations» *IJNMF*, Vol. 25, 475-491, 2000.

[DON 82] DONEA J. «An ALE Finite Element Method for Transient Fluid-Structure Interactions» *Comp. Meth. App. Mech. Eng.*, num. 33, pp. 689-723, 1982.

[FRE 99] FREY P.J. ET GEORGE P.L., *Maillages*, Hermès, Paris, 1999.

[GEO 91] GEORGE P.L., *Automatic mesh generation. Applications to finite element method*, Wiley, 1991.

[GEO 98] George P.L. and Borouchaki H., *Delaunay triangulation and meshing*, Hermès, Paris, 1998.

[HAB 00] HABASHI W. ET AL. «A Step toward mesh-independent and User-Independent CFD» *Barriers and Challenges in CFD*, pp. 99-117, Kluwer Ac. pub.

[HEC 97] HECHT F. AND MOHAMMADI B. «Mesh Adaptation by Metric Control for Multi-scale Phenomena and Turbulence» *AIAA*, paper 97-0859, 1997.

[NKO 94] NKONGA B. , GUILLARD H. «Godunov Type Method on Non-Structured Meshes for Three Dimensional Moving Boundary-Problems» *Comp. Methods in Applied Mech. Eng.* num. 113, pp:183-204, 1994.

[MED 99] MEDIC G. , MOHAMMADI B. , PETRUZZELLI N., STANCIU M. , HECHT F. «3D Optimal Shape Design for Complex Flows : Application to turbomachinery» *AIAA* paper 99-0833, 1999.

[MOH 94] MOHAMMADI B., *CFD with NSC2KE : an User Guide*, Technical report INRIA No.164, 1994.

[MOH 97] MOHAMMADI B. «Practical Applications to Fluid Flows of Automatic Differentiation for Design Problems» *VKI lecture series*, 1997-05.

[MOH 97b] MOHAMMADI B. «A New Optimal Shape Design Procedure for Inviscid and Viscous Turbulent Flows» *Int. J. for Numerical Meth. in Fluid*, vol. 25, 183-203, 1997.

[MOH 99] MOHAMMADI B. «Flow Control and Shape Optimization in Aeroelastic Configurations» *AIAA* paper 99-0182, 1999.

[MOH 99b] MOHAMMADI B. «Dynamical Approaches and Incomplete Gradients for Shape Optimization and Flow Control» *AIAA* paper 99-3374, 1999.

[MOH 00] MOHAMMADI B. «Sensitivity Analysis for Design and Control in an Elastic CAD-Free Framework for Multi-Model Configurations » *Journal Europèene des Elements Finis, special issue on fluid/structure interaction*, 2000.

[MOPI 00] MOHAMMADI B. AND PIRONNEAU O., *Applied Shape Design for Fluids*, Oxford Univ. Press, 2000.