New Progress in Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations

Manolo Castro-Diaz, Frédéric Hecht, Bijan Mohammadi

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

New Progress in Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations

M.J. Castro Díaz F. Hecht and B. Mohammadi

N° 2671

Octobre 1995

PROGRAMME 6

ISSN 0249-6399

INRIA
ROCQUENCOURT

New Progress in Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations

M.J. Castro Díaz* F. Hecht and B. Mohammadi

Programme 6 — Calcul scientifique, modélisation et logiciel numérique
Projet MENUSIN

Rapport de recherche n° 2671 — Octobre 1995 — 24 pages

Abstract: Two new ideas for anisotropic adaptation of unstructured triangular grids are presented with particular emphasis to fluid flows computations. The first concept enables a suitable extension of our mesh adaptation to the case of systems of PDE like Navier-Stokes equations and the second makes possible a correct boundary layer computation which was until now one of the major weakness of anisotropic adaptations in CFD.

Key-words: Mesh, Anisotropic, Adaptation, Delaunay, Metric, CFD.

(Résumé : tsvp)

*Dpto. de Análisis Matemático, Universidad de Málaga, Campus Universitario de Teatinos s/n, 29080 Málaga.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Nouveaux progrès en adaptation anisotropique de maillages pour la simulation d'écoulements visqueux et non visqueux.

Résumé : Deux nouvelles idées pour l'adaptation anisotropique de maillages de triangles non structurés sont présentées, dans l'optique des calculs pour la mécanique des fluides. Le premier concept permet une extension satisfaisante de notre adaptation de maillages au cas des systèmes PDE, comme les équations de Navier-Stokes, et le second rend possible le calcul correct de couches limites, qui était jusqu'à présent le talon d'Achille de l'adaptation anisotropique en CFD.

Mots-clé : Maillages, Anisotrope, Adaptation, Delaunay, Métrique, CFD.

Contents

List of Figures ii

1 Introduction **1**

2 Navier-Stokes solver **1**

3 Metric computations **2**
3.1 Extension to systems 3
3.2 Boundary layer improvement 4

4 Mesh generation **5**

5 Numerical examples **6**
5.1 Supersonic scramjet inlet 6
5.2 Supersonic viscous flow over a NACA0012 6
5.3 Unsteady flow around a cylinder 7
5.4 Subsonic flow around a 4-element airfoil 7

6 Concluding remarks **7**

Bibliography **8**

List of Figures

1 Approximated optimal ellipse of two-ellipse intersection. 4
2 Evolution of the Mesh for the two first consecutive adaptations for a supersonic scramjet inlet. 9
3 Evolution of the Mesh for the last three consecutive adaptations for a supersonic scramjet inlet. 10
4 Evolution of iso-Mach contours for an inviscide Mach 3 scramjet inlet. 11
5 Iso-Mach contours obtained on the last three adapted meshes for an inviscide Mach 3 scramjet inlet. 12
6 Convergence history. 13
7 Partial vue of the pressure coefficient on the initial and final meshes. 13
8 Evolution of the Mesh for five consecutive adaptations for a supersonic viscous flow over an airfoil. 14
9 Iso-Mach contours obtained on the five first adapted meshes for a supersonic viscous flow over an airfoil. 15
10 Convergence history. 16
11 hwall with and without our Dirichlet b.c. 17
12 C_p at step 0 (initial), 5 and 6. C_f at step 0 (initial), 5 and 6. 17
13 Time dependent evolution of the mesh over one period. 18
14 Mesh evolution (continued). 19
15 Snapshots of iso-Mach contours for one period. 20
16 Iso-Mach evolution (continued). 21
17 Evolution of the Mesh for a subsonic viscous flow over a 4-element airfoil. 22
18 Iso-Mach contours for a subsonic viscous flow over a 4-element airfoil. 23
19 Convergence history. 24
20 CP along the 4-element airfoil. 24

1 Introduction

Three major advantages of unstructured grids over structured ones are the ease for complex geometries to be considered and grid adaptation and general mesh anisotropy concept to be incorporated (see [6], [7], [12]). Of course, the introduction of mesh adaptivity and anisotropy reduces the number of grid elements if the simulated physical phenomena are strongly directional (see [10] and [12]) as in the case of shocks and limit layers for fluid flows. However, two major difficulties remain. Firstly, as mesh adaptation is usually based on a change of the local metric based on the interpolation error of some quantity, the extension to systems is not clear; especially when phenomena of different nature interact (for instance in shock-boundary layer interactions). Secondly, boundary layers are usually not correctly resolved and wall coefficients (especially the friction coefficient) obtained over adapted meshes are usually not exploitable as the distance of the first layer points to the wall is not uniform. In this paper, we present two ideas to remove the above difficulties.

The first idea is based on the intersection of the different metrics obtained for the different variables of a PDE which are in our case the conservation variables. This removes also the difficulty we had until now to choose the 'right' variable for building the metric from its interpolation error. Indeed, we used to choose for instance the pressure or the density for Euler computations and for instance the local Mach number or the entropy for viscous computations. Of course, this is unsatisfactory because one variable can not encapsulate all the physics of the whole system.

The second idea is motivated by two requirements coming in fact from structured meshes which are known to be more suitable near the walls:

- we want the meshes to be as orthogonal as possible near the body,
- we want each nodal layer to be at a uniform distance of the wall.

Now the idea consists of a modification of our local metric near the wall.

Therefore, to enforce the orthogonality of the mesh in this area we change the eigen-vectors of our metric to make tem parallel and orthogonal to the wall. The second requirement is satisfied by giving to the user the possibility of prescribing the normal size of the elements along the wall and to propagate this through the domain by some relaxation procedure to obtain quite an orthogonal mesh in the near-wall region with each nodal layer at a uniform distance from the wall.

Both these ideas are extensible to three dimensional case. In fact, to generate $2D$ meshes, we have used our $3D$ surface grid generator after setting z to zero. The generation procedure is fast, generating large grids in very few minutes on workstations (an average being 30000 triangles per minute on a 10 MFlops workstation).

In this paper we describe our adaptation loop which can be formally defined as follow:

Start from an initial (in principle coarse) mesh.

Adaptation loop.

a) Solve the PDE (in this case the fluid dynamics system),

b) Build a metric from the multi-variable solution obtained in **a)**,

c) Build an equilateral mesh in the metric of **b)** using a Delaunay type mesh generator.

End of the loop.

We will discuss in particular the point **b** where we introduce the ingredients presented above. In the following sections, we also give a brief description of the fluid solver and the mesh generator.

2 Navier-Stokes solver

We briefly describe the NSC2KE fluid solver we used for these computations. More details can be found in [9]. A Finite-Volume-Galerkin formulation of Navier-Stokes equations in conservation form

has been considered. A four stage Runge-Kutta scheme is used for time integration. The Hybrid Upwind Scheme [5] has been used for the convective part. Second order accuracy is obtained through MUSCL type reconstruction together with Van Albada limiters [13]. A Stegger-Warming [11] flux splitting has been used at the inflow and outflow boundaries while non penetration or slip boundary conditions are applied to solid walls depending on the nature of the flow. This solver is in free access (anonymous ftp piranha.inria.fr under pub/).

3 Metric computations

During automatic mesh generation it is necessary to have as much information as possible on the nature and the local behaviour of the solution. These indications constitute *the control space* which governs the grid generation. As we are interested in informations concerning the elements shape and size, we need to convert somehow what we know about the solution to something having the dimension of a length and containing directional informations. This might be done by giving at each point $x \in \Omega \subset \mathbb{R}^2$ three parameters (six in $\mathbb{R}^3$). This kind of control is equivalent to an isotropic control with a change of the metric tensor all over the domain[1].

It is easy to prove that if K_0 is a non degenerated element then, there exists an unique metric, where K_0 is equilateral with unity edge lengths. So, in order to obtain triangles with a given stretching and size over a subdomain, it is sufficient to construct an isotropic mesh using the metric tensor $\mathcal{M}$ with an associated refinement function h equal to 1 everywhere. The local metric $\mathcal{M}$ is given at every point $x \in \Omega \subset \mathbb{R}^2$ by

$$\mathcal{M}(x) = \mathcal{R}(x) \begin{pmatrix} \lambda_1(x) & 0 \\ 0 & \lambda_2(x) \end{pmatrix} \mathcal{R}(x)^{-1}, \tag{1}$$

where $\lambda_1(x) > 0$ and $\lambda_2(x) > 0$ are the $\mathcal{M}(x)$ eigen-values and $\mathcal{R}(x)$ is the rotation matrix of angle $\alpha(x)$ that maps the $\mathbb{R}^2$ canonical basis over the $\mathcal{M}(x)$ unit eigen-vectors.

Elementary differential geometry says that the length of a parametric curve $\Gamma(t)$ where $t \in [0,1]$ in the new metric is defined by:

$$L(\Gamma) = \int_0^1 \sqrt{\Gamma'(t)^T \mathcal{M}(\Gamma(t))\Gamma'(t)}dt. \tag{2}$$

We now propose the following discretization of the metric tensor $\mathcal{M}$. Let $\mathcal{M}$ be given at vertices $\{\mathcal{M}(x_i),\ i = 1, \ldots, n_v\}$, then over each mesh triangle, K_0, the $\mathcal{M}$ coefficients are linearly interpolated and a continuous discretization of the metric tensor $\mathcal{M}$ is obtained.

Let $\Gamma = [x_0, x_1]$ be a segment, $\Gamma \subset K_0$. Computing its length using equation (2), we obtain

$$\begin{aligned} L(\Gamma) &= \int_0^1 \sqrt{(\Gamma')^T \mathcal{M}(\Gamma(t))\Gamma'}dt & (3)\\ &= \int_0^1 \sqrt{l_0^2 + t(l_1^2 - l_0^2)}dt & (4)\\ &= \frac{2}{3}\frac{l_0^2 + l_0 l_1 + l_1^2}{l_0 + l_1}, & (5) \end{aligned}$$

where $l_i = \sqrt{(\Gamma')^T \mathcal{M}(x_i)\Gamma'}$, $i = 0, 1$.

We now discuss how the metric tensor $\mathcal{M}$ is defined in order to satisfy an adaptation criterion. Suppose that we only work with one variable denoted η. We are going to determine the metric tensor in order to equilibrate the interpolation error.

[1] For the sake of simplicity, we will consider in this section that $\Omega \subset \mathbb{R}^2$. We obtain the same results if Ω defines a surface or if $\Omega \subset \mathbb{R}^3$. In the latest case, 'triangles' are replaced by 'tetrahedrons'.

Assume that a first solution has been computed over a given mesh and suppose that continuous piecewise linear approximation is applied for η, then (see [1] and [2]) the maximum interpolation error depends on the Hessian matrix of η

$$\mathcal{H} = \begin{pmatrix} \partial^2\eta/\partial x^2 & \partial^2\eta/\partial x\partial y \\ \partial^2\eta/\partial x\partial y & \partial^2\eta/\partial y^2 \end{pmatrix} = \mathcal{R} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathcal{R}^{-1}, \tag{6}$$

and a metric tensor $\mathcal{M}$ can be defined by

$$\mathcal{M} = \mathcal{R} \begin{pmatrix} |\lambda_1| & 0 \\ 0 & |\lambda_2| \end{pmatrix} \mathcal{R}^{-1}. \tag{7}$$

Thus, the error over a mesh edge a_i can be computed as

$$E_i \approx |a_i^T H a_i| \leq \sqrt{c_0} a_i^T M a_i. \tag{8}$$

So, we want to construct a mesh where the length of all the edges in the metric defined by $\mathcal{M}$ are near $\sqrt{c_0}$ to equidistribute the error E_i on all the edges i.

We can also associate an ellipse $\mathcal{E}_\mathcal{M}$ to the metric tensor $\mathcal{M}$ having as axes d_1 and d_2 (eigen-vectors of $\mathcal{M}$) and with lengths $1/\sqrt{\lambda_1}$ and $1/\sqrt{\lambda_2}$, respectively. This ellipse $\mathcal{E}_\mathcal{M}$ is the unit ball in this geometry.

Remarks on metric computations:

1) In the metric definition, we have to introduce the maximum and minimum edge lengths we tolerate in the mesh to remove the cases of unrealistic metrics which sometimes we might get. This is not really a restriction as usually we have a good idea of what these quantities should be.

2) As the key point in the metric definition is the second derivatives. But, in our application the solver is a P1 finite element solver. Therefore, a weak formulation (by Green formula) has to be used to compute the Hessians.

3.1 Extension to systems

Suppose now, that several variables $\eta_1,\ \eta_2, \ldots, \eta_r$ are given, the problem becomes:*find the metric so that the maximum interpolation error is minimized for all the variables.* It is clear, from the geometrical identification ellipse–metric that the solution to the previous minimization problem is to find the biggest ellipse contained in the intersection of all the ellipses $\mathcal{E}_1,\ \mathcal{E}_2, \ldots, \mathcal{E}_r$ corresponding to metrics $\mathcal{M}_1,\ \mathcal{M}_2, \ldots, \mathcal{M}_r$ computed from the variables $\eta_1,\ \eta_2, \ldots, \eta_r$. It is not easy in general to find the optimal solution of this problem. However the following algorithm seems to be suitable enough (see fig. 1). Suppose that initially only two variables (η_1 and η_2) are provided. We find an approximation of the optimal intersection ellipse by the following procedure:

- if the two ellipses curves $\mathcal{E}_1,\ \mathcal{E}_2$ have not any intersection point (this happens when one ellipse is contained into another), the one with smallest area is taken as intersection and the suitable metric is the associated one;
- otherwise, let λ_i^j and v_i^j, $i, j = 1, 2$ the eigen-values and eigen-vectors (resp.) of $\mathcal{M}_j$, $j = 1, 2$. The intersection metric ,$\hat{\mathcal{M}}$, is defined by

$$\hat{\mathcal{M}} = \frac{\hat{\mathcal{M}}_1 + \hat{\mathcal{M}}_2}{2}. \tag{9}$$

  where $\hat{\mathcal{M}}_1$ (resp. $\hat{\mathcal{M}}_2$) has the same eigen-vectors as $\mathcal{M}_1$, $(v_1^1,\ v_2^1)$ and has for eigen-values

$$\tilde{\lambda_i^1} = \max(\lambda_i^1, {v_i^1}^T \mathcal{M}_2 v_i^1), \quad i = 1, 2. \tag{10}$$

Now, if n variables are given, the final intersection metric is computed as follows:

- $\hat{\mathcal{M}} = \text{intersection}(\mathcal{M}_1, \mathcal{M}_2)$.
- for $i = 3, \ldots, n$
  $\hat{\mathcal{M}} = \text{intersection}(\hat{\mathcal{M}}, \mathcal{M}_i)$.

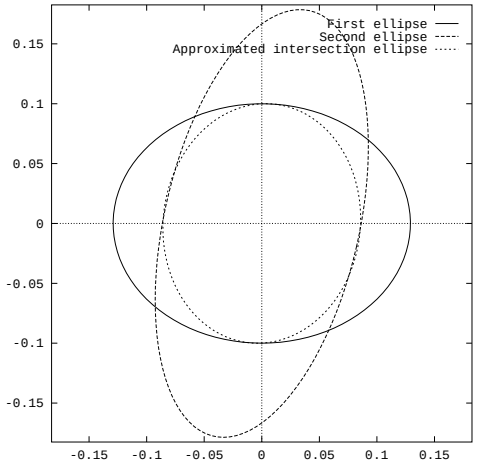


Figure 1: Approximated optimal ellipse of two-ellipse intersection.

3.2 Boundary layer improvement

As said previously, metric evaluation is done using Green's formula with homogenous Neumann boundary condition on all boundaries. This is quite logical and means that elements do not change in size and shape near boundaries. But our experience shows that this is not suitable for viscous computation in boundary layers. This is why structured meshes are usually perfered for these regions.

We have then tried to give a new boundary condition for the metric on the walls to improve the mesh definition in near-wall area. More precisely, along the wall the previous metric $\mathcal{M}(x)$ is replaced by a new metric $\hat{\mathcal{M}}(x)$:

$$\hat{\mathcal{M}}(x) = T\Lambda T^{-1},$$

where

$$\Lambda = diag(h_{wall}, h_\tau) \quad \text{and} \quad T = (\vec{n}(x), \vec{\tau}(x)).$$

In other words, along solid walls the eigen-vectors are now the unit normal and tangent vectors at the wall with h_{wall} and h_τ the corresponding eigen-values. The refinement along the wall can be now monitored through h_{wall}. More precisely, in the adapted mesh the distance from the wall of the first layer nodal will be h_{wall}. This quantity is given by the user and depends on the Reynolds number. On the other hand, h_τ is computed taking the maximum of the two projection lengths on $\vec{\tau}(x)$ of the $\mathcal{M}$ unit eigen-vectors times the corresponding eigen-values.

It is necessary to keep the information coming from the original metric on h_τ to be able to recognize situations such as shock-boundary layer interactions.

A relaxation procedure can now be applied to propagate this orthogonality property further in the mesh. In this paper, this has been done for only two nodal layers. Figuure (11) shows the distance of the points, belonging to these layers, to the wall for a supersonic viscous flow over an airfoil. To give an idea on what we get without this boundary condition on the metric, the corresponding result with only homogeneous Neumann boundary condition for the metric along the wall has also been shown in the same figure.

4 Mesh generation

In this section we give a brief description of our mesh generator, more details can be found in [3] and [4].

Once the metric is given, the mesh generation procedure is applied. This algorithm uses five different local tools for this task: *edge suppression, vertex suppression, vertex addition, edge swapping and vertex reallocation (barycentering step).* We are going to work with three different grids: one defining the domain geometry (*'C.A.D. mesh'*, $\mathcal{T}_G$) which is fixed during the mesh adaptation loop, one that contains the control space informations (i.e. the metric tensor definition) denoted by $\mathcal{T}_0$ and the adapted mesh $\mathcal{T}_1$. We can say that this algorithm solves the following problem:

Find "an optimal mesh" $\mathcal{T}_1$ adapted from $\mathcal{T}_0$ with respect to the criterion imposed by the metric tensor $\mathcal{M}$ and closed to $\mathcal{T}_G$.

So, we suppose that the meshes $\mathcal{T}_G$ and $\mathcal{T}_0$ are given together with the metric tensor $\mathcal{M}$ obtained from a finite element solution over $\mathcal{T}_0$, and we want to compute $\mathcal{T}_1$.

Initially, we have $\mathcal{T}_1 = \mathcal{T}_0$, but $\mathcal{T}_G$ can of course be different of $\mathcal{T}_0$.

Different steps can be distinguished in our mesh adaptation procedure:

Data structure creation and verification. Before the grid adaptation process, it will be necessary to create some auxiliary data, as: mesh connectivity arrays, boundary and inter-subdomain (geometrical and physical)[2] edges with their associated tangent vectors, fixed point localization, etc. A topological verification step together with a triangle orientation is made in order to guarantee that the given meshes are correct and oriented.

Geometry reconstruction. Generalized Farin's algorithm is used in order to define a 'G^1' Bézier surface over $\mathcal{T}_G$.

Initial regularization. Depending on user choices, an initial optimization step (edge swapping) is made in order to improve the initial mesh quality. Numerical experiences show that the final result is better if this initial optimization is made.

Grid adaptation. The topological mesh elements related with $\mathcal{T}_1$ are ordered in a particular data structure called *double dynamic list (DDL).* Three double lists are considered (vertices, edges and triangles). Suppose that the first $i-1$ edges are treated, then the edge a_i is taken. The procedure that we propose is the following one:

1. Let d_i be the a_i length computed with the metric tensor $\mathcal{M}$. We have three possibilities:
   - If $d_i > l_{max}$ ($l_{max} \approx 1.4$) then a_i must be cut in two edges using the add-vertex procedure. The new mesh elements (vertices, edges and triangles) are added at the end of their corresponding DDL. Now, a_i has changed and we again compute its length, d_i. If $d_i > l_{max}$ then we repeat the same process until its associated length $d_i \leq l_{max}$.
   - If $d_i < l_{min}$ ($l_{min} \approx 0.6$) then a_i is suppressed, this leading to new edges creations. We check if their lengths are bigger than l_{max}, in that case the previous process is applied, or smaller than l_{min} where this procees is again repeated.
   - if $l_{min} \leq d_i \leq l_{max}$, a_i is kept.
2. If we have done a local mesh modification, a local optimization step must be considered. In this case, edge swapping algorithm is applied to the new or modified edges. This optimization step increases the mesh quality.

This process is repeated over all the edges of DDL.

Final optimization. Finally, a global optimization process is considered. All non fixed, non inter-subdomain interior vertices with less that 4 elements connected to it are suppressed and a barycentering

[2]The first ones correspond to intersections of regular surfaces and the second ones correspond to intersections of user-defined subdomains.

step is applied: all non fixed vertices are put in the center of gravity of their neighbors with respect to the metric $\mathcal{M}$.

Thus, a final mesh $\mathcal{T}_1$, adapted from $\mathcal{T}_0$ with respect to the metric tensor $\mathcal{M}$ is obtained and it respects the geometrical and topological constraints. After each adaptation step, an initial solution is obtained over the new mesh by interpolation of the solution on the previous mesh. This point is crucial as we will see especially for unsteady computations. As we can see, the remeshing algorithm always works locally. This strategy allows us to remesh surfaces with only a local parametrization.

5 Numerical examples

In this section we describe three configurations of compressible inviscid and viscous flows. For all these cases, the initial meshes have been generated using EMC2 [8].

5.1 Supersonic scramjet inlet

The first case consists of an internal supersonic flow at Mach 3 in a scramjet inlet. Despite the configuration being symmetric, the whole domain has been computed to see if the solution remains symmetric or not. This case is devoted to an evaluation of our multi-variable strategy for the metric definition. The second modification is not used here as it is an inviscid flow.

Five automatic adaptations have been applied. We show the meshes and the corresponding iso-Mach contours from the initial to the last adaptation step in figures 2, 3, 4 and 5. We can see how accurate the discontinuities resolutions are for this flow at step 5. It is interesting to notice that there is no oscillation in the solution and that the fluid solver seems to nicely accept such meshes. Moreover, we see that the solution stays perfectly symmetric even on a clearly non-symmetric mesh.

The initial mesh has around 8000 triangles and the final mesh around 40000. This gives an idea of the number of elements we might need if we want an equivalent resolution on a structured grids. The convergence history is shown in figure 6.

5.2 Supersonic viscous flow over a NACA0012

A Mach 2 supersonic flow at Reynolds 10000 is studied over a classical Naca 0012 airfoil. In this configuration, both the new ingredients we propose here are used. The quality of wall coefficients (especially the friction coefficient) is studied too. In particular, we want to see how much they depend on the mesh.

The adaptation loop length is 6 for this case. The convergence history is shown in figure 10. We can see that the perturbation due to the adaptation on the residual is more important that in the last case. This comes from first the fact that the flow is viscous and second that we have our new boundary condition for the metric along the wall.

The meshes we obtain during the adaptation loop are shown in figure 8 and the corresponding iso-Mach distributions in figure 9. We can see that the multi-variable intersection procedure works well for the identification of shock, boundary layer and wake regions, even the weak shocks at the trailing edge have been detected.

Figure 11 shows the normal distance of the first and second nodes layers to the wall. We can see first that for the first layer this distance is exactly the same for all the nodes which means that our boundary condition on the metric works well and second that the relaxation procedure we have applied to propagate this properties to the next layers also works quite well. For these computations we have just propagated the property of being at the same distance to the wall only to the first two layers but the same procedure can be applied on the other layers. In order to show what we obtain without this boundary condition on the metric, we have also shown the corresponding distances to the wall for the same layers after switching off our correction (i.e. homogeneous Neumann b.c. for the metric)

cooresponding results becomes very poor. This explains the poor quality of the gradients evaluated at the wall until now on adapted meshes and the fact that nice friction coefficients were hard to obtain.

Figure 12 shows the pressure and friction coefficients obtained on the initial mesh, on the mesh at step five and on the last one. Results for steps 5 and 6 are identical.

5.3 Unsteady flow around a cylinder

This is a subsonic flow at Mach 0.4 and Reynolds 80. Our aim here is to see how the mesh adaptation procedure follows an unsteady phenomena.

We make a first computation on a initial mesh having around 6000 triangles until the unsteady flow is established. The adimensional period of the movement being around 7, mesh adaptation has been carried out eight times in a period. These meshes and the corresponding iso-Mach contours are shown in Figures 13 to 16. We can see the mesh adaptation procedure follows the unsteadiness of the flow while the number of nodes remains almost constant.

There is a major difficulty in this computation compared to the previous steady cases. Indeed, as we saw in the convergence history for the two last cases, at each adaptation step, the residual was perturbated. This is due to the fact that the linear interpolation of the solution on the new mesh cannot be compatible with the nonlinear nature of the solution and the equations need time to recover their level of resolution. Basically, we need a relaxation procedure after each adaptation for the solution to be coherent with the previous resolution level. In this work, nothing has been done for this purpose, therefore the time evolution of quantities like the drag or lift coefficients presents the same behaviour than a convergence history in steady computations. More investigations are therefore needed to correctly extend this approach to these flows.

5.4 Subsonic flow around a 4-element airfoil

This case consists of a viscous subsonic flow at Mach 0.2 and Reynolds 10^5 with 8.35 degrees of incidence. The flow is quite complex and unsteady but we have not tried to capture the unsteady features.

Six adaptations have been done for this case. The evolution of the meshes and the iso-Mach contours are shown in figure 17 and 18. The first mesh has around 13000 triangles and the last one around 50000. We can see how complicated is the flow patern at step 6. Convergence history and pressure coefficient distribution are shown in figures 19 and 20. Despite the flow solver is explicit, the convergence is obtained for this complex flow with only 4000 iterations which is quite cheap. Moreover, only 2000 iterations have been done on the finest levels. This represents another interesting consequence of mesh adaptation on convergence speed-up.

6 Concluding remarks

Anisotropic mesh adaptation has been applied with success to compressible viscous flows in a wide range of the Reynolds and Mach number. In particular, two new ideas have been described to remove two basic difficulties with adapted meshes for flow computations concerning multiple physical interactions and boundary layers. It has been clearly shown that these modifications greatly improve the quality of the adapted meshes for these situations.

These ideas are simple to take into account in any adaptation procedure because they involve only local and simple modifications of codes. Therefore, they can be easily applied by other users. In the same way, their extension to 3D configurations is straightforward.

Another important conclusion is that for steady computations, convergence is improved when using mesh adaptation. Indeed, as we saw only a few thousand of iterations were necessary to converge for both the Euler and Navier-Stokes cases on quite large meshes with only a few of them on the finer

levels. We recover there some kind of multigrid behaviour but with a set of independant refined levels where only interpolations (Coarse to Fine) are used but not projections (Fine to Coarse).

Therefore, the global CPU time is significantly reduced compared to a direct computation on a fine mesh.

References

[1] E. F. d'Azevedo and R. B. Simpson, 'On Optimal Interpolation Triangle Incidences'. *SIAM'S Journal on Scientific and Statistical Computing*,

no. 6: pp. 1063–1075, (1989).

[2] E. F. d'Azevedo and R. B. Simpson, 'On Optimal Triangular Meshs for Minimizing the Gradient Error'. *Numerische Mathematik* **59**, no. 4: pp. 321–348, (1991).

[3] M.J. Castro Díaz, 'Mesh refinement over surfaces'. *Rapport de recherche INRIA, Rocquencourt* no. 2462, (1994).

[4] M.J. Castro Díaz and F. Hecht, 'Anisotropic surface mesh generation'. *To appear* (1995).

[5] F. Coquel, 'Hybrid Upwind Scheme'. *To appear* (1994).

[6] P.L. George, *Automatic mesh generation, application to Finite Element Methods*. Wiley & Sons, Chichester, 1991.

[7] P.L. George, F. Hecht and E. Saltel, 'Automatic mesh generation with specified boundary'. *Computer methods in applied mechanics and engineering*. **92**: pp. 269–288 (1991).

[8] F. Hecht and E. Saltel, 'EMC^2 un logiciel d'édition de maillages et de contours bidimensionnels'. *Technical report INRIA, Rocquencourt*, no. RT-0118 (1990).

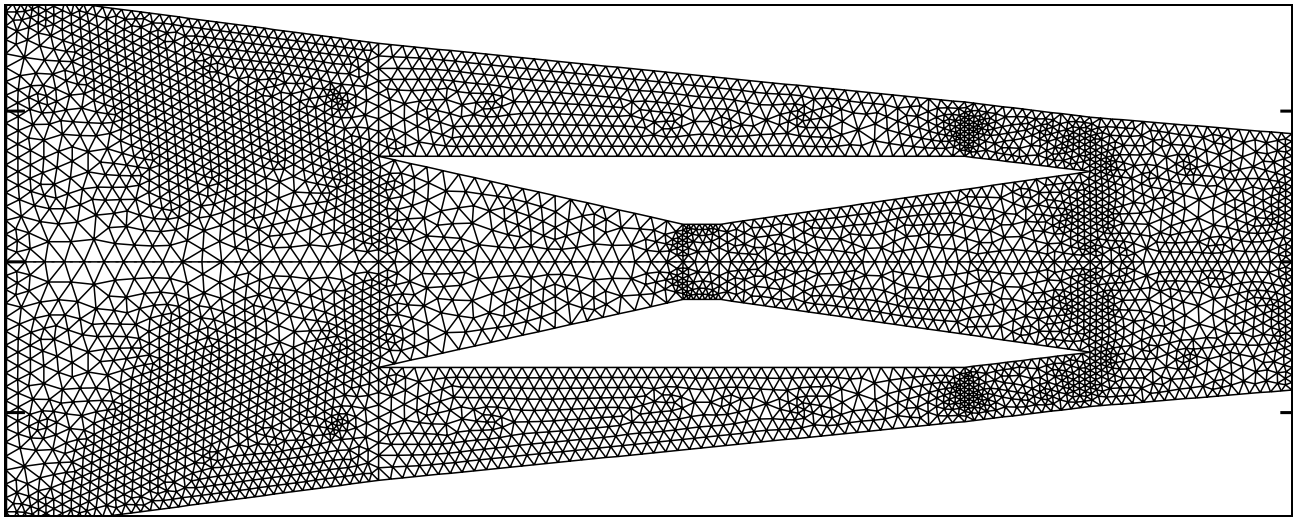
[9] B. Mohammadi, 'CFD with NSC2KE: An User-Guide'. *Technical report INRIA, Rocquencourt* no. RT-0164 (1994).

[10] B. Palmerio and A. Dervieux, 'Application of a FEM Moving Node Adaptive Method to Accurate Shock Capturing'. *Numerical Grid Generation in CFD*. (1986).

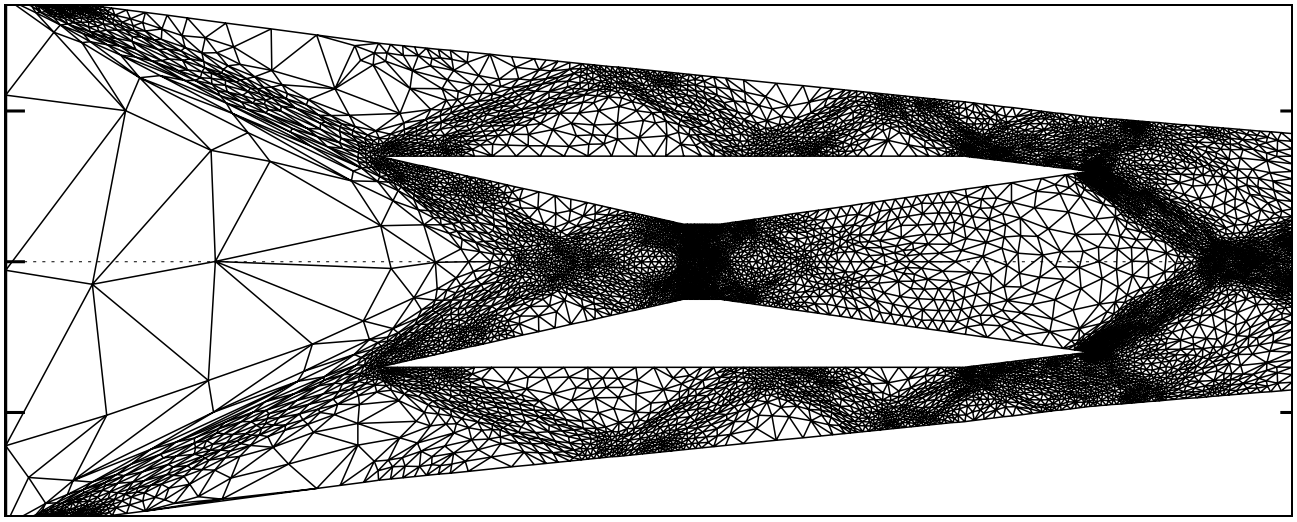
[11] J. Steger and R.F. Warming , 'Flux Vector Splitting for the Inviscid gas dynamic with Applications to Finite-Difference Methods'. *Journal Comp. Phys.* no. 40, pp: 263-293 (1982).

[12] M.G. Vallet, *Génération de maillages Éléments Finis anisotropes et adaptatifs*. Université Paris VI, Paris, 1992.

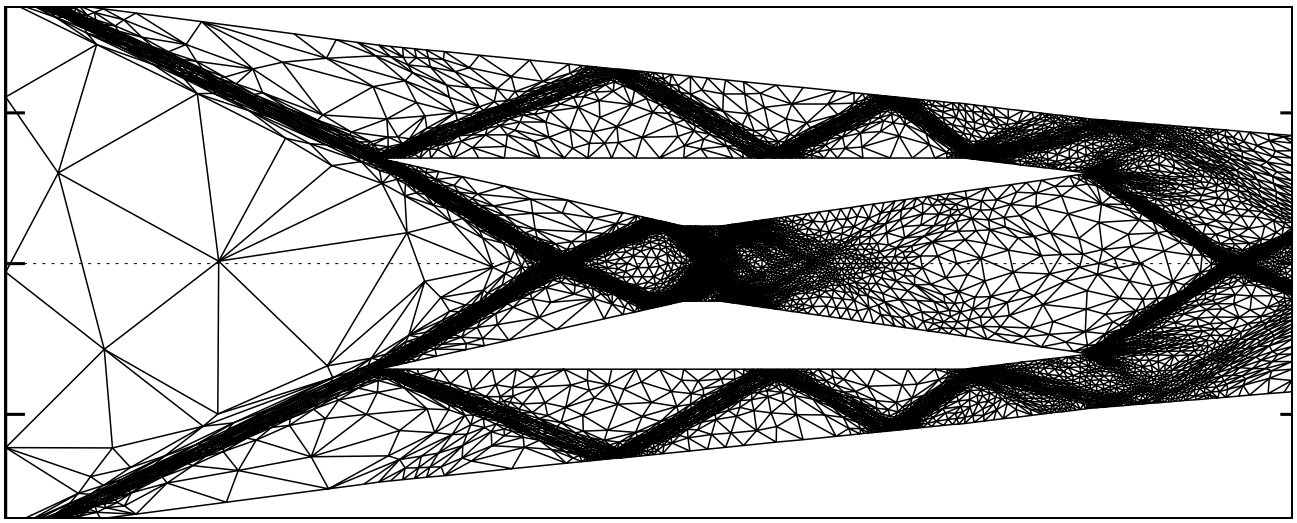
[13] G.D. Van Albada and B. Van Leer, 'Flux Vector Splitting and Runge-Kutta Methods for the Euler Equations'. *ICASE Report* no. 84-27 (1984).



Initial mesh (4532 vertices).

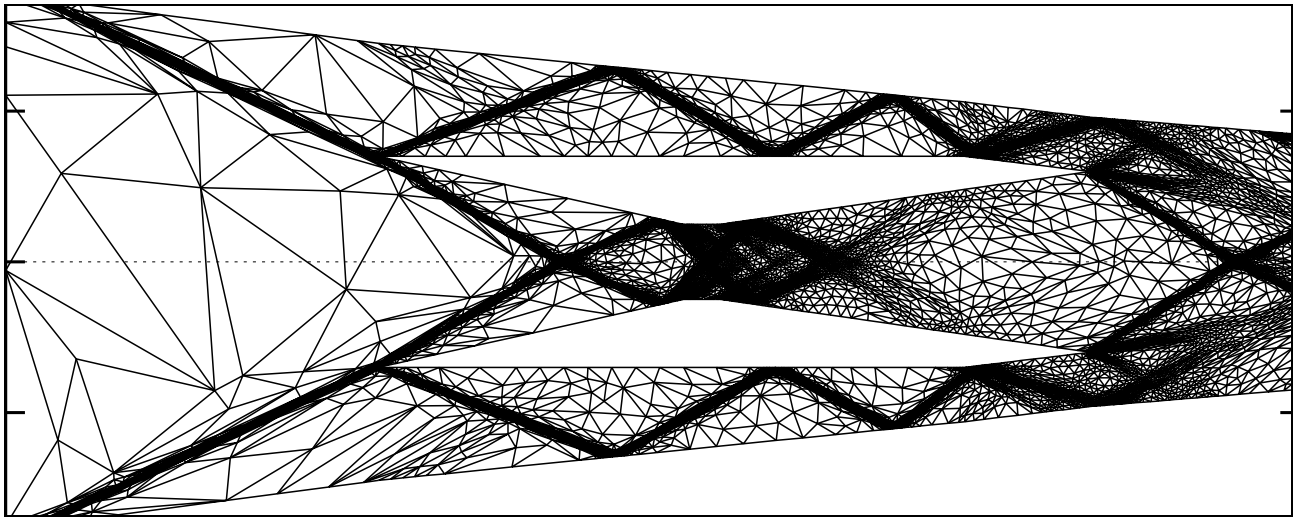


First adaptation (8614 vertices).

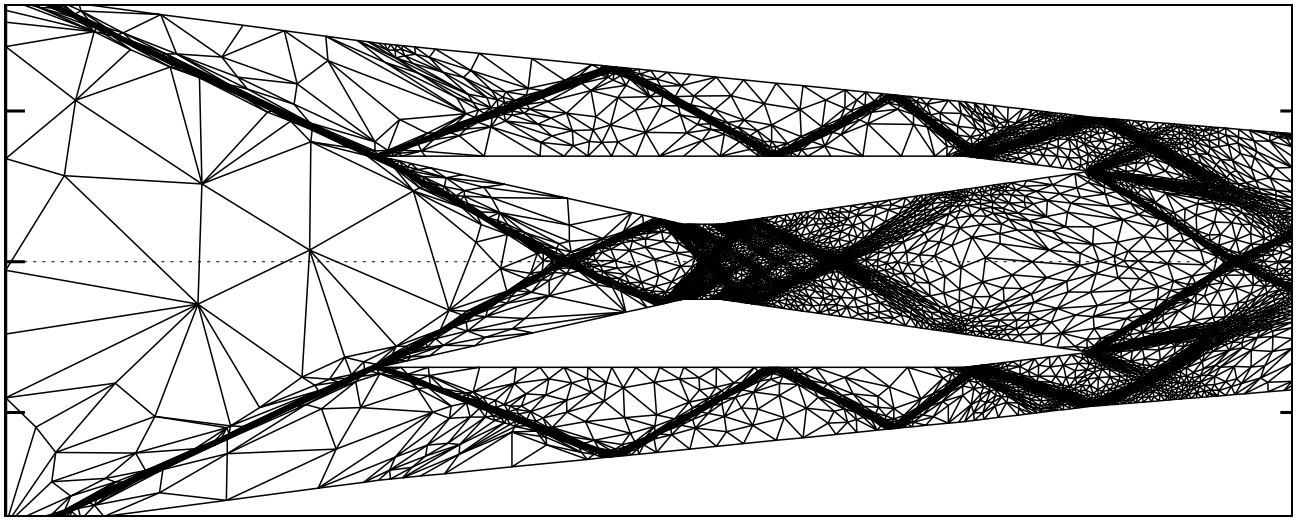


Second adaptation (13939 vertices).

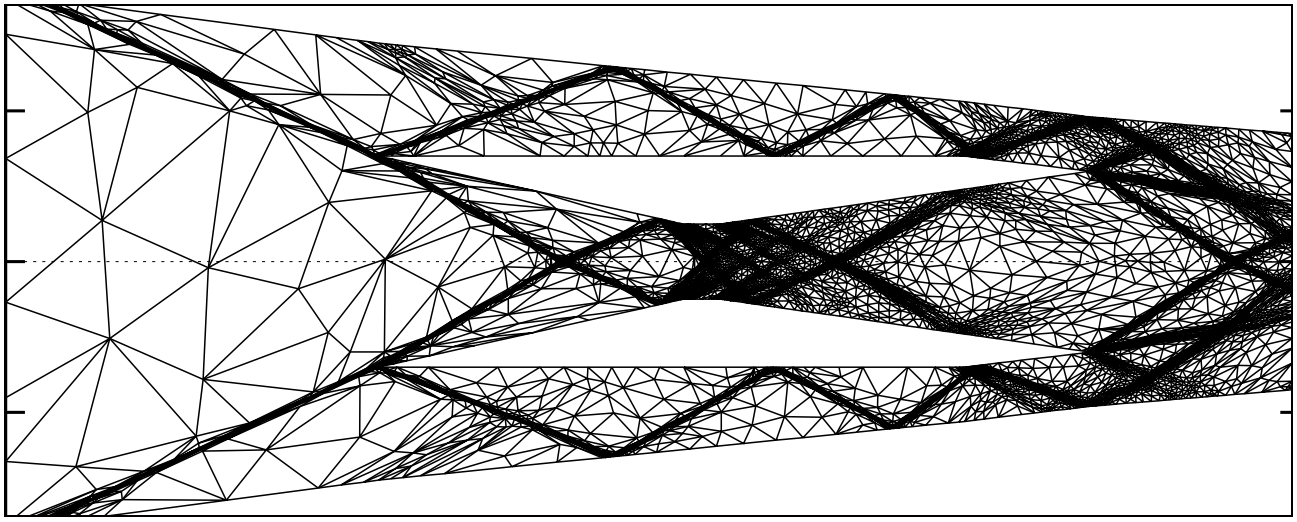
Figure 2: Evolution of the Mesh for the two first consecutive adaptations for a supersonic scramjet inlet.



Third adaptation (20802 vertices).

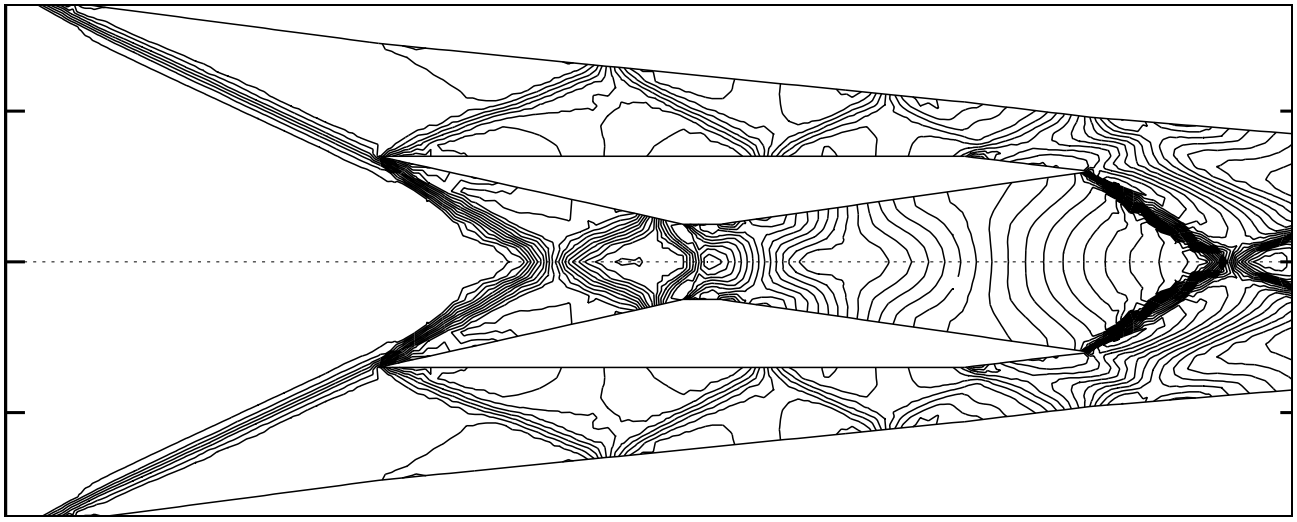


Fourth adaptation (22495 vertices).

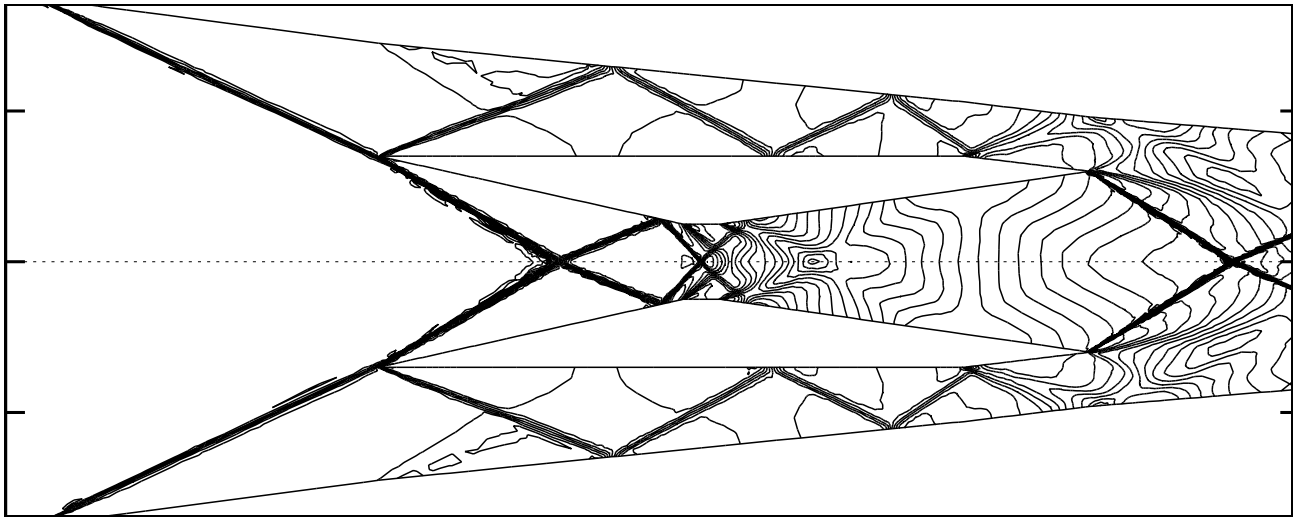


Fifth adaptation (23970 vertices).

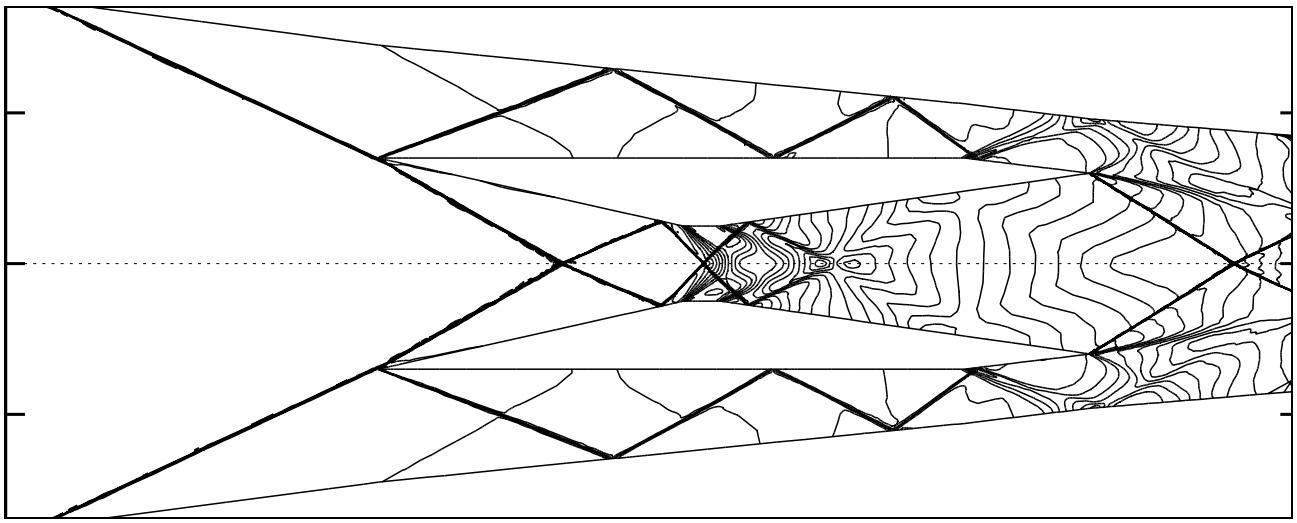
Figure 3: Evolution of the Mesh for the last three consecutive adaptations for a supersonic scramjet inlet.



Iso-Mach contours on the Initial mesh.

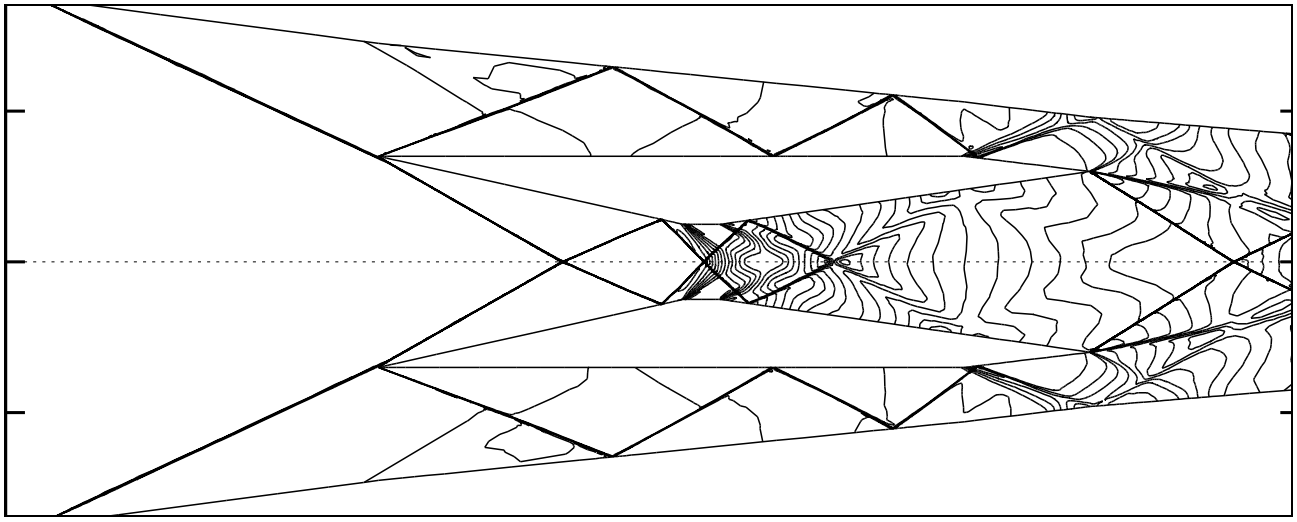


Iso-Mach contours after 1 adaptation.

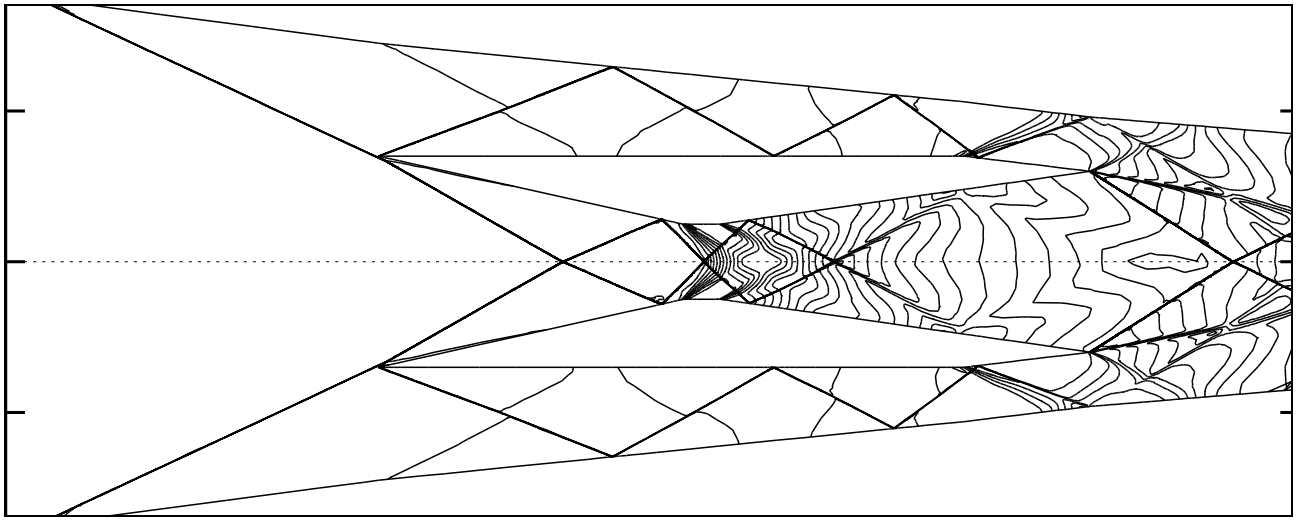


Iso-Mach contours after 2 adaptations.

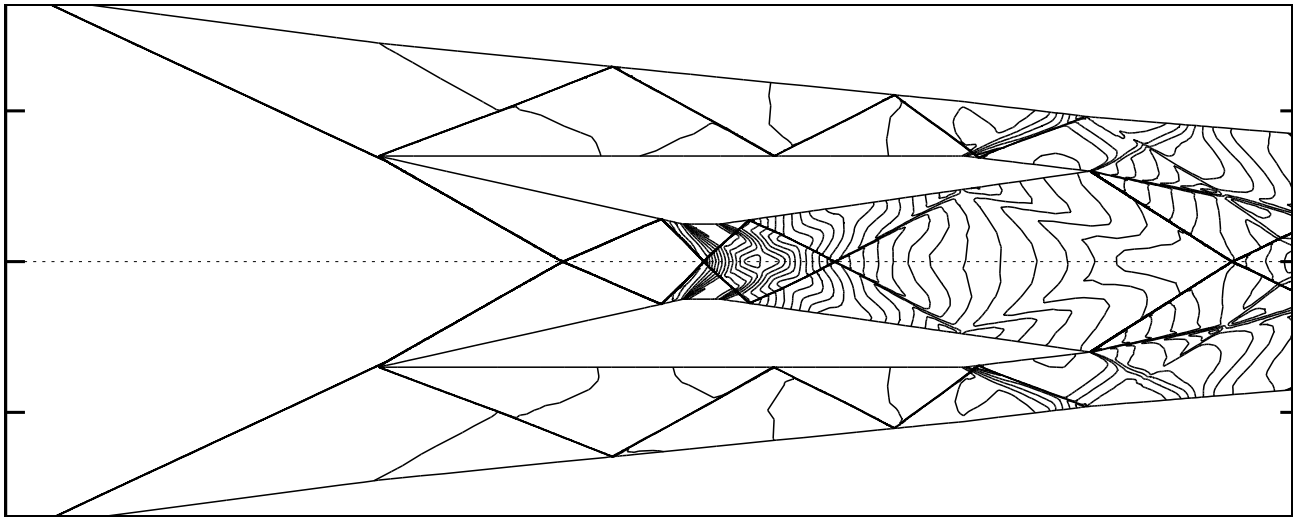
Figure 4: Evolution of iso-Mach contours for an inviscide Mach 3 scramjet inlet.



Iso-Mach contours after 3 adaptations.



Iso-Mach contours after 4 adaptations.



Iso-Mach contours after 5 adaptations.

Figure 5: Iso-Mach contours obtained on the last three adapted meshes for an inviscide Mach 3 scramjet inlet.

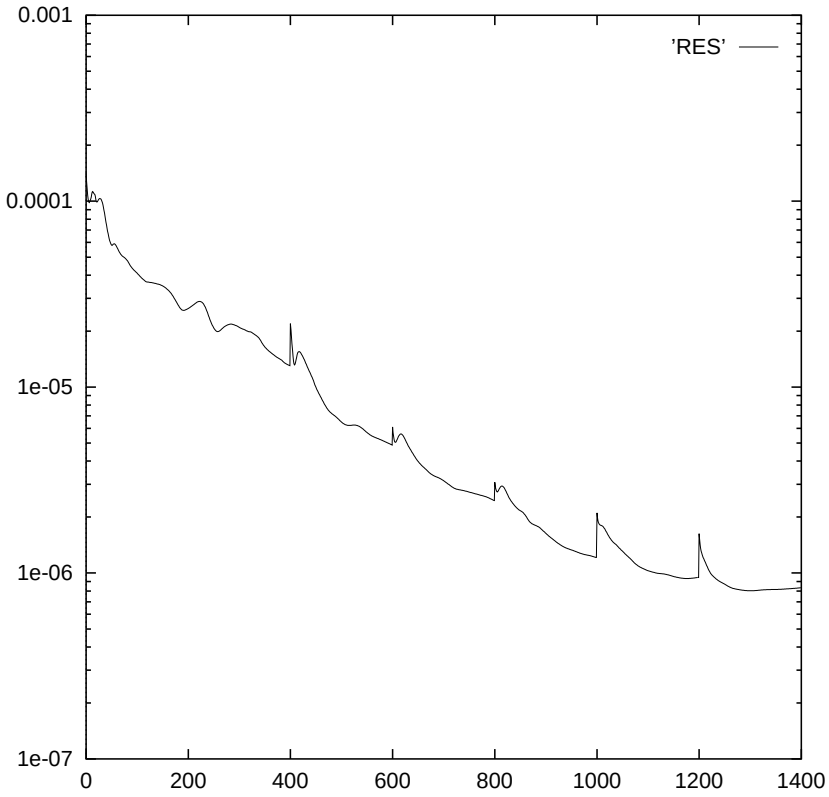


Figure 6: Convergence history.

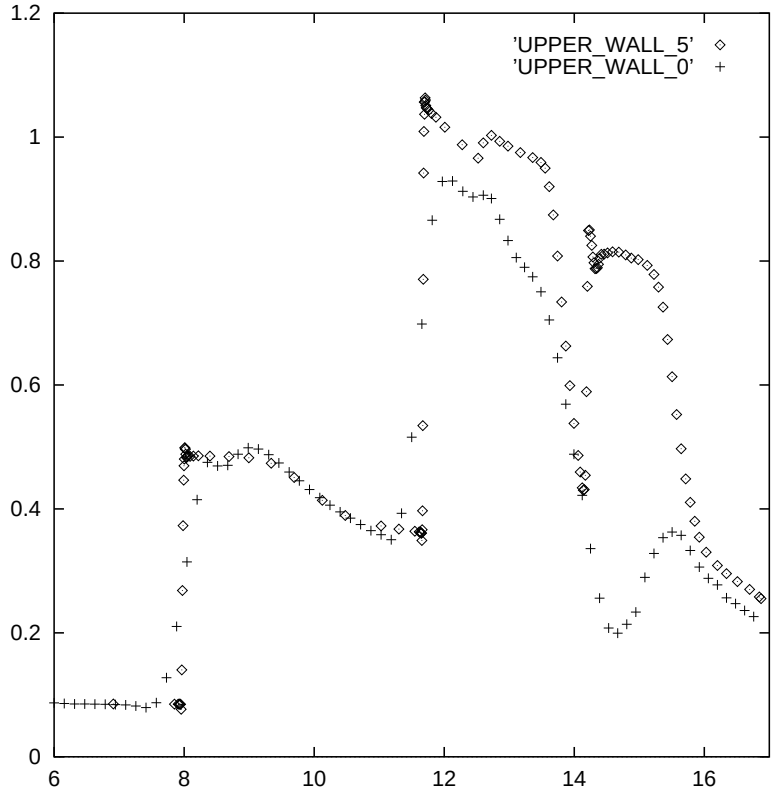
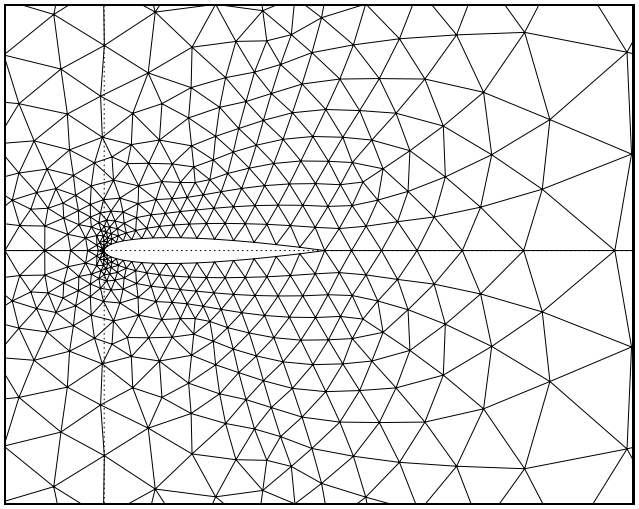
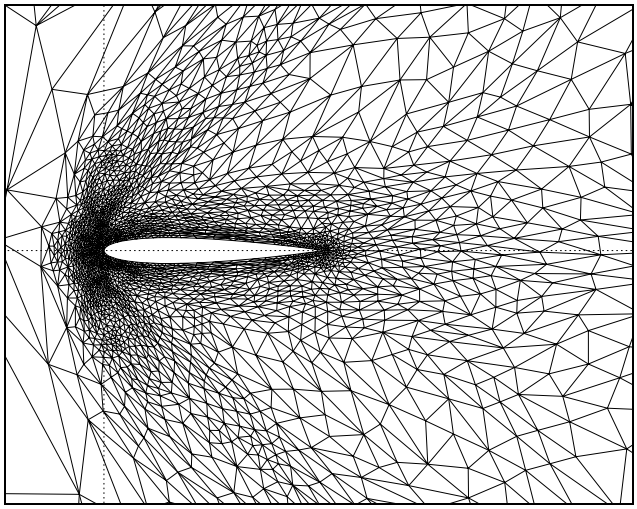


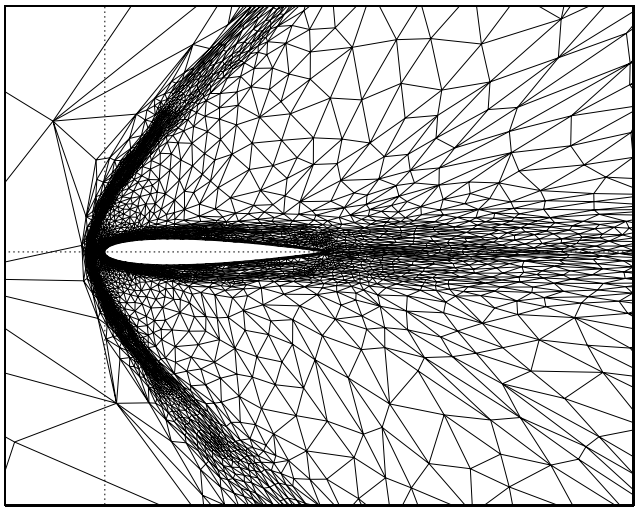
Figure 7: Partial vue of the pressure coefficient on the initial and final meshes.



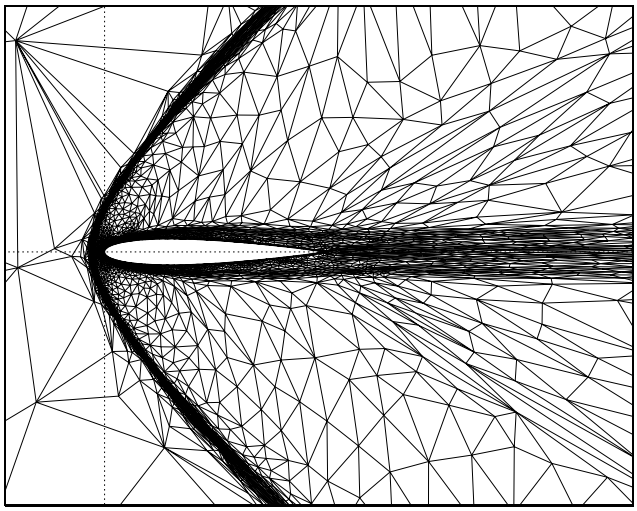
Initial mesh (801 vertices).



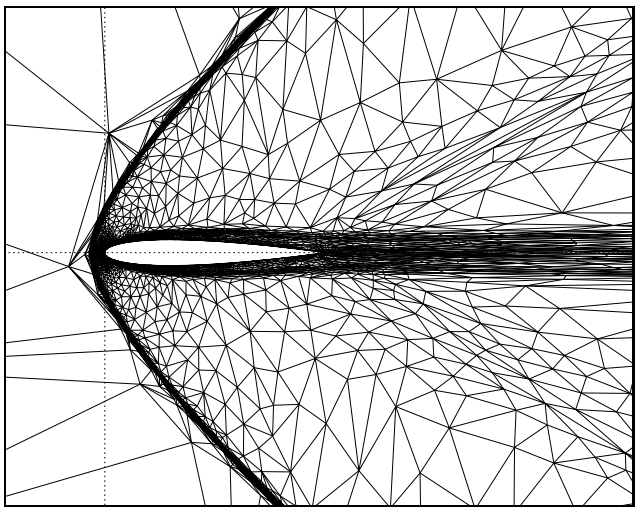
First adaptation (2794 vertices).



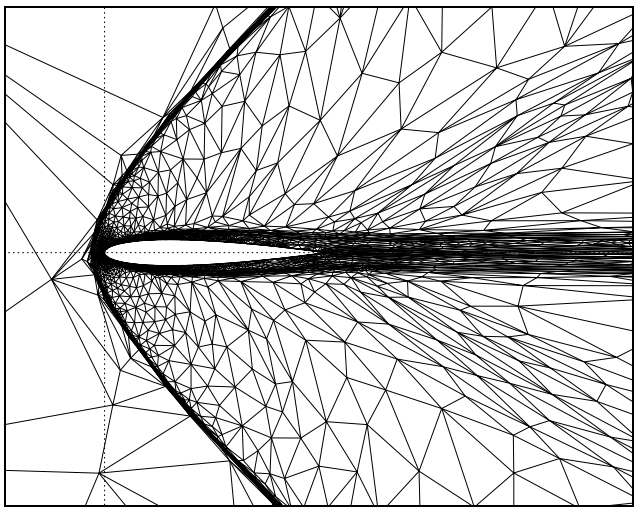
Second adaptation (5119 vertices).



Third adaptation (7543 vertices).

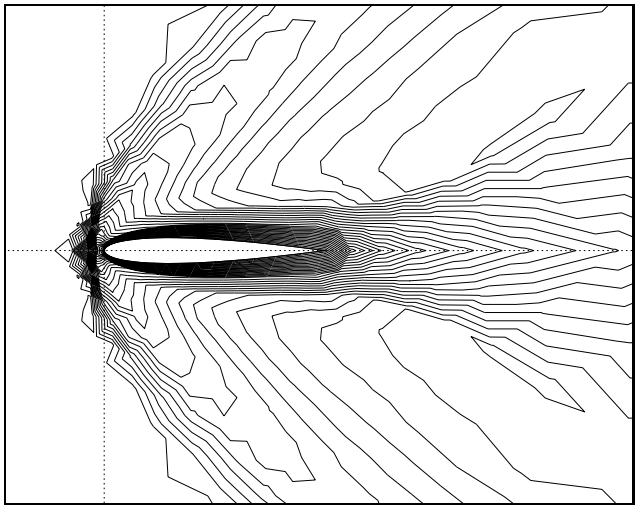


Fourth adaptation (12132 vertices).

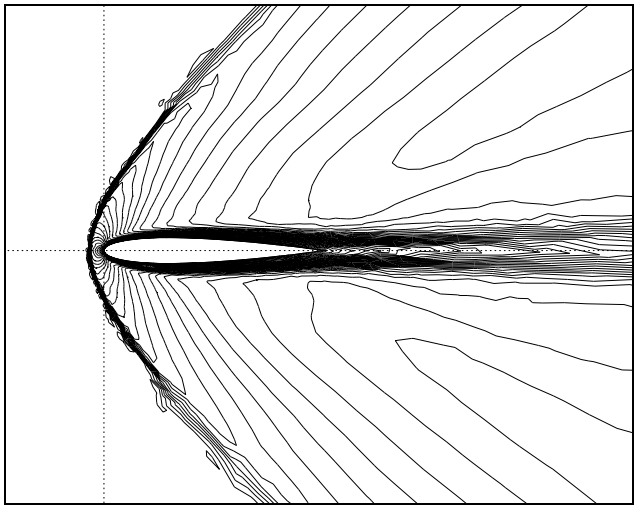


Fifth adaptation (11057 vertices).

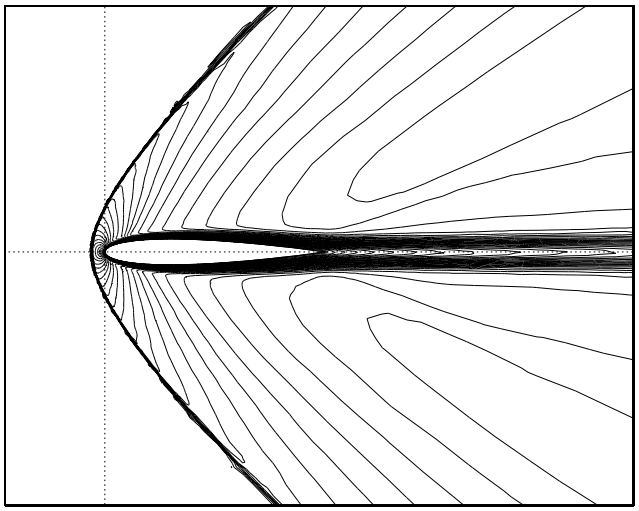
Figure 8: Evolution of the Mesh for five consecutive adaptations for a supersonic viscous flow over an airfoil.



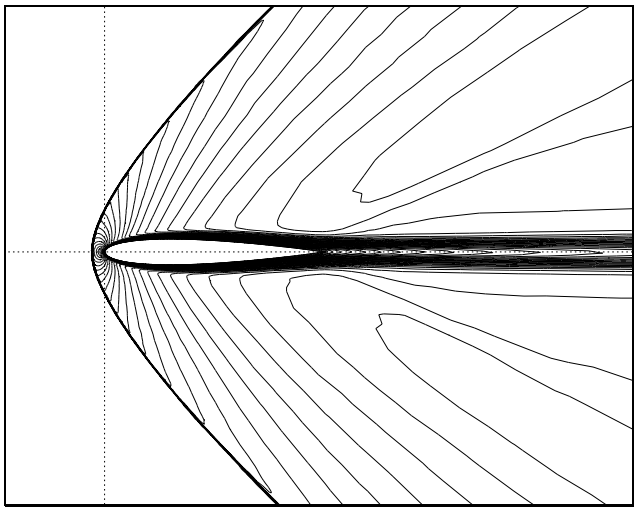
Iso-Mach contours on the Initial mesh.



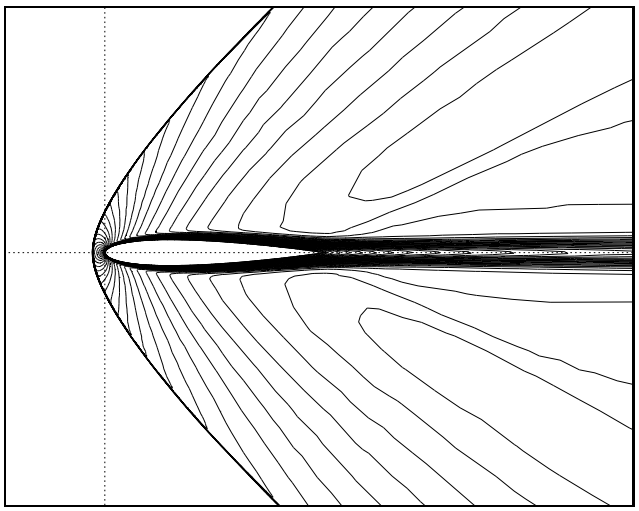
Iso-Mach contours after 1 adaptation.



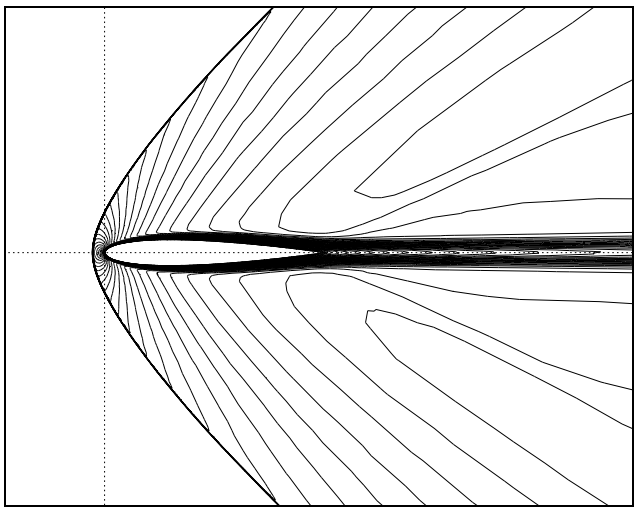
Iso-Mach contours after 2 adaptations.



Iso-Mach contours after 3 adaptations.

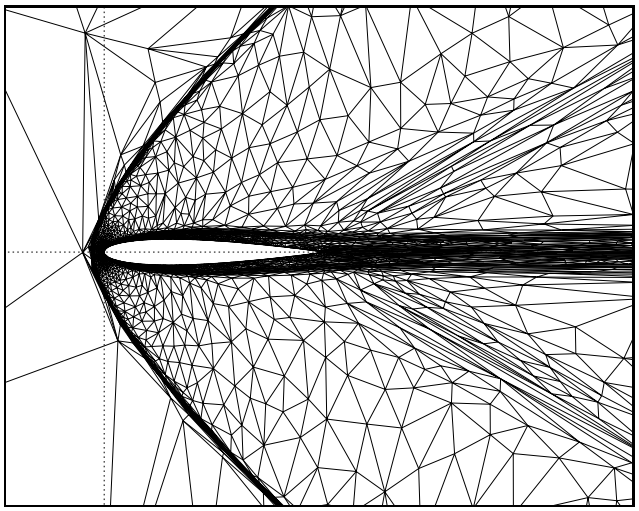


Iso-Mach contours after 4 adaptations.

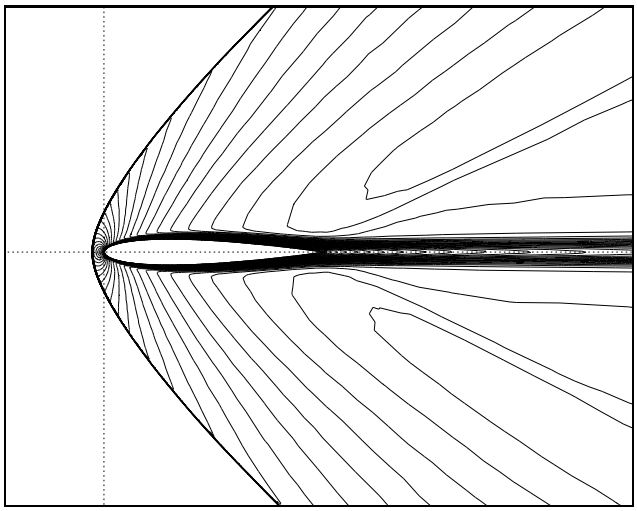


Iso-Mach contours after 5 adaptations.

Figure 9: Iso-Mach contours obtained on the five first adapted meshes for a supersonic viscous flow over an airfoil.



The final mesh at step 6,
13202 vertices, 25901 triangles



The final iso-mach contours.

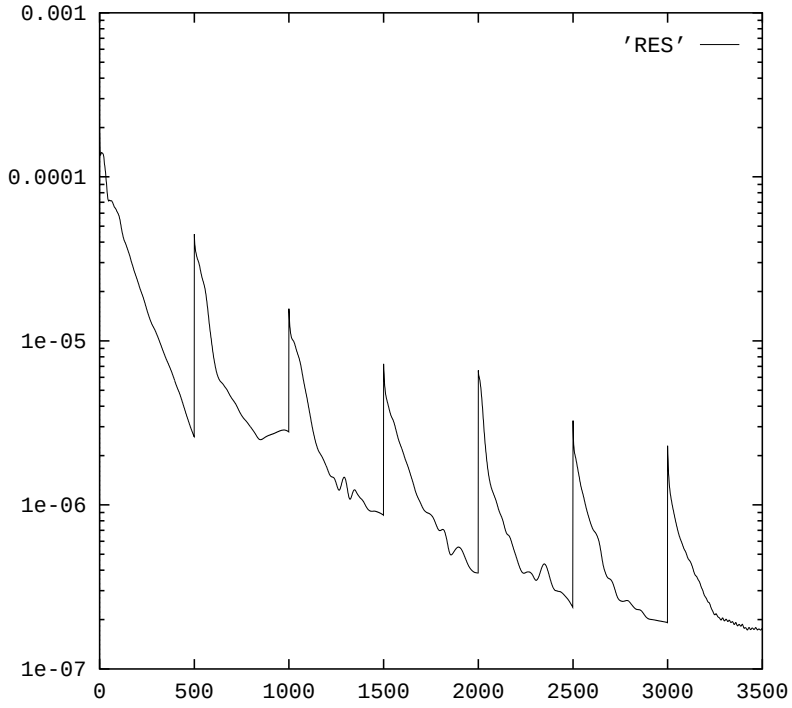


Figure 10: Convergence history.

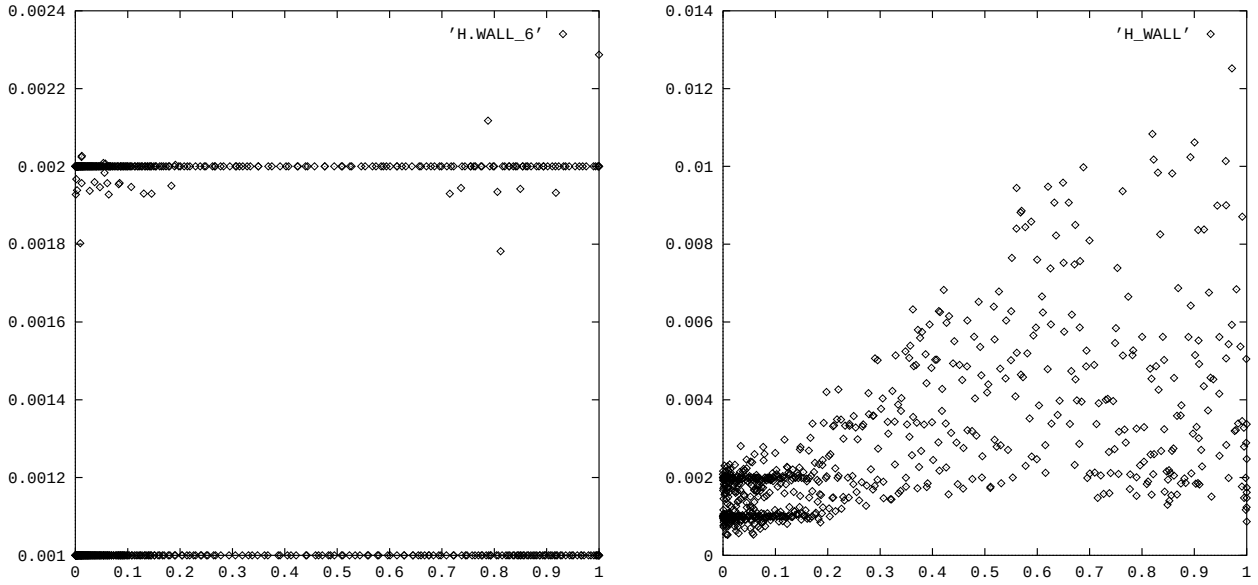


Figure 11: hwall with and without our Dirichlet b.c.

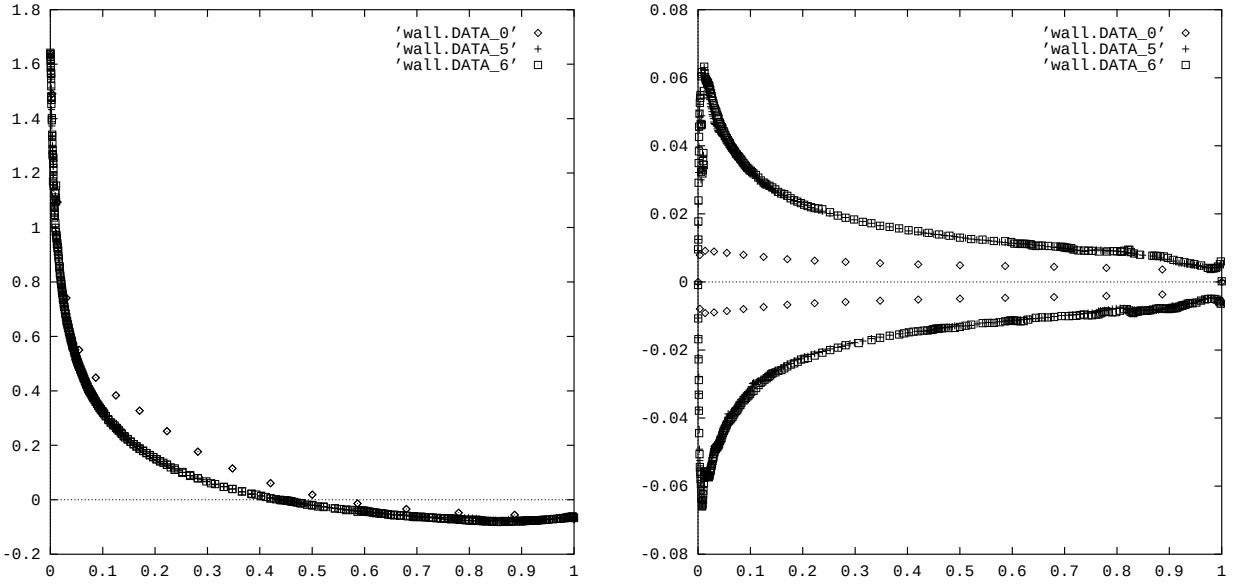
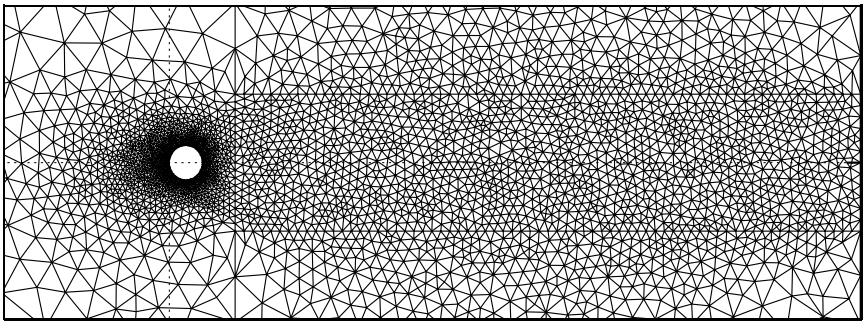
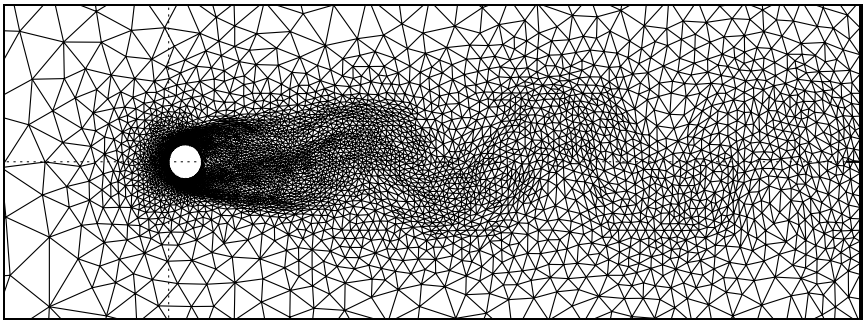


Figure 12: C_p at step 0 (initial), 5 and 6. C_f at step 0 (initial), 5 and 6.

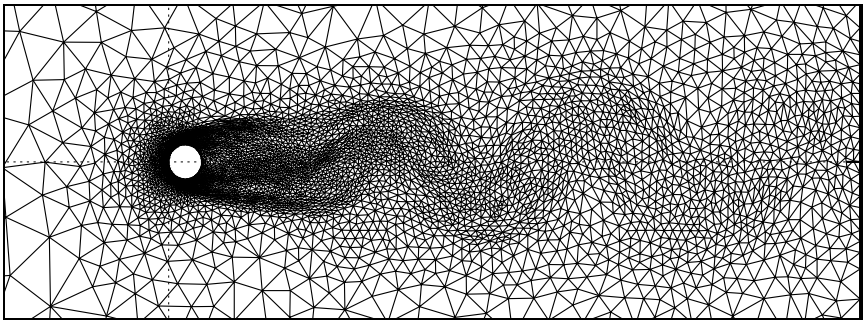
Influence of our Dirichlet boundary condition for the metric on wall quantities.



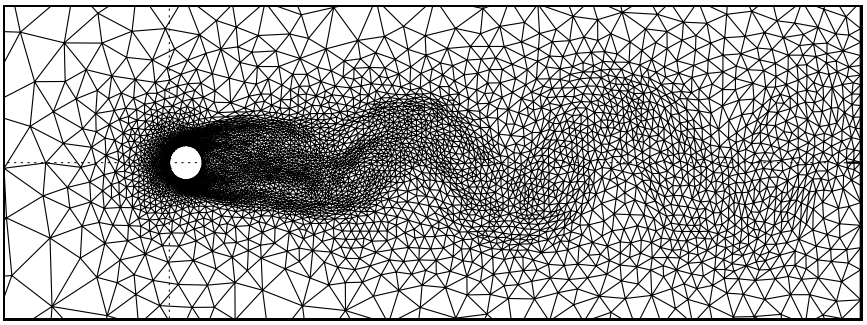
Initial mesh (3724 vertices).



2^{nd} adaptation (5155 vertices).

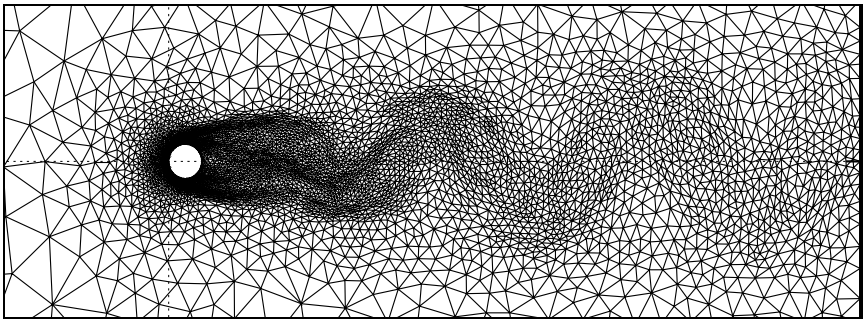


3^{rd} adaptation (5310 vertices).

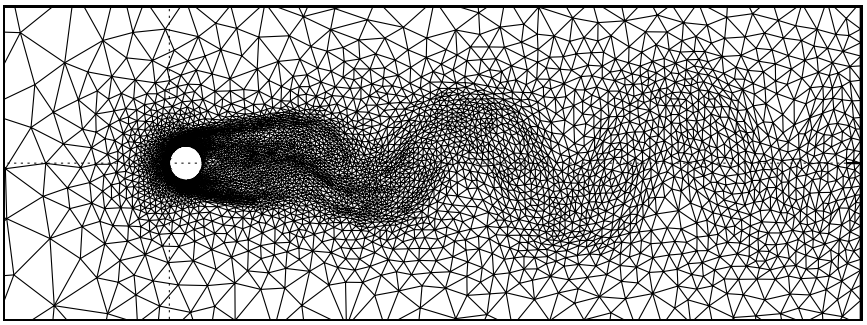


4^{th} adaptation (5355 vertices).

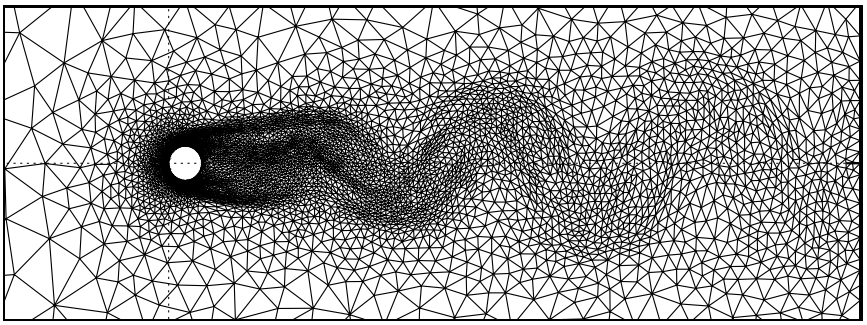
Figure 13: Time dependent evolution of the mesh over one period.



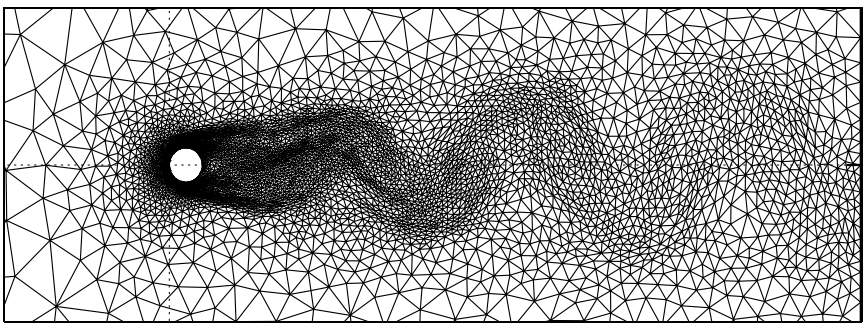
5^{th} adaptation (5437 vertices).



6^{th} adaptation (5474 vertices).



7^{th} adaptation (5528 vertices).



8^{th} adaptation (5581 vertices).

Figure 14: Mesh evolution (continued).

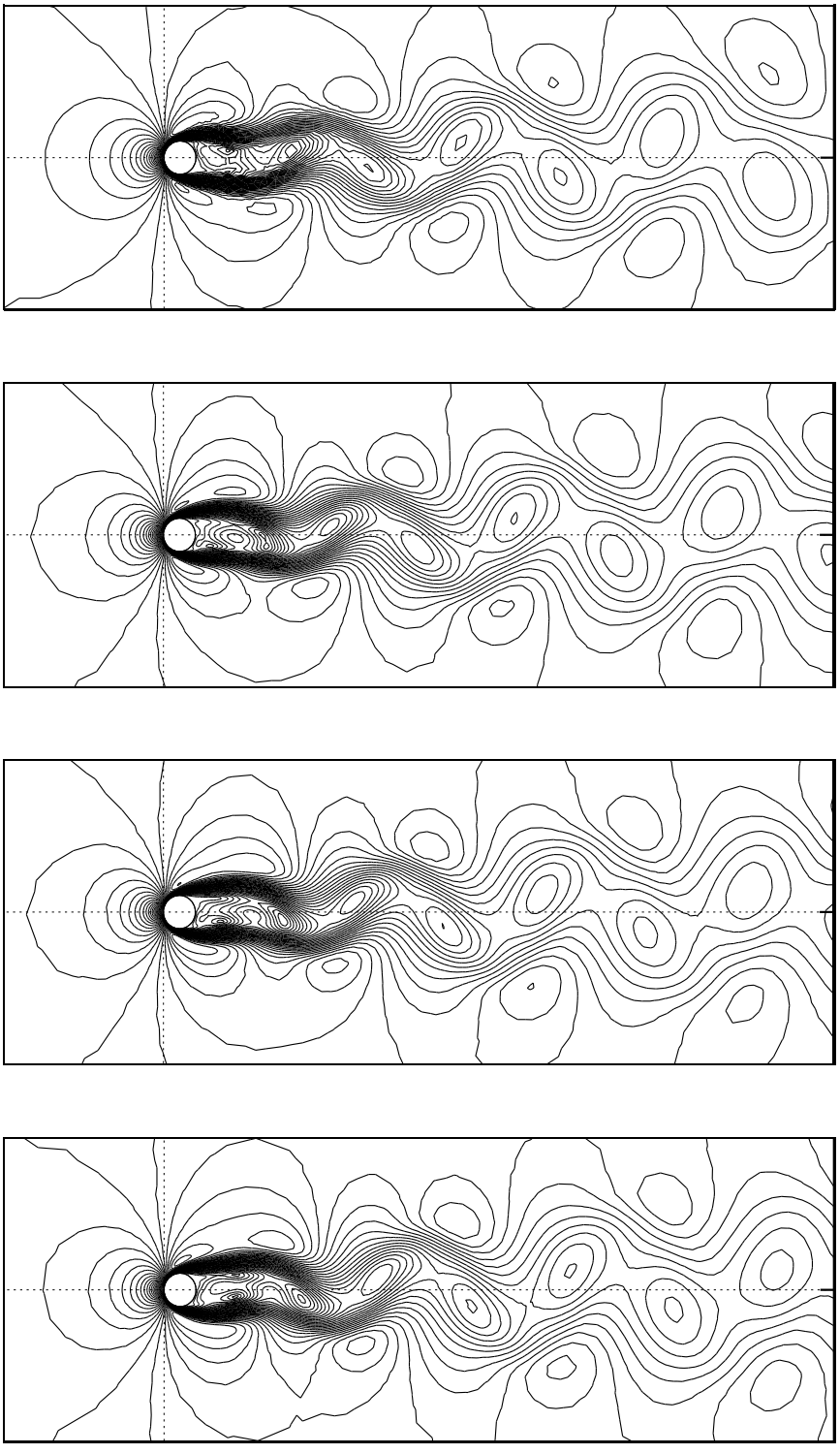


Figure 15: Snapshots of iso-Mach contours for one period.

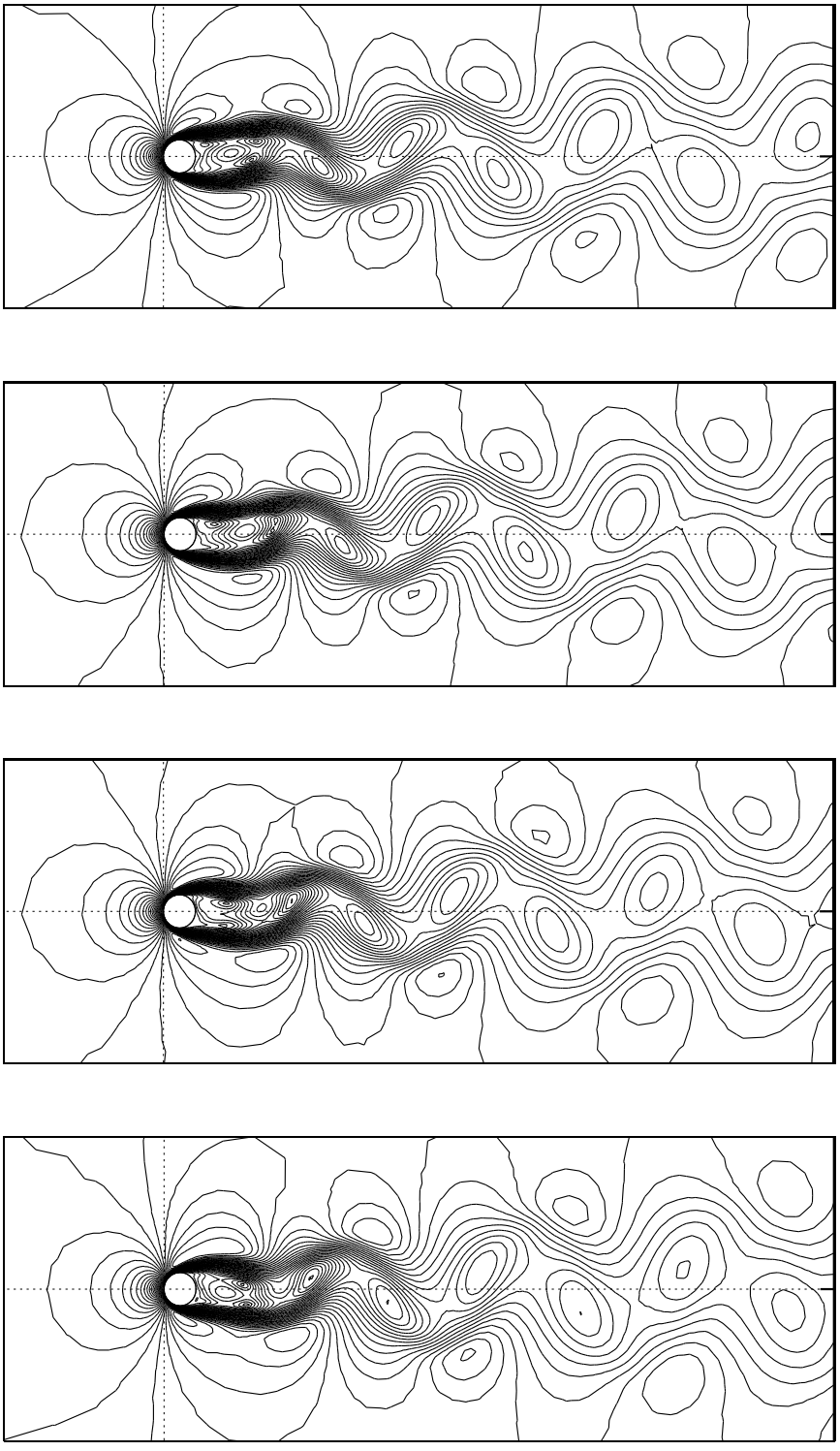
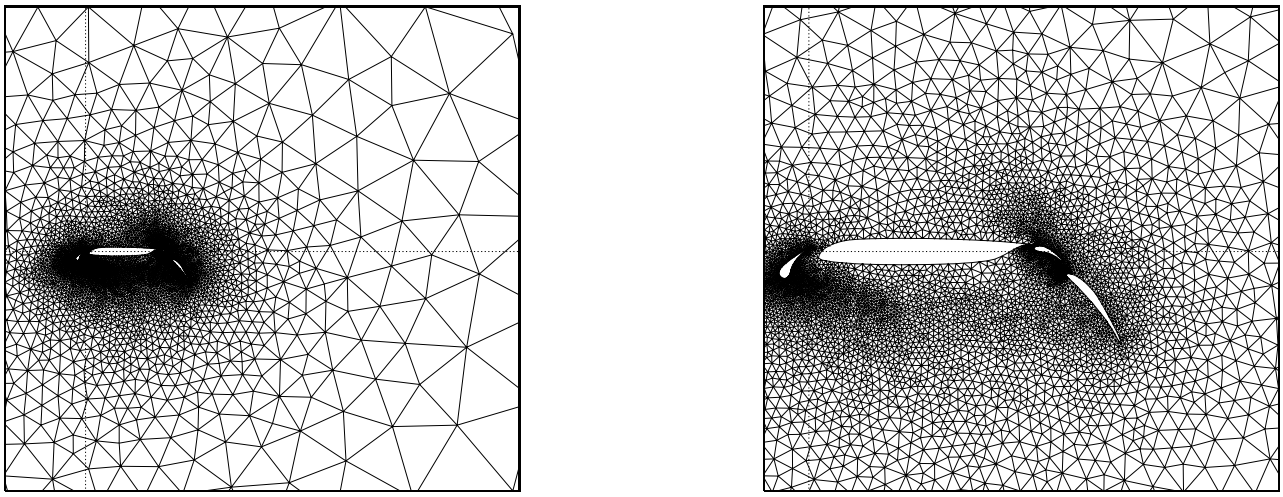
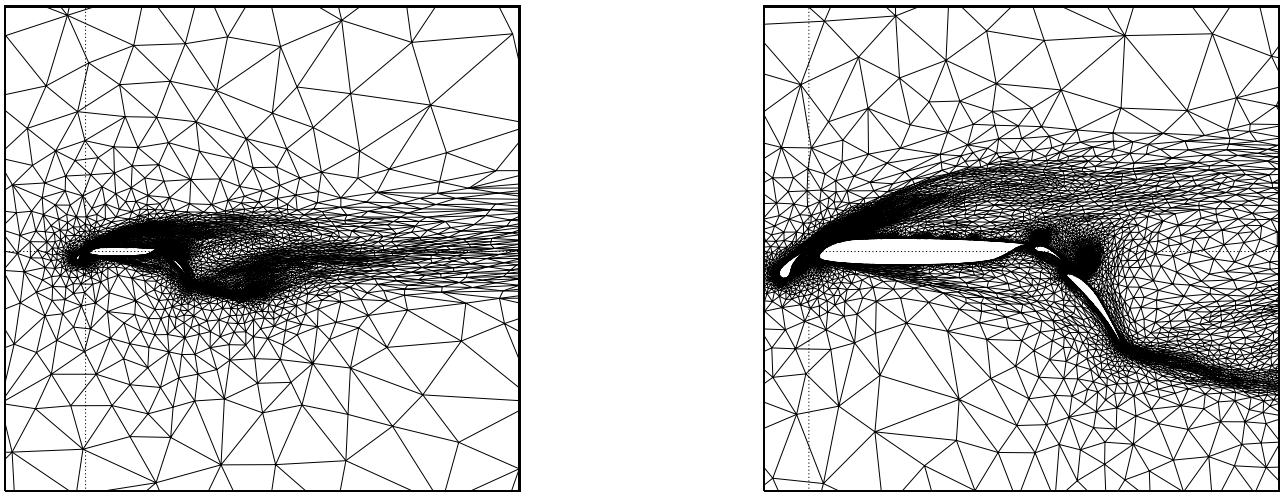


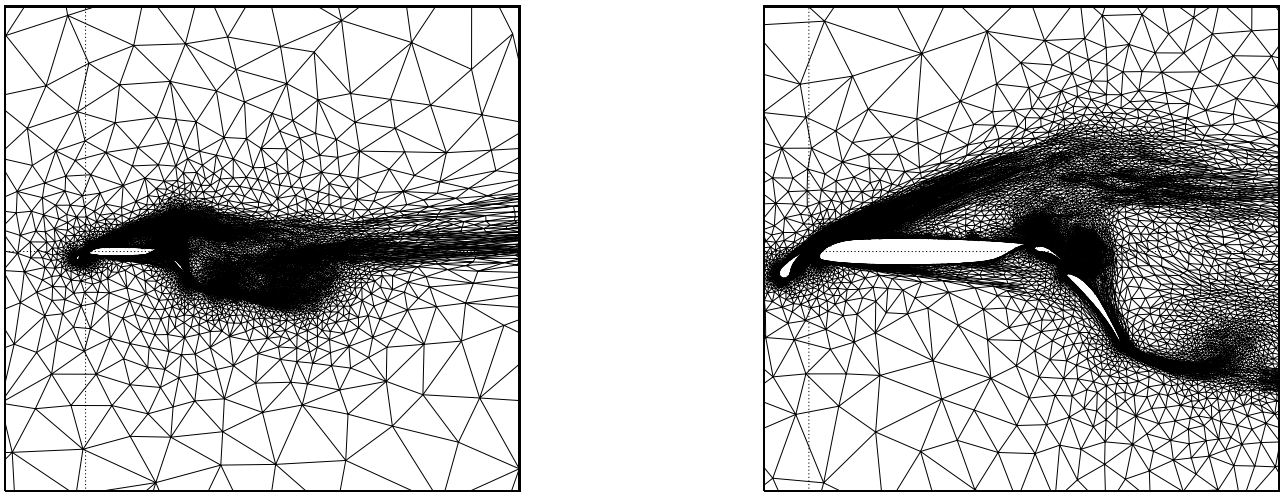
Figure 16: Iso-Mach evolution (continued).



Global and partial vue of the initial mesh (6553 vertices).

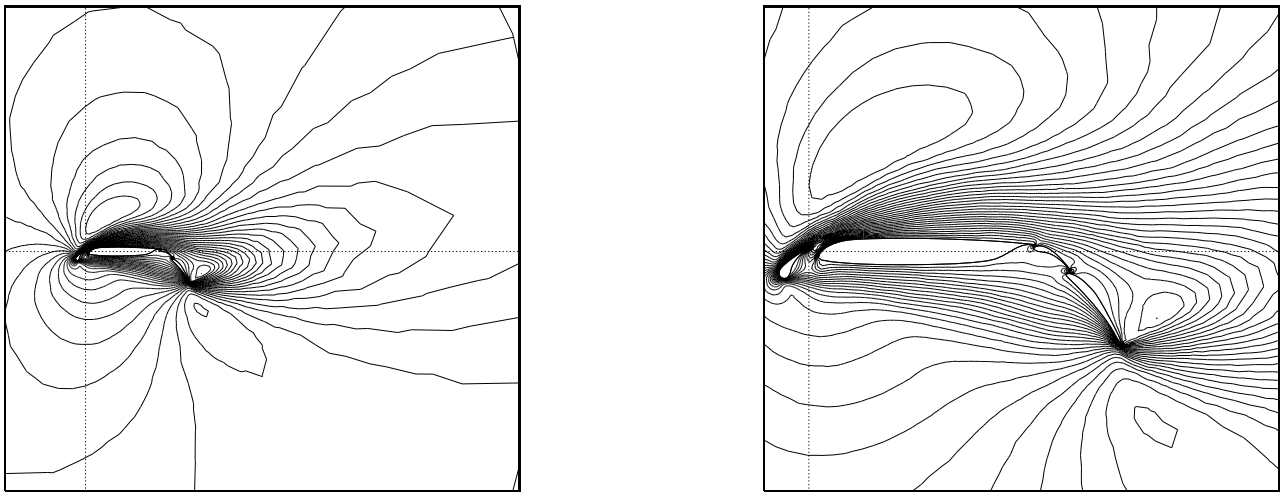


Mesh obtained after 4 adaptations (13860 vertices) (Global and partial vue).

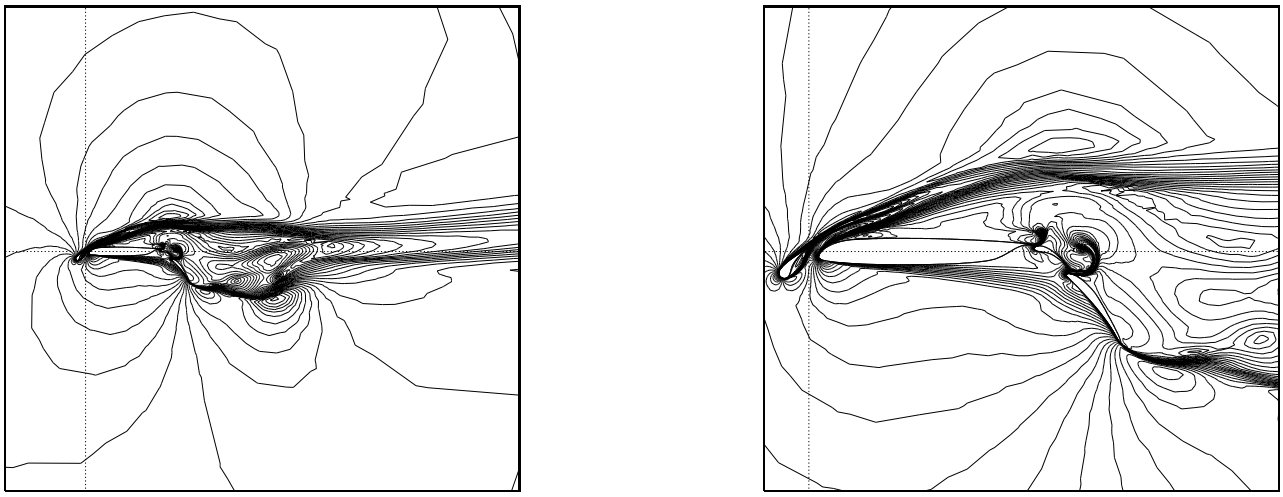


Final adapted mesh (24941 vertices) (Global and partial vue).

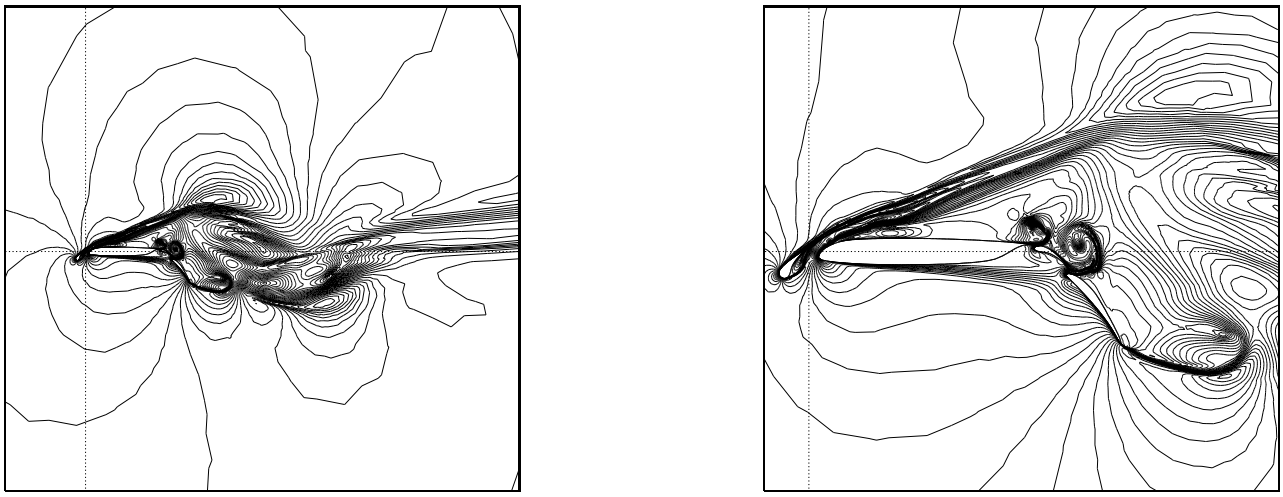
Figure 17: Evolution of the Mesh for a subsonic viscous flow over a 4-element airfoil.



Iso-Mach contours on the Initial mesh (Global and partial vue).



Iso-Mach contours after 4 adaptations (Global and partial vue).



Iso-Mach contours at final mesh (Global and partial vue).

Figure 18: Iso-Mach contours for a subsonic viscous flow over a 4-element airfoil.

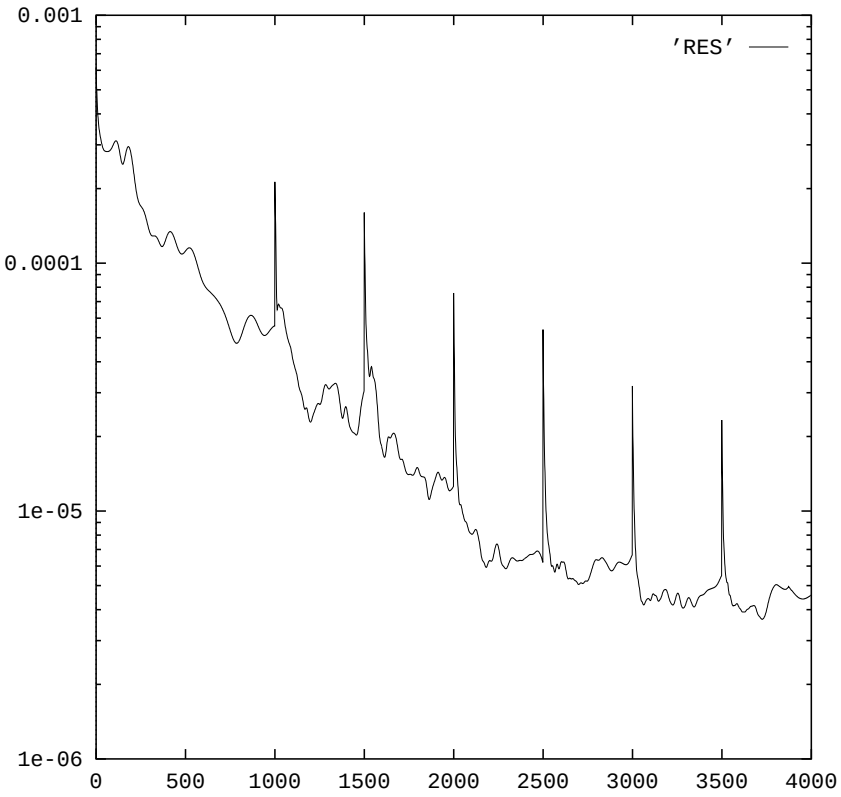


Figure 19: Convergence history.

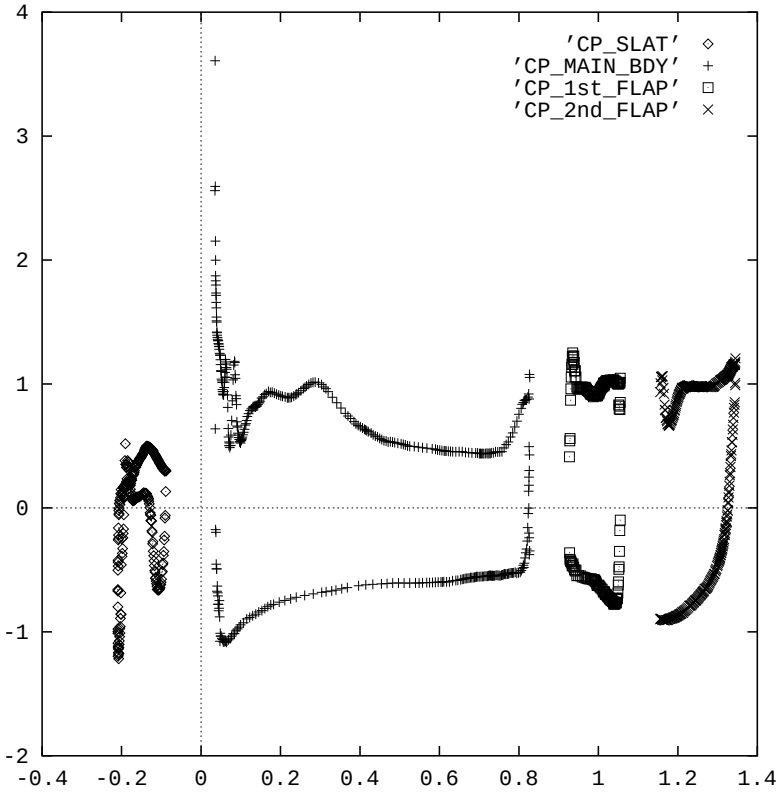


Figure 20: CP along the 4-element airfoil.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399