

---

# A fews snags in mesh adaptation loops

Frederic Hecht<sup>1</sup>

Universit Pierre et Marie, Laboratoire Jacques-Louis Lions, Paris  
frederic.hecht@umpc.fr

## Abstract

The first stage in an adaptive finite element scheme (cf. [CAS95, bor1]) consists in creating an initial mesh of a given domain  $\Omega$ , which is used to perform an initial computation (for example a flow solver). A size specification field is deduced (e.g. at the vicinity of each mesh vertex, the desired mesh size is specified), based on the numerical results. If the mesh does not satisfy the size specification field, then a new constrained mesh, governed by this field, is constructed. The size specification field is usually obtained via an error estimate [FOR, VER96]. Actually, the estimation gives a discrete size specification field. Using an adequate size interpolation over the mesh elements, a continuous field is then obtained.

Metrics are commonly used to normalize the mesh size specification to one in any direction (cf. [VAL92]), and are defined as a symmetric positive definite matrix associated to any point of the domain.

A classical adaptation loop is:

- 0 Build a initial mesh  $\mathcal{T}_h^0$
- 1 loop  $i = 0, \dots$ 
  - Solve your problem on mesh  $\mathcal{T}_h^i$
  - Compute an error indicator , and if the error is small enough then stop.
  - Compute a metric  $\mathcal{M}^{i+1}$  ,
  - Bound, regularize the metric  $\mathcal{M}^{i+1}$  ,
  - Compute a new unit mesh  $\mathcal{T}_h^{i+1}$  with respect to the new metric.

In this kind of algorithm, there are two problematic cases:

- One) if the minimal mesh size is reached then we generally lose the anisotropy of the mesh in this region.
- Two) In the adaptation loop, we use a hidden scheme to evaluate the metric, so some-times the mesh size to compute a good approximation of the solution is incompatible with the scheme to get a good approximation of the metric.

First, we do the numerical experiment to show this two snags. All the experiments are done with **FreeFem++** software, see [freefempp, DAN03].

In this article we present the classical mesh adaptation with metric in section 2. And in section 3 we present the first trouble and some way to solve it. In section 4, a second problem is described and we explain when it occurs.

## 1 Metric and anisotropic mesh adaptation

In this section we recall the notion of mesh *adapted* to a control space or a metric map. Let  $\Omega$  be a domain of  $\mathbb{R}^d$  ( $d = 2$  or  $3$ ) and  $\mathcal{M}_d(\Omega)$  be a *continuous* field of metrics associated with the points of  $\Omega$ . The metric at a point  $P$  of  $\Omega$  characterizes the desired edge- or element-size in the vicinity of  $P$ . By normalizing the desired size to *one*, the metric at  $P$  can be defined as a symmetric positive definite matrix of order  $d$ , denoted as  $\mathcal{M}_d(P)$ ,

Under this metric definition, in the Riemannian space the length  $\mathcal{L}$  of a curve  $\Gamma$  of  $\mathbb{R}^d$ , parametrized by  $\gamma(t)_{t=0..1}$ , is

$$\mathcal{L} = \int_0^1 \sqrt{t\gamma(t)'\mathcal{M}(\gamma(t))\gamma(t)'} dt \quad (1)$$

The metric given is a tool to govern the local mesh size, and so we build a unit mesh in the metric. A unit mesh is such that the size of all the edges in a metric is close to 1.

A way to build a metric with a method of order 2 in space, is to use the Hessian  $\mathcal{H}$  as a good error indicator. 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 [missi, bor2, CAS00, HEC97, GEO99, HAB].

The Key idea is that: the order of the error is given by of the interpolation error, and for a  $P^1$  Lagrange discretization of a variable  $u$ , the interpolation error is bounded by:

$$\mathcal{E} = \|u - \Pi_h u\|_\infty \leq \frac{1}{2} \sup_{T \in \mathcal{T}_h} \sup_{x, y, z \in T} |\mathcal{H}(x)|(y - z).(y - z) \quad (2)$$

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  $\cdot$  is the dot product.

The formular (2) give a way to evaluated the interpolation  $\mathcal{E}_{pq}$  error on an edge  $(p, q)$

$$\mathcal{E}_{pq} = |\mathcal{H}(x)|(q - p).(p - q) \simeq \int_0^1 |\mathcal{H}(tp + (1 - t)q)|(q - p).(p - q) dt$$

Now, we generate, using a Delaunay procedure as a mesh generation method, a mesh with edges close to unit length in the metric

$$\mathcal{M} = \frac{|\mathcal{H}|}{(\mathcal{E})}. \quad (3)$$

The interpolation error  $\mathcal{E}_{pq}$  is equi-distributed with  $\mathcal{E}$  over the edges  $pq$  of the mesh. More precisely, we have

$$(pq)^T \mathcal{M}(pq) = \frac{1}{\mathcal{E}}(pq)^T |\mathcal{H}|(pq) \leq 1. \tag{4}$$

so the interpolation error is less than  $\mathcal{E}$ . Remark, we forgot the coefficient  $\frac{1}{2}$  because the two screw  $\mathcal{E}$  and  $\frac{\mathcal{E}}{2}$  are equivalent.

For systems, the previous approach leads to a metric for each variable. For two metrics  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , we define a metric intersection  $\mathcal{M} = \mathcal{M}_1 \cap \mathcal{M}_2$ , such that the unit ball of  $\mathcal{M}$  is included in the intersection of the two unit balls of metric  $\mathcal{M}_2$  and  $\mathcal{M}_1$  with the procedure defined in [GEO99, chap 10.3.3] for example.

To evaluate the Hessian of  $u_h$  a  $P_1$  finite element function, we can use two formulas like:

$$\mathcal{H}_{h_{ij}}^k = -\frac{1}{3|\Omega_k|} \int_{\Omega_k} \partial_i u_h \partial_j w^k \tag{5}$$

where  $w^k$  is the finite element basis function associated to the vertex  $k$  and  $|\Omega_k|$  is the measure of the support of this basis function.

**Remark**, this approximation generally does not converge when the size of the triangle becomes close to zero.

Or a more stable technique is to make a double projection like:

$$\mathcal{H}_h = I_{L^2} \left( \nabla (I_{L^2}(\nabla u)) \right) \tag{6}$$

where  $I_{L^2}$  is the  $L^2$  projection on the  $P_1$  Lagrange finite element space.

**Remark**, we have no convergence proof, but result is better with using(6, see [alauzet, remark 1.14] for more detail. For the convergence problem, just take the following example approximation of  $f(x, y) = x^2 + y^2$ . If we take a mesh pattern around point  $x_0 = \mathbf{0}$ , and we do a homothety with matrix  $\begin{pmatrix} h & 0 \\ 0 & h \end{pmatrix}$  on this pattern. The value of approximate Hessian at mesh point  $x_0$  given by the two method on the transform pattern are independent of  $h$ , because  $f$  is quadratic.

### 1.1 Mesh generation and adaptation

We use the mesh generation tools developed at INRIA-Gamma project [bamg, GEO99]. One novelty 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 mesh prescribed by the metric is similar to the previous mesh. This is of course suitable 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 ([GEO99, Sec. 7.3.1]) where the insert point procedure is fully described in [GEO99, Sec. 7.6.1].

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

1. Create a bounding box mesh.
2. Discretize the boundary with respect to the metric, Let  $\mathcal{P}$  be 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 the points of  $\mathcal{P}$  one by one.
5. Enforce the boundary mesh.
6. Remove outside triangles.
7. DO: far field point creation  $\mathcal{S} = \emptyset$ .
  - Split all the edges which are too long 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.

Finally, the adaptation loop scheme is:

- 0) Build a initial mesh  $\mathcal{T}_h^0$
- 1) loop  $i = 0, \dots$ 
  1. Solve your problem on mesh  $\mathcal{T}_h^i$
  2. Compute an error indicator , and if the error is small enough then stop.
  3. Compute a metric  $\mathcal{M}$  ,
  4. Bound, regularize the metric  $\mathcal{M}$  ,
  5. Compute a new mesh unit mesh  $\mathcal{T}_h^{i+1}$  with respect to the new metric.

Remark, all these tools are available in the **FreeFem++** software [freefempp].

## 2 Lose of Anisotropy

In case of a solution with shock layer, the minimal mesh size goes to zero perpendicularly to the shock during the adaptation loop, so the user wants a method to bound computations in size or in time. To do this, he gives a minimal mesh size cutoff value to limit the number of mesh points.

We see the problem immediately, if the adaptation loop tries to build an anisotropic mesh with smaller a mesh size than the minimal mesh size cutoff value.

First, to focus on this problem, we just try to build an optimal mesh with respect to the function

$$f(x, y) = yx^2 + y^3 + \tanh(10(\sin(5y) - 2x)).$$

This function has a smooth shock line. the equation of the line is  $\sin(5y) = 2x$ . Figure 3 shows a three dimensional representation of this function  $f$ .

The adaptation algorithm in **FreeFem++** [freefempp] language is:

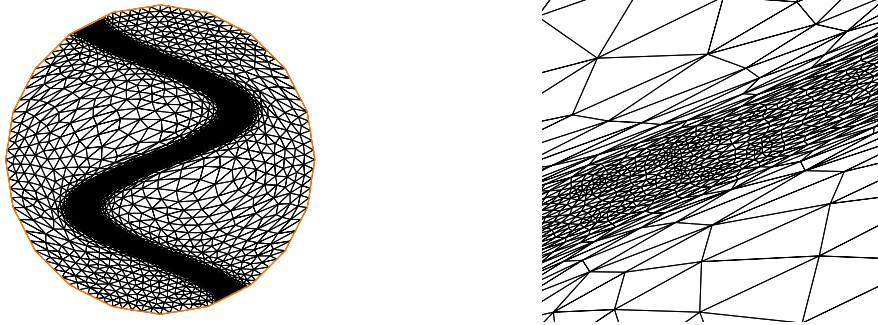
```
// interpolation error: eps = E, see formula (2)
real eps = 0.002;
the user parameter to set the miminal mesh size hmin
real hmin=0.005;// see figure 1
// or hmin=0.000005 see figure 2
func f = y*x*x+y*y*y+h*tanh(10*(sin(5.0*y)-2.0*x));
```

```

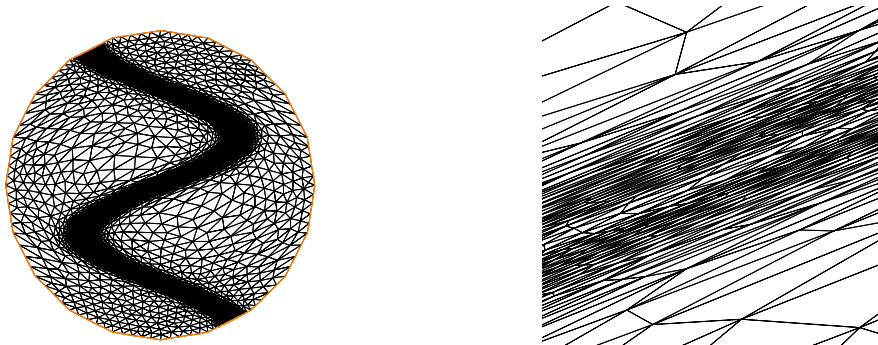
border cercle(t=0,2*pi) x=cos(t);y=sin(t);
mesh Th=buildmesh(cercle(20));

for (int i=0;i<6;i++)
{
Th=adaptmesh(Th,f,hmin=hmin,err=eps,nbvx=100000);
plot(Th,ps="la-Th-"+hmin+".eps");
}

```



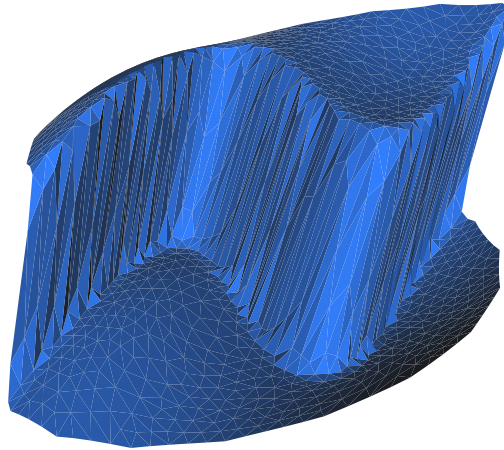
**Fig. 1.** Adaptated mesh with  $hmin = 0.005$ , the left picture seems good but on the picture on the right , we see the lose of anisotropy (Number of Triangles = 11456, Number of Vertices 5806)



**Fig. 2.** Adaptated mesh with  $hmin = 0.00005$ , everything is correct (Nb of Triangles = 14054, Nb of Vertices 7110)

The natural question is: "*Why does this problem of losing anisotropy occur?*"

The answer is given in the three dimensional plot of the  $f$  function on a coarse grid with shading. As you can see thanks to the shading in the shock region it is dented, irregular because the surface is badly approximated. In a tangent direction to the shock layer, the second derivative is high and thus the method puts lots of points in this direction, and the mesh become isotropic.



**Fig. 3.** the 3d representation of function  $f$  on a grid which is too coarse, with  $hmin = 0.05$

Now, how to solve this problem?

- Change the mesh generator or do some optimization, to force points to be on extremal curvature lines to get super convergence propriety (see paper with J.-F. Lage in preparation).
- Change the error indicator, in a shock region, see the work of F. Alauzet [alauzet].
- Smooth the solution before the computation of the metric with for example a convolution with a regularizing real function  $\Phi_h$  of  $\mathbb{R}^d$  to be sure that the built metric respects the constraint on  $hmin$ .

Remember that, the convolution operator  $\star$  of two functions is defined by

$$(f \star g)(x) = \int f(x - y)g(y)dy.$$

The regularizing function must clearly satisfy:

$$\int_{\mathbb{R}^d} \Phi_h(x) dx = 1, \quad \text{and } \|(\partial_{ij}(\Phi_h))_{ij \in (1, \dots, d)^2}\| \leq \frac{1}{h^2} \tag{7}$$

where here  $h$  is the minimal mesh size cutoff value  $hmin$ .

The function can be radial and can be defined with  $\Phi_h(x) = \phi_h(\|x\|)$ . The real function  $\phi_h$  has a small support, must verify

$$\int_0^\infty r^{d-1} \phi_h(r) dr = \frac{1}{S_d}, \quad |\phi_h''| \leq 1/h.$$

where  $S_d$  is the measure of the sphere in dimension  $d$  ( $S_2 = 2\pi, S_3 = 4\pi$ ).

Let us introduce the function  $\rho$  defined by

$$\rho(x) = \begin{cases} 1 - \frac{1}{2}x^2 & \text{if } x \in [0, 1], \\ \frac{1}{2}(x - 2)^2 & \text{if } x \in [1, 2], \\ 0 & \text{otherwise } x \in [2, \infty]. \end{cases}$$

and denote  $\alpha_d = \int_0^\infty x^{d-1} \rho(x) dx$ , we have  $\alpha_2 = 7/12$  and  $\alpha_3 = 1/2$  and we can remark that  $|\rho''| = 1$  on  $]0, 2[$ .

We define the function  $\phi_h$  by

$$\phi_h(x) = \frac{1}{\tilde{h}^d \alpha_d S_d} \rho\left(\frac{x}{\tilde{h}}\right).$$

On the support  $[0, 2\tilde{h}[$  of  $\phi_h(x)$  we have

$$|\phi_h''| = \frac{1}{\tilde{h}^{d+2} \alpha_d S_d} = \frac{1}{h^2} \quad \text{thus } \tilde{h} = \left(\frac{h^2}{\alpha_d S_d}\right)^{\frac{1}{d+2}}$$

So finally the function is

$$\Phi_h(x) = (\alpha_d S_d)^{-\frac{2}{d+2}} h^{-\frac{2d}{d+2}} \rho\left(\left(\frac{h^2}{\alpha_d S_d}\right)^{-\frac{1}{d+2}} \|x\|\right)$$

The same example with this technique solves the problem, as you can see on figure 4.

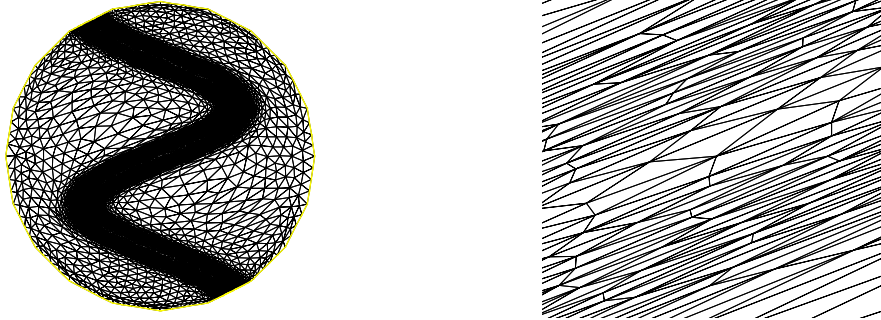
Remark, we preserve the anisotropy but we lose a little bit in the thickness of the shock layer. This is due to the size of the support of function  $\Phi_h$ .

### 3 Problem of convergence in adaptation

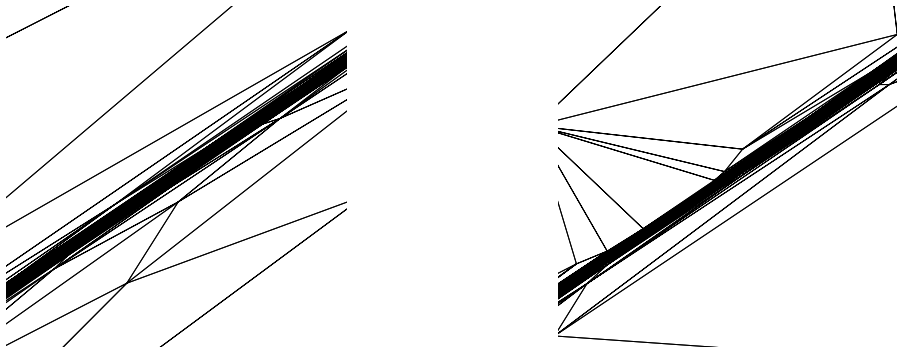
As I say in the introduction, there is a hidden scheme to compute the metric. To compute the metric we use formula (3). Here, to evaluate the Hessian of a discrete function  $u_h$ , we use (6). This hidden scheme has some criterion of convergence, this criterion is hard to define exactly. But anyway, it is possible to show some convergence problems.

Let take the same adaptation loop with a more harder function  $f$ .

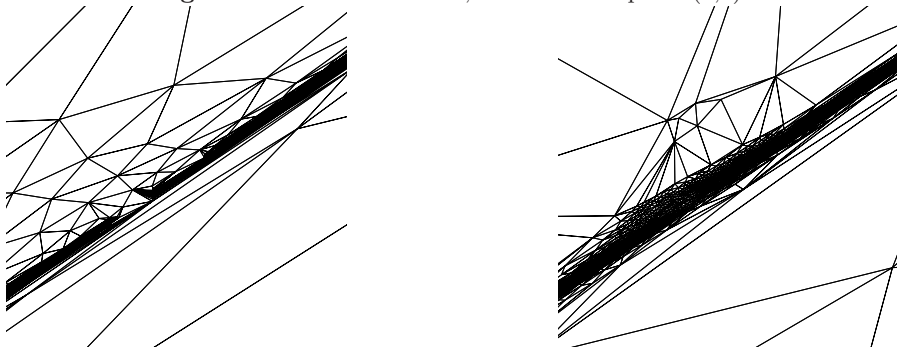
$$f = 0.1 * (y * x * x + y * y * y) + h * \tanh(200 * (\sin(3y) - 2x));$$



**Fig. 4.** Adapted mesh with  $h_{min} = 0.005$ , with the convolution technique, Nb of Triangles = 9659, Nb of Vertices 4905



**Fig. 5.** iteration 12 and 13 , zoom around point (0,0)



**Fig. 6.** iteration 14 and 16, zoom around point (0,0)



The shock on this function is straighter, due to the coefficient set to 200.

```

real eps = 0.01;
real hmin=0.000005; //
int nbiter=40;
func f = 0.1*(y*x*x+y*y*y)+tanh(200*(sin(3*y)-2*x));

```

// same loop and initial mesh as in the previous // freefem++ example at page 2.

We see that at iteration 12 the mesh is correct, but in the next three steps the meshes are bad because the metric is clearly wrong. The reason is: We do not have enough points to compute the second derivative to evaluate the metric of the function  $f$  correctly, from the interpolation of  $f$  on the current mesh.

A classical analysis of a scheme to compute a second order derivative, shows that the error depends of the third derivative. In the example the third derivative is huge compare to the second.

First, This problem happens rarely, if we change the norm to evaluate the error, the  $L^\infty$  norm is changed to a more geometrical error (the Hausdorff distance). In this case, the metric associated to function  $u$  is defined by

$$\mathcal{M} = \frac{|\mathcal{H}|}{(\sqrt{1 + \|\nabla u\|^2} \mathcal{E})}, \quad (8)$$

where  $\mathcal{H}$  is the Hessian of  $u$ .

With this metric, computational results are often better (see figure 7 and 8), but the problem can still occur, and I have no another solution that completely solves this problem.

#### Conclusion:

This kind of adaptation scheme gives very good results but sometimes, as you have seen they are problems. Generally, we can explain why the problem occurs, by analysing an example. Sometimes it is more difficult to correct, but anyway mesh adaptation can help you to solve a lot of problems.

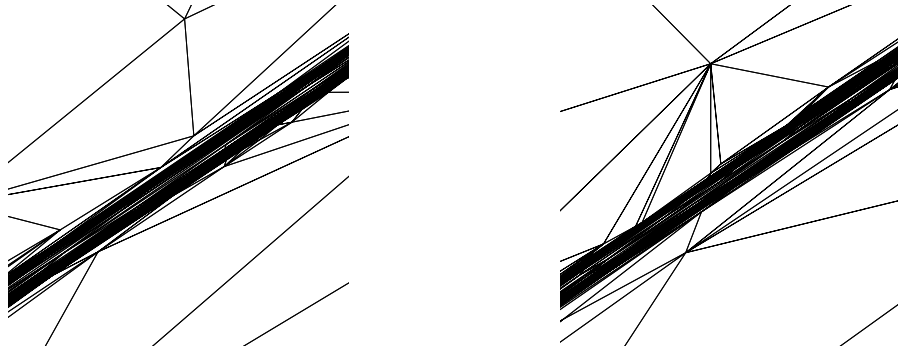
**Acknowledgments:** Thank to the helpful discussion and comment with Frédéric Alauzet, Pascal Frey, Jean-françois Lagüe and Antoine Le Hyaric,

[alauzet] F. ALAUZET, Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en manique des fluides, *The de doctorat de l'UniversitMontpellier II*, 2003.

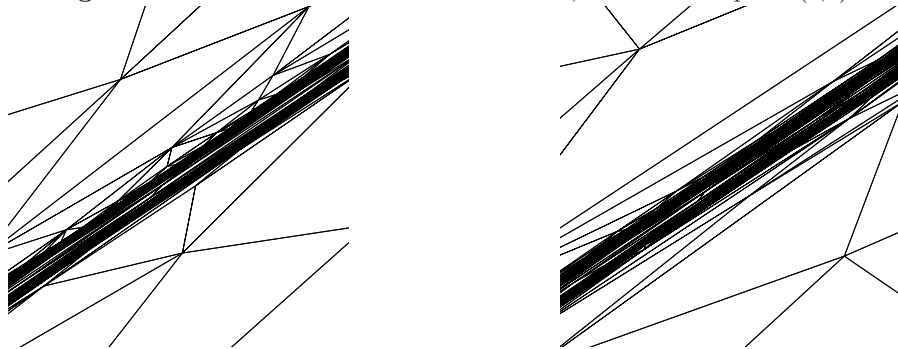
[missi] H. BOROUCAKI H., M.J. CASTRO-DIAZ, P.L. GEORGE, F. HECHT AND B. MOHAMMADI, Anisotropic adaptive mesh generation in two dimensions for CFD *5th Inter. Conf. on Numerical Grid Generation in Computational Field Simulations*, Mississippi State Univ., 1996.

[bor1] H. BOROUCAKI, P.L.GEORGE, F.HECHT, P.LAUG AND E.SALTEL, Delaunay Mesh Generation Governed by Metric Specifications. Part I: Algorithms., *Finite Elements in Analysis and Design*, 25, pp. 61–83, 1997.

[bor2] H.BOROUCAKI, P.L.GEORGE AND B.MOHAMMADI, Delaunay Mesh Generation Governed by Metric Specifications. Part II: Applications., *Finite Elements in Analysis and Design*, 25, pp. 85–109, 1997.



**Fig. 7.** Geometrical Metric iteration 12 and 13 , zoom around point (0,0)



**Fig. 8.** iteration 14 and 16, zoom around point (0,0)

- [CAS95] M.J.CASTRO-DÍAZ, F.HECHT, AND B.MOHAMMADI, New progress in anisotropic grid adaptation for inviscid and viscid flows simulation, *4th International Mesh Roundtable*, Albuquerque, New-Mexico, october 1995.
- [CAS00] M.J. CASTRO-DIAZ , F. HECHT F. AND B. MOHAMMADI Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations *IJNMF*, Vol. 25, 475-491, 2000.
- [DAN03] I. Danaila, F. Hecht, and O. Pironneau. *Simulation numérique en C++*. Dunod, Paris, 2003.
- [GEO99] FREY P.J. ET GEORGE P.L., *Maillages*, Hermès, Paris, 1999.
- [FOR] M. FORTIN, M.G. VALLET, J. DOMPIERRE, Y. BOURGAULT AND W.G. HABASHI Anisotropic Mesh Adaption: Theory, Validation and Applications, *Eccomas 96, PARIS*, CFD book, pp 174–199.
- [HAB] W. HABASHI ET AL. A Step toward mesh-independent and User-Independent CFD *Barriers and Challenges in CFD*, pp. 99-117, Kluwer Ac. pub.
- [HEC97] F. HECHT AND B. MOHAMMADI Mesh Adaptation by Metric Control for Multi-scale Phenomena and Turbulence *AIAA*, paper 97-0859, 1997.

- [bamg] F. HECHT The mesh adapting software: bamg. INRIA report 1998.  
<http://www-rocq.inria.fr/gamma/cdrom/www/bamg/eng.htm>.
- [freefempp] F. Hecht, K. Ohtsuka, and O. Pironneau. *FreeFem++ manual*. Universite Pierre et Marie Curie, 2002–2005. on the web at  
<http://www.freefem.org/ff++/index.htm>.
- [VAL92] M.G. VALLET, Génération de maillages Éléments Finis anisotropes et adaptatifs, *thèse Université Paris VI*, Paris, 1992.
- [VER96] R. VERFÜRTH, A review of a posteriori error estimation and adaptive refinement techniques, *Wiley Teubner*, 1996.

