Metric Generation for a Given Error Estimation

Frédéric Hecht[1] and Raphaël Kuate[2]

[1] Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie
hecht@ann.jussieu.fr

[2] Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie
kuate@ann.jussieu.fr

Abstract. The mesh adaptation is a classical method for accelerating and improving the PDE finite element computation. Two tools are generally used: the metric [10] to define the mesh size and the error indicator to know if the solution is accurate enough.

A lot of algorithms used to generate adapted meshes suitable for a PDE numerical solution for instance [15, 1] for discrete metrics, or [15, 2] for the continuous one, use those tools for the local specification of the mesh size.

Lot of PDE softwares like *Freefem++* [14] use metrics to build meshes, the edges sizes of which are equal with respect to the metric field. The construction of metrics from the hessian matrix [16, 18, 13] is only justify for the piecewise linear Lagrange finite element.

So there is the problem of metric generation when we have another interpolation error estimator [6, 11] that could be used for instance when the Lagrange interpolation needed is a k degree polynomial, $k > 1$.

To answer that question, we propose in this paper an algorithm whose complexity is quasi-linear, in two spacial dimensions; assuming that the error is locally described by a closed curve representing the error level set. Some efficient numerical examples are given. This algorithm allows us to obtain the analytical metric [1], when the error indicator is based on the hessian matrix.

We have also done one comparison in the software *Freefem++* of mesh adaptation with metrics computed using this algorithm with respect to the interpolation error estimation described in [11], and the method with metrics based on the hessian. The results seem to be better for the maximal error.

1 Introduction

We suppose once a metric field on a mesh is known, we can built an adapted mesh for that metric field [8, chap 21.4], [12, chap 20.3]. If we know the metric value on each vertex of the mesh for example, the method described in [12] modify the mesh by evaluating each edge length in the metric field (control space) using an interpolation of the metrics defined on the two vertices of the edge. When an edge length exceeds the "unity", a subdivision of that edge is done in the control space. This method is used in the software *Freefem++* with the mesh generator Bamg [9]

Thanks to Cea lemma, the approximation error is upper bounded by the interpolation error. So our problem is posed for the interpolation error, we'll use the recent results described in [11] which aim to bound the interpolation error of a function on a simplex with the norm of a vector inside a metric given by the $D^{(l)}$ derivative of the function.

We suppose that we know an estimation of the local interpolation error of a function on a 2D mesh around a vertex P in each direction ξ given by a function, and we want to dispose the points M of the triangulation around P such that the local interpolation error should be satisfy.

Let $E_p(\xi)$ be an estimation of the interpolation error in the direction of the vector ξ. If the mesh size is less than or equal to $\|\xi\|$ in the direction of ξ then the interpolation error will be less than $E_p(\xi)$. We want to have locally

$$|E_p(\xi)| \leq \mathcal{E}; \xi \in \mathbb{R}^2. \tag{1}$$

The previous inequality represents an area bounded by the level set

$$|E_p(\xi)| = \mathcal{E}.$$

Dispose the points M of the triangulation around the vertex P as close as possible to the level set while being inside the area delimited by that level set is equivalent to say that the interpolation error evaluated at each vertex of the triangulation going from the vertex P is less than or equal to its value on the level set. If we want to have a mesh whose edges are equal with respect to a metric field, i.e all the points M are equi-distant from P in a local metric $\mathcal{M}$, so they belong to the unit ball of the metric $\mathcal{M}$; the biggest as possible that could be contained inside the closed line described by equation (1).

Let us find a metric prescribing the mesh size around that vertex P such as the triangles vertices should be close as possible to the level set while being inside the area delimited by that level set. How to build that metric?

In the following, after introducing shortly the results of [11] and the metric notion, [15, 10, 1], we formulate the optimization problem that goes with and it resolution algorithm. Then we'll explain the algebraic consequences of the obtained equations in the third section. The fourth section talk about our algorithm convergence and complexity and some illustrating figures. The last section before concluding explains shortly the analytic metric obtainability from the hessian matrix, that our method allows to approach. We also give in that section an example of mesh adaptation for a function with metrics generated by our algorithm with respect to the local interpolation error of [11], and we compare the results obtained with mesh adaptation using metrics based on the hessian matrix, both of them done with the software *Freefem++*.

2 Definitions

We use the notations [3, Chap. VII], which are introduced in the books [7] [5, chap II].

- $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, Sphere $S_d = \{x \in \mathbb{R}^d; ||x|| = 1\}$,
- $D^{(\ell)}u$ is the the derivative of u of order ℓ so if u is a real function, we have $D^{(0)}u = u$.

- $B^{\otimes \ell} : (\xi_1, ..., \xi_\ell) \mapsto (B\xi_1, ..., B\xi_\ell)$
- $W^{\ell,p}(K)$ is the set of functions of $L^p(K)$ such that all the partial derivatives of order $\leq \ell$ are in $L^p(K)$.
- $|u|_{W^{\ell,p}(K)} = (\int_K \sum_{\alpha \in [1..d]^\ell} |\partial_\alpha u|^p)^{\frac{1}{p}}$
- Let F be an affine application of $\mathbb{R}^d$ with values in $\mathbb{R}^d$,

$$||D^{(k)}(v \circ F)||_{L^p(K)} \leq ||D^{(1)}F||^k ||D^{(k)}v||_{L^p(F(K))}$$

d is the space dimension , K is a d-simplex (triangle if $d = 2$, or tetrahedron if $d = 3$), h_K the diameter of K and ρ_K the diameter of its inscribed ball, $|K|$ the measure of K and $|\partial K|$ the measure the border of K.

Definition 1 (d-simplex finite element of degree k)
A finite element (K, P_K, Σ_K) will be called d-simplex of degree k, if K is a d-simplex and if $P_k(K) \subset P_K$ where $P_k(K)$ is the set of polynomials $P[\mathbb{R}^d]$ restricted to K of degree less than or equal to k. Let $\mathcal{I}_K^k$ be the interpolation operator with values in P_K and which keep unchanged the polynomials of $P_k(K)$, i.e if $u \in P_k(K)$, then $u = \mathcal{I}_K^k u$.

Definition 2 (reference element)
The reference element is a regular d-simplex denoted $\hat{K}$, chosen with the size 1.

Let F_K be the affine transformation such that $K = F_K(\hat{K})$, we denote: B_K the linear part of F_K which corresponds to the derivate of F_K; and $\det(B_K)$ the determinant of B_K which is the Jacobian of F_K.

The size $h_{K,\xi}$ of an element K in the direction ξ, the size of the edges in the direction ξ of that edge. $||B_K^{-1}||^{-1}$ is the smallest size of the element denoted $h_K^- = \inf_{\xi \in S_d} h_{K,\xi}$. This element size can be represented by an ellipse associated to the "natural" metric of the element introduced in [17, 4].

Let u be a real function of K, we define its corresponding function
$\hat{u} = u \circ F_K$ on $\hat{K}$. We have

$$x = F_K(\hat{x}), \qquad \hat{u}(\hat{x}) = u(x), \qquad u = \hat{u} \circ F_K^{-1}.$$

A finite element K is affine-equivalent if it has the following property

$$\widehat{\mathcal{I}_K^k u} = \mathcal{I}_{\hat{K}}^k \hat{u}.$$

Main Result
For any finite element (K, P_K, Σ_K) of definition 1 satisfying the previous property, with an associated metric $\mathcal{M}_K$; let u be a function. For all integer $\ell \in [0..k+1]$ and for all $(p, q) \in [1, \infty]^2$ such that $W^{\ell,p}(K)$ be included in $\mathcal{C}^0(K)$ and $u \in W^{\ell,p}(K)$. If it exists a real constant $\kappa \in \mathbb{R}$ such that

$$||(D^{(\ell)}u)(\xi, ..., \xi)||_{L^p(K)} \leq \kappa \, ||\xi||_{\mathcal{M}_K}^\ell (h_K^-)^\ell = \kappa \, (\xi, \mathcal{M}_K \xi)^{\frac{\ell}{2}} (h_K^-)^\ell, \tag{2}$$

then we have

$$||u - \mathcal{I}_K^k u||_{L^q(K)} \leq \kappa C \, |K|^{\frac{1}{q} - \frac{1}{p}} (h_K^-)^\ell,$$

where C is a real positive constant depending only on d, ℓ, p, q.

And if $W^{\ell,p}(K) \subset W^{1,q}(K)$ *then we have*

$$|u - \mathcal{I}_K^k u|_{W^{1,q}(K)} \leq \kappa C\, |K|^{\frac{1}{q}-\frac{1}{p}} (h_K^-)^{\ell-1},$$

with C *positive constant depending only on* d, ℓ, p, q*.*

Definition 3. *A constant metric* $\mathcal{M}$ *of* $\mathbb{R}^2$ *is a symmetric matrix positively definite that allows to define a distance in the euclidian metric space* $\mathbb{R}^2$*.*

In fact the scalar product of two vectors $\overrightarrow{X}$ and $\overrightarrow{Y}$ under the metric space $\mathbb{R}^2$ is defined as: [8]

$$\langle \overrightarrow{X}, \overrightarrow{Y} \rangle_{\mathcal{M}} = {}^t\overrightarrow{X} \mathcal{M} \overrightarrow{Y} \in \mathbb{R}.$$

The euclidian norm of a vector $\overrightarrow{AB}$ or the distance between the points A and B for the metric $\mathcal{M}$ is defined by:

$$d_{\mathcal{M}}(A,B) = \|\overrightarrow{AB}\|_{\mathcal{M}} = \sqrt{{}^t\overrightarrow{AB}\mathcal{M}\overrightarrow{AB}}.$$

There is the situation for which the metric is non constant but continuous. In this case we have a *metric field* $\mathcal{M}(X)$ defined in each point X of the space. In this case, the norm of a vector $\overrightarrow{AB}$ is given by

$$d_{\mathcal{M}}(A,B) = \|\overrightarrow{AB}\|_{\mathcal{M}} = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}(A + t\overrightarrow{AB})\overrightarrow{AB}}dt.$$

A metric $\mathcal{M}$ can be represented by it unit ball [1], The unit ball under the metric $\mathcal{M}$ defined at the vertex P is the set of points M that satisfies:

$$\|\overrightarrow{PM}\|_{\mathcal{M}} = \sqrt{{}^t\overrightarrow{PM}\mathcal{M}\overrightarrow{PM}} = 1.$$

Let us consider $\mathcal{M} = \begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}$ $\quad a > 0, b > 0,\ 4ab - c^2 > 0;\ P(x_p, y_p);\ M(x,y).$

$$\|\overrightarrow{PM}\|^2_{\mathcal{M}} = 1 \qquad \Longleftrightarrow \qquad a(x-x_0)^2 + c(x-x0)(y-y_0) + b(y-y_0)^2 = 1.$$

By writing

$$a = \frac{\cos\theta^2}{\alpha^2} + \frac{\sin\theta^2}{\beta^2};\quad b = \frac{\cos\theta^2}{\beta^2} + \frac{\sin\theta^2}{\alpha^2};\quad c = \sin 2\theta \left(\frac{1}{\alpha^2} - \frac{1}{\beta^2}\right); \tag{3}$$

which are bounded by:

$$\frac{1}{\alpha^2} \leq\ a\ \leq \frac{1}{\beta^2};\quad \frac{1}{\alpha^2} \leq\ b\ \leq \frac{1}{\beta^2};\quad \frac{1}{\alpha^2} - \frac{1}{\beta^2} \leq\ c\ \leq \frac{1}{\beta^2} - \frac{1}{\alpha^2}; \tag{4}$$

we obtain:

$$\frac{((x-x_0)\cos\theta + (y-y_0)\sin\theta)^2}{\alpha^2} + \frac{((y-y_0)\cos\theta - (x-x0)\sin\theta)^2}{\beta^2} = 1,$$

which is an ellipse equation centered at P with semi-axes of length $\alpha > 0$ and $\beta > 0$; $\alpha \geq \beta$; where θ is the angle between the major axis and the abscises axis of the $\mathbb{R}^2$ canonic base.

3 The Problem Description

We do not take care of the different norms used and the different constants of equation (2). We consider that

$$|E_p(\xi)| = ||(D^{(\ell)}u)(\xi, ..., \xi)||_{(K)}. \tag{5}$$

We want to find a local metric $\mathcal{M}$ such that the right member of (2) should be closed to $\mathcal{E}$ in (1).

Let $F(x, y)$ be $E_p(\xi)$. Our problem becomes drawing an ellipse that have the maximum area centered at P contained inside that given error curve or finding the parameters α , β and θ of that ellipse.

Let us consider $\alpha_1 = \dfrac{1}{\alpha^2}$ and $\alpha_2 = \dfrac{1}{\beta^2}$, we obtain from (3)

$$\alpha_2 = \frac{\sqrt{(a-b)^2 + c^2} + (a+b)}{2}; \quad \alpha_1 = \frac{-\sqrt{(a-b)^2 + c^2} + (a+b)}{2}.$$

Maximizing the ellipse area, the parameters of which are α; β; θ is the same as maximizing $\alpha\beta$ or minimizing $\alpha_1\alpha_2$ which is equivalent to minimize $(4ab - c^2)$.

We look at the problem in a numerical point of view and suppose that the vertex P is at the origin.

Let us consider a discretization of our curve with n points M_i of position (x_i, y_i). We have to find three reals a, b and c that minimize $(4ab - c^2)$ under the constraints

$$\begin{cases} a > 0, b > 0, \\ ax_i^2 + by_i^2 + cx_iy_i \geq 1, \ 0 \leq i \leq n-1, \\ 4ab > c^2. \end{cases}$$

Without "good" properties of the minimization function (convexity, quasi-convexity), we didn't find any algorithm that guarantees a global minimum; although it exists because the linear constraints form a close set. The resolution of this problem in this form need the construction of the constraints polyhedron where we'll find the minimum. As $(a, b, c) \in \mathbb{R}^3$ the resolution algorithm will be $\mathcal{O}(n^3)$ complexity in the best case; but it is very cheap inside an a mesh adaptation process. To reduce that cost we propose to solve an approximated problem whose algorithm will be $\mathcal{O}(n^k)$; $k \leq 2$ complexity.

Let us consider two real non negative numbers ϵ_0 and ϵ close to zero, let us call M_0 the point of the error curve the most closed to the origin P of our ellipse, and M_i another point of that error curve. Then we find an ellipse that have the maximal area. Let ϵ_0 be the distance between that ellipse and M_0, and let ϵ be the distance between that ellipse and M_i (figure 1). The distance between the ellipse and a point M_j of the error curve will be consider as the distance between M_j and the intersection point of the vector $\overrightarrow{M_jP}$ and the ellipse. Let us consider a point X of the ellipse on the segment $[P, M_i]$ such that $\|\overrightarrow{M_iX}\| = \epsilon$, the value of X is $(1 - \frac{\epsilon}{\|X_i\|})X_i$. let us consider $r_i = \|X_i\|$.

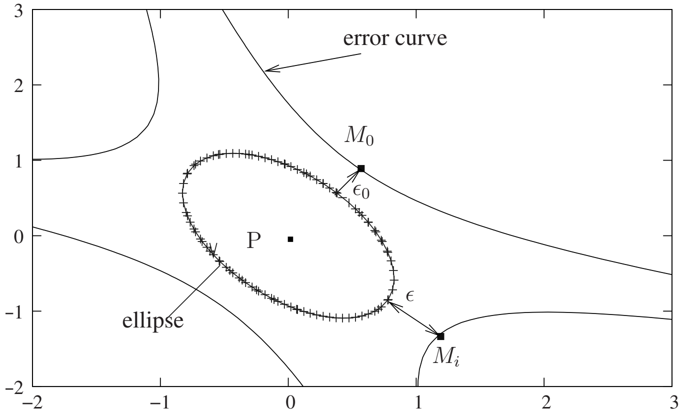

Fig. 1. Parameters

Our problem becomes:

Find three real numbers a, b and c that minimize $(4ab - c^2)$ under the constraints

$$\begin{cases} a > 0, b > 0, \\ ax_0^2 + by_0^2 + cx_0y_0 = \left(\dfrac{r_0}{r_0 - \epsilon_0}\right)^2, \\ ax_i^2 + by_i^2 + cx_iy_i = \left(\dfrac{r_i}{r_i - \epsilon}\right)^2, \quad 1 \le i \le n-1, \\ ax_k^2 + by_k^2 + cx_ky_k \ge 1, \quad 1 \le k \le n-1, \ k \ne i, \\ 4ab > c^2. \end{cases} \tag{6}$$

Numerically, the constraint $4ab > c^2$ need a threshold $\mathcal{T} > 0$ for which a solution of $4ab - c^2 - \mathcal{T}$ will be consider as a minimizer of $4ab - c^2$.

Minimization of $4ab - c^2$.
Let us call R_{max} the euclidian norm of the error point of the curve the most far away from it center P; we obtain:

$$\mathcal{T} = \left(\frac{2}{R_{max}(r_0 - \epsilon_0)}\right)^2$$

4 Algebraic Resolution

Let us examine the two equations of system (6)
If $x_0 = 0$, then $b = \dfrac{1}{(|y_0| - \epsilon_0)^2}$.

- If $x_i = 0$, then $b = \dfrac{1}{(|y_i| - \epsilon)^2}$ so we must have

$$\epsilon = |y_i| - |y_0| + \epsilon_0. \tag{7}$$

- If $y_i = 0$ and $y_k = 0$, we must have

$$\epsilon \geq |x_i| - |x_k|. \tag{8}$$

- If $x_i y_0 - y_i x_0 = 0$ and $x_i \neq 0$, we must have in this case

$$\epsilon = r_i \left(1 - \frac{x_0(r_0 - \epsilon_0)}{x_i r_0}\right). \tag{9}$$

Remark. We observe that if one coefficient of system (6) equal zero or if the points M_i, M_0 and the origin P are aligned, the choice of the parameters ϵ (or ϵ_0 in some cases) is not arbitrary. Those parameters should satisfy (7), (8) or (9) for the previous cases, but it will add new constraints to the problem and will not allow the precision we want.

The points representing those coefficients, when they equal zero, are almost four (the four halve-axes). They will be turn out of the axes with a small angle and also the eventually point on the axe (P, M_0). Doing that will not sensitively modify the problem neither its solution. We will consider that $x_0, x_i, x_k, y_0, y_i, y_k$ are all not zero for all i and for all k, and that $x_i y_0 - y_i x_0$ is not zero for all i. From both two equations of system (6), we extract c and a as a function of b.

Let us write:

$$\mathit{\Delta} = \frac{x_i}{x_0(x_i y_0 - x_0 y_i)}\left(\frac{r_0}{r_0 - \epsilon_0}\right)^2 - \frac{x_0}{x_i(x_i y_0 - x_0 y_i)}\left(\frac{r_i}{r_i - \epsilon}\right)^2;$$

$$\gamma = -\frac{y_i}{x_0(x_i y_0 - x_0 y_i)}\left(\frac{r_0}{r_0 - \epsilon_0}\right)^2 + \frac{y_0}{x_i(x_i y_0 - x_0 y_i)}\left(\frac{r_i}{r_i - \epsilon}\right)^2;$$

$$\tau = \frac{y_i y_0}{x_i x_0}; \qquad \lambda = \frac{x_0 y_i + y_0 x_i}{x_i x_0}; \quad \beta_k = y_k^2 + x_k^2 \tau - x_k y_k \lambda;$$

$$C_k = 1 - \frac{x_k(y_k x_i - y_i x_k)}{x_0(x_i y_0 - x_0 y_i)}\left(\frac{r_0}{r_0 - \epsilon_0}\right)^2 + \frac{x_k(y_k x_0 - x_k y_0)}{x_i(x_i y_0 - x_0 y_i)}\left(\frac{r_i}{r_i - \epsilon}\right)^2;$$

then,

$$c = \mathit{\Delta} - \lambda b; \quad a = \gamma + \tau b.$$

The inequality before the last one of (6) becomes

$$\beta_k b \geq C_k, \quad 1 \leq k \leq n-1;\ k \neq i. \tag{10}$$

So the function to minimize becomes

$$G(b) = (4\tau - \lambda^2)b^2 + 2(2\gamma + \lambda \mathit{\Delta})b - \mathit{\Delta}^2 - \mathcal{T},$$

the roots of which are

$$b_1 = \left(\frac{1}{x_i y_0 - x_0 y_i}\right)^2 \left[\left(\frac{x_0 r_i}{r_i - \epsilon}\right)^2 + \left(\frac{x_i r_0}{r_0 - \epsilon_0}\right)^2 + \mathcal{D}\right];$$

$$b_2 = \left(\frac{1}{x_i y_0 - x_0 y_i}\right)^2 \left[\left(\frac{x_0 r_i}{r_i - \epsilon}\right)^2 + \left(\frac{x_i r_0}{r_0 - \epsilon_0}\right)^2 - \mathcal{D}\right];$$

where

$$\mathcal{D} = \frac{2|x_i x_0|}{(r_0 - \epsilon_0)} \sqrt{\left(\frac{r_i r_0}{r_i - \epsilon}\right)^2 + \left(\frac{x_0 y_i - y_0 x_i}{R_{max}}\right)^2}.$$

The inequalities (4), (10) and the following bounds

$$\frac{(r_0 - \epsilon_0)^2}{R_{max}} \leq \beta \leq r_0 - \epsilon_0 \leq \alpha \leq R_{max},$$

gives b_{min} and b_{max} such that $b \in [b_{min}, b_{max}]$. So, we will compare $G(b_{min})$ and $G(b_{max})$ to determine the value of b that minimize G.

5 Numerical Resolution

The algorithm consists in:

- Discretizing the error curve around the vertex P into n points, determining the point M_0 of the curve which is the most closed to the origin P and the point M_{max} of the curve which is the most far away from P;
- moving out the points on the axes and also the point which is eventually aligned with M_0 and the origin;
- for all fixed M_i, switching the constraints for all the M_k ($k \neq 0;\ k \neq i$) and determining the optimal values of a, b and c.

5.1 Algorithm Order

This algorithm is globally quasi-linear because for each fixed M_i only one (the first) unsatisfied constraint on the M_k implies an interruption of the M_k's loop and switch to the next M_{i+1}(when the distance between a point of the error curve and the ellipse is greater than ϵ, that point will not satisfy any constraints). The points that satisfy the constraints are only those in the regions where the ellipse is tangent to the error curve, and their number is few. The only case where for each fixed M_i we go trough all the M_k is the case where the error curve is already an ellipse and in this case the algorithm order is $\mathcal{O}(n^2)$. But if the error curve is already an ellipse, we don't need this algorithm.

5.2 Convergence of the Approximation

It is natural to think that M_0 will not be the point of the error curve that is the most far away from the ellipse. So we can find ϵ_0 for which the solution of our approach problem converges to the optimal solution.

We can have an upper bound of ϵ_0:

The disc area of radius r_0 is less than the ellipse area of parameters $r_0 - \epsilon_0$ and R_{max} so, $\epsilon_0 < r_0 \left(1 - \frac{r_0}{R_{max}}\right)$. Although we don't have any absolute criteria of

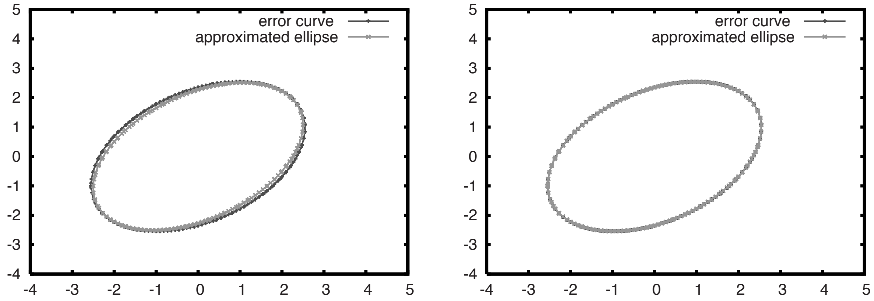

Fig. 2. An ellipse of parameters $\theta = \frac{\pi}{4},\ \alpha = 3,\ \beta = 2,\ n = 159$; left: $\epsilon_0 = 10^{-1}$; right: $\epsilon_0 = 10^{-4}$

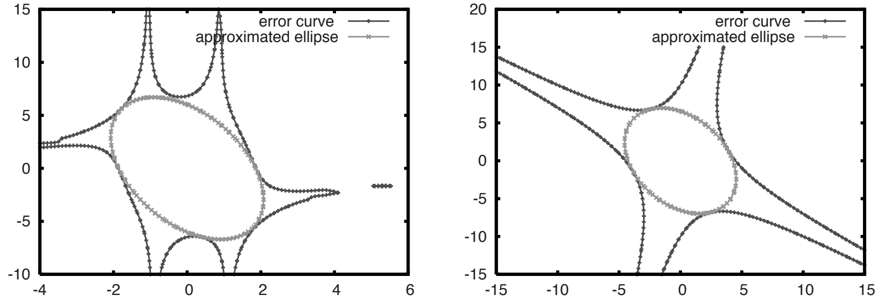

Fig. 3. left: $\mathcal{E} = 59,\ \epsilon_0$ opt. $= 10^{-5},\ n = 285,\ \ F(x,y) = -0.2y^3x^2 - yx^4 + 0.2y^3 - 9x^3 - 0.5xy^2 + 0.005x^2 - 0.1y^2$, right: $F(x,y) = 0.05x^2 - 0.01y^2 + 0.05xy,\ \mathcal{E} = 1,\ \epsilon_0$ opt. $= 0.001,\ n = 304$

choosing ϵ_0, when changing its values within a few iterations (one dozen) inside the code between 10^{-5} and it upper bound, we optimize the approximation (see figures 2, 3).

6 Validation

When solving numerically a partial differential equation with piece wise linear Lagrange finite element, one can use the hessian matrix of the function to approach as error estimator to generate adapted meshes [1].

As the hessian matrix $\mathcal{H}$ is a symmetric matrix, it absolute value define bellow when invertible is positive semi definite so it can be used to define a metric.

$$|\mathcal{H}| = \mathcal{R}|\Lambda|\mathcal{R}^{-1},$$

where $|\Lambda|$ is the diagonal matrix formed by the absolute value of the eigenvalues and $\mathcal{R}$ the matrix of the eigenvectors of $\mathcal{H}$. When $\mathcal{H}$ is not invertible, the zero eigenvalue is replaced by $\frac{1}{h_{max}^2}$, where h_{max} is the maximal authorized edge size in the mesh.

6.1 Some Examples

When applying our algorithm in some cases with

$$F(x,y) = \langle (x,y), \mathcal{H} \cdot (x,y) \rangle,$$

we obtain comparative results with the metric given by the absolute value of $\mathcal{H}$ (analytical) figures 4 and 5.

In the example of figure 4 for the case $\mathcal{H}$ invertible, one can observe that the obtained ellipse is closed to the analytical one when the number n of discretization points is enough, which equation in that example is

$$x^2 + y^2 + xy = 1.$$

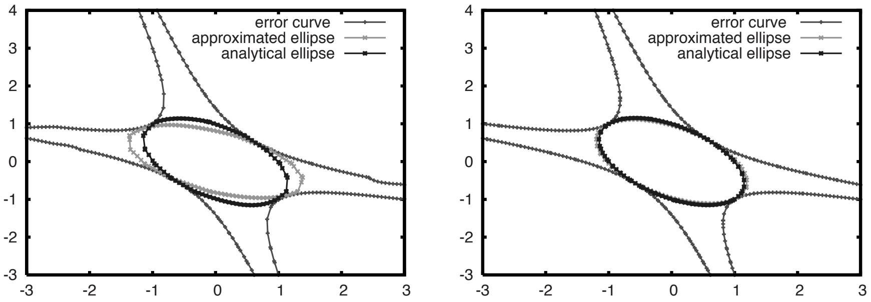

Fig. 4. $\mathcal{E} = 1$; $F(x,y) = 0.5x^2 + 0.5y^2 + 2xy$; left: $n = 228$; right: $n = 288$

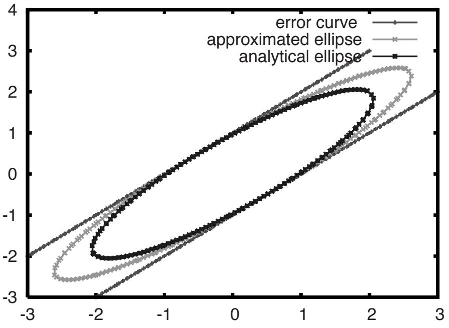

Fig. 5. $\mathcal{E} = 1$; $n = 234$ $F(x,y) = x^2 + y^2 - 2xy$

```
!"#$%&'()*+,-.#)(&
+/,!.$(&0"/,)+"/1($2&
3(45%65748.#*(9:;:;<9=!>1:?@A;9B!>1:?@AC?D
*(#!%=>;B>D
*(#!%,"(0%7E>D
0(42#,(%F5965;GE?D
0(42#,(%H5965;GA?D
0(42#,(%G5965;G>?D

0./,%0%7%)#/59,@94+/99:@B??!9A%@%=???I%9B@=@=?I%2"J9B;K?D
0./,%(**790===@=>@=>@=>IK@0==B@=>@=>@B>IK@0=BB@=>@B>@B>
        %%I0BBB@B>@B>@B>?@,"(0D
LL0===;0==B;0=BB;0BBB%$(*+M#)+M(4%"0%0%#/$%)5(%,"/4)#/)%,%
LL#*(%$(0+/($%+/%0+!(%0"/,)+"/1($2

0"*9+/)%+7ED+NE>D+II?
%%O
%%%%F5%3EE;3EA;3AAD%%%%
%%%%H5%0A70D
%%%%'()*+,-.#)(965;E:>;>1>>>E;K;(**;<3EE<C;3EA<C;3AA<CC;<=>;B>C?D
%%%%*(#!%,,7E>D
%%%%657#$#2)3(45965;,,@3EE;,,@3EA;,,@3AA;P4'()*+,7E;
                %53+/7>1>>>>E;/QM=7E>>>>>>?D
%%%%,".)NN3EE<C13#=NN&%&NN%3EA<C13#=%NN&%&NN%3AA<C13#=%NN(/$!D
%%%%2!")965;J#+)7>;247&651(24&?D
%%%%2!")90A;J#+)7>?D
%%%%G5%(5%7%9#Q490A!0??D
%%%%G5%!(57%!"RE>9(5?D
%%%%*(#!<+/)C%M+4"7<!S;!T;!U;!V;!:;!W;!K;!A;!E;>;ECD
%%%%2!")9!(5;0+!!7E;J#+)7>;M+4"7M+4";M#!.(7E;247&!(51(24&?D
%%%%,".)NN+NN&1111&NN(5<C13+/NN&%&NN(5<C13#=NN&%&NN(5<C14.3L(5<C1/
        %NN&%&NN+/)A$965?9(5?L651#*(#NN&%&NN651/)NN&%&NN651/MNN(/$!D
%%X
65748.#*(9:;:;<9=!>1:?@A;9B!>1:?@AC?D

*(#!%,(**79>1>>:@T1>A:US(!>:?Y>1:%D
0"*9+/)%+7ED+NE>D+II?
%%O
%%%%F5%3EE;3EA;3AAD%%%%
%%%%H5%0A70D
%%%%657#$#2)3(45965;0;(**7,(**;53+/7>1>>>>E;/QM=7E>>>>>>?D
%%%%,".)NN%3EE<C13#=%NN%&%&%NN%3EA<C13#=%NN%&%&%NN%3AA<C13#=%NN(/$!D
%%%%2!")965;J#+)7>;247&65A1(24&?D
%%%%2!")90A;J#+)7>?D
%%%%G5%(5%7%9#Q490A!0??D
%%%%G5%!(57%!"RE>9(5?D
%%%%*(#!<+/)C%M+4"7<!S;!T;!U;!V;!:;!W;!K;!A;!E;>;ECD
%%%%2!")9!(5;0+!!7E;J#+)7>;M+4"7M+4";M#!.(7E;247&!(5A1(24&?D
%%%%,".)NN+NN&1111&NN(5<C13+/NN&%&NN(5<C13#=NN&%&NN(5<C14.3L(5<C1/
        %NN&%&NN+/)A$965?9(5?L651#*(#NN&%&NN651/)NN&%&NN651/MNN(/$!D
%%X
```

The example of figure 5 show the case $\mathcal{H}$ non invertible where h_{max} have been arbitrary chosen in the analytical case which equation in that example is

$$0.5\left(\left(\frac{1}{h_{max}^2}+2\right)(x^2+y^2)+2\left(\frac{1}{h_{max}^2}-2\right)xy\right)=1.$$

We just need to variate the value of h_{max} to coincide with the approached solution.

6.2 Test with Freefem++

This algorithm have been implemented into Freefem++ and we present the obtained results when doing mesh adaptation for a function, in the following code.

In that code, the error function is given by the third derivates of the function f. That error function is written as an homogenous third degree polynomial of the coordinates of the local unit ball, as it's shown in [11]. We use our algorithm with $\mathcal{E} = 1$, to compute the metric on each vertex of the mesh, and the mesh is adapted with the obtained metric in the first loop. In the code, the parameters

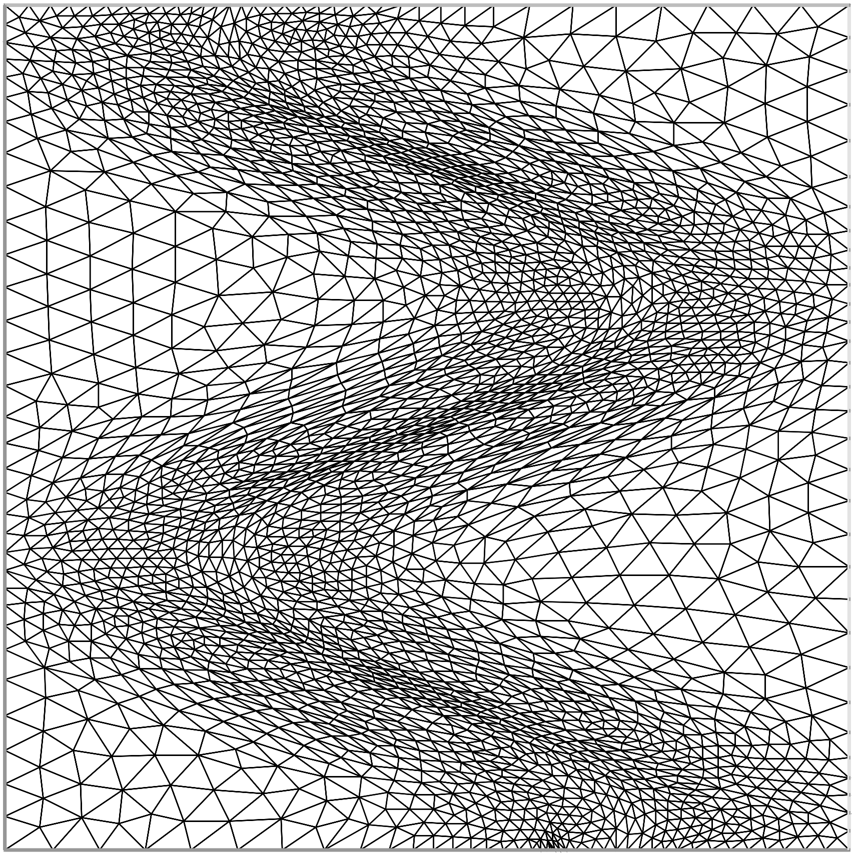

Fig. 6. Adapted meshes for the maximal error level around 10^{-5}, cc=10, coef=10, metrics computed with this algorithm for the (5) error estimator

cc and $coef$ are used to change the value of the error. Let err be the maximal error, and let Ct be a constant. err is given by

$$err = \frac{Ct}{coef \cdot cc^{l/2}}.$$

Using the second loop and the result of the first loop, we computed backward the real $cerr$ useful to have the same error level on each final mesh.

In the second loop, we made the usual mesh adaptation of Freefem++ for the same function f, based on metrics given by the hessian matrix.

Comments

In our example $l = 3$. According to the results of the second table of figure 9, if we take $Ct \simeq 2.6e{-}5$, we obtain $\frac{Ct}{10^{3/2}} = 8.22e{-}5$. It corresponds to the level of the maximal error in the first table.

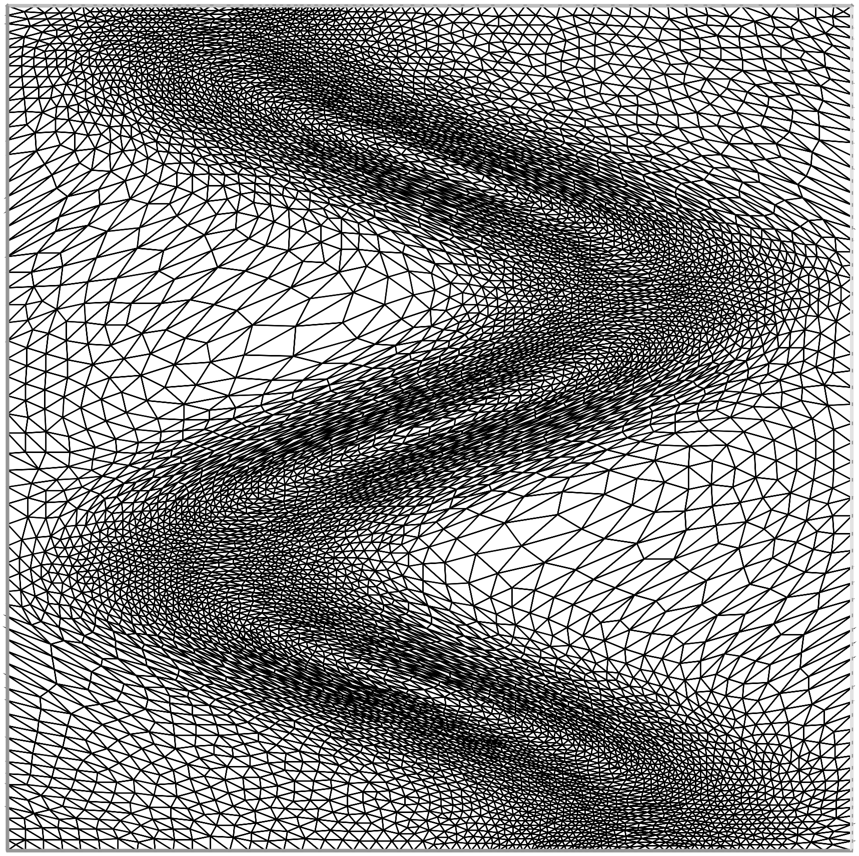

Fig. 7. Adapted meshes for the maximal error level around 10^{-5}, cc=10, coef=10, method based on the hessian

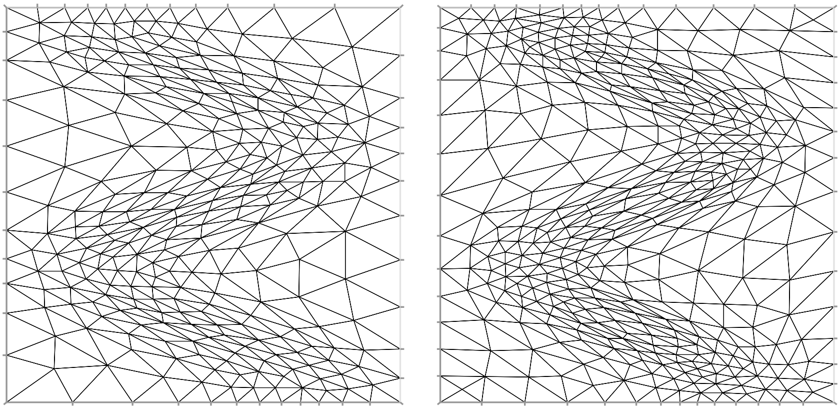

Fig. 8. Adapted meshes for the same maximal error level around 10^{-3}, cc=1, coef=10. Left: metrics computed with this algorithm for the (5) error estimator. Right: method based on the hessian.

iterations	error min 1	error min 2	error max 1	error max 2	vertices 1	vertices2
1	1.20059e-05	4.22866e-06	0.356521	0.358384	2050	3923
2	3.11655e-09	2.48466e-09	0.0043082	0.00437699	5194	17235
3	3.62071e-09	1.52523e-11	9.67516e-05	9.9182e-05	5198	15381
4	3.69877e-09	2.19181e-10	7.95773e-05	0.000103307	5194	15210
5	1.10912e-09	4.76638e-10	7.79361e-05	0.000106652	5264	15188
6	7.05385e-09	3.61698e-10	7.77929e-05	8.02886e-05	5200	15190
7	1.89551e-09	9.00541e-12	8.01738e-05	7.80091e-05	5200	15186
8	5.05435e-10	1.42422e-10	7.75686e-05	8.22038e-05	5200	15186
9	9.78865e-10	9.68075e-10	8.02579e-05	7.7833e-05	5218	15180
1	1.55839e-07	1.00832e-05	0.358347	0.35043	425	460
2	9.65766e-07	3.80745e-08	0.0134088	0.064436	536	743
3	3.58336e-06	2.93163e-08	0.00281739	0.00536942	536	733
4	2.34807e-07	1.97044e-07	0.0027611	0.00278402	540	733
5	1.10585e-06	9.35475e-08	0.00291751	0.0025183	540	732
6	7.26946e-08	3.9567e-07	0.00264979	0.00275217	540	733
7	1.49178e-06	7.63447e-08	0.00266049	0.00270342	540	733
8	8.80737e-08	5.55839e-08	0.00267491	0.00271847	540	733
9	1.04068e-06	2.8896e-07	0.00269423	0.00285419	540	733

Fig. 9.

We can observe on figures 8, 6, 7 and 9 that for the same maximal error level around 10^{-3} or 10^{-5}, the error estimator described in [11] and our algorithm uses much less elements than the classical mesh adaptation method with metrics based on the hessian matrix. We see that the convergence is obtained after five iterations, see figure 9. For the hessian based error built mesh, if we suppose that the function is a polynomial of degree 3, then the adaptation will be accurate. As our function is not a polynomial of degree three, we can observe that on figure 6, there is two thin mesh layers bounding the middle layer, and the corners of the domain are meshed with small size elements for the hessian based method. Those wrong effects are not observed with our new approach.

7 Conclusion

We have proposed a quasi-linear complexity algorithm in two spacial dimension useful to build numerically metrics for any given error estimation represented by a closed curve around a mesh vertex. As a metric can be represented by an ellipse, this algorithm is based on the optimization of an ellipse area. The main purpose of this work is to adapt mesh when doing numerical simulations where the interpolation error estimator is not based on the hessian matrix.

The obtained numerical results of our tests are encouraging. We have presented an example of mesh adaptation using metrics computed with this algorithm with respect to the new interpolation error estimation [11], done in Freefem++. That example seems to give better results than the usual adaptation procedure of Freefem++ with metrics based on the hessian, according the maximal error. We hope that we can expect "good" performances of the future new mesh adaptation routine of the software Freefem++ [14] where this algorithm could be used for the Lagrange finite element interpolation of higher order.

A future work on the local representation of other types of error estimators like residual estimators could be interesting. Then we will see how this algorithm could used or not on those cases. It development into dimension three is a future challenge in so far as the software Freefem++ is now moving from dimension two to dimension three.

We thank the reviewers for their comments which helped to improve the quality of this paper. We also thank Pascal Ventura who help us for the achievement of this work.

References

1. Alauzet, F.: Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en mecanique des fluides. PhD thesis, université de Montpellier II (October 2003)
2. Alauzet, F., Dervieux, A., Frey, P.: Multi-dimensional continuous metric for mesh adaptation. In: 15th International Meshing Roundtable, September 2006, pp. 191–213 (2006)
3. Bernardi, C., Maday, Y., Rapetti, F.: Discrétisations variationnelles de problèmes aux limites elliptiques. Mathématiques & Applications (Berlin) [Mathematics & Applications], vol. 45. Springer-Verlag, Berlin (2004)
4. Cao, W.: On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. SIAM J. Numer. Anal. 43(1), 19–40 (2005) (electronic)

5. Ciarlet, P.G., Lions, J.-L. (eds.): Handbook of numerical analysis. Handbook of Numerical Analysis, vol. II, p. 928. North-Holland, Amsterdam (1991); Finite element methods. Part 1
6. Ciarlet, P.G., Wagschal, C.: Multipoint taylor formulas and applications to the finite element method. Numer. Math 17, 84–100 (1971)
7. Ciarlet, P.G.: The finite element method for elliptic problems. Classics in Applied Mathematics, vol. 40. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2002)
8. Frey, P., Georges, P.L.: Maillages: Applications aux Elements Finis, Hermés Science, Paris (1999)
9. Hecht, F.: The mesh adapting software: bamg. INRIA report (1998), `http://www-c.inria.fr/gamma/cdrom/www/bamg/eng.htm`
10. Hecht, F.: Métriques et indicateurs d'erreur. In: Cours maillage. Paris, INRIA, CEA, EDF, février (2003)
11. Hecht, F.: Quelques estimations d'erreurs d'interpolation anisotropes pour eléments finis de lagrange. Numerical Analysis and Scientific Computing for Partial Differential Equations and their Challenging Applications, février (2008)
12. Hecht, F., Georges, P.L.: Nonisotropic Grids, ch. 20. CRC Press, Boca Raton (1999)
13. Hecht, F., Kuate, R.: Reconstruction de la matrice hessienne d'une fonction discrète. Laboratoire Jacques-Louis Lions (preprint, 2008)
14. Hecht, F., Leharic, A., Pironneau, O.: Freefem++, language for finite element method and Partial Differential Equations (PDE). Université Pierre et Marie, Laboratoire Jacques-Louis Lions, `http://www.freefem.org/ff++/`
15. Leservoisier, D.: Stratégies D'adaptation et de raffinement de maillage en Mécanique des Fluides Numérique. PhD thesis, Université Pierre et Marie Curie (September 2001)
16. Manole, C.-M., Vallet, M.-G., Dompierre, J., Guibault, F.: Benchmarking second order derivatives recovery of a piecewise linear scalar field. Département de génie informatique, Ecole Polytechnique de Montréal Canada (2005)
17. Picasso, M.: An anisotropic error indicator based on Zienkiewicz-Zhu error estimator: application to elliptic and parabolic problems. SIAM J. Sci. Comput. 24(4), 1328–1355 (2003) (electronic)
18. Zienkiewicz, O.C., Zhu, J.Z.: The superconvergent patch recovery and posterior estimates, Part I: The recovery technique. Int. J. Numer. Meth. Engng 33(7), 1331–1364 (1992)