A tensorial-based Mesh Adaptation for a Poisson problem

G. Brèthes*
A. Dervieux**

* INRIA, Projet Ecuador, 2004 route des lucioles - BP 93, 06902 Sophia Antipolis Cedex, France
**INRIA, Projet Ecuador, 2004 route des lucioles - BP 93, 06902 Sophia Antipolis Cedex, France and Lemma, 06410 Biot, France, alain.dervieux@inria.fr

A tensorial-based Mesh Adaptation for a Poisson problem

Abstract

This paper discusses anisotropic mesh adaptation, addressing either a local interpolation error, or the error on a functional, or the norm of the approximation error, the two last options using an adjoint state. This is explained with a Poisson model problem. We focus on metric-based mesh adaptation using *a priori* errors. *Continuous-metric methods* were developed for this purpose. They propose a *continuous* statement of the mesh optimisation problem, which need to be then discretised and solved numerically. *Tensorial-metric based methods* produce directly a *discrete* optimal metric for interpolation error equirepartition. The novelty of the present paper is to extend the tensorial discrete method to addressing (1) L^1 errors and (2) adjoint-based analyses, two functionalities already available with continuous metric. A first interest is to be able to compare tensorial and continuous methods when they are applied to the reduction of approximation errors. Second, an interesting feature of the new formulation is a potentially sharper analysis of the approximation error. Indeed, the resulting optimal metric has a different anisotropic component. The novel formulation is then compared with the continuous formulation for a few test cases involving high gradient layers and gradient discontinuities.

Keywords

Poisson problem, goal-oriented mesh adaptation, anisotropic mesh adaptation, adjoint, metric

1. Introduction

Mesh adaptation is an important component in the research of a better control of numerical error in Computational Mechanics. While the aim of our research is to propose methods applying to the mesh adaption of various PDE's (Partial Differential Equations), we start discussing here the case of a very simplified model useful in Computational Structural Mechanics (CSM) and Computational Fluid Dynamics (CFD), the Poisson problem. The two main ingredients for this will be metric parametrization of mesh and approximation error estimates.

We focus on methods which prescribe a somewhat anisotropic optimal mesh under the form of a parametrization of it by a Riemannian metric. A Riemannian metric is a continuous matrix field defined on the computational domain Ω:

$$\mathcal{M} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^{d^2} \quad \mathbf{x} \mapsto \mathcal{M}(\mathbf{x})$$

where $\mathcal{M}(\mathbf{x})$ is a symmetric matrix, in $\mathbb{R}^2$:

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})^t \begin{pmatrix} \frac{1}{\Delta\xi(\mathbf{x})^2} & \\ & \frac{1}{\Delta\eta(\mathbf{x})^2} \end{pmatrix} \mathcal{R}(\mathbf{x})$$

defining two mesh stretching directions by its eigen vectors and to mesh sizes $\Delta\xi(\mathbf{x}), \Delta\eta(\mathbf{x})$ in those directions. Many mesh generators are able to build meshes in accordance to the specifications (stretching and sizes) of a given metric field. The Riemannian metric should be obtained from an error analysis. One option is the solution of a continuous optimization problem based on a continuous extension of numerical error. This is proposed, among other works, in [26, 27]. Another option defines a discrete equation for a discrete metric on each vertex of the current mesh. It is proposed in [17, 18] and relies on edge-based *tensorial formalism*. Both methods can be equally applied to CFD (see many references in the sequel) and to CSM, we refer to two recent typical works in elasticity, [25] and for fracture problems, [6].

Continuous and *tensorial* metrics both rely on the parametrization of the mesh by a spatial field defining in any point of the computational domain a matrix giving information on mesh size in all the spatial directions.

Both methods solve an optimality system. The continuous metric builds a continuous optimality system which has, afterwards, to be discretised and solved, while the tensorial metric builds a discrete optimality system to be solved directly. Also, the continuous metric theory defines the ideal metric to be choosen. The resulting ideal mesh produced by metric optimization is the so-called *unit mesh*. It is defined from the optimal metric as a mesh with all its edges of unit length with respect to the metric. In contrast, the tensorial metric obtained from an optimization step in [17, 18, 12] is provided by the *modification* to apply to the current mesh in order to obtain the ideal mesh. Then the way to parameterize the final mesh with the two metrics is different, since the ideal mesh is with edges of unit length for the continuous metric, while the tensorial metric defines the ideal mesh from local directional amplifications of the background mesh. Further, the

constraint imposing a prescribed number of nodes is formulated on a vertex by vertex mode for the continuous metric and on an edge by edge mode for the tensorial method.

Let us consider now which error functional is chosen in the two methods. Both methods apply to the minimisation of the P_1-interpolation error committed on one or several *sensors* depending on the PDE solution u, e.g.:

$$\text{Find } \mathcal{M}_{opt} \text{ which minimizes } |u - \Pi_{\mathcal{M}} u|$$

where $\Pi_{\mathcal{M}}$ is the P_1-interpolation operator on the current mesh, parameterized by $\mathcal{M}$. For a representative sample of Hessian-based methods, cf. [15, 19, 2, 32, 22, 33, 24, 16, 3, 36]. Continuous and tensorial Hessian-based methods involve the equi-distribution method, which turns out to finding the metric which minimizes a L^∞ norm of the interpolation error:

$$\mathcal{M}_{opt} = \operatorname{Arg\,min} |u - \Pi_{\mathcal{M}} u|_{L^\infty}.$$

The continuous Hessian-based methods also involves the multiscale method, defined as minimizing the L^p interpolation error of the sensors for $p \neq \infty$.

$$\mathcal{M}_{opt} = \operatorname{Arg\,min} |u - \Pi_{\mathcal{M}} u|_{L^p}.$$

In order to minimize the interpolation error, it is replaced by an asymptotic equivalent (when mesh get finer), which is expressed in terms of the Hessian derivative of the sensor. These methods are refered as *feature-based* or *Hessian-based methods*. While taking into account some features of the solution of the PDE, they do not take into account the features of the PDE itself. Also, when an interpolation-based adaptation is applied to a system, it is not always easy to choose a set of sensors and their weights. However, if the sensors are cleverly chosen, a good convergence of the whole approximate solution field to the exact solution field is usually observed.

Goal-oriented methods allow to take into account the PDE under study. A combination with anisotropic Hessian-based adaption is proposed in [34]. Goal-oriented optimal methods [29, 10, 36], minimize with respect to the metric the approximation error committed on the evaluation of a scalar functional J depending on the PDE solution:

$$\text{Find } \mathcal{M}_{opt} \text{ which minimizes } |J(u) - J(u_{\mathcal{M}})|, \quad u_{\mathcal{M}} \text{ approximate solution of PDE.}$$

They do take into account the features of the PDE, typically through the use of an adjoint state. Goal-oriented methods needs also to rely on an error estimate (and on its sensitivity to mesh). Further, the goal-oriented adaptation criterion is mathematically derived from the functional chosen, and this delivers from the difficult task of choosing sensors as for interpolation-based adaptation.

Several methods have been proposed for reducing the approximation error through an estimate. A pioneering approach is the work of Becker and Rannacher [8] which relies, as many estimate-based work, on an *a posteriori* estimate.

$$|J(u) - J(u_{\mathcal{M}})| \leq \text{function}_{post}(\mathcal{M}, u_{\mathcal{M}}).$$

A good synthesis concerning *a posteriori* estimates is [35]. An interesting feature of an *a posteriori* estimate is that it is directly expressed in terms of the approximate solution, assumed to be available in a mesh adaption loop. A second interest is that it does not require the use of higher order (approximate) derivatives, in contrast to truncation analyses. However, these works do not address anisotropy. Adjoint-based and metric-based anisotropic mesh adaption is a difficult topic. Before going into deeper details of the method we develop, let us mention that an *a priori* analysis relying on element-mapping is proposed in [21]. In [36], a metric optimization is performed from local perturbation of the mesh and of the solution.

A priori estimates depend on the exact solution:

$$|J(u) - J(u_{\mathcal{M}})| \leq \text{function}_{prio}(\mathcal{M}, u).$$

They rely quasi-systematically on Taylor series, either through divided differences, or through polynomial approximation of functions. Then approximations of higher order derivatives of solution need be *recovered* from the approximate solution, typically:

$$|\frac{\partial^2 u}{\partial x^2}| \equiv D_2^{\mathcal{M}}(u_{\mathcal{M}}).$$

This is a delicate job since nothing ensures that a higher order derivative of the approximate solution is a good approximation of the corresponding higher order derivative of the exact solution, see [37] for a fundamental paper on the question. Assuming that we have such a good recovery, Taylor series can be easily used for proposing a somewhat optimal mesh. Further, *a priori* estimates can also provide *correctors* $u'_{\mathcal{M}}$, which are numerically computable fields close to the approximation error $u - u_{\mathcal{M}}$:

$$u'_{\mathcal{M}} \equiv u - u_{\mathcal{M}}.$$

In the present work, we use a corrector defined in [13]. In the present paper we use the tensorial formulation in order to build a novel *a priori* estimate for the Poisson equation which does not explicitly require the evaluation of higher-order derivatives.

Thanks to the goal-oriented formulation, the metric-based mesh adaptation becomes a well-posed optimization problem for the reduction of a genuine approximation error. However, goal-oriented optimal methods are specialized to a given scalar output. Features of the solution field which are not related to this output may be neglected by the automatic mesh improvement. As a consequence, these methods do not systematically provide a globally convergent solution field.

In the present paper, we study a *norm-oriented formulation* (according to [13]). In this third mesh adaptation method, the user can prescribe a norm of error $|u - u_h|$ which the algorithm will minimize with respect to the metric parametrization of the mesh.

$$\text{Find } \mathcal{M}_{opt} \text{ which minimizes } |u - u_{\mathcal{M}}|, \quad u_{\mathcal{M}} \text{ approximate solution of PDE.}$$

As a consequence, with an adequate choice of the norm, the norm-oriented mesh adaptation produces convergent solution fields.

The continuous approach for Hessian-based, goal-oriented, and norm-oriented has been defined in papers like [29, 10, 13].

The purpose of this paper is to analyse the possible novelties which can be derived from the extension and application of a tensorial method to L^1 Hessian-based, to goal-oriented, and to norm-oriented problematics.

The main feature of tensorial approach which we shall exploit is the derivation of the optimal metric thanks to a inversion using the tensorial calculus in the main error term. In order to adapt this feature to L^1-Hessian, to goal-oriented, to norm-oriented problematics, we unify the parametrization by choosing the unit-mesh formulation and by measuring the number of nodes on a vertex basis.

Although the proposed method is a rather general method extending to complex CFD or CSM models, see for example [30] for CFD, we consider in this paper a 2D Poisson problem discretized by the usual linear finite-element method. This choice is motivated first by the rather complete set of theoretical works available for the finite-element approximation of a Poisson problem. This amount of theoretical background reduces as much as possible (although far from completely) the heuristics to introduce in building the mesh adaptation analysis. A second motivation is the easy availability of exact solutions defined in a simple way. This allows to build a kind of benchmark allowing to compare mesh adaptation methods. The proposed approach extends naturally from the Poisson problem to the standard elasticity models. On the other hand, the Poisson problem with variable coefficient is a central equation in CFD, and in particular for two-fluid models (see [23] for a mesh-adaptive example). Let us finally mention that the proposed method extends naturally to systems, which can be useful in case where the choice of sensors of an interpolation-based adaptation is delicate.

Paper overview: in Section 2 we define the Poisson problem under study and propose a simple corrector for the discrete solution, which will be used in Sec.6. Section 3 recall the main features of the continuous metric adaptation. This assume that mesh and approximation errors are converted into continuous fields, namely a continuous metric, and a continuous approximate solution. Then it is possible to formulate a continous optimization problem, which we shall solve analytically. The optimality conditions are then discretized and approximately solved by introducing the mesh generator. Section 4 introduces the discrete context for tensorial metric optimization. A discrete erro field is defined on each edge of the mesh. The optimization of the dicrete metric is formulated edge-by edge and solved and put as parameter in the mesh generator. In Section 5, we focalise on a particular family of errors, the edge-based second-order errors. Three types of second-order errors are introduced: interpolation error, goal-oriented error, norm-oriented error. Section 6 gives the optimal metric for the family of errors. Numerical examples are presented in Section 7 and the paper is concluded by Section 8.

2. Poisson problem approximation

Let us introduce some notations: let $V = H_0^1(\Omega)$, Ω being a smooth enough computational domain of $\mathbb{R}^2$ or $\mathbb{R}^3$. The continuous PDE system is written in short:

$$u \in V, \quad Au = f \quad \text{or} \quad u \in V, \quad \forall\ \phi \in V, \quad a(u,\phi) = (f,\phi). \tag{1}$$

To fix the ideas and simplify notations,

$$A = -\sum \frac{\partial}{\partial x_k}\frac{\partial}{\partial x_k} \quad \Leftrightarrow \quad a(u,\phi) = \int_\Omega \nabla u \cdot \nabla \phi d\mathbf{x}.$$

But the extension to a coercive general case where $A = -\sum \frac{\partial}{\partial x_k}(a_{k\ell}(\mathbf{x})\frac{\partial}{\partial x_\ell}) + a_0(\mathbf{x})$ (where $a_{k\ell}, a_0$ are scalar, possibly discontinuous, fields) is not difficult. Let $\Omega_h = \Omega$ for simplicity, τ_h a triangulation of Ω_h, and V_h be the usual P_1-continuous finite-element approximation space related to τ_h:

$$V_h = \{\phi_h \in \mathcal{C}^0(\bar{\Omega}) \cap V, \phi_h|_T \text{ is affine } \forall T \in \tau_h\}.$$

We denote by Π_h the usual interpolation operator:

$$\Pi_h : \mathcal{C}^0(\bar{\Omega}) \to V_h \quad \Pi_h\phi(\mathbf{x}_i) = \phi(\mathbf{x}_i) \forall \mathbf{x}_i, \text{ vertex of } \tau_h.$$

The finite-element discretisation of (1) is written:

$$u_h \in V_h \ \text{ and } \ \forall\ \phi_h \in V_h \ , \ \ a(u_h,\phi_h) = (f_h,\phi_h) \tag{2}$$

with $f_h = \Pi_h f$. We are interested first in getting estimates of the approximation error $u_h - u$. Let N be the dimension of V_h, that is the number of vertices in τ_h. We observe that (2) is equivalent to computing the array $\mathbf{u}_h$ of the degrees of freedom of the discrete solution:

$$\mathbf{u}_h \in \mathbb{R}^N \ ; \ \mathbf{A}_h\mathbf{u}_h = \mathbf{f}_h. \tag{3}$$

From the above array we derive u_h by

$$u_h = \sum_{i=1,N} \mathbf{u}_{h,i} N_i(\mathbf{x})$$

where the N_i are the canonic finite-element basis of V_h:

$$N_i \in V_h, \quad N_i(\mathbf{x}_j) = 1 \text{ if } i = j, \ 0 \text{ else.}$$

We also introduce the interpolation operator Π_h:

$$for \ \ v \in V \cap H^2(\Omega), \quad \Pi_h v \in V_h, \ \ (\Pi_h v - v)(\mathbf{x}_i) = 0 \ \ \forall \mathbf{x}_i \text{ vertex of } \ \tau_h.$$

Let us now study the approximation error $u - u_h$. We start from the discrete above statement

$$a(u_h,\phi_h) = (f_h,\phi_h) \quad \forall \phi_h \in V_h.$$

and observe that for the exact solution satisfies:

$$a(u, \phi_h) = (f, \phi_h) \quad \forall \phi_h \in V_h.$$

Then

$$a(u_h, \phi_h) = a(u, \phi_h) + (f_h - f, \phi_h) \quad \forall \phi_h \in V_h.$$

Assuming that the solution u is sufficiently smooth, we get:

$$a(\Pi_h u - u_h, \phi_h) = a(\Pi_h u - u, \phi_h) + (f - f_h, \phi_h) \quad \forall \phi_h \in V_h. \tag{4}$$

We call $\Pi_h u - u_h$ the *implicit error*. It differs from the approximation error by an interpolation error:

$$u - u_h = u - \Pi_h u \quad + \quad \Pi_h u - u_h.$$

The rest of the section is devoted to finding a *corrector*, *i.e.* a discrete field u'_{prio} which would be rather easy to compute and would be an approximate of the implicit error:

$$u'_{prio} \equiv \Pi_h u - u_h.$$

Let us evaluate the RHS of (4). The second term of (4)'s RHS is easy to evaluate (we know f and f_h). The first term of (4)'s RHS can be transformed as follows:

$$\begin{aligned} a(\Pi_h u - u, \phi_h) &= \sum_T \int_T \nabla \phi_h \nabla (\Pi_h u - u) \, \mathrm{d}x\mathrm{d}y \\ &= \sum_T \int_{\partial T} (\Pi_h u - u) \nabla \phi_h \cdot \mathbf{n} \, \mathrm{d}\sigma. \end{aligned}$$

Then we get:

$$\begin{aligned} a(\Pi_h u - u, \phi_h) &= K(\phi, u_h) \qquad \text{with} \\ K(\phi, u_h) &= \sum_{\partial T_{ij}} \nabla(\phi_h|_{T_i} - \phi_h|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\Pi_h u - u) \, \mathrm{d}\sigma \end{aligned} \tag{5}$$

where the last sum is taken for all edges $ij = \partial T_{ij}$ (2D case) separating triangles T_{ij}^+ and T_{ij}^- of the triangulation. The unit vector $\mathbf{n}_{ij}$ normal to ∂T_{ij} is pointing outward T_i.

Our *corrector* is defined by:

$$\begin{aligned} a(u'_{prio}, \phi_h) &= K(\phi_h, u_h) + (f - f_h, \phi_h) \qquad \text{with} \\ K(\phi_h, u_h) &= \sum_{\partial T_{ij}} (\nabla \phi_h|_{T_i} - \nabla \phi_h|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\pi_h u_h - u_h) \, \mathrm{d}\sigma \end{aligned} \tag{6}$$

where the term $\pi_h u_h - u_h$ is built on the edge T_{ij} as a quadratic function vanishing at both extremities of T_{ij}, and of second derivative in direction T_{ij} equal to the approximate second derivative in same direction of u_h. The corrector u'_{prio} will be used in Sec.6.3.

3. Continuous metric parametrization

3.1 Mesh parametrization

We recall the continuous mesh framework, introduced in [26, 27]. The main idea of this framework is to model discrete meshes by Riemannian metric fields. It allows us to define a differentiable optimization problem [1, 5], *i.e.,* to apply on the class continuous metrics a calculus of variations which cannot be applied on the class of discrete meshes. This framework lies in the class of metric-based methods. A continuous mesh $\mathcal{M}$ of the computational domain Ω is identified to a Riemannian metric field [11] $\mathcal{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric 3×3 (in 3D, 2×2 in 2D) matrix having $(\lambda_i(\mathbf{x}))_{i=1,3}$ as eigenvalues along the principal directions $\mathcal{R}(\mathbf{x}) = (\mathbf{v}_i(\mathbf{x}))_{i=1,3}$. Sizes along these directions are denoted $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i^{-\frac{1}{2}}(\mathbf{x}))_{i=1,3}$ and the three *anisotropy quotients* r_i are defined by: $r_i = h_i^3\,(h_1h_2h_3)^{-1}$. The diagonalisation of $\mathcal{M}(\mathbf{x})$ writes:

$$\mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x})\,\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & & \\ & r_2^{-\frac{2}{3}}(\mathbf{x}) & \\ & & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x}), \tag{7}$$

The *vertex density* d is equal to: $d = (h_1h_2h_3)^{-1} = (\lambda_1\lambda_2\lambda_3)^{\frac{1}{2}} = \sqrt{\det(\mathcal{M})}$. By integrating it, we define the *total number of vertices* $\mathcal{C}$:

$$\mathcal{C}(\mathcal{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}. \tag{8}$$

Given a continuous mesh $\mathcal{M}$, we shall say, following [26, 27], that a discrete mesh $\mathcal{H}$ with edges $\mathbf{x}_{ij} = \mathbf{x}_j\mathbf{x}_i$ of the same domain Ω is a **unit mesh with respect to** $\mathcal{M}$, if each edge $\mathbf{x}_{ij}$ of $\mathcal{H}$ verifies:

$$\forall i \in [1,3], \quad \ell_{\mathcal{M}}(\mathbf{x}_{ij}) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right],$$

in which the length of an edge $\ell_{\mathcal{M}}(\mathbf{x}_{ij})$ is defined as follows:

$$\ell_{\mathcal{M}}(\mathbf{x}_{ij}) = \int_0^1 \sqrt{{}^t\mathbf{x}_{ij}\;\mathcal{M}(\mathbf{x}_i + t\,\mathbf{x}_{ij})\;\mathbf{x}_{ij}}\;\mathrm{d}t.$$

We want to emphasize that the set of all the discrete meshes that are unit meshes with respect to a unique $\mathcal{M}$ contains an infinite number of meshes, but these meshes produce approximates solutions of (1) which are sufficiently close to each others, so that we consider these meshes as an equivalence class of meshes. We henceforward denote by $\mathbf{x}^{\mathcal{M}}$ a unit mesh for metric $\mathcal{M}$. The unit edge property of unit mesh writes in short:

$$\text{For a unit mesh } \mathbf{x}^{\mathcal{M}} \text{ , any edge } \mathbf{x}_{ij}^{\mathcal{M}} \text{ satisfies } \left(\mathbf{x}_{ij}^{\mathcal{M}}, \mathcal{M}\mathbf{x}_{ij}^{\mathcal{M}}\right) = 1.$$

3.2 Optimal continuous metric

We recall, following [26, 27], the main features of the metric-based analysis initiated in several papers like [19, 15, 2]. The continuous interpolation error of a function u defined on the computational domain is denoted now:

$$u - \pi_{\mathcal{M}} u = |tr(\mathcal{M}^{-\frac{1}{2}} |H_u| \mathcal{M}^{-\frac{1}{2}})| \tag{9}$$

where H_u is the Hessian of u. Let denote also $\mathcal{M}$ a unit mesh for metric $\mathcal{M}$. We shall use the estimate

$$|u - \Pi_{\mathcal{M}} u| \approx \frac{1}{8} |u - \pi_{\mathcal{M}} u|. \tag{10}$$

Once we have a continuous tensorial error kernel, we consider minimizing:

$$j_p(\mathcal{M}) = \|u - \pi_{\mathcal{M}} u\|_{\mathbf{L}^p(\Omega_h)} \tag{11}$$

and we define as optimal metric the one which minimizes the right hand side under the constraint of a total number of vertices equal to a parameter N. In the case of a bounded p, after solving analytically this optimization problem, we get -without using the fact that H is anything but a positive symmetric matrix- the unique optimal $(\mathcal{M}_{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ as:

$$\mathcal{M}_{\mathbf{L}^p} = \mathcal{K}_p(1, H) \tag{12}$$

where we use (throughout this paper) the following notation defined for a scalar field k and for $1 < p \leq \infty$:

$$\mathcal{K}_p(k, H) = D_{\mathbf{L}^p} \left(\det(kH)\right)^{\frac{-1}{2p+2}} kH \quad \text{and} \quad D_{\mathbf{L}^p} = N^{\frac{2}{2}} \left(\int_\Omega (\det(kH))^{\frac{p}{2p+2}} \right)^{-\frac{2}{2}}, \tag{13}$$

In this formulation, $D_{\mathbf{L}^p}$ is a real number imposing that the continuous mesh has a complexity N. The scalar field $(\det(H))^{\frac{-1}{2p+2}}$ is a local normalization term accounting for the sensitivity of the $\mathbf{L}^p$ norm. *A particular case: L^∞-norm/iso-distribution* It is important to remark that error iso-distribution is taken into account by setting $p = \infty$, a limiting case for which we get:

$$(\det(H))^{\frac{-1}{2\infty+2}} = 1.$$

and

$$\mathcal{M}_{\mathbf{L}^\infty} = \mathcal{K}_\infty(1, H) \;\text{ with }\; \mathcal{K}_\infty(1, H) = D_{\mathbf{L}^\infty} H$$

where $D_{\mathbf{L}^\infty}$ is defined from the specification of the number of nodes of the mesh.

Another way to see it is to write that the error is uniform, indeed:

$$\mathcal{M}_{\mathbf{L}^\infty}(\mathbf{x}) = \text{const.(indep. of } \mathbf{x}) \; H$$

implies that:

$$\text{trace}\big(\mathcal{M}_{\mathbf{L}^\infty}^{-\frac{1}{2}}(\mathbf{x}) H(\mathbf{x}) \mathcal{M}_{\mathbf{L}^\infty}^{-\frac{1}{2}}(\mathbf{x})\big) = \text{const.(indep. of } \mathbf{x}).$$

Main case under study: L^1-norm optimisation The rest of the paper concentrates with the case:

$$p = 1.$$

Replacing the optimal metric $\mathcal{M}_{\mathbf{L}^1}$ in the L^1 norm shows that second-order convergence is obtained for smooth contexts. This can also be extended to non-smooth ones, cf. [28].

Let k a sufficiently smooth scalar function defined on Ω. We shall be, in the sequel, interested in minimizing the right-hand side of:

$$|(k, u - \Pi_{\mathcal{M}} u)_\Omega| \approx \int_\Omega \mathrm{trace}\big(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})|k(\mathbf{x})H(\mathbf{x})|\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\big)\mathrm{d}\mathbf{x}. \tag{14}$$

The optimum metric is given by:

$$\mathcal{M}^{1,k}_{opt} = \mathcal{K}_1(k, H) \ \text{ with } \ \mathcal{K}_p(k, H) \ \text{ defined in (13).} \tag{15}$$

It is interesting to compare this result with the result of equidistribution, at least for the particular case of an interpolation error. We observe that:

$$\begin{aligned} \mathcal{M}^{1,k}_{opt} = \text{const.} \ \ |k|^{\frac{3}{4}}|(\det|H|)^{-\frac{1}{4}}|H| \ &= \ \text{const.} \ \ |H_k| \\ H_k &= |k|^{\frac{3}{4}}|(\det|H|)^{-\frac{1}{4}}H. \end{aligned} \tag{16}$$

This means that the error minimisation in L^1 weighted by k is equivalent to an equidistribution process with a matrix H corrected by a scalar factor $|k|^{\frac{3}{4}}|(\det|H|)^{-\frac{1}{4}}$:

$$\mathcal{M}^{1,k}_{opt} = \text{const.} \ \ \mathcal{K}_\infty(|k|^{\frac{3}{4}}|(\det|H|)^{-\frac{1}{4}}, H).$$

In order to evaluate approximatively H, it is necessary to numercally differentiate the approximate solution by using a recovery as introduced in [37]. The precise recovery which we use in this paper is described in [4].

To synthetize, the continuous metric method yields the mesh adaptation solution under the form of a continuous optimality system involving:
- the continuous initial PDE,
- its continuous adjoint, and
- a stationarity condition explicitly solved by (15).

In practice, this optimality system is discretized and then numerically solved.

4. Edge-based tensorial approach

This section recalls in short the main features of the length distribution tensor method using edge-based errors. This method is introduced in [17]. We concentrate on the more recent formulation of [18]. Let us consider a mesh $\mathbf{x}$ described by its edges $\mathbf{x}^{ij}$ between

vertex i and vertex j. We call a *unit metric* of this arbitrary mesh a metric $\mathbb{M}_1(\mathbf{x})$ defined on each vertex of the mesh $\mathbf{x}$ which measures the mesh $\mathbf{x}$ as a unit mesh, in other words which satisfies (approximatively in practice) the relation:

$$\forall\ (i,j)\quad \big(\mathbb{M}_1(\mathbf{x})\mathbf{x}^{ij},\mathbf{x}^{ij}\big)=1.$$

Let $\Gamma(i)$ be the set of vertices which are neigbors if vertex i. We can write at vertex i:

$$\sum_{j\in\Gamma(i)}\big(\mathbb{M}_1(\mathbf{x})\mathbf{x}^{ij},\mathbf{x}^{ij}\big)=\sum_{j\in\Gamma(i)}1\quad\Rightarrow\quad\mathbb{M}_1(\mathbf{x})\ :\ \Big(\sum_{j\in\Gamma(i)}\mathbf{x}^{ij}\otimes\mathbf{x}^{ij}\Big)=|\Gamma(i)|,$$

where $|\Gamma(i)|$ is the cardinality of $\Gamma(i)$. When there exists at least d non-aligned edges around i we can solve for the value $\mathbb{M}^i$ of unit metric $\mathbb{M}_1(\mathbf{x})$ at vertex i as follows:

$$\mathbb{M}^i=\mathbb{M}_1(\mathbf{x})^i=\frac{1}{d}\Big(\sum_{j\in\Gamma(i)}\frac{1}{|\Gamma(i)|}\mathbf{x}^{ij}\otimes\mathbf{x}^{ij}\Big)^{-1}.$$

This metric, when applied for transforming the initial mesh $\mathbf{x}^{ij}$ into a new mesh, gives a new mesh with uniform edge length $||\widetilde{\mathbf{x}^{ij}}||=1$.

An second-order approximation with local edge error

$$e|_{\mathbf{x}^{ij}}=e_{ij}$$

equal to e_{ij} on the edge of length $||\mathbf{x}_{ij}||$ would have its error changed as follows if the length of $\mathbf{x}^{ij}$ is changed:

$$\widehat{\mathbf{x}^{ij}}=s_{ij}\ \mathbf{x}^{ij}\quad\Rightarrow\quad e|_{\widehat{\mathbf{x}^{ij}}}=s_{ij}^2\ e_{ij}.$$

Looking for a *uniform error* $e|_{\widehat{\mathbf{x}^{ij}}}=1$ we have to impose $s_{ij}=(1/e_{ij})^{1/2}$, that is to transform the initial mesh with the metric:

$$\mathbb{M}^i=\frac{1}{d}\Big(\sum_{j\in\Gamma(i)}\frac{1}{|\Gamma(i)|}e_{ij}^{-1}\mathbf{x}^{ij}\otimes\mathbf{x}^{ij}\Big)^{-1}.\tag{17}$$

Then it remains to multiply the metric by a constant allowing to control the total number of vertices in the new mesh (see [18]).

When comparing this formulation with the previous one, we observe a couple of differences.

- Formulation (17) is a discrete one while the continuous metric (12) is not.

- Formulation (17) takes into account errors which are defined along mesh edges while continuous metric (12) takes into account error fields which can then be integrated into L^p norms.

- Formulation (17) provides a corrected mesh from the initial one *instead of*, like the continuous metric method (12), giving the novel mesh as the unit mesh of an optimal metric.

In the sequel, we show that edge-based errors can also model error fields, and we unify the mesh parametrization to an optimal metric formulation.

5. Approximation of metric properties

The optimality system of the tensorial formulation relies on an edge-based error modelling. Then most of the important discrete fields need to be cast in an *edge-based format*. We introduce a few notations for this.

5.1 Generic mesh notations

Given a mesh $\mathcal{H}_{\mathbf{x}}$, we can define the following partitions.

- A *mesh-vertex* is a vertex of numero i and coordinates $\mathbf{x}_i$ of an element of the mesh.

- When there is an *edge* between vertex i and vertex j, we denote $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$.

- Two tetrahedra m and n having a common face have face mn or face nm as common face.

- *Elements* : triangles (i, j, k) or tetrahedra (i, j, k, l). Elements are divided in *sub-elements*: 6 *subtriangles* using medians and 24 *subtetrahedra* using median plans. The vertices of a subtetrahedron are : a mesh-vertex i , a center I_{ij} of an edge ij having i as extremity, the centroid g_{ijk} of a face ijk containing vertices i and j, the element centroid G_{ijkl}. The measure of a subtetrahedron of the tetrahedron T is $1/24\ meas(T)$.

- *Cell* i : for a vertex i of the mesh, cell i is union of sub-elements having i as vertex of the sub-element. A cell measure is defined as

$$meas_{\mathbf{x}}(i) = \frac{1}{dim+1} \sum_{T_{\mathbf{x}} \ni i} \quad meas(T_{\mathbf{x}})$$

where $T_{\mathbf{x}}$ are elements of $\mathcal{H}_{\mathbf{x}}$ containing vertex i.

- *2D-diamond* D_{ij} : union of the 4 subtriangles (of triangles ijk and ijl) having a side included in edge ij.

- *Face-diamond* $\bar{D}_{mn}$, where m and n are two tetrahedra having a common face ijk : union of 6 subtetrahedra having a subtriangle of the common face ijk as face.

- *Edge-diamond* D_{ij}: union of subtetrahedra having having a side included in edge ij.

The integral of a function e_{ij} defined on the edges can be approximated by:

$$err_{L^1} = \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_j e_{ij}$$

where the sum is taken over vertices(=cells), or introducing the diamond partition $\Omega = \cup \bar{D}_{mn}$ where m and n are elements with a common face:

$$err_{L^1} = \frac{1}{3} \sum_{\bar{D}_{mn}} meas_{\mathbf{x}}(\bar{D}_{mn})\ (e_{ij} + e_{ik} + e_{jk}).$$

where i, j, k are vertices of the face mn.

5.2 Discretizing an arbitrary continuous metric on a background mesh

In order to find the optimal metric we are given a background mesh $\mathbf{x}$. We assume that the unknown metric $\mathcal{M}$ is defined on the vertices $\mathcal{M}(\mathbf{x}_i) = \mathcal{M}^i$ of the background mesh and that it is P^1-continuously interpolated. The total number of nodes can be approximated on the mesh $\mathbf{x}$ by a quadrature of (8) as follows:

$$\mathcal{C}(\mathcal{M}) = \textstyle\sum_i meas_{\mathbf{x}}(i)\sqrt{det(\mathcal{M}^i)}.$$

To simplify, *we assume that the unit mesh is a deformation of* $\mathbf{x}$*, and that* $\mathbf{x}_{ij}^{\mathcal{M}}$ *and* $\mathbf{x}_{ij}$ *are colinear*. Then we can derive from the unit-mesh property a relation between the edge lengths of unknown mesh and the edge lengths of the background mesh:

$$(\mathbf{x}_{ij}^{\mathcal{M}}, \mathcal{M}\mathbf{x}_{ij}^{\mathcal{M}}) = 1 = (\mathbf{x}_{ij}\frac{|\mathbf{x}_{ij}^{\mathcal{M}}|}{|\mathbf{x}_{ij}|}, \mathcal{M}\mathbf{x}_{ij}\frac{|\mathbf{x}_{ij}^{\mathcal{M}}|}{|\mathbf{x}_{ij}|}) = (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})\frac{|\mathbf{x}_{ij}^{\mathcal{M}}|^2}{|\mathbf{x}_{ij}|^2}$$

$$\Rightarrow \mathbf{x}_{ij}^{\mathcal{M}} \approx \mathbf{x}_{ij}(\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-\frac{1}{2}}.$$

In order now to evaluate the approximation error provoked by the application of the unit-mesh, we need to define a *generic error model*.

6. Second-order error of a metric on a background mesh

To any given metric, *i.e.* to any given mesh, should correspond a numerical error field. Let us define a generic family of error field with values on mesh edges. We restrict to second-order *i.e.* quadratic errors, on the model of P_1-interpolation error.

Definition : An *edge-based second-order (or quadratic) error* produced by the use of the unit mesh $\mathbf{x}_{\mathcal{M}}$ of metric $\mathcal{M}$ has an intensity defined on edge $\mathbf{x}_{ij}^{\mathcal{M}}$ by:

$$e_{ij}^{\mathcal{M}} = \bar{e}_{ij}\ |\mathbf{x}_{ij}^{\mathcal{M}}|^2.$$

in which $\bar{e}_{ij}$ depends only on location and direction of $\mathbf{x}_{ij}^{\mathcal{M}}$, and is $O(1)$ when mesh becomes finer. Typically:

$$e_{ij}^{\mathcal{M}} = \quad |\mathbf{x}_{ij}^{\mathcal{M}}|^2 \quad \bar{e}_{ij}(\tfrac{1}{2}(\mathbf{x}_i^{\mathcal{M}} + \mathbf{x}_j^{\mathcal{M}}), \tfrac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|}).\square$$

Since we *a priori* know neither the optimal metric nor its mesh, it is useful to evaluate this error on a given background mesh $\mathbf{x}$. We use that the unit mesh is a deformation of $\mathbf{x}$ in such a way that $\mathbf{x}_{ij}^{\mathcal{M}}$ and $\mathbf{x}_{ij}$ are colinear. Then the intensity $e_{ij}^{\mathcal{M}}$ of the error with the unit mesh evaluated at middle of $\mathbf{x}_{ij}$ of the background mesh writes:

$$e_{ij}^{\mathcal{M}} = \quad |\mathbf{x}_{ij}|^2 \quad (\mathbf{x}_{ij}, \mathcal{M}_{ij}\mathbf{x}_{ij})^{-1} \quad \bar{e}_{ij}(\frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j), \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}) \tag{18}$$

where $\mathcal{M}_{ij}$ is evaluated on $\frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$. The mesh adaptation problem will be set as the research of the discrete metric, defined on mesh vertices and linearly interpolated, of a given number of nodes N

$$\mathcal{C}(\mathcal{M}) = N,$$

and minimizing the discrete error norm:

$$j(\mathcal{M}) = \sum_i meas_{\mathbf{x}}(i) \frac{1}{\Gamma(i)} \sum_{ij \ni i} e_{ij}^{\mathcal{M}}. \tag{19}$$

In Section 7.6 we determine the optimal mesh for this type of error, as far as $\bar{e}_{ij}$ is identified. The rest of the present section is devoted to the description of three examples of quadratic errors.

6.1 First example: interpolation error

The error committed in interpolating a smooth function on a P^1 mesh is a quadratic error. Indeed, the weighted P_1-interpolation error of a smooth function u on $\mathbf{x}_{ij}^{\mathcal{M}}$ can be estimated similarly to (9),(10) as follows:

$$\int_\Omega |g||u - \Pi_h u| \mathrm{d}\Omega \preceq \frac{1}{8} \sum_i meas_{\mathbf{x}}(i) \Gamma(i)^{-1} \sum_j e_{ij}^{\mathcal{M},g,u}(\mathbf{x}_{ij})$$

with:

$$e_{ij}^{\mathcal{M},g,u} = \quad |\mathbf{x}_{ij}^{\mathcal{M}}|^2 \quad |g_{ij}| \, |H_{ij}| \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \quad ,$$

and where $H_{ij} = H(\frac{1}{2}(\mathbf{x}_i^{\mathcal{M}} + \mathbf{x}_j^{\mathcal{M}}))$, $H(\mathbf{x})$ being the Hessian of u at point $\mathbf{x}$, and $g_{ij} = g(\frac{1}{2}(\mathbf{x}_i^{\mathcal{M}} + \mathbf{x}_j^{\mathcal{M}}))$. Here $\preceq$ holds for an inequality applying for a sufficiently fine mesh, with a multiplicative constant close to 1. The error can be evaluated on a background mesh as follows:

$$e_{ij}^{\mathcal{M},g,u}(\mathbf{x}_{ij}) = \quad |\mathbf{x}_{ij}^{\mathcal{M}}|^2 \quad \bar{e_{ij}}(\mathbf{x}_{ij}) = \; (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \quad |\mathbf{x}_{ij}|^2 \quad \bar{e_{ij}}(\mathbf{x}_{ij})$$

with:

$$\bar{e_{ij}}(\mathbf{x}_{ij}) = |g_{ij}(\mathbf{x}_{ij})| \, |H_{ij}(\mathbf{x}_{ij})| \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \quad = \quad |g_{ij}(\mathbf{x}_{ij})| \, |H_{ij}(\mathbf{x}_{ij})| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}.$$

We observe that $\bar{e_{ij}}(\mathbf{x}_{ij})$ is $O(1)$ when mesh gets finer. Then this first example of error takes place into the context of (18)(19).

6.2 Goal-oriented error

Let u be the solution of (1) and $u_{\mathcal{M}}$ the discrete solution of (2) where the mesh is an unit mesh for metric $\mathcal{M}$. A typical goal-oriented analysis relies on the minimization of the error $\delta j_{goal}(\mathcal{M})$ committed in the evaluation of the scalar output $j = (g, u)$, error which we write as follows:

$$\delta j_{goal}(\mathcal{M}) = |(g, u - u_{\mathcal{M}})| = |(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u)|. \tag{20}$$

According to the Aubin-Nitsche analysis ([7, 31]), this error is second-order with respect to mesh size. Let us define the discrete adjoint state u^*_{goal}:

$$\forall \psi_{\mathcal{M}} \in \ V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{goal}) = (\psi_{\mathcal{M}}, g). \tag{21}$$

In the sequel, we use a fixed-point in which the adjoint is frozen with respect to the metric $\mathcal{M}$. Injecting (21) in (20) we get:

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u_{\mathcal{M}}, u^*_{goal}) + (g, u - \Pi_{\mathcal{M}} u)$$

and, using (4),

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u, u^*_{goal}) + (f - \Pi_{\mathcal{M}} f, u^*_{goal}) + (g, u - \Pi_{\mathcal{M}} u)$$

thus

$$\delta j_{goal}(\mathcal{M}) \approx |a(\Pi_{\mathcal{M}} u - u, u^*_{goal}) + (f - \Pi_{\mathcal{M}} f, u^*_{goal}) + (g, u - \Pi_{\mathcal{M}} u)|$$

or:

$$\delta j_{goal}(\mathcal{M}) \preceq |a(\Pi_{\mathcal{M}} u - u, u^*_{goal})| + |(f - \Pi_{\mathcal{M}} f, u^*_{goal})| + |g||u - \Pi_{\mathcal{M}} u| \tag{22}$$

The RHS of (22) involves three terms. The *second* and *third terms* give Hessian-like quadratic errors $e_{ij}^{\mathcal{M},u^*_{goal},f}$ and $e_{ij}^{\mathcal{M},g,u}$:

$$\begin{aligned} &|(f - \Pi_{\mathcal{M}} f, u^*_{goal})| + |g||\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}| \\ &\qquad \preceq \ \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_{ij \ni i} \left(e_{ij}^{\mathcal{M},u^*_{goal},f} + e_{ij}^{\mathcal{M},g,u} \right) \\ &\qquad \preceq \ \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_{ij \ni i} \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \quad |\mathbf{x}_{ij}|^2 \left(\bar{e_{ij}}^{u^*_{goal},f} + \bar{e_{ij}}^{g,u} \right) \end{aligned}$$

with

$$\bar{e_{ij}}^{u^*_{goal},f}(\mathbf{x}_{ij}) = |u^*_{goal,ij}| \ |H^f_{ij}| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \quad ; \quad \bar{e_{ij}}^{g,u}(\mathbf{x}_{ij}) = |g_{ij}| \ |H^u_{ij}| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

and

$$u^*_{goal,ij} = u^*_{goal}(\frac{\mathbf{x}_i + \mathbf{x}_j}{2})$$

$$g_{ij} = g(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}) \quad ; \quad H^f_{ij} = H^f(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}) \quad ; \quad H^u_{ij} = H^u(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}).$$

The *first term* of (22)'s RHS is more complex. It can be estimated in a different way from the continuous method presented in [9] and used in [13]. Indeed,

$$\begin{aligned} |a(\Pi_{\mathcal{M}} u - u, u^*_{goal})| &= |\int_\Omega \nabla(\Pi_{\mathcal{M}} u - u) \nabla \Pi_{\mathcal{M}} u^*_{g,\mathcal{M}} \mathrm{d}\mathbf{x}| \\ &\preceq \sum_{\partial T_{mn}} \; | \; [\, \nabla u^*_{goal}|_{T_m} - \nabla u^*_{goal}|_{T_n}] \cdot \mathbf{n}_{mn} \; | \; \int_{\partial T_{mn}} |\Pi_{\mathcal{M}} u - u| \; \mathrm{d}\sigma. \end{aligned} \tag{23}$$

Study of the 2D case. In the 2D case, ∂T_{mn} is exactly an edge ij. We introduce the interpolation error estimate on ij, and its measure . We get from (23):

$$|a(\Pi_{\mathcal{M}} u - u, u^*_{goal})| \preceq \sum_{ij} \kappa_{ij}(u^*_{goal}) \; |\mathbf{x}_{ij}|^3 (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \bar{e}^u_{ij}$$

where the sum is taken over the edges and with, for any edge ij

$$\kappa_{ij}(u^*_{goal}) = | \; [\, \nabla u^*_{goal}|_{T_{ij}} - \nabla u^*_{goal}|_{T_{ji}}] \cdot \mathbf{n}_{ij} \; |$$

in which T_{ij} and T_{ji} are the triangles having ij as common edge and $\mathbf{n}_{ij}$ is the normal to edge ij. We need know to identify the *local intensity of the error term* by comparing the RHS with an integral over the computational domain. This integral is taken as a sum over the diamond cells D_{ij} around each edge ij:

$$|a(\Pi_{\mathcal{M}} u - u, u^*_{goal})| \preceq \sum_{ij} |D_{ij}||D_{ij}|^{-1} \kappa_{ij}(u^*_{goal}) \; |\mathbf{x}_{ij}|^3 (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \bar{e}^u_{ij}$$

which shows that $|D_{ij}|^{-1} \kappa_{ij}(u^*_{goal}) \; |\mathbf{x}_{ij}|^3 (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \bar{e}^u_{ij}$ is the local error intensity. The cellwise error integral then writes:

$$\mathcal{E}^{\mathcal{M},a} = \sum_i \frac{1}{\Gamma(i)} \sum_{ij \ni i} |\mathbf{x}_{ij}|^2 (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \bar{e}^a_{ij}$$

with

$$\bar{e}^a_{ij} = |\mathbf{x}_{ij}||D_{ij}|^{-1} \kappa_{ij}(u^*_{goal}) \bar{e}^u_{ij}.$$

We observe that for a Cartesian mesh of mesh size Δx, term $|\mathbf{x}_{ij}|$ is $O(\Delta x)$, term $|D_{ij}|^{-1}$ is $O(\Delta x)^{-2}$, term $\kappa_{ij}(u^*_{goal})$ is $O(\Delta x)$ (non divided difference of normal gradient), and $\bar{e}^u_{ij}$, which is a directional second derivative, is $O(1)$. The error intensity $\bar{e}^a_{ij}$ is then $O(1)$ when mesh size gets finer.

Study of the 3D case. The intersection ∂T_{mn} of two elements T_m and T_n is a common face with vertices i, j, k and an area $area(mn)$. The following quantity is again known:

$$\kappa_{mn}(u^*_{goal}) = | \left(\nabla u^*_{goal}\right)|_{T_m} \cdot \mathbf{n}_{mn} - \left(\nabla u^*_{goal}\right)|_{T_n} \cdot \mathbf{n}_{mn}|.$$

The remaining expression can be expressed in terms of interpolation errors:

$$\int_{\partial T_{mn}} |\Pi_{\mathcal{M}} u - u| \approx \frac{1}{3} area(mn)(e_{ij}^{\mathcal{M},u} + e_{ik}^{\mathcal{M},u} + e_{kj}^{\mathcal{M},u})$$

with (for $\alpha\beta$=ij,ik and kj):

$$e_{\alpha\beta}^{\mathcal{M},u} = \ (\mathbf{x}_{\alpha\beta}, \mathcal{M}\mathbf{x}_{\alpha\beta})^{-1} \quad |\mathbf{x}_{\alpha\beta}|^2 \ \bar{e}_{\alpha\beta}^{u}$$

and:

$$\bar{e}_{\alpha\beta}^{u}(\mathbf{x}_{\alpha\beta}) = \ |H_{\alpha\beta}^{u}| \cdot \frac{\mathbf{x}_{\alpha\beta}}{|\mathbf{x}_{\alpha\beta}|} \cdot \frac{\mathbf{x}_{\alpha\beta}}{|\mathbf{x}_{\alpha\beta}|}.$$

We get:

$$|a(\Pi_{\mathcal{M}} u - u, u_{goal}^*)| \ \preceq \ \sum_{\bar{D}_{mn}} \frac{area(mn)}{3} \ (e_{ij}^{\mathcal{M},u} + e_{ik}^{\mathcal{M},u} + e_{jk}^{\mathcal{M},u}) \ \kappa_{mn}(u_{goal}^*)$$

Let us convert the RHS into an edge-by-edge sum:

$$|a(\Pi_{\mathcal{M}} u - u, u_{goal}^*)| \ \preceq \ \sum_{\bar{D}_{mn}} \quad \sum_{\alpha\beta=ij,ik,jk} area(mn) \ \frac{1}{3} e_{\alpha\beta}^{\mathcal{M},u} \kappa_{mn}(u_{goal}^*)$$

$$= \ \sum_{\text{edges } ij} \ \sum_{\bar{D}_{mn} \ni ij} area(mn) \ \frac{1}{3} e_{ij}^{\mathcal{M},u} \kappa_{mn}(u_{goal}^*) \ = \ \sum_{\text{edges } ij} \ e_{ij}^{\mathcal{M},a} \ |D_{ij}|$$

where we recognize the edge-by-edge integral of a field $e_{ij}^{\mathcal{M},a}$ defined on edges, with the notation:

$$e_{ij}^{\mathcal{M},a} = \frac{1}{3} \ \frac{1}{|D_{ij}|} \ e_{ij}^{\mathcal{M},u} \ \sum_{\bar{D}_{mn} \ni ij} area(mn) \ \kappa_{mn}(u_{goal}^*). \tag{24}$$

Equivalently (at the second order) we get the(18)(19) format:

$$|a(\Pi_{\mathcal{M}} u - u, u_{goal}^*)| \ \preceq \ \sum_i meas_{\mathbf{x}}(i) \frac{1}{\Gamma(i)} \sum_{ij \ni i} \ e_{ij}^{\mathcal{M},a}.$$

We can then define:

$$\bar{e}_{ij}^{a} = \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \quad |\mathbf{x}_{ij}|^{-2} \ e_{ij}^{\mathcal{M},a} = \frac{1}{3} \ \frac{1}{|D_{ij}|} \ \bar{e}_{ij}^{u} \ \sum_{\bar{D}_{mn} \ni ij} area(mn) \ \kappa_{mn}(u_{goal}^*)$$

which does not depend on $\mathcal{M}$.

Synthesis. Finally, gathering the estimate of the three RHS, we get:

$$\delta j_{goal}(\mathcal{M}) \ \preceq$$
$$\sum_i meas_{\mathbf{x}}(i) \Gamma(i)^{-1} \ \sum_{ij \ni i} \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \quad |\mathbf{x}_{ij}|^2 \left(\bar{e}_{ij}^{a} + \bar{e}_{ij}^{u_{goal}^*, f} + \bar{e}_{ij}^{g,u} \right)$$

which takes place in the context of (18)(19).

Remark: The *a priori estimates* at the starting of this analysis relies on edge-based terms which are essentially products of : $\kappa_{mn}(u^*_{goal})$, a second-order directional derivative, normal to edge in 2D, of the adjoint, times $\bar{e}^u_{ij}$, a second-order directional derivative in edge direction (in 2D). In the analysis proposed in [9] and used in [13], the majoration of the directional adjoint derivative consists in using the largest eigenvalue $\rho(H_{u^*})$ of its Hessian. Further, the demonstration is obtained thanks to the assumption that the mesh stretching is bounded. *In the present study, the second directional derivative of u^*_{goal} is directly taken into account, and gives without any extra assumption a more accurate estimate.*

6.3 Norm-oriented error

The norm-oriented analysis is defined in details in the case of the continuous metric method in [14]. In short, this method focusses on the minimization of the following norm with respect to the mesh $\mathcal{M}$:

$$\delta j(\mathcal{M}) = ||u - u_{\mathcal{M}}||^2_{L^2(\Omega)}. \tag{25}$$

Introducing $g_{\mathcal{M}} = u - u_{\mathcal{M}}$, we get a formulation similar to the goal-oriented formulation:

$$\delta j(\mathcal{M}) = (g_{\mathcal{M}}, u - u_{\mathcal{M}}). \tag{26}$$

But in the practical application $u - u_{\mathcal{M}}$ is not known. We approximate it by a function close to it, which we call a corrector. Let us define:

$$g_{\mathcal{M}} = \bar{u}'_{prio,\mathcal{M}} - (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})$$

in which $\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}$ is a Hessian-based approximation of the interpolation error and in which $\bar{u}'_{prio,\mathcal{M}}$ is the solution of:

$$a(\bar{u}'_{prio,\mathcal{M}}, \phi) =$$
$$\sum_{\partial T_{ij}} (\nabla\phi|_{T_i} - \nabla\phi|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}) \, \mathrm{d}\sigma \; - \; (\phi, \pi_{\mathcal{M}} f_{\mathcal{M}} - f_{\mathcal{M}}). \tag{27}$$

Another example with a RHS evaluated on a two-times finer grid is given in [14].

Let us define the discrete adjoint state u^*_{norm}:

$$\forall \psi_{\mathcal{M}} \in \; V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{norm}) = (\psi_{\mathcal{M}}, g_{\mathcal{M}}). \tag{28}$$

Then, similarly to previous section we shall minimize:

$$\delta j_{norm}(\mathcal{M}) \approx |a(\Pi_{\mathcal{M}} u - u, u^*_{norm}) + (f - \Pi_{\mathcal{M}} f, u^*_{norm}) + (g_{\mathcal{M}}, u - \Pi_{\mathcal{M}} u)|.$$

Turning now to the *tensorial formulation*, we minimize:

$$\mathcal{E}(\mathcal{M}) \; = \; \sum_i meas_{\mathbf{x}}(i) \Gamma(i)^{-1} \sum_{ij \ni i} (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \quad |\mathbf{x}_{ij}|^2 \left(\bar{e_{ij}}^{\mathcal{M},a} + \bar{e}^{u^*_{norm},f}_{ij} + \bar{e}^{g,u}_{ij} \right)$$

with

$$\bar{e_{ij}}^{u^*_{norm},f} = |u^*_{norm,ij}| \ |H^f_{ij}| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

$$\bar{e_{ij}}^{g,u} = |g_{ij}| \ |H^u_{ij}| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

$$\bar{e_{ij}}^{\mathcal{M},a} = \ |\mathbf{x}_{ij}| \ |D_{ij}|^{-1} \kappa_{ij}(u^*_{norm}) \ \bar{e}^u_{ij} \tag{29}$$

and with $\kappa_{mn}(u^*_{norm}) = |\left(\nabla u^*_{norm}\right)|_{T_m} \cdot \mathbf{n}_{mn} - \left(\nabla u^*_{norm}\right)|_{T_n} \cdot \mathbf{n}_{mn}|$. The error intensities $\bar{e_{ij}}^{u^*_{norm},f}, \bar{e_{ij}}^{g,u}, \bar{e_{ij}}^{\mathcal{M},a}$ are $O(1)$ when mesh gets finer. This again takes place in the context of (18)(19).

7. Optimal metric

The purpose is to minimize with respect to the metric for a given number of vertices N a functional of the form:

$$\mathcal{E}(\mathcal{M}) \ = \ \sum_i meas_{\mathbf{x}}(i) \Gamma(i)^{-1} \sum_{\mathbf{x}_{ij}} \ (\mathbf{x}_{ij})^2 (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \ \bar{e}_{ij}$$

which is a discrete model for the L^1 norm of a generic quadratic error. We solve this in two steps as in [26, 27]: first we minimize the functional in a point of the computational domain and get a first property of the optimal solution, second we finish determining the optimum by solving a sub-problem on the whole domain.

7.1 Pointwise optimal metric

The purpose of the pointwise metric optimisation is to look for the optimal stretching of the metric, independantly of mesh density. The number of vertices is fixed. We consider metric $\mathcal{M}_0$ such that the determinant, or product of eigenvalues is equal to unity, i.e. $\lambda_1\lambda_2\lambda_2 = 1$ or, equivalently $det(\mathcal{M}_0) = 1$. We know that:

$$(\mathbf{x}_{ij})^2 \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \ \ \bar{e}_{ij} = e^{\mathcal{M}}_{ij} \ \ \forall j.$$

In that expression, $(\mathbf{x}_{ij})^2$ and $(\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1}$ are not vanishing for any couple of neighboring vertices i and j, which implies

$$e^{\mathcal{M}}_{ij} = 0 \quad \Leftrightarrow \quad \bar{e}_{ij} = 0.$$

Now, for any i and any j belonging to $\Gamma(i)$ such that $\bar{e}_{ij} \neq 0$,

$$(\mathbf{x}_{ij})^{-2} \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \ \ (\bar{e}_{ij})^{-1} = (e^{\mathcal{M}}_{ij})^{-1}.$$

Summing around the vertex i, it gives:

$$\sum_{\substack{j \in \Gamma(i) \\ |\bar{e}_{ij} \neq 0}} (\mathbf{x}_{ij})^{-2} \ (\bar{e}_{ij})^{-1} \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) = \sum_{\substack{j \in \Gamma(i) \\ |\bar{e}_{ij} \neq 0}} (e^{\mathcal{M}}_{ij})^{-1}$$

For the sake of simplicity, let us denote: $D_i = \sum_{\substack{j\in\Gamma(i)\\ |\bar{e}_{ij}\neq 0}} (e_{ij}^{\mathcal{M}})^{-1}$.

We note that each $e_{ij}^{\mathcal{M}}$ is positive and therefore so is D_i. This implies:

$$D_i = \sum_{j\in\Gamma(i)} (\mathcal{M}\ \bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}, \bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}) = \mathcal{M}\ :\ \sum_{j\in\Gamma(i)} \bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}\otimes\bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}.$$

Now, remembering that $A : B = tr({}^tA.B)$, it is interesting to choose (among other solutions):

$$\mathcal{M}^i = \frac{D_i}{dim}\left(\sum_{j\in\Gamma(i)} \bar{e}_{ij}^{-1}|\mathbf{x}_{ij}|^{-2}\mathbf{x}_{ij}\ \otimes\ \mathbf{x}_{ij}\right)^{-1}. \tag{30}$$

The optimal pointwise metric is then defined as:

$$\mathcal{M}_0^i = (det(\mathcal{M}^i))^{-\frac{1}{2}}\mathcal{M}^i. \tag{31}$$

7.2 Global optimal metric

The global optimal metric will be obtained by multiplying the pointwise metric by a scalar field C_i defined on any vertex i and which remains to be determined:

$$\mathcal{M}_{opt}^i = C_i\ \mathcal{M}_0^i.$$

We search $(C_i)_i$ which minimizes

$$err_{L^1} = \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1}\sum_{\mathbf{x}_{ij}}\ (\mathbf{x}_{ij})^2(\mathbf{x}_{ij}, C_i\mathcal{M}_0^i\mathbf{x}_{ij})^{-1}\ \bar{e}_{ij}$$

or

$$err_{L^1} = \sum_i\ \alpha_i\ C_i^{-1}\ \ ;\ \ \text{with}\ \ \alpha_i = meas_{\mathbf{x}}(i)\Gamma(i)^{-1}\sum_{\mathbf{x}_{ij}}\ (\mathbf{x}_{ij})^2(\mathbf{x}_{ij}, \mathcal{M}_0^i\mathbf{x}_{ij})^{-1}\ \bar{e}_{ij}$$

while satisfying to the constraint: $\sum_i meas_{\mathbf{x}}(i)\sqrt{det(C_i\mathcal{M}_0^i)} = N$ or:

$$\sum_i\ \mu_i\ C_i^{\frac{dim}{2}} = N\ \ \text{with}\quad \mu_i\ =\ meas_{\mathbf{x}}(i)\sqrt{det(\mathcal{M}_0^i)}.$$

This can be simply solved by applying the variable change $d_i = \mu_i C_i^{\frac{dim}{2}}$, which gives:

$$\text{Min}\ \ \sum_i \eta_i d_i^{\frac{-2}{dim}}\ \ \text{under the constraint}\ \ \sum_i d_i = N, \tag{32}$$

with $\eta_i = \alpha_i\mu_i^{\frac{2}{dim}}$. The solution of (32) writes:

$$d_i = \left(\sum_j \eta_j^{\frac{dim}{2+dim}}\right)^{-1}\ \eta_i^{\frac{dim}{2+dim}}\ N.$$

Lemma: *The optimal metric is defined by:*

$$\mathcal{M}^i \;=\; C_i\, \mathcal{M}_0^i$$

with

$$\mathcal{M}_0^i \;=\; (det(\mathcal{M}_1^i))^{-\frac{1}{2}}\mathcal{M}_1^i, \quad \mathcal{M}_1^i \;=\; \frac{1}{dim}\left(\sum_{j\in\Gamma(i)} \bar{e}_{ij}^{-1}|\mathbf{x}_{ij}|^{-2}\mathbf{x}_{ij} \;\otimes\; \mathbf{x}_{ij}\right)^{-1},$$

$$C_i = \mu_i^{-\frac{2}{dim}}\left(\sum_j \eta_j^{\frac{dim}{2+dim}}\right)^{-\frac{2}{dim}} \eta_i^{\frac{2}{2+dim}}\; N^{\frac{2}{dim}},$$

$$\eta_i = \alpha_i\mu_i^{\frac{2}{dim}} \quad ; \quad \alpha_i = \frac{meas_{\mathbf{x}}(i)}{\Gamma(i)}\sum_{\mathbf{x}_{ij}} \frac{(\mathbf{x}_{ij})^2}{(\mathbf{x}_{ij},\mathcal{M}_0^i\mathbf{x}_{ij})}\,\bar{e}_{ij} \quad ; \quad \mu_i \;=\; meas_{\mathbf{x}}(i)\sqrt{det(\mathcal{M}_0^i)}.\square$$

8. Numerical examples

The analysis developed in this paper gives a purely discrete answer to the same mesh adaptation problems as in [13] in which the continuous approach were introduced and a series of test cases were presented for its evaluation. Our evaluation of the new method will apply it to recompute these test cases and compare the results with the results of [13]. We refer to [13] for a more detailed presentation of each test case.

8.1 A 2D boundary layer test case

This test case is taken from [20]. We solve the Poisson problem $-\Delta u = f$ in $]0,1[\times]0,1[$ with Dirichlet boundary conditions and a right-hand side f chosen for having:

$$u(x,y) = [1 - e^{-\alpha x} - (1-e^{-\alpha})x]4y(1-y).$$

The coefficient α is chosen equal to 100. The graph of the solution is depicted in Figure 1. We study the 2D boundary layer test case for five different methods: uniformly-refined Full-Multi-Grid (FMG), continuous Hessian-based adaptative FMG, tensorial Hessian-based adaptative FMG, continuous norm-oriented adaptative FMG and tensorial norm-oriented adaptative FMG. We can first compare the meshes obtained with the four different adaptative methods. At the begining, we have the uniform mesh given by Figure 2, right. Using this mesh, we compute an approximate solution and we use it to create an adapted mesh with the four methods:

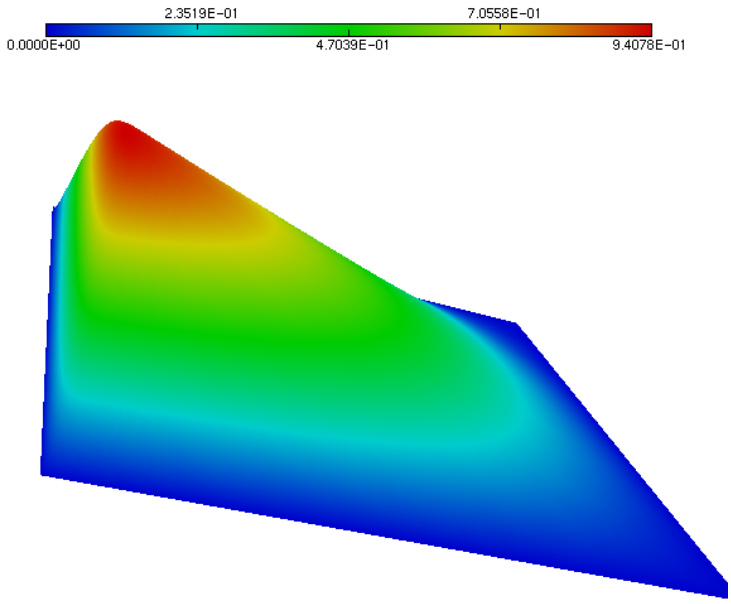

Figure 1: Fully 2D Boundary layer test case : sketch of the solution.

- continuous Hessian-based adaptation gives Figure 2, center,
- tensorial Hessian-based adaptation gives Figure 2, right,
- continuous norm-oriented adaptation gives Figure 3, right,
- tensorial norm-oriented adaptation gives Figure 3, left.

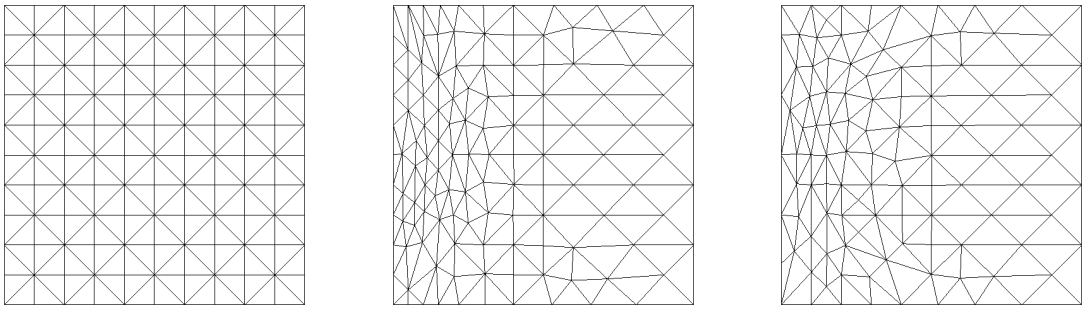

Figure 2: 2D boundary layer test case: initial uniform mesh (left), adapted mesh obtained by *continuous* Hessian-based adaptation (center) and *tensorial* Hessian-based adaptation (right).

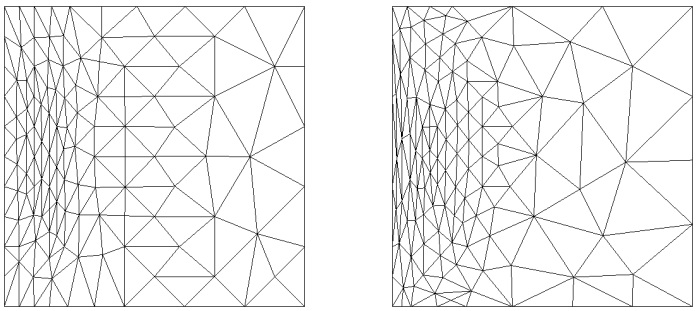

Figure 3: 2D boundary layer test case: adapted mesh obtained with *continuous* norm-oriented adaptation (left)and *tensorial* norm-oriented adaptation (right).

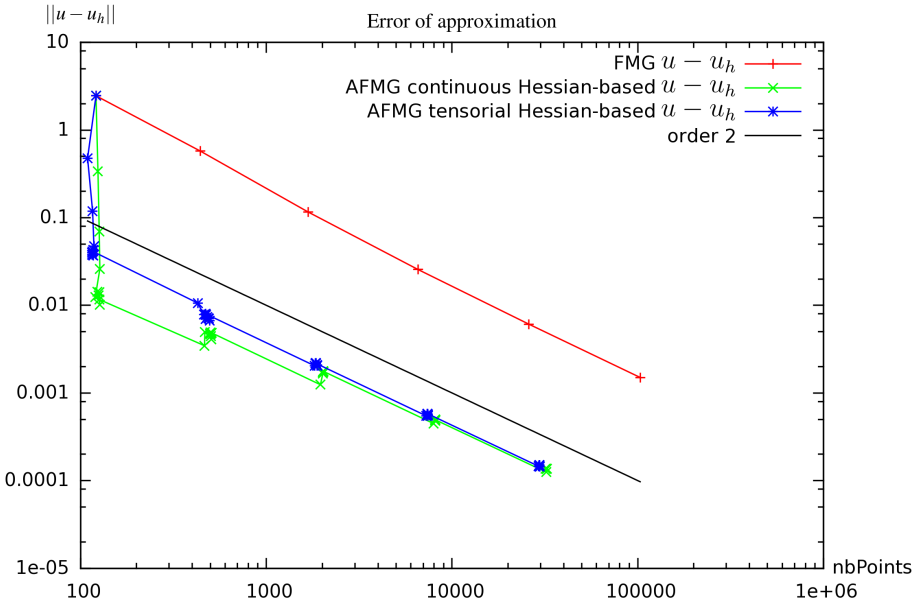

Figure 4: 2D boundary layer test case, Hessian-based methods: error convergence in terms of number of vertices.

We have computed the results for the continuous case and for the tensorial case. For both options, 5 FMG phases corresponding to 5 numbers of nodes, from 128 to $20,000$ are applied. During each FMG phase, the number of nodes is fixed, and 10 mesh adaptations are applied interleaved with a few MG cycles. The approximation error convergence curves of the different methods are depicted in Figures 4 and 5 in function of the number of nodes. We can observe the uniform case in red, the Hessian-based continuous and tensoriel respectively in green and dark blue and the norm-oriented continuous and tensorial respectively in pink and clear blue, the black line being simply the order 2 (legends with symbols are also given in figures). The two Hessian-based cases are very similar and, in

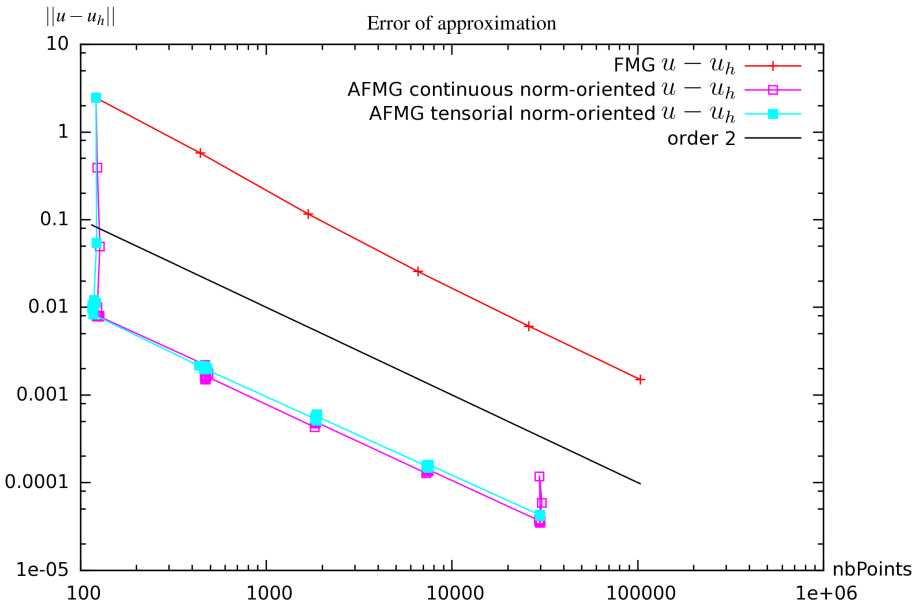

Figure 5: 2D boundary layer test case, norm-oriented methods: approximation error convergence in terms of number of vertices.

the same way, the two norm-oriented cases are very similar too. This tends to indicate that our tensorial method is good, at least for this test case.

8.2 Bubble-like test case with thick interface

We are interested by a Poisson problem the solution of which is a function u equal to 1 on a disk and to 0 in the rest of the domain. This function is the prototype of the pressure in a multi-fluid flow involving capillary forces. The source term is a Dirac derivative. We smooth this computation by defining a thickness ε for defining an annular region separating the two subdomains (outside the disk, inside the disk) and in which u is smoothly varying from 0 to 1: if (x, y) is located inside the annular region, $u(x, y)$ is given by the formula: $u(x, y) = \frac{1}{2} + \frac{1}{2}sin(\frac{\pi\psi}{\varepsilon})$ with $\psi = 0.25 - \sqrt{(x_C - x)^2 + (y_C - y)^2}$. From this solution, a right-hand side f is computed. Given a mesh, vertex values of $f_h(\mathbf{x}_i)$ are prescribed as the analytic values $f(\mathbf{x}_i)$. As a result, for rather coarse meshes, the zone where f is not zero can be simply missed and f_h can be zero even in the neighborhood of the high values of f. We consider first a quite large thickness of $\varepsilon = 0.1$. An approximate solution u_h is shown in as shown in Figure 6. Applying the four above methods give convergence curves which are depicted in Figures 7 and 8. Like in the previous test case, we observe that the tensorial version and the continuous version produce very similar results.

8.3 Bubble-like test case with thin interface

In order to evaluate the robustness of the methods with respect to steeper gradients, we consider the same test case with a thinner transition: $\varepsilon = 0.02$. Figures 9 and 10 give

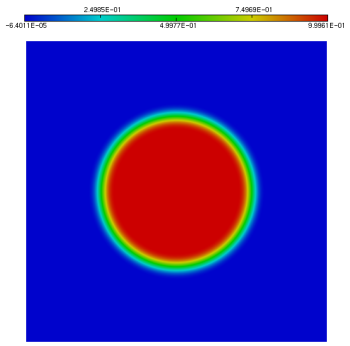

Figure 6: Circular-test-case-domain: sketch of the solution u.

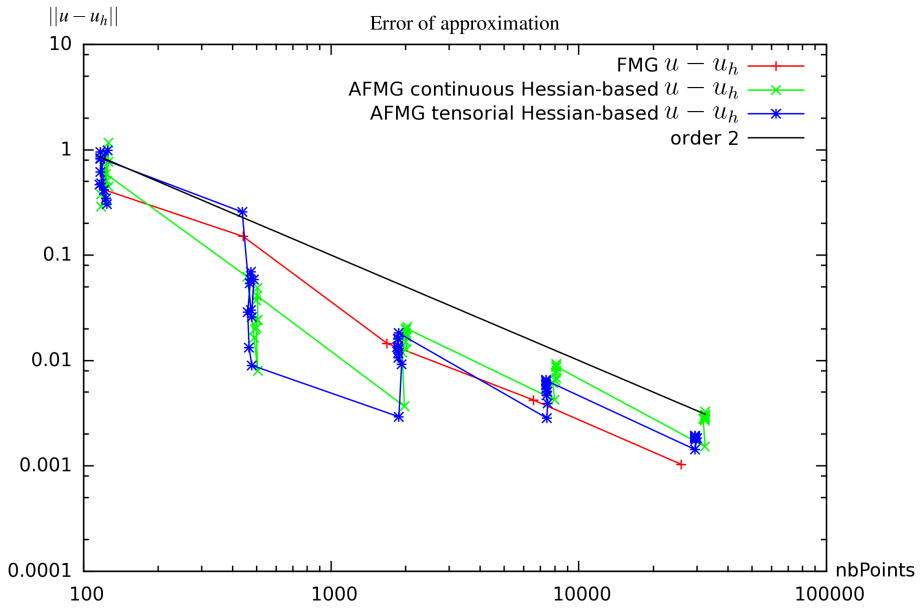

Figure 7: Bubble-like test case with thick interface, Hessian-based methods: approximation error convergence in terms of number of vertices.

us the results. In this case, the tensorial version and the continuous version perform with very similar efficiency. Hessian-based methods give now a notable improvement with respect to uniform refining. Norm-oriented are much better, but adaptation phases appear still rather noisy, since the adaptation stabilizes only after 10 remeshings. For both norm-oriented algorithms, the improvement is of two orders of magnitude with the $30,000$ nodes calculations. Some differences appear when the resulting meshes are compared, see Figures 11. On global mesh views, we observe that the quasi-uniform inner and outer regions contain much more vertices with the tensorial version, in particular close to the boundary. This can be related to the fact that for one case, the non-refined region took about 2000 nodes from the total of $30,000$ while the other option tooks only 700. On

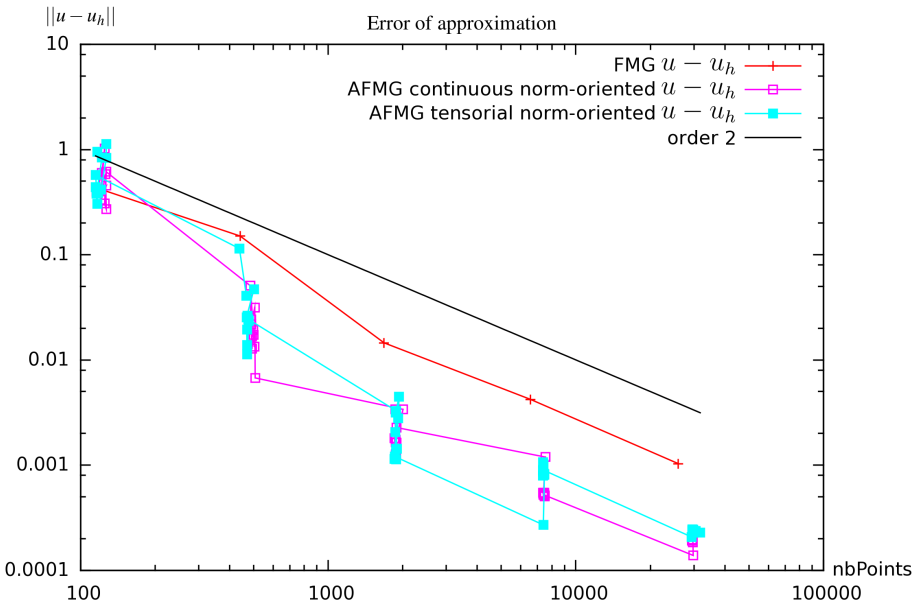

Figure 8: Bubble-like test case with thick interface, norm-oriented methods: approximation error convergence in terms of number of vertices.

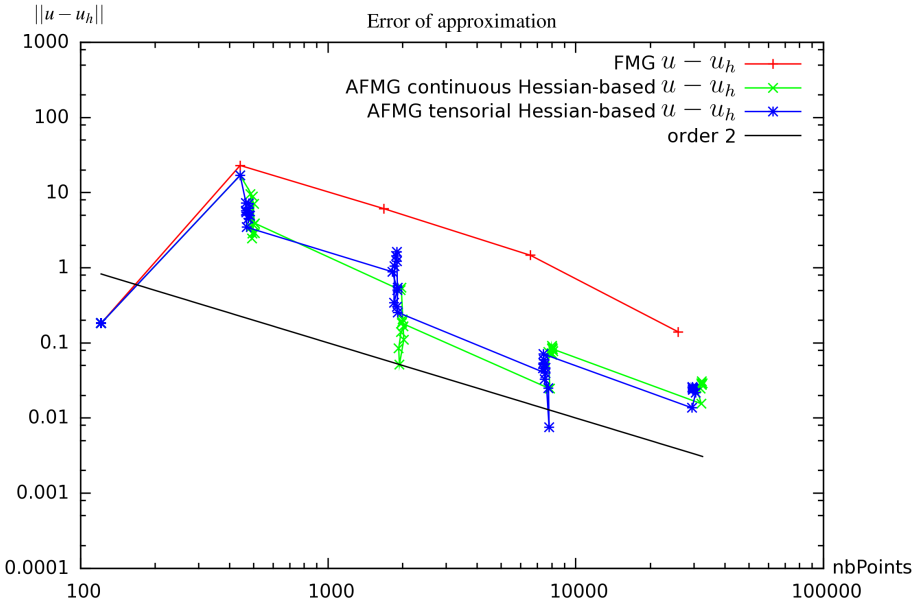

Figure 9: Bubble-like test case with thin interface, Hessian-based methods: approximation error convergence in terms of number of vertices.

the annular region of high variation, the behavior of both method are very similar, and produce stretched meshes with streching ratios both of order 10.

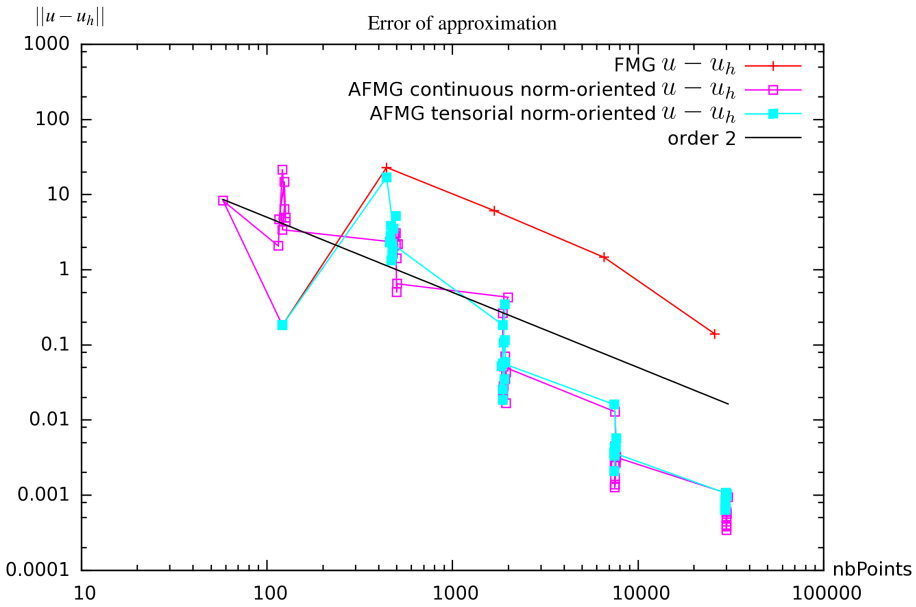

Figure 10: Bubble-like test case with thin interface, norm-oriented methods: approximation error convergence in terms of number of vertices.

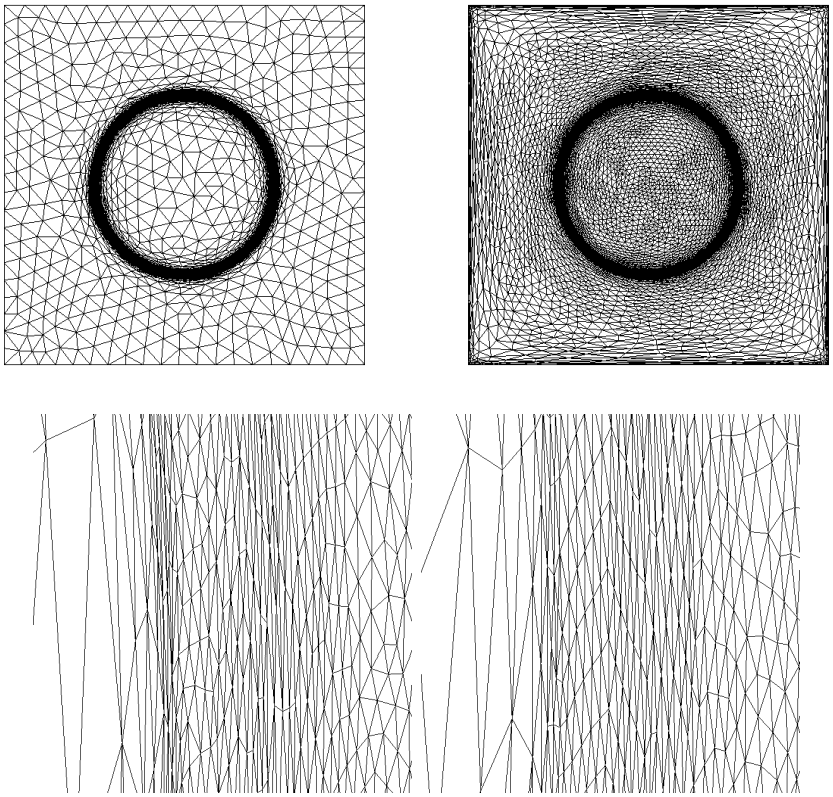

Figure 11: Bubble-like test case with thin interface, norm-oriented methods, sketch of meshes: top, global viewsof continuous option, left and tensorial option, right. Bottom, zooms near the point of discontinuity of maximal abscissa, of continuous option (left) and tensorial option (right).

8.4 Poisson problem with discontinuous coefficient [13]

This test case exemplifies the singularity which is met in the simulation of multi-fluid flows with a large deviation between the densities ρ_1 and ρ_2 of each phase. In the case where a projection algorithm is applied to solve the Navier-Stokes equations for incompressible flow, a Poisson problem with discontinuous coefficients has to be solved. An example can be found in [23]. The present case does not satisfy the smoothness assumptions introduced for deriving our method. However, a usual expectation in mesh adaptation is that the methods should also apply well on non-smooth contexts. We consider the equation of Poisson $-div(\frac{1}{\rho}\nabla u) = rhs$ with a discontinuous coefficient taking two different values $1/\rho_1$ and $1/\rho_2$ on two sub-domains Ω_1 and Ω_2 separated by an interface which is a sufficiently smooth curve for having a normal vector. This PDE is mathematically referred as a transmission problem and the solution is continuous across the interface but of discontinuous normal derivatives since:

$$1/\rho_1 \nabla u_1 \cdot \mathbf{n} = 1/\rho_2 \nabla u_2 \cdot \mathbf{n}$$

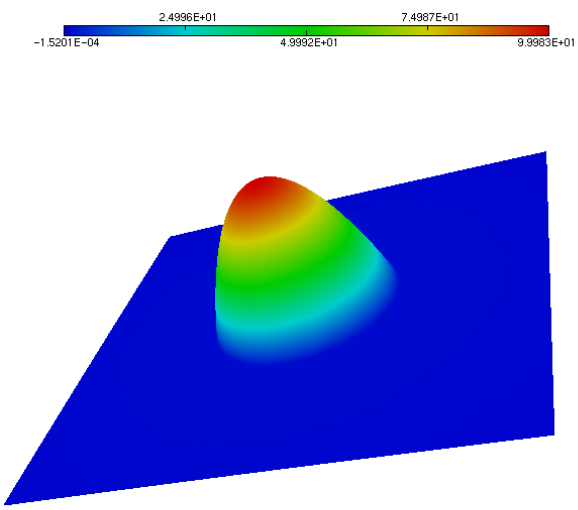

Figure 12: Poisson problem with discontinuous coefficient: view of the solution.

where u_1 and u_2 are the restrictions of the solution u on Ω_1 and Ω_2. In our example, we define them as follows

$$u|_{\Omega_i} = u_i = \alpha_i + \beta_i(x^2 + y^2) \qquad i = 1, 2.$$

Further, Ω_2 is the disk of center $(0.5, 0.5)$ and of radius 0.2 in the computational domain $]0, 1[\times]0, 1[$ and we have:

$$\begin{array}{rcl} 1/\rho_1 = 1000. & ; & \alpha_1 = 1.23579... \;\; ; \;\; \beta_1 = -2.47158... \\ 1/\rho_2 = 1. & ; & \alpha_2 = 100. \qquad\quad ; \;\; \beta_2 = -2471.58... \end{array} \tag{33}$$

This is sketched in Figure 12. In the discrete model, the interface appears only as values of $1/\rho$ evaluated on the vertices of each grid.

Results of Figure 13 are also good but Figure 14 shows results which are disappointing. The two Hessian-based cases and the continuous norm-oriented cases present very good results, of order two. Unfortunately, the tensorial norm-oriented case present a result very different, of order one, whereas it should look very much like the continuous norm-oriented. An examination of the meshes generated and displayed in Figures 15 shows that while the continuous option keeps a good anisotropy in the generated meshes, anisotropy is completely lost by the tensorial option. Note however that in practical applications related to level set calculations, the Heaviside coefficient is generally replaced by a smoother one, see for example [23].

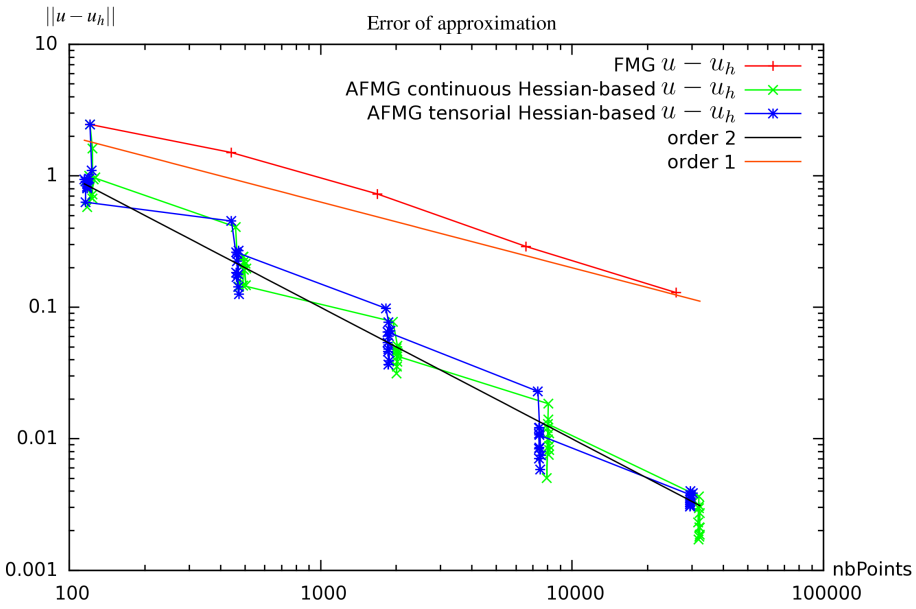

Figure 13: Poisson problem with discontinuous coefficient, Hessian-based methods: approximation error convergence in terms of number of vertices.

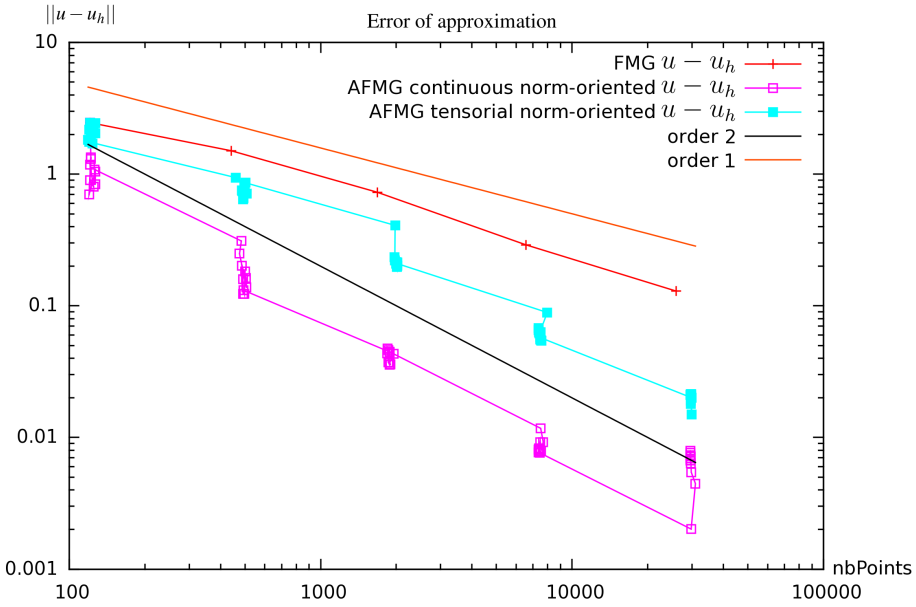

Figure 14: Poisson problem with discontinuous coefficient, norm-oriented methods: approximation error convergence in terms of number of vertices.

8.5 A 1D boundary layer test case

Figures 16 and 17 give us the results in the case of the 1D boundary layer. The two Hessian-based results are similar but we can observe an important difference bewteen the

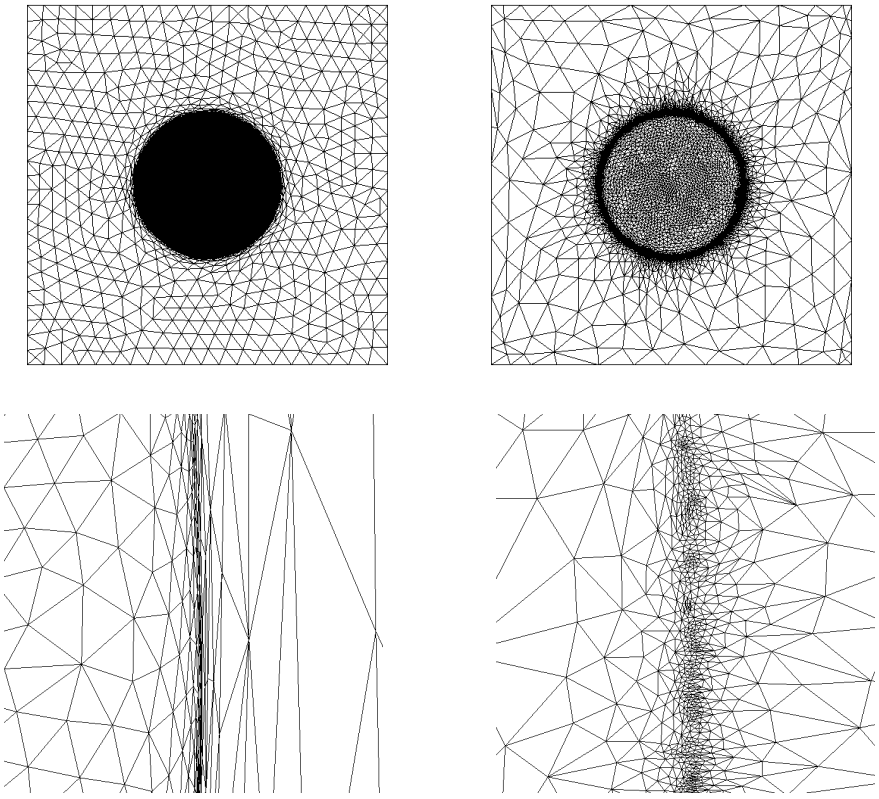

Figure 15: Poisson problem with discontinuous coefficient, sketch of meshes: top, global views of continuous option, left and tensorial option, right. Bottom, zooms near the point of discontinuity of maximal abscissa, of continuous norm oriented option (left) and tensorial norm oriented option, right.

two norm-oriented results. The continuous norm-oriented gives a bad convergence which loses the order two at the end of the computation whereas the tensorial norm-oriented remains of order two. Because of that, the tensorial norm-oriented is better than the continuous norm-oriented.

9. Conclusion

We have proposed several extensions of the discrete tensorial metric method for the metric-based mesh adaptation of a Poisson problem.

The choice of a simplified model, the Poisson equation, allows to analyse in details the different steps in adaptation and to rely on a well-established set of solution-smooothness and approximation-error analyses.

The extensions done here concern first its formulation in terms of an equation defining an intrinseque optimal metric, giving the optimal adapted mesh as a unit mesh of the optimal metric. Second, the method is extended to the minimization of L^p norms. Third, it is extended to anisotropic goal-oriented mesh adaptation. It is also extended to the norm-oriented analysis.

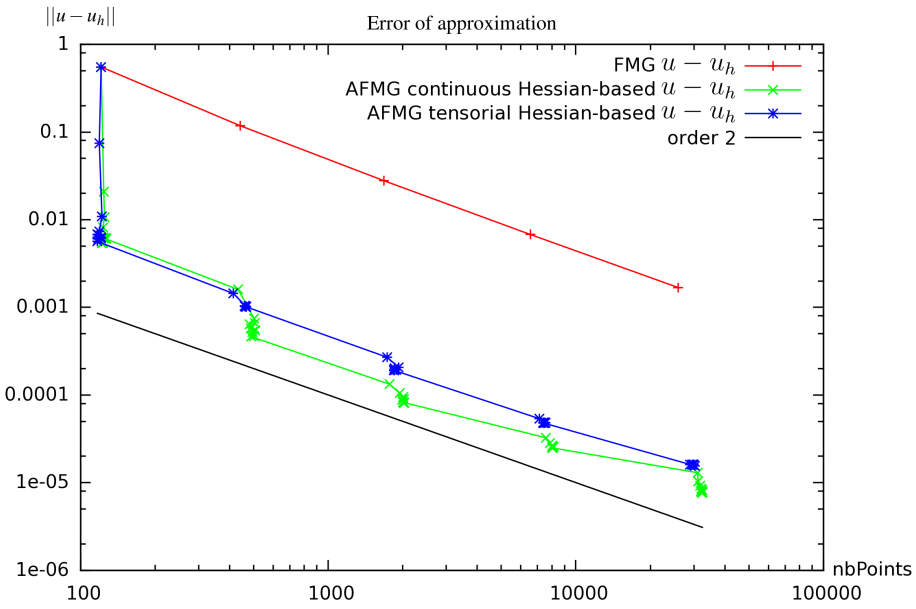

Figure 16: 1D boundary layer, Hessian-based methods: approximation error convergence in terms of number of vertices.

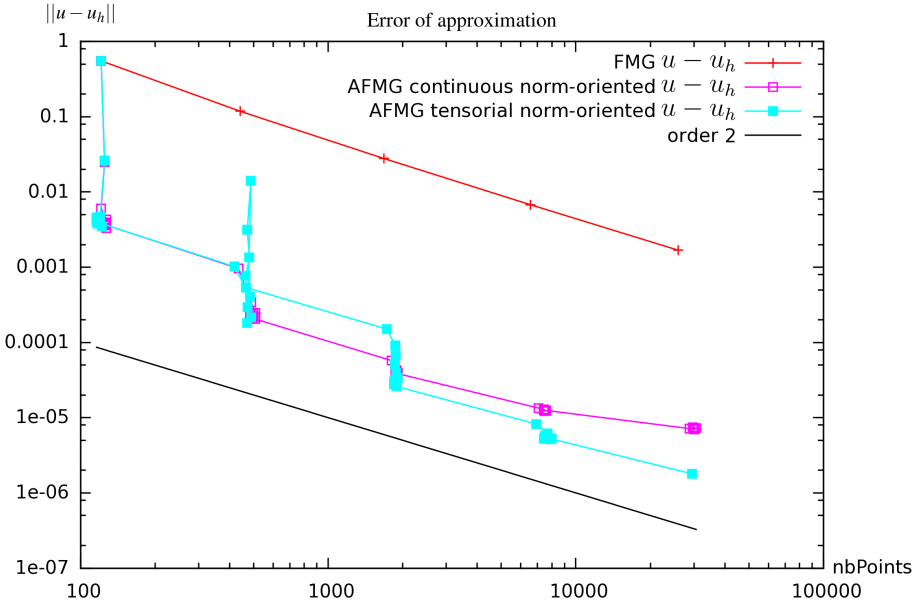

Figure 17: 1D boundary layer, norm-oriented methods: approximation error convergence in terms of number of vertices.

The proposed novel tensorial approach assumes, like the initial tensorial formulations, that the iterated mesh is locally of same edge directions as the background mesh while

the continuous metric never uses this assumption, but this assumption is just a way of reasoning and not a constraint in adaptation. This is illustrated by the fact that the tensorial method produces optimality systems which are essentially discretisations of the optimality systems given by the continuous metric method.

The novel tensorial method shows different features from the continuous metric method. In the continuous metric method, discrete fields are theoretically mapped into a continuous one in order to define a continuous optimality system for the metric. In the tensorial treatment of Hessian-based, goal-oriented, and norm-oriented error analysis, no continuous context needs to be invoked. Further, the error analysis in the tensorial case does not require any anisotropy bound while the continuous analysis does (at least in theoretical arguments).

Two-dimensional numerical experiments on a benchmark already used for continuous Hessian-based, goal-, and norm-oriented adaptation show that both continuous approach and tensorial approach behave similarly on smooth test cases. In particular, both methods produce anisotropic meshes. The tensorial method appears just slightly less smooth than the continuous one. The comparison will in the next future be continued with strongly anisotropic mesh adaptation test cases (shape aspect ratio much larger than 100) by introducing new versions of the mesh generator. In contrast, when applied to a strictly discontinuous context, the tensorial method looses its anisotropy. We have not found yet a simple parameter-free improvement to this defect, and further studies are necessary.

This work also proposes a 3D analysis. 3D experiments will soon be produced.

10. Acknowledgment

This work was partly done in the MAIDESC ANR project which is supported by the French Ministry of Research under contract ANR-13-MONU-0010. The fellowship of Gautier Brèthes is supported by Lemma and région Provence-Alpes-Côte dAzur. We thank Thierry Coupez for fruitful discussions. This work was granted access to the HPC resources of CINES under the allocations 2017-A0022A05067 and 2017- A0022A06386 made by GENCI (Grand Equipement National de Calcul Intensi

References

[1] Absil, P.-A., Mahony, R., & Sepulchre, R. 2008. *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press.

[2] Agouzal, A., Lipnikov, K., & Vasilevskii, Y. 1999. Adaptive generation of quasi-optimal tetrahedral meshes. *East-West Journal*, **7**(4), 223–244.

[3] Alauzet, F. 2003. *Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en Mécanique des Fluides*. Ph.D. thesis, Université Montpellier II, Montpellier, France. (in French).

[4] Alauzet, F., & Loseille, A. 2010. High Order Sonic Boom Modeling by Adaptive Methods. *J. Comp. Phys.*, **229**, 561–593.

[5] Arsigny, V., Fillard, P., Pennec, X., & Ayache, N. 2006. Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors. *Magn. Reson. Med.*, **56**(2), 411–421.

[6] Artina, M., Fornasier, M., Micheletti, S., & Perotto, S. 2013. Benefits of Anisotropic Mesh Adaptation for Brittle Fractures Under Plane-Strain Conditions. *Pages 43–67 of:* S. Perotto, L. Formaggia Eds. (ed), *Proceedings of Tetrahedron IV in New Challenges in Grid Generation and Adaptivity for Scientific Computing.* Springer. Series: SEMA SIMAI Springer, Vol. 5.

[7] Aubin, J. P. 1967. Behaviour of the error of the approximate solution of boundary value problems for linear elliptic operators by Galerkin's and finite difference methods. *Ann. Scuola Norm. Sup. Pisa*, **21**, 599–637.

[8] Becker, R., & Rannacher, R. 1996. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West J. Numer. Math.*, **4**, 237–264.

[9] Belme, A. 2011. *Aérodynamique instationnaire et méthode adjointe.* Ph.D. thesis, Université de Nice Sophia Antipolis, Sophia Antipolis, France. (in French).

[10] Belme, A., Dervieux, A., & Alauzet, F. 2012. Time Accurate Anisotropic Goal-Oriented Mesh Adaptation for Unsteady Flows. *J. Comp. Phys.*, **231**(19), 6323–6348.

[11] Berger, M. 2003. *A panoramic view of Riemannian geometry.* Berlin: Springer Verlag.

[12] Billon, L., Mesri, Y., & Hachem, E. 2016. Anisotropic boundary layer mesh generation for immersed complex geometries. *Engineering With Computers*, 1–12.

[13] Brèthe, G., & Dervieux, A. 2016. Anisotropic Norm-Oriented Mesh Adaptation for a Poisson problem. *J. Comp. Phys.*, **322**, 804 826.

[14] Brèthes, G. 2015. *Algorithmes multigrilles adaptatifs et scalables.* Ph.D. thesis, Université de Nice.

[15] Castro-Díaz, M.J., Hecht, F., Mohammadi, B., & Pironneau, O. 1997. Anisotropic Unstructured Mesh Adaptation for Flow Simulations. *Int. J. Numer. Meth. Fluids*, **25**, 475–491.

[16] Chen, L., Sun, P., & Xu, J. 2007. Optimal anisotropic meshes for minimizing interpolation errors in L^p-norm. *Math. Comp.*, **76**(257), 179–204.

[17] Coupez, T. 2011. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *J. Comp. Phys.*, **230**, 2391–2405.

[18] Coupez, T., Jannoun, G., Nassif, N., Nguyen, H.C., Digonnet, H., & Hachem, E. 2013. Adaptive time-step with anisotropic meshing for incompressible flows. *J. Comp. Phys.*, **241**, 195–211.

[19] Dompierre, J., Vallet, M.G., Fortin, M., Bourgault, Y., & Habashi, W.G. 1997 (Jan). Anisotropic mesh adaptation: towards a solver and user independent CFD. *In: AIAA 35th Aerospace Sciences Meeting and Exhibit.* AIAA-1997-0861, Reno, NV, USA.

[20] Formaggia, L., & Perotto, S. 2003. Anisotropic a priori error estimates for elliptic problems. *Numer. Math.*, **94**, 67–92.

[21] Formaggia, L., Micheletti, S., & Perotto, S. 2004. Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems. *Appl. Numer. Math.*, **51**(4), 511–533.

[22] Gruau, C., & Coupez, T. 2005. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Comput. Methods Appl. Mech. Engrg.*, **194**(48-49), 4951–4976.

[23] Guégan, D., Allain, O., Dervieux, A., & Alauzet, F. 2010. An L^∞-L^p mesh adaptive method for computing unsteady bi-fluid flows. *Int. J. Numer. Meth. Engng*, **84**(11), 1376–1406.

[24] Huang, W. 2005. Metric tensors for anisotropic mesh generation. *J. Comp. Phys.*, **204**(2), 633–665.

[25] Jensen, K.E. 2016. Anisotropic Mesh Adaptation and Topology Optimization in Three Dimensions. *J. Mech. Des.*, **138**(6), 061401.

[26] Loseille, A., & Alauzet, F. 2011a. Continuous mesh framework. Part I: well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, **49**(1), 38–60.

[27] Loseille, A., & Alauzet, F. 2011b. Continuous mesh framework. Part II: validations and applications. *SIAM Journal on Numerical Analysis*, **49**(1), 61–86.

[28] Loseille, A., Dervieux, A., Frey, P.J., & Alauzet, F. 2007 (Jun). Achievement of global second-order mesh convergence for discontinuous flows with adapted unstructured meshes. *In: 37th AIAA Fluid Dynamics Conference and Exhibit.* AIAA-2007-4186, Miami, FL, USA.

[29] Loseille, A., Dervieux, A., & Alauzet, F. 2010 (Jan). A 3D Goal-Oriented Anisotropic Mesh Adaptation Applied to Inviscid Flows in Aeronautics. *In: 48th AIAA Aerospace Sciences Meeting and Exhibit.* AIAA-2010-1067, Orlando, FL, USA.

[30] Loseille, A., Dervieux, A., & Alauzet, F. 2015 (Jan). Anisotropic Norm-Oriented Mesh Adaptation for Compressible Flows. *In: 53rd AIAA Aerospace Sciences Meeting.* AIAA-2015-2037, Kissimmee, Florida.

[31] Nitsche, J. 1968. Ein Kriterium fr die quasi-optimalitat des Ritzschen Verfohrens. *Numer. Math.*, **11**, 346 348.

[32] Vasilevski, Y.V., & Lipnikov, K.N. 1999. An adaptive algorithm for quasi-optimal mesh generation. *Comput. Math. Math. Phys.*, **39**(9), 1468–1486.

[33] Vasilevski, Y.V., & Lipnikov, K.N. 2005. Error bounds for controllable adaptive algorithms based on a Hessian recovery. *Computational Mathematics and Mathematical Physics*, **45**(8), 1374–1384.

[34] Venditti, D.A., & Darmofal, D.L. 2003. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. Comp. Phys.*, **187**(1), 22–46.

[35] Verfürth, R. 2013. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Oxford: Oxford University Press.

[36] Yano, M., & Darmofal, D. 2012. An optimization framework for anisotropic simplex mesh adaptation. *J. Comp. Phys.*, **231**(22), 7626–7649.

[37] Zienkiewicz, O.C., & Zhu, J.Z. 1992. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *Int. J. Numer. Meth. Engng*, **33**(7), 1331–1364.