Journal of Computational Physics 230 (2011) 2391–2405

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing ☆,☆☆

T. Coupez *

Ecole des Mines de Paris, Centre de Mise en Forme des Matériaux (CEMEF), UMR CNRS 7635, Sophia-Antipolis, France

ARTICLE INFO

Article history:
Received 30 March 2010
Received in revised form 25 October 2010
Accepted 26 November 2010
Available online 29 December 2010

Keywords:
Metric
Length distribution tensor
Anisotropic meshing
Interpolation error
Edge error estimate

ABSTRACT

Metric tensors play a key role to control the generation of unstructured anisotropic meshes. In practice, the most well established error analysis enables to calculate a metric tensor on an element basis. In this paper, we propose to build a metric field directly at the nodes of the mesh for a direct use in the meshing tools. First, the unit mesh metric is defined and well justified on a node basis, by using the statistical concept of length distribution tensors. Then, the interpolation error analysis is performed on the projected approximate scalar field along the edges. The error estimate is established on each edge whatever the dimension is. It enables to calculate a stretching factor providing a new edge length distribution, its associated tensor and the corresponding metric. The optimal stretching factor field is obtained by solving an optimization problem under the constraint of a fixed number of edges in the mesh. Several examples of interpolation error are proposed as well as preliminary results of anisotropic adaptation for interface and free surface problem using a level set method.

1. Introduction

Recent years have seen the emergence of anisotropic mesh adaptation techniques with very convincing applications [1–6]. Indeed, such a technique enables to capture scale heterogeneities that can appear in numerous physical or mechanical applications including those having boundary layers, shock waves and edge singularities. In particular, anisotropic adaptive meshing can play a key role in simulation problem involving interfaces. In these cases, discontinuities or gradient of the solution are highly directional and can be captured with a good accuracy by using anisotropic meshing at a low extra number of elements. Contrary to classically used isotropic meshing, anisotropic meshes are more complicated to generate, requiring full control of the shape, size, stretching factor and orientations of elements. Unstructured anisotropic adaptive meshes are most often based on local modifications [7–10], etc. of an existing mesh. In this approach the construction of the mesh in the anisotropic case, is quite easy to implement. Indeed, it mainly requires extending the way to measure lengths following the space directions and can be done by using a metric field to redefine the geometric distances. In parallel, theories of anisotropic a posteriori error estimation [11–14] have been well consolidated, leading to some standardization of the adaptation process, *i.e.*, production of metrics from the error analysis of the discretization error and steering of remeshing by these metrics. It follows that most adaptive anisotropic meshing techniques take a metric map as input. For practical reasons, meshing tools are often demanding a nodal metric map. In fact, during the remeshing operations, the elements are much more volatile

☆ This document is a collaborative effort.
☆☆ The second title footnote which is a longer than the first one and with an intention to fill in up more than one line while formatting.
* Tel.: +33 (0)4 93 95 75 94.
E-mail address: Thierry.Coupez@mines-paristech.fr

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.11.041

than the mesh nodes and therefore defining fields on a continuous basis ease their reconstruction, interpolation or extrapolation. From a theoretical point of view, a continuous metric field could be a direct way to represent the underlying Riemannian space, in which the measurement of length varies at each point and direction [15,1,5]. However, a flat element P^1 can be regarded as a representation of the metric space of the underlying tangent space. Indeed, the metric associated with the transformation of the reference element to the physical element is constant throughout the portion of space covered by it and consequently the representation of space by the field of basic element metric seems somewhat natural. When the metrics are defined at the nodes of the mesh, it is assumed to be an interpolation of the continuous metric associated with the underlying Riemannian space and questions rise about the adequate extrapolation at any point in the domain. Another reason why it must be done, in one way or another, using an automatic metric based mesher driven by a node metric field. We propose in this work a different route for the metric construction. It is done directly at the node of the mesh without any direct information from the element, neither considering any underlying interpolation. It is performed by introducing a statistical concept: the length distribution function. Following the standard idea used in computational rheology to account for orientation [16], we have introduced the second order tensor, approximate of the distribution of lengths defined by gathering the edges at the node. The proposed metric is shown to be the inverse of the tensor, which definition is only depending on edges and can be used in any space dimension.

The second part of this paper is concerned with the interpolation error analysis along the edges. It is shown that once the mesh metric can be defined from the edge length distribution, the remaining task is then to recalculate the length of each edge in its own direction. This can be done by using a simple 1D analysis and it is sufficient to define such metric map based on the length redistribution.

The purpose of this paper is to describe a new theoretical basis giving rise to a numerical technique quite easy to implement. Note also that the proposed analysis is implemented in the context of adaptive meshing under the constraint of a fixed number of nodes, with the advantage to avoid evaluating precisely the constant arising in the error analysis and also to provide a useful tool for practical applications.

The paper is structured as follows. In Section 2, a brief review on the standard element based metric field. Section 3 introduces the node based metric framework and describes the anisotropic mesh adaptation procedure governed by the length distribution tensor. In Section 4, the edge based interpolation error estimate is described and detailed. The error estimate and metric construction are developed in Section 5. Whereas the metric construction using equi-distribution and number constraint are detailed in Section 6. Section 7 provides numerical results and examples to demonstrate the mesh adaptation procedure for given functions, implicit geometry and for a 3D interface capturing using the levelset method.

2. Mesh metric

In this section, we review briefly the definition of the mesh metric field and introduce some useful notations (see Table 1).

A unit metric field can be associated with any unstructured mesh. Indeed, the metric associated with element K can be built using the affine transformation to a reference element which has to be equilateral of edge length equal to unity:

$$\begin{aligned} T_K: \quad & K \to \widehat{K}, \\ & \mathbf{X} \to \widehat{\mathbf{X}} + \mathbb{A}_K(\mathbf{X} - \widehat{\mathbf{X}}). \end{aligned} \tag{1}$$

It means for each edge:

$$1 = |\mathbb{A}_K \mathbf{X}^{ij}| = ({}^t\mathbb{A}_K \mathbb{A}_K \mathbf{X}^{ij}, \mathbf{X}^{ij})^{1/2},$$

Consequently, the unit metric associated with each element can be simply defined by:

$$\mathbb{M}_K = {}^t\mathbb{A}_K \mathbb{A}_K. \tag{2}$$

Such a construction leads to a piecewise constant tensor field and thus discontinuous from element to element. The anisotropic adaptation involves to build a mesh conforming a metric map, which means edges of unit length for the given metric field. It provides both the size and the stretching of elements in a very condensed information data. It is worth mentioning

Table 1
Main notations and definitions.

Notations	Definitions
d	dimension of space
$\mathcal{K}$	set of elements
$\mathcal{N}$	set of nodes
$K \in \mathcal{K}$	mesh element
$\mathbf{X}^i,\ i \in \mathcal{N}$	vector of coordinates for the ith node
$\mathbf{X}^{ij} = \mathbf{X}^j - \mathbf{X}^i$	edge vector made of nodes i and j sharing at least one element
$h_{ij} = \lvert\mathbf{X}^{ij}\rvert$	edge length
$\Gamma(i) = \{j \in \mathcal{N},\ \exists K \in \mathcal{K},\ \mathbf{X}^{ij} \in K\}$	set of nodes connected to node i

that from a meshing point of view, it is easier to store the metric at node than at element, the latter being much more volatile during the remeshing process. This is clearly true for meshers based on local change and in which the metric field is diffused from node to node during derivation [7,8,17]. However, in order to get a continuous metric, the element tensor field must be transferred to the mesh nodes. In this paper we propose a very different way to build the metric field at nodes rather than just averaging the element based metrics, and how to modify the edge length within such a framework. For this purpose we need first to visit again the unit mesh metric definition.

3. Length distribution tensor

We propose in this section a new procedure to construct a metric field directly at the nodes of the mesh. The first step consists in building a continuous natural metric field of a given mesh. In another words, the desired metric field is for which all the edges have a unit length. Moreover, since we want to store this information in a unique tensor defined at the node of the mesh, it is in the averaging process that the method deeply differ from common procedures.

Starting from a given vertex, we examine what information we can construct from the set of edges. Since more than only two edges can be encountered for a node, it is necessary then to find an approximation or an averaging process of the information. For this reason, we first state that the length size of the edges sharing a given node is exactly the interpolation of the continuous length distribution function defined in the space at the considered point.

Let $\mathbf{X}^{ij}$ be the edge vector and $\mathbb{A}^{ij}$ be a transformation changing it into a unit length vector:

$$|\mathbb{A}^{ij}\mathbf{X}^{ij}| = 1 \text{ then for } \mathbb{M} = {}^t\mathbb{A}^{ij}\mathbb{A}^{ij}, \quad \text{we get } (\mathbb{M}\mathbf{X}^{ij}, \mathbf{X}^{ij}) = 1.$$

$\mathbb{M}$ is a positive symmetric tensor but not definite yet. In order to build $\mathbb{M}^i$ such as the edge length of node, i, is almost equal to 1, the immediate solution is to sum up the previous relation:

$$\sum_{j\in\Gamma(i)} (\mathbb{M}^i\mathbf{X}^{ij}, \mathbf{X}^{ij}) = \sum_{j\in\Gamma(i)} 1 \quad \Rightarrow \quad \mathbb{M}^i : \left(\sum_{j\in\Gamma(i)} \mathbf{X}^{ij}\otimes\mathbf{X}^{ij}\right) = |\Gamma(i)|, \tag{3}$$

where $|\Gamma(i)|$ denotes the cardinality of subset $|\Gamma(i)|$. Consequently, the length distribution tensor at node i and denoted as $\mathbb{X}^i$ is defined by:

$$\mathbb{X}^i = \frac{1}{|\Gamma(i)|}\sum_{j\in\Gamma(i)} \mathbf{X}^{ij}\otimes\mathbf{X}^{ij}. \tag{4}$$

Note that, $\mathbb{X}^i$ is a positive symmetric matrix when there exist at least d non aligned edge vectors (see Fig. 1). Under this assumption we can propose the following relation:

$$\mathbb{M}^i = \frac{1}{d}(\mathbb{X}^i)^{-1}. \tag{5}$$

The solution defined by (5) will be considered as the immediate solution. In order to justify the above construction, we introduce in the following the natural solution of an optimization problem, showing also in which sense the immediate solution is a good approximation of the natural one.

Proposition 3.1. *Let* $\{\mathbf{p}^j, j\in\Gamma(i)\}$ *be a sample of vectors and*

$$^2\mathbb{X} = \sum_{j\in\Gamma(i)} \mathbf{p}^j\otimes\mathbf{p}^j,$$

$$^4\mathbb{X} = \sum_{j\in\Gamma(i)} \mathbf{p}^j\otimes\mathbf{p}^j\otimes\mathbf{p}^j\otimes\mathbf{p}^j$$

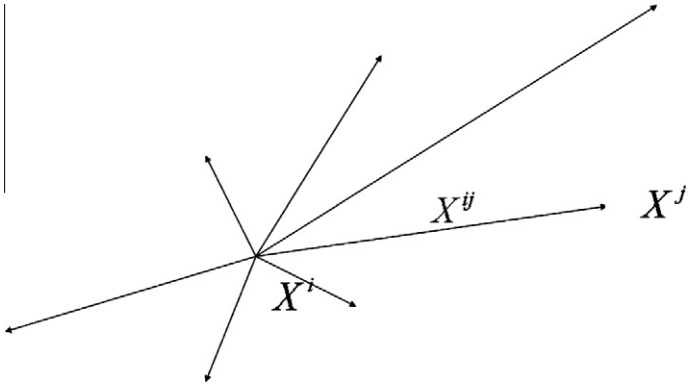

Fig. 1. The symmetric positive definite tensor, $\mathbb{X}^i = \frac{1}{|\Gamma(i)|}\sum_{j\in\Gamma(i)}\mathbf{X}^{ij}\otimes\mathbf{X}^{ij}$, is made from the collected edges at node. This tensor is a statistical representation of length distribution function at a given location..

be respectively the length distribution second and fourth order tensor. Let $\mathbb{M}$ *be solution of*

$$^4\mathbb{X}\ \mathbb{M} = {}^2\mathbb{X} \tag{6}$$

then

$$\mathbb{M} = \arg\min\left(\sum_{j\in\Gamma(i)}((\mathbb{M}\ \mathbf{p}^j,\ \mathbf{p}^j) - 1)^2\right). \tag{7}$$

Proof. By a standard derivation of the functional of (7), we get immediately:

$$\sum_{j\in\Gamma(i)}\left(\left(M_{kl}\ p_k^j,\ p_l^j\right) - 1\right)\ p_m^j\ p_n^j = 0, \quad 1 \leqslant m,n \leqslant d$$

and thus (6). □

Note that Eq. (6) accounts for a unique solution if the fourth order tensor is invertible which requires a large sampling of vectors. The immediate solution is similar to this natural solution when considering a quadratic approximation of the 4th order tensor:

$$^4\mathbb{X} \approx {}^4\widetilde{\mathbb{X}} = {}^2\mathbb{X} \otimes {}^2\mathbb{X} \qquad [18]$$

Indeed:

$$({}^2\mathbb{X} \otimes {}^2\mathbb{X})\ \mathbb{M} = {}^2\mathbb{X}\ ({}^2\mathbb{X} : \mathbb{M}) = {}^2\mathbb{X}.$$

We use the immediate solution in the sequel since it is not only easier to calculate but also the 4th order tensor requires much more edges at node to be invertible. In the sequel, we denote by $\mathbb{X}$ for ${}^2\mathbb{X}$ and $\mathbb{X}^i$ will be the one made of the edges of a node for any dimension. Note that for dimension 1, the edges are the element of a node; for dimension 2, the edges are also the side of triangles and for dimension 3, the edges are shared by numerous tetrahedra.

Remark. If at least d edges are not aligned then $\mathbb{X}^i$ is symmetric definite positive. It only requires one non degenerate element of the node, showing that the above metric construction demands very few mesh regularity.

The following proposition helps to understand the simple hypothesis required in here and thus the efficiency and robustness of the construction.

Proposition 3.2. *Let* $\mathbf{u}$ *and* $\mathbf{v}$ *be two non colinear vectors of* $\mathbb{R}^2$*, and* $\mathbb{M} = (\mathbf{u}\otimes\mathbf{u} + \mathbf{v}\otimes\mathbf{v})^{-1}$ *then:*

$$(\mathbb{M}\ \mathbf{u},\ \mathbf{u}) = 1,$$
$$(\mathbb{M}\ \mathbf{v},\ \mathbf{v}) = 1.$$

Proof. We prove this proposition using a simple straightforward derivation that yields the following:

$$(\mathbb{M}\ \mathbf{u},\ \mathbf{u})\frac{1}{(u_1v_2 - u_2v_1)^2}\begin{pmatrix} u_2^2 + v_2^2 & -(u_1u_2 + v_1v_2) \\ -(u_1u_2 + v_1v_2) & u_1^2 + v_1^2 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}\cdot\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \frac{(u_1v_2 - u_2v_1)^2}{(u_1v_2 - u_2v_1)^2} = 1$$

and both vectors play a symmetric role. □

Conjecture 3.3. *If* $\mathbf{u}^i$*,* $i \in [|1,d|]$ *are* d *independent vectors of* $\mathbb{R}^d$ *then, the tensor defined by*

$$\mathbb{M} = \left(\sum_{i=1}^{d}\mathbf{u}^i\otimes\mathbf{u}^i\right)^{-1}$$

is symmetric definite positive and

$$\left(\mathbb{M}\ \mathbf{u}^i,\ \mathbf{u}^i\right) = 1, \quad 1 \leqslant i \leqslant d.$$

Example 3.1. As shown in Fig. 2, we assume that the extremities of $\mathbf{X}^{ij}$ are located on the circumference of an ellipsoid.

Lemma 3.4. *If the extremity of* $\mathbf{X}^{ij}$ *is located on a unique ellipsoid for any* j *connected to* i *then:*

$$\exists\mathbb{M},\ (\mathbb{M}\mathbf{X}^{ij},\ \mathbf{X}^{ij}) = 1, \quad \forall j \in \Gamma(i).$$

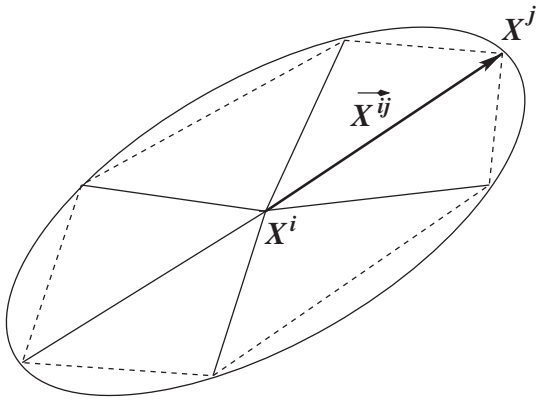

Fig. 2. Edges along an ellipsoid.

Proof. In dimension d, the ellipsoid can be described by the following equation:

$$\sum_{k,l=1}^{d} a_{kl}\left(x_k - X_k^{ij}\right)\left(x_l - X_l^{ij}\right) = 1.$$

The associated metric can be written as:

$$\mathbb{M} = \begin{pmatrix} a_{11} & \cdots & a_{1d} + a_{d1} \\ \vdots & \ddots & \vdots \\ a_{d1} + a_{1d} & \cdots & a_{dd} \end{pmatrix}$$

Thus, $\mathbb{M} : \mathbb{X}^i = 1$ is solution of (7). □

Remark. The difference with the immediate metric may be understood by following the theory of closure approximations [18], however their purpose was to rewrite the fourth order tensor from the second order one only.

4. Edge based interpolation error estimate

4.1. Gradient and strong continuity along the edges

Let u be a regular scalar field in Ω and u_h its interpolation with respect to a given mesh of N nodes such that:

$$u_h(X^i) = u(X^i), \quad 1 \leqslant i \leqslant N.$$

For sake of simplicity, we introduce the following variable change:

$$U^i = u_h(X^i) = u(X^i) \quad \text{and} \quad U^{ij} = U^j - U^i.$$

In the case of continuous element P^1, the gradient of u_h is a piecewise constant field but it is continuous along the edges in the direction of the edges. By introducing the usual approximation space

$$\mathcal{V}_h = \{ v_h \in \mathcal{C}^0(\Omega),\ v_h|_K \in P^1(K) \} \tag{8}$$

we obtain the following:

$$U^j = U^i + \nabla u_h|_K \cdot \mathbf{X}^{ij}$$

and

$$\nabla u_h|_K \cdot \mathbf{X}^{ij} = U^{ij}, \quad \forall K \supset \mathbf{X}^{ij}. \tag{9}$$

The gradient of u_h is a piecewise constant vector field, discontinuous from element to element. But its projection along the edges is continuous since it depends only on the scalar value at the extremities: the nodal values of the field. This is due to the continuity of the approximation whatever the space dimension is. Note that this remains true for higher continuous interpolation of the gradient field along the edges depending only on the values on the edges.

In the following, (9) will be written as:

$$\nabla u_h \cdot \mathbf{X}^{ij} = U^{ij} \tag{10}$$

showing that it does not depends on the shared elements and we define exactly the gradient projection of the interpolated field along the edge simply by the difference of the field value at the edge extremities.

The interpolation error along the edge is of order 2 for P^1 element. More precisely, the strong form is obtained by developing the following proposition:

Proposition 4.1. *Error gradient inequality combined with the above notations imply that:*

$$|(\nabla u_h - \nabla u(X^i)) \cdot \mathbf{X}^{ij}| \leqslant \max_{Y \in [X^i, X^j]} |\mathbb{H}(u)(Y)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|, \tag{11}$$

where $\mathbb{H}(u) = \nabla^{(2)}u$ is the associated Hessian.

Proof. Let $u \in \mathcal{C}^2(\Omega)$ be a smooth scalar field, its projection along $\mathbf{X}^{ij}$ reads the following:

$$u(x) = u(X^i + s\mathbf{X}^{ij}), \quad s \in [0,1]. \tag{12}$$

The mean value theorem implies that:

$$\exists\, A^{ij} \in [X^i, X^j], \quad U^{ij} = \nabla u(A^{ij})\mathbf{X}^{ij}.$$

Thus, from (10) we obtain:

$$\nabla u_h \cdot \mathbf{X}^{ij} = \nabla u(A^{ij})\mathbf{X}^{ij}. \tag{13}$$

Let us write $A^{ij} = X^i + s^{ij}X^{ij}$ with $s \in [0,1]$

Using again the mean value argument, we can state:

$$\exists\, B^{ij} \in [X^i, A^{ij}] \subset [X^i, X^j], (\nabla u(A^{ij}) - \nabla u(X^i)) \cdot \mathbf{X}^{ij} = s^{ij}\mathbb{H}(u)(B^{ij})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \tag{14}$$

and thus:

$$|U^{ij} - \nabla u(X^i) \cdot \mathbf{X}^{ij}| = |(\nabla u_h - \nabla u(X^i)) \cdot \mathbf{X}^{ij}| = |s^{ij}\mathbb{H}(u)(B^{ij})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \tag{15}$$

Consequently:

$$|U^{ij} - \nabla u(X^i) \cdot \mathbf{X}^{ij}| \leqslant \max_{Y \in [X^i, X^j]} |\mathbb{H}(u)(Y)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \tag{16}$$

In the next sections, we will assume the following compact notation:

$$|\mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}| \;\sim\; |s^{ij}\mathbb{H}(u)(B^{ij})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}| \;\sim\; \max_{Y \in [X^i, X^j]} |\mathbb{H}(u)(Y)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|,$$

when there is no ambiguity in the evaluation. □

4.2. Gradient interpolation operator

In this section we introduce an interpolation operator enabling to build a continuous gradient defined directly at the node of the mesh and depending only on the given interpolation values. This simple operator is based again on the length distribution tensor and the projection of the gradient along the edges. This recovery operator would play the role of the Clément interpolant classically used in the error estimation framework [12].

Let us denote by $\mathbf{G}^i$ the proposed gradient reconstruction defined by:

$$\mathbf{G}^i = \arg\min_{\mathbf{G}} \left(\sum_{j \in \Gamma(i)} |(\mathbf{G} - \nabla u_h) \cdot \mathbf{X}^{ij}|^2 \right) = \arg\min_{\mathbf{G}} \left(\sum_{j \in \Gamma(i)} |\mathbf{G} \cdot \mathbf{X}^{ij} - U^{ij}|^2 \right). \tag{17}$$

The solution is obtained by solving:

$$0 = \sum_{j \in \Gamma(i)} (\mathbf{G}^i\mathbf{X}^{ij})\mathbf{X}^{ij} - U^{ij}\mathbf{X}^{ij} = \sum_{j \in \Gamma(i)} (\mathbf{G}^i(\mathbf{X}^{ij} \otimes \mathbf{X}^{ij}) - U^{ij}\mathbf{X}^{ij}). \tag{18}$$

Then, by taking $\mathbf{U}^i = \sum_{j \in \Gamma(i)} U^{ij}\mathbf{X}^{ij}$, we get:

$$\mathbf{G}^i = (\mathbb{X}^i)^{-1}\mathbf{U}^i. \tag{19}$$

Remark. As already noted, the tensor $\mathbb{X}^i$ is positive definite once at least one non degenerate element exists at the node. Consequently $\mathbf{G}^i$ is always defined under this only assumption. Moreover, (19) can be written also by using the natural metric introduced in (5) as follow:

$$\mathbf{G}^i = d\ \mathbb{M}^i \mathbf{U}^i.$$

4.3. A posteriori error analysis

We now take a closer look at the estimates in the neighbourhood of the target node given in the following proposition:

Proposition 4.2. $\forall i \in [|1,N|]$

$$\left(\sum_{j\in\Gamma(i)} |(\mathbf{G}^i - \nabla u)\cdot \mathbf{X}^{ij}|^2\right)^{1/2} \leqslant 2\left(\sum_{j\in\Gamma(i)} |\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}|^2\right)^{1/2}. \tag{20}$$

Proof

$$\sum_{j\in\Gamma(i)} |(\mathbf{G}^i - \nabla u)\cdot \mathbf{X}^{ij}|^2 = \sum_{j\in\Gamma(i)} |\mathbf{G}^i\cdot\mathbf{X}^{ij} - U^{ij} + U^{ij} - \nabla u\cdot\mathbf{X}^{ij}|^2 \leqslant 4\sum_{j\in\Gamma(i)} |U^{ij} - \nabla u\cdot\mathbf{X}^{ij}|^2 \leqslant 4\sum_{j\in\Gamma(i)} |\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}|^2. \quad \square \tag{21}$$

The above result concerns the neighbourhood of the target node. Hereafter, we set the estimate for the edge only.

Proposition 4.3. $\forall i \in [|1,N|]$ *and* $\forall j \in \Gamma(i)$, *we have:*

a) $|(\mathbf{G}^i - \nabla u)\cdot\mathbf{X}^{ij}| \leqslant 2\left(\sum_{k\in\Gamma(i)} |\mathbb{H}u\mathbf{X}^{ik}\cdot\mathbf{X}^{ik}|^2\right)^{1/2}$,

b) $|\mathbf{G}^i\cdot\mathbf{X}^{ij} - U^{ij}| \leqslant 2\left(\sum_{k\in\Gamma(i)} |\mathbb{H}u\mathbf{X}^{ik}\cdot\mathbf{X}^{ik}|^2\right)^{1/2} + |\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}|$,

c) $|\mathbf{G}^{ij}\cdot\mathbf{X}^{ij}| \leqslant 2\left(\sum_{k\in\Gamma(i)} |\mathbb{H}u\mathbf{X}^{ik}\cdot\mathbf{X}^{ij}|^2\right)^{1/2} + 2\left(\sum_{k\in\Gamma(j)} |\mathbb{H}u\mathbf{X}^{jk}\cdot\mathbf{X}^{jk}|^2\right)^{1/2} + |\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}|$,

where $\mathbf{G}^{ij} = \mathbf{G}^j - \mathbf{G}^i$.

Proof. Following the lines in Proposition 4.1 and combining with result of Proposition 4.2 we get:

$$\begin{aligned} &\nabla u(\mathbf{X}^j)\cdot\mathbf{X}^{ij} - \nabla u(\mathbf{X}^i)\cdot\mathbf{X}^{ij} = \mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij} \\ &\qquad\qquad \text{hence} \\ &\underbrace{\nabla u(\mathbf{X}^j)\cdot\mathbf{X}^{ij} - \mathbf{G}^j\cdot\mathbf{X}^{ij}}_{-\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}-\cdots} + \mathbf{G}^{ij}\cdot\mathbf{X}^{ij} + \underbrace{\mathbf{G}^i\cdot\mathbf{X}^{ij} - \nabla u(\mathbf{X}^i)\cdot\mathbf{X}^{ij}}_{+\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}+\cdots} = \mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}. \quad \square \end{aligned} \tag{22}$$

Remark. Looking at the underlying patch associated with this approach we find that the estimate given by b) is only the neighborhood of the node (element connected to the node) and it is the neighborhood of the edge for estimate c). We will use the latter in the sequel. However, we do not prove the reverse estimate: $|\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}| \leqslant |\mathbf{G}^{ij}\cdot\mathbf{X}^{ij}|$.

5. Error estimate and metric construction

We can use these results to develop the error estimate and the metric construction.

5.1. Edge error estimate

We do use the proposed estimate of the projection along the edge of the second derivative:

$$e_{ij} = |\mathbf{G}^{ij}\cdot\mathbf{X}^{ij}|. \tag{23}$$

Note that further work need to be done in order to extend this interpolation error estimate to an approximation error estimate and the following items would be investigated or assumed:

i) $|U_h^{ij} - U^{ij}| \leqslant |\mathbf{G}^{ij}\cdot\mathbf{X}^{ij}|$,
ii) $|\nabla u(\mathbf{X}^j)\cdot\mathbf{X}^{ij} - \nabla u(\mathbf{X}^i)\cdot\mathbf{X}^{ij}| \leqslant |\mathbf{G}^{ij}\cdot\mathbf{X}^{ij}|$,
iii) $|\mathbb{H}u\mathbf{X}^{ij}\cdot\mathbf{X}^{ij}| \leqslant |\mathbf{G}^{ij}\cdot\mathbf{X}^{ij}|$.

We are now in position to do the analysis of the error variation when the length of the edge is changed in the edge direction of a stretching factor denoted by s in the following proposition.

Proposition 5.1. *let s be a stretching factor and consider the edge transformation in its own direction, then:*

$$\begin{aligned} &\mathbf{X}^{ij} \mapsto s\mathbf{X}^{ij}; \quad s \in \mathbb{R}^+, \\ &e_{ij}(s) = |\mathbf{G}^{ij}(s) \cdot s\mathbf{X}^{ij}| \leqslant s^2 e_{ij}. \end{aligned} \tag{24}$$

Proof. We begin to examine the solution along a given edge direction when the extremity varies (the interpolation points).

$$\begin{aligned} &x(s) = \mathbf{X}^i + s\mathbf{X}^{ij}, \\ &u(x(s)) = U^i + s\nabla u(X^i) \cdot \mathbf{X}^{ij} + s^2 \mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \end{aligned} \tag{25}$$

It follows:

$$U^{ij}(s) = s\nabla u(X^i) \cdot \mathbf{X}^{ij} + s^2 \mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \tag{26}$$

Since $U^{ij}(s) = \nabla u_h(s) \cdot \mathbf{X}^{ij}$, we get:

$$|\nabla u_h(s) \cdot \mathbf{X}^{ij} - s\nabla u(X^i) \cdot \mathbf{X}^{ij}| = s^2 |\mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \tag{27}$$

Finally, we obtain:

$$e_{ij}(s) = |\mathbf{G}^{ij}(s) \cdot s\mathbf{X}^{ij}| = |\mathbb{H}u(s) s\mathbf{X}^{ij} \cdot s\mathbf{X}^{ij}| = s^2 |\mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}| = s^2 e_{ij}. \quad \square \tag{28}$$

The quadratic behaviour of the error has been obtained when scaling the edge. It is also an implicit proof of the optimality of the proposed error estimate.

5.2. Edge estimate and element estimate

The objective of this section is to show how the 1D edge error estimate can be brought back to the element error analyis by using the appropriate integration rules.

Let u_h be an approximation of the scalar field u for a given mesh. The edge error reads:

$$\begin{aligned} &u(x(s)) = U^i + s\nabla u(X^i) \cdot \mathbf{X}^{ij} + s^2 \mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}, \\ &u_h(x(s)) = U_h^i + s\nabla u_h \cdot \mathbf{X}^{ij}, \\ &u(x(s)) - u_h(x(s)) = s(\nabla u(X^i) - \nabla u_h) \cdot \mathbf{X}^{ij} + s^2 \mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}, \\ &u(x(s)) - u_h(x(s)) \equiv e(s). \end{aligned} \tag{29}$$

By applying the interpolation $U^i = U_h^i$, it yields that:

$$u(x(1)) - u_h(x(1)) = 0 = \nabla u \cdot \mathbf{X}^{ij} - U^{ij} + \mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \tag{30}$$

Thus,

$$e(s) = s(s-1)\mathbb{H}u\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \tag{31}$$

As a result, we get almost here a polynomial of degree 2. In order to evaluate the L^1 error onto the element, we use an exact integration rule which is based on points located only at nodes and edges, i.e.

$$\int_K p(x) = \sum_i w^i p(\mathbf{X}^i) + \sum_{ij} w^{ij} p\left(\frac{\mathbf{X}^j + \mathbf{X}^i}{2}\right) \tag{32}$$

with $(w^i, w^{ij}) = (0, 1/6)$ for triangles and $(w^i, w^{ij}) = (-1/120, 1/30)$ for tetrahedra.

Proposition 5.2

$$\begin{aligned} &\int_K e(x) = \sum_{i,j=1,\cdots,3} \tfrac{1}{8} |\mathbf{G}^{ij} \cdot \mathbf{X}^i| |K| = \tfrac{1}{8} |K| \sum_{i,j=1,\cdots,3} e^{ij} \qquad \text{in 2D} \\ &\int_K e(x) = \sum_{i,j=1,\cdots,4} \tfrac{1}{30} |\mathbf{G}^{ij} \cdot \mathbf{X}^i| |K| = \tfrac{1}{30} |K| \sum_{i,j=1,\cdots,4} e^{ij} \qquad \text{in 3D.} \end{aligned} \tag{33}$$

Proof. We prove this proposition by induction. The conclusion is obviously true by using (32) and by still assuming that the interpolation error is null at the vertices. $\square$

This shows that the edge error analysis is sufficient to provide at least the L^1 evaluation of the error.

6. Metric construction by equi-distribution and number constraint

6.1. Metric construction

In what follows, we assume that the approximation error is proportional to the interpolation error, allowing using directly the proposed estimate. However, this matter will be the subject of further investigations in a near future.

We begin by setting the following hypothesis:

$$\exists c > 0, \quad \|u - u_h\| < c\|u - \pi_h u\| \quad \text{with} \quad \pi_h u(\mathbf{X}^i) = u(\mathbf{X}^i), \quad \forall i \in \mathcal{N}. \tag{34}$$

For a "good" finite element method, this is correct and the main difficulty is to exhibit precisely the constant c. This depends clearly on the equations to be solved. By assuming that the constant is global over the whole domain, then the interpolation error estimate works correctly if one seeks for the optimal mesh under some convenient other constraints. In this work, we choose to fix the total number of edges. In this case the exact value of the approximation error is not necessary to build the optimal mesh.

In order to obtain a new metric depending on the error analysis, one has to calculate first a new length for each edge and then to use it for rebuilding the length distribution tensor introduced in the first sections.

Let $s_{ij},\ i,\ j \in \mathcal{N}$ be the set of edges scaling (stretching) factors defined by:

$$\begin{aligned} \tilde{e}_{ij} &= s_{ij}^2 e_{ij} \\ |\widetilde{\mathbf{X}}^{ij}| &= s_{ij}|\mathbf{X}^{ij}|, \end{aligned} \tag{35}$$

where $\tilde{e}$ and $|\widetilde{\mathbf{X}}|$ denote respectively the target error and associated edge length.

Theorem 6.1. *Let:*

i) A be a given number of edges of the mesh,
ii) $(e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|)$ *be the calculated error along and in the direction of the edges,*
iii) $p \in [1,d]$ *be an exponent to be defined.*

Then, the continuous metric field defined at the mesh nodes is obtained from:

$$\mathbb{M}^i = \left(\frac{1}{d} \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1}, \tag{36}$$

where

$$s_{ij} = \left(\frac{\lambda}{e_{ij}} \right)^{1/p} \tag{37}$$

and

$$\lambda = \left(\frac{\sum_i \sum_{j \in \Gamma(i)} e_{ij}^{\frac{p}{p+2}}}{A} \right)^{\frac{p+2}{p}} \tag{38}$$

minimizes the error for a fixed number of edges. Moreover, when $s_{ij} = 1$, $\forall i$, $\forall j \in \Gamma(i)$ *then the mesh is optimal.*

Proof. We introduce first $S = s_{ij},\ i,\ j \in \mathcal{N}$ and the following functional:

$$\varphi(S) = \frac{1}{2} \sum_{ij} s_{ij}^2 e_{ij}. \tag{39}$$

We seek for a minimum under the constraint of a fixed number:

$$\min_{S,\ \sum_{i<j} 1 = A} \varphi(S). \tag{40}$$

We rewrite this minimisation problem by introducing the following Lagrangian expression:

$$\mathcal{L}(S,\lambda) = \varphi(S) + \frac{\lambda}{p} \left(\left(\sum_{ij} n_{ij} \right) - A \right), \tag{41}$$

where n_{ij} is the evaluation of the number of created edges in relation with the scaling factor s_{ij}. This number depends on space dimension and must be 1 when $s = 1$.

When scaling the edges of the mesh length by a factor s, the number of edges is multiply by s^{-d} and we choose:

$$n_{ij} = s_{ij}^{-d} \tag{42}$$

then, on one hand, we have:

$$\frac{\partial \mathcal{L}}{\partial S} = 0 \;\Rightarrow\; s_{ij} e_{ij} - \lambda s_{ij}^{-(p+1)} = 0 \;\Rightarrow\; s_{ij} = \left(\frac{\lambda}{e_{ij}}\right)^{\frac{1}{p+2}} \tag{43}$$

and on the other hand, we get:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \;\Rightarrow\; \sum_{ij} \left(\frac{e_{ij}}{\lambda}\right)^{\frac{p}{p+2}} = A \;\Rightarrow\; \lambda^{\frac{p}{p+2}} = \frac{\sum_{ij} e_{ij}^{\frac{p}{p+2}}}{A} \tag{44}$$

Substituting λ by the above expression yields the proposed result. □

6.2. Regularisation and background metric

According to previous proposition, a null error yields an infinite edge which causes a problem in practice and for different applications. It can be fixed by the error analysis of a uniform isotropic mesh under the constraint of a fixed number of edges.

Proposition 6.2. *The edge error defined by:*

$$e_{ij} = \varepsilon |\mathbf{X}^{ij}|^2 \tag{45}$$

is uniform if and only if:

$$|\mathbf{X}^{ij}| = h = C^{st} \tag{46}$$

i.e. if all edges are of same length.

Proof

i) Assuming that $|\mathbf{X}^{ij}| = h$, then it follows:

$$\lambda^{\frac{p}{p+2}} = \frac{1}{A} \sum_{ij} (\varepsilon |\mathbf{X}^{ij}|^2)^{\frac{p}{p+2}} \;\Rightarrow\; \lambda = \varepsilon h^2 \left(\frac{\sum_{ij} 1}{A}\right)^{\frac{p+2}{p}} = \varepsilon h^2 \;\Rightarrow\; s_{ij} = \left(\frac{\varepsilon h^2}{\varepsilon h^2}\right)^{\frac{1}{p+2}} = 1.$$

ii) Let us assume that $e_{ij} = \varepsilon |\mathbf{X}^{ij}|^2$. In order to obtain a uniform mesh for which the edge length is h, the following scaling factor is required:

$$s_{ij} = \frac{h}{|\mathbf{X}^{ij}|}.$$

The assiocated error is equal to:

$$s_{ij}^2 e_{ij} = \left(\frac{h}{|\mathbf{X}^{ij}|}\right)^2 \varepsilon |\mathbf{X}^{ij}|^2 = \varepsilon h^2$$

and it is uniform when $s_{ij} = 1$. □

Remark. The robust definition of error in Theorem 6.1 takes into account the above regularisation according to two forms:

i) Additive form:

$$e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| + \varepsilon |\mathbf{X}^{ij}|^2. \tag{47}$$

ii) Maximal form:

$$\begin{cases} e_{ij} = \max(|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|, \varepsilon_{\min} |\mathbf{X}^{ij}|^2), \\ e_{ij} = \min(|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|, \varepsilon_{\max} |\mathbf{X}^{ij}|^2). \end{cases} \tag{48}$$

7. Applications

For illustration purposes, we present in this section three numerical examples to demonstrate the application of the anisotropic mesh adaptation procedure. In the first example, we use an analytical function to specify the mesh metric field

and to show the quality aspects of the final mesh. In the second example, the mesh is adapted to render the interface of an immersed cubic objects, whereas in the third example, we solve a 3D unsteady two-phase flow using a convected levelset method with dynamical mesh adaptation.

7.1. Test case1

We consider the following functions defined in a unit square/cube domain:

$$\begin{aligned} &a(x) = \tanh\left(E\sin\left(5\frac{\pi}{2}\|X\|\right)\right),\\ &b(x) = \tanh\left(E\sin\left(5\frac{\pi}{2}\|X-C\|\right)\right), \quad X \in [0,1]^d,\\ &C = \begin{pmatrix} 1 \\ \cdot \\ 1 \end{pmatrix},\\ &u(x) = a\circ a(x) + a\circ b(x). \end{aligned} \tag{49}$$

The parameter E varies from 1 to 32. The larger is E, the sharper is the gradient of this function favoring anisotropic meshing. Despite the simplicity of this function, it has characteristics of a very complex situation where the addition of the two contribution mimics interferences between two sources forcing isotropic mesh at the intersecting regions. Figs. 3–5 show the final adapted mesh for different values of E in 2D and 3D and also the qualities of the mesh and the capability of the proposed method to capture the small details.

7.2. Interface capturing

In order to test this interpolation error estimator and the associated metric construction we propose the following example. The idea is to immerse a geometrical object into the mesh. This method, referred as the immersed volume method (IVM) is presented in [19,20] and [4]. A simple mesh adaptation technique was already used in this reference so that the present work can be considered as an extension for such kind of applications. The distance function defined from the immersed boundary is first computed. This function need only to be evaluated at the node of the mesh and it is nothing else that a level set function:

$$\begin{aligned} &\alpha(x) = d(x,\Gamma) = \min_{y\in\Gamma} d(x,y),\\ &\Gamma = x, \alpha(x) = 0, \end{aligned} \tag{50}$$

where $\Gamma \subset \Omega$ being an immersed boundary and the interpolation being given by $\alpha_h(X^i) = \alpha(X^i),\ i \in \mathcal{N}$. The following example are based on the hyperbolic tangent of the distance function as a smooth approximation of the Heaviside function, with

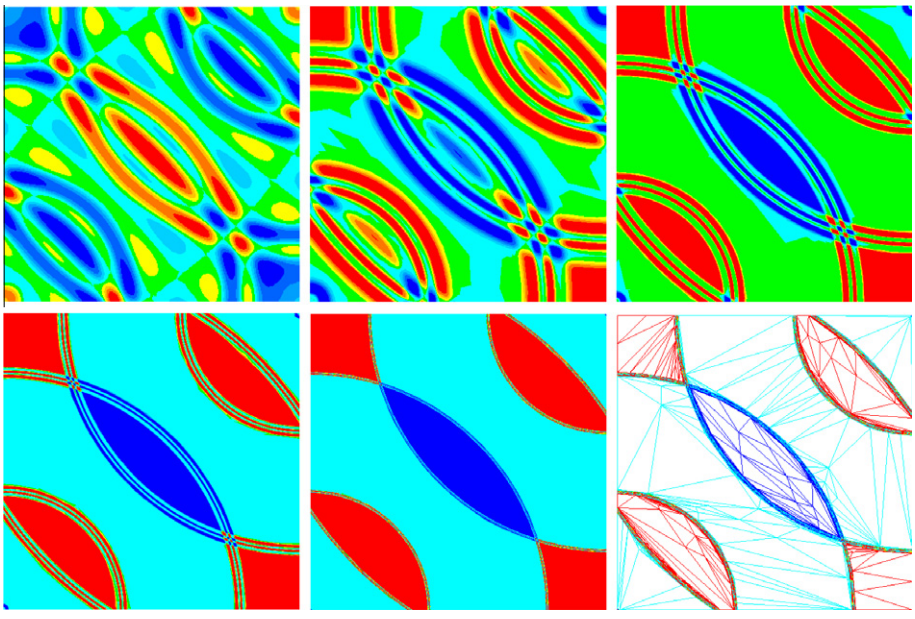

Fig. 3. 2D case: anisotropic mesh obtained for different values of E = 1, 2, 4, 8, 32.

two maxima of the second derivative near the interface. The parameter E enables to control the steepness and the location of the variation of the function.

$$u(x) = E \tanh\left(\frac{\alpha(x)}{E}\right). \tag{51}$$

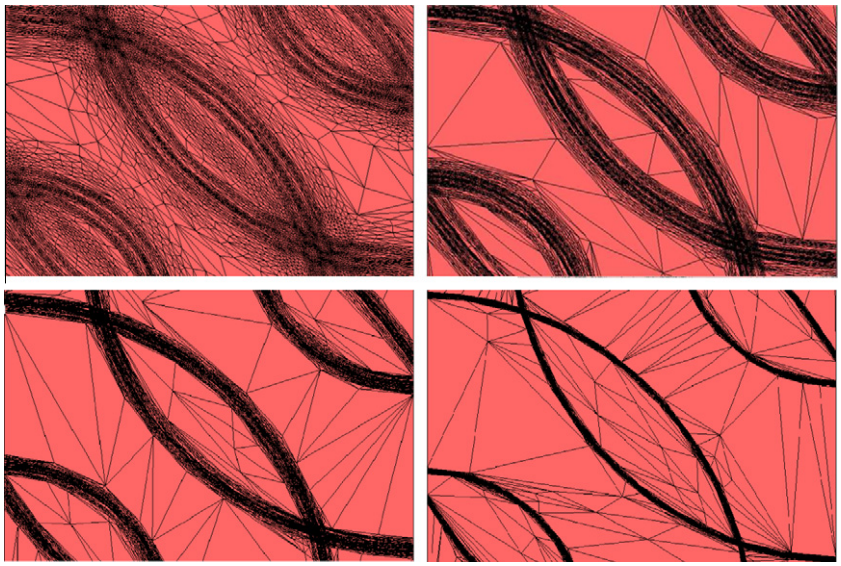

Fig. 4. 2D case: anisotropic mesh obtained for different values of $E = 1, 2, 4, 8, 32$ and for a given fixed number of about 8000 mesh nodes.

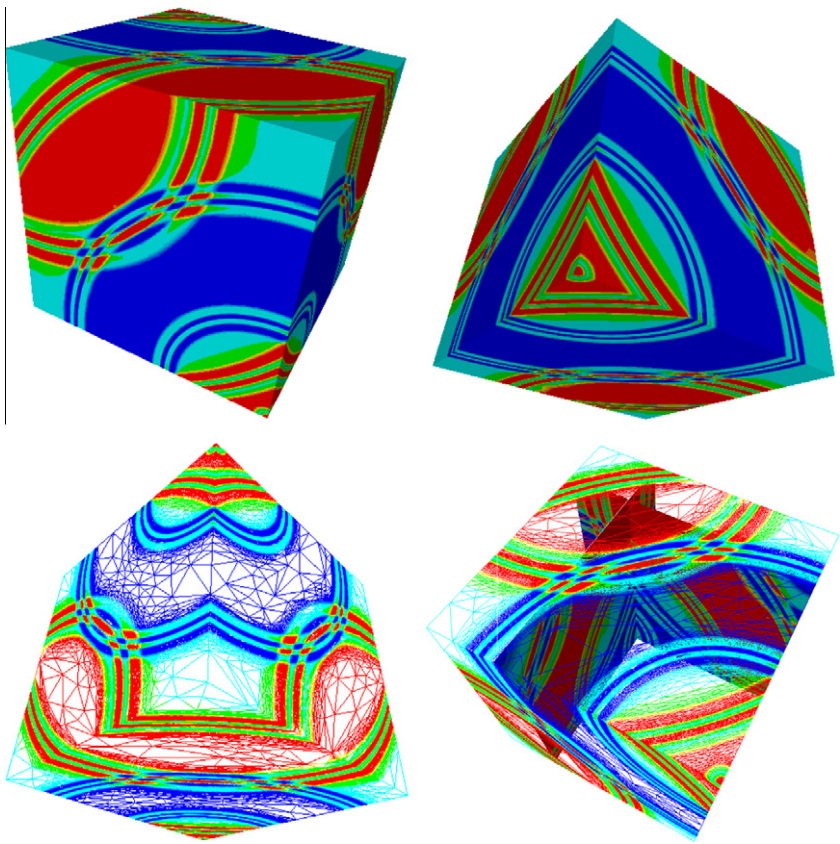

Fig. 5. 3D case: the adapted mesh has about 1 million nodes and it is the 'lighter' possible mesh required just to represent the proposed function.

Remark. In our implementation, the metric calculation requires only loops over nodes and elements. Indeed, we do not need the data structure describing the mesh edges. In fact we assemble the length distribution tensor by summing up the edge vector from element to element. Clearly edges can be counted several times in the averaging process, without theoretical consequences.

Fig. 6 shows the quality of the final anisotropically refined mesh. If the geometrical edges and corners are poorly rendered with the levelset without an important mesh refinement, we can cleary see from Fig. 6 that the implemented method is the appropriate solution for this kind of problems.

7.3. Multiphase calculation by the convected level set method with adaptation

The purpose of this test case is to demonstrate the effectiveness and robustness of the method on a 3D unsteady two-phase flow. A stable/stabilized finite element method is used for both Navier–Stokes [21] and the convected level set [22] equations provided by the CimLib finite element tool box [23]. It is important to mention that the used stabilization coefficients take into account the anisotropy automatically and the element stretching does not cause any difficulty.

We adapt the mesh periodically using the heuristic function defined by (51) as an input. As shown in Figs. 7 and 8, the results seem very realistic rendering of the fluid bucking phenomenon while maintaining an affordable mesh. The mesh adaptation strategy is very crude on this example since the mesh is only adapted periodically without extra control on the time dependency and the extrapolation. The purpose is only to illustrate the proposed approach and to show the possibility in terms of unsteady simulation. In fact such an example is not affordable without mesh adaptation since the filament thinning is important and its position is not predictable a priori. Consequently a uniform mesh would lead to huge numbers of element (several millions as a first evaluation).

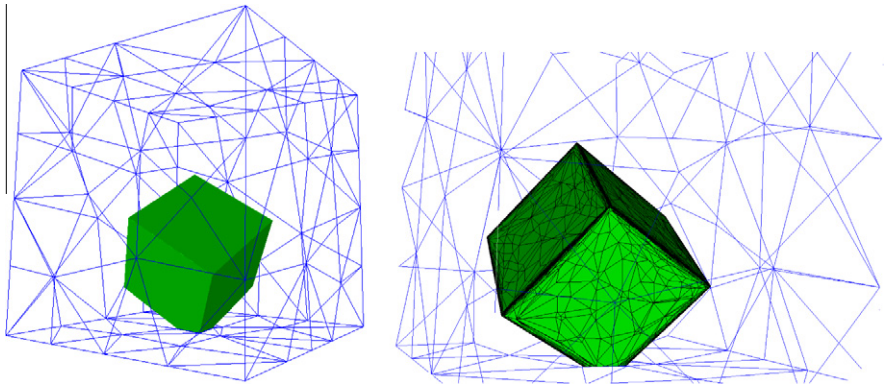

Fig. 6. Anisotropic mesh adaptation for capturing level set representation.

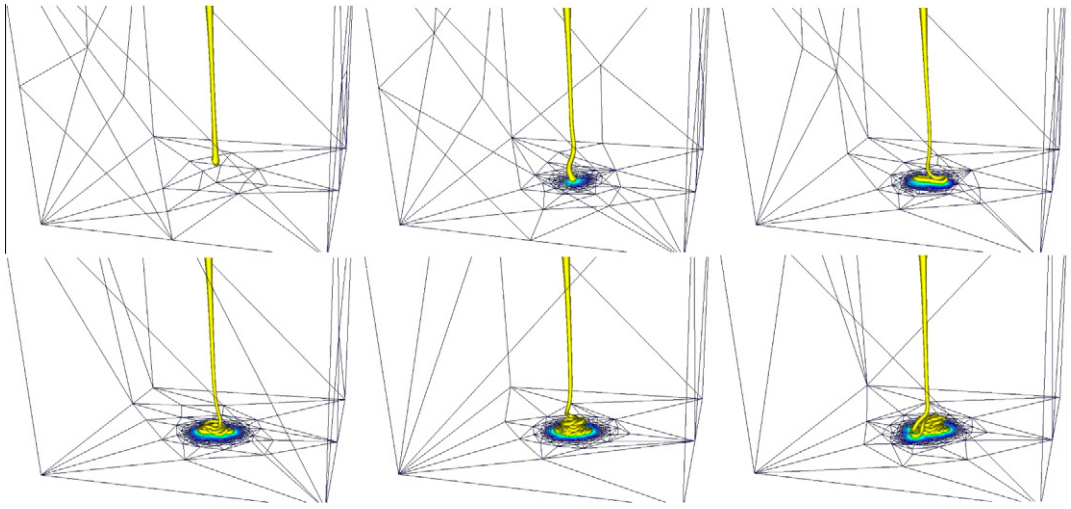

Fig. 7. Fluid buckling simulation by the convected level set method with anisotropic adaptive meshing. Mesher by local mesh topology optimization, metric calculated by the length distribution tensor-objective: a fixed number of nodes (50,000).

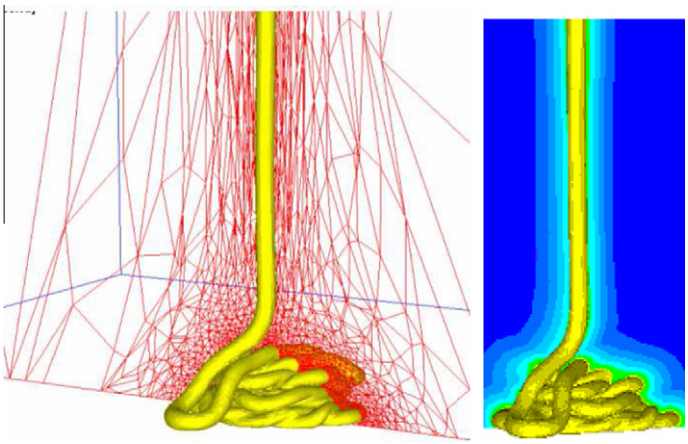

Fig. 8. Fluid buckling simulation, vertical cutting plane. Detail of the 3D mesh trace on the left and the local convected level set function on the right.

8. Conclusion

In this paper we propose a different route to define the metric directly at the node of the mesh. In order to get an averaging procedure that produces a symmetric definite positive tensor we use a statistical descriptor: the length distribution tensor. It is shown that adaptivity is obtained by just redistributing the edge length via this tensor. A 1d error analysis is proven to be sufficient when considering the interpolation error. Several stiff examples of interpolation are proposed showing that the proposed procedure drives correctly the mesher to highly adapted solution. In order to show the efficiency of the method, we propose to apply it directly by connecting the geometrical error to the interpolation error of a level set function. Here again, the error is exact for the static case (representation of an immersed geometry) and becomes heuristic in the dynamic case. The entire process looks quite simple and competes immediately to Hessian based metric and patch recovery procedure. Extension to approximation error would be done by plugging in edge based error or residual estimate.

Acknowledgment

The author gratefully acknowledge support from the CIM research Team at CEMEF.

References

[1] F. Alauset, Adaptation de maillage anisotrope en trois dimension. Applications aux simulations instationnaires en MTcanique des Fluides, Ph.d. thesis, UniversitT de Montpellier II, 2003.
[2] J. Dompierre, M.G. Vallet, M. Fortin, W.G. Habashi, D. Ant-Ali-Yahia, S. Boivin, Y. Bourgault, A.Tam: Edge-based mesh adaptation for CFD. International Conference on Numerical Methods for the Euler and Navier–Stokes Equations, in: Proceedings of Eighth IEEE Symp. pn Parallel and Distributed Processing, Montréal 1995, pp. 265–299.
[3] P.J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Computer Methods in Applied Mechanics and Engineering 194 (48–49) (2005) 5068–5082.
[4] C. Gruau, T. Coupez, 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric, Computer Methods in Applied Mechanics and Engineering 194 (2005) 4951–4976.
[5] A. Loseille, Adaptation de maillage anisotrope 3D multi-Tchelles et ciblTe a une fonctionnelle pour la mTcanique des flluides, Application a la prTdiction haute fidTlitT du bang sonique, Ph.d. thesis, UniversitT Pierre et Marie Curie, 2008.
[6] Y. Mesri, W. Zerguine, H. Digonnet, S. Luisa, T. Coupez, Dynamic parallel mesh adaption for three dimensional unstructured meshes: application to interface tracking, International Meshing Roundtable (2008).
[7] T. Coupez, A mesh improvement method for 3D automatic remeshing, in: Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Pineridge Press, 1994, pp. 615–626.
[8] T. Coupez, Génération de maillage et adaptation de maillage par optimisation locale, Revue européenne des éléments finis 9 (4) (2000) 403–423.
[9] X. Li, M. Shephard, M. Beall, 3D anisotropic mesh adaptation by mesh modification, Computer Methods in Applied Mechanics and Engineering 194 (48-49) (2005) 4915–4950.
[10] J.-F. Remacle, X.M. Shephard, J. Flaherty, Anisotropic adaptive simulation of transient flows, International Journal for Numerical Methods in Engineering 62 (2005) 899–923.
[11] S. Formaggia, L. Perotto, New anisotropic a priori error estimates, Numerical Mathematics 89 (2001) 641–667.
[12] S. Formaggia, L. Perotto, Anisotropic error estimates for elliptic problems, Numerical Mathematics 94 (1) (2003) 67–92.
[13] T. Apel, Anisotropic Finite Elements: Local Estimates and Applications, Teubner, Stuttgart, 1999.
[14] G. Kunert, A Posteriori Error Estimation for Anisotropic Tetrahedral and Triangular Finite Element Meshes, Ph.d. thesis, Fakultätur Mathematik der Technischen Universitä Chemnitz, 1999.
[15] Y. Mesri, F. Alauzet, A. Loseille, L. Hascoet, A. Koobus, B. Dervieux, Continuous metric for computational fluid dynamics 16 (4) (2007) 346–355.
[16] S. Advani, C. Tucker, The use of tensors to describe and predict fiber orientation in short fiber composites, Rheology 31 (8) (1987) 751–784.
[17] T. Coupez, Mesh generation and adaptive remeshing by a local optimisation principle, in: Proceedings of the NAFEMS world congress, 1997, pp. 1051–1060.
[18] S. Advani, C. Tucker, Closure approximation for 3-dimensional structure tensors, Rheology 34 (1990) 367–386.

[19] E. Hachem, Stabilized Finite Element Method for Heat Transfer and Turbulent Flows inside Industrial Furnaces, Ph.d. thesis, Ecole Nationale Supérieure des Mines de Paris, 2009.
[20] E. Hachem, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element solution to handle complex heat and fluid flows in industrial furnace using the immersed volume method, International Journal for Numerical Methods in Fluids (http://dx.doi.org/10.1002/fld.2498).
[21] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element method for incompressible flows with high Reynolds number, Journal of Computational Physics 229 (2010) 8643–8665.
[22] L. Ville, T. Silva, T. Coupez, Convected Level Set method for the numerical simulation of Fluid Buckling, Accepted in the International Journal for Numerical Methods in Fluids.
[23] Y. Mesri, H. Digonnet, T. Coupez, Advanced parallel computing in material forming with CimLib, Accepted in the European Journal of Computational Mechanics.