Anisotropic Delaunay mesh adaptation for unsteady simulations

Dobrzynski C.[1] and Frey P.[2,3]

[1] IMB, UMR 5251, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence cedex, France and INRIA Futurs, Scalapplix team
`cecile.dobrzynski@math.u-bordeaux1.fr`

[2] UPMC Univ Paris 06, UMR 7598, Laboratoire J.L. Lions, F-75005 Paris, France
`frey@ann.jussieu.fr`

[3] Universidad de Chile, UMI 2807, Centro de Modelamiento Matemático, Santiago, Chile

Summary. Anisotropic mesh adaptation is a key feature in many numerical simulations to capture the physical behavior of a complex phenomenon at a reasonable computational cost. It is a challenging problem, especially when dealing with time dependent and interface capturing or tracking problems. Here, we describe an extension of the Delaunay kernel for creating anisotropic mesh elements based on adequate metric tensors. The accuracy and efficiency of the method is assessed on various numerical examples of complex three-dimensional simulations.

1 Introduction

In the context of numerical simulations based on variational methods, adaptive and anisotropic triangulations have proven to be very effective for solving complex physical and biomedical problems described by a set of partial differential equations; see for instance [5, 20, 29, 31]. Actually, many applications (e.g., in solid and fluid dynamics, combustion, heat transfer, etc.) require localized regions of the computational domain to have a larger mesh density, *i.e.* closely spaced nodes, to capture the singular or nearly singular solutions that develop in such regions and to resolve large solution variations sufficiently accurately. Solving these equations with a uniform mesh would require a huge number of mesh nodes, often out of reach of the current computer technology. Indeed, the aim of mesh adaptation is twofold: improving the efficiency of the method for better accuracy and stability at a lower computational expense.

On the other hand, dynamically evolving surfaces arise in numerous computational applications, such as free surfaces in multiphase flows or moving and deforming interfaces in fluid-structure interactions, biomedical surfaces, etc. These applications require or involve potentially large displacements or deformations of the domain geometry in time. Furthermore, this moving geometry is generally part of the solution of a set of partial differential equations and thus it is not known by or described by an analytical model. Such surfaces can be successfully handled by level set formulations [28] or implicit surfaces. In many cases, it is again more efficient to use anisotropic elements and to adapt the mesh to capture the interface or to follow the severe deformations of the geometry. The goal of this paper is to provide such mesh adaptation features for unstructured simplicial meshes in view of time-dependent and geometry evolving simulations.

In general, anisotropic mesh adaptation aims at equidistributing the approximation error by adjusting locally the mesh density according to a metric tensor field based on the Hessian of the numerical solution [18, 19]. It relies on the ability to control the *size*, the *shape* and the *orientation* of the mesh elements. In addition to improving the accuracy of the solution, anisotropy allows to preserve the order of convergence of the computational schemes [20]. It has already been largely shown that highly stretched mesh elements can interpolate a smooth function much more accurately than an isotropic mesh with regular elements [3, 32, 34]. As we will emphasize here, anisotropic elements also have the advantage of introducing regularity in the approximation of interfaces between physical domains.

The contributions of this paper are the following. First, it provides a general context for anisotropic Delaunay-based mesh adaptation in three dimensions. In this respect, it can be considered as an extension of previous works on anisotropic meshing for planar domains. Second, it explains how to build an anisotropic metric tensor for level set interface tracking, following the ideas of [16]. Then, we show how this method can be efficiently used to resolve a moving mesh problem. We demonstrate benchmark and simulation results on rigid-body and large deformations for fluid dynamics simulations. However, we will consider in all the simulations that the initial surface esh is not concerned by adaptation.

The remainder of this paper is organized as follows. Section 2, we review the main issues of anisotropic mesh adaptation based on Riemannian metric tensors. This notion of anisotropic metric tensor has been described in a general purpose book [19] and in many research papers. In particular, we outline the definition and construction of a metric tensor based on error estimates, the notion of metric intersection and interpolation. In Section 3, we show how the classical Delaunay mesh generation procedure can be extended to the anisotropic context. In particular, we show how to compute the Delaunay kernel used for point insertion in the Bowyer-Watson approach to mesh generation. Numerical examples of simulations are given Section 4 to show the efficiency of the approach.

2 Anisotropy and error estimates

As pointed out already [19, 23], anisotropic meshing is closely related to differential geometry concepts and numerical error estimates. We will briefly review this theoretical material (curvature estimates, metric tensor field, Delaunay-based adaptation) for anisotropic meshing as it has already been thoroughly described in various papers [1, 3, 9, 14, 15, 17, 22, 20].

2.1 Basic definitions

We introduce the following notations: Ω denotes a simply connected open bounded domain in $\mathbb{R}^3$, $\overline{\Omega}$ is the closure of Ω and $|\Omega|$ is the d-dimensional measure or the volume of Ω. We suppose given a conforming triangulation T_h on Ω, h representing the characteristic element size. Each element $K \in T_h$ is a closed subdomain of Ω and we assume that $\overline{\Omega} = \bigcup_{K\in T_h} K$ and that the usual finite element requirements are satisfied (*i.e.* non-overlapping nor intersecting elements).

A *uniform mesh* of Ω is then a mesh in which all elements are equally sized and regular. In such case, if $|T_h|$ represents the number of mesh elements and $h_K = diam(K)$ the diameter of K, the size $h = \max\limits_{K\in T_h} h_K$ is given by the relation:

$$h_K \approx \left(\frac{|\Omega|}{|T_h|}\right)^{1/d} \qquad \forall\, K \in T_h\,.$$

A *quasi-uniform* mesh is a mesh for which (i) there exist a constant τ such that

$$\frac{h_K}{\rho_K} \leq \tau \qquad \forall K \in \bigcup_h T_h\,,$$

where, for any open ball $B_i \subset K$, $\rho_K = \sup_i\{diam(B_i)\}$ is the in-diameter of K and (ii) the variation of h is bounded by a constant.

2.2 Metric tensors

Essential to mesh adaptation is the ability to control the *size*, the *shape* and the *orientation* of mesh elements. This specification is usually based on an error estimate or an error indicator. Typically, it uses a matrix-valued field for anisotropic mesh adaptation.

On the continuous level, it is suitable to consider that mesh elements are represented by ellipsoids. In this geometric representation, the size of the element is its volume, the shape is associated with the ratio of the lengths of its semi-axes and the orientation is provided by its principal axis vectors. Then, their control can be achieved by specifiying a metric tensor $M(x)$ to indicated the size, shape and orientation of mesh elements on the whole domain. Here, $M(x)$ is a 3×3 symmetric positive definite matrix, sometimes referred to as *monitor function.* The function $\rho(x) = \sqrt{\det(M(x))}$ is called the *adaptation function.* This metric tensor is used to generate a quasi-uniform mesh of Ω in the metric related to M. The volume of an element $K \in T_h$ is unitary:

$$\int_K \sqrt{det(M(x))}\, dx = \int_K \rho(x)\, dx \;=\; 1\,, \qquad \forall K \in T_h$$

and corresponds to the discrete formulation:

$$|K|\,\sqrt{det(M_K)} \;=1\,, \qquad \forall K \in T_h$$

where M_K is an average of $M(x)$ on K. By extension, the length of a curve γ in a metric M given by $M(x)$ for any $x \in \Omega$ is defined as:

$$|\gamma|_M = \int_0^1 \sqrt{\langle \gamma'(t), M(\gamma(t))\gamma'(t)\rangle} dt\,,$$

where $\gamma(t) : [0,1] \to \mathbb{R}^d$ is a parametrization of γ. By analogy, the length of a mesh edge $\mathbf{e}$ is defined as:

$$l_M(\mathbf{e}) = \int_0^1 \sqrt{\mathbf{e}^t M(t)\mathbf{e}}\, dt\,.$$

Since the metric tensor $M(x)$ is supposed a symmetric positive definite matrix, the spectral decomposition theorem allows to decompose M as:

$$M = P\,\Lambda\,P^t \;=\; \sum_{i=1}^{d} \lambda_i \mathbf{e_i}\mathbf{e_i}^t\,,$$

where the normalized eigenvectors of M are the columns of matrix $P = [\mathbf{e_1}, \ldots, \mathbf{e_d}]$ such that $P\,P^t \;=\; P^t\,P \;=\; I_d$ and Λ is the diagonal matrix of the eigenvalues λ_i. It is obvious to see that the matrix P prescribes the orientation and the matrix Λ prescribes the size and shape of any element K.

Metric intersection

Suppose that two metric tensors are specified at a vertex $p \in T_h$. For mesh generation purposes, we would like to deal with a single metric at the vertex. To this end, we define a metric intersection procedure. Geometrically speaking, it consists in defining the largest ellipsoid E included in the intersection of the two ellipsoids E_1 and E_2 associated with the two metric tensors. From the algebraic point of view, we use the simutaneous reduction of the two underlying quadratic forms to find a basis of the vector space in which the matrices M and N associated with E_1 and E_2 are represented respectively by I_n and D, a diagonal matrix of $M_n(\mathbb{R})$. In other words, the idea is to find a basis (e_1, e_2, e_3) in which M and N are congruent to a diagonal matrix and then to deduce the metric tensor of the intersection.

To achieve this result, the matrix $N' = M^{-1}\,N$ is introduced. The matrix N' is diagonalizable in $\mathbb{R}$ and admits a basis of eigenvectors (orthogonal). In this basis, the terms of the diagonal matrices associated to the metric tensors M and N are obtained using the Rayleigh formula: $\lambda_i = e_i^T\,Me_i$ and $\mu_i = e_i^T\,Ne_i$, $i = 1,3$. Denoting by $P = \begin{pmatrix} e_1 \; e_2 \; e_3 \end{pmatrix}$ the invertible matrix of $GL_3(\mathbb{R})$ formed by the eigenvectors $e_i, i = 1,3$ of the matrix N' leads to define the intersection matrix $M_\cap$ as:

$$M_\cap = (P^T)^{-1} \begin{pmatrix} \max(\lambda_1,\mu_1) & 0 & 0 \\ 0 & \max(\lambda_2,\mu_2) & 0 \\ 0 & 0 & \max(\lambda_3,\mu_3) \end{pmatrix} P^{-1}\,.$$

$M_\cap$ is a symmetric positive definite matrix since $\det M_\cap = \left(P^{-1}\right)^2 \det \Lambda$, where $\Lambda = (\max(\lambda_i,\mu_i))$.

Metric interpolation

Consider a parametrization of a mesh edge pq as $c : [0,1] \to \mathbb{R}^3$, $c(t) = (1-t)p + tq$ and two metric tensors M_p and M_q associated with the endpoints. We are looking for the metric tensor at t, hence for a matrix $M(t)$ defined along the segment $c(t)$ for any value of the parameter $t \in [0,1]$. The definition of this matrice $M(t)$ involves the interpolation of the eigenvalues of the matrices M_p and M_q. This procedure allows to define a continuous metric field along the segment. To this end, we suggest the following linear interpolation scheme:

$$M(t) = \left((1-t)M_p^{-\frac{1}{2}} + tM_q^{-\frac{1}{2}}\right)^{-2}\,, \quad 0 \le t \le 1\,. \tag{1}$$

Finding the interpolated metric tensor M_t requires to express the two matrices in a basis $\{e_i\}$ in which both are congruent to a diagonal matrix and then to deduce the metric tensor at point t. In other words, this scheme is similar to reducing simultaneously the two quadratic forms associated with the metrics.

2.3 Mesh quality measures

There are several reasons for assessing a mesh and controlling its quality. In particular, it is useful to know if the mesh elements are aligned with the physical solution, especially in the anisotropic context. Then, in the adaptive context, it is important to know how closely the equiditribution and alignement conditions are satisfied by the mesh. Finally, in three dimensions, a quasi-uniform mesh is not easy to produce if only controlling the lengths of mesh edges. Well-known *slivers* (null volume) elements may occur that cannot be detected by simply checking the length of edges.

The topics of mesh quality and mesh assessment have been studied in the context of finite element methods in numerous papers. We refer the reader to [2] and the references therein for more details on the related issues. Classical quality measures include minimal angle condition [38] or maximal angle condition [4] as well as aspect ratio considerations. Other works include error estimates to account for the shape of the element and the solution behavior [8]. The adequacy between anisotropic elements and the anisotropy of the solution can be measured by a *matching function* [26].

Given a metric tensor field $M(x)$, it is natural to define critera to measure how closely the mesh elements are aligned and equidistributed with respect to $M(x)$. For practical reasons, we introduce a single measure to evaluate the quality of an element K (given here in dimension d):

$$Q_{ani}(K) = \alpha_d \frac{\left(\displaystyle\sum_{i=1}^{k} \langle \mathbf{e_i}, M_K \mathbf{e_i} \rangle \right)^d}{|K| \sqrt{\det(M_K)}} ,$$

where $\mathbf{e_i}$ represents any of the k egdes of K and α_d is a normalisation constant such that $Q_{ani}(K) = 1$ for a regular element. Notice that $Q_{ani} \geq 1$ for all $K \in T_h$ and thus the larger $\max\limits_K Q_{ani}(K)$ is, the less the triangulation T_h matches the metric specifications.

In addition, we define an *efficiency index* τ that provides a single scalar value to evaluate how well a mesh complies with the metric requirements:

$$\tau = \exp\left(\frac{1}{n_e} \sum_{1 \leq i \leq n_e} \ell_M(\mathbf{e}_i) \right) , \quad \ell_M(\mathbf{e}_i) = \begin{cases} l_M(\mathbf{e}_i) - 1 \text{ if } l_M(\mathbf{e}_i) < 1 \\ l_M^{-1}(\mathbf{e}_i) - 1 \quad \text{else} \end{cases} \tag{2}$$

and n_e denotes the total number of mesh edges.

2.4 Error estimates

Suppose we want to solve the following model problem:
find $u \in W$ *such that* $a(u, v) = f(v)$*, for all* $v \in V$*,*
where V and W are Hilbert spaces, $f \in V'$ and $a \in \mathcal{L}(W \times V, \mathbb{R})$. Suppose also that the bilinear form a satisfies the hypothesis of the Nečas theorem, *i.e.* that this problem is well-posed. Given a mesh T_h of the domain Ω, and approximation spaces W_h and V_h, we consider the numerical problem:
find $u_h \in W_h$ *such that* $a(u_h, v_h) = f(v_h)$*, for all* $v_h \in V_h$*,*
that is also assumed to be well-posed.

A function $e(h, u_h, f)$ is called *a posteriori error* if $\|u - u_h\|_W \leq e(h, u_h, f)$. Furthermore, if $e(h, u_h, f)$ can be written as:

$$e(h, u_h, f) = \left(\sum_{K \in T_h} e_K(u_h, f)^2 \right)^{1/2} ,$$

then $e_K(u_h, f)$ is called an *error indicator*.

Generally speaking, the aim is to obtain an anisotropic bound where the physical derivatives are related to the size, the shape and the orientation of mesh elements. Research has been very active these recent years to develop mathematically-based error estimates and several references are provided in the bibliography section of this paper. From the numerical point of view, it is interesting to obtain estimates for the classical L^1 and L^2 norms or the H^1 seminorm. Such estimates have been provided for the interpolation error on linear Lagrange finite elements and involve the eigenvalues and eigenvectors of the Jacobian matrix of the affine mapping between the reference element and a mesh element [17, 30] or based on the Hessian matrix of the solution [19, 1].

Following this idea, a local error model is defined at a mesh vertex p as:

$$e_M(p) = \sum_{i=1}^{d} h_i^2 \left| \frac{\partial^2 u}{\partial \alpha_i^2} \right| ,$$

where the α_i are the coefficients of the diagonal matrix Λ (see above) and h_i indicate the local sizes in the directions of the eigenvectors of the Hessian matrix. Actually, the local error indicator can be defined in L_∞ norm as:

$$e_K = \|u - P_h u\|_{\infty,K} \ \leq \ C \max_{y \in K} \max_{\mathbf{c} \subset K} \langle \mathbf{v}, |D^2 u(y)| \mathbf{v} \rangle ,$$

where $D^2 u$ represents the hessian matrix of the function u and C is a constant independent of h.

3 Anisotropic Delaunay mesh generation

As indicated, our approach is based on a modification of the classical point insertion procedure in a Delaunay triangulation to acount for anisotropic metric specifications. At first, we briefly recall the Delaunay kernel in the isotropic context and then, we introduce the modifications in the anisotropic context.

3.1 The Delaunay kernel

In the classical isotropic context, mesh points are inserted in an existing Delaunay triangulation using the so-called *Delaunay kernel* [19]. This incremental yet practical method provides an efficient mean for constructing a Delaunay triangulation.

Theorem 1. (Delaunay kernel) *Let T_i be the Delaunay triangulation of the convex hull of a set of point $\mathcal{S} \subset \mathbb{R}^d$ and let $p \notin \mathcal{S}$ be a vertex enclosed in T_i. The* Delaunay kernel *procedure can be written as:*

$$T_{i+1} = T_i \setminus \mathcal{C}(p) + \mathcal{B}(p) , \qquad (3)$$

and provides T_{i+1}, a Delaunay triangulation of the convex hull of $\mathcal{S} \cup \{p\}$.

In this fundamental result, $\mathcal{C}(p)$ stands for the *cavity* of point p: the set of simplices in T_i such that their circumspheres contain point p and $\mathcal{B}(p)$ denotes the *ball* of point p: the set of simplices formed by joining p to the external faces of $\mathcal{C}(p)$ (Figure 1). Practically, this result ensures that the cavity $\mathcal{C}(p)$ is a star-shaped polytope with respect to point p.

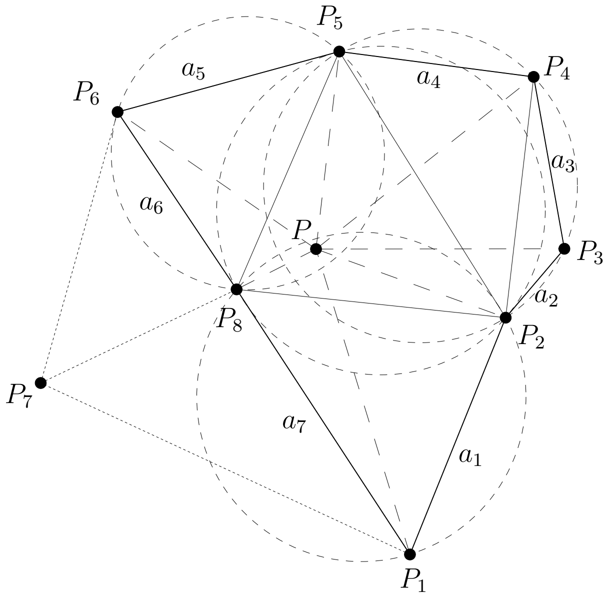

Fig. 1. *Construction of the Delaunay triangulation using the Delaunay kernel (bold lines indicate the external edges of $\mathcal{C}(P)$ in two dimensions).*

However, in numerical simulations, at least two specific problems arise, related to the necessity of:

i) inserting specific entities in the triangulations (a given set of edges defining the domain boundary, for instance) and
ii) creating additional vertices in the triangulation that are not part of the initial set $\mathcal{S}$ (for instance internal vertices during mesh adaptation).

Regarding the last requirement, the following result provides the existence of a triangulation:

Lemma 1. *Let T_i be an arbitrary triangulation and let $p \in \mathbb{R}^d$ be a point enclosed in T_i, p not being a vertex of T_i. Then, a valid conforming triangulation T_{i+1} having p as vertex can be created using the Delaunay kernel, Theorem 1.*

Defining the cavity $\mathcal{C}(p)$ of a point p requires identifying all simplices having a circumsphere that contains point p. The radius of the sphere circumscribed to a simplex K corresponds to the distance between any vertex p_i of

K and the circumcenter O of K: $r_K = d(p_i, O) = \|Op_i\|$. The circumcenter O can be computed as the solution of a linear system of the form:

$$\|Op_1\| = \|Op_i\|\,,\ \forall p_i \in K\,,\, p_i \neq p_1,\ i = 2, \ldots, d\,.$$

Hence, given a point p, an element K belongs to the cavity $\mathcal{C}(p)$ iif the following relation holds:

$$\alpha(p, K) = \frac{\|Op\|}{r_K} < 1\,,$$

$\alpha(p, K)$ being called the *Delaunay measure* of point p with respect to the simplex K.

Inserting a point p in a given (Delaunay) triangulation T_h requires at first to identify all simplices that belong to the cavity $\mathcal{C}(p)$. This can be achieved by finding the simplex or, in some peculiar cases, the set $\{K \in T_h\,, p \in K\}$ of all simplices containing p. Then, using adjacency relationships, it is possible to enumerate the set $\mathcal{C}(p)$. Although Theorem 1 ensures that the cavity $\mathcal{C}(p)$ is star-shaped polytope with respect to point p, this must be check and eventually enforced numerically.

3.2 Anisotropic point insertion

The extension to the anisotropic case consists in introducing a metric tensor, hence a symmetric positive definite matrix M_p at each point $p \in \mathbb{R}^d$. This will allow us to consider the Euclidean norm of any vector in $\mathbb{R}^d$ given the inner product $\langle \cdot, \cdot \rangle_M$. Hence, all distance checks involved in the Delaunay measure will be replaced by length checks according to the given matrix M_p, namely:

$$\alpha_{M_p}(p, K) = \frac{\|Op\|_{M_P}}{(r_K)_{M_p}} < 1\,.$$

Obviously, taking into account only the metric tensor at the given point p is not sufficient and accurate in many applications. It is therefore highly recommended to consider all matrices related to the vertices of element K [21]. By doing so however, leads to solving a non-linear system of equations; overcoming this problem may be tedious and we propose the following answer. The center O_K of the sphere circumscribed to tetrahedra K is the solution of the following system:

$$\begin{cases} l_{M_p}(O_K, p_1) = l_{M_p}(O_K, p_2) \\ l_{M_p}(O_K, p_1) = l_{M_p}(O_K, p_3) \\ l_{M_p}(O_K, p_1) = l_{M_p}(O_K, p_4) \end{cases}$$

with $(p_i)_{i=1..4}$ denote the vertices of K. Then, the length of an edge $O_K p_i$ in the metric M_p is given by:

$$\begin{aligned} l_{M_p}(O_K, p_i) = {} & 2\, m_{12}\,(x_i - O_x)(y_i - O_y) + m_{11}\,(x_i - O_x)^2 \\ & + 2\, m_{13}\,(x_i - O_x)(z_i - O_z) + m_{22}\,(y_i - O_y)^2 \\ & + 2\, m_{23}\,(y_i - O_y)(z_i - O_z) + m_{33}\,(z_i - O_z)^2, \end{aligned}$$

where $p_i = (x_i, y_i, z_i)^t$, $O_K = (O_x, O_y, O_z)^t$ and the metric at point p is defined as $M_p = (m_{ij})_{1\leq i,j\leq 3}$. Finding the center O_K simply leads to solving the following linear system:

$$\begin{pmatrix} a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{pmatrix} \begin{pmatrix} O_x \\ O_y \\ O_z \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix},$$

with the coefficients:

$$\begin{aligned} a_i &= 2\,(m_{11}\,(x_i - x_1) + m_{12}\,(y_i - y_1) + m_{13}\,(z_i - z_1)), \\ b_i &= 2\,(m_{22}\,(y_i - y_1) + m_{12}\,(x_i - x_1) + m_{23}\,(z_i - z_1)), \\ c_i &= 2\,(m_{33}\,(z_i - z_1) + m_{13}\,(x_i - x_1) + m_{23}\,(y_i - y_1)), \\ d_i &= m_{11}\,x_i^2 + 2\,m_{12}\,x_i\,y_i + 2\,m_{13}\,x_i\,z_i + m_{22}\,y_i^2 + 2\,m_{23}\,y_i\,z_i + m_{33}\,z_i^2 \\ &\quad -(m_{11}\,x_1^2 + 2\,m_{12}\,x_1\,y_1 + 2\,m_{13}\,x_1\,z_1 + m_{22}\,y_1^2 + 2\,m_{23}\,y_1\,z_1 + m_{33}\,z_1^2). \end{aligned}$$

In two dimensions, we have the following result:

Lemma 2. *The Delaunay measure, $\alpha_{M_p}(p, K) < 1$, provides a valid anisotropic Delaunay kernel.*

The proof consists in checking that the so-defined cavity is a star-shaped polygon with respect to point P [21].

From the numerical point of view, it can be of interest to take into account the metric tensors at all element vertices M_{p_i} to define the cavity. Finally, we hae found practical and efficient to assess the cavity with two inequalities:

$$\begin{cases} \alpha_{M_p}(p, K) < 1, \\ \displaystyle\sum_{i=1}^{4} \alpha_{M_{p_i}}(p, K) + \alpha_{M_p}(p, K) < 5. \end{cases} \tag{4}$$

3.3 Local mesh adaptation

Adaptive meshing methods belong to two categories depending on whether they proceed by global or local remeshing of the computational domain at each iteration. Global remeshing techniques consist in constructing a new mesh of the domain at each iteration, to ensure that the elements are in good agreement with the anisotropic metric-related prescriptions. The latter are supplied at the vertices of the previous mesh that is then acting as a control space. Obviously, the order of complexity of the meshing method remains the same throughout the whole adaptation scheme. One disadvantage of this technique for steady-state adaptative simulations is that the amount of modifications shall decrease with the iterations since a fixed point of the pair (mesh,solution) is targeted. In other words, once mesh features have been identified and captured by adjusting the local node density, numerical accuracy is only a matter of introducing a few more nodes in critical regions while most of the mesh is kept unchanged.

In our approach, we rely on local mesh modifications. Vertex insertion is handled using the anisotropic extension of the Delaunay kernel described hereabove. The description of the other operators that are involved can be found in the book [19]. Their extension to the anisotropic case is straightforward. Let us simply mention here that we use a combination of edge flips, edge collapsing, node relocation and vertex insertion operations, all driven by the anisotropic metric specifications.

4 Application examples

We provide here a set of numerical examples to assess the proposed approach.

4.1 Analytical metric

At first, we prescribe the size, shape and orientation of the mesh elements using the following analytical metric:

$$M = \begin{pmatrix} h_1^{-2} & 0 & 0 \\ 0 & h_{max}^{-2} & 0 \\ 0 & 0 & h_{max}^{-2} \end{pmatrix}. \tag{5}$$

with $h_1 = h_{max}|1 - e^{-|x-0,5|}| + 0,003$ for $h_{max} = 0,2$. This metric simulates the capture of a planar shock in the computational domain: $\Omega = [-1\,,\,1]^3$.

Since the boundary mesh is not adapted, we started the adaptation loop with an initial mesh T_0 containing no internal vertex and for which the boundary triangulation was already adapted to the metric (Figure 2). This mesh T_0 contains 4399 vertices and 13978 tetrahedra. Table 1 reports the main char-

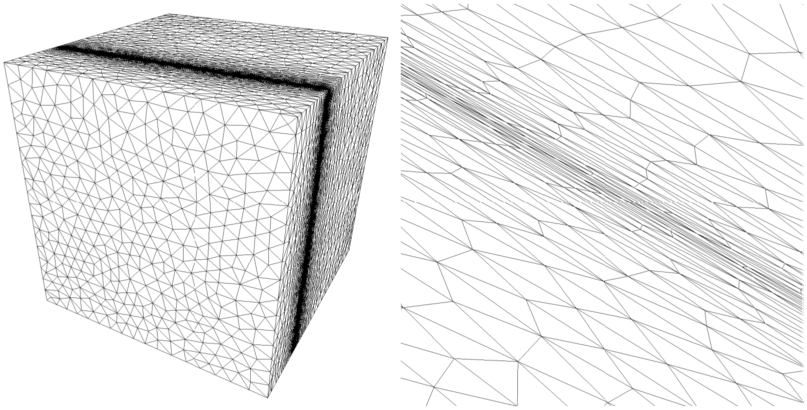

Fig. 2. *Initial mesh with adapted surface and zoom.*

acteristics of the first and final adapted mesh after 5 iterations. We can notice that the final mesh is in good agreement with the prescribed metric as indicated by the efficiency index. Moreover, more than 99% of the elements have a shape quality better than 3. The computational time is obviously decreasing at each iteration, since the mesh is close to the optimal mesh, only minor adjustements are needed to match the metric presciptions. Figure 3 shows the tetrahedral elements intersected by an arbitrary cutting plane.

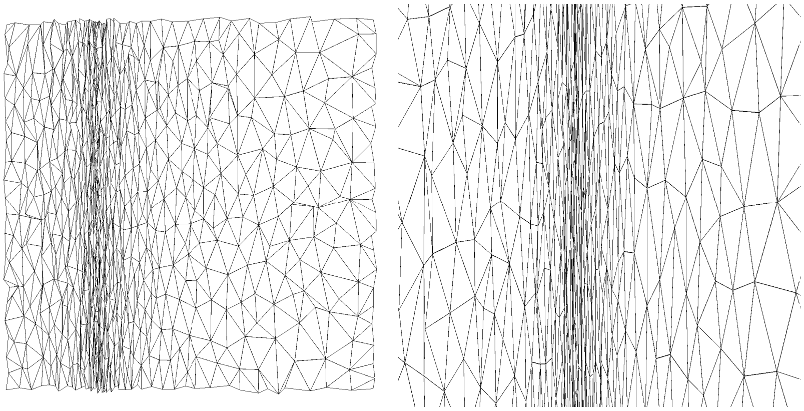

Fig. 3. *Mesh adaptaton: cutting plane through the final adapted tetrahedral mesh and local enlargement on the anisotropic elements.*

4.2 Capture of an interface

An interesting potentiality of local anisotropic mesh adaptation is related to the accuracte tracking and aproximation of a dynamically evolving interface [16]. In this approach, the metric tensor field is related to the intrinsic properties of the manifold of codimension one that correspond to the interface.

Here, we consider an implicitly defined scalar valued function u on a domain $\Omega \subset \mathbb{R}^3$ and we denote by Γ the surface associated to the isovalue $u = 0$. The objective is to produce a mesh where the density is high in the vicinity of the isosurface so as to minimize the piecewise affine approximation of this interface. To this end, we defined the following metric tensor at the vertices of all mesh elements intersected by the manifold:

	ne	τ	CPU (sec.)	$Q < 3$ (%)
1st iteration	81 336	0.8526	21.39	92.05
5th iteration	51 440	0.8712	1.94	99.96

Table 1. *Mesh features of the anisotropic adaptated meshes.*

$$M = R \begin{pmatrix} 1/\varepsilon^2 & 0 & 0 \\ 0 & |\lambda_2|/\varepsilon & 0 \\ 0 & 0 & |\lambda_3|/\varepsilon \end{pmatrix} {}^tR,$$

with $R = \begin{pmatrix} \nabla u \; v1 \; v2 \end{pmatrix}$, where (v_1, v_2) is a basis of the tangent plane to the surface and λ_i are the eigenvalues of the Hessian of u. At all other vertices, we define the metric αI_3 with $\alpha \in \mathbb{R}^+$.

Figure 4 shows an example of analytical surface captured with this method, defined in spherical coordinates as:

$$r = 0.45 + 0.3\,\sin(3\phi),$$

with $\theta \in [0; 2\pi]$ and $\phi \in [-\frac{\pi}{2}; \frac{\pi}{2}]$. We started from an initial uniform mesh of size $h = 0.2$. In the final mesh after 8 iterations, the minimal element size is $h = 10^{-3}$ and the mesh contains 10^6 vertices. The approximation error in L_∞ norm between the surface and its piecewise affine discretization is lesser than 10^{-4}.

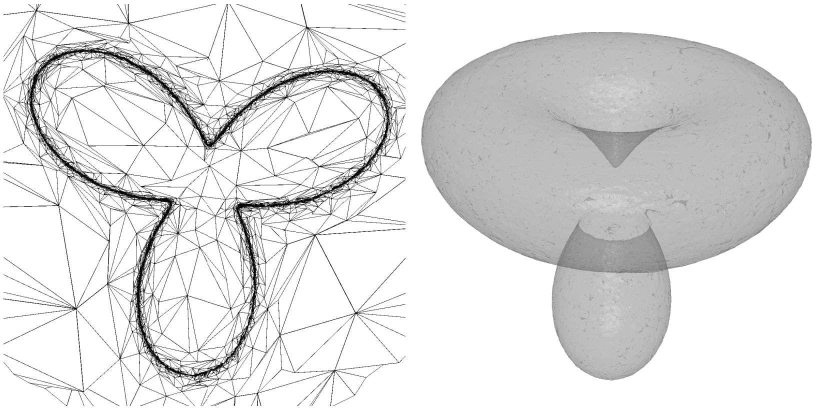

Fig. 4. *Example of mesh adaptation to capture an implicitely defined surface: cutting plane through tetrahedral elements (left) and isosurface reconstruction (right).*

4.3 Rigid bodies displacements

As mentioned previously, our local anisotropic mesh adaptation approach allows to handle rigid-body displacement problems without difficulty. The procedure is straightforward and can be decomposed in three successive stages: given a field of displacements prescribed at the domain boundaries, *e.g.* resulting from a fluid calculation, (ii) solve a linear elasticity equation as suggested by [6] to define a discrete displacement field at all mesh vertices and (iii) move the mesh vertices to the positions prescribed by this field and (iv) optimize the resulting mesh. Notice that this routine preserves the number of mesh vertices. Mesh optimization is usually based on local vertex reconnection (em i.e. edge flips) and vertex relocation, but it could also take advantage of vertex insertion or vertex deletion if needed or supported by the application.

Moving spheres through an anisotropic mesh

This aim of this analytical test case is to ilustrate the coupling between rigid-body displacement and anisotropic specifications. Here, we consider two spheres of radius 0.5 in a computatonal domain $\Omega = [-2\,,\,2] \times [-2\,,\,6] \times [-10\,,\,1.5]$. The initial mesh is adapted to a planar shock corresponding to a metric specification very similar to that given by Equation (5): with $h_1 = h_{max}|1 - e^{-|z+4|}| + 0.008$ and $h_{max} = 0.5$.

Here, we prescribed the constant translation vector $(0\,,\,0\,,\,-0.3)$ to all mesh vertices on the sphere boundaries and the two spheres will reach the bottom of the box after 34 iterations. In this example, the main problem is to preserve the anisotropic refinement throughout the simulation. To this end, we decided to allow vertex insertion and vertex deletion, in addition to vertex relocation and edge flips, during the adaptation loop. Table 2 reports some statistics on the mesh quality. Figure 5 represents several adapted meshes during the simulation, in the vicinity of the anisotropically refined region.

	np	ne	τ	CPU (sec.)	$Q < 3$ (%)
Initial mesh	12 200	60 323	0.85	-	98.66
14th move	11 026	53 224	0.85	209.40	98.30
16th move	11 316	54 950	0.85	189.03	95.72
18th move	11 133	53 567	0.85	3.57	98.12
34th move	11 282	54 441	0.86	34.49	98.83

Table 2. *Mesh features for the moving spheres test case.*

Airflow around a rotating helicopter propeller

Finally, we have coupled the moving mesh method with an airflow simulation around an helicopter propeller. Here, the flow is governed by the classical compressible Euler equations of the fluid dynamics and the numerical resolution uses the software `Fluidbox` based on finite volume and Arbitrary Lagrangian Eulerian (ALE) method [27]. The solver uses implicit time stepping scheme. Furthermore, we consider that the propeller turns with a constant angular velocity $\theta = 2\pi/10\ rad.s^{-1}$, *i.e.* thus making 10 full rotations per second. In this simulation, one mesh is generated at each time step $dt = 1e - 4$ seconds, thus 1000 meshes are needed to achieve a complete revolution.

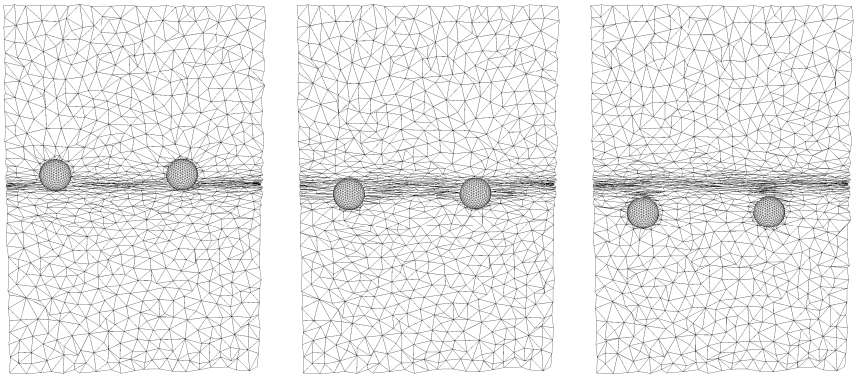

Fig. 5. *Moving spheres test case: adapted meshes at iterations 14, 16 and 18.*

In this simulation, ALE formulation imposes the meshes to have a constant number of vertices and the connectivity to remains identical to that of the initial mesh. Unfortunately, this appeared too much of a constraint. Inevitably, twisted elements will be created thus leading to invalid meshes. To overcome this problem, a few edge flips were introduced on badly shaped elements to improve the overall mesh quality and the solution had then to be interpolated on the adapted optimized mesh. In this simulation, we applied edge flips every 100 rotations, but the number of mesh vertices remained constant at all time.

Table 3 reports statistics about adapted meshes for this simulation. The overall quality of the adapted meshes can be considered as good: the vast majority of elements has a quality value above 3. At iteration 99 (just before edge flips occur), the worst element quality is close to 47 and does not perturbate the simulation. Figure 6 presents some cutting planes through the tetrahedral adapted meshes after several rotations. Figure 7 show several streamlines of the airflow.

	ne	τ	CPU (sec.)	$Q < 3$ (%)	Q_{bad}
Initial mesh	301 380	0.83	-	98.55	26
1st rotation	301 380	0.83	20.59	98.55	26
99th rotation	301 380	0.814	24.77	97.46	47
100th rotation	301 418	0.82	20.46	98.57	26
500th rotation	302 427	0.80	31.31	98.45	26
1000th rotation	302 918	0.794	27.91	98.29	32

Table 3. *Mesh features for the ALE airflow simulation.*

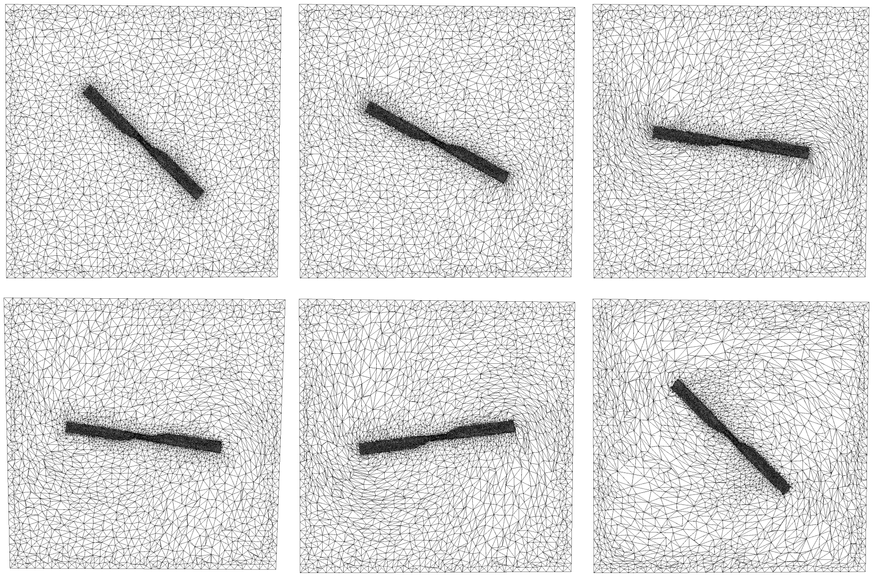

Fig. 6. *Airflow simulation: adapted meshes at iterations 1, 45, 99, 100, 145 and 500.*

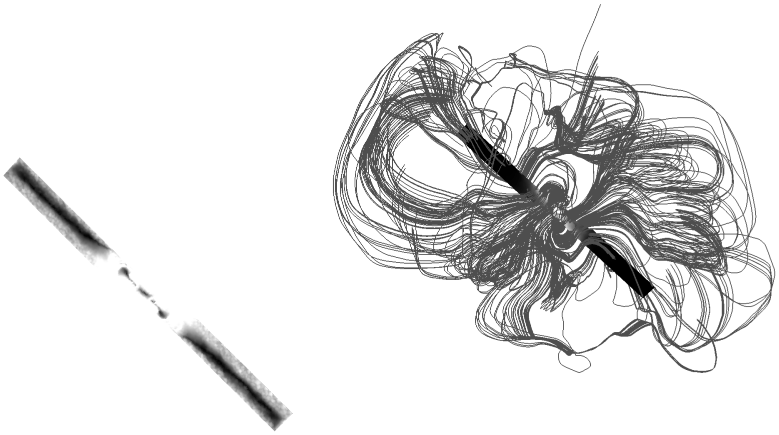

Fig. 7. *Airflow simulation: velocity modulus and streamlines.*

5 Conclusions and perspectives

In this paper, we have presented an efficient method for obtaining anisotropic adapted meshes based on Riemannian metric specifications. This approach is based on a modification of the classical Delaunay kernel and involve local mesh modification operations. The results obtained so far in the numerical simulations are promising. The next stage will be to handle dynamically evolving domains where the geometry and the topology of the domains may change in time.

References

1. F. Alauzet, A. Loseille, A. Dervieux and P. Frey, *Multi-dimensional continuous metric for mesh adaptation*, Proc. 15th Int. Meshing Roundtable, (2006), 191-214.
2. T. Apel, M. Berzins, P. K. Jimack, G. Kunert, A. Plaks, I. Tsukerman, and M. Walkley, *Mesh shape and anisotropic elements: theory and practice*, in J. R. Whiteman, editor, The Mathematics of Finite Elements and Applications X, Elsevier, Oxford, (2000), 367-376.
3. E.F. DAzevedo and R.B. Simpson, *On optimal triangular meshes for minimizing the gradient error*, Numer. Math., **59** (1991), 321-348.
4. I. Babuška and A.K. Aziz, *On the angle condition in the finite element method*, SIAM J. Numer. Anal., **13** (1976), 214-226.
5. T. Baker, *Mesh adaptation strategies for problems in fluid dynamics*, Finite Elem. Anal. Des., **25**(3-4), (1997), 243-273.
6. Timothy J. Baker, *Mesh Movement and Metamorphosis*, Eng. Comput., **18**(3) (2002), 188-198.
7. R.E. Bank and R.K. Smith, *Mesh smoothing using a posteriori error estimates*, SIAM J. Numer. Anal., **34**, (1997), 979-997.
8. M. Berzins, *A solution-based triangular and tetrahedral mesh quality indicator*, SIAM J. Sci. Comput., **19** (1998), 2051-2060.
9. F.J. Bossen and P.S. Heckbert, *A pliant method for anisotropic mesh generation*, Proc. 5th Int. Meshing Roundtable, (1996) 63-74.
10. J.U. Brackbill, *An adaptive grid with directional control*, J. Comput. Phys., **108**, (1993), 38-50.
11. G. Carey, *Computational grids: Generation, adaptation, and solution strategies*, Taylor & Francis, Washington, (1997).
12. M.J. Castro-Diaz, F. Hecht, B. Mohammadi and O. Pironneau, *Anisotropic unstructured mesh adaption for flow simulations*, Int. J. Numer. Meth. Fluids, **25** (1997), 475-491.
13. P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, (1978).
14. V. Dolejsi, *Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes*, Computing and Visualisation in Science, **1** (1998), 165-178.

15. J. Dompierre, M.-G. Vallet, Y. Bourgault, M. Fortin and W.G. Habashi, *Anisotropic mesh adaptation: towards user-indepedent, mesh-independent and solver-independent CFD. Part III: unstructured meshes*, Int. J. Numer. Meth. Fluids, **39** (2002), 675-702.
16. V. Ducrot, P. Frey, *Contrôle de l'approximation géométrique d'une interface par une métrique anisotrope*, C.R. Acad. Sci. Paris, **345**, Série I (2007), 537-542.
17. L. Formaggia and S. Perotto, *New anisotropic a priori error estimates*, Numer. Math., **89** (2001), 641-667.
18. M. Fortin, *Etude numérique d'estimations d'erreur* a posteriori, REEF, **9**, (2000), 467-486.
19. P.J. Frey and P.L. George, Mesh generation. Application to finite elements, 2nd ed., Wiley, 2008.
20. P.J. Frey and F. Alauzet, *Anisotropic mesh adaptation for CFD simulations*, Comput. Methods Appl. Mech. Engrg., **194**(48-49), (2005), 5068-5082.
21. P.L. George, *Premières expériences de maillage automatique par une méthode de Delaunay anisotrope en trois dimensions*, RT-0272, Inria, (2002).
22. C. Gruau and T. Coupez, *3d unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric*, Comput. Methods Appl. Mech. Engrg, **194**(48-49), (2005), 49514976.
23. W. Huang, *Metric tensors for anisotropic mesh generation*, J. Comput. Phys., **204** (2005), 633-665.
24. X. Jiao, A. Colombi, X. Ni and J. Hart, *Anisotropic mesh adaptation for evolving triangulated surfaces*, Proc. 15th Int. Meshing Roundtable, (2006), 173-190.
25. P.M. Knupp and N. Robidoux, *A framework for variational grid generation: conditioning the Jacobian matrix with matrix norms*, SIAM J. Sci. Comput., **21** (2000), 2029-2047.
26. G. Kunert, *A local problem error estimator for anisotropic tetrahedral finite element meshes*, SIAM J. Numer. Anal., **39** (2001), 668-689.
27. B. Nkonga, *On the conservative and accurate CFD approximations for moving meshes and moving boundaries*, Comput. Methods Appl. Mech. Engrg., **190** (2000) 1801-1825.
28. S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag New York, Applied Mathematical Sciences, **153**, (2002)
29. J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, *Adaptive remeshing for compressible flow computations*, J. Comput. Phys., **72** (1997), 449-466.
30. M. Picasso, *An anisotropic error indicator based on Zienkiewicz-Zhu error estimator: Application to elliptic and parabolic problems*, SIAM J. Sci. Comput., **24** (2003), 1328-1355.
31. J. Remacle, X. Li, M.S. Shephard and J.E. Flaherty, *Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods*, Int. J. Numer. Meth. Engr., (2003), - .
32. S. Rippa, *Long and thin triangles can be good for linear interpolation*, SIAM J. Numer. Anal., **29**, (1992), 257-270.
33. B. Semper and G. Liao, *A moving grid finite-element method using grid deformation*, Numer. Methods in PDEs, **11**, (1995), 603-615.
34. J.R. Shewchuk, *What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures*, 11th Int. Meshing Roundtable (2002), 115-126.
35. K.G. Siebert, *An a posteriori error estimator for anisotropic refinement*, Numer. Math., **73**, (1996), 373-398.

36. R.B. Simpson, *Anisotropic mesh transformations and optimal error control*, Appl. Numer. Math., **14** (1994), 183-198.
37. J.F. Thompson, B.K. Soni and N.P. Weatherill, *Handbook of grid generation*, CRC Press, London, (1999).
38. M. Zlámal, *On the finite element method*, Numer. Math., **12** (1968), 394-409.