On a Basis Framework for High Order Anisotropic Mesh Adaptation

Thierry Coupez*  
*ICI, Ecole Centrale de Nantes, 1 rue de la no , 4430 Nantes, France

Abstract
In this paper we propose a new framework to build high order (mainly P2) meshes (with curved edges) driven by a metric field. In contrast to the common way to obtain such meshes that is to deform a P1 mesh by using an appropriate differential operator, we study a completely different route based on a fully adaptive strategy. Meshes are built in a full Riemann framework, edges being redefined as the geodesics of the underlying Riemannian manifold. The iterative process is made of three ingredients, the mesh topology operation that are defined by basic algebraic topological operations, the on flight edge calculations and the element conformity and shape measurement extension. This paper will show first results of such an approach and will discuss the performance issues.

Keywords: 2nd order meshing; curved element; geodesic edges; mesh topology; anisotropic mesh adaptation;

1. INTRODUCTION

Anisotropic mesh adaptation techniques have been shown to be very efficient in the last decade with very convincing results [1–6] and even completely new domains of applications [7]. In modern anisotropic Finite Element, the element stretching can attain very high level as 1 for 1000 to 10000 when capturing boundary layers. The gain in terms of mesh points versus a uniform mesh is thus about 1000 to 10000 and it has been proven to be the only successful known solution for specific applications. These techniques are well understood and fully developed for P1 simplex element. Until now, the gain in terms of mesh entities was enough to compensate a low order numerical solution. Today, supercomputers of thousands of cores are available and enable to run computations on meshes of several millions of points including parallel mesh adaptation [8], and it changes our former point of view, since the extra cost for high order meshing could be compensated by a higher convergence order. 

The most efficient way to build anisotropic adaptive meshes is based on local modifications ([9], [10], [11]) of an existing mesh and the proposed extension to high order element follows this strategy. Beyond P1 element, the interpolation error analysis gives rise to element metrics that are not any-more constant and therefore unit meshes are not made of straight elements. Two ways can be followed in the meshing community to go P2 elements. The first one is to mesh with classical P1 (straight) simplex elements and thereafter transform them in P2 (generally for curved boundary recovery layer.) The second way we want to explore here is a direct metric driven construction of P2 element. For that purpose we propose to revisit the framework proposed in [5, 12] for mesh generation, that combine simple algebraic topology operations and basic differential geometric considerations, with the extension and generalization of the tensor approach and associated edge based error first introduced in [1]. The novelty of this framework is to give

*Corresponding author  
E-mail address: thierry.coupez@ec-nantes.fr
a clue for a straightforward construction of unit meshes that provide automatically curved edges as a consequence of the minimal path definition in non-Euclidian geometry, while the conformity remains guaranty by a minimal volume principle. First examples of P2 anisotropic meshes will be shown and potential impact in applications will be discussed.

2. Mesh topology

The mesh topology represents the connectivity of the mesh. A mesh is made of piece-wise transformations of the space that are described by elements. The connection of elements by the common vertices is described by a graph of the mesh node numbers which properties gave rise to the mesh topology.

Let \( N \subset \mathbb{N} \) be a finite set of node numbers. A topological element is a set of \( D \) node numbers

\[ K \in \mathcal{P}(N), \text{card}(K) = D \]

If \( d \) is a space dimension, a \( d \)-simplex is composed of \( D = d + 1 \) vertices. The boundary \( \partial K \) of a \( d \)-simplex \( K \), is made of all subsets of \( D - 1 \) nodes in \( K \).

\[ \partial K = \{ F \subset K, \text{card}(F) = D - 1 \} \]

A \( d \)-simplex will be a (topological) element in the sequel. Let consider a set of elements \( \mathcal{K} \subset \mathcal{P}(N) \) an let us define the set of faces of \( \mathcal{K} \) by

\[ \mathcal{F}(\mathcal{K}) = \bigcup_{K \in \mathcal{K}} \partial K \]

The elements sharing a node numbered \( i \) is:

\[ \mathcal{K}(i) = \{ K \in \mathcal{K}, i \in K \} \quad (1) \]

The elements sharing an edge made of two node numbers \( ij \) is:

\[ \mathcal{K}(ij) = \mathcal{K}(i) \cap \mathcal{K}(j) \quad (2) \]

And more generally, the set of elements sharing a subset of nodes \( F \) (a face for instance) is given by:

\[ \mathcal{K}(F) = \bigcap_{i \in F} \mathcal{K}(i) \quad (3) \]

\( \mathcal{K} \) is a mesh topology if each face of \( \mathcal{K} \) belongs at most to two element:

\[ \text{card}(F) \leq 2 \forall F \in \mathcal{F} \]

The boundary of a mesh topology is given by the set of faces that belong to only one element:

\[ \partial \mathcal{K} = \{ F \in \mathcal{F}, \text{card}(\mathcal{K}(F)) = 1 \} \]

and to simplify, a mesh topology can be completed in order to be without boundary:

\[ \mathcal{K} = \mathcal{K} \cup \{ T = \{ 0 \} \cup F, F \in \partial \mathcal{K} \} \]

3. Derivation of a mesh topology

3.1. Elementary operator

The generic cut and paste operators

\[ \mathcal{K} \leftarrow \mathcal{K} - a + b \]
The elementary operator that transforms an element $T$, can be defined by substituting vertex $i$ by vertex $j$ in $T$:

$$T_{i,j} = \begin{cases} T, & T \not\in \mathcal{K}(i) \\ \emptyset, & T \in \mathcal{K}(i) \cap \mathcal{K}(j) \\ T - i + j & \text{otherwise} \end{cases}$$

The mesh topology derivation operator is obtained by applying the above elementary operator to all elements by:

$$\mathcal{K}_{i,j} = \{T_{i,j}, \ T \in \mathcal{K}\} = \bigcup_{T \in \mathcal{K}}\{T_{i,j}\}$$

A specific operation is always the result of the elementary substitution operator applied to a subset of connected elements Node operator 1 includes the node deletion: $\mathcal{K} \rightarrow \mathcal{K}_{i,j}$

![Figure 1: node substitution operator](image)

**Figure 1: node substitution operator**

Edge operator 2 includes the node creation: $\mathcal{K} \rightarrow \mathcal{K}_{ij,k}$

![Figure 2: edge swapping by vertex number substitution](image)

**Figure 2: edge swapping by vertex number substitution**

$$T_{i,j,k} = \begin{cases} \{T\}, & T \not\in \mathcal{K}(i) \cap \mathcal{K}(j) \\ \emptyset, & T \in \mathcal{K}(i) \cap \mathcal{K}(j) \cap \mathcal{K}(k) \\ \{T - i + k\} \cup \{T - j + k\} & \text{otherwise} \end{cases}$$

$$\mathcal{K}_{i,j,k} = \{T_{i,j,k}, \ T \in \mathcal{K}\} = \bigcup_{T \in \mathcal{K}}\{T_{i,j,k}\}$$

Face/element operator 3:

![Figure 3: element operation](image)

$$\mathcal{K} \rightarrow \mathcal{K}_{ijk,l}$$
3.2. Mesh topology conformity

Any mesh topology operation must be constrained to preserve the conformity by means of the definition of the previous section: It means that if $\mathcal{K}$ is a mesh topology under which condition $\mathcal{K}' = \mathcal{K}_{i,j}$ is still a mesh topology.

**Theorem 3.1** $\mathcal{K}' = \mathcal{K}_{i,j}$ is a mesh topology if:

$$N(\mathcal{K}(i)) \cap N(\mathcal{K}(j)) = N(\mathcal{K}(i) \cap \mathcal{K}(j))$$

We get immediately the following generalization: $\mathcal{K}_{i,j,k}$ is a mesh topology if:

$$N(\mathcal{K}(i)) \cap N(\mathcal{K}(j)) \cap ... \cap N(\mathcal{K}(k)) = N(\mathcal{K}(i) \cap \mathcal{K}(j) \cap ... \cap \mathcal{K}(k))$$

4. Mesh conformity

Let us consider the piece-wise mapping defined from the mesh topology to the physical space:

$$x : \begin{cases} \mathcal{K} \to \mathbb{R}^d \\ K \to x(K) \end{cases} \quad (4)$$

Mapping from the set of nodes to $\mathbb{R}^d$ provides the set of node coordinates of the mesh, $\mathbf{X} = \{ \mathbf{X}_i \in \mathbb{R}^d, i = 1, \ldots, N \}$. The $d$-simplexes $\Omega_K$ generate a domain, $\Omega(\mathcal{K}, \mathbf{X})$ and the boundary faces generate a surface, $\Gamma(\partial \mathcal{K}, \mathbf{X})$. $\langle \mathcal{K}, \mathbf{X} \rangle$ defines a conform mesh of $\Omega$ for the conditions given below.

**Theorem 4.1 Minimal volume theorem.**

Let us assume that $\langle \partial \mathcal{K}, \mathbf{X} \rangle$ is a conform triangulation of the boundary domain $\Omega$ (i.e. $\Gamma(\partial \mathcal{K}, \mathbf{X}) = \partial \Omega$). Then, if $|\Omega_K|$ is the volume of $\Omega_K$.

1. $\sum_{T \in K} |x(T)| \geq |\Omega|$
2. $\langle T, x \rangle$ is a mesh of $\Omega$ if and only if:
$$\sum_{T \in K} |x(T)| = |\Omega|$$

For P1 element $|x(T)|$ is the volume of element which is generalized for high order element by:

$$|x(T)| = \int_T |\det(\nabla x)| d\mathbf{T}$$

5. Mapping

in this section is considered the P2 case. Let $\psi^i, i = 1, \ldots, D$ the standard P1 basis function and consider the hierarchical following form for a P2 element:

$$x(T) = \sum_{i \in I} \psi^i X^i + \sum_{i,j \in I, i \neq j} Y^{ij}(4 \times \psi^i \psi^j)$$

where $Y^{ij}$ is the deviation vector from the mid-edge node position. The mid-edge node position is simply defined by $X^i = \frac{x_i + x_j}{2} + Y^{ij}$ the transformation gradient is made of the linear part and the quadratic one:

$$\nabla x(T) = \nabla X + \nabla Y$$
the conformity by reusing the minimal volume theorem

\[ \min_{K \in \mathcal{P}(N)} \sum_{T \in K} |x(T)| = \min_{K \in \mathcal{P}(N)} \int_T |\det(\hat{\nabla}x)| d\hat{T} \]

the shape factor extended definition by:

\[ c(T)^2 = \int_T \frac{\det(\hat{\nabla}x(T) \hat{\nabla}x(T))}{\frac{1}{d} \int_T \text{trace}(\hat{\nabla}x(T) \hat{\nabla}x(T))d} \]

where

\[ c(\hat{T})^2 = \int_T \frac{\det(Id)}{\frac{1}{d} \int_T \text{trace}(Id)d} = 1 \]

The extension to anisotropic high order mesh is as usual by embedding a metric field in the calculation of the invariants of the transformation:

\[ c(T)^2 = \int_T \frac{\det(\hat{\nabla}x(T) M(T) \hat{\nabla}x(T))}{\frac{1}{d} \int_T \text{trace}(\hat{\nabla}x(T) M(T) \hat{\nabla}x(T))d} \]

the mapping restriction to the edge is depending only on the two vertices (and from the topology side can be defined by the only knowledge of these vertices) and the edge deviation vector:

\[ x(s) = \frac{1 - s}{2} X^i + \frac{1 + s}{2} X^j + Y^{ij}(1 - s^2) \]

and the edge is curved when \( Y^{ij} \neq 0 \)

\[ x(0) = \frac{X^i + X^j}{2} + Y^{ij} \]

6. Geodesic calculation

Under a metric field, \( M \), edges are the geodesics between vertices. Following a differential geometry argument we can assume that there is a unique path of minimal length between two vertices for a given metric field. As a consequence it is possible to calculate on flight the mid-node position from the knowledge of the metric along the edge.

\( M(ij) \implies Y^{ij} \) let us introduce the following definition: Energy of the curve:

Euclidian: \( E^{ij} = \int_{X^i} |dx|^2 \) \quad Riemannian: \( E^{ij}(M) = \int_{X^i} (M dx, dx) \)
Length of a curve:

\[ L^{ij} = \int_{X^i}^{X^j} |dx| \quad \rightarrow \quad \text{Euclidian:} \quad L^{ij}(\mathcal{M}) = \int_{X^i}^{X^j} (\mathcal{M} \, dx, \, dx)^{\frac{1}{2}} \]

For a parametric curve:

\[ dx = x'(s)ds \quad \rightarrow \quad \text{Riemannian:} \quad L^{ij}(\mathcal{M}) = \int_{-1}^{1} (\mathcal{M} \, x'(s), \, x'(s))^\frac{1}{2} ds \]

For the P2: \( x(s) = \frac{1-s}{2}X^i + \frac{1+s}{2}X^j + \frac{Y^{ij}}{2}(1-s^2) \quad s \in [-1, 1] \)

\[ dx = (\frac{1}{2}X^i - 2sY^{ij})ds \]

We are in position now to give the trick to calculate the geodesic. Let introduce the following interpolation of the metric along the edge:

\[ m(s) = \frac{\mathcal{M}(s)}{(\mathcal{M}(s), x'(s), x'(s))^\frac{1}{2}} \quad (5) \]

and the straightforward interpolation of the "square root of the metric":

\[ m(s) = \frac{1-s}{2}m^i + \frac{1+s}{2}m^j + m^{ij}(1-s^2) \]

we optimize the energy of the curve under \( m \) by solving a simple quadratic optimisation in \( Y^{ij} \) by:

\[ \min_{Y^{ij}} E(Y^{ij}) = \int_{-1}^{1} \left( m(s)(\frac{1}{2}X^i - 2sY^{ij}), (\frac{1}{2}X^j - 2sY^{ij}) \right) ds \quad (6) \]

the solution is:

\[ Y^{ij}(m) = \left( m^i + m^j + \frac{4}{5}m^{ij} \right)^{-1} \left( m^i - m^j \right) \frac{X^{ij}}{4} \quad (7) \]

the problem is in fact non linear since \( Y^{ij}(m) \) is depending on \( m \) which depends on \( Y \). But in practice very few iterations are necessary to get the solution and thus:

**Lemma 6.1** \( Y^{ij}(m) \) minimizes the length path:

\[ Y^{ij}(m) = \arg \min L^{ij}(\mathcal{M}) = \int_{-1}^{1} (\mathcal{M} \, x'(s), \, x'(s))^\frac{1}{2} ds \]

7. Illustrations

In the example plotted on Figure 5 we consider the following Riemannian metric: \( \mathcal{M} = \frac{1}{h^2} (Id + \nabla u \times \nabla u) \) with \( u = \sqrt{R^2 - r^2} \). On Figure 6, the P1 resulting mesh is compared to the P2 one.

The C++ 2011 has been used to program the mesh topology management set, intersection of sets, and Eigen library for tensorial calculation (http://eigen.tuxfamily.org).

8. Conclusion

In this short note we gave a basis framework with mathematical results that can be used to perform anisotropic mesh of higher order by means of stretched and curved elements. The theory is solid enough, both in 2d and in 3d and seems to be valid for higher order element. The use of a Riemann metric enforces the element to be curved not only at the boundary but everywhere in the computational domain. It is a consequence of the edge definition as the geodesic between two vertices. The geodesic can be calculated on flight and the topological operation can be done almost as usual. There is here a clear possibility to build high order mesh in a flexible way as for the popular P1 element.
References


