

Ph. D. Thesis

submitted to

L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS

by

**Cyril Gruau**

in partial fulfillment of the requirements for the degree of

DOCTOR

of

COMPUTATIONAL MECHANICS

---

**Metric generation  
for anisotropic mesh adaption  
with numerical applications  
to material forming simulation**

---

approved on 10<sup>th</sup> December 2004 by the examining committee :

Professor	Michel RASCLE	President
Professor	Paul-Louis GEORGE	Reviewer
Professor	Frédéric HECHT	Reviewer
Doctor	Éric BONNEAU	Inspector
Doctor	Richard DUCLOUX	Inspector
Professor	Thierry COUPEZ	Thesis advisor
Professor	Jean-François AGASSANT	Thesis advisor



*with the memory of my grandfather*



# Contents

<b>Main introduction</b>	<b>3</b>
Framework . . . . .	3
Rem3D . . . . .	3
Initial context . . . . .	4
Organization . . . . .	5
<b>I 3D unstructured and anisotropic mesh generation</b>	<b>7</b>
Introduction . . . . .	9
<b>1 Meshes and metrics</b>	<b>11</b>
1.1 Metric . . . . .	11
1.1.1 Definition . . . . .	11
1.1.2 Diagonalization . . . . .	11
1.2 Simplex . . . . .	12
1.2.1 Definition . . . . .	12
1.2.2 Simplex measure . . . . .	13
1.2.3 Metric associated with a simplex . . . . .	14
1.2.4 Simplex quality . . . . .	14
1.3 Mesh . . . . .	15
1.3.1 Definition . . . . .	15
1.3.2 Mesh volume . . . . .	17
1.3.3 Mesh conformity . . . . .	18
<b>2 Meshing a domain from a mesh of its boundary</b>	<b>19</b>
2.1 State of the art: three classical methods . . . . .	19
2.1.1 Meshing by tree . . . . .	19
2.1.2 Advancing front . . . . .	20
2.1.3 Delaunay method . . . . .	22
2.1.3.1 Delaunay triangulation . . . . .	22
2.1.3.2 Delaunay mesh generation . . . . .	23
2.2 Topological optimization . . . . .	25
2.2.1 Mathematical background . . . . .	25
2.2.1.1 Mesh topology . . . . .	25
2.2.1.2 Minimal volume theorem . . . . .	26
2.2.1.3 Shape criterion and order relationship . . . . .	29
2.2.2 Mesh generation . . . . .	29
2.2.2.1 Local topology optimization . . . . .	29
2.2.2.2 Convergence . . . . .	31
2.2.2.3 Boundary treatment . . . . .	33
2.3 Conclusion . . . . .	34

<b>3</b>	<b>Mesh adaption driven by a metric field</b>	<b>35</b>
3.1	State of the art: an hybrid Delaunay / advancing front method . . . . .	36
3.1.1	Constrained advancing front . . . . .	36
3.1.2	Algorithm modifications . . . . .	37
3.2	Topological optimization . . . . .	39
3.2.1	Anisotropic optimization strategy . . . . .	39
3.2.1.1	Anisotropic shape criterion and size criterion . . . . .	39
3.2.1.2	Order relationship between local topologies . . . . .	40
3.2.1.3	Algorithm modifications . . . . .	40
3.2.1.4	Metric update . . . . .	41
3.2.2	Validation . . . . .	42
3.2.2.1	Anisotropic case in 2D . . . . .	43
3.2.2.2	Isotropic case in 2D . . . . .	45
3.2.2.3	Anisotropic case in 3D . . . . .	47
3.3	Conclusion . . . . .	50
<b>II</b>	<b>Natural metric</b>	<b>51</b>
	Introduction . . . . .	53
<b>4</b>	<b>Natural metric generated a priori</b>	<b>55</b>
4.1	Local anisotropy . . . . .	58
4.1.1	Definition . . . . .	58
4.1.2	Neighborhood growing . . . . .	59
4.2	Discrete elliptic interpolation . . . . .	60
4.2.1	Semi-definite programming problem . . . . .	60
4.2.2	Tensor decomposition . . . . .	63
4.2.3	Elementary metric combination . . . . .	66
4.2.4	Comparisons . . . . .	67
4.3	Appropriate order . . . . .	68
4.3.1	Interpolation error . . . . .	68
4.3.2	Final algorithm . . . . .	70
4.4	Several element layers through the thickness . . . . .	71
4.4.1	Selective division . . . . .	71
4.4.2	Iteration process . . . . .	71
4.4.3	Multiple thickness . . . . .	75
4.5	Numerical applications . . . . .	77
4.5.1	Natural mesh of cavities . . . . .	77
4.5.2	Limitations and conclusion . . . . .	80
<b>5</b>	<b>Natural metric corrected a posteriori</b>	<b>83</b>
5.1	Scalar field heterogeneity . . . . .	84
5.1.1	Typical simulation . . . . .	84
5.1.1.1	Simulation conditions . . . . .	84
5.1.1.2	Potentially interesting fields . . . . .	86
5.1.2	Natural metric correction . . . . .	86
5.1.2.1	Additional term . . . . .	86
5.1.2.2	Application . . . . .	89
5.2	Error indicator on anisotropic meshes . . . . .	91
5.2.1	Metric corrected by an indicator uniformity . . . . .	91
5.2.1.1	Computation of the local correction factor . . . . .	91
5.2.1.2	A possible strategy . . . . .	93
5.2.2	Error estimation with the mesh anisotropy taken into account . . . . .	93

<b>III</b>	<b>Multidomain metric</b>	<b>95</b>
	Introduction . . . . .	97
<b>6</b>	<b>Multidomain metric generated a priori</b>	<b>99</b>
6.1	Natural sub-mesh . . . . .	101
6.1.1	Exact interfaces . . . . .	101
6.1.2	Fuzzy interfaces . . . . .	102
6.2	Discontinuity surface capturing . . . . .	103
6.2.1	Interface tensor . . . . .	103
6.2.2	Discontinuous $P^0$ interpolation and voxelization . . . . .	105
6.2.3	Discontinuous $P^0$ gradient and triple point treatment . . . . .	106
6.2.4	Capillary number and final algorithm . . . . .	107
6.2.5	Iteration process and convergence criterion . . . . .	110
<b>7</b>	<b>Numerical applications</b>	<b>115</b>
7.1	Polymer forming . . . . .	115
7.1.1	Validation of the fuzzy multidomain approach . . . . .	115
7.1.2	Injection molding with a complex geometry and thermal coupling . . . . .	119
7.1.3	Over-molding . . . . .	119
7.2	Computations at microscopic scale . . . . .	123
7.2.1	Rigid inclusion movement in a pure shearing flow . . . . .	123
7.2.2	Polycrystalline deformation . . . . .	125
7.3	Boundary layers . . . . .	127
7.3.1	Monodomain mesh improvement . . . . .	127
7.3.2	Boundary layer with a fixed thickness size . . . . .	127
7.4	Mesh generation . . . . .	132
	<b>Main conclusion</b>	<b>139</b>
	Summarize . . . . .	139
	Industrial applications . . . . .	141
	Mathematical contributions . . . . .	141
	Publications . . . . .	141
	Future work . . . . .	142
	<b>Appendices</b>	<b>147</b>
<b>A</b>	<b>Complements on meshes and metrics</b>	<b>147</b>
A.1	Metric . . . . .	147
A.1.1	Conic and parametrization . . . . .	147
A.1.2	Tensorial product . . . . .	147
A.1.3	Operators between metrics . . . . .	148
A.1.3.1	Several averaged values . . . . .	148
A.1.3.2	Intersection . . . . .	150
A.1.4	Analytical metrics . . . . .	151
A.1.4.1	Cartesian metric . . . . .	151
A.1.4.2	Cylindrical metric . . . . .	152
A.1.4.3	Spherical metric . . . . .	154
A.2	Simplex . . . . .	155
A.2.1	Orientation . . . . .	155
A.2.2	Outgoing normal vector . . . . .	156
A.2.3	Equilateral simplex . . . . .	157
A.3	Mesh . . . . .	160

A.3.1	Reverse topology . . . . .	160
A.3.2	Neighborhood . . . . .	161
A.3.3	4D mesh . . . . .	161
<b>B</b>	<b>Complements on natural and multidomain metrics</b>	<b>167</b>
B.1	Local computation of the layer number . . . . .	167
B.1.1	Minimal order evaluation . . . . .	167
B.1.2	Convergence criterion . . . . .	168
B.2	Subdomain boundary . . . . .	169
B.2.1	Boundary nodes . . . . .	169
B.2.2	Boundary faces . . . . .	170
B.3	Voxelization . . . . .	171
B.3.1	Sampling . . . . .	171
B.3.2	Mesh screening . . . . .	171
<b>C</b>	<b>Rem3D simulation example</b>	<b>173</b>
C.1	Pre-treatment . . . . .	174
C.1.1	Mesh . . . . .	174
C.1.2	Materials . . . . .	175
C.1.3	Initial conditions and boundary conditions . . . . .	175
C.2	Post-treatment . . . . .	176
C.2.1	Flow during injection . . . . .	176
C.2.2	Weld line study . . . . .	180
<b>D</b>	<b>Notations</b>	<b>183</b>
	<b>List of figures</b>	<b>189</b>
	<b>List of tables</b>	<b>191</b>
	<b>Bibliography</b>	<b>193</b>
	<b>Index</b>	<b>203</b>
	<b>Abstract</b>	<b>208</b>

# Main introduction



## Framework

The work presented here has been achieved between October 2001 and September 2004. It takes place in the Rem3D project, managed by the Rem3D Consortium, involving:

- Atofina (Cerdato)
- Dow Chemicals
- Essilor International
- FCI (Areva)
- Plastic Omnium
- Schneider Electric
- SNECMA Propulsion Solide
- Transvalor
- and the Material Forming Center (CEMEF), research laboratory of l'École des Mines de Paris.

## Rem3D

The CIMlib library, from which Rem3D is built, is currently developed at CEMEF, by the High Performance for Material team. Several projects have contributed to the CIMlib improvements like [Magnin 1994, Pichelin 1998, Nouatin 2000, Bigot 2001, Fournier 2003]. Here we draw a list of major functionalities related to the mesh technology dealt with in the present document.

All CIMlib solvers, including:

- a mixed solver  $P^1 + /P^1$  (mini-element velocity, pressure) for Navier-Stokes equations [Saez 2003] with compressibility and viscoplasticity [Silva 2004]
- a mixed Galerkin solver  $P^0/P^0+$  (discontinuous temperature, discontinuous flux) for heat transfer equation [Batkam 2002]
- a  $P^0$  discontinuous Galerkin solver for transport equation [Batkam 2002, Bruchon 2004] and tensorial equation of orientation [Silva 2004, Redjeb 2003]

are using:

- 3D tetrahedral, linear and unstructured elements (that is the reason why we focus our attention on that type of meshes in this document)
- only one mesh for all the parts of a simulation (the multidomain mesh notion is introduced for that purpose in chapter 6)
- an Eulerian kinematics with a R-adaption of the initial mesh, which remains the same during all the process (a good initial mesh generation as in part II is required)
- a space-time formulation, for which 4D mesh would be very interesting.

Furthermore, industrial processes addressed by Rem3D are:

- extrusion and injection molding (filling, compacting and cooling [Silva 2004]) for polymers, as in figure 1

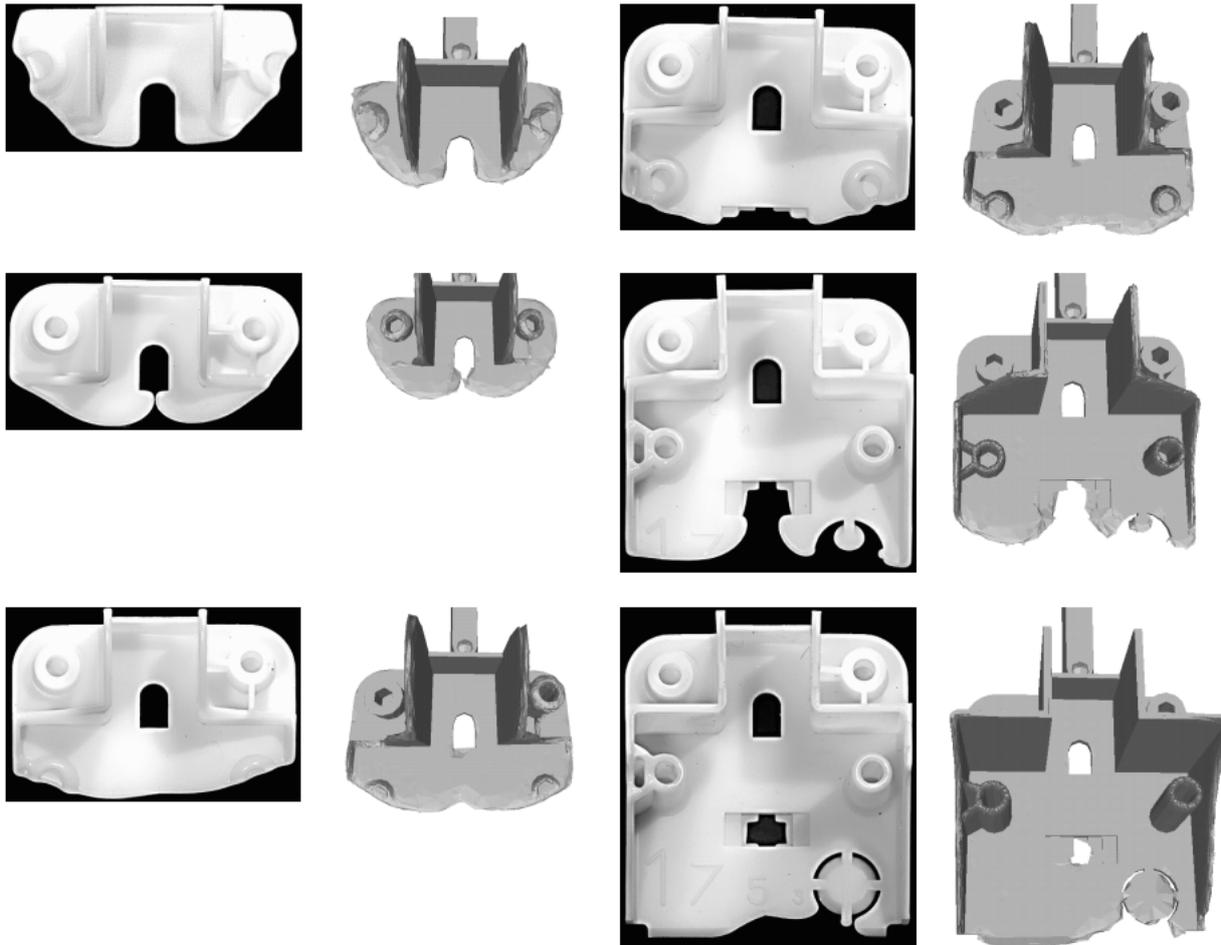


Figure 1: Comparisons between experimental short shots (in white on black) and Rem3D computation

- all multidomain versions of extrusion and injection molding: co-extrusion, co-injection, gas or water assisted injection [Daboussy 2000], over-molding [Batkam 2002] and fiber reinforced injection [Megally *et al.* 2004]
- other multidomain process, as casting [Saez 2003], polymer foam expansion [Bruchon 2004] and glass fiber spinning [Rinaldi-Mareel 2004].

All these process simulations have specific needs about the mesh (following section).

## Initial context

At the beginning of this work, the picture is as follows:

- industrial parts are geometrically complex (including curvature and sudden change of the thickness size)
- thin zones are frequent (for cooling time reduction), while thick zones are also present
- the mold around the cavity has to be taken into account because of heterogeneous thermal conditions.

Altogether, these requirements make the meshing process very difficult (when not impossible), even more difficult than the simulation itself. The need of an automatic tool for meshing all configurations is obvious. The aim of this work is to address this problem of a priori mesh generation for complex geometries.

## Organization

The three parts of this document (I, II and III) are depicted on the mesh lifecycle (figure 2):

- the first part (page 9) is devoted to the existing results: mesh generation from a mesh of the boundary and adaption to a metric field
- the second part (page 53) describes a new metric field, designed for building a suitable mesh of the cavity, may it be improved by a solver and an error estimator
- the third part (page97) extends this metric field to the multidomain framework, that is, when several parts have to be meshed simultaneously.

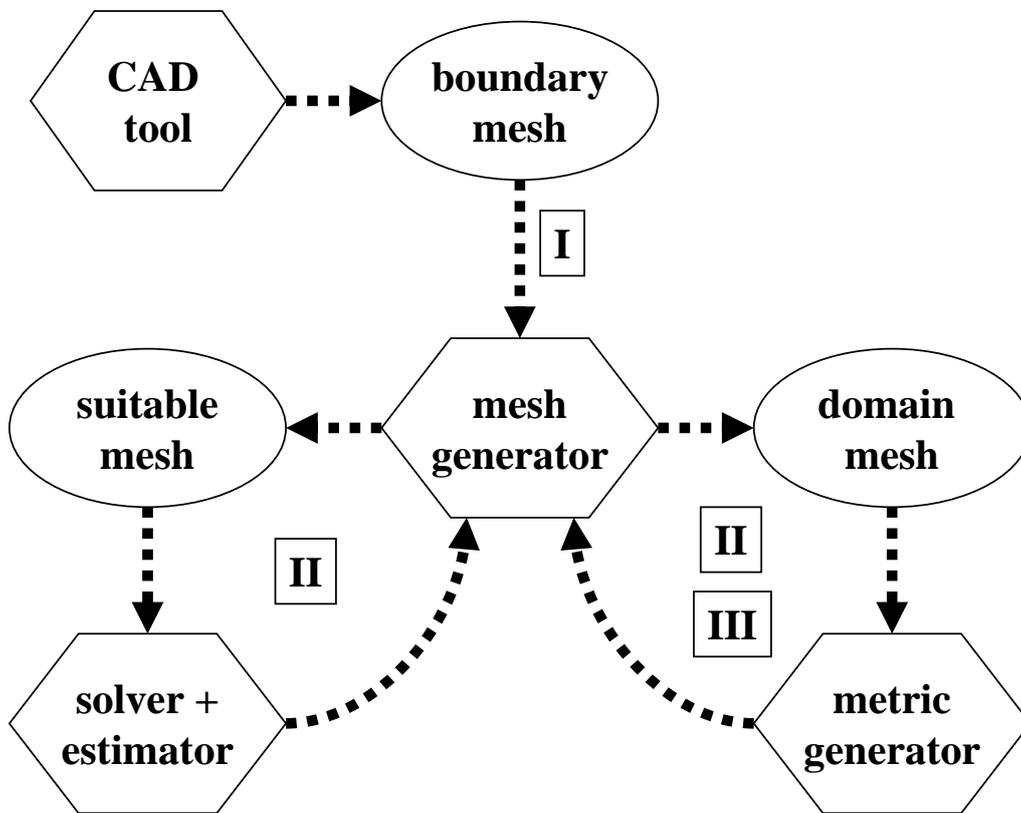


Figure 2: The overall mesh generation and adaption process

Except the initial boundary meshing, obtained with a CAD tool, which is beyond the scope of this work, all the steps of figure 2 are concerned with a part of this document, including, some appendices dedicated to technical information about meshes and metrics (page 147).



## Part I

# 3D unstructured and anisotropic mesh generation



## Introduction

This document is not only devoted to new developments about metrics (parts II and III), but also to the opportunity of capitalizing on the CEMEF meshing tools.

In this part, several mathematical results are proved in a more general framework than before. In particular, the minimal volume theorem is proved without assumption about orientation. Furthermore, several minor results are generalized for all dimension (2, 3 or more).

We focus our attention on unstructured mesh composed of simplices, generated by the so-called topological mesh generator [Coupez 1991]. This topological mesh generator can deal with isotropic adaption, since [Coupez 1994], whose technique have been further used by [Zavattieri *et al.* 1996].

Meanwhile, pioneering investigations on anisotropic adaption were performed in 2D, without metric, by [Peraire *et al.* 1987, Löhner 1988, Löhner 1989] and with the use of a metric, by [Vallet 1992]. Here, we emphasize on 3D anisotropic adaption, following the work of [Coupez and Bigot 2000].

The aim of this part is to compare the topological mesh generator with classical methods (octree, advancing front and Delaunay), in what concerns generation and anisotropic adaption. The major results are a quantification of the topological mesh generator efficiency and a sensitivity study between several strategies.

Some preliminaries are given in chapter 1 (page 11), in what concerns meshes and metrics. A comparative study is then depicted between the different mesh generation methods (chapter 2 page 19). Finally, anisotropic adaption techniques, driven by a metric field, are discussed in chapter 3 (page 35).



# Chapter 1

## Meshes and metrics

We intensively use the notion of metric (section 1.1), mesh (section 1.3) and simplex (section 1.2). Here we propose some definitions and useful properties.

We denote by  $d \in \mathbb{N} \setminus \{0\}$  the number of coordinates (the spatial dimension, not to be confused with the topological dimension). We will see that most results do not depend on the dimension  $d$ . Actually, all implementations have been made dimension independent, so as to work in 2D, 3D and even in 4D.

The computational domain is a polytope denoted by  $\Omega \subset \mathbb{R}^d$  (in other words, a polygon in 2D and a polyhedron in 3D) that is bounded, internally not void and with an orientation.

### 1.1 Metric

To determine the volume of a simplex we need the notion of metric. That is the reason why we begin with the metric definition.

#### 1.1.1 Definition

**Definition 1.1** a *metric* of  $\mathbb{R}^d$  is a real matrix with  $d$  columns and  $d$  rows that is symmetrical and semi-definite (positive).

A metric can be used to measure the space. Indeed, with a metric  $M$ , the bilinear application

$$\langle x, y \rangle_M = y^\top Mx \quad (1.1)$$

is a scalar product, whose associated norm is

$$\|x\|_M = \sqrt{x^\top Mx} \quad (1.2)$$

We will use this norm to measure segment length.

#### 1.1.2 Diagonalization

We can see a metric in different ways. For us, the most useful is the following diagonalized form.

**Proposition 1.1** let  $M$  be a metric of  $\mathbb{R}^d$ , then  $M$  can be diagonalized by

$$M = R \begin{pmatrix} \frac{1}{h_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{h_d^2} \end{pmatrix} R^\top \quad (1.3)$$

where  $R$  is a rotation composed by eigenvectors of  $M$  and  $\left(\frac{1}{h_i^2}\right)_{1 \leq i \leq d}$  are the corresponding eigenvalues.

Thus, we can see a metric  $M$  as the given of  $d$  principal directions (the columns of  $R$ ) and one mesh size  $h_i$ , in each principal direction.

**Proof :**  $M$  is symmetrical, so, we can diagonalize it in the orthogonal group (at this stage,  $R$  is an orthogonal matrix). Furthermore,  $M$  is semi-definite (positive), so, its eigenvalues  $\lambda_i$  are strictly positives. Thus, there exists  $h_i \neq 0$  such that  $\lambda_i = 1/h_i^2$ . Now, for  $R$  we can choose unitary vectors (so that  $|\det(R)| = 1$ ). If  $\det(R) = -1$ , then we can choose the opposite of one eigenvector, so as to obtain  $\det(R) = 1$  (finally,  $R$  is a rotation matrix).  $\square$

Since the metric  $M$  is diagonalizable, we can consider it to any power  $\alpha \in \mathbb{R}$

$$M^\alpha = R \begin{pmatrix} \frac{1}{h_1^{2\alpha}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{h_d^{2\alpha}} \end{pmatrix} R^\top \quad (1.4)$$

Note that, with this notation  $M^{-1}$  is still the inverse of  $M$ . In this document, we will mostly use  $M$ ,  $M^{1/2}$ ,  $M^{-1}$  and  $M^{-1/2}$ .

## 1.2 Simplex

Meshes of interest are composed of triangles in 2D and tetrahedra in 3D. Here we define the notion of simplex (valid for all dimension  $d$ ).

### 1.2.1 Definition

In fact, we need a definition valid for all dimension and for all number of vertex (for example, a triangle in 3D has 3 vertices with 3 coordinates).

**Definition 1.2** with  $0 \leq k \leq d$

- a  $k$ -simplex of  $\mathbb{R}^d$  is the convex hull of  $k+1$  points of  $\mathbb{R}^d$ , called *vertices*
- a  $k$ -face of a simplex  $T$  is a  $k$ -simplex whose  $k+1$  vertices are distinct vertices of  $T$ , and, in order to simplify the following definition 1.4, we adopt this convention: a  $(-1)$ -face is the empty set.

Classical denominations are adopted in what concerns a simplex:

- a 2-simplex is a *triangle*, a 3-simplex is a *tetrahedron* and a 4-simplex is a *pentatope* (simplicial 4D meshes are composed of pentatopes)
- a  $(k-1)$ -face of a  $k$  simplex is simply a *face* of that simplex and a 1-face is simply an *edge*, whose length is called its *size*
- a simplex is *equilateral* if all its sizes are equal and *unitary equilateral* when this size is 1
- a simplex is said to be *not degenerated* when the edge vectors from one of its vertices are linearly independent (in  $\mathbb{R}^d$ )
- a simplex is a *reference* simplex when not degenerated and when the coordinates of each vertex are 0 except one, which is 1.

### 1.2.2 Simplex measure

We need to define the measure of a simplex (the volume of a tetrahedron, the area of a triangle, the length of a segment, etc.).

**Proposition 1.2** let  $T$  be a  $k$ -simplex of  $\mathbb{R}^d$  whose vertices are  $S_0, \dots, S_k$  and let  $A$  be the rectangular matrix whose columns are the edge vectors from  $S_0$  :  $A = (S_1 - S_0, \dots, S_k - S_0)$ . The measure of  $T$  is

$$|T| = \frac{\sqrt{\det(A^\top A)}}{k!} \quad (1.5)$$

Thanks to (1.5), it is possible to compute the measure of a simplex whatever type  $k$  and whatever dimension  $d$ .

**Proof :** let  $\hat{T}$  be the  $k$ -reference simplex of  $\mathbb{R}^k$ , we could recursively (on  $k$ ) prove that its measure is

$$|\hat{T}| = \frac{1}{k!} \quad (1.6)$$

Let us denote by  $f$  the affine function from  $\mathbb{R}^k$  to  $\mathbb{R}^d$ , whose matrix is  $A$ , such that  $f(\hat{T}) = T$ . From Riemannian geometry (used in this proof in an Euclidian framework), the induced metric on  $T$ , with map  $f$ , by the Euclidian canonical metric (identity matrix) that  $\mathbb{R}^d$  is equipped with, is  $M = A^\top A$ . Furthermore, according to [do Carmo 1988, Jost 1998], we have

$$|T| = \int_{\hat{T}} \sqrt{\det(M)} \, dx \quad (1.7)$$

Since  $M$  is a constant metric, it comes

$$\int_{\hat{T}} \sqrt{\det(M)} \, dx = |\hat{T}| \sqrt{\det(M)} \quad (1.8)$$

by substituting the value of  $|\hat{T}|$  and the value of  $M$ , we obtain (1.5).  $\square$

For specific values of  $k$ , we have a simplification of (1.5):

$$\begin{aligned} \text{when } k = d & \quad |T| = \frac{|\det(A)|}{d!} \\ \text{when } k = d - 1 & \quad |T| = \frac{\|S_0 S_1 \wedge \dots \wedge S_0 S_{d-1}\|}{(d-1)!} \\ \text{when } k = 1 & \quad |T| = \|S_0 S_1\| \end{aligned} \quad (1.9)$$

where  $\wedge \dots \wedge$  is the vectorial product of  $\mathbb{R}^d$  and  $\|\cdot\|$  is the canonical Euclidian norm.

Since the measure is given by (1.5), its value is positive or zero ( $|T| \geq 0$ ). The following results says that the case  $|T| = 0$  corresponds to the degenerated case.

**Corollary 1.1** a simplex  $T$  is not degenerated if and only if

$$|T| > 0 \quad (1.10)$$

This corollary gives a simple criterion to determine whether a simplex is degenerated or not.

**Proof :**

- if  $T$  is not degenerated then columns of  $A$  are linearly independent, so, the rank of  $A$  is  $k$ , the rank of  $A^\top$  and the rank of  $A^\top A$  also, which implies that  $\det(A^\top A) > 0$
- conversely, if  $|T| > 0$ , then there exists a sub-matrix of  $A$  with  $k$  columns and  $k$  rows, whose determinant is not zero, so, the rank of  $A$  is  $k$ ; it follows that the columns of  $A$  are linearly independent, so,  $T$  is not degenerated.  $\square$

### 1.2.3 Metric associated with a simplex

One of the interpolation techniques used for the natural metric (page 55) needs the computation of the canonical metric associated with a  $d$ -simplex. Here, we indicate two methods to achieve this computation.

**Proposition 1.3** let  $T$  be a  $d$ -simplex of  $\mathbb{R}^d$ , not degenerated, whose vertices are  $S_0, \dots, S_d$

- (i) there exists only one metric  $M_T$ , for which  $T$  is unitary equilateral
- (ii) let  $T^0$  with vertices  $O, S_1^0, \dots, S_d^0$  be a unitary equilateral  $d$ -simplex (in the canonical Euclidian metric, see appendix A.2.3 page 157 for its construction), then we have

$$M_T = A_T^\top A_T \quad (1.11)$$

$$\text{with the matrix } A_T = \left( S_1^0, \dots, S_d^0 \right) \left( S_1 - S_0, \dots, S_d - S_0 \right)^{-1} \quad (1.12)$$

Note that, given a metric  $M_T$ , there exists several  $d$ -simplices  $T$  that are unitary equilateral in  $M_T$ .

**Proof :** (i)  $M_T$  is solution of a the linear system

$$\left( (S_j - S_i)^\top M (S_j - S_i) = 1 \right)_{1 \leq i < j \leq d} \quad (1.13)$$

whose unknowns are the upper triangular coefficients of  $M$ ; since  $T$  is not degenerated, this system is invertible, so,  $M_T$  is unique

- (ii) since  $T$  is not degenerated, the matrix  $\left( S_1 - S_0, \dots, S_d - S_0 \right)$  is invertible, so,  $A_T$  is well defined; besides, we have

$$\|S_j - S_i\|_{A_T^\top A_T}^2 = (S_j - S_i)^\top A_T^\top A_T (S_j - S_i) \quad (1.14)$$

$$= \|A_T(S_j - S_i)\|^2 \quad (1.15)$$

$$= 1 \quad (1.16)$$

because  $A_T$  is the matrix of an affine function that transforms  $T$  in  $T^0$ ; indeed, the matrix  $\left( S_1 - S_0, \dots, S_d - S_0 \right)^{-1}$  transforms  $T$  in  $\hat{T}$  and the matrix  $\left( S_1^0, \dots, S_d^0 \right)$  transforms  $\hat{T}$  in  $T^0$ ; since  $M_T$  is unique and since matrix  $A_T^\top A_T$  satisfies the system (1.13), we have  $M_T = A_T^\top A_T$ .  $\square$

### 1.2.4 Simplex quality

The quality of a  $k$ -simplex  $T$  is measured by a shape criterion [Dompierre *et al.* 2003].

**Definition 1.3** a *shape criterion* is a scalar continuous function  $c$  that associates a not degenerated simplex  $T$  to  $c(T)$ , such that

- (i)  $0 < c(T) \leq 1$
- (ii)  $c(T) = 1$  if and only if  $T$  is equilateral
- (iii)  $c(T) \rightarrow 0$  when  $T$  degenerates
- (iv)  $c(T)$  does not change under translation, rotation, homothety and orientation changing.

Examples of shape criteria

$$c(T) = c_0 \frac{h_T}{\rho(T)} \quad (1.17)$$

$$c(T) = c_0 \frac{|T|}{h_T^k} \quad (1.18)$$

and [Baker 2000, Pébay and Baker 2001]

$$c(T) = \frac{d}{\kappa_2(M_T)} \quad (1.19)$$

where  $h_T$  is the diameter of  $T$  (its greatest edge size),  $\rho(T)$  is the inner radius of  $T$ ,  $M_T$  is the metric associated with  $T$  (previous section) and  $\kappa_2$  is the conditioning operator (the greatest eigenvalue over the smallest eigenvalue). The constant  $c_0$  is introduced, so as to make the shape criterion = 1 for equilateral simplices.

According to [Knupp 2001], other shape criterion are either equivalent to both (1.17) and (1.18) or equivalent to the third (1.19). Many other criteria do not satisfy the shape criterion definition.

In what concerns the topological mesh generator, the shape criterion (1.18) is used (however,  $h_T$  is replaced by the averaged edge size). This is also the shape criterion recommended by [Dompierre *et al.* 2003].

## 1.3 Mesh

With the simplex notion in mind, we can now define the notion of simplicial mesh. For the following chapter, we need to define a mesh not only for the computational domain  $\Omega$ , but also for its boundary, denoted by  $\partial\Omega$ . That is the reason why we write the following definition for a variety  $\mathcal{O}$ , which is, in practice,  $\Omega$  or  $\partial\Omega$ .

### 1.3.1 Definition

Let  $\mathcal{O}$  be a variety of  $\mathbb{R}^d$ , whose dimension is  $D-1$  (with  $1 \leq D \leq d+1$ ). For instance, a surface in  $\mathbb{R}^3$  is a dimension 2 variety (and  $D = 3$ ). In fact,  $D$  is the number of vertices of each simplex in the future mesh. We call  $D$  the *topological dimension* of the mesh, contrary to  $d$  the *spatial dimension*.

**Definition 1.4** let  $\mathcal{N}$  be a finite set of points belonging to  $\mathcal{O}$  (they are called *nodes*) and let  $\mathcal{T}$  be a set of  $(D-1)$ -not degenerated simplices (they are called *elements*), whose vertices belong to  $\mathcal{N}$ . We denote by  $\mathcal{F}$  the set of faces of those simplices. The couple  $(\mathcal{N}, \mathcal{T})$  is a *mesh* of  $\mathcal{O}$  if and only if:

- (i) each face of  $\mathcal{F}$  belongs to one or two elements of  $\mathcal{T}$ , but not more
- (ii) intersection of two distinct elements of  $\mathcal{T}$  is a  $k$ -face of both elements with

$$k \in \{-1, 0, \dots, D-2\} \quad (1.20)$$

- (iii) and  $\bigcup_{T \in \mathcal{T}} T = \mathcal{O}$ .

Some remarks about this definition:

- condition (i) excludes manifold meshes (like the surface mesh of figure 1.1(a), where two tetrahedra share the same edge)
- condition (ii) does not allow two distinct elements to cover each other (contrary to figure 1.1(b))
- condition (ii) forbids also any node to be in the middle of an edge (contrary to figure 1.1(c)).

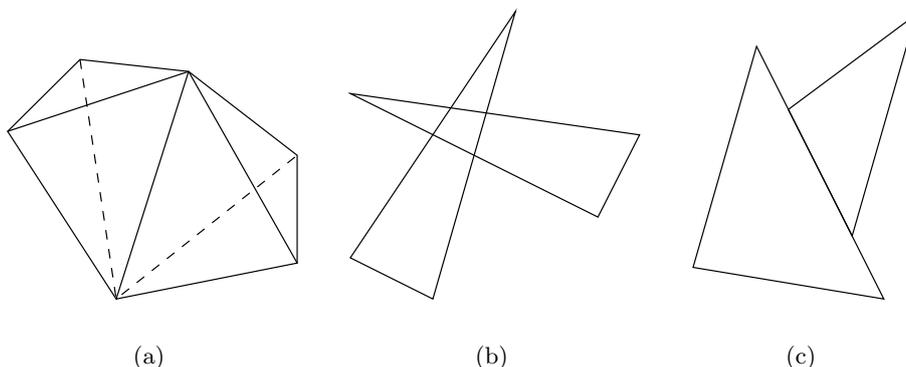


Figure 1.1: Forbidden situations for a mesh

Some classical denominations about meshes:

- the set  $\mathcal{T}$  is called the *topology* of the mesh; it is equivalent to the given of element vertices and that information can be inverted to know the connectivity of the nodes (see section A.3.1 page 160);
- the *neighbors* of a node  $S$  are the other vertices of all element that  $S$  belongs to (figure 1.2(a))
- the (topological) *boundary* of a mesh is denoted by  $\partial\mathcal{T}$  and contains the set of faces that belongs to only one element.

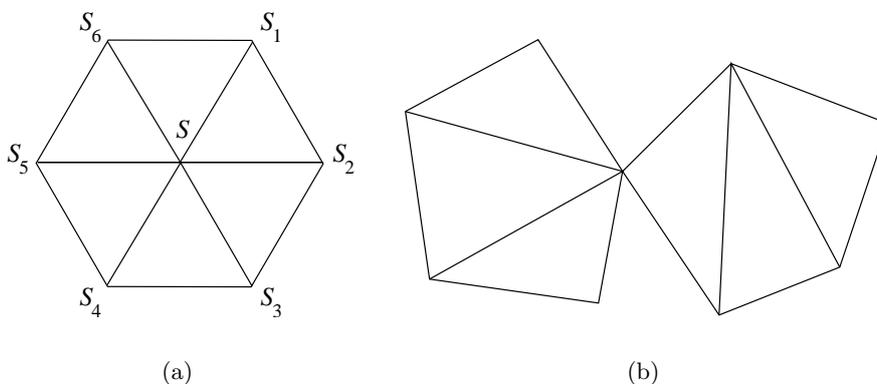


Figure 1.2:  $\{S_1, \dots, S_6\}$  are the neighbors of  $S$  (a) and a mesh whose boundary is not a mesh (b), because of condition (i) violation (a point, which is a face of the 1D boundary mesh, shares 4 elements)

We note that the boundary of a mesh of  $\Omega$  is not always a mesh of  $\partial\Omega$ , since condition (i) may not be satisfied (like in figure 1.2(b)). Furthermore, the topological boundary  $\partial\mathcal{T}$  (which is a set of faces)

should not be confused with the geometrical boundary  $\partial \left( \bigcup_{T \in \mathcal{T}} T \right)$  (which is a set of points). However, with a mesh we have the following property

$$\bigcup_{F \in \partial \mathcal{T}} F = \partial \left( \bigcup_{T \in \mathcal{T}} T \right) \quad (1.21)$$

It will not necessarily be true anymore, with the notion of mesh topology (definition 2.2 page 25).

### 1.3.2 Mesh volume

The topological mesh generator described in section 2.2 page 25 strongly rely on the computation of mesh volume (measure).

**Proposition 1.4** let  $(\mathcal{N}, \mathcal{T})$  be a mesh of  $\Omega$ . The measure of this mesh is

$$|\Omega| = \sum_{T \in \mathcal{T}} |T| \quad (1.22)$$

$$= \frac{1}{d} \int \bigcup_{F \in \partial \mathcal{T}} F x \cdot \vec{n} \, dx \quad (1.23)$$

where  $\vec{n}$  is the outgoing unitary normal vector (which is defined almost everywhere on  $F$ ).

So, we have two computation ways for the measure of the domain  $\Omega$ . In particular, the formulae (1.23) uses only the mesh boundary. The topological mesh generator relies on the fact that the equality between  $\sum_{T \in \mathcal{T}} |T|$  and  $\frac{1}{d} \int \bigcup_{F \in \partial \mathcal{T}} F x \cdot \vec{n} \, dx$  is only reached by the meshes.

**Proof :** from the Lebesgue measure definition and from condition **(iii)** of the mesh definition, we have

$$|\Omega| = \int_{\Omega} dx \quad (1.24)$$

$$= \int \bigcup_{T \in \mathcal{T}} T \, dx \quad (1.25)$$

Besides, from condition **(ii)** of the mesh definition, elements of  $\mathcal{T}$  do not cover each other, so,

$$\int \bigcup_{T \in \mathcal{T}} T \, dx = \sum_{T \in \mathcal{T}} \int T \, dx \quad (1.26)$$

$$= \sum_{T \in \mathcal{T}} |T| \quad (1.27)$$

Furthermore, from Stokes theorem [Weisstein *et al.*], since  $\nabla \cdot x = \text{div}(x) = d$  and thanks to the fact that  $\bigcup_{T \in \mathcal{T}} T = \Omega$  is a polytope with an orientation, we have

$$\int \bigcup_{T \in \mathcal{T}} T \, dx = \frac{1}{d} \int \bigcup_{T \in \mathcal{T}} T \, \nabla \cdot x \, dx \quad (1.28)$$

$$= \frac{1}{d} \int \partial \left( \bigcup_{T \in \mathcal{T}} T \right) x \cdot \vec{n} \, dx \quad (1.29)$$

which leads to (1.23) by combination with (1.21).  $\square$

### 1.3.3 Mesh conformity

After an anisotropic adaption to a metric field  $M$  (chapter 3 page 35), the obtained mesh does never respect perfectly the anisotropy of  $M$ . So as to quantify the conformity of a mesh with a metric field  $M$ , [Labbé *et al.* 2000, Labbé *et al.* 2001, Dompierre *et al.* 2003] have proposed a non conformity coefficient.

Let  $(\mathcal{N}, \mathcal{T})$  be a mesh, for all  $T \in \mathcal{T}$  we denote by

$$M'_T = \frac{1}{|T|} \int_T M(x) dx \quad (1.30)$$

the averaged value of  $M$  on  $T$  and we still denote by  $M_T$  the associated metric of  $T$  (section 1.2.3 page 14).

If the mesh was perfectly conforming the metric field  $M$ , then we would have for all  $T \in \mathcal{T}$ ,  $M_T = M'_T$ . It is thus proposed to look at  $\epsilon_T$ , the *Frobenius norm* of

$$R_T = \underbrace{\left( M_T^{-1} M'_T - I \right)}_{\text{detects } |T| \rightarrow 0} + \underbrace{\left( M_T'^{-1} M_T - I \right)}_{\text{detects } |T| \rightarrow \infty} \quad (1.31)$$

in other words

$$\epsilon_T = \sqrt{\text{Trace} \left( R_T^\top R_T \right)} \quad (1.32)$$

The non conformity coefficient is then defined by

$$\epsilon_{\mathcal{T}} = \frac{1}{\text{card}(\mathcal{T})} \sum_{T \in \mathcal{T}} \epsilon_T \quad (1.33)$$

In this way, one element that degenerates makes  $\epsilon_{\mathcal{T}}$  tend to  $\infty$  (which is a good point, because such a mesh is not suitable for a finite element computation).

This coefficient takes into account the size and the shape of the elements. It does not suffer from scaling and it is defined in all dimensions. That is the reason why we use it to validate the topological mesh generator in section 3.2 page 39.

### Memory allocation coefficient

With the same notations, we can estimate the size of the mesh after adaption from the size of the mesh before adaption and from the metric used for the adaption. We can use the actual edge sizes in the mesh and the edge sizes prescribed by the metric to evaluate the growing factor to be used to allocate the required workspace.

We denote by  $\lambda_i(M_T)$ , the eigenvalues of  $M_T$  and by  $\lambda_i(M'_T)$ , those of  $M'_T$ . The proposed growing factor is the integer part +1 of

$$c_{\text{allocation}} = \frac{1}{\text{card}(\mathcal{T})} \sum_{T \in \mathcal{T}} \prod_{i=1}^d \sqrt{\frac{\lambda_i(M'_T)}{\lambda_i(M_T)}} \quad (1.34)$$

In practice, this growing factor overestimates the workspace needed for the refining or the coarsening. It gives only a useful upper bound.

## Chapter 2

# Meshing a domain from a mesh of its boundary

In this chapter, we deal with the following question: given a mesh of the boundary  $\partial\Omega$ , how to build a mesh of  $\Omega$  itself? We focus our attention on unstructured meshes composed of simplices.

In 3D, there exists four major ways to answer this question. Classical ways are: meshing by tree, advancing front and Delaunay method (section 2.1). The fourth way is the topological mesh generator, which has been introduced by [Coupez 1991]. There are several advantages to use the latter (section 2.2).

### 2.1 State of the art: three classical methods

Let us recall the three classical mesh generation methods.

#### 2.1.1 Meshing by tree

Introduced by [Yerry and Shephard 1983], the meshing technique by tree is widely used, even in 3D [Tchon *et al.* 2003].

A mother cell (a rectangle in 2D or a parallelepiped in 3D), englobing the domain  $\Omega$ , is considered. The first step is to decompose recursively each cell containing several nodes in  $2^d$  equivalent cells (figure 2.1(b)). A simple convergence criterion of this process could be: each cell contains not more than one node. It gives a tree structure with  $2^d$  branches, that is, a quadtree in 2D and an octree in 3D.

Then, the method consists in:

- removing exterior cells
- triangulating the cells that contain one node with respect to the boundary mesh
- removing exterior elements
- triangulating empty cells, whose vertices are new nodes (figure 2.1(c)).

With this technique, some problems may be faced:

- one node could lie near the border or at the corner of a cell, which could lead to bad quality elements
- the aforementioned convergence criterion is too poor, because a cell could contain one node and be traversed by another face
- numerous useless nodes are introduced on the boundary

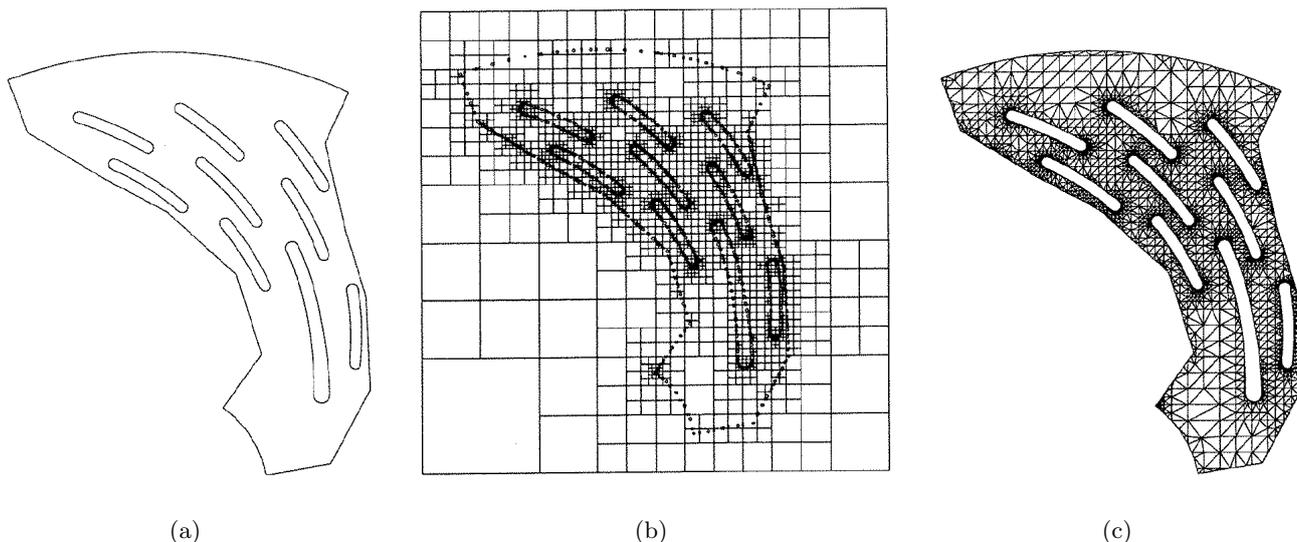


Figure 2.1: Quadtree (from [Frey and George 1999])

- to avoid an ill balanced tree, the  $2 : 1$  rule is often enforced (this rule stipulates that the size factor between two adjacent cells is not more than 2), which introduces far too much nodes in the mesh.

However, the principal advantage of this technique is the tree structure, which is very useful (especially for localization computations in the mesh). Its main drawback is the quasi impossibility to deal with anisotropic elements.

### 2.1.2 Advancing front

Introduced by [Lo 1985], the advancing front method is also widely used and one could cite, for example, the work of [Bonet and Peraire 1991].

In this method, new nodes are created by connecting them to existing elements, with an equilateral goal. A set of faces, which is said to be the front, scans progressively all the domain  $\Omega$  (figure 2.2 steps i to iii). The process stops when the front is empty, that is, when the domain is completely triangulated (figure 2.2 step iv).

More precisely, major steps of the method are the following:

- initially, the front is the mesh of the boundary
- then, a good quality element is built from a face of the front (a new node can be introduced)
- if this new element intersects the mesh already built, then another node is chosen, otherwise the element is added in the mesh
- we get a new front and the process is iterated, while the front is not empty.

Major difficulties encountered by this techniques are summarized by these questions:

- which face to choose among those of the front?
- how to determine the best position of a new node? (especially in 3D)
- how to determine efficiently if an element intersects the mesh?

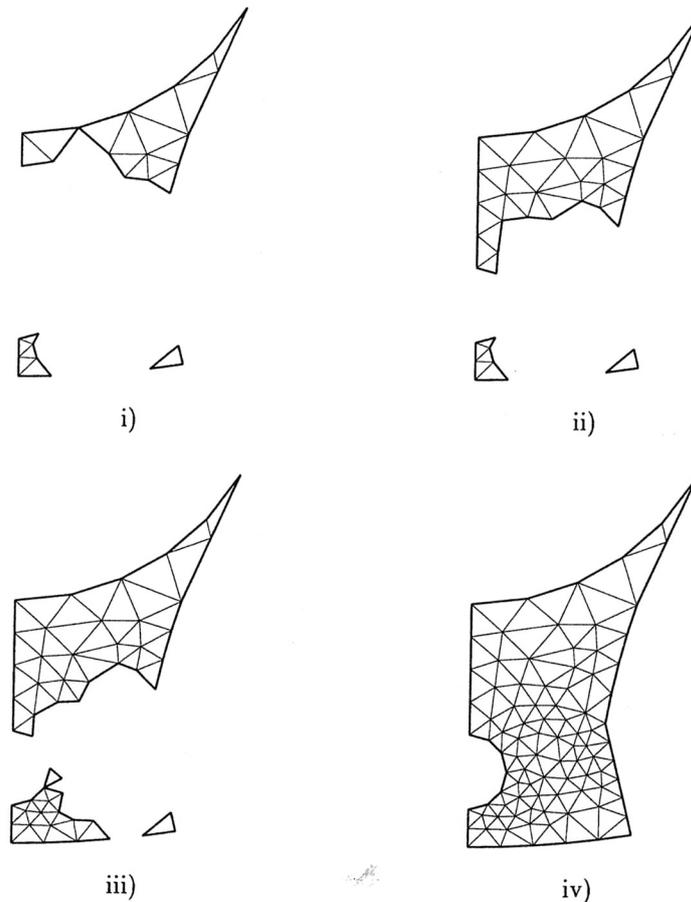


Figure 2.2: Advancing front in 2D (from [Frey and George 1999])

In what concerns the front face choice, the smallest one or the poorest one could be chosen. In what concerns the position of a new node, candidates are (in descending priority order): surrounding existing nodes, the optimal node(s) (attaining the best reachable quality) and, finally, candidates given a priori. In what concerns the intersection computation, also called front collision, it is generally a costly operation. An efficient localization computation is needed (see page 24).

The advancing front is the best technique to produce good quality elements. However, the drawbacks of this method are: the front can be very noisy (even for simple geometry, especially in 3D) and convergence may not be attained (except in 2D). Robustness is not ensured with the advancing front technique.

### 2.1.3 Delaunay method

Before the Delaunay mesh generator is detailed, some important results are given here (proofs are available in [George and Borouchaki 1997]).

#### 2.1.3.1 Delaunay triangulation

Firstly, let us look at the existence and the construction of a mesh from a point cloud. Let  $\mathcal{N}$  be a finite set of points  $(S_i)_{1 \leq i \leq n}$  from  $\mathbb{R}^d$ .

**Definition 2.1** before we talk about Delaunay triangulation, we introduce the Voronoï diagram with those definitions:

- a *d-convex polytope* is the convex hull of several points from  $\mathbb{R}^d$  (it generalizes the notion of convex polygon in 2D and of convex polyhedron in 3D)
- the *Voronoï diagram* of  $\mathcal{N}$  is the set of all *d-convex polytopes*  $(V_i)_{1 \leq i \leq n}$  called *Voronoï cells* and defined by

$$V_i = \left\{ x \in \mathbb{R}^d, \forall j \neq i \quad \|S_i x\|_2 \leq \|S_j x\|_2 \right\} \quad (2.1)$$

that is, the points that are closer to  $S_i$  than to any other  $S_j$ .

A Voronoï diagram has several very interesting properties.

- Proposition 2.1**
- the Voronoï cells cover  $\mathbb{R}^d$ , without covering each other (figure 2.3 on the left)
  - if the points of  $\mathcal{N}$  are not cocyclical (that is,  $d + 2$  of them are never aligned on the same ball), then the dual  $\mathcal{T}$  of the Voronoï diagram is a set of simplices
  - otherwise this dual contains some *d-polytopes*, which could be decomposed in simplices, so as to obtain a set of simplices, still denoted by  $\mathcal{T}$
  - $\mathcal{T}$  is called the *Delaunay triangulation* and constitutes, with  $\mathcal{N}$ , a mesh of the convex hull of  $\mathcal{N}$  (figure 2.3 on the right).

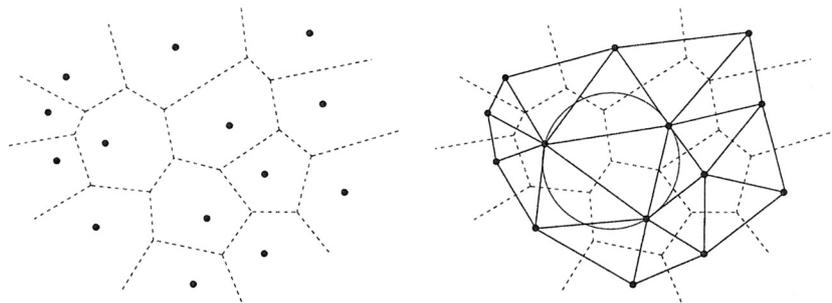


Figure 2.3: Voronoï diagram of a set of points and corresponding Delaunay triangulation in 2D (from [Frey and George 1999])

In fact, to construct the Delaunay triangulation, the Voronoï diagram is not required. In practice, the following criterion is used to obtain a Delaunay triangulation.

**Theorem 2.1** (Delaunay criterion) Let  $\mathcal{T}$  be a triangulation of the convex hull of  $\mathcal{N}$ .  $\mathcal{T}$  is a Delaunay triangulation of  $\mathcal{N}$  if and only if the interior of the outer ball of each element does not contain any node.

The problem is that  $\Omega$  is rarely the convex hull of the nodes  $\mathcal{N}$ . So, we need to modify the dual of the Voronoï diagram, so as to obtain a mesh of  $\Omega$  (next section).

### 2.1.3.2 Delaunay mesh generation

A Delaunay mesh generation method consists in:

- triangulating the convex hull of the nodes (figure 2.4(a)) in two steps:
  - triangulate a bounding box containing all the nodes in its interior (in 2D, this bounding box is a rectangular decomposed in two triangles, while in 3D, it is a parallelepiped decomposed in 5 or 6 tetrahedra)
  - introducing each node  $S_n$  one by one, in the triangulation by using the Delaunay kernel:
    - \* remove all elements whose outer ball contains  $S$  (these adjacent elements forms the Delaunay cavity)
    - \* add new elements built with  $S$  and the remaining faces of the removed elements (this operation is called the *star operation*)
- making the boundary mesh to be respected by:
  - edge swapping in 2D (figure 2.4(b)) or edge and face swapping in 3D
  - introduction of new nodes if required
- removing exterior elements (by coloring).

Coloring the exterior elements can be achieved by:

- choosing a vertex of the bounding box
- considering all elements of this vertex as exterior
- adjacent elements, except through a boundary face, are also exterior.

We get one connex compound by this first pass. Several passes are required to get all the connex compounds of the exterior.

This technique raises several difficulties:

- centers and radii of all outer balls have to be computed and stored (this computation involves a linear system inversion; except if transport of those quantities is possible)
- the Delaunay cavity is not that easy to compute, because it may involve complex situations
- several algorithm steps require the localization of a point in the triangulation (see the following paragraph)
- in 2D, the boundary can always be recovered, because edge swapping can visit all possibilities; but in 3D, it is no more the case and boundary recovering can be impossible without relaxing the boundary mesh
- a special tetrahedron, called a *sliver*, who tends to degenerate (without any edge or face degenerating) and who still satisfies the Delaunay criterion (this criterion is not a quality criterion); unfortunately in 3D, a Delaunay mesh generator can introduce such undesirable elements in the mesh.

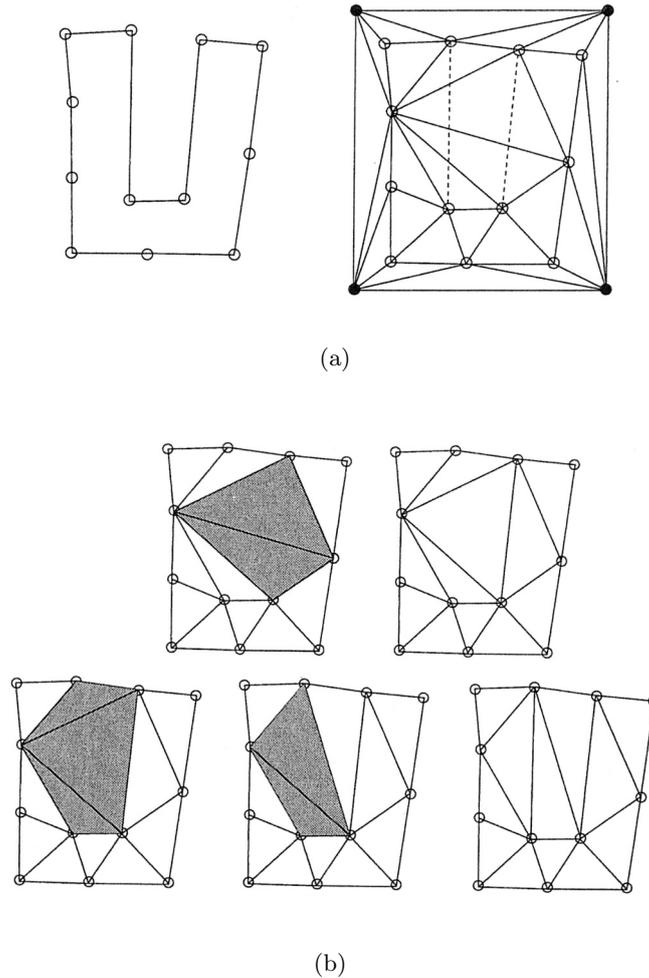


Figure 2.4: Delaunay method in 2D: triangulation of the convex hull (a) and recovering of the boundary (b) (from [Ern and Guermond 2002])

The localization of a point  $S$  (which is always a costly operation) can be efficiently driven as follows:

- let  $T$  be an element that does not contain  $S$ , then  $S$  sees at least one face  $F$  of  $T$ ; the other element, to which  $F$  belongs is tested and we continue until we find an element containing  $S$  (it works well when we have a mesh of the convex hull of the domain (because there is no hole between  $S$  and  $T$ ), which is the case with a Delaunay mesh generator)
- if  $S$  sees several faces  $F$ , then one of them is chosen randomly (otherwise it could loop forever)
- the first tested element  $T$  should not be far away from  $S$ , thus, a tree structure of initial elements (even coarse) would be helpful.

A robust and very efficient implementation of such a Delaunay mesh generator has been achieved at INRIA Rocquencourt by the Gamma team. But this approach is very technical and a neophyte would face a lot of numerical errors (especially in what concerns geometrical computations, like centers and radii of outer balls).

## 2.2 Topological optimization

The three preceding techniques are often coupled with a last step that consists in improving the quality of the mesh (like a Laplacian smoothing [Frey and Field 1991] or a more elaborated method). The alternative, depicted in this section, relies only on that optimization step. The method has been introduced by [Coupez 1991], when all current 3D mesh generators were investigated.

This method is the topological mesh generator, already described in [Coupez 2000] and also in [Coupez and Bigot 2000]. Here, we propose a new and more general description.

### 2.2.1 Mathematical background

Before the algorithm is detailed, let us prove that a mesh is the solution of an optimization problem, which justifies the use an optimization strategy to build it.

#### 2.2.1.1 Mesh topology

A mesh is a very difficult objet to construct directly (especially when  $d \geq 3$ ). Let us consider the less constrained objet of the following definition.

**Definition 2.2** let  $\mathcal{N}$  be a finite set of nodes from  $\Omega$ , let  $\mathcal{T}$  be a set of  $d$ -simplices whose vertices belong to  $\mathcal{N}$ . We denote by  $\mathcal{F}$  the set of faces of those simplices.  $\mathcal{T}$  is a *mesh topology* of  $\Omega$  if and only if:

- (i) each face of  $\mathcal{F}$  belongs to one or two elements of  $\mathcal{T}$ , but not more
- (ii) and  $(\mathcal{N}, \partial\mathcal{T})$  is a mesh of the boundary  $\partial\Omega$ .

Of course, the topology of a mesh is a mesh topology (except when condition **(iii)** of definition 1.4 is not fulfilled for the mesh of the boundary). Our goal is to exhibit a sufficient condition for a mesh topology to be a mesh.

Some remarks about this definition:

- condition **(i)** is the same than condition **(i)** of the mesh definition
- again, in condition **(ii)**, the boundary  $\partial\mathcal{T}$  of a mesh topology  $\mathcal{T}$  is the set of faces that belongs to only one element.

Only one operator is needed to build a mesh topology, namely the star operator. It consists in connecting one node  $S$  to a set of faces  $\mathcal{F}$ , by considering

$$\mathcal{T}^*(S, \mathcal{F}) = \{\text{convex hull of } \{S\} \cup F, \text{ where } F \in \mathcal{F} \text{ and } S \notin F\} \quad (2.2)$$

The following property says that the star operator build a mesh topology when the faces are the boundary of a volume.

**Proposition 2.2** let  $S$  be a node and  $\mathcal{F}$  be a set of faces. If  $\mathcal{F}$  is a mesh topology without boundary (that is,  $\partial\mathcal{F} = \emptyset$ ), then  $\mathcal{T}^*(S, \mathcal{F})$  is a mesh topology and its boundary is  $\partial(\mathcal{T}^*(S, \mathcal{F})) = \mathcal{F}$ .

Note that this property uses a mesh topology with a set of  $(d-1)$ -simplices (containing the faces). For this case, the definition is the same than definition 2.2.

**Proof :** suppose that there exists  $F_0$  a face of  $\mathcal{T}^*(S, \mathcal{F})$  who belongs to three elements:  $T, T'$  and  $T''$ , which are convex hulls of  $\{S\}$  and respectively  $F, F'$  et  $F''$ . Then, we have  $F \cap F_0 = F' \cap F_0 = F'' \cap F_0 = G$ . In this context,  $G$  belongs to three distinct elements of  $\mathcal{F}$ , which is impossible because  $\mathcal{F}$  is a mesh topology.

Furthermore, by construction we have  $\mathcal{F} \subset \partial\mathcal{T}^*(S, \mathcal{F})$ . For the reverse inclusion, suppose that there exists a face  $F$  of  $\partial\mathcal{T}^*(S, \mathcal{F})$ , such that  $F \notin \mathcal{F}$ . Then  $F$  is a face of the convex hull of  $\{S\} \cup F'$ , where  $F' \in \mathcal{F}$ . We denote by  $G$  the intersection  $F \cap F'$ . As  $\mathcal{F}$  is without boundary, there exists  $F'' \neq F' \in \mathcal{F}$ , such that  $G$  belong to  $F'$  and  $F''$ . In this context,  $F$  belong to the convex hull of  $\{S\} \cup F'$  and of  $\{S\} \cup F''$ , which are distinct. So,  $F \notin \partial\mathcal{T}^*(S, \mathcal{F})$ , which is impossible.  $\square$

### 2.2.1.2 Minimal volume theorem

Thanks to condition (ii) of definition 2.2, we always have

$$|\Omega| = \frac{1}{d} \int \bigcup_{F \in \partial\mathcal{T}} F x \cdot \vec{n} dx \quad (2.3)$$

However, the elements of a mesh topology can be degenerated, can cover each other and can run outside the domain  $\Omega$ . So, with this definition we do not have  $|\Omega| = \sum_{T \in \mathcal{T}} |T|$ , but only the following result.

**Lemma 2.1** if  $\mathcal{T}$  is a mesh topology of  $\Omega$ , then

$$\Omega \subset \bigcup_{T \in \mathcal{T}} T \quad (2.4)$$

In other words, thanks to condition (ii) of definition 2.2, each point of the computational domain  $\Omega$  lies inside, at least, one element of  $\mathcal{T}$ .

**Proof :** let us consider the front series  $(\Gamma_n)_{n \in \mathbb{N}}$  and the topology series  $(\mathcal{T}_n)_{n \in \mathbb{N}}$  defined by:

$$\begin{cases} \Gamma_0 = \partial\mathcal{T} \\ \mathcal{T}_0 = \emptyset \end{cases} \quad (2.5)$$

$$\forall n \in \mathbb{N} \quad \text{if } \Gamma_n = \emptyset \text{ then } \begin{cases} \Gamma_{n+1} = \Gamma_n \\ \mathcal{T}_{n+1} = \mathcal{T}_n \end{cases} \quad (2.6)$$

$$\text{otherwise } \begin{cases} \Gamma_{n+1} = \Gamma_n \Delta \partial\{T_n\} \\ \mathcal{T}_{n+1} = \mathcal{T}_n \cup \{T_n\} \end{cases} \quad (2.7)$$

where  $T_n$  is an element of  $\mathcal{T} \setminus \mathcal{T}_n$  who has one face belonging to  $\Gamma_n$ . In this proof, we use the set operator  $\Delta$  defined by  $A \Delta B = (A \cup B) \setminus (A \cap B)$ .

In other words, we consider an advancing front that scans progressively the mesh topology, from the boundary. Our goal is to prove that this front scans all points of  $\Omega$ .

Since  $\mathcal{T}$  is a finite set, there exists  $N \in \mathbb{N}$ , such that  $\Gamma_{N-1} \neq \emptyset$  and such that  $\forall n \geq N \quad \Gamma_n = \emptyset$ . Firstly, we can prove that

$$\forall 0 \leq n < N \quad \Gamma_n = \partial(\mathcal{T} \setminus \mathcal{T}_n) \quad (2.8)$$

which ensures that the front series is well defined. For  $n = 0$ , we have  $\Gamma_0 = \partial\mathcal{T} = \partial(\mathcal{T} \setminus \mathcal{T}_0)$ , since  $\mathcal{T}_0 = \emptyset$ . For  $0 \leq n < N$ , we suppose that  $\Gamma_n = \partial(\mathcal{T} \setminus \mathcal{T}_n)$  et we prove that  $\Gamma_{n+1} = \partial(\mathcal{T} \setminus \mathcal{T}_{n+1})$ .

Let us start by  $\Gamma_{n+1} \subset \partial(\mathcal{T} \setminus \mathcal{T}_{n+1})$ . We take  $F \in \Gamma_{n+1}$  and we distinguish two cases:

- if  $F \in \Gamma_n$ , then  $F \notin \partial\{T_n\}$  (since  $\Gamma_{n+1} = \Gamma_n \Delta \partial\{T_n\}$ ). So, there exists a unique element  $T$  in  $\mathcal{T} \setminus \mathcal{T}_n$ , different from  $T_n$  and such that  $F \in \partial\{T\}$ . Thus,

$$F \in \partial(\mathcal{T} \setminus \mathcal{T}_n \setminus \{T_n\}) = \partial(\mathcal{T} \setminus \mathcal{T}_{n+1}) \quad (2.9)$$

- if  $F \notin \Gamma_n$ , then  $F \in \partial\{T_n\}$  and  $F \notin \partial\mathcal{T}$ . So, condition (i) of definition 2.2 allows us to say that there exists only one element  $T$  of  $\mathcal{T}$ , different from  $T_n$  and such that  $F \in \partial\{T\}$ ; we have  $T \notin \mathcal{T}_n$ , otherwise  $F \in \partial(\mathcal{T} \setminus \mathcal{T}_n) = \Gamma_n$ . Thus,

$$F \in \partial(\mathcal{T} \setminus \mathcal{T}_n \setminus \{T_n\}) = \partial(\mathcal{T} \setminus \mathcal{T}_{n+1}) \quad (2.10)$$

Now, we have to treat the reverse inclusion  $\partial(\mathcal{T} \setminus \mathcal{T}_{n+1}) \subset \Gamma_{n+1}$ . We take  $F \in \partial(\mathcal{T} \setminus \mathcal{T}_{n+1})$  and we know that there exists a unique element  $T \in \mathcal{T} \setminus \mathcal{T}_n \setminus \{T_n\}$ , such that  $F \in \partial\{T\}$ . Again, we distinguish two cases:

- if  $F \in \partial\{T_n\}$ , then  $F \notin \partial(\mathcal{T} \setminus \mathcal{T}_n)$ , since  $T \in \mathcal{T} \setminus \mathcal{T}_n$  and  $T_n \in \mathcal{T} \setminus \mathcal{T}_n$ , so

$$F \in \partial(\mathcal{T} \setminus \mathcal{T}_n) \Delta \{T_n\} = \Gamma_n \Delta \{T_n\} = \Gamma_{n+1} \quad (2.11)$$

- if  $F \notin \partial\{T_n\}$ , then  $T$  is unique in  $\mathcal{T} \setminus \mathcal{T}_n$ , so,  $F \in \partial(\mathcal{T} \setminus \mathcal{T}_n)$ , thus

$$F \in \partial(\mathcal{T} \setminus \mathcal{T}_n) \Delta \{T_n\} = \Gamma_n \Delta \{T_n\} = \Gamma_{n+1} \quad (2.12)$$

We get the equality (2.8). Thus, the front series  $(\Gamma_n)_{n \in \mathbb{N}}$  is well defined (because, if  $\Gamma_n \neq \emptyset$ , then  $\partial(\mathcal{T} \setminus \mathcal{T}_n) \neq \emptyset$ , so, there exists  $T_n \notin \mathcal{T}_n$  such that  $\Gamma_n \cap \partial\{T_n\} \neq \emptyset$ ).

Secondly, we prove that

$$\forall 0 \leq n < N \quad \partial(\Omega \Delta T_0 \Delta \dots \Delta T_{n-1}) \subset \bigcup_{F \in \Gamma_n} F \quad (2.13)$$

which gives an upper bound (for the set relationship order) of  $\Omega \setminus (T_0 \cup \dots \cup T_N)$ , by the empty set. For  $n = 0$ , since  $(\mathcal{N}, \partial\mathcal{T})$  is a mesh of  $\partial\Omega$ , we have  $\partial\Omega = \bigcup_{F \in \partial\mathcal{T}} F = \bigcup_{F \in \Gamma_0} F$ . For  $0 \leq n < N$ , we assume that

$$\partial(\Omega \Delta T_0 \Delta \dots \Delta T_{n-1}) \subset \bigcup_{F \in \Gamma_n} F \quad (2.14)$$

and we prove that

$$\partial(\Omega \Delta T_0 \Delta \dots \Delta T_n) \subset \bigcup_{F \in \Gamma_{n+1}} F \quad (2.15)$$

Let us begin with the proof of

$$\left( \partial(\Omega \Delta T_0 \Delta \dots \Delta T_{n-1}) \Delta \partial T_n \right) \subset \bigcup_{F \in \Gamma_{n+1}} F \quad (2.16)$$

We take a point  $x \in \left( \partial(\Omega \Delta T_0 \Delta \dots \Delta T_{n-1}) \Delta \partial T_n \right)$  and we distinguish two cases:

- if  $x \in \partial(\Omega \Delta T_0 \Delta \dots \Delta T_{n-1})$  and  $x \notin \partial T_n$ , then there exists  $F \in \Gamma_n$ , such that  $x \in F$  and such that  $\forall F' \in \partial\{T_n\}$  we have  $x \notin F'$ , so,  $F \in \Gamma_{n+1}$
- if  $x \notin \partial(\Omega \Delta T_0 \Delta \dots \Delta T_{n-1})$  and  $x \in \partial T_n$ , then there exists  $F \in \partial\{T_n\}$ , such that  $x \in F$  and such that  $\forall F' \in \Gamma_n$  we have  $x \notin F'$ , so,  $F \in \Gamma_{n+1}$ .

Next, since  $\bigcup_{F \in \Gamma_{n+1}} F$  is a closed set, the inclusion (2.16) is also valid for the closure

$$\overline{\partial(\Omega \triangle T_0 \triangle \dots \triangle T_{n-1}) \triangle \partial T_n} \subset \bigcup_{F \in \Gamma_{n+1}} F \quad (2.17)$$

We use the formulae  $\partial(A \triangle B) = \overline{\partial A \triangle \partial B}$ , so as to obtain (2.15).

Thus, we have recursively proved that

$$\partial(\Omega \triangle T_0 \triangle \dots \triangle T_N) \subset \bigcup_{F \in \Gamma_N} F = \emptyset \quad (2.18)$$

and, as  $(\Omega \triangle T_0 \triangle \dots \triangle T_N)$  is a bounded set, it comes  $\Omega \triangle T_0 \triangle \dots \triangle T_N = \emptyset$ .

Lastly, since  $(A \setminus B) \subset (A \triangle B)$ , we have proved that

$$(\Omega \setminus (T_0 \cup \dots \cup T_N)) \subset (\Omega \triangle T_0 \triangle \dots \triangle T_N) \quad (2.19)$$

Thus,  $\Omega \setminus (T_0 \cup \dots \cup T_N) = \emptyset$ . □

Thanks to this lemma, we do not need the orientation assumption of the mesh topology, contrary to [Coupez 2000], in what concerns the proof of the following theorem.

**Theorem 2.2** let  $\mathcal{T}$  be a mesh topology whose vertices are  $\mathcal{N}$ , a finite node set of  $\Omega$ .  $(\mathcal{N}, \mathcal{T})$  is a mesh of  $\Omega$  if and only if the simplices of  $\mathcal{T}$  are not degenerated and

$$\sum_{T \in \mathcal{T}} |T| = |\Omega| \quad (2.20)$$

In other words, among all mesh topologies, meshes are those who satisfy the equality (2.20), also known as the *minimal volume criterion*.

**Proof :** we only have to prove that if  $\sum_{T \in \mathcal{T}} |T| = |\Omega|$  then  $(\mathcal{N}, \mathcal{T})$  is a mesh of  $\Omega$ . Condition **(i)** of the mesh definition 1.4 being already satisfied, we have to treat conditions **(ii)** and **(iii)**.

Firstly, let us suppose that  $\bigcup_{T \in \mathcal{T}} T \neq \Omega$ . Since  $\Omega \subset \bigcup_{T \in \mathcal{T}} T$ , it would come  $\left| \bigcup_{T \in \mathcal{T}} T \right| > |\Omega|$ . But, we still have  $\sum_{T \in \mathcal{T}} |T| \geq \left| \bigcup_{T \in \mathcal{T}} T \right|$ , which would raise a contradiction. Thus, condition **(ii)** is satisfied.

Secondly, let us suppose that there exists  $T_1 \neq T_2 \in \mathcal{T}$  such that  $\overset{\circ}{T}_1 \cap \overset{\circ}{T}_2 \neq \emptyset$ . Then, we would have  $|T_1| + |T_2| > |T_1 \cup T_2|$ . We would obtain  $\sum_{T \in \mathcal{T}} |T| > \left| \bigcup_{T \in \mathcal{T}} T \right|$ , which would raise the same contradiction. Thus, elements do not cover each other.

Thirdly, let us suppose that there exists  $T_1 \neq T_2 \in \mathcal{T}$ , such that  $\partial T_1 \cap \partial T_2$  is not a  $k$ -face of  $T_1$  and  $T_2$  with  $k \in \{-1, 0, \dots, d-1\}$ . We could denote by  $F_1$  a face de  $T_1$  and by  $F_2$  a face of  $T_2$ , such that  $F_1 \cap F_2$  would not be a  $k$ -face of  $F_1$  and  $F_2$ , with  $k \in \{-1, 0, \dots, d-2\}$ . If  $F_1 \notin \partial\{\mathcal{T}\}$ , then there would exists  $T_3 \in \mathcal{T} \setminus \{T_1, T_2\}$ , with  $F_1$  as a face of  $T_3$ . Since  $T_3$  is not degenerated,  $T_3$  would cover  $T_2$ , which raises the same contradiction. So, we have  $F_1 \in \partial\mathcal{T}$  and, by the same way, we have  $F_2 \in \partial\mathcal{T}$ . But, since  $F_1 \cap F_2$  is not supposed to be a  $k$ -face of  $F_1$  and  $F_2$ , with  $k \in \{-1, 0, \dots, d-2\}$ ,  $(\mathcal{N}, \partial\mathcal{T})$  could not be a mesh of  $\partial\Omega$ . This contradiction with condition **(ii)** of the mesh topology definition, induces the condition **(iii)** of the mesh definition. □

### 2.2.1.3 Shape criterion and order relationship

In the next section, from two distinct mesh topologies with the same volume, we shall prefer the one whose elements show a better quality. The quality of a simplex  $T$  can be measured with the shape criterion

$$c(T) = c_0 \frac{|T|}{h(T)^d} \quad (2.21)$$

where  $h(T)$  is the averaged edge size.

This shape criterion allows the introduction of a relationship order between mesh topologies.

**Definition 2.3** let  $\mathcal{T}_1 = (T_1^i)_{1 \leq i \leq I_1}$  and  $\mathcal{T}_2 = (T_2^i)_{1 \leq i \leq I_2}$  be two mesh topologies, whose elements are ordered such that

$$\forall 1 \leq i_1 < i_2 \leq I_j \quad c(T_j^{i_1}) \leq c(T_j^{i_2}) \quad (2.22)$$

(in other words, elements of mesh topologies  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are ordered by ascending shape criterion, in order to compare worst elements at first). We consider the *lexicographical order*, denoted by  $<$  and defined by

$$\mathcal{T}_1 < \mathcal{T}_2 \quad \Leftrightarrow \quad \exists 1 \leq i_0 \leq \min(I_1, I_2), \text{ such that } \begin{cases} \forall i < i_0 & c(T_1^i) = c(T_2^i) \\ \text{and} & c(T_1^{i_0}) < c(T_2^{i_0}) \end{cases} \quad (2.23)$$

## 2.2.2 Mesh generation

Now we can propose a technique to build a mesh of  $\Omega$  from a mesh of  $\partial\Omega$ . In fact, like in the Delaunay method, we proceed by local cut-and-past operations.

### 2.2.2.1 Local topology optimization

For the following algorithm, we make the assumption that the boundary of  $\Omega$  is connex. However, this assumption can be removed with the mesh generation approach, derived from the multidomain metric (section 7.4 page 132).

**Algorithm 2.1** let  $(\mathcal{N}, \partial\mathcal{T})$  be a mesh of  $\partial\Omega$ , the topological mesh generator proceeds like this:

– build a very first mesh topology  $\mathcal{T} = \mathcal{T}^*(S_0, \partial\mathcal{T})$  of  $\Omega$ , where  $S_0$  is a node of the boundary mesh (if  $\Omega$  is not convex, then  $(\mathcal{N}, \mathcal{T})$  is not a valid mesh, like on figure 2.5 top left)

**while** the minimal volume is not reached **do**

**for** each node and each edge **do**

    – remove the local topology  $\mathcal{T}_c$  around this node or this edge (a cavity)

    – replace it by a topology  $\mathcal{T}_e$  that minimizes the volume  $\sum_{T \in \mathcal{T}} |T|$

      and that is maximal in the relationship order (2.23),  
      among all candidates generated by the star operator

**done**

**done**

Iterating around each node and each edge is enough. There is no need to iterate around each element.

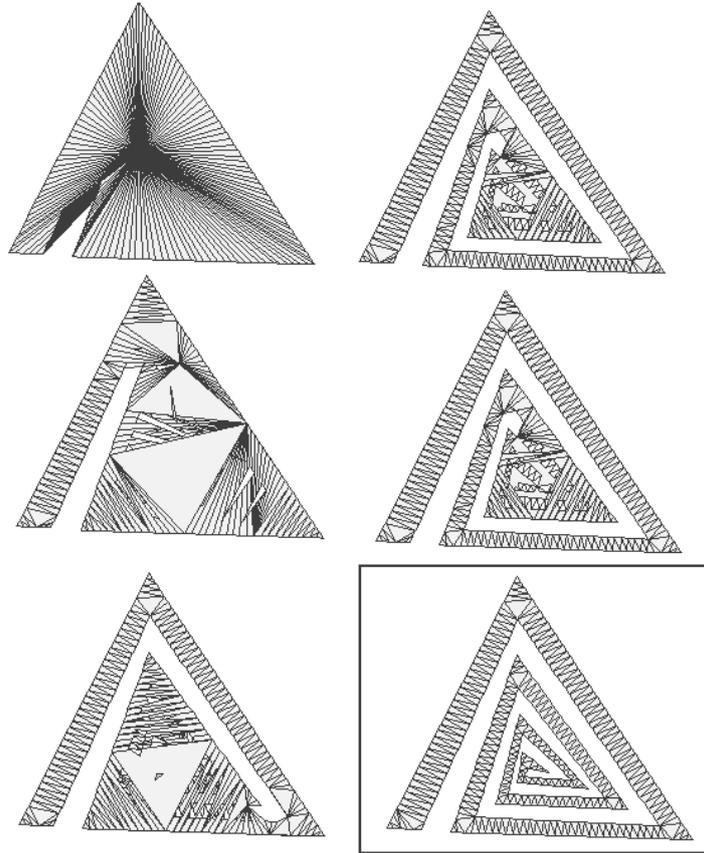
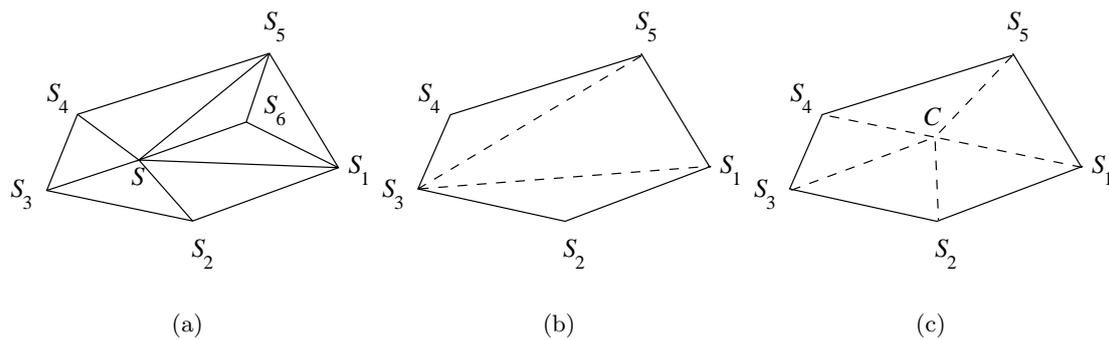


Figure 2.5: Mesh generation by topological optimization in 2D

The local topology  $\mathcal{T}_c$  around a node  $S$  is composed by the elements of  $\mathcal{T}$ , whose all vertices belong to  $\mathcal{S} = \{S\} \cup \mathcal{N}(S)$ , where  $\mathcal{N}(S)$  is the neighborhood of  $S$  (according to this definition, the triangle  $S_1S_5S_6$  on figure 2.6(a) belongs to  $\mathcal{T}_c$ ). In this case, candidates for  $\mathcal{T}_e$  are the topologies  $\mathcal{T}^*(S'', \partial\mathcal{T}_c)$ , where  $S''$

Figure 2.6: The local topology  $\mathcal{T}_c$  around a node  $S$  (a) and two candidates for  $\mathcal{T}_e$  (b-c).

is:

- $S$  itself (which leads to the elimination of  $S_6$  on figure 2.6)
- one of the vertices of  $\partial\mathcal{T}_c$  (elimination of  $S$  and  $S_6$ )
- or  $C$ , the center of vertices of  $\partial\mathcal{T}_c$  (position smoothing of  $S$  and elimination of  $S_6$ ).

The local topology  $\mathcal{T}_c$  around an edge  $[S, S']$  is composed by the elements of  $\mathcal{T}$ , whose all vertices belong to  $\mathcal{S} = \{S, S'\} \cup \mathcal{N}(S, S')$ , where  $\mathcal{N}(S, S') = \mathcal{N}(S) \cap \mathcal{N}(S')$  (as on figure 2.7(a)). In this case,

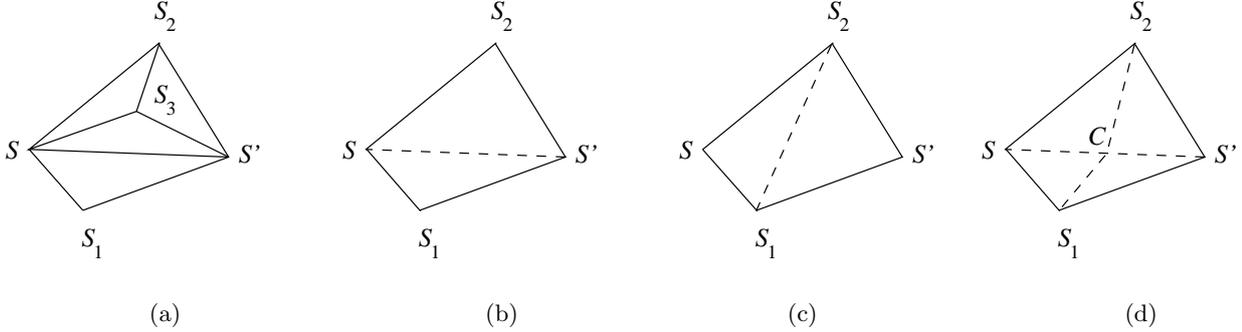


Figure 2.7: The local topology  $\mathcal{T}_c$  around an edge  $[S, S']$  (a) and three candidates for  $\mathcal{T}_e$  (b-d).

candidates for  $\mathcal{T}_e$  are the topologies  $\mathcal{T}^*(S'', \partial\mathcal{T}_c)$ , where  $S''$  is:

- $S$  or  $S'$  themselves (elimination of  $S_3$  on figure 2.7)
- one of the vertices of  $\partial\mathcal{T}_c$  (edge swapping)
- or  $C$ , the middle of the edge  $[S, S']$  (introduction of a new node).

### 2.2.2.2 Convergence

Empirically, algorithm 2.1 always reaches the minimal volume  $|\Omega|$  (which is computed by (2.3) page 26). According to theorem 2.2,  $(\mathcal{N}, \mathcal{T})$  is then a valid mesh of  $\Omega$  (as on figure 2.5 bottom right).

Unfortunately, there is no convergence theorem at that time. However, the cut and paste operation is legal (for all dimension  $d$ ), since we have the following result (already mentioned in [Coupez 2000] and completed here).

**Proposition 2.3** with the notations of algorithm 2.1, we denote by  $\mathcal{T}_f$  the elements of  $\mathcal{T}$  whose all vertices except one, belong to  $\mathcal{S}$  (figure 2.8). If:

- (i)  $\partial\mathcal{T}_c$  is a mesh topology,
- (ii) all faces that compose  $\partial\mathcal{T}_c$  are also faces of  $\mathcal{T}_f$ s whose all vertices belong to  $\mathcal{S}$ ,

then the new topology  $\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$  is also a mesh topology of  $\Omega$ .

In other words, after the cut and past operation, we still have a mesh topology (except when condition (ii) is not satisfied).

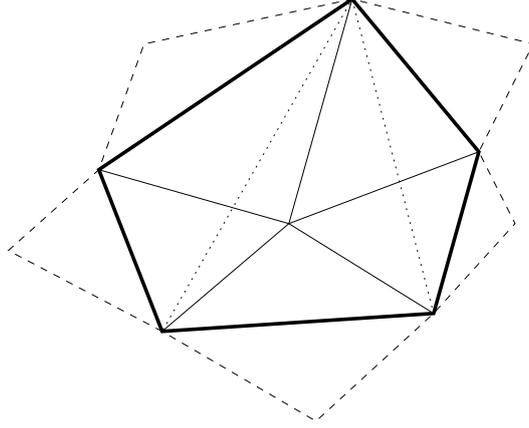


Figure 2.8:  $\mathcal{T}_c$  is in normal lines,  $\partial\mathcal{T}_c$  in bold lines,  $\mathcal{T}_e$  in dotted lines and  $\mathcal{T}_f$  in dashed lines

**Proof :** let us suppose that there exists  $F$ , a face of  $\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$  that shares strictly more than two elements. Firstly, since  $\partial\mathcal{T}_c$  is a mesh topology (condition **(i)**),  $\mathcal{T} \setminus \mathcal{T}_c$  and  $\mathcal{T}_e$  are also mesh topologies. So,  $F$  would necessarily be a face of  $\mathcal{T} \setminus \mathcal{T}_c$  and a face of  $\mathcal{T}_e$ .

Secondly, let  $T$  be an element of  $\mathcal{T} \setminus \mathcal{T}_c$ , with  $F$  as a face. Since  $T \notin \mathcal{T}_c$ , all vertices of  $T$  could not belong to  $\{S\} \cup \mathcal{N}(S)$ . Since  $F$  is a face of  $\mathcal{T}_e$ , all vertices of  $F$  would belong to  $\{S\} \cup \mathcal{N}(S)$ . Since  $F$  is a face of  $T$ , it would come  $T \in \mathcal{T}_f$ . So, condition **(ii)** would say that  $F$  belongs to  $\partial\mathcal{T}_c$ .

Thirdly, since  $F \in \partial\mathcal{T}_c$ , we would have  $F \in \partial\mathcal{T}$  or  $F \in \partial(\mathcal{T} \setminus \mathcal{T}_c)$ , but in both cases,  $F$  would not belong to more than one simplex of  $\mathcal{T} \setminus \mathcal{T}_c$ . Furthermore, lemma 2.2 says that  $\partial\mathcal{T}_e = \partial\mathcal{T}_c$ , so,  $F$  would not share more than one simplex of  $\mathcal{T}_e$ , which raises a contradiction. Thus, no face of  $\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$  can share more than two elements.

Fourthly, we still have to prove that  $\partial(\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e) = \partial\mathcal{T}$ . In what concerns the first inclusion  $\partial\mathcal{T} \subset \partial(\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e)$ , we consider a face  $F \in \partial\mathcal{T}$ . There exists a unique element  $T \in \mathcal{T}$ , such that  $F \in \partial\{T\}$ . Then, we distinguish two cases:

- if  $T \notin \mathcal{T}_c$ , then  $F \notin \partial\mathcal{T}_c$ , so,  $\forall T' \in \mathcal{T}_e$   $F \notin \partial\{T'\}$ ; thus,  $T$  is unique in  $\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$
- if  $T \in \mathcal{T}_c$ , then  $F \in \partial\mathcal{T}_c$  (since  $T$  is unique in  $\mathcal{T}_c$ ), so, there exists a unique element  $T' \in \mathcal{T}_e$ , such that  $F \in \partial\{T'\}$ ; since  $\forall T'' \in \mathcal{T} \setminus \mathcal{T}_c$   $F \notin \partial\{T''\}$ ,  $T'$  is unique in  $\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$ .

In what concerns the reverse inclusion  $\partial(\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e) \subset \partial\mathcal{T}$ , we consider a face  $F \in \partial(\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e)$ . There exists a unique element  $T \in \mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$ , such that  $F \in \partial\{T\}$ . Again, we distinguish two cases:

- if  $T \notin \mathcal{T}_e$ , then  $T \in \mathcal{T} \setminus \mathcal{T}_c$  and  $F \notin \partial\mathcal{T}_e$ ; we would have  $\forall T' \in \mathcal{T}_c$   $F \notin \partial\{T'\}$ , otherwise,  $T'$  would be unique in  $\mathcal{T}_c$  (because of the presence of  $T$  in  $\mathcal{T}$ ); thus,  $F \in \partial\mathcal{T}_c$ , which is impossible because  $\partial\mathcal{T}_e = \partial\mathcal{T}_c$  (lemma 2.2)
- if  $T \in \mathcal{T}_e$  then  $F \in \partial\mathcal{T}_e = \partial\mathcal{T}_c$  (lemma 2.2); so, there exists a unique element  $T' \in \mathcal{T}_c$ , such that  $F \in \partial\{T'\}$ ; by construction, we have  $\mathcal{T}_e \cap (\mathcal{T} \setminus \mathcal{T}_c) = \emptyset$ ; since  $T$  is unique in  $\mathcal{T} \setminus \mathcal{T}_c \cup \mathcal{T}_e$ ,  $\forall T'' \in \mathcal{T} \setminus \mathcal{T}_c$   $F \notin \partial\{T''\}$ ; thus,  $T'$  is unique in  $\mathcal{T}$ .  $\square$

In practice, the assumption **(ii)** is not always satisfied, but, surprisingly enough, the topological mesh is always able to correct those ill-posed situations.

### 2.2.2.3 Boundary treatment

With the star operator:

- a node  $S$  can be eliminated or smoothed, but only if  $S$  is not a vertex of  $\partial\mathcal{T}_c$
- an edge  $[S, S']$  can be swapped and a node  $C$  can be introduced, but only if  $[S, S']$  is not an edge of  $\partial\mathcal{T}_c$ .

Unfortunately, when  $S$  is a boundary node,  $S$  is always a vertex of  $\partial\mathcal{T}_c$ , though we would like to be able to eliminate or smooth it. This is also the case for a boundary edge  $[S, S']$ .

A solution to make it possible for a boundary node  $S$  not to be a vertex of  $\partial\mathcal{T}_c$ , is to connect all boundary faces  $F \in \partial\mathcal{T}$  to a fictitious node (the node 0 on figure 2.9). We obtain new elements, that do

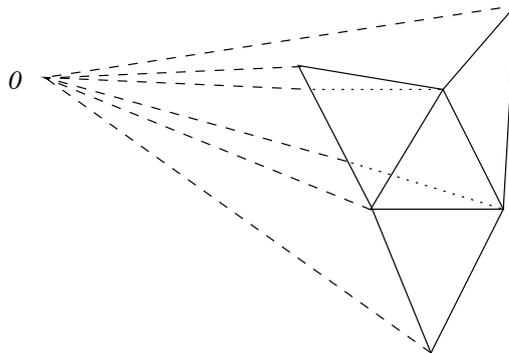


Figure 2.9: A mesh topology without boundary, obtained by connecting old boundary faces to the node 0, so as to form virtual elements (in dashed lines)

not directly participate to the mesh. They are said to be virtual elements. In this way, the topology is without boundary ( $\partial\mathcal{T} = \emptyset$ ). The real boundary is still there, it is simply defined by the set of faces that are connected to node 0 (via the virtual elements).

Virtual elements do not participate to the computation of the volume  $\sum_{T \in \mathcal{T}} |T|$  and, during the algorithm 2.1, node 0 has locally the same coordinates than node  $S$ . Thus, boundary nodes can be eliminated, smoothed or introduced and boundary edges can be swapped, if it does not induce volume changing.

## 2.3 Conclusion

The mesh generation by topological optimization does not try to generate directly a mesh, instead, it generates a simple mesh topology that is progressively improved, until it becomes a mesh. This method operates iteratively, contrary to the direct methods (tree method, advancing front and Delaunay), whose benefits and drawbacks are summarized in table 2.1 (limited to the context of material forming).

mesh generator	benefits	drawbacks
tree method	- a very useful tree structure - good mesh inside	- many nodes introduced - poor boundary mesh - poor anisotropy possibilities
advancing front (unconstrained)	- best geometrical quality of the generated elements	- convergence problems (noisy front)
Delaunay method	- very efficient	- difficulties to recover the boundary - poor quality elements can be generated (slivers)
topological optimization	- very robust - dimension-independent (4D) - simple implementation	- quite slow

Table 2.1: Comparison between mesh generation techniques

The topological mesh generator is not that easy to understand, but quite simple to implement. Indeed, only topological operations are performed by the algorithm 2.1 and this kind of operations are less subject to numerical roundoff errors than geometrical operations (which are intensively used by a Delaunay mesh generator, for instance). Furthermore, node 0 introduction allows the treatment of boundary nodes and boundary edges as if they were inside the domain. There is no specific treatment, which simplifies the implementation.

Besides, operations in algorithm 2.1 are local and dimension-independent, thus, the topological mesh generator has been parallelized [Coupez *et al.* 2000] and has succeeded in 4D mesh generation (see appendix A.3.3 page 161). 4D meshes shall be useful for space-time solvers, dedicated to equations formulated on  $\Omega \times [0, T]$ . This 4D mesh generation is (as far as we know) a first attempt in unstructured mesh generation.

However, the topological mesh generator is an heuristical approach, that succeeds in all tested cases (up to now). This robustness is mainly due to the optimization strategy chosen by [Coupez 1991].

Eventually, the algorithm of the topological mesh generator costs more than a Delaunay mesh generator, because much more topologies are tested. But the idea to treat a mesh by a local optimization, is easily extended to the anisotropic adaption in the next chapter.

## Chapter 3

# Mesh adaption driven by a metric field

Once the first mesh of the domain  $\Omega$  has been built, by one of the preceding methods, the next question is: how to improve this mesh, so that it becomes suitable for a solver?

In fact, the first mesh is often not fine enough (as on figure 2.5 bottom right). We need to refine it, which implies the introduction of several element layers through the thickness, so as to well capture phenomena. Unfortunately, isotropic meshes with several element layers through the thickness lead to dozens of  $10^6$  nodes, which implies too many degrees of freedom for an implicit solver on a single workstation.

In our case, the solver can deal with an anisotropic mesh, where element stretching is operated in the low gradient directions. Chapter 4 describes a possible method to generate anisotropic meshes with several element layer through the thickness. Techniques without metrics are possible, but the method of interest relies on the computation of a metric field.

The aim of this chapter is to point out the existing methods in mesh adaption to a metric field. Those methods can proceed by optimization (section 3.2) or by a complete mesh regeneration (section 3.1).

We remark that the metric field can be:

- constant on the domain  $\Omega$ , so that we stay in an Euclidian framework
- heterogeneous over  $\Omega$ , so that we leave the Euclidian framework (without being in a Riemannian framework, due to the lack of regularity)
- isotropic (in other words, proportional to the identity matrix:  $\frac{I}{h^2}$ , where  $h$  is the mesh size field)
- or truly anisotropic (orthotropic, in fact).

### 3.1 State of the art: an hybrid Delaunay / advancing front method

Basically, a Delaunay method do not introduce new nodes during the mesh generation process (except Steiner points that are necessary in 3D, for the boundary recovering). To enrich a Delaunay mesh, a coupling with one of the following techniques is needed:

- introduction of new nodes at the center of too big elements
- cutting of too big edges
- introduction of new nodes by the advancing front method.

Experience of [George and Borouchaki 2002] shows that the best technique is the introduction of new nodes by the advancing front method (next section).

#### 3.1.1 Constrained advancing front

In section 2.1.2 page 20 the front is advancing in the void. This is the unconstrained advancing front. The major drawback of this technique is the non convergence problem, in most complex 3D situations.

Fortunately, a front can also advance in an already meshed domain, which is called the constrained advancing front (figure 3.1). Elements of the initial mesh are considered as bad ones and elements gen-

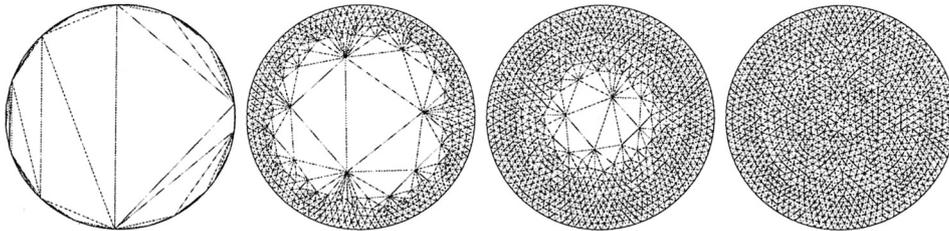


Figure 3.1: Constrained advancing front in 2D (from [Ern and Guermond 2002])

erated by the constrained advancing front as good ones. The front is then the set of faces that separate good elements from bad ones.

The steps of the constrained advancing front method are almost the same:

- initially the front is the boundary mesh
- then, a good quality element is built from a front face (with the possible introduction of a new node)
- this new node is inserted by the Delaunay kernel (which implies that there is no need to deal with any front collision)
- we get a new front and the method is repeated while the front is not empty
- the resulting mesh can be optimized.

The method do no more suffer from any convergence problem, since at each step a valid mesh of  $\Omega$  is obtained.

### 3.1.2 Algorithm modifications

Deep modifications [Borouchaki and George 2000] are needed to make this hybrid Delaunay / advancing front method compliant with a metric field:

- in the advancing front algorithm, the best point chosen to construct a good quality element from a front face is no more unique: in each vertex  $S$  of that face, there is a metric  $M(S)$  with which we can compute a point  $P(M(S))$ ; the provisory candidate is the center of all these candidates
- but, if the edges obtained with this provisory candidate do not all have a size equal to 1, then new candidates  $P(M(S))$  are considered, which are the points at length 1 on these edges; this selection process is repeated until convergence (which may take several iterations)
- then, the retained candidate  $S$ , is inserted by the anisotropic Delaunay kernel: this time, the cavity around  $S$  is defined by the set of all elements (with vertices  $S_0, \dots, S_d$ ), such that all there outer elliptic balls (respectively evaluated in metrics  $M(S), M(S_0), \dots, M(S_d)$ ) contain  $S$
- eventually, in what concerns the optimization step, the quality criterion of an element with vertices  $S_0, \dots, S_d$  is considered to be the weaker quality criterion, among those evaluated in the metrics  $M(S_0), \dots, M(S_d)$ .

The adaption strategy adopted by [George and Borouchaki 2002] is then the following:

- boundary extraction of the previous mesh
- anisotropic remeshing of this boundary [Castro-Díaz and Hecht 1995]
- building of a new mesh for the domain (with recovering of the remeshed boundary), according to the anisotropy defined by the metric field  $M$  (which is still interpolated on the old mesh).

Since the old mesh is used as a background mesh dedicated to the interpolation of  $M$ , lots of localizations are involved, so as to evaluate the metric field on new points. Besides, the linear matrix interpolations are not performed directly on the metric field  $M$  but rather on  $M^{-1/2}$ , which is more accurate (see appendix A.1.3.1 page 148), but also more expensive.

The obtained results (figure 3.2) are very impressive, since a good conformity seems to be achieved

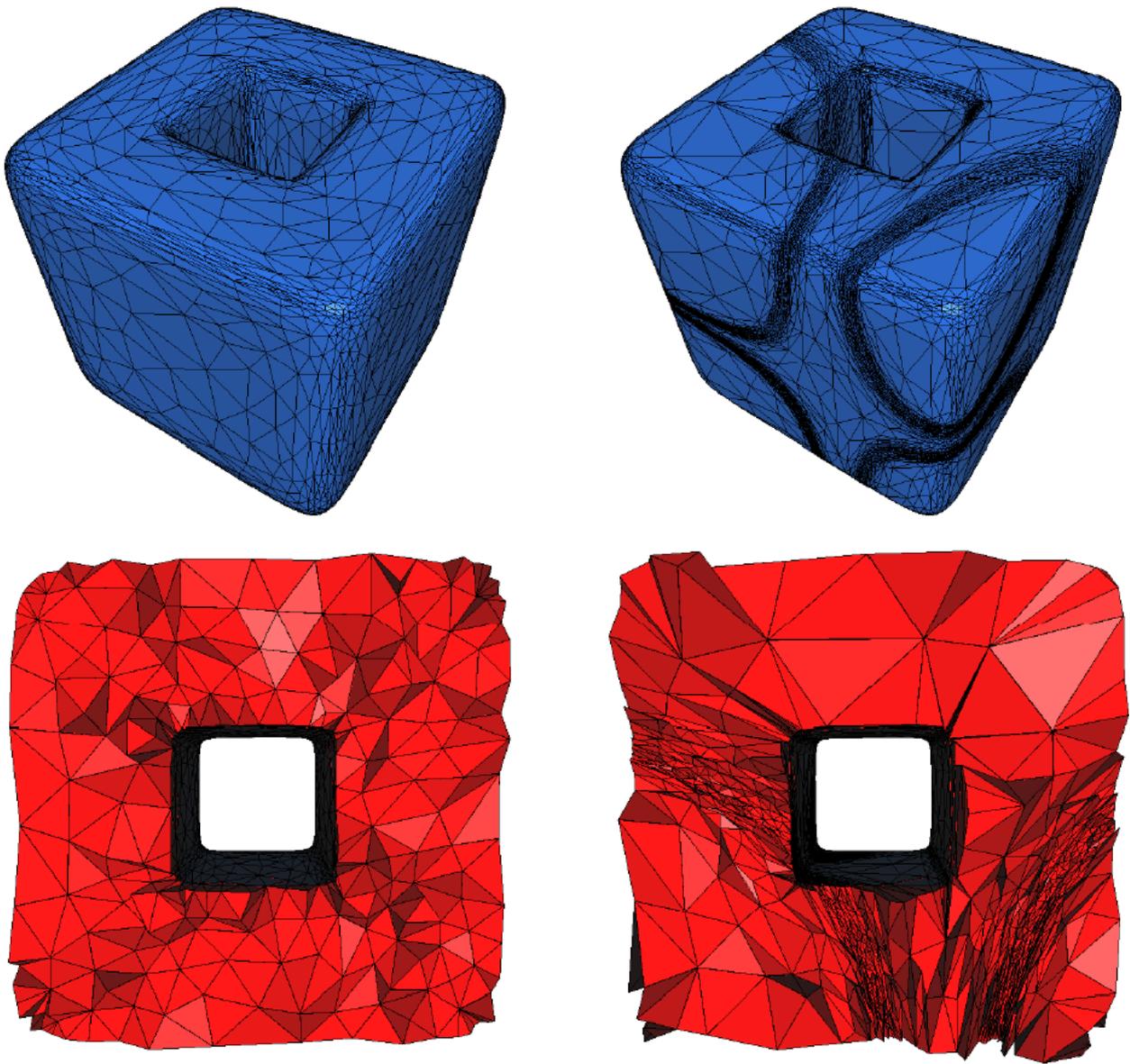


Figure 3.2: Anisotropic mesh adaptation by an hybrid Delaunay / advancing front method, the initial mesh on the left and the final mesh (after 8 iterations) on the right, from [Alauzet 2003] (bottom figures are sectional views)

and the element anisotropy is real. However, the algorithm changes and the method coupling are very technical and time consuming, in what concerns the development phase.

## 3.2 Topological optimization

Anisotropic adaption by optimization is much more simple. Several teams have been identified in this research field, such as [Tam *et al.* 2000, Pain *et al.* 2001, Remacle *et al.* 2002, Bottasso 2004] in 3D and [Dompierre *et al.* 2002] in 2D. But, the first approach can be granted to [Coupez 2000], whose technique is exposed hereafter.

### 3.2.1 Anisotropic optimization strategy

The optimization strategy, presented here, uses:

- a relationship order, instead of a cost function
- the star operator and no other operator
- a simple interpolation between metrics, which surprisingly reveals to be the most accurate (in our context).

#### 3.2.1.1 Anisotropic shape criterion and size criterion

Thanks to the topological mesh generation (or other technique, in fact), the minimal volume has been reached. Now, we have to conserve this minimal volume and to improve the element quality, so that these elements become conform to the metric field  $M$ . So as to evaluate the element quality in the metric field  $M$ , we need to define an anisotropic shape criterion.

**Definition 3.1** the *anisotropic shape criterion* used to measure the quality of an element  $T$ , with vertices  $S_0, \dots, S_d$ , is

$$c_0 \frac{|T|_{M(T)}}{h_{M(T)}^d} \quad (3.1)$$

where

$$\text{the averaged metric is} \quad M(T) = \frac{1}{d+1} \sum_{i=0}^d M(S_i) \quad (3.2)$$

$$\text{the Euclidian volume is} \quad |T|_{M(T)} = |T| \sqrt{\det(M(T))} \quad (3.3)$$

$$\text{the } L^2 \text{ averaged size is} \quad h_{M(T)} = \left( \frac{2}{d(d+1)} \sum_{0 \leq i < j \leq d} \|S_j - S_i\|_{M(T)}^2 \right)^{1/2} \quad (3.4)$$

$$\text{and the normalization factor is} \quad c_0 = \frac{d!}{\sqrt{d+1}} 2^{d/2} \quad (3.5)$$

the classical notation,  $\|x\|_M^2 = x^\top M x$ , being used.

Some remarks about this shape criterion:

- we work with a metric  $M(T)$  that is constant on the element  $T$
- volumes and lengths are thus evaluated in a Euclidian framework (but not canonical, except when  $M$  is the identity matrix)
- $h(T)$  is not the diameter of  $T$ ; however, considering the averaged size and not the maximal length, makes it possible to distinguish between two elements with the same diameter but with different edge sizes (besides the fact that they have different volumes)

- $c_0$  is such that (3.1) is equal to 1 when  $T$  is equilateral in the metric  $M(T)$  (see section A.2.3 page 159)
- it has been proved by [Dompierre *et al.* 2003] that this anisotropic criterion shows great properties.

Furthermore, in order to introduce or to remove enough nodes before the quality optimization, we use the size criterion

$$\min \left( h_{M(T)}^d, \frac{1}{h_{M(T)}^d} \right) \quad (3.6)$$

Finally, we use the quality criterion  $c(T)$ , defined as the minimum between the shape criterion (3.1) and the size criterion (3.6)

$$c(T) = \min \left( c_0 \frac{|T|_{M(T)}}{h_{M(T)}^d}, h_{M(T)}^d, \frac{1}{h_{M(T)}^d} \right) \quad (3.7)$$

### 3.2.1.2 Order relationship between local topologies

This quality criterion enables the introduction of the same relationship order, than in section 2.2.1.3 page 29. This relationship order is dedicated to the comparison between candidate topologies  $\mathcal{T}_e$  for replacing the old topology  $\mathcal{T}_c$ , in the cut-and-past process of algorithm 2.1 page 29.

**Definition 3.2** let  $\mathcal{T}_1 = (T_1^i)_{1 \leq i \leq I_1}$  and  $\mathcal{T}_2 = (T_2^i)_{1 \leq i \leq I_2}$  be two candidate topologies whose elements are ordered such that

$$\forall 1 \leq i_1 < i_2 \leq I_j \quad c(T_{j}^{i_1}) \leq c(T_{j}^{i_2}) \quad (3.8)$$

(in other words, topologies  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are ordered by ascending quality criterion, so as to compare worst elements at first). We consider the *lexicographical order*, denoted by  $<$  and defined by

$$\mathcal{T}_1 < \mathcal{T}_2 \quad \Leftrightarrow \quad \exists 1 \leq i_0 \leq \min(I_1, I_2), \text{ such that } \begin{cases} \forall i < i_0 & c(T_1^i) = c(T_2^i) \\ \text{and} & c(T_1^{i_0}) < c(T_2^{i_0}) \end{cases} \quad (3.9)$$

This relationship order ensures that the worst element of the best topology is as good as possible. Conversely, techniques based on cost functions, like [Bottasso 2004], can keep some degenerated or poor quality elements.

### 3.2.1.3 Algorithm modifications

The overall strategy of algorithm 2.1 page 29 is still used. However, the aim has changed since we now need to preserve the minimal volume (so as to work with valid meshes) and to maximize the topology  $\mathcal{T}$  (according to the previous relationship order). Again, we proceed by local optimizations.

**Algorithm 3.1** so as to adopt a mesh  $(\mathcal{N}, \mathcal{T})$  to a metric field  $M$ :

**while** topology  $\mathcal{T}$  has been modified **do**

**for** each node and each edge of the mesh **do**

- remove the local topology  $\mathcal{T}_c$  around this node or this edge
- among all candidates  $\mathcal{T}_e$  whose volume is minimal (these candidates are obtained by the star operator), insert a maximal topology (according to the relationship order between topologies)
- transport the metric field  $M$ , if necessary

**done**

**done**

Again, only the star operator (2.2) (defined page 25) is required. In fact, combinations of this operator are able to reproduce other operators used in [Tam *et al.* 2000, Pain *et al.* 2001, Remacle *et al.* 2002, Bottasso 2004].

Since the boundary nodes are treated in the same way than the interior nodes (section 2.2.2.3 page 33), the adaption of the boundary mesh is a faithful trace of the interior mesh adaption (unlike an hybrid Delaunay / advancing front method, where a separate adaption is performed for the boundary).

Unfortunately, we do not get a perfect mesh after the algorithm 3.1 is performed for the first time. If we iterate, by recomputing the metric field on the new mesh and by executing the algorithm 3.1 again, then a conforming mesh to the metric field  $M$  is built. In other words, the loop denoted by II on figure 2 page 5 is run several times before the final adaption is achieved.

#### 3.2.1.4 Metric update

Each time a new node  $C$  is introduced, the metric field needs to be updated. When  $C$  is the center of  $S_1, \dots, S_n$ , vertices of  $\partial\mathcal{T}_c$ , we simply consider the averaged value

$$M(C) = \frac{1}{n} \sum_{i=1}^n M(S_i) \quad (3.10)$$

and when  $C$  is the middle of an edge  $[S, S']$ , we take

$$M(C) = \frac{M(S) + M(S')}{2} \quad (3.11)$$

This interpolation choice may be mathematically arguable (appendix A.1.3.1 page 148), but it leads to an efficient update of the field  $M$ . Surprisingly, the imprecision made by this interpolation cannot be improved by better interpolations (see next section). In fact, the iterative process does not need the better interpolation, but the most favorable interpolation for the convergence.

Besides, due to this update step, the topological mesh adaption could not work with a  $P^0$  metric field (in other words, a metric field defined on the elements), although it could be advantageous (in mesh coarsening [Janka 2002], for example). A  $P^0$  field update during the cut and paste process would be very difficult and expensive, unlike the update of a node defined information, which is easier to implement.

### 3.2.2 Validation

In this section, several analytical metric fields are given and we follow the non conformity coefficient  $\epsilon_{\mathcal{T}}$  (defined in section 1.3.3 page 18) evolution during the iterative process of adaption by topological optimization. The more  $\epsilon_{\mathcal{T}}$  is close to 0, the better the conformity between the mesh and the metric field used in the adaption.

We use the  $P^1$  interpolation of the metric field  $M$  described previously, but we also test other interpolations:

- a  $P^1$  interpolation of the transformation field  $M^{1/2}$ , by considering

$$M^{1/2}(T) = \frac{1}{d+1} \sum_{i=0}^d M^{1/2}(S_i) \quad (3.12)$$

$$|T|_{M(T)} = |T| \det(M^{1/2}(T)) \quad (3.13)$$

$$h_{M(T)} = \left( \frac{2}{d(d+1)} \sum_{0 \leq i < j \leq d} \|M^{1/2}(T)(S_j - S_i)\|^2 \right)^{1/2} \quad (3.14)$$

$$M^{1/2}(C) = \frac{1}{n} \sum_{i=1}^n M^{1/2}(S_i) \quad \text{or} \quad \frac{M^{1/2}(S) + M^{1/2}(S')}{2} \quad (3.15)$$

- a  $P^1$  interpolation of the inverse transformation field  $M^{-1/2}$ , by considering

$$M^{1/2}(T) = \left( \frac{1}{d+1} \sum_{i=0}^d M^{-1/2}(S_i) \right)^{-1} \quad (3.16)$$

$$|T|_{M(T)} = |T| \det(M^{1/2}(T)) \quad (3.17)$$

$$h_{M(T)} = \left( \frac{2}{d(d+1)} \sum_{0 \leq i < j \leq d} \|M^{1/2}(T)(S_j - S_i)\|^2 \right)^{1/2} \quad (3.18)$$

$$M^{-1/2}(C) = \frac{1}{n} \sum_{i=1}^n M^{-1/2}(S_i) \quad \text{or} \quad \frac{M^{-1/2}(S) + M^{-1/2}(S')}{2} \quad (3.19)$$

- a  $P^1$  interpolation of the minimal mesh size  $h = \max_{\lambda, \text{eigenvalue of } M} \lambda^{-1/2}$ , by considering

$$h(T) = \frac{1}{d+1} \sum_{i=0}^d h(S_i) \quad (3.20)$$

$$|T|_{M(T)} = |T| \quad (3.21)$$

$$h_{M(T)} = \frac{1}{h(T)} \left( \frac{2}{d(d+1)} \sum_{0 \leq i < j \leq d} \|S_j - S_i\|^2 \right)^{1/2} \quad (3.22)$$

$$h(C) = \frac{1}{n} \sum_{i=1}^n h(S_i) \quad \text{or} \quad \frac{h(S) + h(S')}{2} \quad (3.23)$$

### 3.2.2.1 Anisotropic case in 2D

We start with the 2D anisotropic metric field defined on the simple rectangular domain  $[0, 7] \times [0, 9]$ , by [George and Borouchaki 1997]. First iterations of the adaptation process, with metric field  $M$ , are given on figure 3.3. CPU times are less than 1 seconde. Those CPU times will be more significant in 3D.

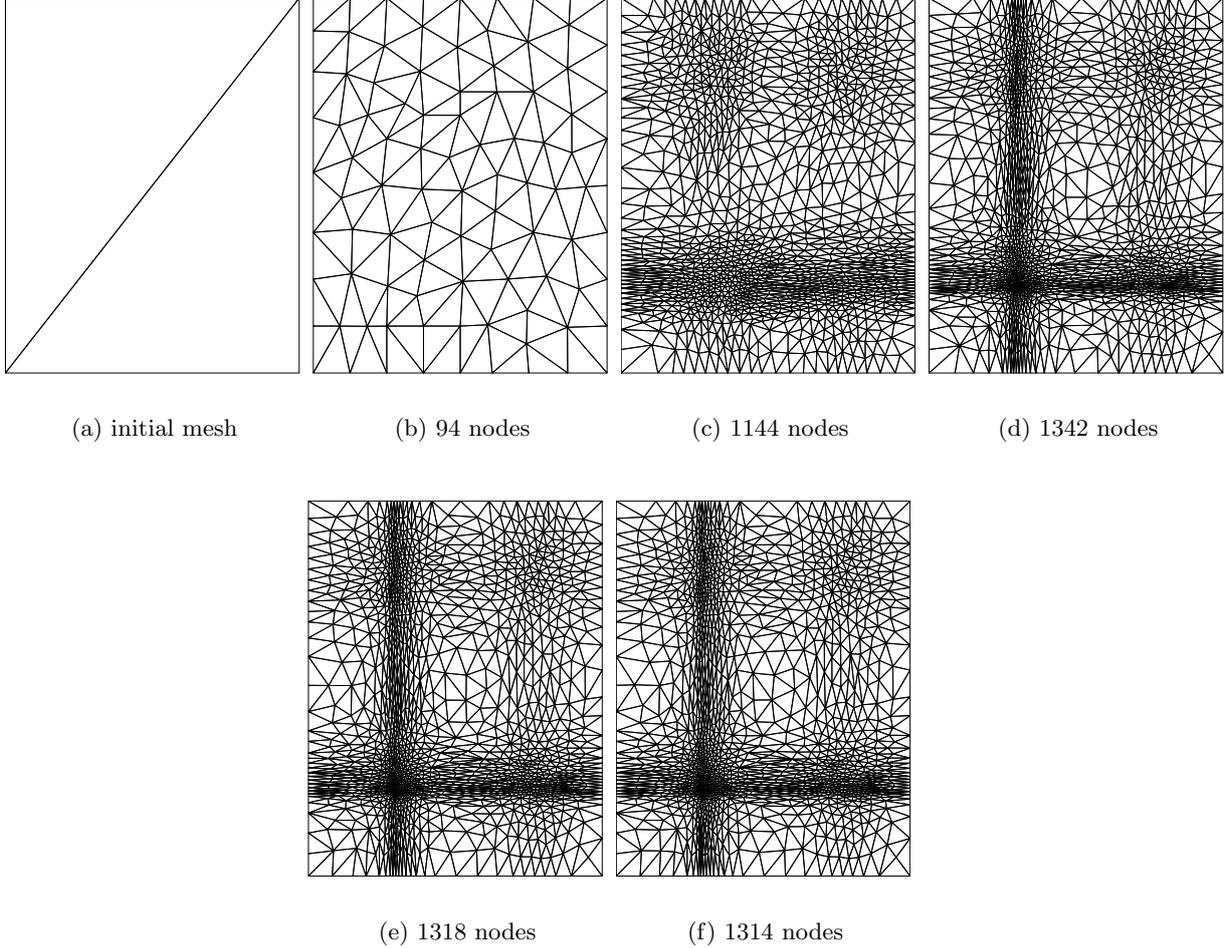


Figure 3.3: Iterative adaptation process on a 2D anisotropic test case

Evolutions of the non conformity coefficient, for all configurations ( $M$ ,  $M^{1/2}$ ,  $M^{-1/2}$  and  $h$ ), are plotted on figure 3.4. Obviously, with this anisotropic case, the use of an isotropic mesh size  $h$  do not lead a good metric compliant mesh. Indeed, the curve obtain with  $h$  tends to a high asymptotical value of  $\epsilon_{\mathcal{T}}$ . Conversely, the three other curves (for  $M$ ,  $M^{1/2}$  et  $M^{-1/2}$ ) converge well to 0, in less than 8 iterations. It implies that the adaptation has succeeded.

More precisely (figure 3.5), the three curves tends to quite similar asymptotical values. These asymptotical values are intrinsic thresholds of the adaptation method. The lowest one is obtained with  $M^{1/2}$ , but a faster convergence is obtained with  $M$ .

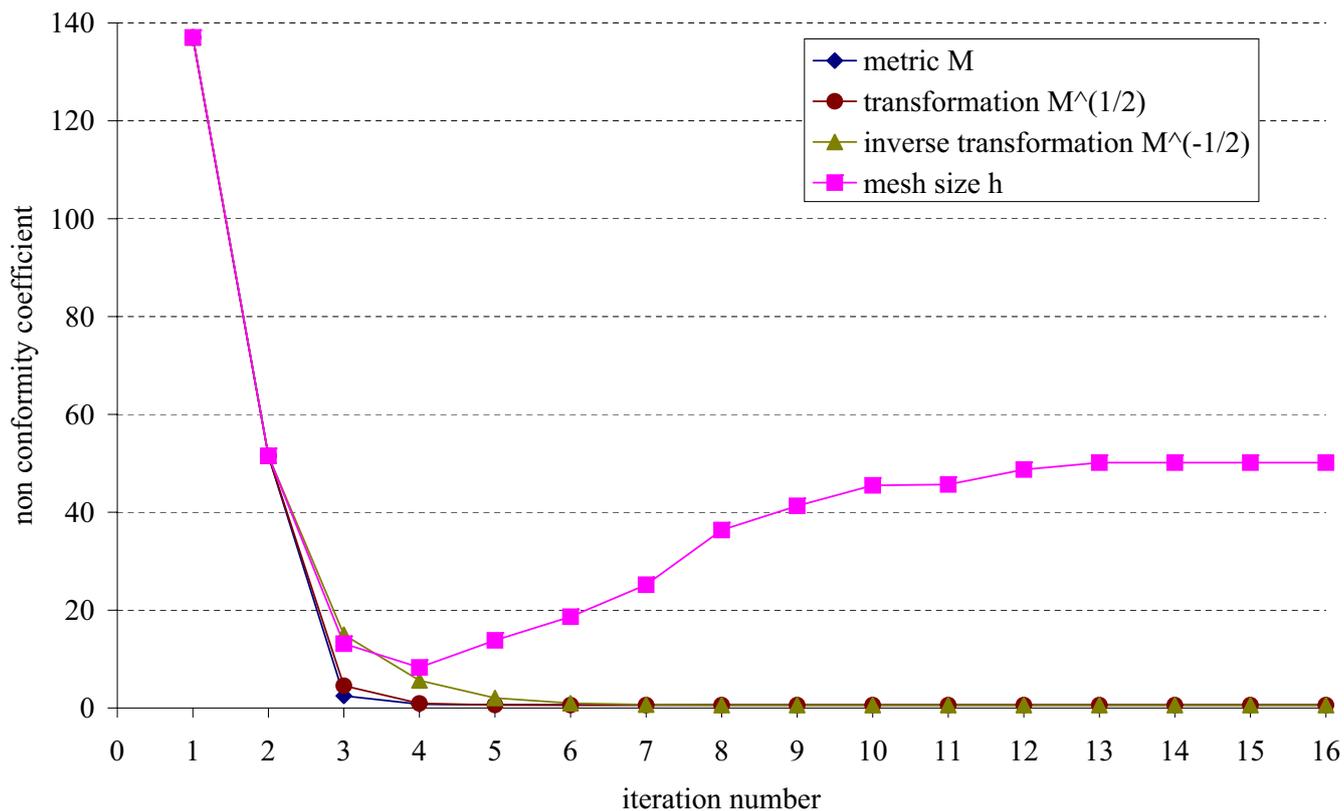


Figure 3.4: Convergence history for the 2D anisotropic test case

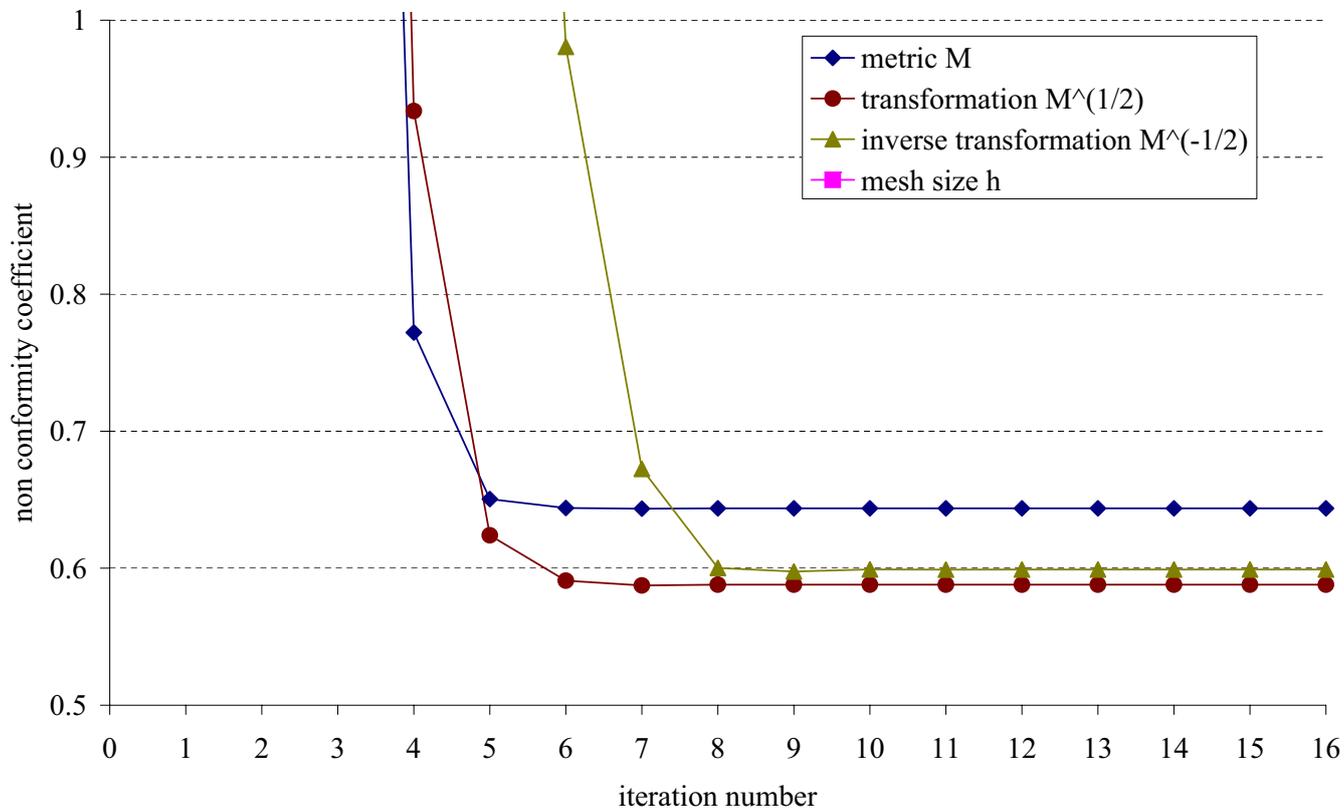


Figure 3.5: Enlargement of the convergence curves of figure 3.4

### 3.2.2.2 Isotropic case in 2D

Before the 3D case, we propose to observe the behavior of the anisotropic adaption algorithm with an isotropic metric field (in 2D). This is the same test case than previously, except that only the smallest mesh size is considered [George and Borouchaki 1997]. First iterations of the adaption process, with metric field  $M$ , are given on figure 3.6. Surprisingly, the obtained meshes do not show a good symmetry,

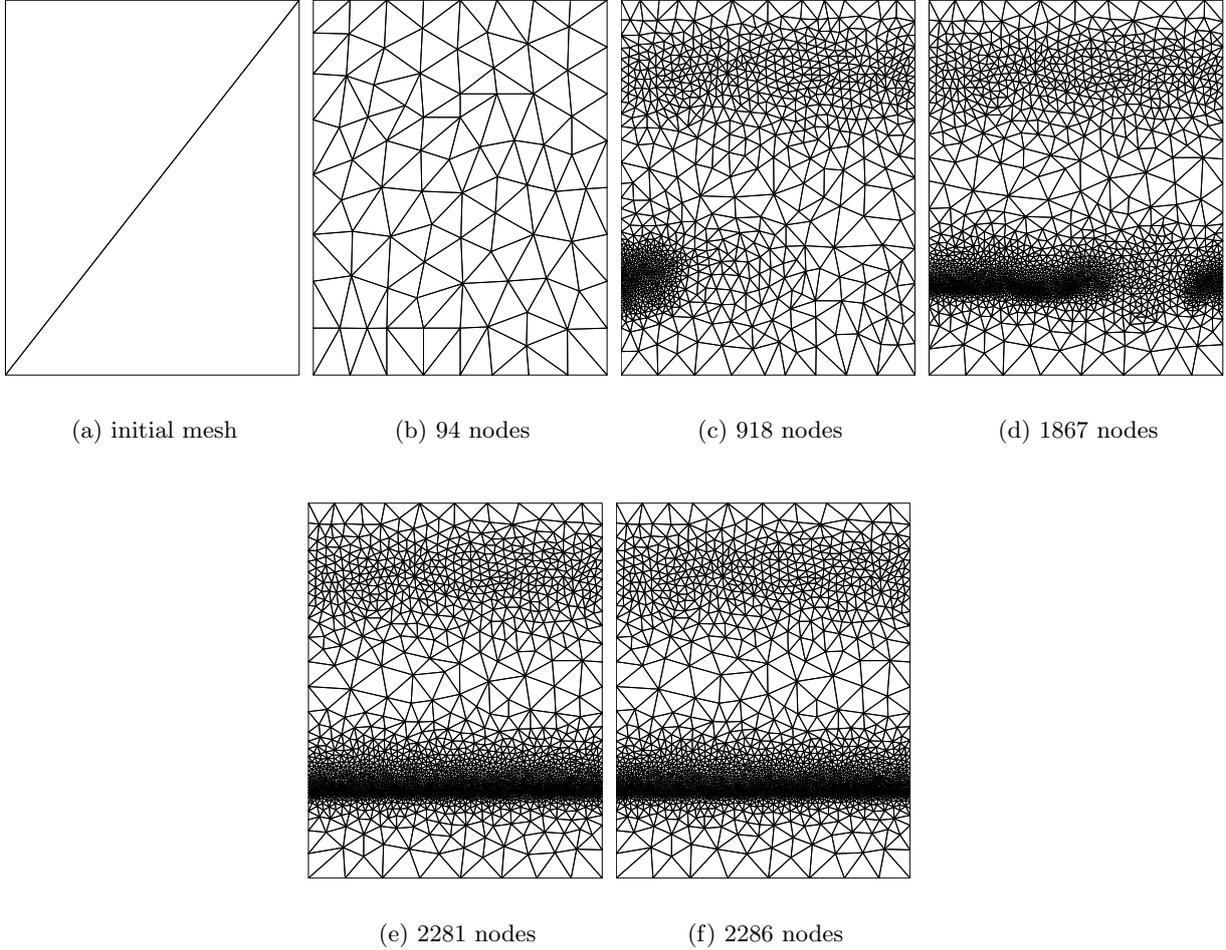


Figure 3.6: Iterative adaption process on a 2D isotropic test case

despite the symmetry of the problem.

This time, all convergence curves, including the mesh size field  $h$ , show a good adaption (figure 3.7). Curves concerning  $M^{-1/2}$  and  $h$  are perfectly superimposed, which verifies the fact that formula (3.16 - 3.23) are equivalent, in the isotropic framework.

However, convergence with  $M^{-1/2}$  and  $h$  is less stable than convergence with  $M$  or  $M^{1/2}$  and the asymptotical value is 10 times higher (figure 3.8). This implies that the treatment of isotropic cases has become better, since the anisotropic strategy is implemented (which is not usual). Again, between  $M$  and  $M^{1/2}$ , the asymptotical value is lower with  $M^{1/2}$  but convergence is faster with  $M$ .

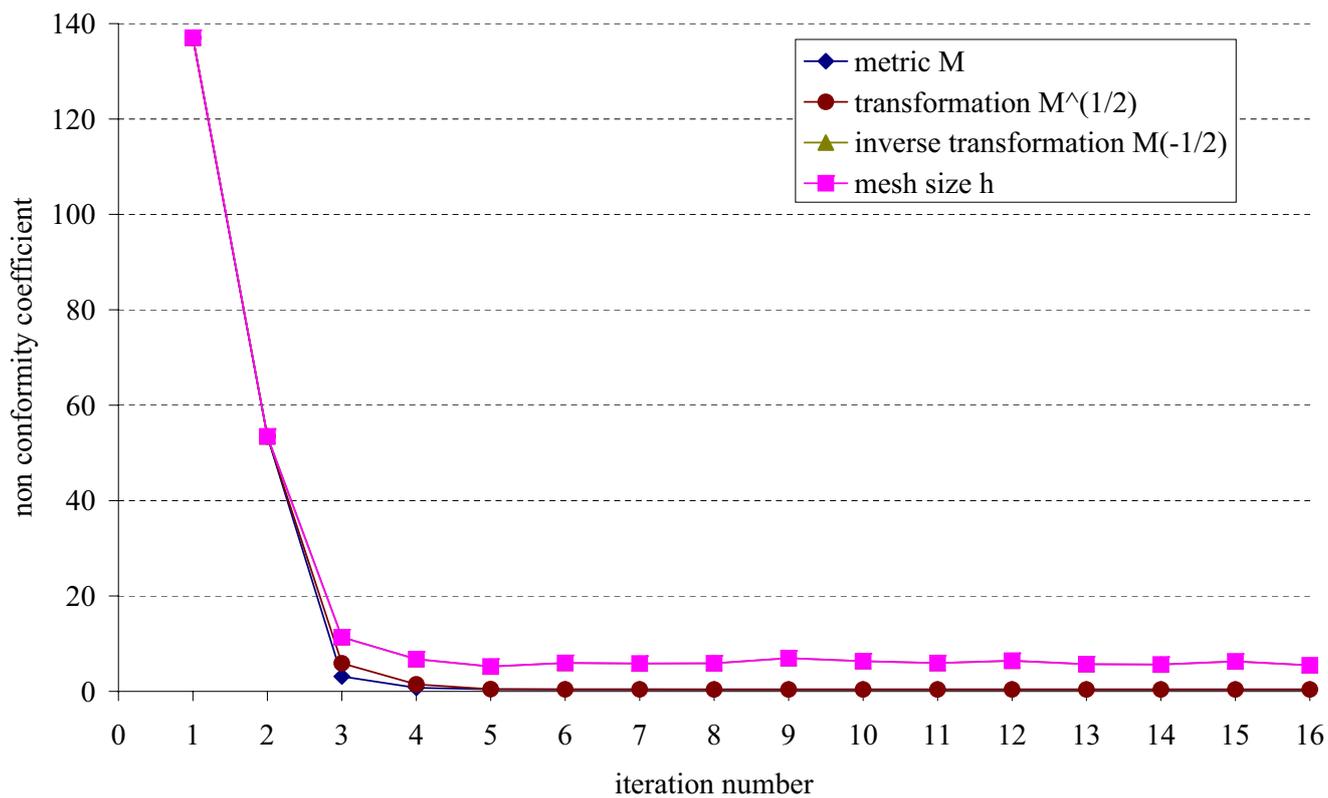


Figure 3.7: Convergence history for the 2D isotropic test case

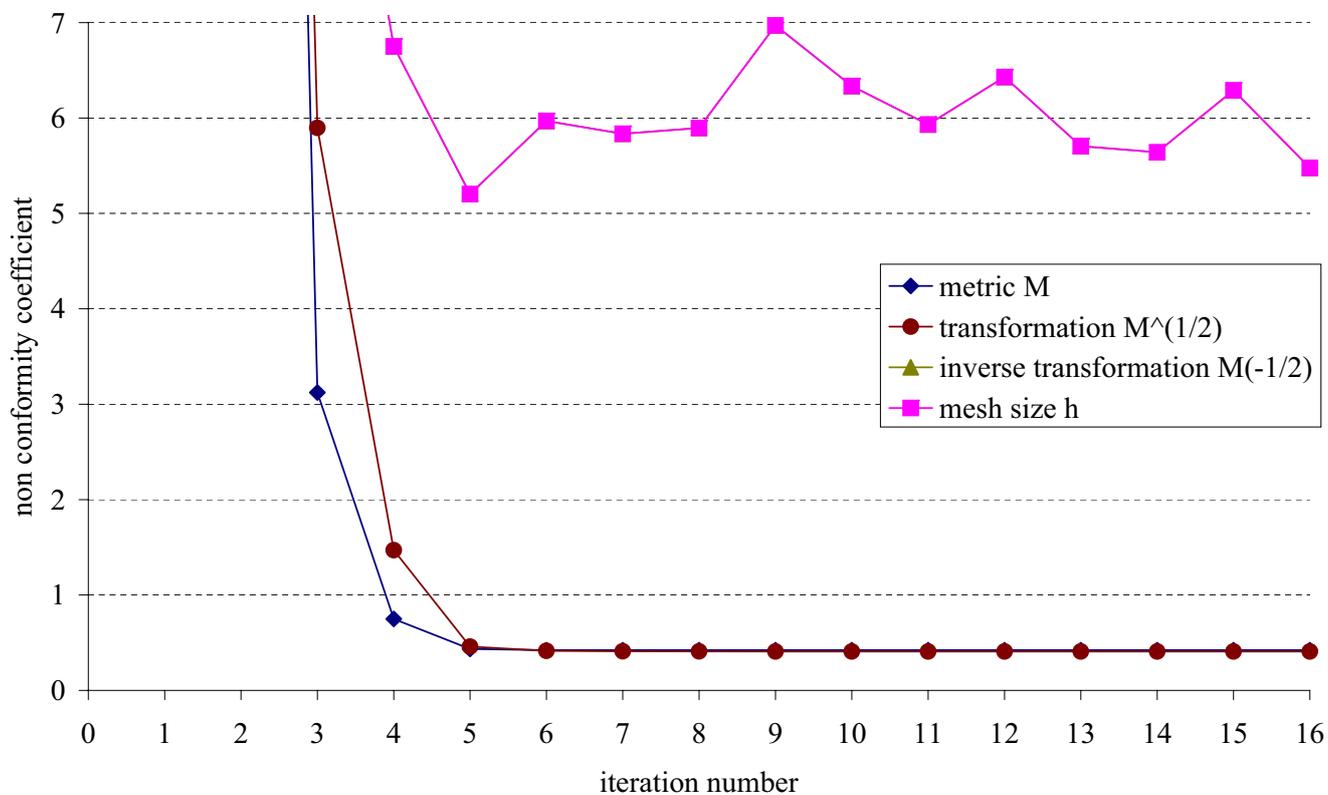


Figure 3.8: Enlargement of the convergence curves of figure 3.7

### 3.2.2.3 Anisotropic case in 3D

2D test cases are interesting, but the most important validation concerns the behavior in 3D. Here, we consider the test case defined by [Bottasso 2003, Bottasso 2004] on the parallelepiped  $[0, 0.1] \times [0, 1] \times [0, 1]$ . First iterations of the adaption process, with metric field  $M$ , are given on figure 3.9. CPU times are obtained on a 220 Mflops workstation. When the mesh is close to the final one, the adaption process is not expensive (between figure 3.9(e) and figure 3.9(f), it takes 1 minute and 42 seconds), but at the beginning, many refining operations are needed (between figure 3.9(b) and figure 3.9(c), it takes 12 minutes).

Again, the mesh size field  $h$  fails (figure 3.10). Furthermore, the curves obtained with  $M^{1/2}$  and  $M^{-1/2}$  show some erratic high values during the convergence process, which expresses the presence of at least one bad element in the mesh.

In this case, the curve obtained with  $M$  is the only one that converges monotonically and with the lowest asymptotical value (figure 3.11). Interpolation with  $M$  seems to be the best one for our optimization process.

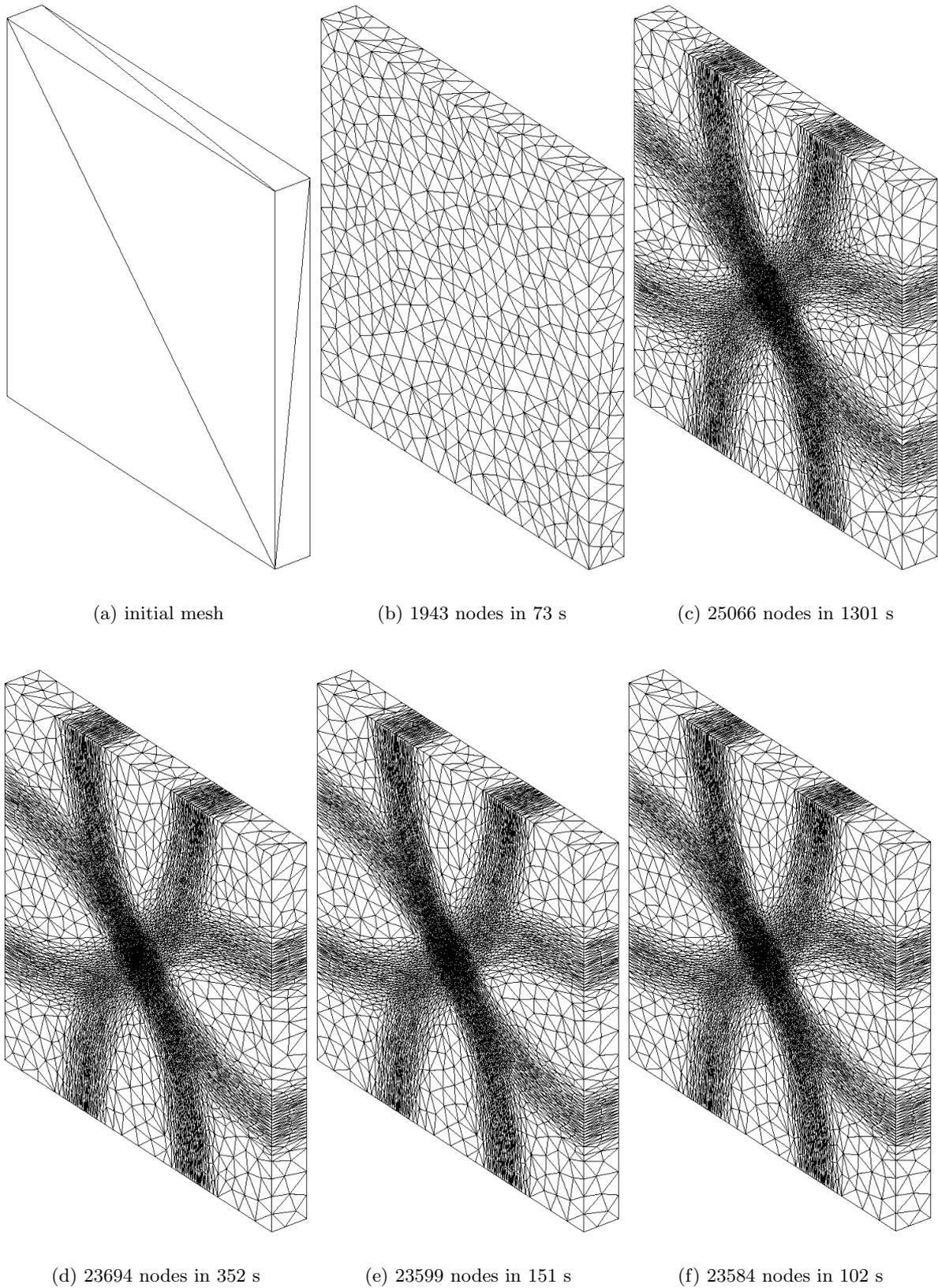


Figure 3.9: Iterative adaption process on a 3D anisotropic test case

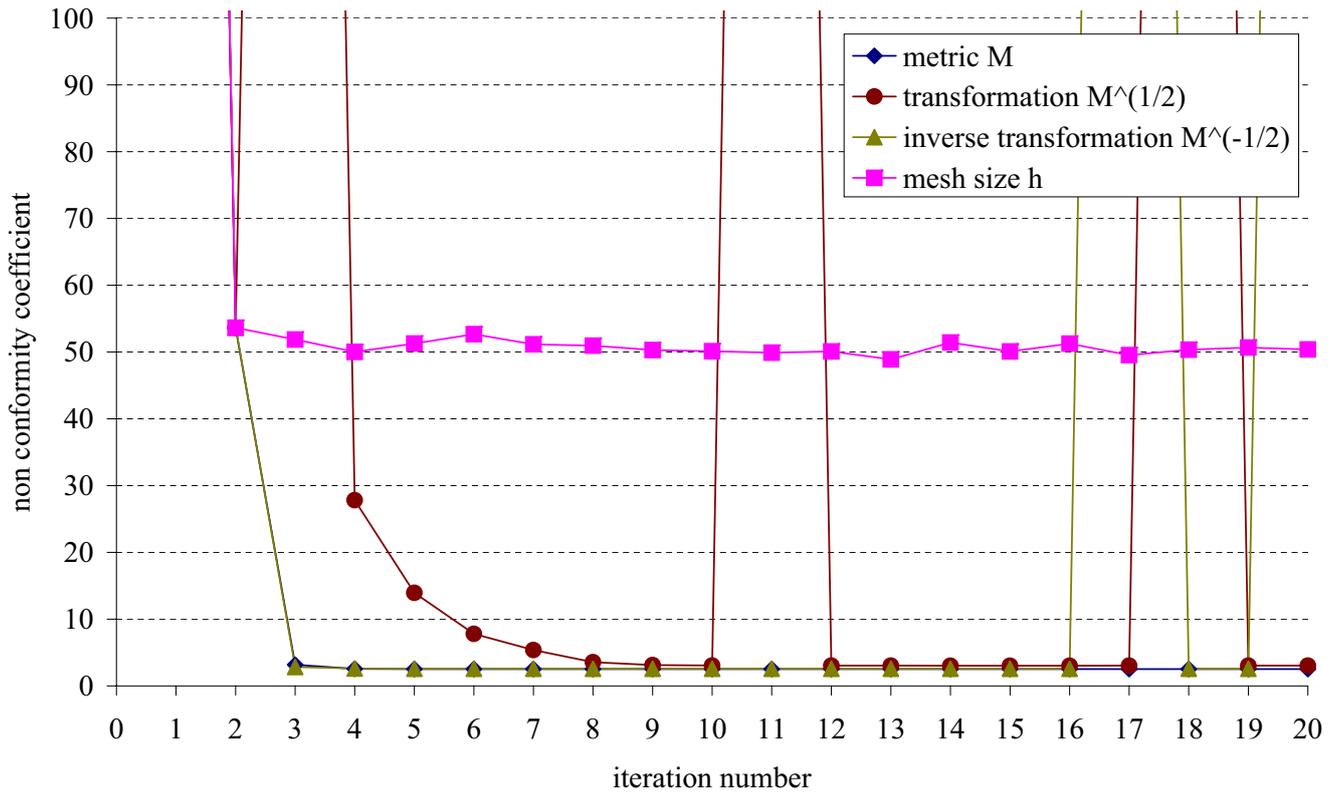


Figure 3.10: Convergence history for the 3D anisotropic test case

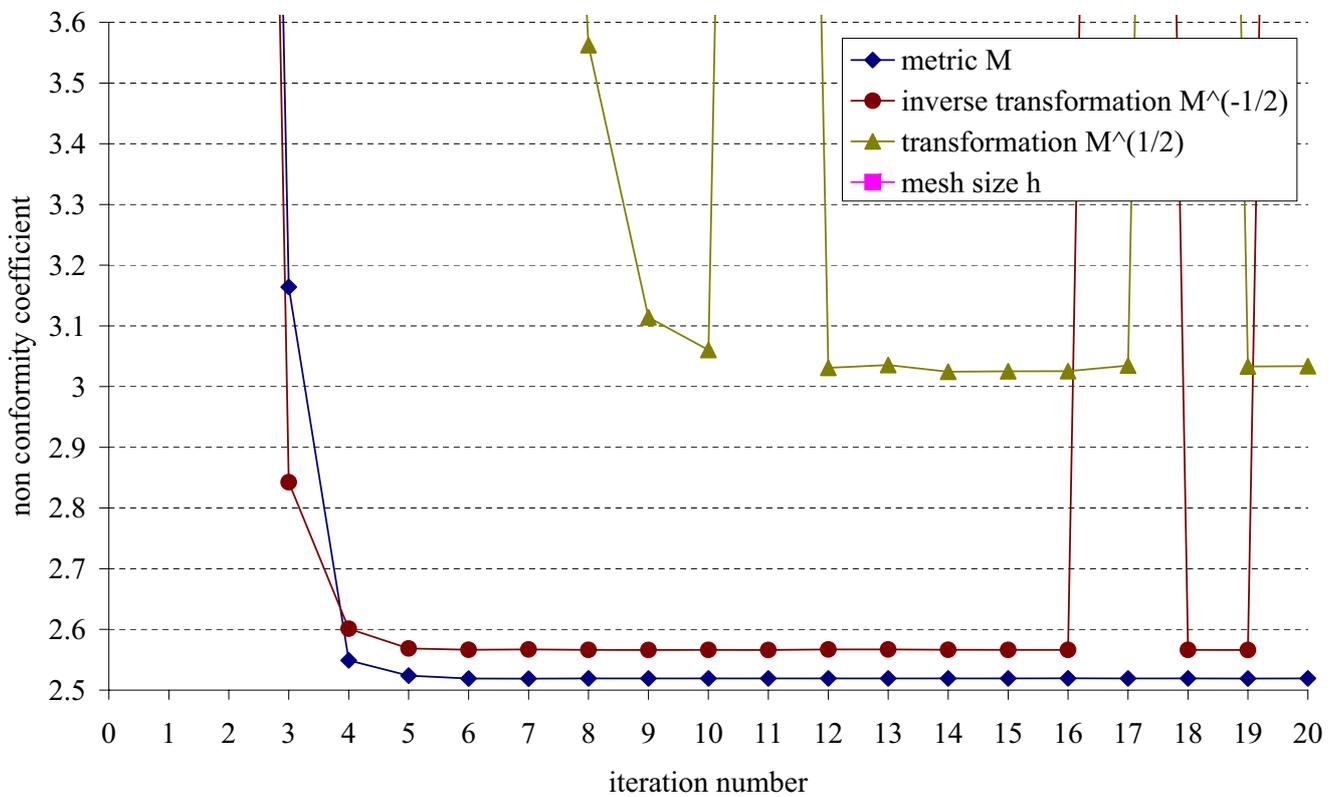


Figure 3.11: Enlargement of the convergent curves of figure 3.10

### 3.3 Conclusion

The topological optimization strategy, introduced in the previous chapter, succeeds (empirically) in the anisotropic adaption and we have quantified the convergence.

No improvement is obtained for this anisotropic adaption process by replacing the metric field  $M$  by the transformation field  $M^{1/2}$  or by the inverse transformation field  $M^{-1/2}$ . It would be more expensive, slower and conformity would not be better. In other words, the optimization strategy is compatible with the direct handling of a metric field  $M$ .

Furthermore, there is no need to replace the  $L^2$  averaged size (3.4) by a  $L^1$  averaged size (the latter being more expensive). Indeed, we see on figure 3.12 that the convergence is better with the  $L^2$  averaged

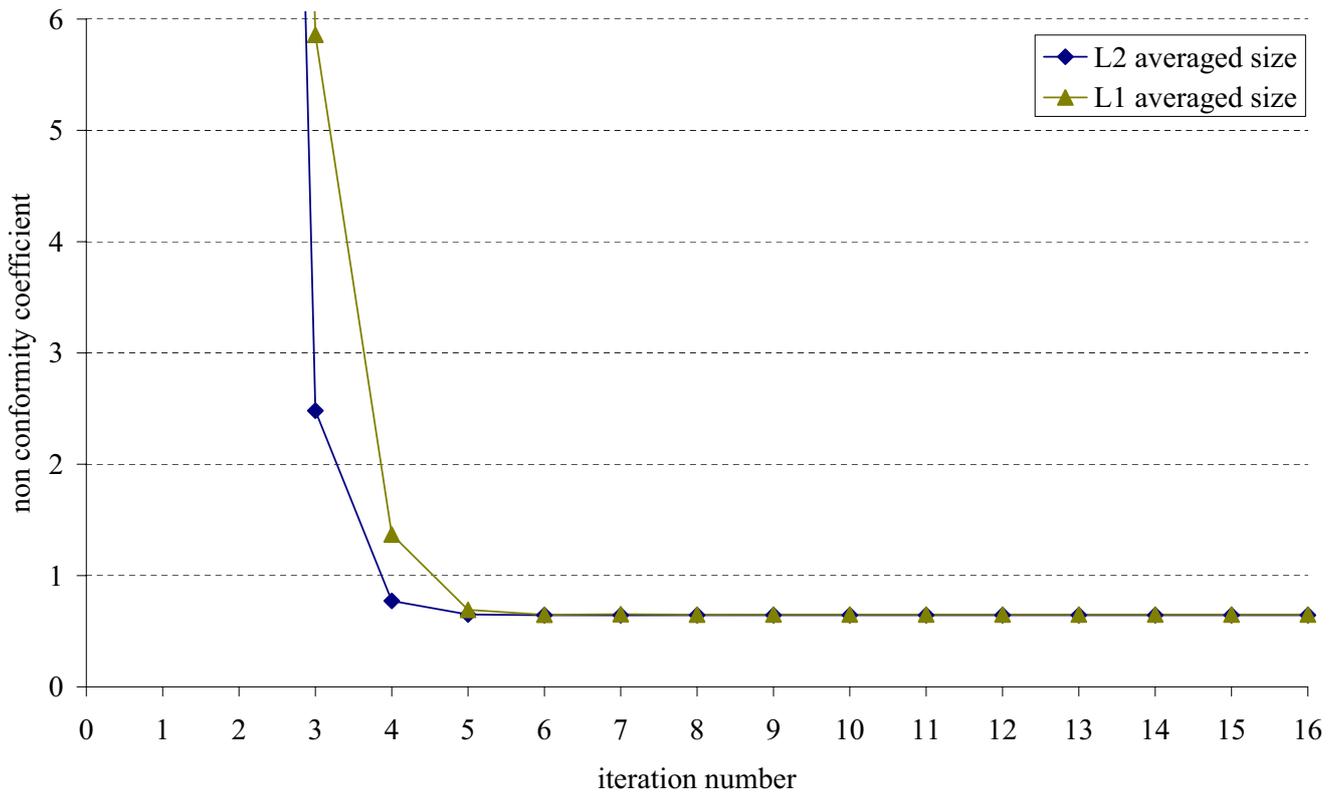


Figure 3.12: Comparison between different averages for the 2D anisotropic test case

size than the  $L^1$  one (the 2D anisotropic test case, where the metric field  $M$  interpolation has been used).

Thanks to this validation benchmark, the tuning of the anisotropic adaption by topological optimization is achieved. Fortunately, less expensive formula are also the best ones.

Compared to other methods, may it be an hybrid Delaunay / advancing front technique or other optimization techniques, the topological optimization keeps the same advantages than in the previous chapter, namely: dimension independence, robustness and simple implementation (efficient formula and simple topological operations).

The building of a good metric field is now required. This is the aim of the following parts.

## Part II

# Natural metric



## Introduction

Metric fields used in anisotropic adaption are usually computed by an a posteriori error estimator [Tam *et al.* 2000, Pain *et al.* 2001, Rémacle *et al.* 2002, Alauzet *et al.* 2003, Dobrzynski *et al.* 2004]. This metric field can be the absolute value of a Hessian matrix (computed from a scalar field), eigenvalues being truncated by a minimal and a maximal mesh sizes.

However, this error estimation is not available when simulation fails due to the initial mesh. Reasons for such a failure may come from: the isotropy of the initial mesh, leading to an excessive memory consumption within an implicit solver or the excessive coarsening of the initial mesh, leading to a solution, so poor, that the Hessian matrix is null.

That is the reason why, in our opinion, efforts must be spent on building a metric field before the solver is run. This part is devoted to the building of an a priori metric field that leads to a suitable mesh for Rem3D solvers. This a priori metric field uses only geometrical information about the computational domain.

For that purpose, we define in chapter 4 (page 55) a so-called natural metric that detects the local anisotropy of a geometry. A computational way for this natural metric is proposed, relying on a semi-definite programming algorithm. The natural metric is then applied to introduce several element layers through the thickness of industrial parts (figure 3.13), which can be thin and curved.

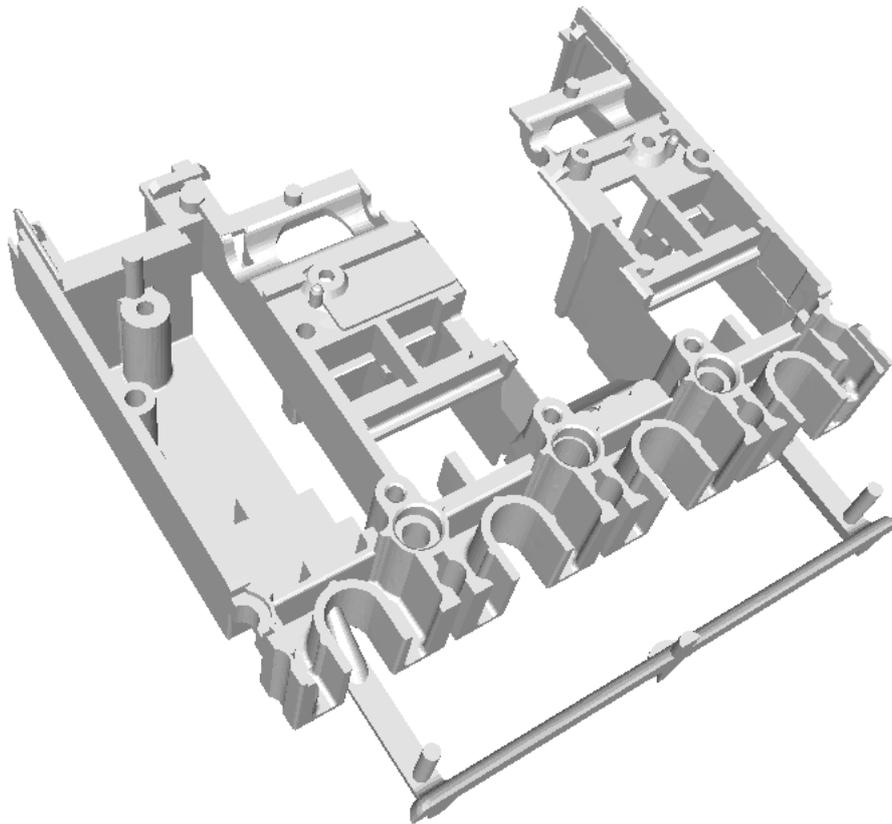


Figure 3.13: Complex geometry that could not be decomposed in simple patterns (courtesy of Schneider Electric)

Unfortunately, the natural metric is not always satisfying. For some configurations, we need the developments of chapter 5 (page 83), where two distinct ways are explored, so as to improve the natural metric with the use of a simple, but relevant, simulation result.



## Chapter 4

# Natural metric generated a priori

The first problem that we address with a metric field computation is the need of a mesh with several element layers everywhere through the thickness. In the Rem3D context, this need is due to the fact that a coarse mesh of the mould cavity is able to interpolate correctly a (linear) newtonian flow, but not a (non linear) viscoplastic flow. Another reason is the strong temperature gradient between the injected thermoplastic (at about 200°C) and the surrounding mould (at about 50°C). The thermal solver needs a fine mesh trough the thickness, otherwise cooling could be overestimated and the thermoplastic could even obstruct.

In order to get a mesh with several element layers through the thickness, several techniques without metric could be employed:

- the mesh could be extruded from its median surface [Knockaert 2001], but the median surface is not easy to build and local thickness size must be given everywhere
- the mesh could be extruded from its boundary [Garimella and Shephard 1998], but curved geometries imply a lot of connecting zone treatments
- an initial coarse mesh could be systematically refined in a direction that is orthogonal to the boundary [Peraire and Morgan 1997] or the thickness direction, which has to be computed, like in [Garimella 1998] (figure 4.1).

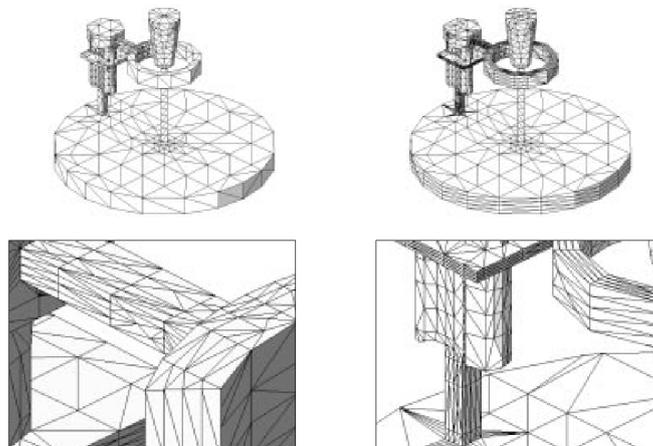


Figure 4.1: Thickness refining (from [Garimella 1998])

All this techniques can be employed with moderately complex geometries. Their main drawback is the impossibility to coarsen the mesh in the directions that are orthogonal to the thickness direction. Mesh sizes in these directions are decided by the initial mesh, which may lead to an expensive triangulation

with curved geometries (figure 4.4).

On the contrary, a metric based method is more flexible, which enables coarsening and complex geometry treatment. The main difficulty is to build the good metric field. When the computational domain can be divided in simple plate, cylindrical and spherical zones (figure 4.2) or divided in zones

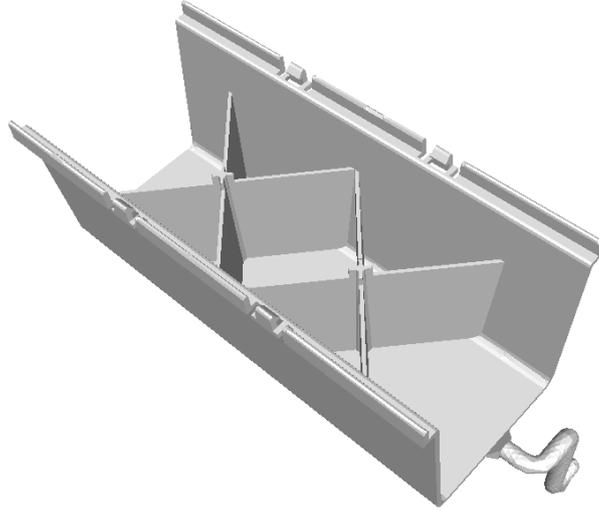
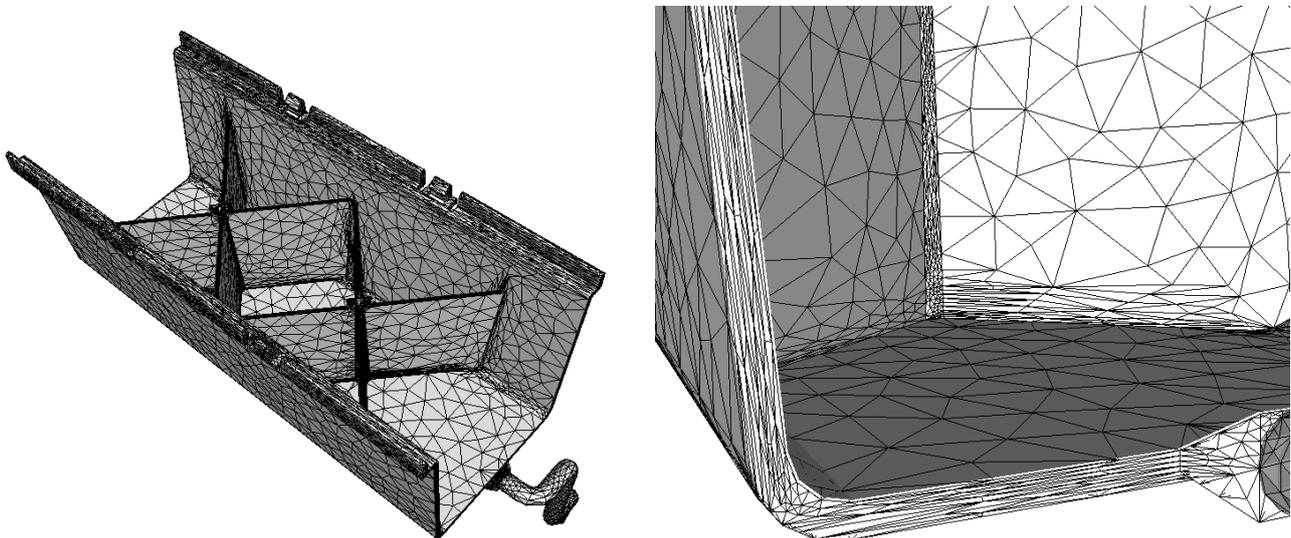


Figure 4.2: A geometry that can be divided in plates and cylinders (courtesy of Plastic Omnium)

where the anisotropy is known [Gobeau 1996], we can easily build an analytical metric field (see section A.1.4 page 151). Unfortunately, this manual division of the computational domain in simple zones is tedious. For instance, so as to obtain the mesh of figure 4.3, three days have been spent on the an-



(a) suitable mesh for Rem3D (48 646 nodes)

(b) enlargement

Figure 4.3: Mesh obtained with an analytical metric field

alytical metric computation. The most difficult step concerns the rotation computation of inclined plates.

Such a manual division becomes inconceivable for industrial complex geometries (figure 3.13). Furthermore, an analytical metric is quasi impossible to exhibit when curvature changes in a complex way (figure 4.4).

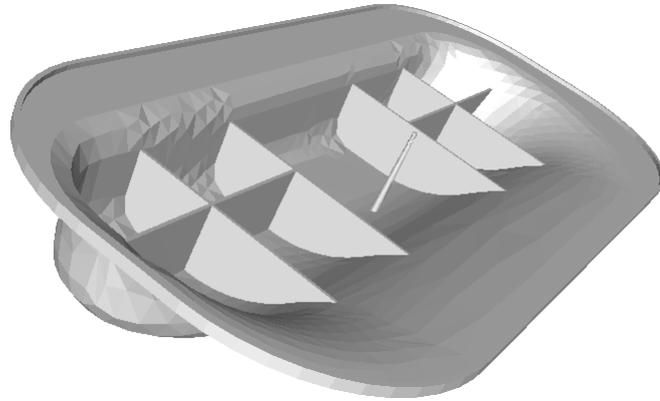


Figure 4.4: Curved industrial geometry (courtesy of Atofina)

For these reasons, we focus our attention on the automatic building of a natural metric that detects the local anisotropy of the computational domain. Three steps are required:

- hull growing according to successive neighborhoods (section 4.1), until the appropriate order is attained (section 4.3)
- elliptic interpolation of that hulls, so as to obtain a metric (section 4.2)
- and refining this metric, in order to introduce several element layers through the thickness (section 4.4).

## 4.1 Local anisotropy

With the introduction of a relationship order between metrics, the local anisotropy of a geometry can be defined. However, a numerical computation of the natural metric cannot be directly derived from such a definition. That is the reason why we propose a practical method to approximate the local anisotropy from a given mesh of the geometry (even a very coarse mesh).

### 4.1.1 Definition

Let us introduce a relationship order between metrics.

**Definition 4.1** let  $M_1$  and  $M_2$  be two metrics, whose eigenvalues  $\lambda_1^j \leq \dots \leq \lambda_d^j$  are sorted (in ascending order). We consider the *lexicographical order* between metrics, denoted by  $<$  and defined by

$$M_1 < M_2 \quad \Leftrightarrow \quad \exists 1 \leq i_0 \leq d \text{ such that } \begin{cases} \forall 1 \leq i < i_0 & \lambda_i^1 = \lambda_i^2 \\ \text{and} & \lambda_{i_0}^1 > \lambda_{i_0}^2 \end{cases} \quad (4.1)$$

The unit ball of a metric  $M$ , centered on  $x \in \mathbb{R}^d$ , is the elliptical conic

$$\mathcal{E}(M, x) = \{y \in \mathbb{R}^d \text{ such that } \|y - x\|_M \leq 1\} \quad (4.2)$$

The first idea, for the natural metric definition on  $x \in \Omega$ , is to consider the greatest unit ball containing  $x$  and entirely included in  $\Omega$ . Unfortunately, this metric would degenerate when  $x$  is located at a corner of the geometry.

The second idea is to consider the greatest metric centered on  $x$  and entirely included in  $\Omega$ . Unfortunately, this metric would again degenerate when  $x$  is close to the boundary. However, this definition is fine when  $x$  belongs to the median surface, which is denoted by  $\partial_{\frac{1}{2}}\Omega$ . That is the reason why we employ the following definition, with the notations of figure 4.5.

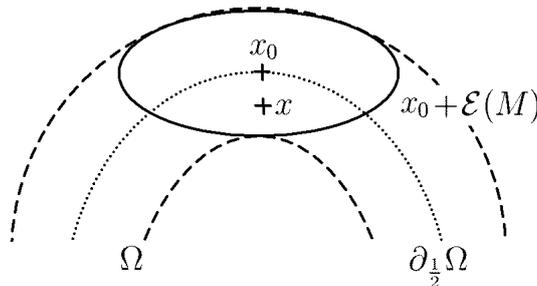


Figure 4.5: Notations for the natural metric definition

**Definition 4.2** the *natural metric* of  $\Omega$ , on  $x \in \Omega$ , is defined by:

- the greatest metric centered on  $x$  and entirely included in  $\Omega$ , when  $x \in \partial_{\frac{1}{2}}\Omega$
- the natural metric of the closest point  $x_0 \in \partial_{\frac{1}{2}}\Omega$  from  $x$ , otherwise
- the averaged value, if  $x_0$  is not unique.

Implementing that definition would involve too many operations. An approximation is then introduced in the following section.

### 4.1.2 Neighborhood growing

Anisotropic adaption, as described in the previous chapter, uses a continuous  $P^1$  metric field. Thus, the natural metric has to be defined on each node  $S$  of  $\Omega$ 's mesh. In practice, the local anisotropy of  $\Omega$  is detected by aggregating mesh elements around each node  $S$ . Aggregation is performed by successive neighborhoods.

**Definition 4.3** the  $k$ -th order *neighborhood*  $\mathcal{N}_k(S)$  of a node  $S$  (with  $k \in \mathbb{N}$ ) is recursively defined by

$$\mathcal{N}_0(S) = \{S\} \quad (4.3)$$

$$\mathcal{N}_k(S) = \mathcal{V}(\mathcal{N}_{k-1}(S)) \setminus \bigcup_{0 \leq l < k} \mathcal{N}_l(S) \cup \boxed{\bigcup_{0 \leq l < k} (\mathcal{N}_l(S) \cap \partial\Omega)} \quad (4.4)$$

where  $\mathcal{V}(\mathcal{S})$  is the set of neighbors of a node set, denoted by  $\mathcal{S}$ .

In this definition, the  $k$ -th order neighborhood of  $S$  contains not only the neighbors of the  $(k-1)$ -th order neighborhood of  $S$  that are not already neighbors of  $S$  (with order  $< k$ ), but also all intersections between previous neighborhoods of  $S$  and the boundary (boxed part in (4.4)). Those intersections contain the most important information about the geometry.

The  $k$ -th order neighborhood of  $S$  separates the topology  $\mathcal{T}$  (including its virtual elements connected to node 0, see page 33) into two subtopologies: the subtopology whose elements have all their vertex in the  $k$  first neighborhoods  $\bigcup_{0 \leq l \leq k} \mathcal{N}_l(S)$  and the subtopology  $\mathcal{T}_k^{\text{ext}}(S)$  that contains all other elements (including the virtual elements connected to node 0). The subtopology  $\mathcal{T}_k^{\text{ext}}(S)$  is very interesting since its topological boundary can be seen as the  $k$ -th order hull around  $S$  (figure 4.6).

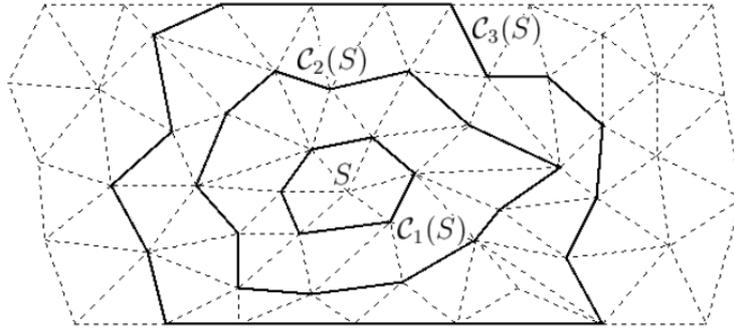


Figure 4.6: Three successive neighborhoods around a node  $S$ .

**Definition 4.4** the (geometrical)  $k$ -th order *hull* around a node  $S$  is the surface defined by

$$\mathcal{C}_k(S) = \bigcup_{F \in \partial\mathcal{T}_k^{\text{ext}}(S)} F \quad (4.5)$$

Some of the  $k$ -th order neighbors of  $S$  may not be vertices of the  $k$ -th order hull. The previous definition eliminates deliberately useless neighbors. This leads to the introduction of the filtered  $k$ -th order neighborhood of  $S$ , defined by

$$\tilde{\mathcal{N}}_k(S) = \mathcal{N}_k(S) \cap \mathcal{C}_k(S) \quad (4.6)$$

that is to say, the  $k$ -th order neighbors of  $S$  that are also vertices of  $\partial\mathcal{T}_k^{\text{ext}}(S)$ .

Then, candidates for the natural metric in  $S$  are the elliptic interpolations of the successive hulls  $(\mathcal{C}_k(S))_{k \in \mathbb{N} \setminus \{0\}}$ . So as to compute such an elliptic interpolation, we use the faces of  $\partial\mathcal{T}_k^{\text{ext}}(S)$  and the nodes of  $\tilde{\mathcal{N}}_k(S)$ .

## 4.2 Discrete elliptic interpolation

For each hull vertex  $S' \in \tilde{\mathcal{N}}_k(S)$ , we denote by  $\partial\mathcal{T}_k^{\text{ext}}(S, S')$  the set of faces from  $\partial\mathcal{T}_k^{\text{ext}}(S)$ , whose  $S'$  is a vertex. This allows the introduction of  $C_k(S)$ , the weighted center of the  $k$ -th order hull around  $S$

$$C_k(S) = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} S' \quad (4.7)$$

$$\text{with } \alpha_{S'} = \frac{\sum_{F \in \partial\mathcal{T}_k^{\text{ext}}(S, S')} |F|}{d \sum_{F \in \partial\mathcal{T}_k^{\text{ext}}(S)} |F|} \quad (4.8)$$

In this section, the following problem is addressed: to find the metric whose unit ball centered on  $C_k(S)$  interpolates at best the hull  $C_k(S)$ . In 2D, this is a classical interpolation problem [Halř and Flusser 1998]. But, as far as we know, 3D solutions (a fortiori dimension-independent solutions) proposed here, are new.

### 4.2.1 Semi-definite programming problem

From a least-square point of view, this interpolation problem can be reformulated as: find the minimum of the following function

$$f(M) = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\|S' - C_k(S)\|_M^2 - 1)^2 \quad (4.9)$$

Hereafter, several properties of that function are investigated.

**Proposition 4.1** the gradient and the Hessian of  $f(M)$  are

$$\nabla f(M) = 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\|S' - C_k(S)\|_M^2 - 1) (S' - C_k(S)) \otimes (S' - C_k(S)) \quad (4.10)$$

$$H(f) = 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (S' - C_k(S)) \otimes (S' - C_k(S)) \otimes (S' - C_k(S)) \otimes (S' - C_k(S)) \quad (4.11)$$

This formulae are used in the minimum  $M$  computation.

**Proof :** since  $\|x\|_M^2 = \sum_{1 \leq i, j \leq d} x_i M_{ij} x_j$ ,  $f(M)$  is quadratic, so,  $f$  has a  $C^\infty$  regularity and

$$\nabla f(M) = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \nabla (\|S' - C_k(S)\|_M^2 - 1)^2 \quad (4.12)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} ((\|S' - C_k(S)\|_M^2 - 1) \nabla (\|S' - C_k(S)\|_M^2 - 1)) \quad (4.13)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\|S' - C_k(S)\|_M^2 - 1) (S' - C_k(S)) \otimes (S' - C_k(S)) \quad (4.14)$$

because  $\nabla \|x\|_M^2 = x \otimes x$ . Indeed, we have

$$\frac{\partial \|x\|_M^2}{\partial M_{i_0 j_0}} = \frac{\partial \left( \sum_{1 \leq i, j \leq d} x_i M_{ij} x_j \right)}{\partial M_{i_0 j_0}} \quad (4.15)$$

$$= x_{i_0} x_{j_0} \quad (4.16)$$

Besides,

$$H(f) = \nabla(\nabla f(M)) \quad (4.17)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \nabla(\|S' - C_k(S)\|_M^2 - 1) (S' - C_k(S)) \otimes (S' - C_k(S)) \quad (4.18)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (S' - C_k(S)) \otimes (S' - C_k(S)) \otimes (S' - C_k(S)) \otimes (S' - C_k(S)) \quad (4.19)$$

which ends the proof.  $\square$

We denote by  $\vec{M} \in \mathbb{R}^{d^2}$  the column vector, whose coefficients are those of  $M$  (row by row). We also use  $\overrightarrow{x \otimes x}$  and we introduce  $\overrightarrow{x \otimes x \otimes x \otimes x} = \overrightarrow{x \otimes x} \otimes \overrightarrow{x \otimes x}$ . Then, the Hessian  $H(f)$  can be seen as the matrix  $\overline{H(f)}$  with  $d^2$  rows and  $d^2$  columns.

**Proposition 4.2**  $f$  is convex and a local extremum of  $f$  is a global minimum of  $f$ .

In other words, it legitimates the minimization of  $f$ .

**Proof :** let us consider  $\vec{h} \in \mathbb{R}^{d^2}$ , we have

$$\overline{H(f)} \vec{h} \cdot \vec{h} \quad (4.20)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \vec{h}^\top (S' - C_k(S)) \otimes (S' - C_k(S)) \otimes (S' - C_k(S)) \otimes (S' - C_k(S)) \vec{h} \quad (4.21)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \left( (S' - C_k(S)) \otimes (S' - C_k(S)) \vec{h} \right)^\top \left( (S' - C_k(S)) \otimes (S' - C_k(S)) \vec{h} \right) \quad (4.22)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \left( (S' - C_k(S)) \otimes (S' - C_k(S)) \vec{h} \right)^2 \quad (4.23)$$

$$\geq 0 \quad (4.24)$$

So,  $f$  is convex. Furthermore, since the set of all real matrices with  $d$  columns and  $d$  rows is also a convex, each local extremum of  $f$  is a global minimum of  $f$ .  $\square$

Since  $f$  is quadratic, all minima  $M$  of  $f$  satisfy the following linear equation system

$$\overline{H(f)} \vec{M} = -\overline{\nabla f(0)} \quad (4.25)$$

However,  $f$  can admits several minima, especially when the population  $\tilde{\mathcal{N}}_k(S)$  is not rich enough. So as to make the minimum unique, the diagonal part of  $\overline{H(f)}$  could be penalized. Unfortunately, such a Tikhonov regularization [Jonas and Louis 2001] selects a particular solution, which may not be the most interesting.

That is the reason why we prefer to enrich the set  $\tilde{\mathcal{N}}_k(S)$ , by introducing the centers of faces  $\partial \mathcal{T}_k^{\text{ext}}(S)$  or even better, by introducing the middle of each edge between two neighbors from  $\tilde{\mathcal{N}}_k(S)$ . Empirically, this leads to a strictly convex function  $f$ .

We still have to study the properties of the minimum admitted by  $f$ . For that purpose, we consider the decomposition

$$\overrightarrow{(S' - C_k(S)) \otimes (S' - C_k(S))} = \begin{pmatrix} U \\ D \\ L \end{pmatrix} \quad (4.26)$$

in upper triangular  $U$ , diagonal  $D$  and lower triangular  $L$  parts. The same decomposition is performed on

$$\vec{M} = \begin{pmatrix} U(M) \\ D(M) \\ L(M) \end{pmatrix} \quad (4.27)$$

$$\text{and } \overrightarrow{\nabla f(0)} = \begin{pmatrix} U(\nabla f(0)) \\ D(\nabla f(0)) \\ L(\nabla f(0)) \end{pmatrix} \quad (4.28)$$

**Proposition 4.3** if  $f$  admits a unique minimum  $M$ , then  $M$  is a symmetrical matrix whose coefficients are solution of the reduced system

$$2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \begin{pmatrix} 2U \otimes U & U \otimes D \\ 2D \otimes U & D \otimes D \end{pmatrix} \begin{pmatrix} U(M) \\ D(M) \end{pmatrix} = - \begin{pmatrix} U(\nabla f(0)) \\ D(\nabla f(0)) \end{pmatrix} \quad (4.29)$$

In other words, there is not  $d^2$  equations to be solved, so as to find  $M$ , but only  $\frac{d(d+1)}{2}$ .

**Proof :** firstly, we remark that

$$\overline{\overline{H(f)}} = 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \begin{pmatrix} U \otimes U & U \otimes D & U \otimes L \\ D \otimes U & D \otimes D & D \otimes L \\ L \otimes U & L \otimes D & L \otimes L \end{pmatrix} \quad (4.30)$$

$$= 2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \begin{pmatrix} U \otimes U & U \otimes D & U \otimes U \\ D \otimes U & D \otimes D & D \otimes U \\ U \otimes U & U \otimes D & U \otimes U \end{pmatrix} \quad (4.31)$$

because  $(S' - C_k(S)) \otimes (S' - C_k(S))$  is symmetrical, so,  $L = U$ .

Secondly, we denote by  $\overline{\overline{P}}$  the transposition matrix, namely, the matrix such that, for all matrix  $M$  with  $d$  rows and  $d$  columns, we have

$$\overline{\overline{P}} \vec{M} = \overrightarrow{M^\top} \quad (4.32)$$

$$\text{in other words, } \overline{\overline{P}} \begin{pmatrix} U \\ D \\ L \end{pmatrix} = \begin{pmatrix} L \\ D \\ U \end{pmatrix} \quad (4.33)$$

Then, we have  $\overline{\overline{H(f)}} \overline{\overline{P}} = \overline{\overline{H(f)}}$ , since the first and third columns of  $\begin{pmatrix} U \otimes U & U \otimes D & U \otimes U \\ D \otimes U & D \otimes D & D \otimes U \\ U \otimes U & U \otimes D & U \otimes U \end{pmatrix}$  are identical. Thus, it comes

$$-\overrightarrow{\nabla f(0)} = \overline{\overline{H(f)}} \vec{M} = \overline{\overline{H(f)}} \overline{\overline{P}} \vec{M} = \overline{\overline{H(f)}} \overrightarrow{M^\top} \quad (4.34)$$

Since  $M$  is unique, we have  $M^\top = M$ , so,  $M$  is symmetrical.

Thirdly, the system (4.25) can be rewritten in the following form

$$2 \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \begin{pmatrix} U \otimes U & U \otimes D & U \otimes U \\ D \otimes U & D \otimes D & D \otimes U \\ U \otimes U & U \otimes D & U \otimes U \end{pmatrix} \begin{pmatrix} U(M) \\ D(M) \\ U(M) \end{pmatrix} = - \begin{pmatrix} U(\nabla f(0)) \\ D(\nabla f(0)) \\ U(\nabla f(0)) \end{pmatrix} \quad (4.35)$$

because  $M$  and  $\nabla f(0)$  are symmetrical. In this new system (4.35), the third row is identical to the first one and can thus be eliminated. Then, by grouping  $U(M)$  coefficients, we get the reduced system (4.29).  $\square$

Thus, the solution  $M$  is always symmetrical. However,  $M$  may not be definite positive and, thus, may not be a metric. It may happen when, for example, the nodes of  $\tilde{\mathcal{N}}_k(S)$  are aligned on a hyperbolae in 2D (in this case  $M$  is the matrix corresponding to this hyperbolae and has a negative eigenvalue).

So as to obtain a metric, the function  $f$  should be minimized under the following constraint:  $M$  is definite positive. Unfortunately, the set of definite positive symmetrical matrices is not a closed set (the series of positive definite matrices  $(\frac{1}{n}I)_{n \in \mathbb{N} \setminus \{0\}}$  tends to the null matrix, which is not definite).

Thus, a minimal eigenvalue  $h_{\max}^{-2}$  should be chosen and minimization should be performed under the constraint: each eigenvalue of  $M$  is lower or equal  $h_{\max}^{-2}$ , so as to make this problem well-posed. The choice of  $h_{\max}$  corresponds to the choice of a maximal radius for the unit ball of  $M$ , which is reasonable.

However, this is a semi-definite programming problem and the constraint is difficult to treat by usual optimization techniques. This constraint could be ignored and solution without constraint could be projected on the constrained space. This is the case, when a metric is built from a Hessian matrix [Tam *et al.* 2000, Leservoisier *et al.* 2001, Pain *et al.* 2001, Remacle *et al.* 2002, Alauzet *et al.* 2003].

In the following sections we propose two original methods, in order to solve the constrained problem. The first method relies on a tensorial decomposition of a size distribution function, around the center  $C_k(S)$  (section 4.2.2). The second method uses a linear combination of metrics associated to the elements generated by the star operator, from center  $C_k(S)$  to the hull  $\partial\mathcal{T}_k^{\text{ext}}(S)$  (section 4.2.3).

### 4.2.2 Tensor decomposition

The first way to find a minimum  $M$  of  $f$  that satisfies the definite positive constraint, is to search  $M^{-1}$  among all linear combinations of tensors  $((S' - C_k(S)) \otimes (S' - C_k(S)))_{S' \in \tilde{\mathcal{N}}_k(S)}$ .

For that purpose, we consider a second order tensorial approximation of the length distribution function, from the center  $C_k(S)$  to the hull  $\mathcal{C}_k(S)$ . We denote by  $S^{d-1} = \{x \in \mathbb{R}^d, \|x\|_2 = 1\}$ , the unit sphere of  $\mathbb{R}^d$ .

**Definition 4.5** the *length distribution function* around the center  $C_k(S)$  is defined by the following function defined on  $S^{d-1}$

$$h(x) = \min_{\lambda \in \mathbb{R}} \lambda^2 \quad (4.36)$$

such that  
 $C_k(S) + \lambda x \in \mathcal{C}_k(S)$

Thus,  $h(x)$  is the squared distance between  $C_k(S)$  and the closest point of  $\mathcal{C}_k(S)$ , in direction  $x$  (figure 4.7).

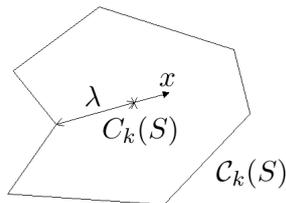


Figure 4.7: Notations for the length distribution function

**Proposition 4.4** the  $h$  function can be approximated by

$$h(x) \simeq \frac{1}{2^{d-1}\pi} \int_{S^{d-1}} h(p) dp + x^\top \left( \frac{d}{d-1} \frac{1}{2^{d-1}\pi} \int_{S^{d-1}} h(p) \left( p \otimes p - \frac{1}{d} I \right) dp \right) x \quad (4.37)$$

In other words,  $h$  can be represented by an averaged value and a traceless second order tensor.

**Proof :** we denote by  $H_{2i}(x)$  the  $2i$ -th order tensor, defined on the unit sphere by

$$H_{2i}(p) = \text{Traceless} (p \otimes \dots (2i \text{ times}) \dots \otimes p) \quad (4.38)$$

where Traceless is the operator that removes the trace part of a  $2i$ -th order tensor. For instance, when  $i = 0$  and  $i = 1$ , we have

$$H_0(p) = 1 \quad (4.39)$$

$$H_2(p) = p \otimes p - \frac{1}{d} I \quad (4.40)$$

However, when  $i = 2$ , it is more complicated [Onat 1982].

From [Onat and Leckie 1988], the series  $(H_{2i})_{i \in \mathbb{N}}$  is an orthogonal series of  $L_2(S^{d-1})$ . By making the assumption that  $h \in L_2(S^{d-1})$ , the Fourier series of  $h$  on the basis  $(H_{2i})_{i \in \mathbb{N}}$  is

$$h(x) = \sum_{i \in \mathbb{N}} \|H_{2i}\|_{L_2(S^{d-1})}^{-1} \left( \int_{S^{d-1}} h(p) H_{2i}(p) dp \right) : H_{2i}(x) \quad (4.41)$$

where the operator  $:$  denotes the complete contraction between two tensors of the same order. Since  $h(-x) = h(x)$ , odd terms are null.

Rewriting the first terms, it gives

$$h(x) = \frac{1}{2^{d-1}\pi} \left( \int_{S^{d-1}} h(p) dp + \frac{d}{d-1} \left( \int_{S^{d-1}} h(p) \left( p \otimes p - \frac{1}{d} I \right) dp \right) : \left( x \otimes x - \frac{1}{d} I \right) \right) + \dots \quad (4.42)$$

because first norms are

$$\|H_0\|_{L_2(S^{d-1})} = \int_{S^{d-1}} dp = 2^{d-1}\pi \quad (4.43)$$

$$\|H_2\|_{L_2(S^{d-1})} = \int_{S^{d-1}} H_2(p) : H_2(p) dp \quad (4.44)$$

$$= \int_{S^{d-1}} \left( 1 - \frac{1}{d} \right) dp = \frac{d-1}{d} 2^{d-1}\pi \quad (4.45)$$

Formula (4.42) can be simplified since

$$\left( \int_{S^{d-1}} h(p) \left( p \otimes p - \frac{1}{d} I \right) dp \right) : I = \text{Trace} \left( \int_{S^{d-1}} h(p) \left( p \otimes p - \frac{1}{d} I \right) dp \right) \quad (4.46)$$

$$= \int_{S^{d-1}} h(p) \text{Trace} \left( p \otimes p - \frac{1}{d} I \right) dp = 0 \quad (4.47)$$

and it comes

$$h(x) = \frac{1}{2^{d-1}\pi} \left( \int_{S^{d-1}} h(p) dp + \frac{d}{d-1} \left( \int_{S^{d-1}} h(p) \left( p \otimes p - \frac{1}{d} I \right) dp \right) : (x \otimes x) \right) + \dots \quad (4.48)$$

For each matrix  $A$  with  $d$  rows and  $d$  columns, we have  $x^\top A x = A : (x \otimes x)$ , so, we obtain (4.37).  $\square$

Following the preceding result, we employ a second order tensor (its trace corresponds to the averaged value), so as to approximate the function  $h$

$$h(x) \simeq x^\top \left( \mu \left( \int_{\tilde{\mathcal{C}}_k(S)} (y - C_k(S)) \otimes (y - C_k(S)) \, dy \right)^{-1} \right) x \quad (4.49)$$

where  $\mu$  is a scalar and  $\tilde{\mathcal{C}}_k(S)$  is the image of  $S^{d-1}$  by  $h$ .

Thus, the following approximation is proposed to interpolate the hull  $\mathcal{C}_k(S)$ , with a metric unit ball

$$M_2 = \mu_k M \quad (4.50)$$

$$\text{with } M = \left( \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (S' - C_k(S)) \otimes (S' - C_k(S)) \right)^{-1} \quad (4.51)$$

$$\text{and } \mu_k = \frac{\sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \|S' - C_k(S)\|_M^2}{\sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \|S' - C_k(S)\|_M^4} \quad (4.52)$$

It results from a  $P^1$  quadrature of the integral (4.49), where  $\alpha_{S'}$  is defined by (4.8). This quadrature could be improved since the function  $y \mapsto (y - C_k(S)) \otimes (y - C_k(S))$  is, in fact,  $P^2$  on  $\mathcal{C}_k(S)$ .

We can see (4.51) as a second order tensorial approximation of the vector  $S' - C_k(S)$  collection (figure 4.8), whose orientations and lengths are taken into account. This idea has been inspired by [Advani and Tucker 1987].

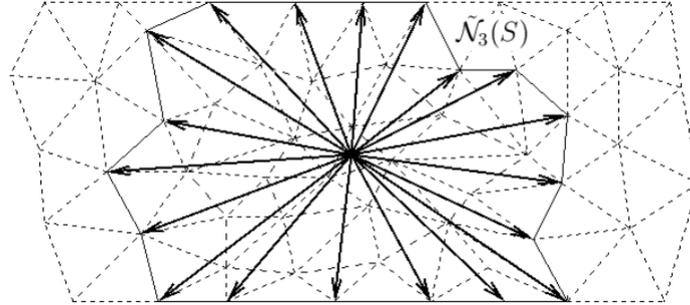


Figure 4.8: Vectors  $S' - C_3(S)$

According to the following result, the scalar  $\mu_k$  given by (4.52) is the best multiplicative factor.

**Proposition 4.5**  $\mu_k$  is the minimum of the scalar function

$$f(\mu) = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\|S' - C_k(S)\|_{\mu M}^2 - 1)^2 \quad (4.53)$$

This is the same function of the preceding section, except that only one degree of freedom remains.

**Proof :** we have

$$f(\mu) = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\|S' - C_k(S)\|_{\mu M}^2 - 1)^2 \quad (4.54)$$

$$= \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\mu \|S' - C_k(S)\|_M^2 - 1)^2 \quad (4.55)$$

$$= \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (\mu^2 \|S' - C_k(S)\|_M^4 - 2\mu \|S' - C_k(S)\|_M^2 - 1) \quad (4.56)$$

so,  $f$  is quadratic and

$$\frac{df}{d\mu}(\mu) = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \frac{d}{d\mu} (\mu^2 \|S' - C_k(S)\|_M^4 - 2\mu \|S' - C_k(S)\|_M^2 - 1) \quad (4.57)$$

$$= \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (2\mu \|S' - C_k(S)\|_M^4 - 2\|S' - C_k(S)\|_M^2) \quad (4.58)$$

since there exists  $S' \neq C_k(S) \in \tilde{\mathcal{N}}_k(S)$ , it comes

$$\frac{df}{d\mu}(\mu) = 0 \quad \Leftrightarrow \quad \mu = \frac{\sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \|S' - C_k(S)\|_M^2}{\sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} \|S' - C_k(S)\|_M^4} \quad (4.59)$$

thus, the unique minimum of  $f(\mu)$  is given by (4.52).  $\square$

### 4.2.3 Elementary metric combination

An other way to compute the best interpolating metric has been given by [Janka 2002]. Firstly, the star operator (defined by (2.2) page 25) is used from the center  $C_k(S)$  to the hull  $\partial\mathcal{T}_k^{\text{ext}}(S)$  (figure 4.9)

$$\mathcal{T}_k(S) = \mathcal{T}^*(C_k(S), \partial\mathcal{T}_k^{\text{ext}}(S)) \quad (4.60)$$

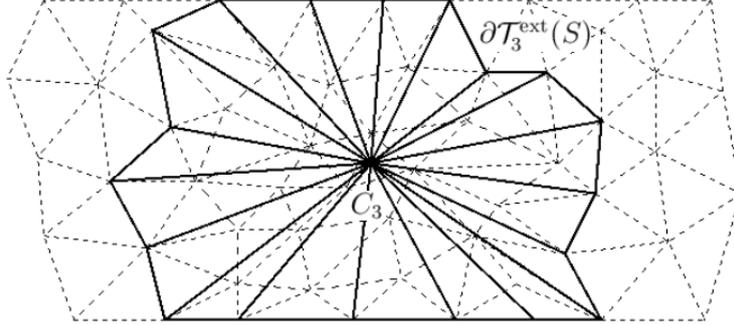


Figure 4.9: Star operating with the hull of order  $k = 3$

Then, the matrix  $M^{-1/2}$  is searched among all linear combinations of inverse transformations  $M_T^{-1/2}$ , where the elements  $T$  result from the star operating  $\mathcal{T}_k(S)$  (figure 4.9). According to [Janka 2002], if we consider

$$M_3 = \left( \frac{1}{\text{card}(\mathcal{T}_k(S))} \sum_{T \in \mathcal{T}_k(S)} M_T^{-\frac{1}{2}} \right)^{-2} \quad (4.61)$$

then eigenvectors of  $M_3$  are relevant but their eigenvalues should be corrected isotropically.

We denote by  $M_T = R_T D_T R_T^\top$ , the diagonalization of  $M_T$  and by  $M_3 = R D R^\top$ , those of (4.61), then better results are obtained when considering

$$M_4 = R \left( \frac{1}{\text{card}(\mathcal{T}_k(S))} \sum_{T \in \mathcal{T}_k(S)} D_T^{-\frac{1}{2}} \right)^{-2} R \quad (4.62)$$

#### 4.2.4 Comparisons

Here we compare  $M_2$  (obtained by orientation tensors),  $M_3$  and  $M_4$  (obtained by elementary metric combination), with a metric  $M_1$ , solution of the unconstrained optimization (according to the reduced system (4.29) page 62). The metric computation is performed on a mesh containing nearly 100 000 nodes, whose geometry represents all difficulties encountered in injection molding of thin parts (figure 4.10).

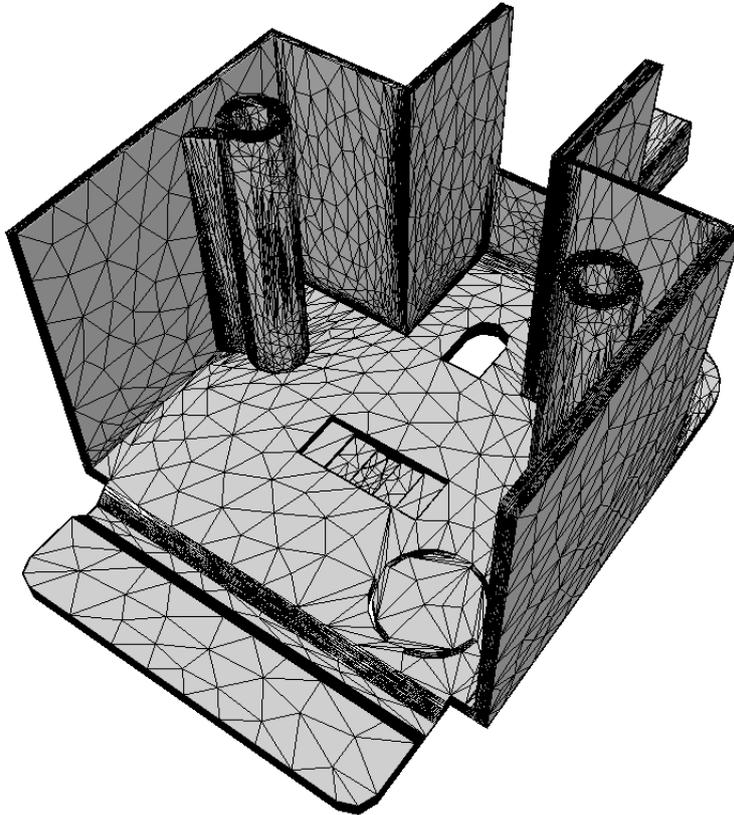


Figure 4.10: Test case mesh containing 96 033 nodes (courtesy of Schneider Electric)

The CPU time in table 4.1 takes into account the computation of the appropriate order (next section). It also takes into account the diagonalization used to divide mesh sizes (section 4.4).

interpolation type	CPU time en seconds on a 220 Mflops workstation	failure number	averaged interpolation error
$M_1$	1330	2710	0.86052
$M_2$	1668	0	0.80979
$M_3$	4048	12	1.82066
$M_4$	4526	12	0.87447

Table 4.1: Comparisons between four elliptic interpolation methods

With our implementation, the metric  $M_2$  is not only the less expensive (except  $M_1$ , but  $M_1$  fails 2710 times in elliptic matrix resulting), but also the most robust and accurate. That is the reason why we use it in priority, in the remainder. In practice, when  $M_2$  fails,  $M_4$  is tried and, if necessary,  $M_1$  also (with a projection on the constrained space).

### 4.3 Appropriate order

In fact, the elliptic interpolation is not computed for each neighborhood order  $k$ , but only for  $k$  between  $k_{\min}$  and  $k_{\max}$ .

Since the local geometry thickness has to be detected, the minimal order  $k_{\min} \geq 1$  is the first order, such that  $\forall k \geq k_{\min}$  the hull  $\mathcal{C}_k(S)$  locally obstructs the thickness. Computing  $k_{\min}$  is quite technical and described in appendix B.1.1 page 167.

Then, the neighborhood order is chosen as great as possible: while a not-so-bad elliptic interpolation of the hull  $\mathcal{C}_k(S)$  is reachable, the neighborhood growing continues.

#### 4.3.1 Interpolation error

The neighborhood growing stops at order  $k_{\max}$ , defined as the first order  $k = k_{\max}$ , such that the elliptic interpolation error is too high, according to the following criterion

$$\max_{S' \in \mathcal{N}_k(S)} \left| \|S' - C_k(S)\|_{M(S,k)}^2 - 1 \right| > 0.76 \quad (4.63)$$

where  $M(S, k)$  is the natural metric computed on node  $S$ , at order  $k$ .

In other words, the greatest order searching is finished when the hull becomes really non elliptic. In practice, 76% is an empirical threshold that discriminates well the elliptic or non elliptic shape of a hull.

For the determination of the 76% threshold, a unit square is considered in 2D (figure 4.11(a)), while a unit cube is considered in 3D (figure 4.11(b)) (with an additional node at the center). Since an ellipse

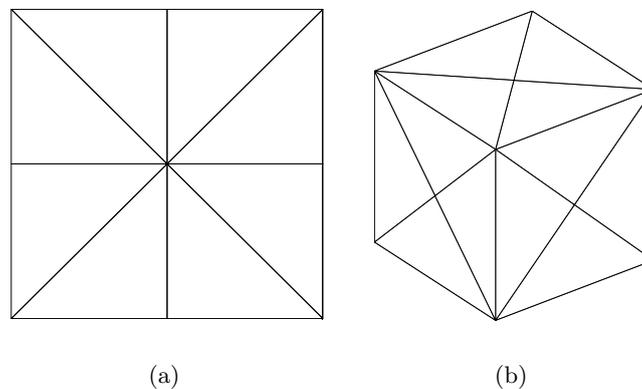


Figure 4.11: 2D and 3D meshes for determining a lower bound of the elliptic interpolation error threshold

does not interpolate perfectly a square and an ellipsoid does not interpolate perfectly a cube, the interpolation error in these cases is not zero. In fact, with the meshes of figure 4.11, this error is 0.757255 in 2D and 0.615385 in 3D. In other words, so as to accept the elliptic interpolation of those simple objects, the threshold value has to be greater than 0.757256.

Furthermore, for the T-shaped configurations of figures 4.12(a) and 4.12(b), (an additional node lies

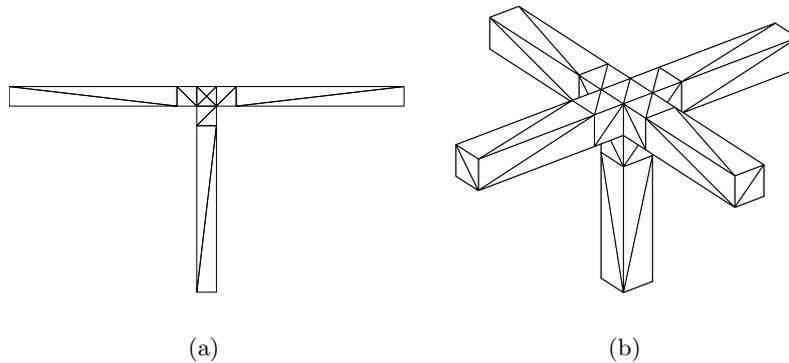


Figure 4.12: 2D and 3D meshes for determining an upper bound of the elliptic interpolation error threshold

inside the 3D geometry, at the T center), the elliptic interpolation errors at the center node are given on table 4.2. In these cases, we want a hull growing stopped at  $k = 1$ , since a T-shaped geometry is really

order	error in 2D	error in 3D
$k = 1$	0.000000	0.000000
$k = 2$	0.850210	0.761029
$k = 3$	0.964311	0.984728

Table 4.2: Elliptic interpolation error at different neighborhood orders

non elliptic. From 3D results at  $k = 1$ , it comes that the threshold must be lower than 0.761029. A value of 76% for the elliptic interpolation error threshold satisfies the lower and the upper bounds.

This empirical value works well in most situations, but pathological cases may be exhibited, where a threshold value of 76% does not fit. Further investigations should be performed, so as to distinguish the admissible noise of the hull  $C_k(S)$  around its elliptic interpolation and the loss of ellipticity of the local geometry, around the node  $S$ . It could lead to a threshold value that depends on the spatial dimension  $d$  and on the order  $k$ .

### 4.3.2 Final algorithm

Thus, the appropriate order for the natural metric is  $K = k_{\max}$  (except when  $k_{\min} > k_{\max}$ , in this case,  $K = k_{\min}$ ).

Finally, the computed natural metric is

$$M_{\text{natural}}(S) = \mu_K L_K^{-1} \quad (4.64)$$

$$\text{where } L_K = \sum_{S' \in \tilde{\mathcal{N}}_k(S)} \alpha_{S'} (S' - C_k(S)) \otimes (S' - C_k(S)) \quad (4.65)$$

$$\text{and } \mu_K = \frac{\sum_{S' \in \tilde{\mathcal{N}}_K(S)} \alpha_{S'} \|S' - C_K(S)\|_{L_K^{-1}}^2}{\sum_{S' \in \tilde{\mathcal{N}}_K(S)} \alpha_{S'} \|S' - C_K(S)\|_{L_K^{-1}}^4} \quad (4.66)$$

According to the previous section (and the following section), the full algorithm to compute the natural metric field  $M_{\text{natural}}(P^1)$  is as follows.

**Algorithm 4.1** after preliminary works on the mesh, dedicated to the neighborhood computation (annexe A.3.2 page 161), the applied procedure is:

**for** each node  $S$  **do**

**for**  $k$  from  $k_{\min}$  to  $k_{\max}$  **do**

- determinate the  $k$ -th order neighborhood of  $S$
- interpolate the hull  $\mathcal{C}_k(S)$
- compute the elliptic interpolation error, so as to update  $k_{\max}$

**done**

- diagonalize the obtained metric
- divide selectively the mesh sizes in the thickness direction

**fait**

Algorithmic complexity can hardly be evaluated, since  $k_{\max}$  strongly depends on the local geometry and its current mesh. For instance, for a cube mesh with 100 000 nodes,  $k_{\max}$  is much greater than for another mesh with 100 000 nodes, involving thin zones. The former situation, where each node sees all other nodes, is particularly expensive for the natural metric computation.

Furthermore, with the same geometry, but for different numbers  $N$  of nodes, the algorithm 4.1 is not linear, since  $k_{\max}$  depends on  $N$ . In the worst case (consisting in meshes obtained by successive isotropic refining),  $k_{\max}$  depends linearly on  $N$ . In the best case (consisting in meshes obtained by successive refining in only one direction),  $k_{\max}$  depends linearly on  $N^{\frac{1}{d}}$ .

These observations about  $k_{\max}$  are also true for the number of faces involved in the hull  $\mathcal{C}_k(S)$ , which is also the cardinal of  $\partial\mathcal{T}_k^{\text{ext}}(S)$ . At worst, the algorithm complexity is proportional to  $N^3$  and, at best, proportional to  $N^{1+\frac{2}{d}}$ .

However,  $k_{\min}$  may have the same order than  $k_{\max}$ , which reduces the number of operations. Still, the order range  $k_{\max} - k_{\min}$  remains unknown a priori, so, a numerical study should be run to compute the algorithm complexity.

Computing the natural metric only at very interior nodes and transporting it to the rest of the mesh could save numerous operations. However, the local anisotropy detection could be degraded.

## 4.4 Several element layers through the thickness

At this stage, the local anisotropy of the geometry is known, thanks to the natural metric. In particular, the thickness direction and the thickness size are embedded in the natural metric (without any skeleton technique, employed for instance by [Tchon *et al.* 2003]). This thickness size has to be divided, in order to introduce several element layers through the thickness, without introducing too many nodes in other directions.

### 4.4.1 Selective division

We denote by  $h_1^{-2} \geq \dots \geq h_d^{-2}$ , the eigenvalues of  $M_{\text{natural}}(S)$  (in descending order) and we consider the rotation  $R$ , such that

$$M_{\text{natural}}(S) = R \begin{pmatrix} \frac{1}{h_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{h_d^2} \end{pmatrix} R^\top \quad (4.67)$$

So as to obtain  $n$  element layers through the thickness (where  $n$  is an end user value), one mesh size has to be divided by  $n$ , the one corresponding to the thickness, namely  $h_1$ . However, when the geometry shows a local isotropy, that is, when  $M_{\text{natural}}(S)$  has almost the same sizes in all directions,  $h_1$  is not the only one to be divided, but also  $h_2, \dots, h_d$  (in order to generate isotropic elements).

In practice, the sizes  $h_1, \dots, h_p$  to be divided are those who lies between  $h_1$  and  $(1 + \alpha_h)h_1$  where  $\alpha_h > 0$  is a security factor. Empirically, we take  $\alpha_h = 1$ , which implies that a metric is anisotropic when its sizes differ with a factor greater than 2. Then, the sizes  $h_1, \dots, h_p$  are divided by  $n$ , while  $h_{p+1}, \dots, h_d$  remain unchanged. Finally, the following selectively divided natural metric is obtained

$$M_{\text{natural}}^n(S) = R \begin{pmatrix} \frac{n^2}{h_1^2} & & & & 0 \\ & \ddots & & & \\ & & \frac{n^2}{h_p^2} & & \\ & & & \frac{1}{h_{p+1}^2} & \\ 0 & & & & \ddots & \\ & & & & & \frac{1}{h_d^2} \end{pmatrix} R^\top \quad (4.68)$$

### 4.4.2 Iteration process

Let us consider the part of figure 4.13(a) (which is a biomedical application, only one half is modelled, for symmetrical reasons). This part is a test case for Rem3D software validation [Silva and Coupez 2002].

The first mesh generated by topological optimization, driven by a natural metric with  $n = 4$  element layers required through the thickness, does not entirely respects this requirement (figure 4.13(b)). So as to obtain a suitable mesh (figure 4.13(e)), the process has to be iterated. It means that the natural metric is computed again on the new mesh and that the topological optimization is performed again to achieved another adaption to this new metric.

We observe that the topological optimization also concerns the mesh boundary, especially on the symmetry plane. The boundary mesh is not optimized separately, but reflects the trace of the interior

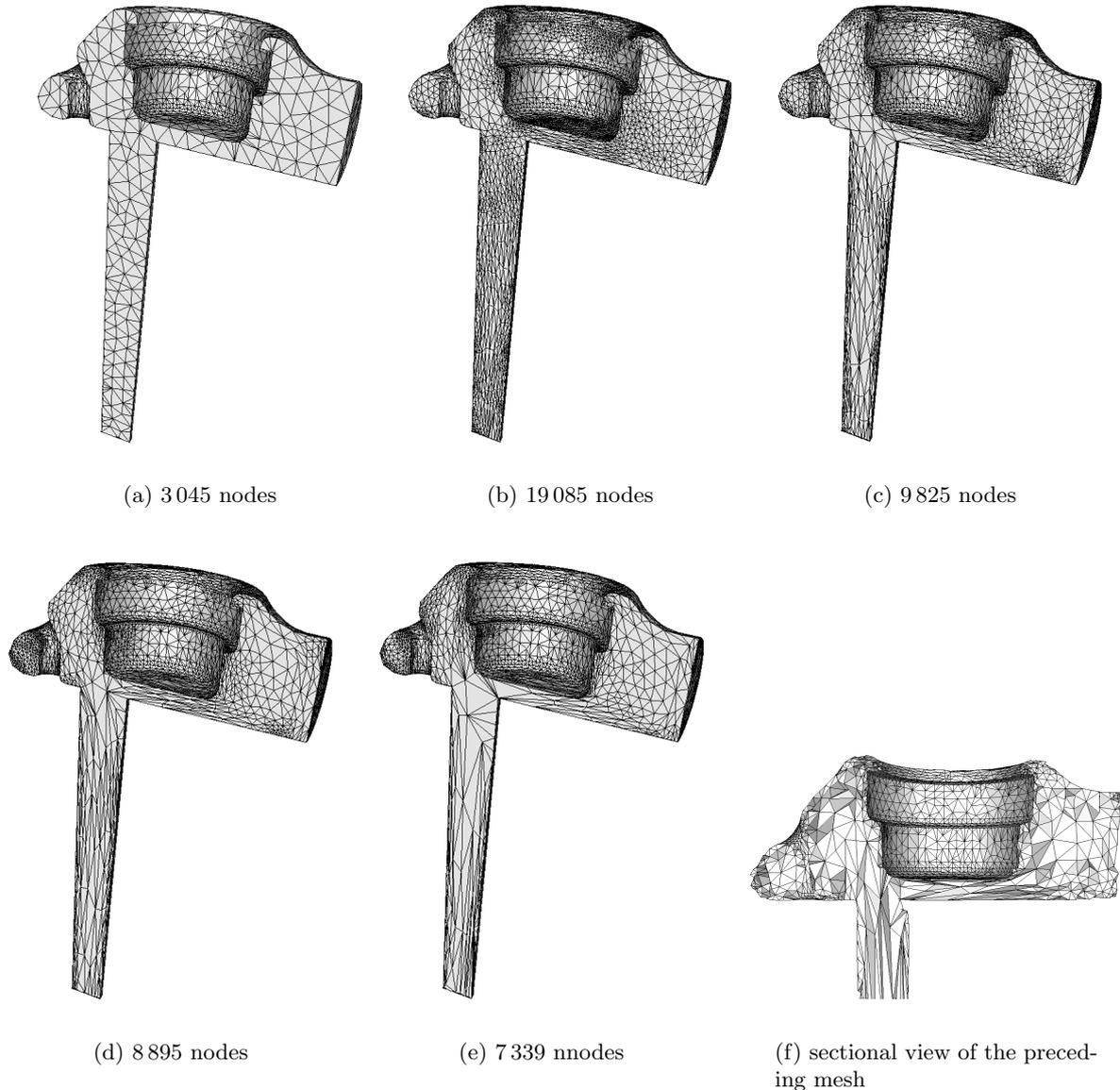


Figure 4.13: Initial mesh (a) and first iterations (courtesy of Sophysa), with a sectional view (f) of step (e)

natural metric. Unfortunately, this trace is sometimes delusive. For instance, the interior mesh of figure 4.13(f) is more refined than we may think, when we only look at its boundary (figure 4.13(e)).

Two indicators can be used to quantify the mesh evolution during the iterative process: the number of nodes and the averaged number of element layers through the thickness (computation of the latter being described in appendix B.1.2 page 168). These indicators are plotted on figure 4.14, for the first 16 iterations (the geometry of figure 4.13 being considered). These curves lead to several remarks:

- iteration 0 is the initial mesh, so, the peak is attained at the first adapted mesh (figure 4.13(b)); this peak is always observed (with all geometries), its presence must then be taken into account, especially when memory is allocated for the adaption process
- the averaged number of layers converges, but to an asymptotical value lower than the requirement (3 instead of 4); this is a usual phenomenon: the partially divided natural metric tends to coarsen the mesh too much, when too many iterations are performed (we recommend about 4 iterations only)

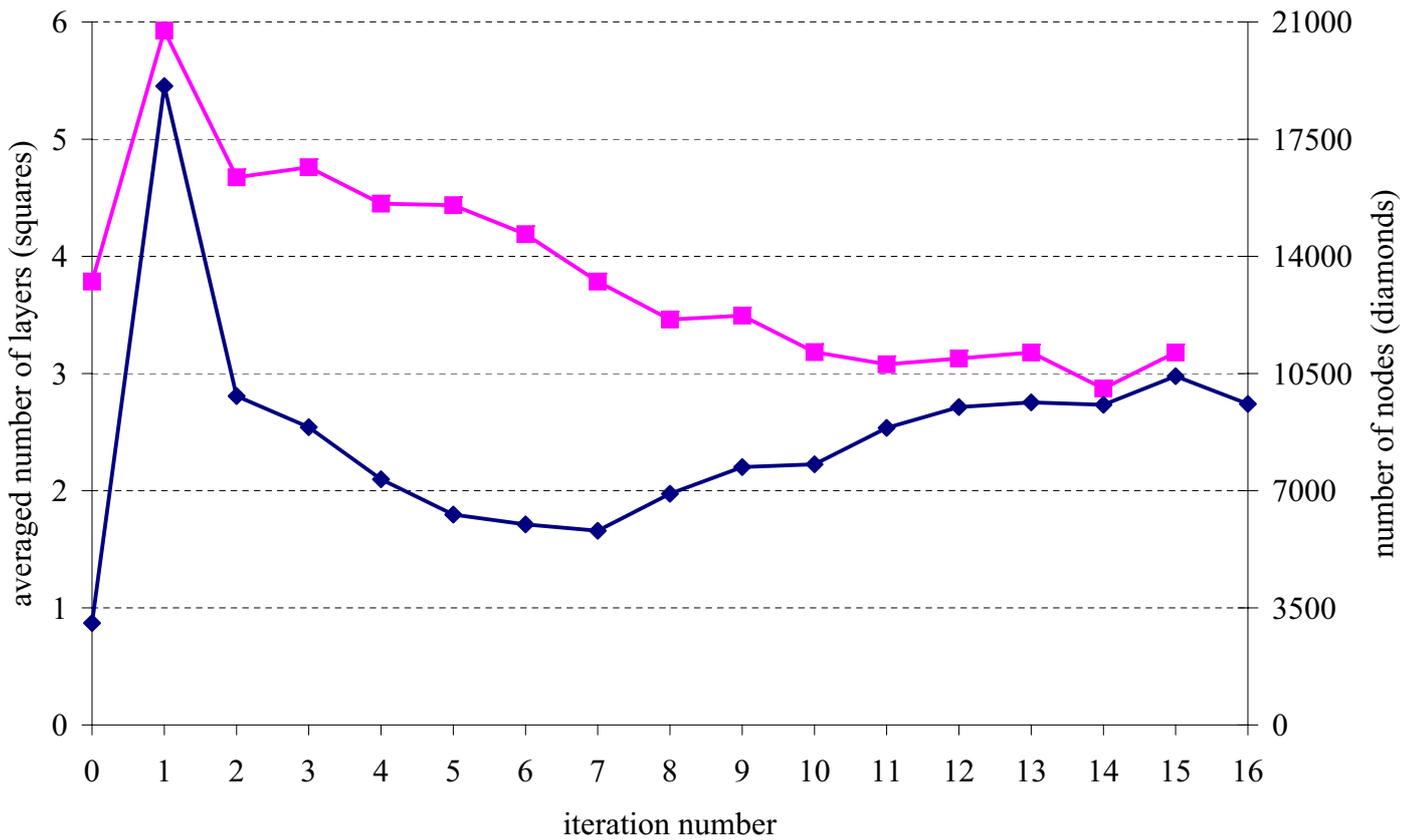


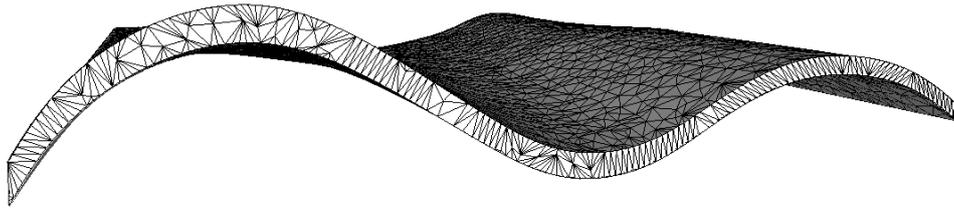
Figure 4.14: Convergence history of the iterative process introducing several element layers through the thickness

- the number of nodes hardly converges, which expresses a lack of local stability; thus, a full stabilization should not be expected for the process.

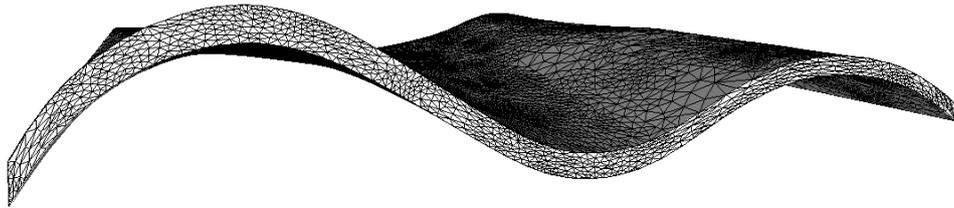
The preceding geometry is quite massive, but industrial plastic parts are generally thin and curved, as the one of figure 4.15. This configuration is not manageable manually, so, this is an important challenge for an automatic tool.

So as to correctly interpolate the curvature, the boundary mesh is particularly fine. The boundary mesh size is even lower than the thickness size, which leads to an anisotropic initial mesh (figure 4.15(a)), but its anisotropy is opposed to the natural one. For that reason, the first iteration gives a merely isotropic mesh (figure 4.15(b)), which emphasizes the peak about the number of nodes. Fortunately, after 3 or 4 iterations, we get a suitable mesh, with curved element layers (figures 4.15(d) and 4.15(e)). In order to show the good quality of the natural mesh inside the domain, a sectional view of the final mesh is displayed (figure 4.15(f)).

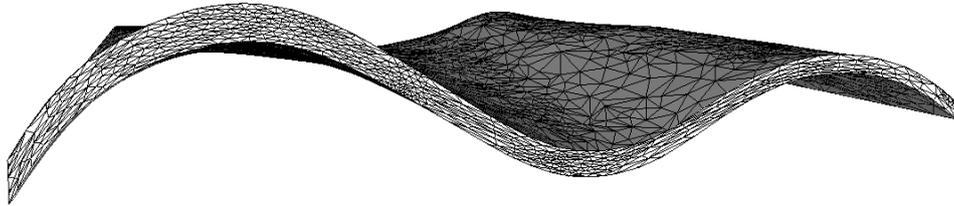
Memory consumption of the algorithm is linear with the number of nodes. However, the peak evoked previously may block the iterative process, when an intermediate mesh needs more than available memory. In order to reduce this peak, we may not ask for  $n$  layers at first iterations, but rather 2, 3 and progressively  $n$  layers.



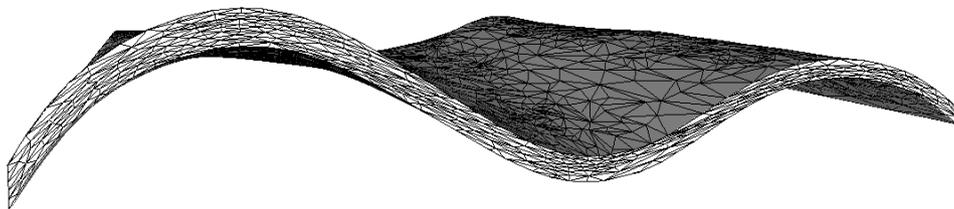
(a) 6 100 nodes



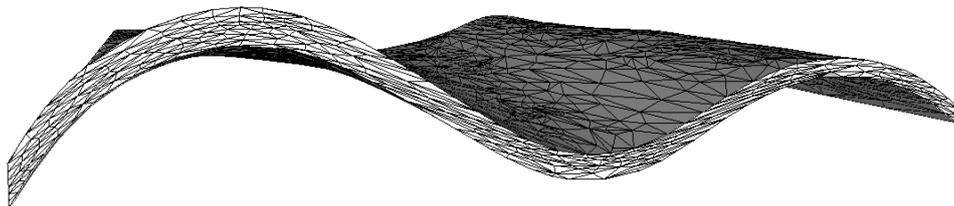
(b) 50 504 nodes



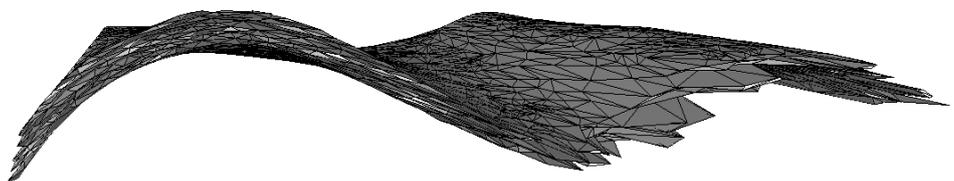
(c) 17 453 nodes



(d) 7 445 nodes



(e) 4 930 nodes



(f) sectional view of the preceding mesh

Figure 4.15: Initial mesh (a) and first iterations (academic geometry)

### 4.4.3 Multiple thickness

In 3D, a geometry can have not only one thickness size but two different thickness sizes (in other words, a thickness and a width). For instance, we consider the spiral mold of figure 4.16: its thickness

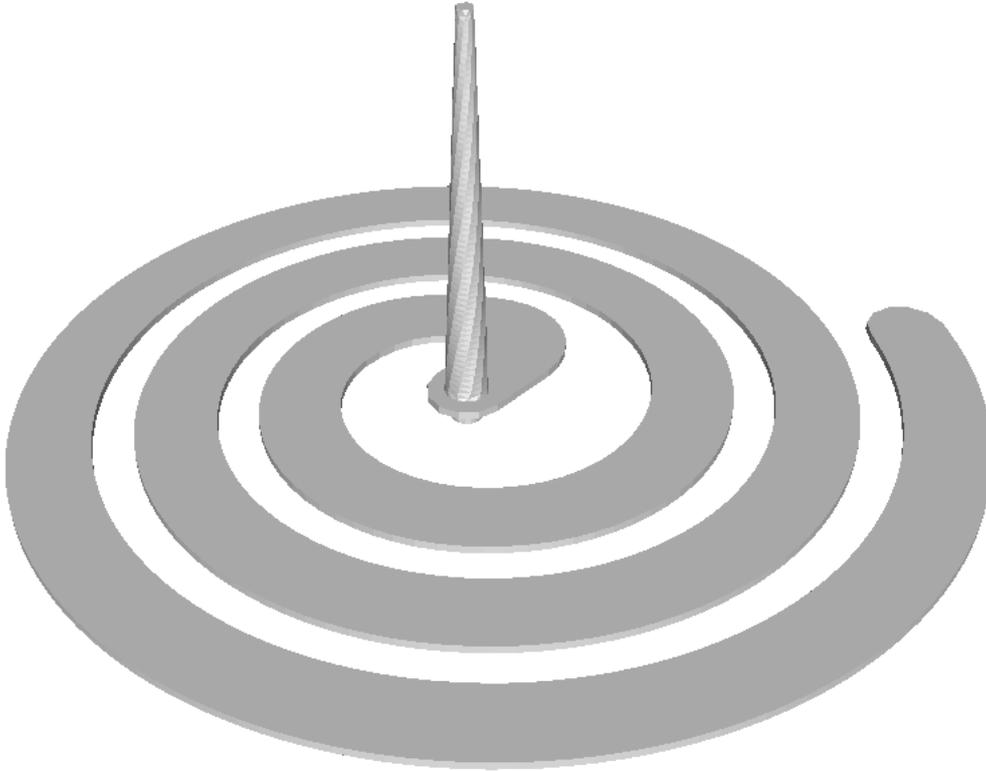


Figure 4.16: Spiral mold (courtesy of Atofina)

is 2 mm and its width is 20 mm. Several element layers have to be introduced, not only through the thickness but also through the width.

Without any treatment, several element layers are only introduced through the thickness (figure 4.17). A remedy consists in tuning the aforementioned security factor  $\alpha$ . For example, we can take  $\alpha = 9$  and get a better mesh (figure 4.18). In this case, for each selective division of the natural metric, mesh sizes between the smallest  $h_1$  and  $10 h_1$  are divided by  $n$ .

The end user may also want  $n_1$  element layers through the thickness,  $n_2$  through the width and  $n_3$  through the length (in 3D). For that purpose, two security factors  $\alpha_1$  and  $\alpha_2$  are needed, so that mesh sizes between  $h_1$  and  $(1 + \alpha_1) h_1$  are divided by  $n_1$ , those between  $(1 + \alpha_1) h_1$  and  $(1 + \alpha_1 + \alpha_2) h_1$  by  $n_2$ , others by  $n_3$ .

In 4D (space-time mesh), we could use a fourth number of layer  $n_4$  and a third security factor  $\alpha_3$ .

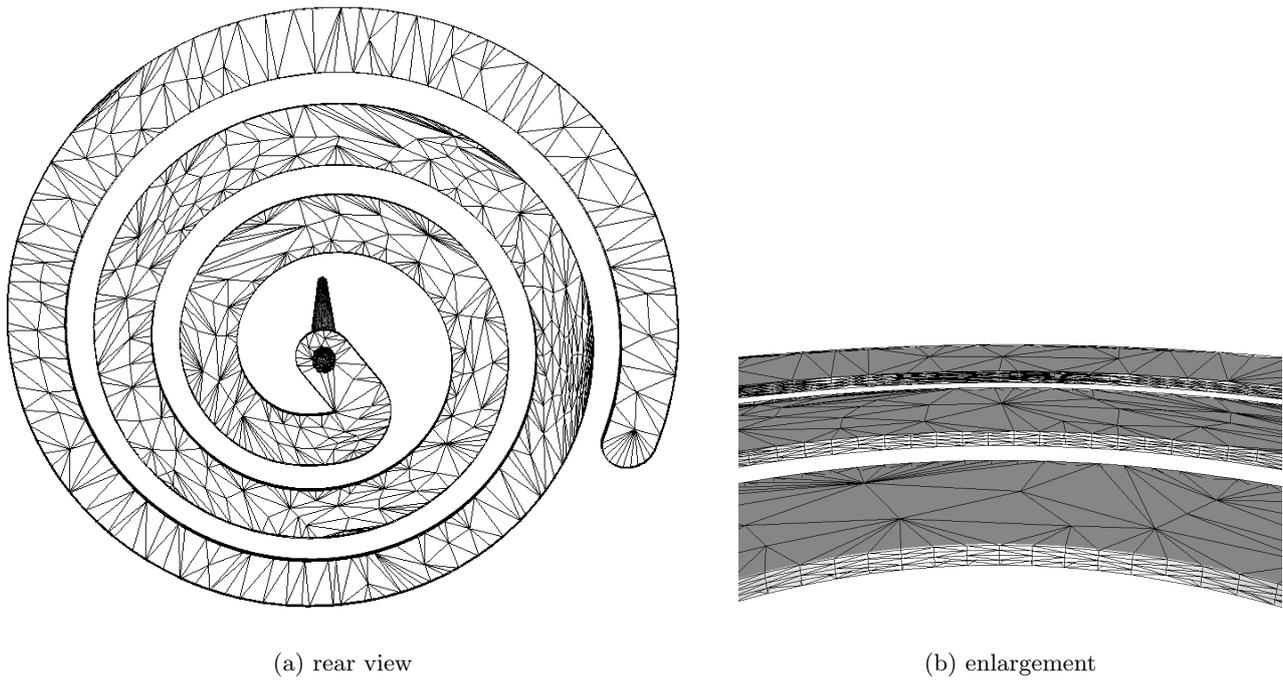


Figure 4.17: Mesh with several element layers through the thickness, but with few element layers through the width

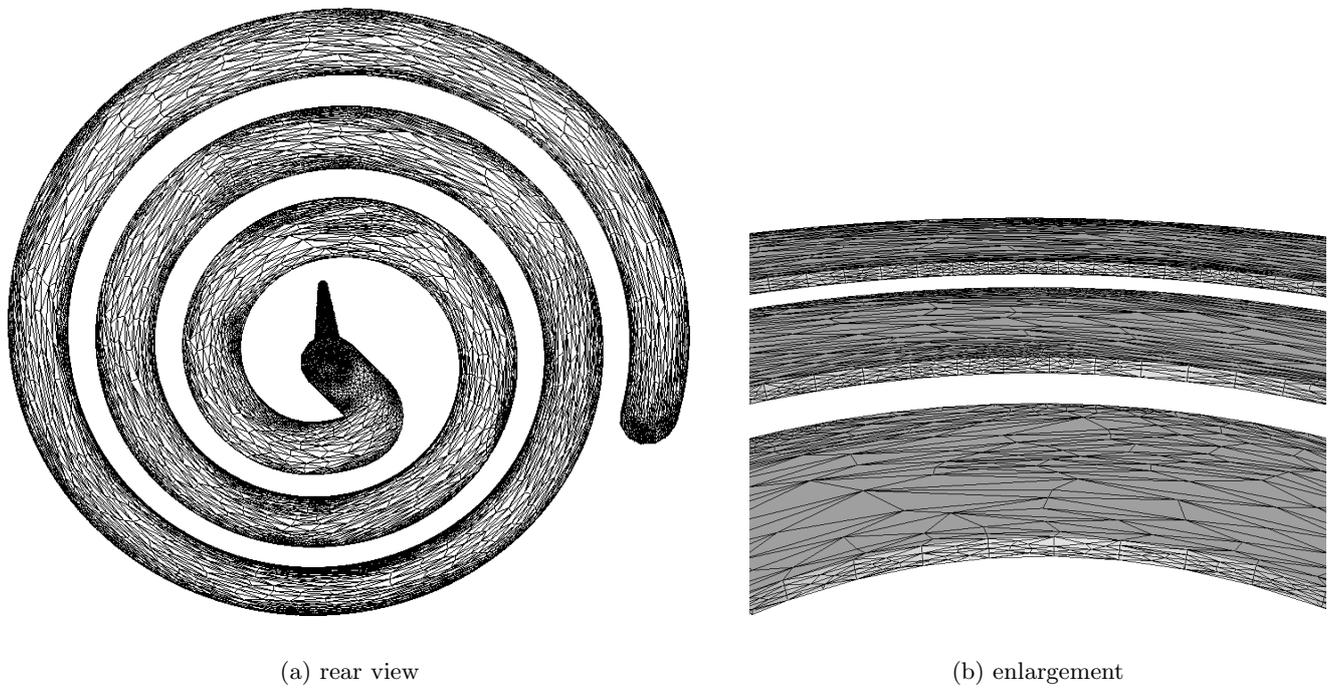


Figure 4.18: Mesh with several element layers through the thickness and through the width

## 4.5 Numerical applications

Apart from generating several element layers through the thickness, others applications are possible for the elliptic interpolation. It can be used for determining the orientation of a polyhedral grain, in polycrystalline studies [Meskini 2004]. It can also be used to progressively coarsen a mesh, in a multigrid framework [Janka 2002]. Here, we focus our attention on suitable mesh elaboration, dedicated to mold filling with Rem3D.

### 4.5.1 Natural mesh of cavities

Lots of Rem3D simulations concern only the mold cavity, where the fluid is injected. Injection molding leads to a final part with the same geometry than the mold cavity (except residual strain effects).

A natural metric can be used during the pre-treatment phase, so as to deal with the cavity geometry. In practice, when the process model is isotherm,  $n = 5$  element layers are enough to interpolate correctly a viscoplastic front profile with linear elements. When thermal coupling is considered,  $n = 8$  element layers can be requested, so that the low order heat transfer solver finds enough elements along the highest temperature gradient (along the thickness direction).

Here, we could not plot all the geometries already treated by a natural metric, since this tool is provided with Rem3D current versions and is widely used. Instead, we focus our attention on the most representative cases.

Let us start with the geometry of figure 4.2 page 56, which is composed of several plates whose thicknesses and orientations are not the same. Since the fluid flow, inside this cavity, is symmetrical, the non symmetrical gate has been removed and only half of the cavity has been meshed (figure 4.19).

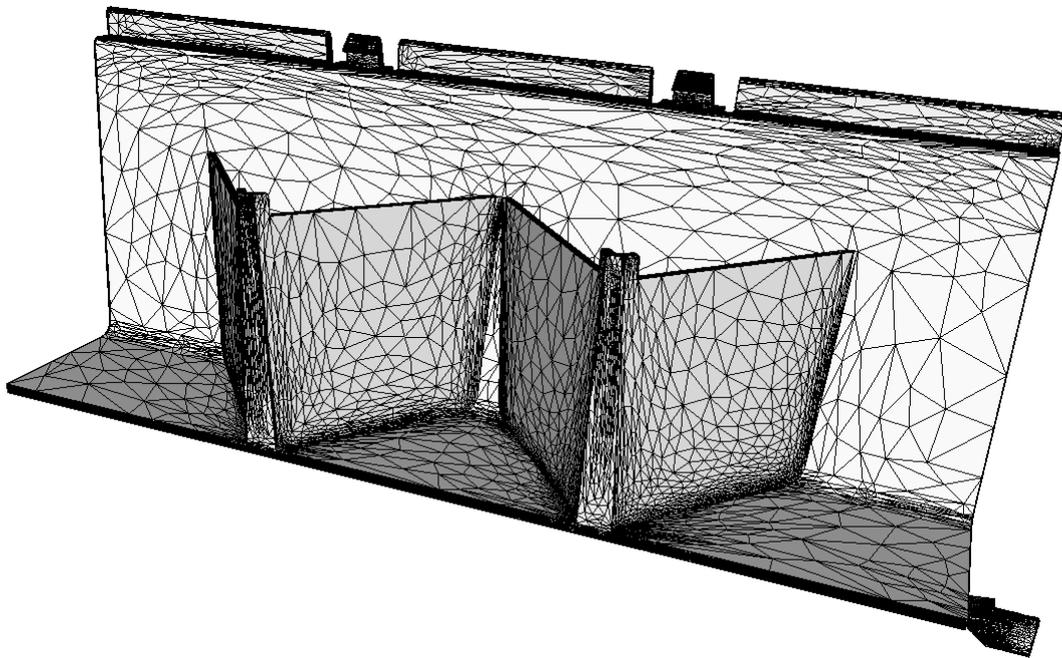


Figure 4.19: Natural mesh of a Plastic Omnium part (87 054 nodes)

With  $n = 8$ , we obtain between 8 and 14 element layers through the thickness (figure 4.20). Transitions are correctly treated when anisotropy directions change, especially at the interior of plate crossings (figure 4.21). A good mesh refinement inside these transition zones is very important, so as to make a 3D solver more accurate than a 2.5D solver.

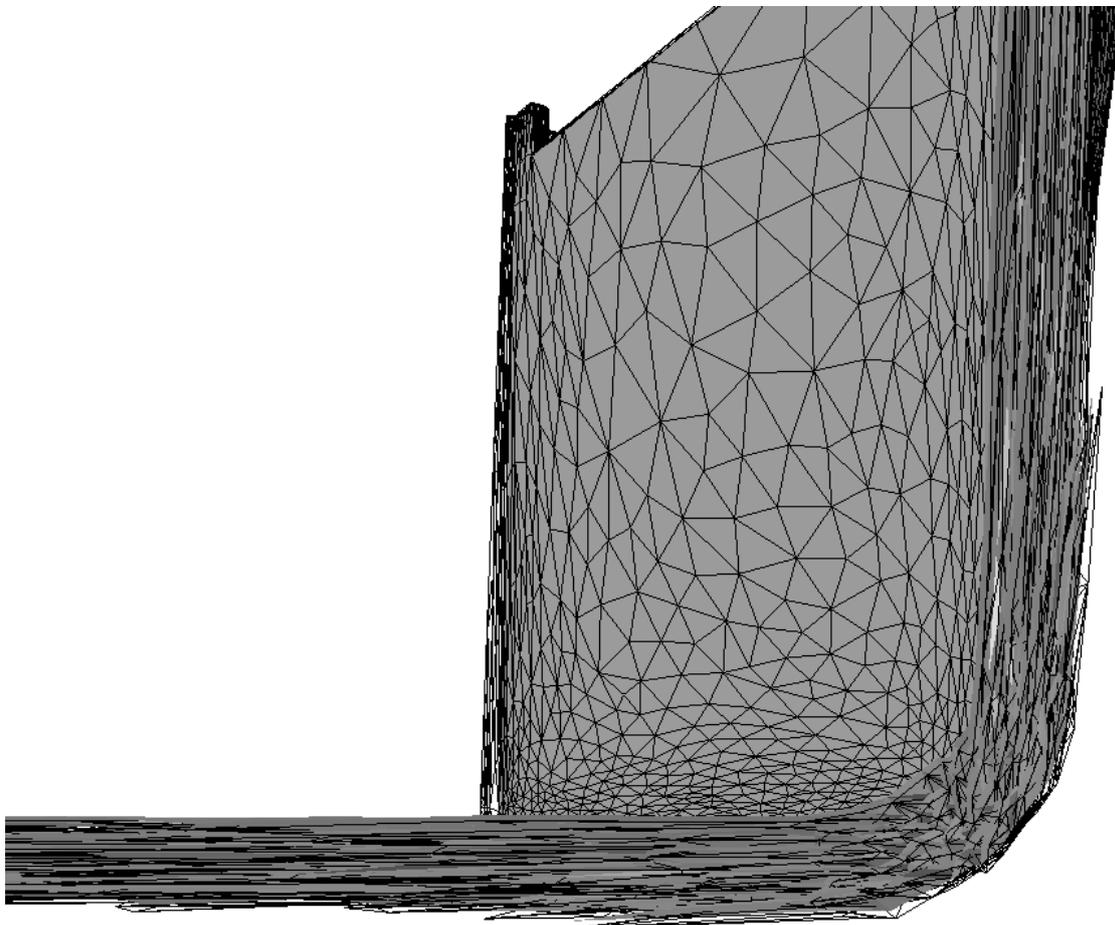


Figure 4.20: The same natural mesh (sectional view through the thickness)

Now, we consider the thin geometry of figure 4.4 page 57, whose particularity is a varying curvature. Thanks to the natural metric, several anisotropic element layers can be introduced automatically along the curvature (figure 4.22). This type of curved and thin geometries was not easy to manage before.

In what concerns injection gates, an accurate computation is important to determine when the flow is frozen during the packing and cooling stages [Silva 2004]. Since the geometry is very thin around injection gates, a coarse mesh, even if a natural metric is employed (figure 4.23(a)), would lead to an overestimation of the cooling. Thus, a very fine mesh is required in the vicinity of injection gates and we recommend a manual correction of the natural metric by a more isotropic and finer metric (figure 4.23(b)).

However, the robustness of the natural metric algorithm is enough to treat any industrial geometry, even a very complex one in 3D (figure 4.24). Without the natural metric, a Rem3D computation could not have been performed on such a part.

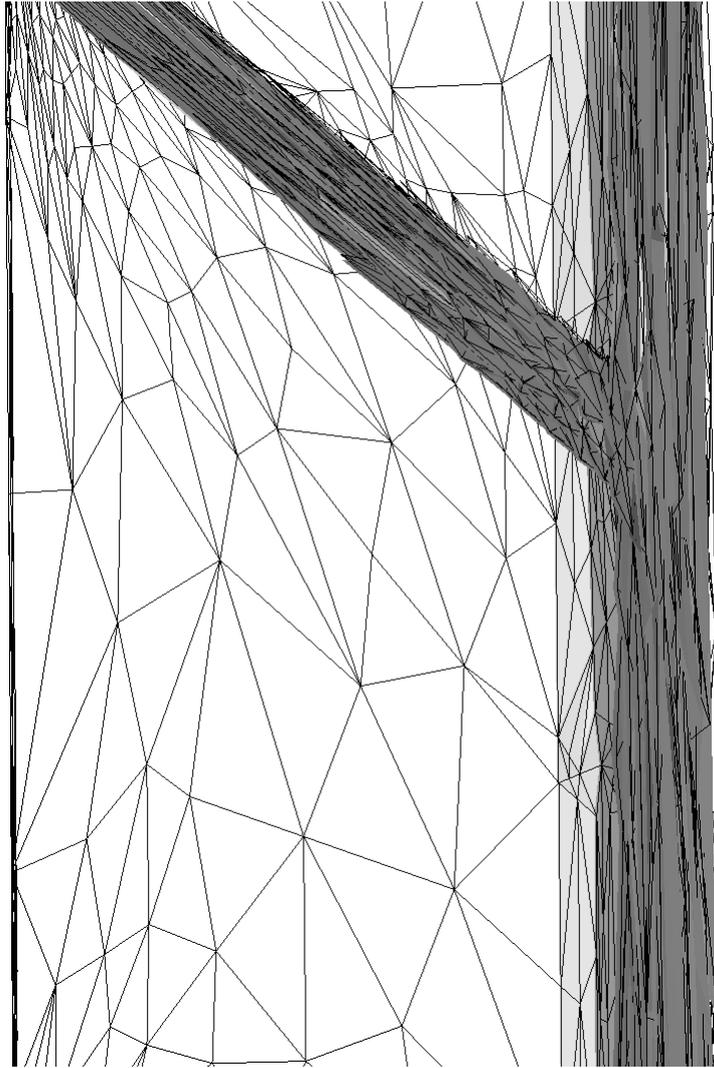


Figure 4.21: Correct transition between a vertical and a  $45^\circ$  oriented anisotropy

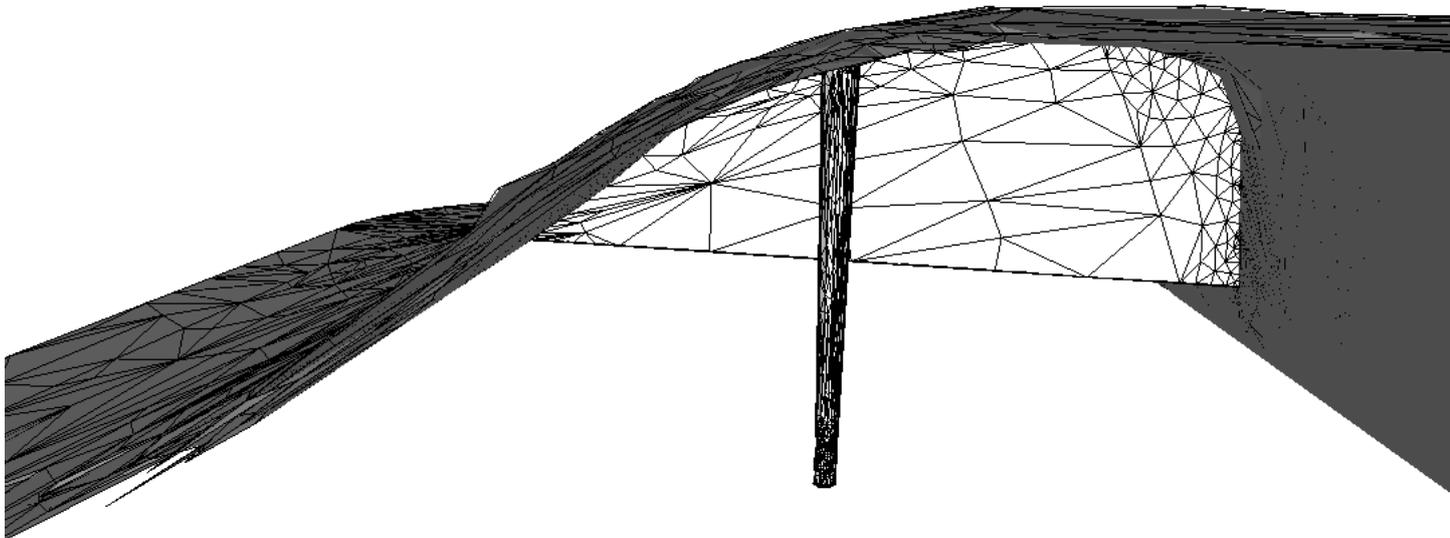


Figure 4.22: Natural mesh of a curved and thin geometry (sectional view)

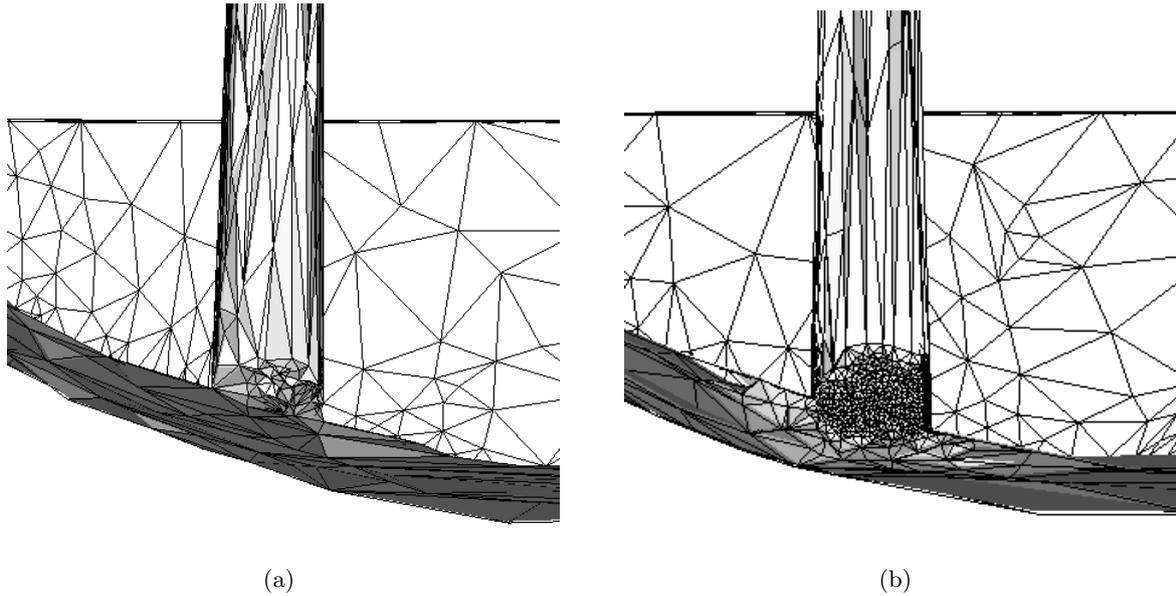


Figure 4.23: Manual correction of the natural metric in the vicinity of an injection gate (sectional view)

#### 4.5.2 Limitations and conclusion

The natural metric computed a priori leads usually to suitable meshes for Rem3D, but sometimes those meshes are not perfect. In fact, four different limitations have been identified. Illustrations of those limitations are given on figure 4.25 in 2D, so as to fully understand what can happen. However, the same phenomena are observed in 3D.

The first problem concerns crossing zones (figure 4.25(a)). In this case, an isotropic transition is needed, instead, the natural metric chooses one dominating anisotropy. This phenomenon is accentuated during the iterative process. Such crossing zones are widespread in industrial parts.

The second problem concerns notch zones (figure 4.25(b)). In those zones, the thickness size changes suddenly and several element layers are still needed (so that the flow is not artificially blocked). However, this local zone is too small for the natural metric computation. Such notch zones are frequent in electro technical objects, involving almost cut pieces.

The third problem is raised by small details that are traditionally carved on the mold (figure 4.25(c)). This kind of detail are not big enough for the natural metric computation. And the flow is artificially blocked by the natural mesh.

The fourth problem is observed in curved geometries, when too many iterations are performed (figure 4.25(d)). In this case, element orientation does not follow the local curvature anymore. In fact, the local anisotropy detection is not stable enough during the iterative process.

Thus, in particular situations, a suitable mesh might not be reachable by an a priori natural metric. Some of these difficulties disappear in the multidomain framework (third part, page 97). However, in our monodomain framework, one could improve a natural mesh by a posteriori information, that is, information gathered after a first Rem3D computation (next chapter).

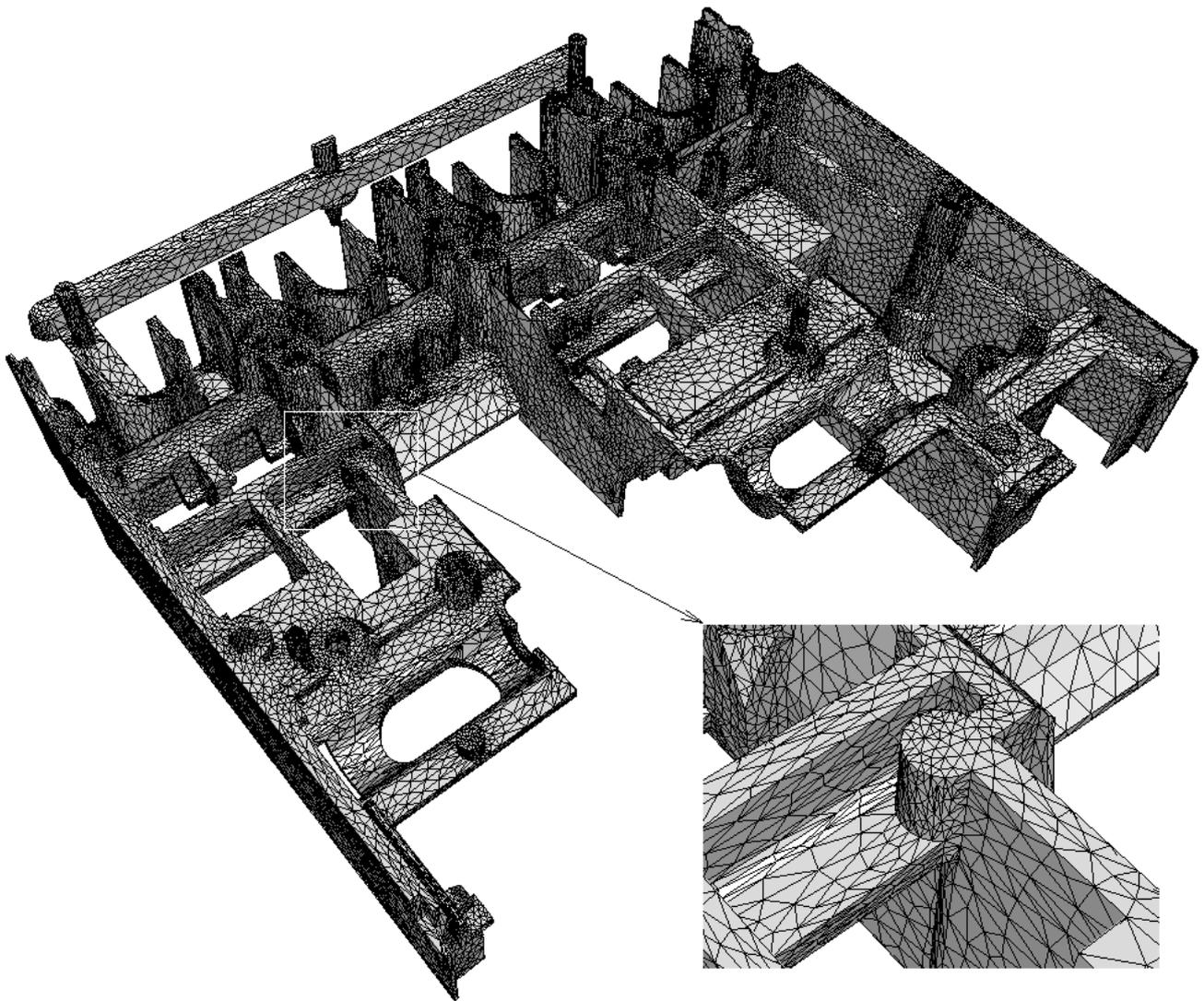
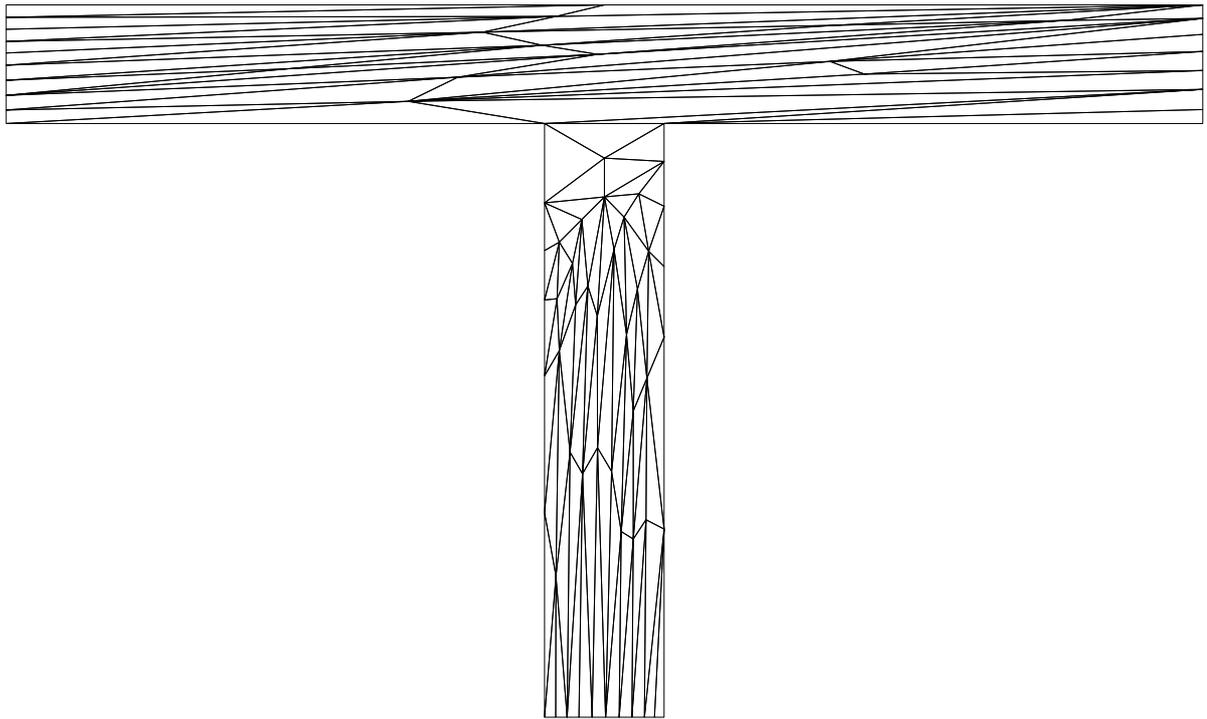
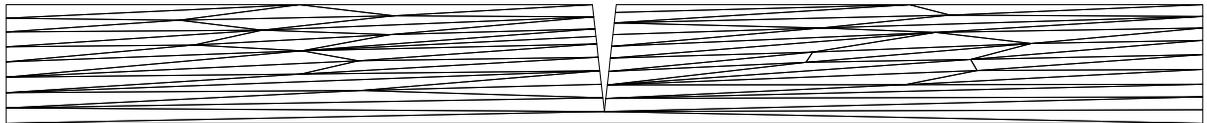


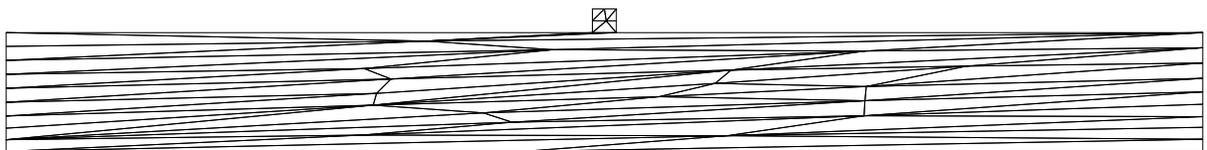
Figure 4.24: Natural mesh (174 759 nodes) of figure 3.13's part



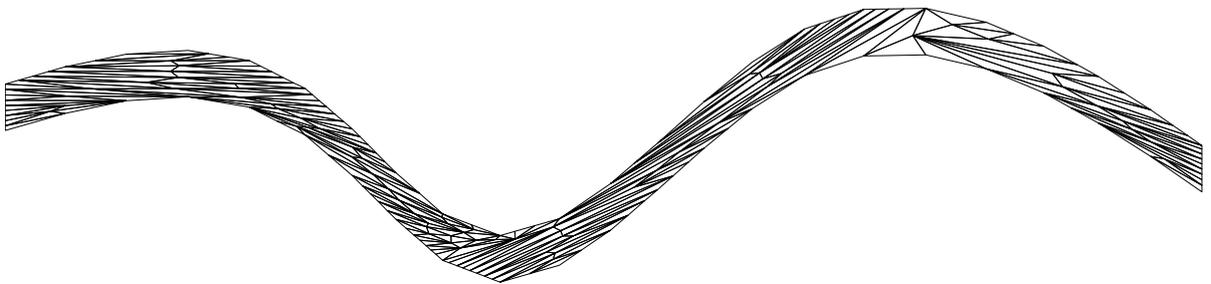
(a) crossing problem



(b) notch problem



(c) small detail problem



(d) curvature problem

Figure 4.25: Problems encountered by the a priori natural metric (each mesh is obtained after 16 iterations)

## Chapter 5

# Natural metric corrected a posteriori

With an a priori natural metric (previous chapter), limitations can be observed, preventing us from generating meshes that are perfectly suitable for Rem3D. We now look for improvements of such meshes, relying on a posteriori information, based on some computational results.

We still want to deal with anisotropic meshes. In this context, an anisotropic adaption can be based on a posteriori information, issuing from:

- state variables such as velocity or temperature, interpolated over the computational domain  $\Omega$ : these fields present their own heterogeneity; this field heterogeneity can be exploited to orient and stretch the elements of the mesh
- equation solving on  $\Omega$  leads to numerical errors, which can be measured and smoothed by a mesh refining.

Even not perfect, a mesh obtained by an a priori natural metric allows a preliminary computation to be performed, so as to obtain that kind of a posteriori information. Then, the mesh can be improved and the compulsory computation can rely on it. In this case, the natural metric is still an a priori metric, but corrected a posteriori.

In the literature, anisotropic adaption based on error estimation can be performed without metric, like in [Belhamadia *et al.* 2004]. In this paper, authors use a hierarchical error estimator, as a goal in topological mesh optimization. The error and its gradient are thus minimized.

However, anisotropy is generally treated by adaption to a metric, which is generated by a scalar field Hessian matrix. This Hessian matrix is then truncated by a minimal and a maximal mesh size [Tam *et al.* 2000, Leservoisier *et al.* 2001, Pain *et al.* 2001, Remacle *et al.* 2002, Alauzet *et al.* 2003]. As already mentioned, this approach (based on interpolation error estimation) induces that, when the Hessian matrix is hyperbolic, an arbitrary projection is applied on the elliptic matrix space.

Here, we propose an alternative (section 5.1) based on a tensorial product of a vector (the gradient of a scalar field, for instance). The advantages are:

- we cannot obtain a hyperbolic matrix
- a gradient computation is less difficult and more accurate than a Hessian matrix computation.

In what concerns residual error estimation, usual estimators can not deal with mesh anisotropy. We propose an approach in section 5.2 (page 91), dedicated to error smoothing by a priori metric refining.

## 5.1 Scalar field heterogeneity

In this section, we address the following issue: how to improve a mesh after some computation results have been obtained with it? Our answer is based on considering a scalar field  $u$ , whose anisotropy is taken into account. Then, the anisotropic adaptation described in chapter 3 (page 35) is used, so as to locally align mesh elements with the principal directions of  $u$ 's heterogeneity and to adapt mesh sizes in each direction.

In fact, all we need is  $\nabla u$ , the gradient vector of  $u$ :

- principal directions of  $u$ 's heterogeneity are the direction of  $\nabla u$  and the directions of  $d-1$  orthogonal vectors that are also orthogonal to  $\nabla u$
- the more the gradient norm  $\|\nabla u\|$ , the more we need small mesh sizes in the direction of  $\nabla u$ .

For other directions, the mesh sizes prescribed by the a priori natural can be kept. In this context, we propose to correct the natural metric by the tensor  $\nabla u \otimes \nabla u$  (section 5.1.2 page 86), which is an extension of a 2D study, obtained with the topological mesh generator [Liu 2004].

But previously, a typical computation is described in order to obtain a relevant scalar field  $u$  (next section).

### 5.1.1 Typical simulation

The basic idea is to compute a representative fluid flow, without processing a full Rem3D simulation (which would be transient and non linear). A linear Stokes problem is solved once (section 5.1.1.1), for a cavity full of fluid, with an imposed pressure at one entrance gate and a free pressure at one or several exit gates (other boundary conditions being zero velocity). Several relevant scalar fields are proposed, so as to adapt the mesh to the flow (section 5.1.1.2).

#### 5.1.1.1 Simulation conditions

We consider a cavity full of a Newtonian fluid, which is assumed to be incompressible, with sticky boundary conditions, except at symmetry planes, at entrance and at exit gates (as in figure 5.1). Inertia forces and gravity are assumed to be negligible, with respect to viscous forces. The fluid flow (with its aforementioned boundary conditions) obeys to the Stokes equations

$$\begin{cases} 2\eta \nabla \cdot \epsilon(v) - \nabla p = 0 \\ \nabla \cdot v = 0 \end{cases} \quad (5.1)$$

where  $\nabla \cdot$  is the divergence operator,  $\eta$  is the fluid viscosity,  $p$  is the pressure,  $v$  is the velocity vector and

$$\epsilon(v) = \frac{1}{2} \left( \nabla v + (\nabla v)^\top \right) \quad (5.2)$$

is the strain rate tensor.

Equation solving is performed via a mixed and stable finite element formulation: continuous  $P^1$  pressure and continuous  $P^1$  velocity, with a bubble term at each element center, dedicated to approximation stability (inf-sup condition). This formulation leads to the following linear equation system

$$\begin{pmatrix} A & B \\ B^\top & -C \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (5.3)$$

where  $U$  contains the velocity degrees of freedom,  $P$  contains the pressure degrees of freedom,  $A$  and  $C$  are two definite positive symmetrical matrices and  $F$  contains boundary conditions about pressure.

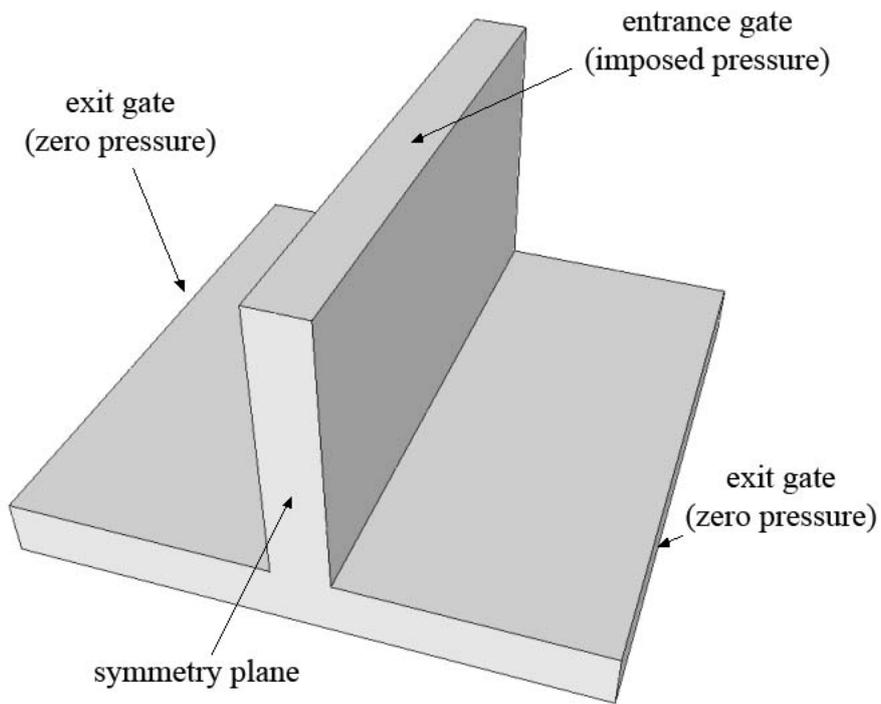


Figure 5.1: Boundary conditions

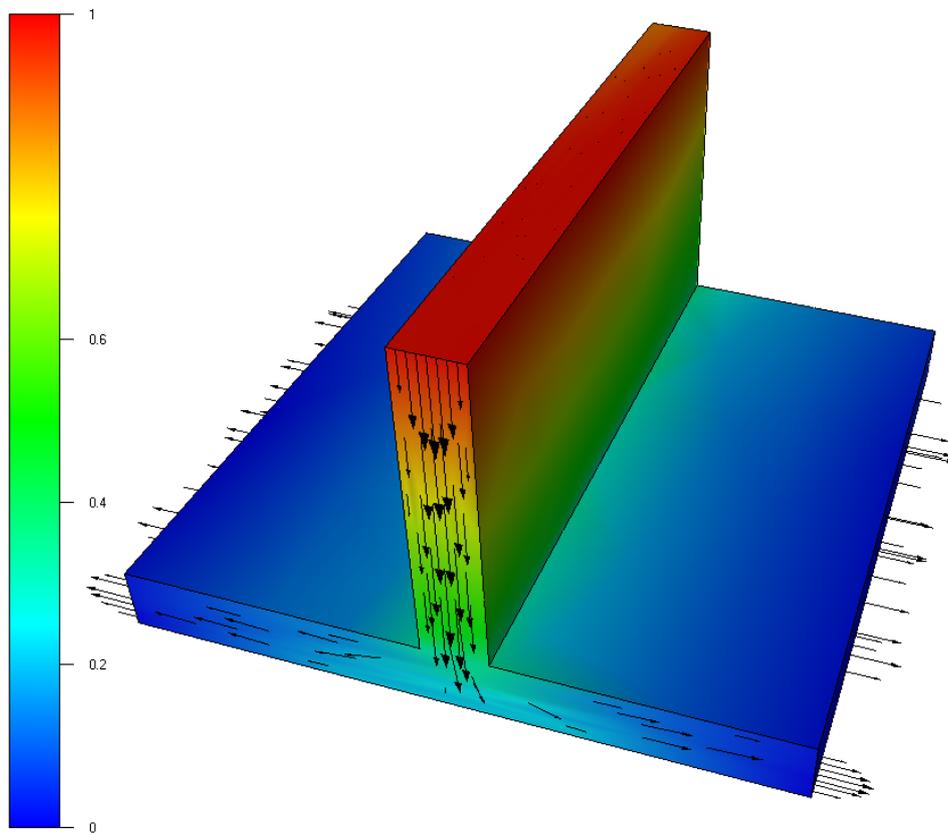


Figure 5.2: Velocity (vectors) and pressure (color scale)

Bubble terms are condensed in matrix  $-C$ , which makes the system (5.3) symmetrical but not definite positive. Thus, convergence of the iterative linear solver is not monotonous. System (5.3) is sparse and its size is four times the number of nodes in the mesh. It is solved iteratively by a conjugate residual method with ILU(0) preconditioning (ILU(0) stands for incomplete Gauss factorization, without enrichment).

A result is presented on figure 5.2. On the symmetry plane, we can see the Newtonian velocity profile at the entrance gate and at the exit gate. Furthermore, pressure decreases regularly along the flow.

### 5.1.1.2 Potentially interesting fields

Among all output fields, we look for a scalar field  $u$ , whose heterogeneity would lead to an improvement of the initial mesh. In our example, the initial mesh (figure 5.3(a)) is built with an a priori natural metric. However, this mesh does not have enough element layers in the crossing zone (figure 5.3(b)).

In this context, we need a scalar field whose level sets capture well this crossing zone. Pressure  $p$  is not a candidate, since its level sets are orthogonal to the flow. However, the acceleration norm  $\bar{\gamma} = \|(\nabla v) v\|$  has interesting level sets that capture quite well the crossing zone (figure 5.4). This scalar field can be used to improved the initial mesh.

On the other hand, the equivalent shear rate

$$\|\epsilon(v)\| = \left( \sum_{1 \leq i, j \leq d} \epsilon_{ij}^2(v) \right)^{1/2} \quad (5.4)$$

could be an acceptable candidate, but only in plate zones, not in the crossing zones (figure 5.5).

## 5.1.2 Natural metric correction

Once the computation is performed and once the scalar field  $u$  is chosen, this information has to be treated in our anisotropic adaption process.

### 5.1.2.1 Additional term

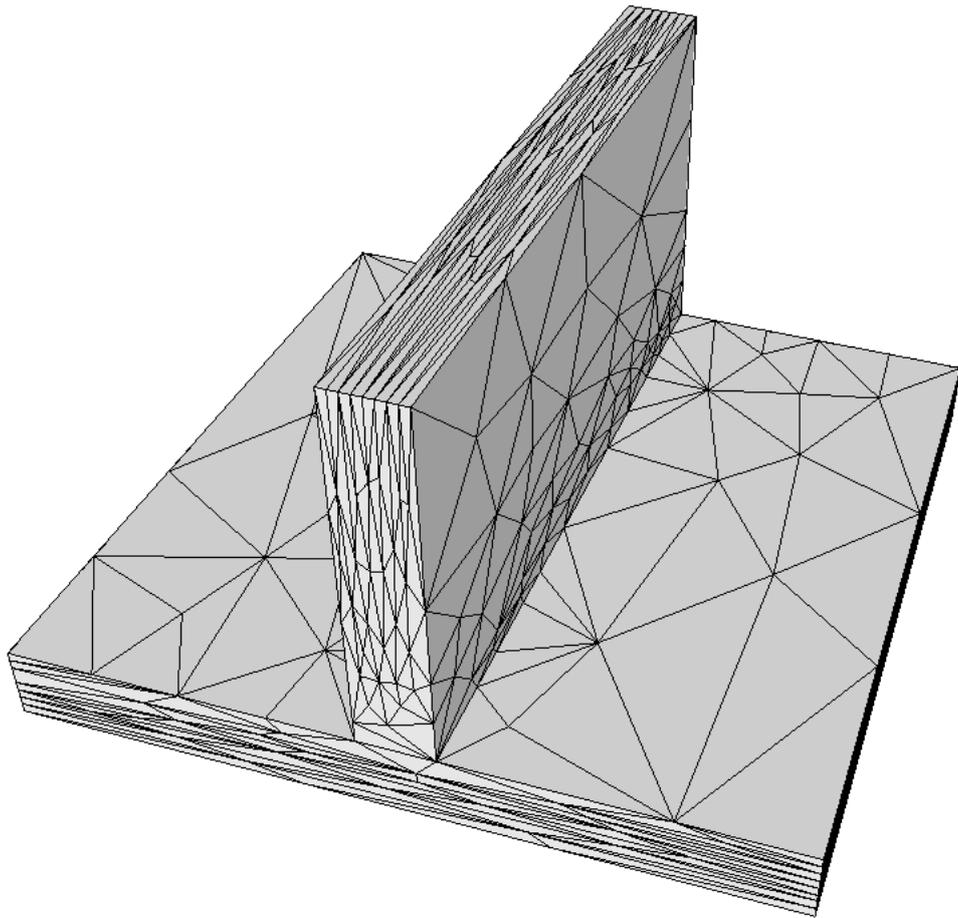
So as to adapt the mesh to  $u$ 's heterogeneity, we do not compute its Hessian matrix (which is difficult to compute from a continuous  $P^1$  field and more difficult from a discontinuous  $P^0$  field). We prefer to deal with its gradient vector. In fact, we use the following result, which has been proved within the multidomain framework (section 6.2.1 page 103).

The graph of  $u$ , can be seen as a surface in  $\mathbb{R}^{d+1}$ , which can be triangulated according to the following metric

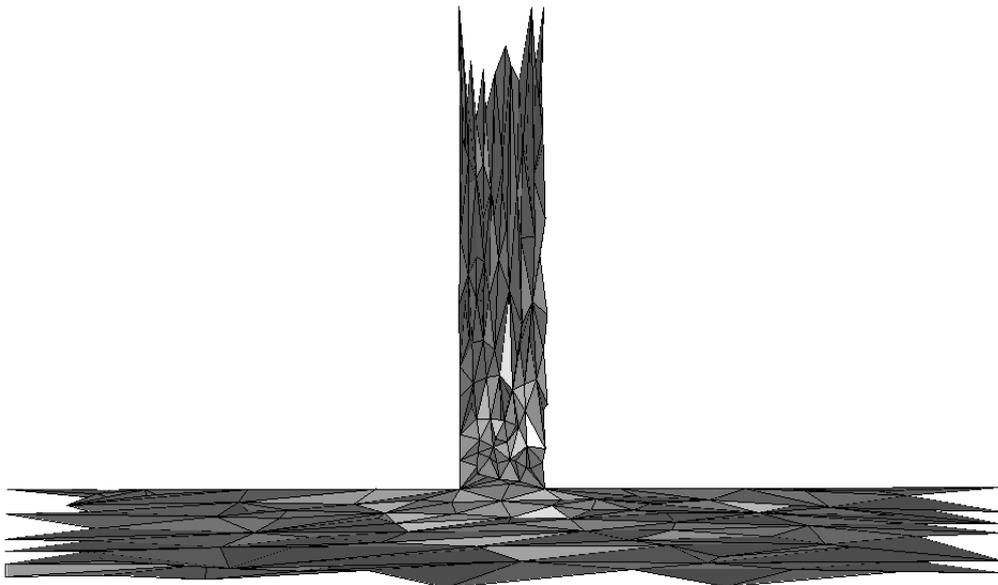
$$\begin{pmatrix} & & & 0 \\ & M_{\text{natural}}^n & & \vdots \\ & & & 0 \\ 0 & \dots & 0 & \mu^2 \end{pmatrix} \quad (5.5)$$

where  $1/\mu$  is the  $(d+1)$ -th mesh size that must be introduced, so as to make a triangulation in  $\mathbb{R}^{d+1}$ . This triangulation is then orthogonally projected on  $\Omega$ , in order to obtain a mesh of this computational domain. Proposition (6.1) page 104, tells us that it is equivalent to use directly the following metric on  $\Omega$

$$M_{\text{a posteriori}} = M_{\text{natural}}^n + \mu^2 \nabla u \otimes \nabla u \quad (5.6)$$



(a) global view



(b) sectional view

Figure 5.3: A natural mesh to be improved (1 327 nodes)

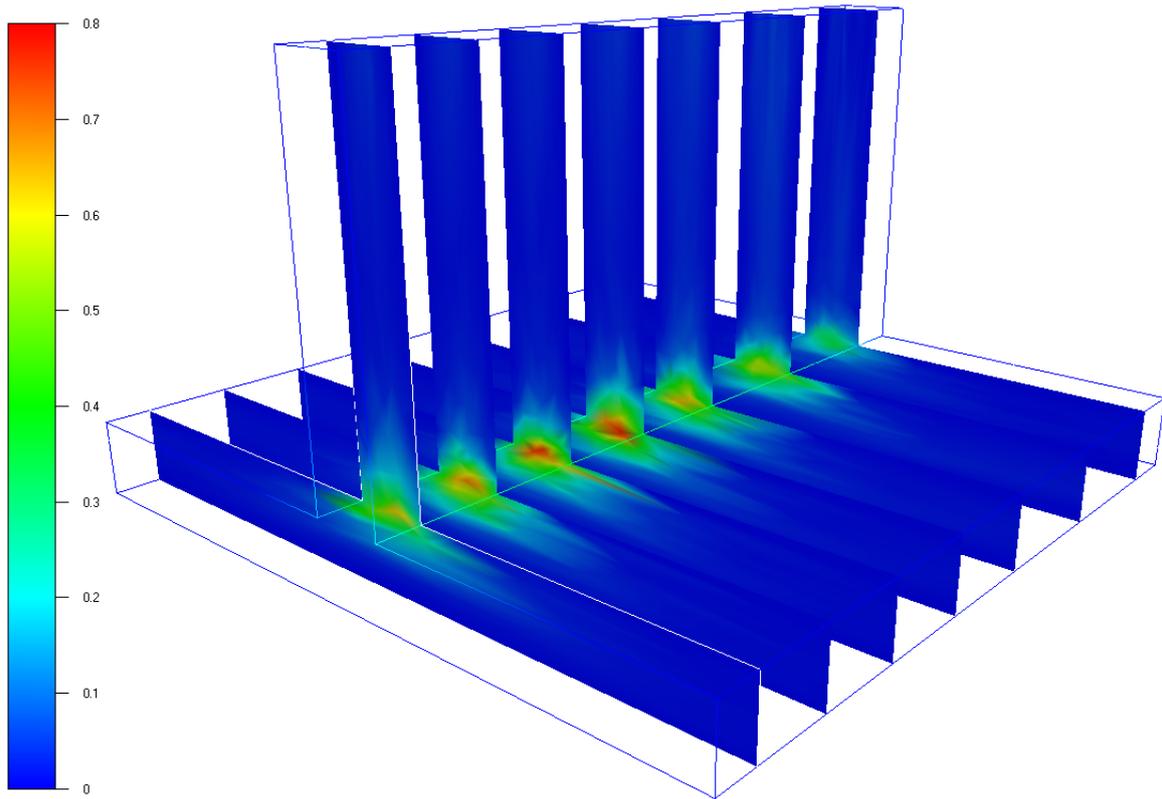


Figure 5.4: Acceleration values  $\bar{\gamma}$  on several cutting planes

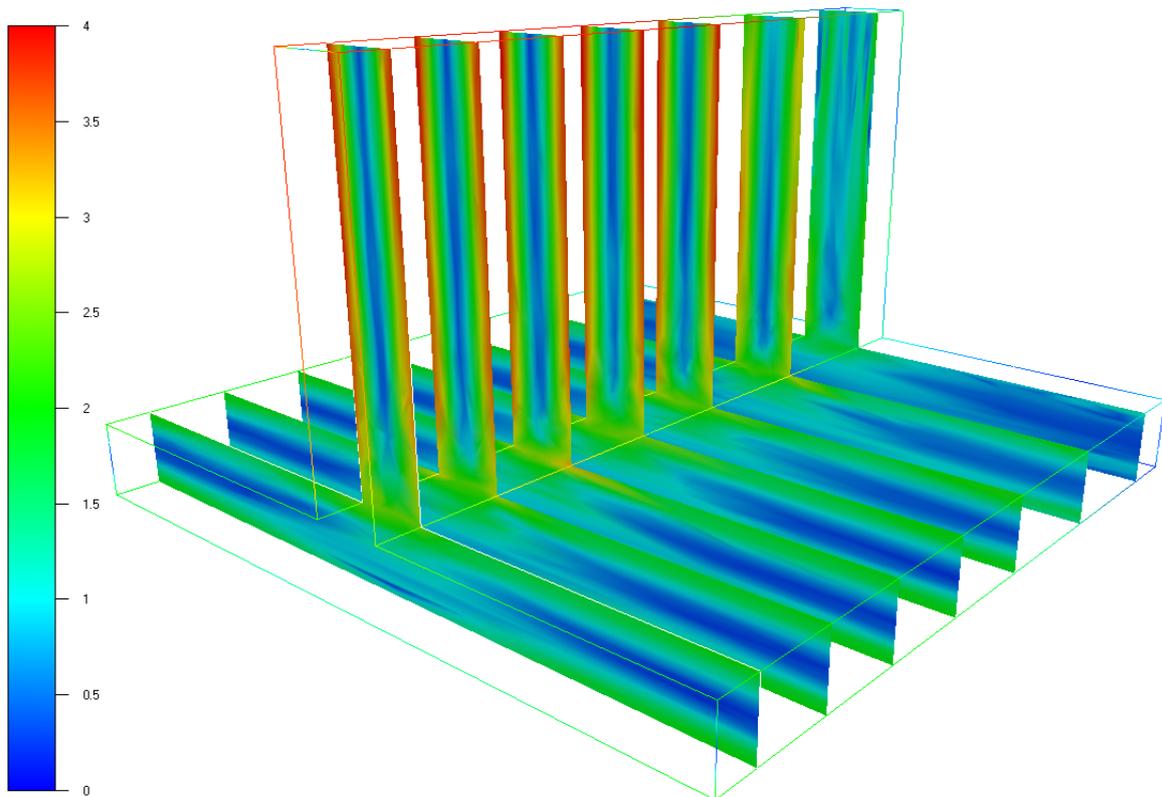


Figure 5.5: Equivalent shear rate values  $\|\epsilon(v)\|$  on the same cutting planes

The major difficulty is the tuning of parameter  $\mu$ . This scalar can be interpreted as a penalty coefficient that weights the terms  $\nabla u \otimes \nabla u$  versus the (a priori) natural metric, except where  $\nabla u$  is null. Empirically,  $\mu$  should be such that the greatest eigenvalue of  $\mu^2 \nabla u \otimes \nabla u$  is greater than all eigenvalues of  $M_{\text{natural}}^n$ , with two orders of magnitude.

Note that other candidate fields could be found among vector fields  $\vec{u}$ . In this case, we could not use directly  $\nabla \vec{u} \otimes \nabla \vec{u}$ , since it would lead to a fourth order tensor. However, we could use

$$(\nabla \vec{u})^\top \nabla \vec{u} \quad (5.7)$$

$$\text{or } \epsilon(\vec{u}) = \frac{1}{2} \left( \nabla \vec{u} + (\nabla \vec{u})^\top \right) \quad (5.8)$$

which has not been tested yet, but further investigations should be performed in this way.

### 5.1.2.2 Application

Back to the figure 5.1's case, with scalar field  $u = \bar{\gamma}$  (the acceleration norm), if we employ the additional term  $\mu^2 \nabla \bar{\gamma} \otimes \nabla \bar{\gamma}$  alone (without natural metric), then the mesh obtained after adaptation is fine enough in the crossing zone, but not fine enough where acceleration is almost homogeneous (figure 5.6).

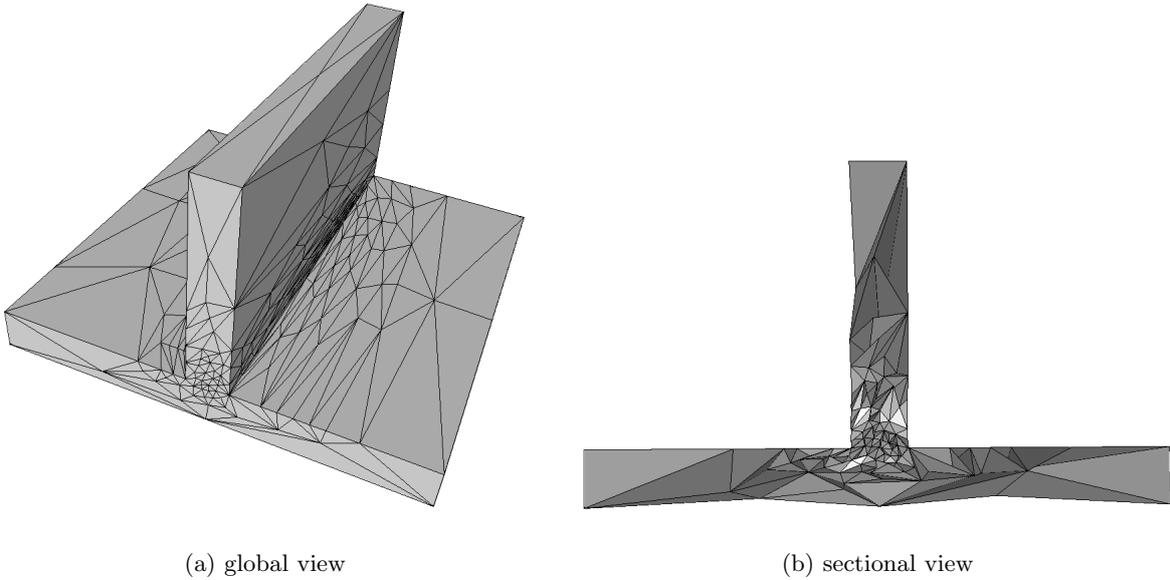
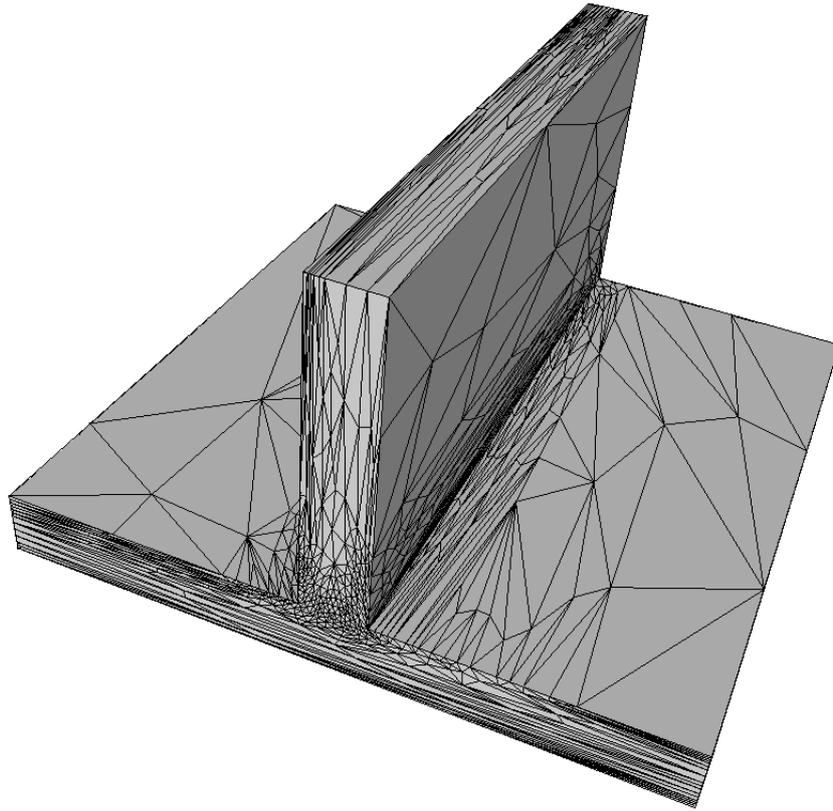


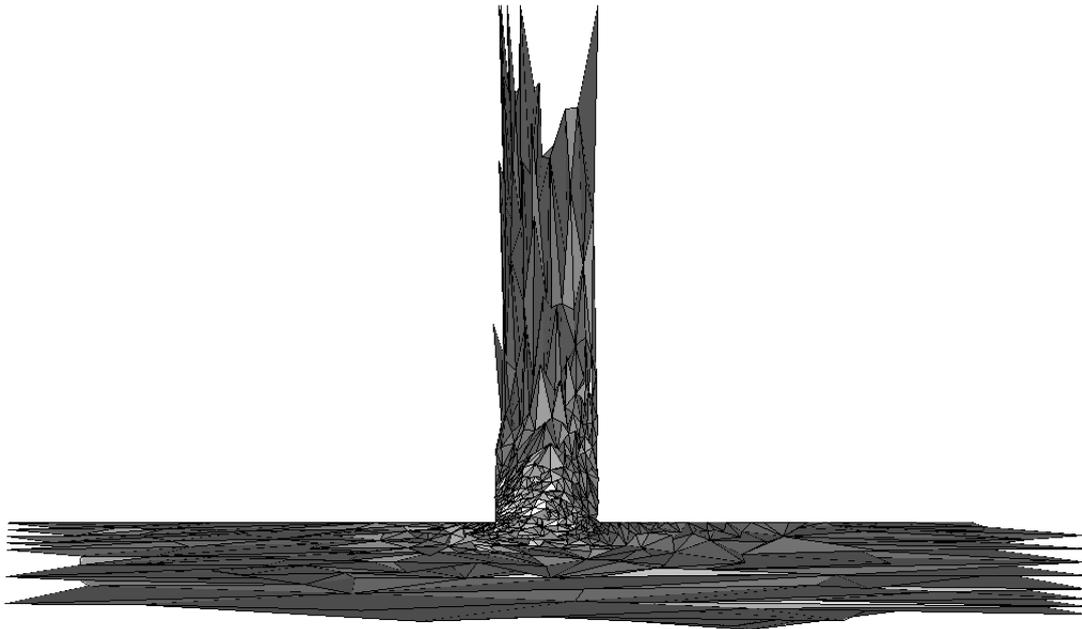
Figure 5.6: Mesh obtained with the acceleration heterogeneity (920 nodes)

We remark that, in this case, the anisotropic adaptation is driven by a matrix field that is not a metric field, since 0 is always an eigenvalue of  $\nabla \bar{\gamma} \otimes \nabla \bar{\gamma}$  (section A.1.2 page 147). However, when the topological algorithm is performed, an average between nodal matrices is computed, which leads to a metric when  $d$  independent vectors  $\nabla \bar{\gamma}$  are involved (eigenvalue 0 disappears from the averaged matrix). At worst, a degenerated matrix can lead to an edge with a zero length, which is not crippling for the process.

However, when  $\mu^2 \nabla \bar{\gamma} \otimes \nabla \bar{\gamma}$  is used in conjunction with the natural metric, as in (5.6), we obtain a metric field. Furthermore, the natural mesh is automatically refined in the crossing zone and remains suitable where the acceleration heterogeneity is useless (figure 5.7).



(a) global view



(b) sectional view

Figure 5.7: Mesh obtained with a natural metric correction by acceleration heterogeneity (5 686 nodes)

## 5.2 Error indicator on anisotropic meshes

Another way to improve the natural metric consists in mesh refinement or coarsening (according to solver needs), without changing anisotropy directions. A multiplicative correction factor can be introduced, which modifies the local mesh sizes in the natural metric field, in order to smooth the numerical error generated by the solver on the current mesh (section 5.2.1).

So as to determine whether the solver needs a finer mesh or a coarser one, one can use:

- an error indicator, even without an accurate error quantification (section 5.2.1)
- or an error estimator, more precise in error quantification, but also harder to compute on an anisotropic mesh (section 5.2.2).

### 5.2.1 Metric corrected by an indicator uniformity

Let  $\alpha_T$  be an error indicator on a mesh element  $T \in \mathcal{T}$ , and  $M$  be a metric to be optimized locally, so as to smooth the values of  $\alpha_T$  over the computation domain  $\Omega$ .

The first step consists in computing the optimal mesh size  $h_T^{\text{optimal}}$  of each element  $T$  that leads to an uniformization (and maybe a reduction) of the indicated error. Then, then correction factor, obtained for  $h_T^{\text{optimal}}$ , can be used with the inverse transformation matrix  $M^{-1/2}$  (its eigenvalues are directly the mesh sizes), so as to get a correction factor for the metric field  $M$ .

#### 5.2.1.1 Computation of the local correction factor

We denote by  $\alpha_{\text{imposed}}$  the error level desired for the indicator  $\alpha_T$ . Let  $p$  be the convergence order of the finite element method and  $d$  be the spatial dimension. Then, mesh adaption theory [Boussetta 2005] leads to

$$h_T^{\text{optimal}} = \frac{\alpha_{\text{imposed}}^{\frac{1}{p}}}{\alpha_T^{\frac{2}{2p+d}} \left( \sum_{T \in \mathcal{T}} \alpha_T^{\frac{2d}{2p+d}} \right)^{\frac{1}{2p}}} h_T \quad (5.9)$$

In fact,  $\alpha_{\text{imposed}}$  is computed from  $\text{card}(\mathcal{T})_{\text{imposed}}$ , which is the imposed number of elements for the next adapted mesh, via

$$\alpha_{\text{imposed}} = \frac{\left( \sum_{T \in \mathcal{T}} \alpha_T^{\frac{2d}{2p+d}} \right)^{\frac{2p+d}{2d}}}{\text{card}(\mathcal{T})_{\text{imposed}}^{\frac{p}{d}}} \quad (5.10)$$

then, (5.9) can be rewritten as

$$h_T^{\text{optimal}} = \frac{\left( \sum_{T \in \mathcal{T}} \alpha_T^{\frac{2d}{2p+d}} \right)^{\frac{2p+d}{2p} - \frac{1}{2p}}}{\text{card}(\mathcal{T})_{\text{imposed}}^{\frac{1}{d}} \alpha_T^{\frac{2}{2p+d}}} h_T \quad (5.11)$$

$$= \frac{\left( \frac{1}{\text{card}(\mathcal{T})_{\text{imposed}}} \sum_{T \in \mathcal{T}} \alpha_T^{\frac{2d}{2p+d}} \right)^{\frac{1}{d}}}{\alpha_T^{\frac{2}{2p+d}}} h_T \quad (5.12)$$

Now, if we change our indicator  $\alpha_T$  into  $\alpha_T^{\frac{2}{2p+d}}$ , we have

$$h_T^{\text{optimal}} = \frac{\left( \frac{1}{\text{card}(\mathcal{T})_{\text{imposed}}} \sum_{T \in \mathcal{T}} \alpha_T^d \right)^{\frac{1}{d}}}{\alpha_T} h_T \quad (5.13)$$

then, introducing  $\text{card}(\mathcal{T})$ , which is the number of elements of the current mesh, it comes

$$h_T^{\text{optimal}} = \beta \frac{\left( \frac{1}{\text{card}(\mathcal{T})} \sum_{T \in \mathcal{T}} \alpha_T^d \right)^{\frac{1}{d}}}{\alpha_T} h_T \quad (5.14)$$

where  $\beta = \left( \text{card}(\mathcal{T}) / \text{card}(\mathcal{T})_{\text{imposed}} \right)^{1/d}$  is the desired error reduction.

We can now introduce the global error on the topology  $\mathcal{T}$ , defined by

$$\alpha_{\mathcal{T}} = \frac{1}{\text{card}(\mathcal{T})} \left( \sum_{T \in \mathcal{T}} \alpha_T^d \right)^{\frac{1}{d}} \quad (5.15)$$

thus,

$$h_T^{\text{optimal}} = \beta \frac{\alpha_{\mathcal{T}}}{\alpha_T} h_T \quad (5.16)$$

Since we want to improve a metric field  $M$ , defined on each mesh node  $S \in \mathcal{N}$ , this correction factor (5.16) is averaged around each node

$$\frac{1}{\sum_{T \in \mathcal{T}(S)} |T|} \sum_{T \in \mathcal{T}(S)} \left( \beta \frac{\alpha_{\mathcal{T}}}{\alpha_T} \right) = \beta \frac{\alpha_{\mathcal{T}}}{\alpha_S} \quad (5.17)$$

$$\text{with } \alpha_S^{-1} = \frac{\sum_{T \in \mathcal{T}(S)} |T| \alpha_T^{-1}}{\sum_{T \in \mathcal{T}(S)} |T|} \quad (5.18)$$

where  $\mathcal{T}(S)$  is the set of elements from  $\mathcal{T}$ , whose  $S$  is a vertex.

Since  $M^{-1/2}$  has the same unit than  $h_T$ , we apply the averaged correction factor with (5.17) and it comes

$$M_{\text{optimal}}^{-1/2}(S) = \beta \frac{\alpha_{\mathcal{T}}}{\alpha_S} M^{-1/2}(S) \quad (5.19)$$

Finally, the correction for the metric field itself is

$$M_{\text{optimal}}(S) = \left( \beta \frac{\alpha_{\mathcal{T}}}{\alpha_S} \right)^{-2} M(S) \quad (5.20)$$

which can be seen as

$$M_{\text{after}} = \left( \beta \frac{\text{global error}}{\text{local error}} \right)^{-2} M_{\text{before}} \quad (5.21)$$

### 5.2.1.2 A possible strategy

A question arises of how the coefficient  $\beta$  can be determined. In practice, an end user can choose between:

- smoothing the error and preserving the same averaged error level, then  $\beta = 1$  and  $\text{card}(\mathcal{T})_{\text{imposed}} = \text{card}(\mathcal{T})$
- smoothing and reducing the averaged error level with a factor 2 (for instance), then it comes  $\beta = 1/2$  and in 3D,  $\text{card}(\mathcal{T})_{\text{imposed}} = 8 \text{card}(\mathcal{T})$
- smoothing and reducing the averaged error level until a given number of elements is attained; in this case,  $\beta = \left( \text{card}(\mathcal{T}) / \text{card}(\mathcal{T})_{\text{imposed}} \right)^{1/d}$
- or smoothing and rising the averaged error level, which corresponds to  $\beta > 1$ .

A low cost strategy could consists in smoothing and not reducing the error, with an error indicator based on the normalized velocity gradient

$$\alpha_T = \left\| \left( \nabla \frac{v}{\|v\|} \right) (T) \right\| \quad (5.22)$$

In this case, when the velocity direction changes, the mesh becomes more refined and the more the change is sudden, the more the mesh is refined. This approach has not been tested yet. Nonetheless, it is an extension of the natural metric that worths mentioning.

### 5.2.2 Error estimation with the mesh anisotropy taken into account

A second mean to build an error indicator dedicated to smoothing, is to derive an error estimator. Among all numerical errors that could be estimated, we focus our attention on the error generated by equation solving. For instance, the residual error estimator for a Newtonian Stokes problem (5.1) is of a great interest.

We consider the following indicator based on the residual error

$$\alpha_{\mathcal{T}} = \left( \sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} \quad (5.23)$$

$$\text{with } \eta_T = \sum_{F \in \partial T} \lambda_T^F \left\| 2 \eta [\epsilon(v)]_T^F \cdot \vec{n}(F) \right\|_{L_2(F)}^2 h_F + \left\| \nabla p \right\|_{L_2(T)}^2 h_T^2 \quad (5.24)$$

where  $\lambda_T^F$  is one half, except when  $F$  is a boundary face (in this case,  $\lambda_T^F$  is zero). In (5.24),  $[\epsilon(v)]_T^F$  is the strain rate jump across a face  $F$  from the element  $T$ . Furthermore,  $h_F$  is the diameter of  $F$  and  $h_T$  the diameter of  $T$ .

It has been proved in [Coupez and Bigot 2000, Bigot 2001] that (5.24) is an error estimator for the Stokes problem. However, this estimator does not work on anisotropic meshes, since it relies on the Clément lemma, which is only valid for isotropic meshes. In practice, a 1D Stokes problem (a planar Poiseuille) can be solved with an increasing accuracy (so, a decreasing exact error) on an anisotropic mesh series, built with the same element diameter (so that estimation remains constant), by a unidirectional refinement.

This limitation is avoided by modifying the Clément lemma with a matching function [Kunert 1998, Kunert 1999, Kunert 2001], also know as an alignment measure [Creusé *et al.* 2003], since it measures

the alignment between the mesh anisotropy and the error heterogeneity. Denoting the error by  $e$ , this matching function  $m_1$  is defined by

$$m_1(e, \mathcal{T}) = \frac{\left( \sum_{T \in \mathcal{T}} h_{\min}^{-2}(T) \left\| C_T^\top \nabla e \right\|_{L_2(T)}^2 \right)^{1/2}}{\|\nabla e\|_{L_2(T)}} \quad (5.25)$$

where  $h_{\min}(T)$  is the minimal height of element  $T$  and  $C_T^\top$  is the matrix that transforms  $T$  in a unitary equilateral element. This scalar function is used as a multiplicative factor for the upper bound of a residual error estimator. Its value is 1 when the alignment is perfect and strictly greater than 1, otherwise.

Unfortunately, computing  $m_1(e, \mathcal{T})$  involves the estimation of error  $e$ . Authors advise the use of a recovering method or the use of super convergence points, which is beyond the scope of this work.

Here we propose a different approach, based on the following remark inspired by the previous work of [Coupez and Bigot 2000, Bigot 2001]: with the planar Poiseuille problem, where the flow is orthogonal to the  $y$  direction, when diameters are replaced by mesh sizes in direction  $y$  in (5.24), then a relevant error estimation is recovered.

That is the reason why diameters  $h_F$  and  $h_T$  should be replaced by  $h(F, \vec{d})$  and  $h(T, \vec{d})$ , which are the sizes of  $F$  and  $T$ , computed in a specific direction  $\vec{d}$ . This specific direction could be the direction of a gradient (computed from a scalar field, like the velocity norm  $\|v\|$ ), which would lead to an automatic selection of direction  $y$  in the planar Poiseuille example.

Computing  $h(T, \vec{d})$  can be efficient, if we use the real symmetrical matrix  $N_T$ , with  $d$  columns and  $d$  rows, solution of the following linear system

$$\left( \|S_j - S_i\|_{N_T} = \|S_j - S_i\|^2 \right)_{1 \leq i < j \leq d} \quad (5.26)$$

where  $S_0, \dots, S_d$  are the vertices of element  $T$ . In this case, when  $\vec{d}$  is unitary, we have

$$h(T, \vec{d}) = \|\vec{d}\|_{N_T} \quad (5.27)$$

In what concerns  $h(F, \vec{d})$ , it is more complicated, since it involves the computation of  $N_F$ , the same matrix associated to a face  $F$ , but in the hyperplane containing  $F$ .

With these considerations, we obtain the following error indicator

$$\alpha_T = \sum_{F \in \partial T} \lambda_T^F \left\| 2 \eta [\epsilon(v)]_T^F \cdot \vec{n}(F) \right\|_{L_2(F)}^2 \|\vec{d}\|_{N_F} + \|\nabla p\|_{L_2(T)}^2 \|\vec{d}\|_{N_T}^2 \quad (5.28)$$

$$\text{with } \vec{d} = \frac{\nabla \|v\|}{\|\nabla \|v\|\|} \quad (5.29)$$

This is not an error estimator, because an upper bound in direction  $\vec{d}$  may not always be a global upper bound in all directions. However, it would be very interesting to use this new indicator in the smoothing process, described previously.

These thoughts about an a posteriori metric correction give some further investigation orientations for future works.

## **Part III**

# **Multidomain metric**



## Introduction

At this stage, a suitable mesh for a monodomain Rem3D simulation can be built. Monodomain means that the simulation involves only one body (for instance, the mold cavity in the injection process). However, a simulation can involve several bodies at the same time. In this case, the computational domain  $\Omega$  is a compound of  $p$  subdomains  $\omega_i$ , such that

$$\Omega = \omega_1 \cup \dots \cup \omega_p \quad (5.30)$$

In this document, a subdomain is not a partition generated by a domain decomposition for parallel computing like in [Löhner *et al.* 1999].

Among all possible multidomain simulation topics, we emphasize the injection process with mold coupling [Batkam and Coupez 2001] and other material forming processes, like forging with deformable tools [Barboza *et al.* 2002]. Other multidomain simulations can be motivated by a computation at mesoscopic or microscopic scale, like: foam expansion like in [Bruchon and Coupez 2003], over-molding like in [Coupez *et al.* 2004] or fiber reinforced molding [Bouchard *et al.* 2000, Fournier 2003, Megally *et al.* 2004].

In what concerns the injection process with mold coupling, the following subdomains are taken into account:

- the mold cavity where the fluid (in red on figure 5.8) flows and where the air is trapped
- the mold itself, maybe divided in several parts (in blue, green and yellow on figure 5.8)
- some thermal regulation channels (appearing by transparency on figure 5.8), where a cooling fluid flows.

To take the mold into account is very important for a heat transfer study [Batkam 2002], because thermal boundary conditions are only well known at the exterior of the mold (not between the cavity and the mold).

When a mesh involves several bodies in interaction (chapter 6 page 99), the natural metric can be enriched by a surface term, dedicated to interface tracking between subdomains. Thanks to some Riemannian geometry results, we obtain the multidomain metric that can be used in conjunction with a discontinuous  $P^0$  interpolation of each subdomain characteristic function (VoF, Volume of Fluid).

This multidomain metric can then be applied to discontinuity surface tracking in many meshing situations (chapter 7 page 115).

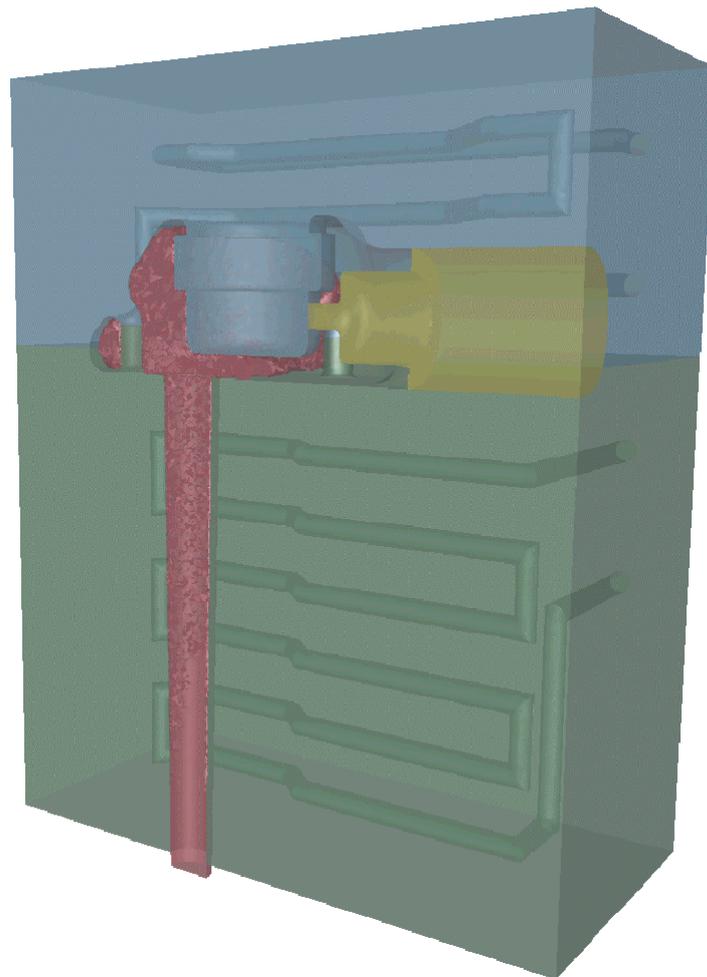


Figure 5.8: A multidomain configuration: mold coupling with thermal regulation (courtesy of Sophysa and Transvalor)

## Chapter 6

# Multidomain metric generated a priori

So as to treat a computational domain  $\Omega = \omega_1 \cup \dots \cup \omega_p$ , one can proceed by three different meshing strategies:

- to generate one mesh for each subdomain, independently, which leads to  $p$  meshes that are non conforming (figure 6.1(a))
- to generate one (conforming) mesh whose internal faces give a precise triangulation of interfaces between subdomains (figure 6.1(b))
- to generate one (conforming) mesh where interfaces between subdomains are (fuzzily) immersed (figure 6.1(c)); in this case, an element of the mesh can be filled with two distinct subdomains (or more).

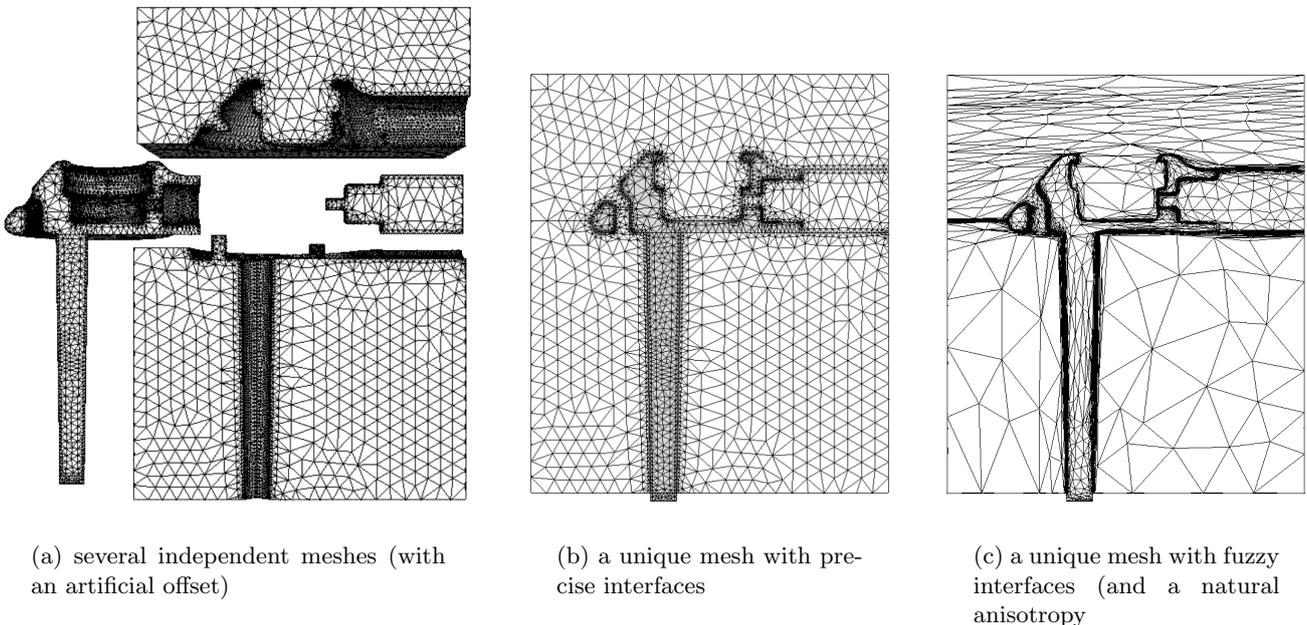


Figure 6.1: Three meshing strategies for a multidomain simulation

The first approach is widely used in multi-body forging [Barboza *et al.* 2002] or in aero-elasticity [Lesoinne and Farhat 1996]. From a meshing point of view, this is the most simple approach since meshing  $p$  bodies independently is straightforward. However, its main drawback is the contact treatment, because information has to be transported from one mesh to another (so that interactions between subdomains are correctly taken into account). The use of contact elements [Barboza *et al.* 2002], mortar

elements [Ben Belgacem *et al.* 1995] or other techniques like [Maman and Farhat 1995], leads to many geometrical localization during the computation.

The second approach, used for instance by [Jacquotte 1989], has been initially chosen for Rem3D so as to avoid those expensive localizations [Batkam 2002]. Indeed, with one (conforming) mesh for the whole domain, sub-meshes corresponding to subdomains are perfectly connected. However, in practice, this approach involves several complex meshing operations:

- firstly, a mesh of the first subdomain is generated (from a first mesh of its boundary)
- secondly, the boundary of this mesh is extracted and merged with the mesh of the second subdomain boundary
- thirdly, a mesh of the second subdomain is generated without modifying the common boundary mesh with the first subdomain
- then, these operations are iterated, so as to treat all subdomains
- all sub-meshes are merged in a unique global mesh (this step is the less difficult one).

This approach implies three major drawbacks:

- the end user has a lot of manual work to do, with an exponential growth when the subdomain number rises
- CAD operations on each subdomain are constrained by the presence of other subdomains
- meshing (and remeshing) operations are also strongly constrained by the internal interface tracking.

Currently, because of these difficulties, the multidomain capabilities of Rem3D are not widely used.

For that purpose, we propose a third approach that consists in generating a single mesh, which captures interfaces between subdomains. Such an interface capture is automatically performed by a mesh refining in the vicinity of subdomain boundaries. The end user has no manual operation, however, some new drawbacks arise:

- the refinement, even with a relevant anisotropy, introduces more nodes than other approaches (computation is then more expensive, but also more accurate)
- interfaces are fuzzy, which requires that the solver<sup>1</sup> can deal with mixed elements (a mixing between several subdomains).

In this chapter, before this third approach is described (section 6.2), section 6.1 shows how to use a natural metric inside each subdomain (in the framework of both second and third approaches).

---

<sup>1</sup>fortunately, this is the case with Rem3D solvers

## 6.1 Natural sub-mesh

A natural mesh for the computational domain  $\Omega$  is no more interesting in the multidomain framework. In fact, a mesh is required with several element layers through the thickness of each subdomain. In the subdomain  $\omega_1$ , a natural metric for  $\omega_1$  needs to be computed, in the subdomain  $\omega_2$ , it is a natural metric for  $\omega_2$  that is required and so on.

### 6.1.1 Exact interfaces

If each subdomain  $\omega$  is triangulated by a sub-mesh of  $\Omega$ 's mesh (the second approach, with exact interfaces), then this sub-mesh could be extracted and a natural metric can be computed like in the previous part. In practice, such an extraction is not performed, but the natural metric of each subdomain is computed directly on  $\Omega$ 's mesh, by introducing minor modifications to chapter 4:

- if the node  $S \in \omega$ , then the natural metric computed in  $S$  is the natural metric of  $\omega$  (when  $S$  belongs to several subdomains, an averaged metric is considered)
- in order to get a metric that is a natural one of  $\omega$ , nodes not belonging to  $\omega$  are not taken into account during the neighborhood  $\mathcal{N}_k(S)$  growing
- boundary nodes and boundary faces of interest are not those of  $\Omega$  but those of  $\omega$ .

These modifications work well with a multidomain mesh, whose interfaces are exact (second approach). It is the case on figure 6.2, where a tube is twisted in a cube geometry, each subdomain (inside and outside the tube) being triangulated with a natural metric.

In this second approach, nodes lying on an interface are often locked during the adaption process, so as to ensure conforming interfaces. This constraint is removed when fuzzy interfaces are considered (the third approach, next section).

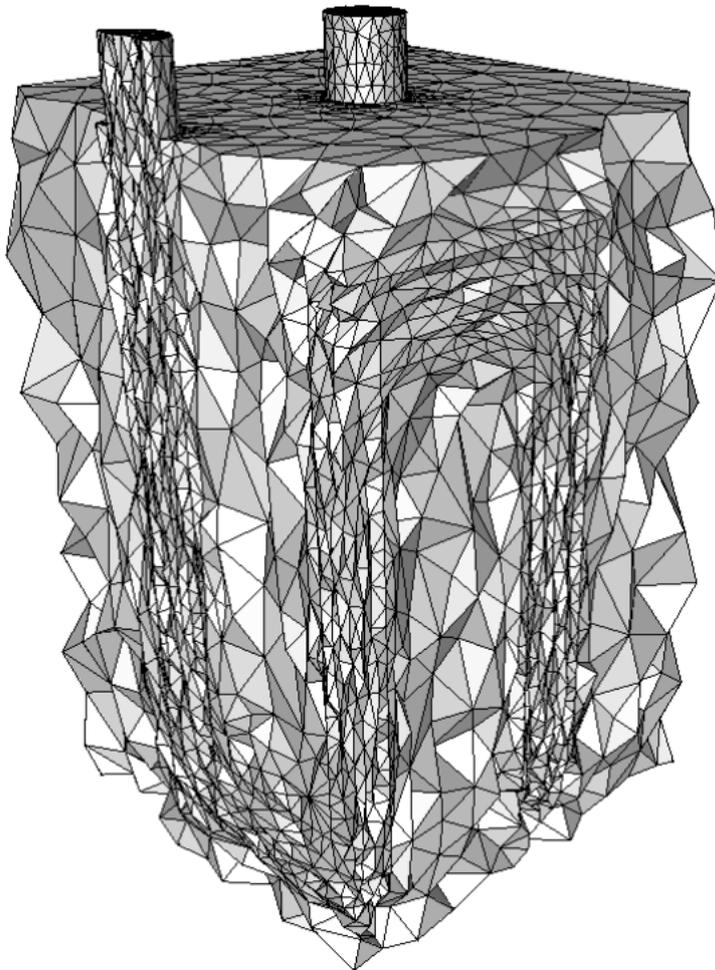


Figure 6.2: A bidomain mesh (18 714 nodes) with an exact interface (sectional view)

### 6.1.2 Fuzzy interfaces

When interfaces are fuzzily interpolated, to determine boundary nodes and boundary faces of a subdomain  $\omega$  is more complicated. Some technical adjustments (described in appendix B.2 page 169) are required, but the computation of a natural metric field  $M_{\text{natural}}$  on  $\Omega$  that is locally a natural metric for each subdomain  $\omega_1, \dots, \omega_p$  is still possible.

Again, mesh sizes of this metric field can be divided, so as to obtain  $n$  element layers through the thickness in each subdomain. The notation  $M_{\text{natural}}^n$  is still used, despite the fact that  $n$  can be different from one subdomain to another.

At this stage, thanks to  $M_{\text{natural}}^n$ , the mesh of  $\Omega$  is adapted specifically in each subdomain, but the mesh is not fine enough in the vicinity of interfaces. For that purpose, a surface term has to be added (next section).

## 6.2 Discontinuity surface capturing

Each subdomain can be represented by its characteristic function, whose discontinuities are located at the subdomain boundary. In 3D, interfaces between subdomains are internal discontinuity surfaces of  $\Omega$ . For some interpolation reasons, the mesh needs to be refined in the vicinity of such discontinuity surfaces.

### 6.2.1 Interface tensor

Let us first define the characteristic function of a subdomain.

**Definition 6.1** the *characteristic function* of a subdomain  $\omega$  is the function  $\mathbb{1}_\omega : \Omega \rightarrow \{0, 1\}$  such that

$$\mathbb{1}_\omega(x) = 1 \quad \Leftrightarrow \quad x \in \omega \quad (6.1)$$

Our multidomain metric computation relies on the evaluation of a characteristic function gradient. Since the gradient of  $\mathbb{1}_\omega$  is zero almost everywhere, we consider a regular approximation of the characteristic function. For instance, we could consider the following sigmoid

$$f_\omega(x) = \frac{1}{1 + e^{\beta d(x, \partial\omega)}} \quad (6.2)$$

where  $\beta \gg 1$  and  $d(x, \partial\omega)$  is the signed distance between  $x \in \Omega$  and the boundary  $\partial\omega$  (more precisely,  $d(x, \partial\omega) \leq 0$  when  $x \in \omega$  et  $d(x, \partial\omega) > 0$  when  $x \notin \omega$ ). For a circular subdomain  $\omega$ , with center  $c = (0, 0)$  and radius  $r = 1$ , this sigmoid is plotted on figure 6.3.

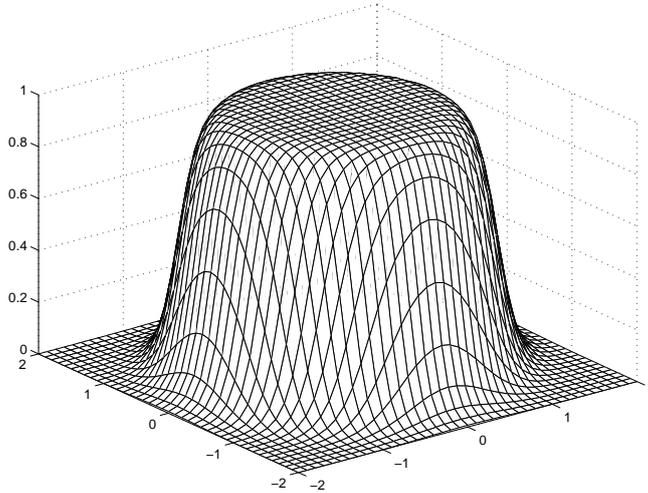


Figure 6.3: Function  $f_\omega$  with  $d(x, \partial\omega) = \|x - c\|_2 - r$

The graph of  $f_\omega$ , denoted by

$$F_\omega = \left\{ x \in \mathbb{R}^{d+1} \text{ such that } (x_1, \dots, x_d) \in \Omega \text{ and } x_{d+1} = f_\omega(x_1, \dots, x_d) \right\} \quad (6.3)$$

is a variety of  $\mathbb{R}^{d+1}$  and its dimension is  $d$ . Basically, our method consists in triangulating this variety with a natural metric in what concerns the  $d$  first cartesian directions and with a mesh size of  $1/m$  in the last direction. After orthogonal projection on  $\Omega$ , we obtain a mesh of  $\Omega$ . This mesh is then in the vicinity of  $\partial\omega$ , with  $m$  element layers in this zone.

In other words, the variety  $F_\omega$  is triangulated with the following metric

$$M(x_1, \dots, x_{d+1}) = \begin{pmatrix} & & & 0 \\ & M_{\text{natural}}^n(x_1, \dots, x_d) & & \vdots \\ & & & 0 \\ 0 & \dots & & 0 \quad m^2 \end{pmatrix} \quad (6.4)$$

This is a metric of  $\mathbb{R}^{d+1}$  that induces (by orthogonal projection) a metric of  $\mathbb{R}^d$  on  $\Omega$  (see the following proposition). In practice,  $F_\omega$  is not triangulated and there is no orthogonal projection on  $\Omega$ , but we use directly this induced metric (it gives the same result).

**Proposition 6.1** the induced metric on  $\Omega$  by orthogonal projection of the metric  $M$  defined by (6.4) on the variety  $F_\omega$  is

$$M_\omega(x_1, \dots, x_d) = M_{\text{natural}}^n(x_1, \dots, x_d) + m^2 \nabla f_\omega(x_1, \dots, x_d) \otimes \nabla f_\omega(x_1, \dots, x_d) \quad (6.5)$$

In other words, the aforementioned technique is equivalent to the addition of tensor  $m^2 \nabla f_\omega \otimes \nabla f_\omega$  to the natural metric.

**Proof :** we denote by  $\pi$  the orthogonal projection on  $\mathbb{R}^d$

$$\pi(x_1, \dots, x_{d+1}) = (x_1, \dots, x_d) \quad (6.6)$$

From Riemannian geometry [do Carmo 1988, Jost 1998], the induced metric of  $M$  on  $\Omega$  (with its map  $\pi$ ) has the following components

$$M_\omega^{ij} = \left( \frac{\partial \pi^{-1}}{\partial x_j} \right)^\top M \frac{\partial \pi^{-1}}{\partial x_i} \quad (6.7)$$

Since on  $F_\omega$  we have

$$\pi^{-1}(x_1, \dots, x_d) = (x_1, \dots, x_d, f_\omega(x_1, \dots, x_d)) \quad (6.8)$$

it comes

$$\frac{\partial \pi^{-1}}{\partial x_i} = \left( \delta_{1i}, \dots, \delta_{di}, \frac{\partial f_\omega}{\partial x_i} \right)^\top \quad (6.9)$$

Thus, we obtain

$$\begin{aligned} M_\omega^{ij} &= \left( \delta_{1j}, \dots, \delta_{dj}, \frac{\partial f_\omega}{\partial x_j} \right) M \left( \delta_{1i}, \dots, \delta_{di}, \frac{\partial f_\omega}{\partial x_i} \right)^\top \\ &= (\delta_{1j}, \dots, \delta_{dj}) M_{\text{natural}}^n (\delta_{1i}, \dots, \delta_{di})^\top + m^2 \frac{\partial f_\omega}{\partial x_j} \frac{\partial f_\omega}{\partial x_i} \\ &= (M_{\text{natural}}^n)^{ij} + m^2 (\nabla f_\omega \otimes \nabla f_\omega)^{ij} \end{aligned}$$

which is exactly the metric (6.5).  $\square$

This analytical result has already been used by [Hoch and Rasclé 2003] in 2D. Chapter 7 is devoted to 3D applications of this results, which is in fact established for all dimensions.

However at this stage, the natural metric computation would involve the expensive evaluation of function  $f_\omega$ . In practice,  $f_\omega$  is replaced by a less expensive function  $g_\omega$  such that  $g_\omega = 1$  inside  $\omega$ ,  $g_\omega = 0$  outside and  $0 < g_\omega < 1$  in the vicinity of  $\partial\omega$  (following section).

### 6.2.2 Discontinuous $P^0$ interpolation and voxelization

We consider a discontinuous  $P^0$  interpolation of the characteristic function  $\mathbb{1}_\omega$ , via the function  $g_\omega : \Omega \rightarrow [0, 1]$  defined almost everywhere by

$$\forall T \in \mathcal{T} \quad \forall x \in \overset{\circ}{T} \quad g_\omega(x) = g_\omega(T) = \frac{|T \cap \omega|}{|T|} \quad (6.10)$$

where  $|\cdot|$  is the Lebesgue measure ( $\omega$  is assumed to be measurable). This function is used in the multidomain metric computation, for some reasons related to the following proposition.

**Proposition 6.2** approximation errors are exactly

$$\int_{\Omega} (g_\omega - \mathbb{1}_\omega) \, dx = 0 \quad (6.11)$$

$$\|g_\omega - \mathbb{1}_\omega\|_{L_2(\Omega)} = \left( \sum_{T \in \mathcal{T}} g_\omega(T) (1 - g_\omega(T)) |T| \right)^{1/2} \quad (6.12)$$

In other words,  $g_\omega$  is a conservative approximation of  $\mathbb{1}_\omega$  (6.11) and diffusion is low, when partially filled element  $T$  (those such that  $0 < g_\omega(T) < 1$ ) are small (6.12).

**Proof :** we have

$$\int_{\Omega} (g_\omega - \mathbb{1}_\omega) \, dx = \sum_{T \in \mathcal{T}} \int_T (g_\omega - \mathbb{1}_\omega) \, dx \quad (6.13)$$

$$= \sum_{T \in \mathcal{T}} \left( \frac{|T \cap \omega|}{|T|} |T| - |T \cap \omega| \right) \quad (6.14)$$

$$= 0 \quad (6.15)$$

which proves (6.11). Furthermore, we have

$$\|g_\omega - \mathbb{1}_\omega\|_{L_2(\Omega)} = \left( \int_{\Omega} (g_\omega - \mathbb{1}_\omega)^2 \, dx \right)^{1/2} \quad (6.16)$$

$$= \left( \sum_{T \in \mathcal{T}} \int_T (g_\omega - \mathbb{1}_\omega)^2 \, dx \right)^{1/2} \quad (6.17)$$

Since  $g_\omega$  is constant on  $T$ , it comes

$$\int_T (g_\omega - \mathbb{1}_\omega)^2 \, dx = g_\omega(T)^2 |T| - 2g_\omega(T) \int_T \mathbb{1}_\omega \, dx + \int_T \mathbb{1}_\omega^2 \, dx \quad (6.18)$$

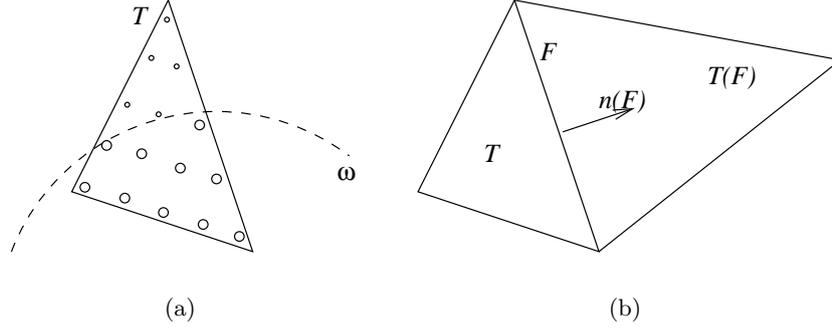
Besides,  $\int_T \mathbb{1}_\omega \, dx = |T \cap \omega|$  and  $\mathbb{1}_\omega^2 = \mathbb{1}_\omega$ , so

$$\int_T (g_\omega - \mathbb{1}_\omega)^2 \, dx = g_\omega(T)^2 |T| - 2g_\omega(T) |T \cap \omega| + |T \cap \omega| \quad (6.19)$$

Using  $|T \cap \omega| = g_\omega(T) |T|$ , we obtain

$$\int_T (g_\omega - \mathbb{1}_\omega)^2 \, dx = g_\omega(T) |T| (g_\omega(T) - 2g_\omega(T) + 1) \quad (6.20)$$

which proves (6.12).  $\square$

Figure 6.4: Element sampling and notations for the  $P^0$  gradient

In practice, the function  $g_\omega$  is computed with an approximated filling ratio of each element. This approximation is performed for an element  $T \in \mathcal{T}$ , by sampling  $T$  with many interior points (such a sampling algorithm is given in appendix B.3.1 page 171). Then, the points that belong to  $\omega$  are counted (figure 6.4(a)).

The estimated filling ratio of  $T \in \mathcal{T}$  is

$$\frac{|T \cap \omega|}{|T|} \simeq \frac{\text{number of points in } \omega}{\text{total number of points}} \quad (6.21)$$

So as to efficiently determine whether a point belongs to  $\omega$  or not, no expensive localization of this point, in a mesh of  $\omega$ , is used. Instead, we prefer use a voxelization of  $\omega$ , because this objet gives the answer in a constant time (appendix B.3.2).

### 6.2.3 Discontinuous $P^0$ gradient and triple point treatment

The function  $g_\omega$  is  $P^0$  discontinuous and a we need to define a numerical gradient of  $g_\omega$  that is not zero almost everywhere. For that purpose, we consider the vectorial  $P^0$  discontinuous function  $\nabla g_\omega : \Omega \rightarrow \mathbb{R}^d$  defined almost everywhere by

$$\forall T \in \mathcal{T} \quad \forall x \in \overset{\circ}{T} \quad \nabla g_\omega(x) = \nabla g_\omega(T) = \sum_{F \in \partial T} \frac{g_\omega(T(F)) - g_\omega(T)}{|T(F)| + |T|} |F| \vec{n}(F) \quad (6.22)$$

where  $\partial T$  contains the faces of  $T$ ,  $T(F)$  is the opposite element of  $T$  through  $F$  (there exists always one opposite element, since virtual elements are considered, making  $\mathcal{T}$  without boundary) and  $\vec{n}(F)$  is the outgoing normal of  $T$  on  $F$  (figure 6.4(b)).

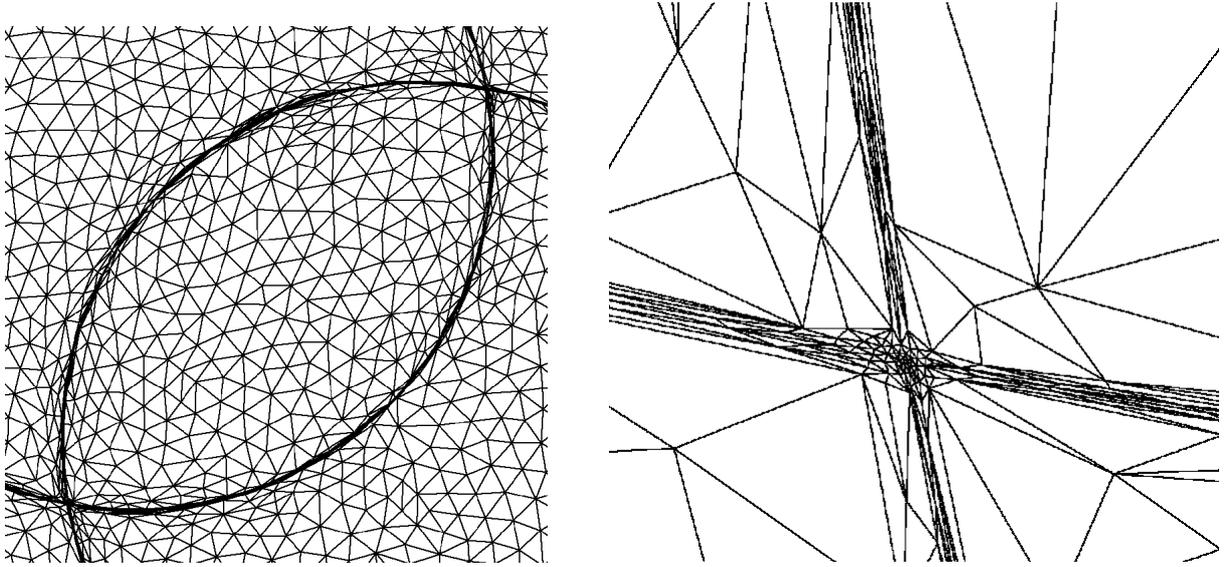
Even with virtual elements, such a  $P^0$  gradient requires boundary conditions. Generally, when  $F$  is a boundary face, we take  $|T(F)| = \infty$ , which is an adiabatic boundary condition. However, if we take  $|T(F)| = 0$  and  $g_\omega(T(F)) \neq g_\omega(T)$ , then a boundary layer is generated in the vicinity of  $\Omega$ 's boundary, which can be sometimes useful.

This gradient definition is motivated by the fact that it is a first order approximation of  $\mathbb{1}_\omega$ 's gradient [Coupez and Baranger]. In fact,  $\nabla g_\omega(T)$  is a linear combination of outgoing normals of  $T$ , each term being weighted by the jump of  $g_\omega$  across the face, by the face measure and by the measures of elements who share this face.

This gradient enables us to apply the previous metric building on  $g_\omega$  (and not on  $f_\omega$ ). However,  $\Omega$  is composed of several subdomains  $\omega_1, \dots, \omega_p$  with respective functions  $g_{\omega_1}, \dots, g_{\omega_p}$ . So, for each component  $1 \leq i \leq d$ , we consider the maximal gradient

$$\nabla g(T)_i = \max(\nabla g_{\omega_1}(T)_i, \dots, \nabla g_{\omega_p}(T)_i) \quad (6.23)$$

which leads, empirically, to a good treatment of triple (or more) points (figure 6.5(b)). Indeed, according



(a) multidomain mesh with triple points

(b) enlargement of the bottom left triple point

Figure 6.5: Two overlapping subdomains (equivalent to three subdomains), involving two triple points to appendix A.1.3.2 page 151, the max operator leads often to an isotropic metric when applied to distinct metrics. In our case, an isotropic mesh is relevant around a triple point.

The final  $P^1$  (continuous) multidomain metric field is obtained by applying formula (6.5) and by regularizing the  $P^0$  discontinuous tensor  $\nabla g \otimes \nabla g$  around each node  $S$ , which gives

$$M_{\text{multidomain}}(S) = M_{\text{natural}}^n(S) + m^2 \nabla g(S) \otimes \nabla g(S) \quad (6.24)$$

$$\text{with } \nabla g(S) = \left( \sum_{T \in \mathcal{T}(S)} |T| \nabla g(T) \right) / \left( \sum_{T \in \mathcal{T}(S)} |T| \right) \quad (6.25)$$

where  $\mathcal{T}(S)$  contains the elements of  $\mathcal{T}$  whose  $S$  is a vertex.

It should be noted that the mesh of figure 6.5(a) has been generated by considering two circular subdomains that overlap themselves. In other words, even if it has no physical meaning, overlapping subdomains can be treated by the method presented here.

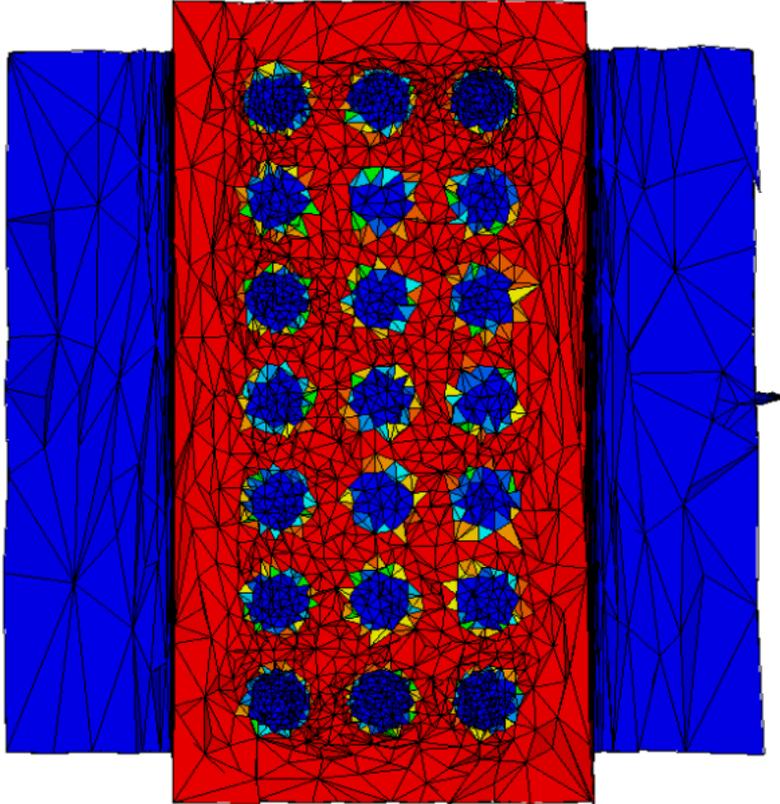
#### 6.2.4 Capillary number and final algorithm

Before the complete algorithm is described, the multiplicative parameter  $m$  of (6.25) has to be determined. Since  $m$  can be interpreted as the desired number of element layers in the vicinity of an interface, its value could be chosen by the end user. However,  $m$  is treated as a numerical parameter whose value is automatically chosen, in order to ensure the interface capturing.

So as to understand the influence of  $m$ , we consider a bidomain over-molding case where a simple part is injected on a pierced reinforcement (figure 6.6). In this configuration, if we choose a constant (and quite low) value for  $m$ , then, the quality of interface capturing depends strongly on the natural mesh around interfaces (figure 6.7).



Figure 6.6: Over-molding case, including a pierced plate

Figure 6.7: Ill interface capturing  
(the scalar field in color scale is  $g_\omega$ : red elements are full, blue ones are empty)

In fact in (6.24),  $m^2$  behaves like a penalty coefficient between the natural metric  $M_{\text{natural}}^n$  and the additional tensor  $\nabla g \otimes \nabla g$ . Empirically, to ensure a good capturing of interfaces between subdomains, the term  $m^2 \nabla g \otimes \nabla g$  needs to be two order of magnitude greater than the natural metric, except where the gradient  $\nabla g$  is zero.

That is the reason why,  $m$  is computed automatically from the following capillary number

$$\frac{\lambda_{\max}(M_{\text{natural}}^n)}{\lambda_{\max}(\nabla g \otimes \nabla g)} \quad (6.26)$$

which is the ratio between the volume term  $M_{\text{natural}}^n$  and the surface term  $\nabla g \otimes \nabla g$  (where  $\lambda_{\max}$  is the greatest eigenvalue operator). Locally, for each node  $S$  we consider

$$m^2(S) = 256 * \frac{\lambda_{\max}(M_{\text{natural}}^n(S))}{\lambda_{\max}(\nabla g(S) \otimes \nabla g(S))} \quad (6.27)$$

From our experience, 256 is the first power of 2 that always leads to a good interface capturing (figure 6.8).

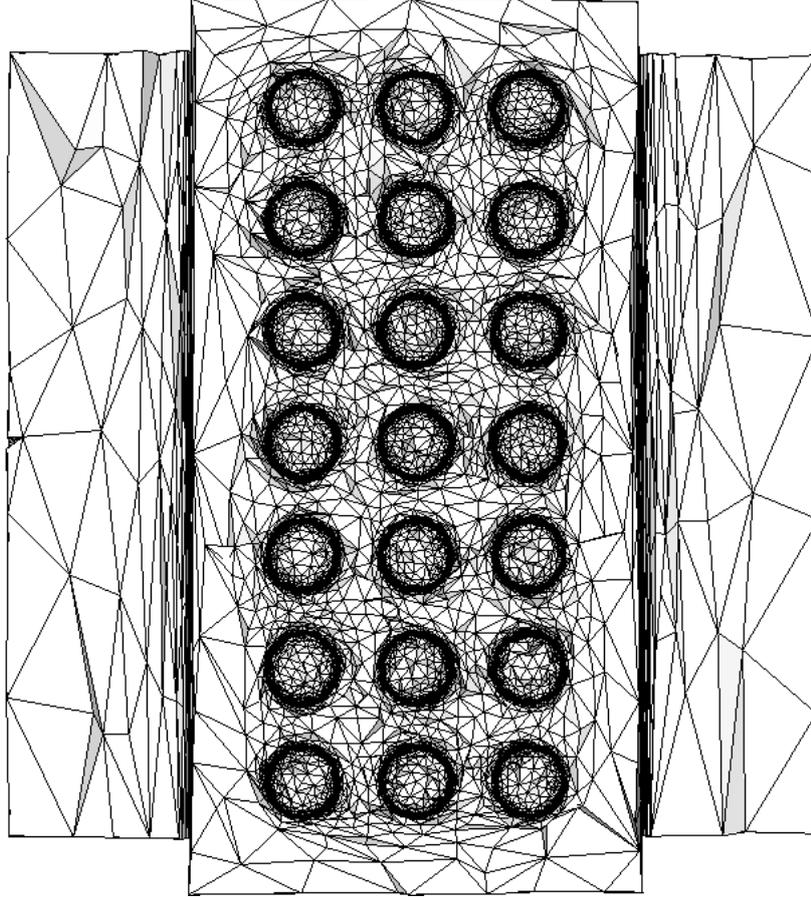


Figure 6.8: Good interface capturing after that coefficient  $m$  has been tuned

At this stage, the computational algorithm for the multidomain metric field  $M_{\text{multidomain}}$  is complete.

**Algorithm 6.1** after voxelization of the subdomains, we proceed as follows:

**for** each subdomain  $\omega$  **do**

- computation of  $\omega$ 's natural metric and assembling in the global natural metric field
- computation of  $g_\omega$ , the  $P^0$  discontinuous interpolation of  $\omega$ 's characteristic function
- computation of  $g_\omega$ 's gradient and assembling in the global gradient  $\nabla g$

**done**

- regularization of the global gradient  $\nabla g$  into a  $P^1$  continuous field

- computation of the tensorial product  $\nabla g \otimes \nabla g$

**for** each node  $S$  of the mesh **do**

- computation of the penalty factor  $m^2(S)$
- addition of the term  $m^2(S)\nabla g(S) \otimes \nabla g(S)$  to the global natural metric field

**done**

Except for the natural metric computation, the algorithmic complexity of these operations is linear with the problem size. The complexity of the multidomain metric computation has the same order than the natural one. Memory consumption is linear with the problem size.

### 6.2.5 Iteration process and convergence criterion

The first mesh generated by adaption to the multidomain metric is never perfect. So, the preceding algorithm is employed in an iterative process between metric generation and mesh generation. In order to illustrate such an iterative process, we consider an industrial bidomain case, where a subdomain  $\omega$  is the cavity of our biomedical test case (voxelization is given on figure B.2 page 172). The mold around the cavity is simply a parallelepiped (figure 6.9(a)).

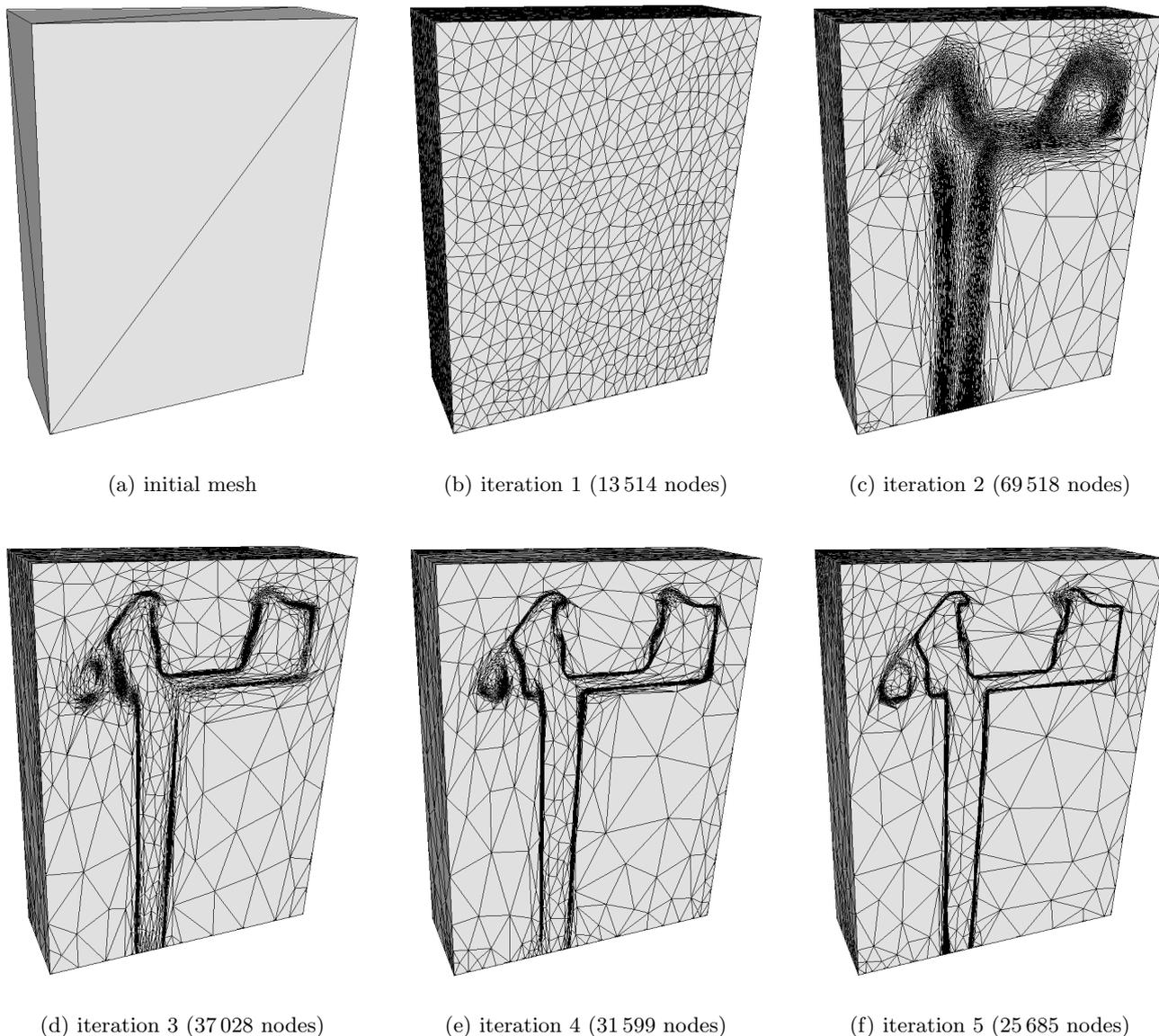


Figure 6.9: Iterative process of the anisotropic adaption to a multidomain metric

Without limitation, this iterative process would indefinitely refine the mesh around interfaces. That is the reason why a minimal mesh size  $h_{\min}$  is prescribed to truncate the multidomain metric (all eigenvalues greater than  $h_{\min}^{-2}$  are corrected). The value of  $h_{\min}$  should be small enough to ensure interface capturing, but still greater than  $\epsilon$ , the voxelization precision (otherwise, the voxel shape appears in the adapted mesh). The use of this minimal mesh size makes the process converge (figure 6.10).

The relative  $L^2$  approximation error, defined by (6.12), measures the diffusion (the fuzzyness) around interfaces. Its stability is a good convergence indicator, in complement with the number of nodes. History

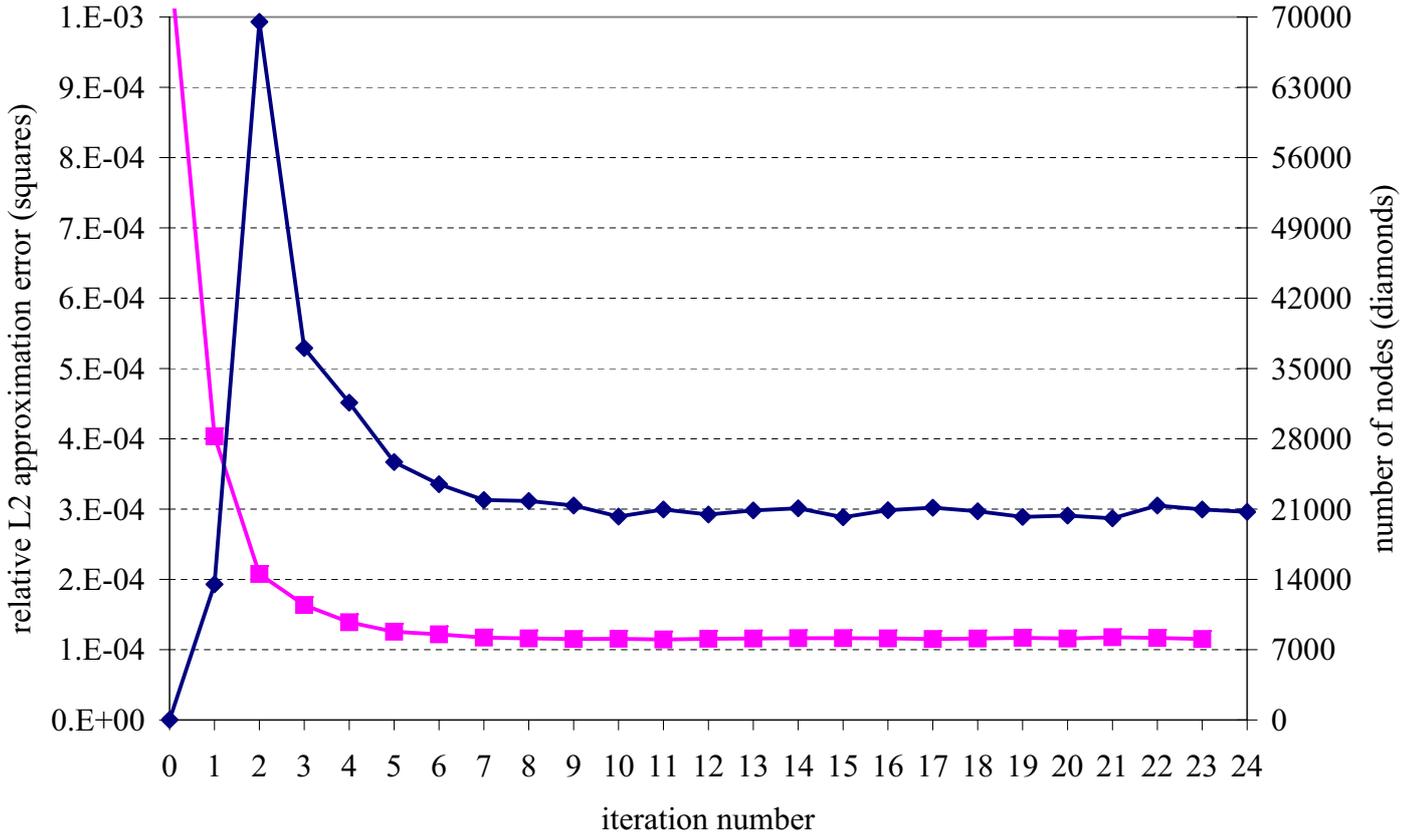


Figure 6.10: Convergence history of the multidomain metric iterative process

against iteration number (figure 6.10) shows that, in this case, the  $L^2$  error indicator converges faster than the number of nodes. The final value of this indicator depends on intrinsic limitations of the method, on  $h_{\min}$  and on interface geometry.

In what concerns this example, the mesh is quite stable after 8 iterations, which is a common value. Remembering that 8 iterations are not enough to obtain a convergence of the natural metric, in the monodomain framework (figure 4.14 page 73). The multidomain framework seems to speed up the natural metric convergence. Especially the number of element layers (figure 6.11), even if the final value is not exactly the required number of element layers. This better convergence is certainly due to the fact that the position of a subdomain boundary is not constrained on a surface but in the diffusion zone (it is a fuzzy interpolation advantage).

A sectional view of the last mesh (24-th iteration) shows that a good adaption is, not only, attained on the symmetry plane (figure 6.12(b)), but also, inside the computational domain (figure 6.12(a)). On this sectional view, the mesh is clearly refined in the vicinity of the interface between  $\omega$  and its complement  $\Omega \setminus \omega$ , with a relevant anisotropy (aligned with this surface curvature). Furthermore, the number of element layers around the interface is about 4 or 5 (figure 6.12(b)). This value is enough for our applications, but it would be hard to increase or to decrease it (since we automatically use the parameter  $m$  as a penalty factor).

The scalar field  $g_\omega$ , which is the discontinuous  $P^0$  approximation of  $\omega$ 's characteristic function, is plotted for the symmetry plane on figure 6.13. It shows that, at most, two elements layers are crossed by the boundary of  $\omega$ . These elements are partially filled, that is why they are neither blue nor red on figure 6.13(b). Since these elements are very thin (figure 6.14(b)), the  $L^2$  interpolation error is low.

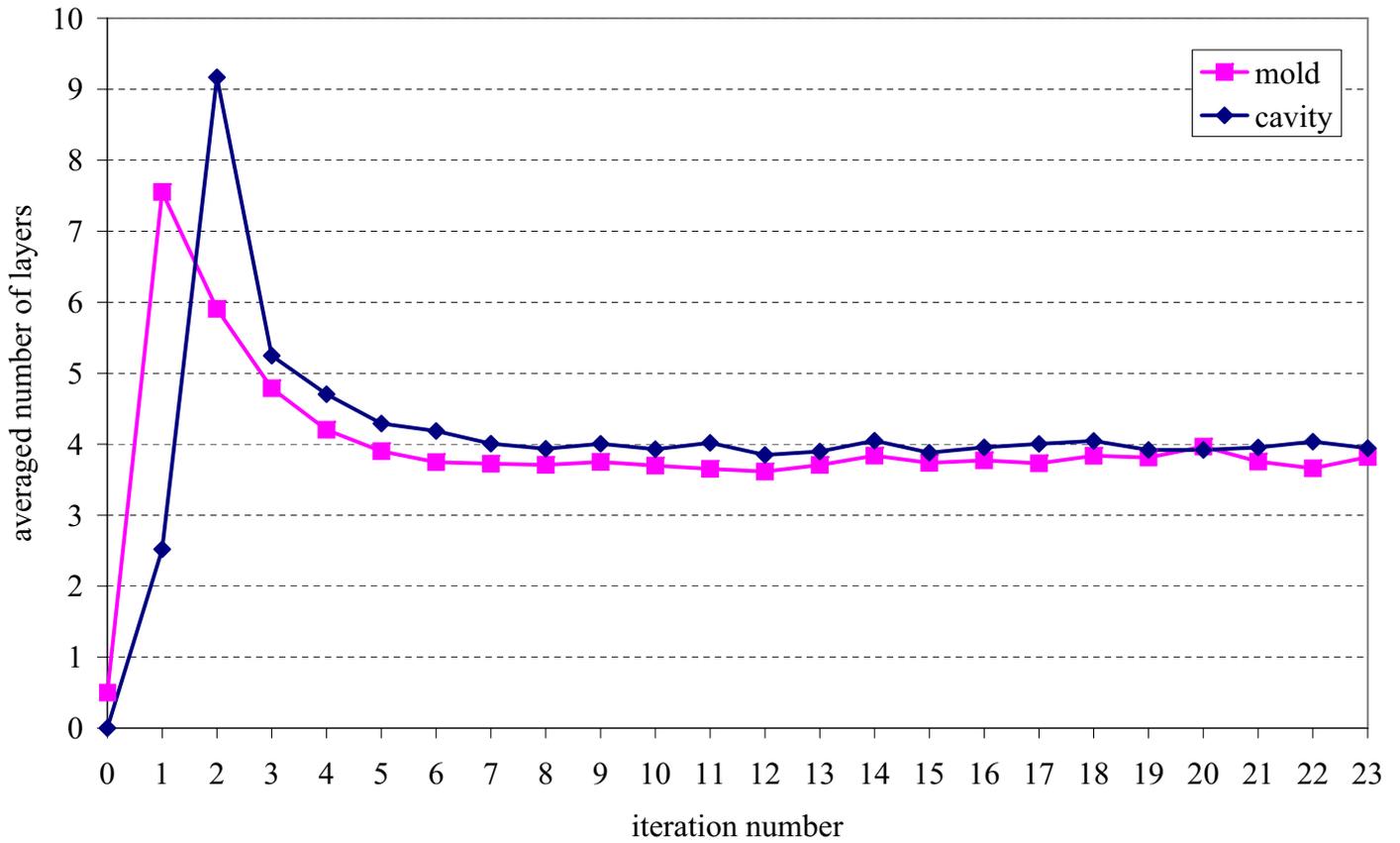
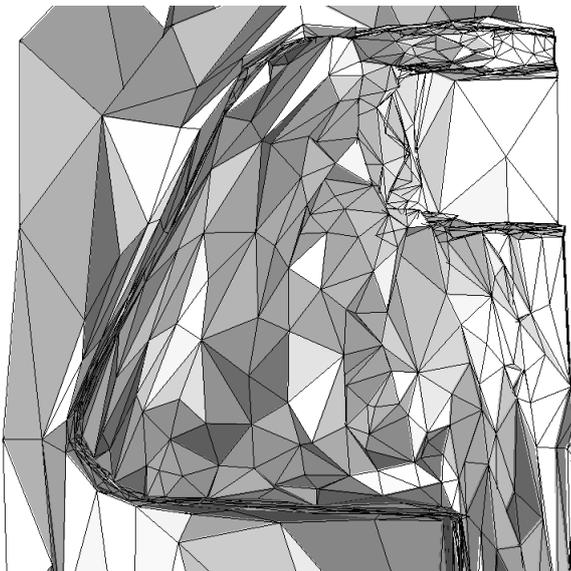
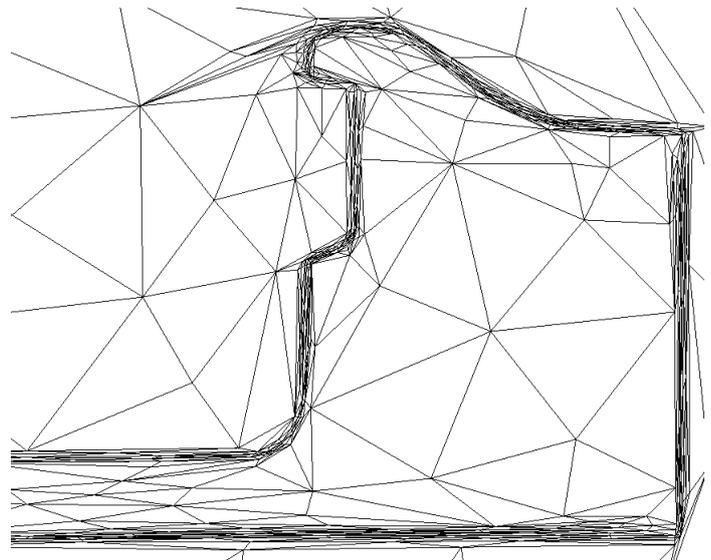


Figure 6.11: Convergence of the number of element layers, for the same case

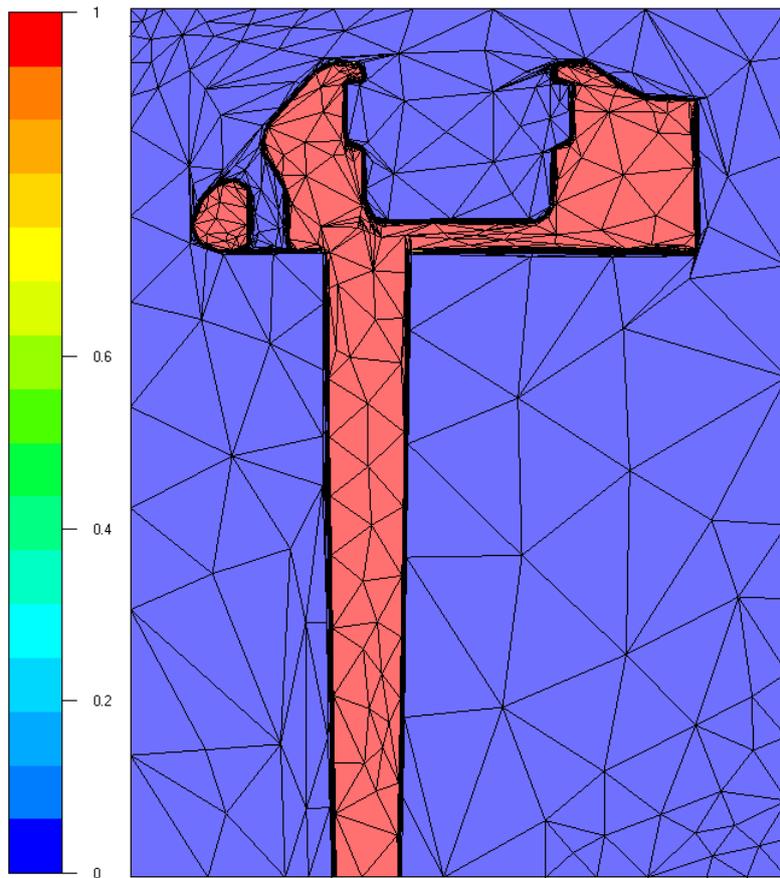


(a) sectional view (orthogonal to the symmetry plane)

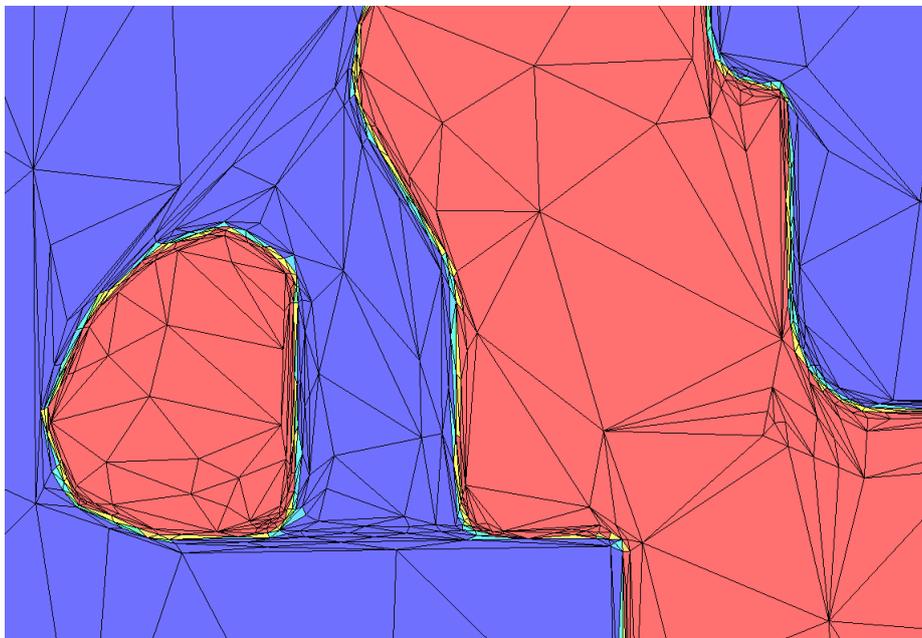


(b) enlargement of the symmetry plane

Figure 6.12: 24-th iteration mesh



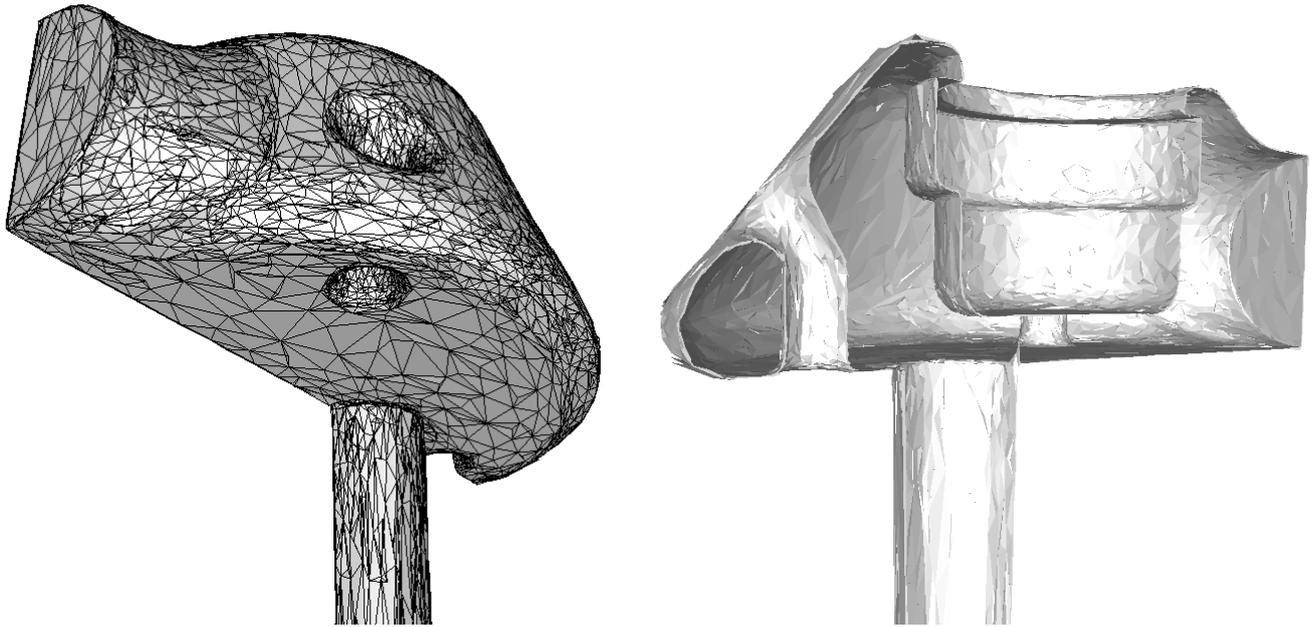
(a) symmetry plane



(b) enlargement

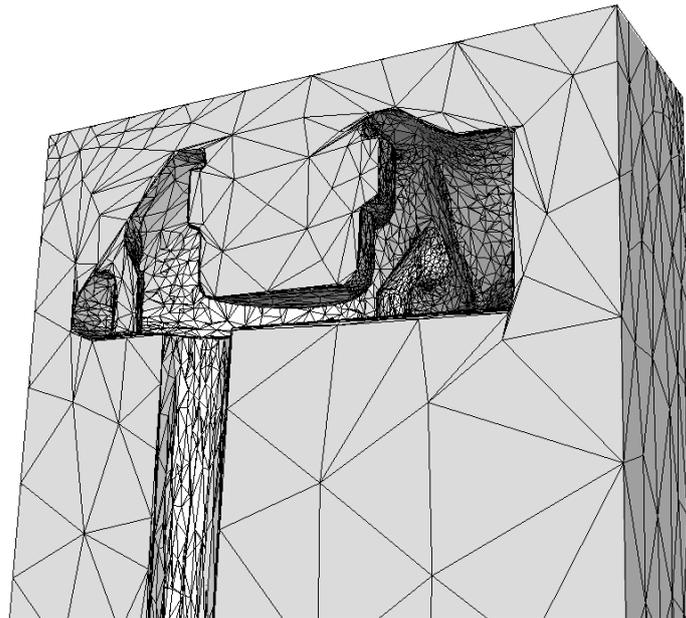
Figure 6.13: Approximated characteristic function  $g_\omega$ , interpolated on the 24-th iteration mesh

Besides, after extraction of sub-meshes corresponding to  $\omega$  (figure 6.14(a)) and its complement  $\Omega \setminus \omega$



(a) elements such that  $g_\omega \geq 0.9$

(b) elements such that  $0.1 \leq g_\omega < 0.9$



(c) elements such that  $g_\omega < 0.1$

Figure 6.14: Sub-meshes of the 24-th iteration (extraction is performed by filtering  $g_\omega$ )

(figure 6.14(c)), we can see the natural mesh inside each subdomain.

Finally, this method can be seen as detection and iterative refinement of elements in the vicinity of an interface, like in [Tezduyar *et al.* 1998, Stalický and Roos 1999, Aguilar-Villegas 2000]. The originality of the technique presented here, is the automatic anisotropy obtained for these elements and the fact that very complex 3D geometries can be treated (next chapter).

# Chapter 7

## Numerical applications

Many application fields could be concerned with multidomain meshes. We focus our attention on material forming (by finite element technology). Firstly, some Rem3D problems are addressed (section 7.1), then, some applications devoted to modelling at fine scale are presented (section 7.2 page 123), before the multidomain metric is proved to be efficient in boundary layer meshing (section 7.3 page 127). Lastly, a new mesh generation process can be derived from multidomain adaptation and interface capturing (section 7.4 page 132).

Another multidomain approach has already been investigated by [Tezduyar and Osawa 2001], where different numerical solvers are run successively on overlapping subdomains. The coupling presented here between subdomains is thought to be stronger, since all degrees of freedom are solved simultaneously.

### 7.1 Polymer forming

Within the framework of Rem3D, the concept of a multidomain simulation has been successfully introduced by [Batkam 2002]. Such a simulation can concern thermal coupling with the mold or over-molding with a solid insert. Previous computations were performed with the painfully technique of exact interfaces.

Such configurations are now revisited with the use of fuzzy interfaces (sections 7.1.2 and 7.1.3). But before, the section 7.1.1 is devoted to the comparison between the results obtained with exact and with fuzzy interfaces.

#### 7.1.1 Validation of the fuzzy multidomain approach

So as to study the influence of the fuzzy interface approach on Rem3D results, we consider the bidomain test case of figure 7.1 (a twisted tube in a cube). The exact interface mesh is already given on figure 6.2 (page 102) and a sectional view of the fuzzy interface mesh, generated by a multidomain metric is plotted on figure 7.2.

The same computation is performed on both meshes. It consists in a viscoplastic fluid (with a power law, whose consistency is 1230 Pa and exponent 0.75), initially at 250°C and injected by one side of the tube with a constant pressure of 1 MPa (the other side of the tube being at zero pressure). The steel mold (considered as quasi-rigid), initially at 100°C, is cooled by the ambient temperature of 25°C, while the air trapped inside the tube is initially at 50°C (in fact, the air is a third subdomain that flows in front of the fluid subdomain, in the same sub-mesh representing the tube).

In what concerns the computation with a fuzzy interface, if we consider a linear mixing law between subdomains, then a thin layer of air is artificially trapped between the mold and the fluid, during the filling stage. Since the air is modelled with a low conductivity, heat transfer between the fluid and the mold is wrong, which results in a wrong temperature field and wrong mechanical parameter values. The

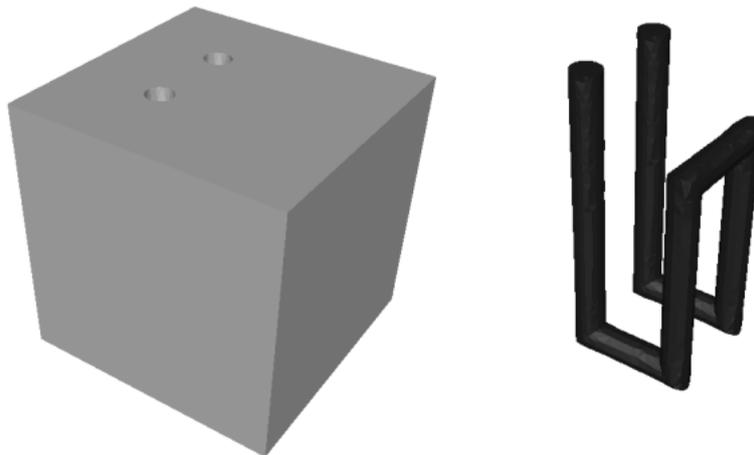


Figure 7.1: Validation test case for the fuzzy interface approach

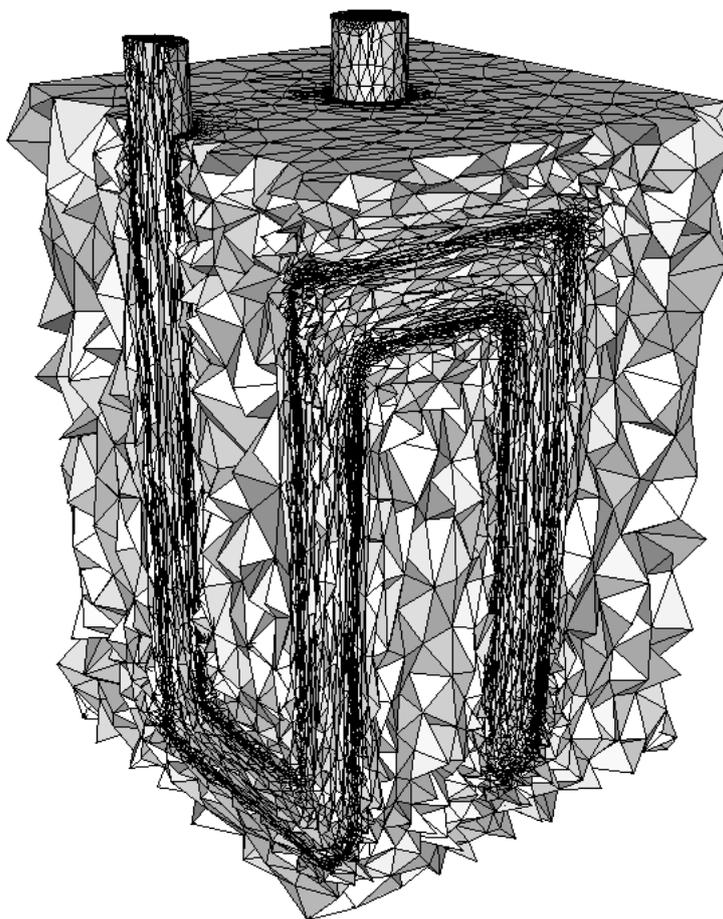


Figure 7.2: Sectional view of the bidomain mesh (69 201 nodes)

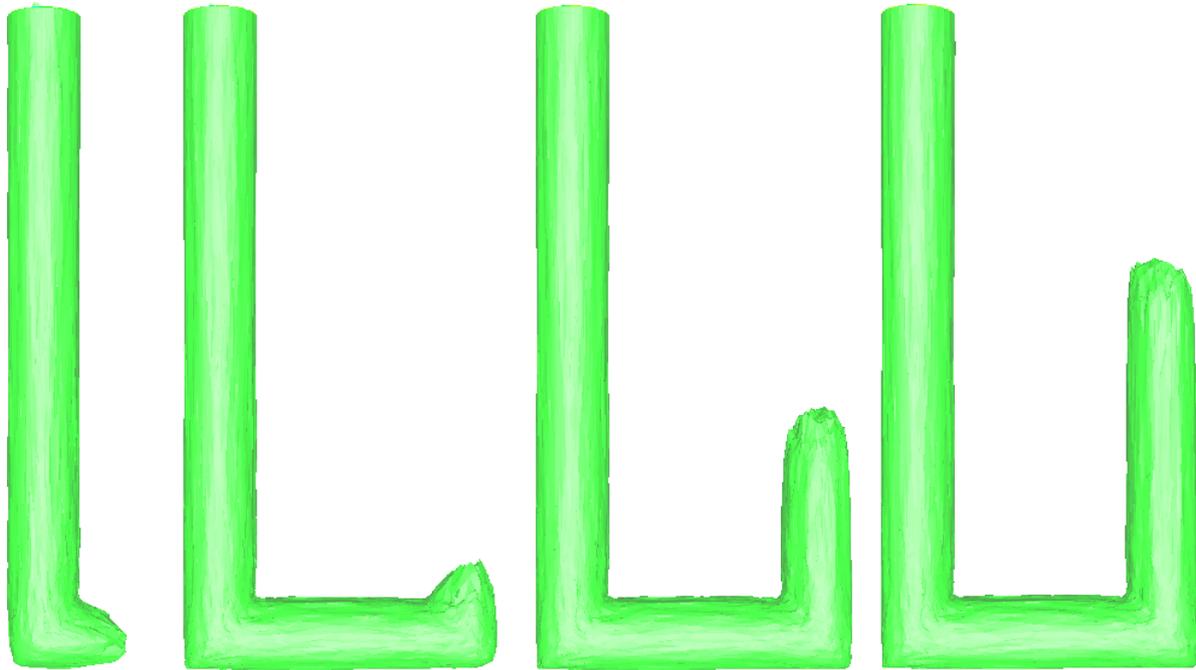
mixing law has thus been modified in disfavor of the air material.

Meshes of both approach have different sizes:

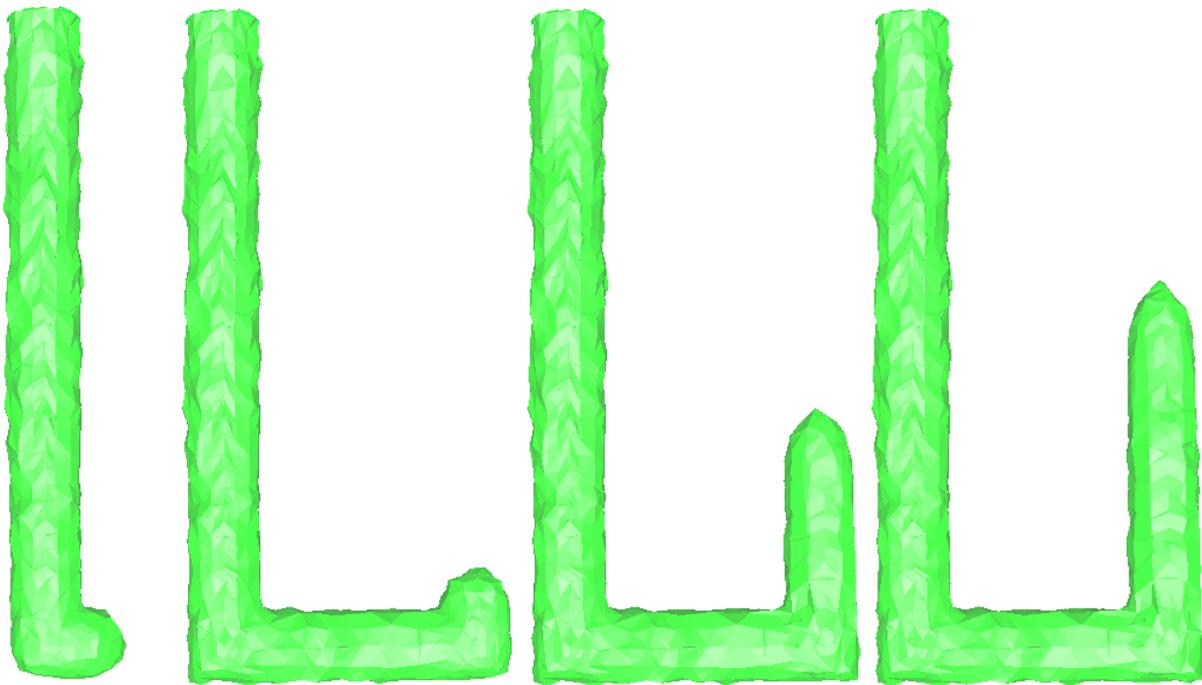
- 18 714 nodes for the mesh with an exact interface
- vs. 69 201 nodes for the mesh with a fuzzy interface.

This explains the difference between computation times (2.5 hours vs. 13 hours). However, the finer

mesh gives a better accuracy about the flow front (figure 7.3) and the temperature field (figure 7.4).



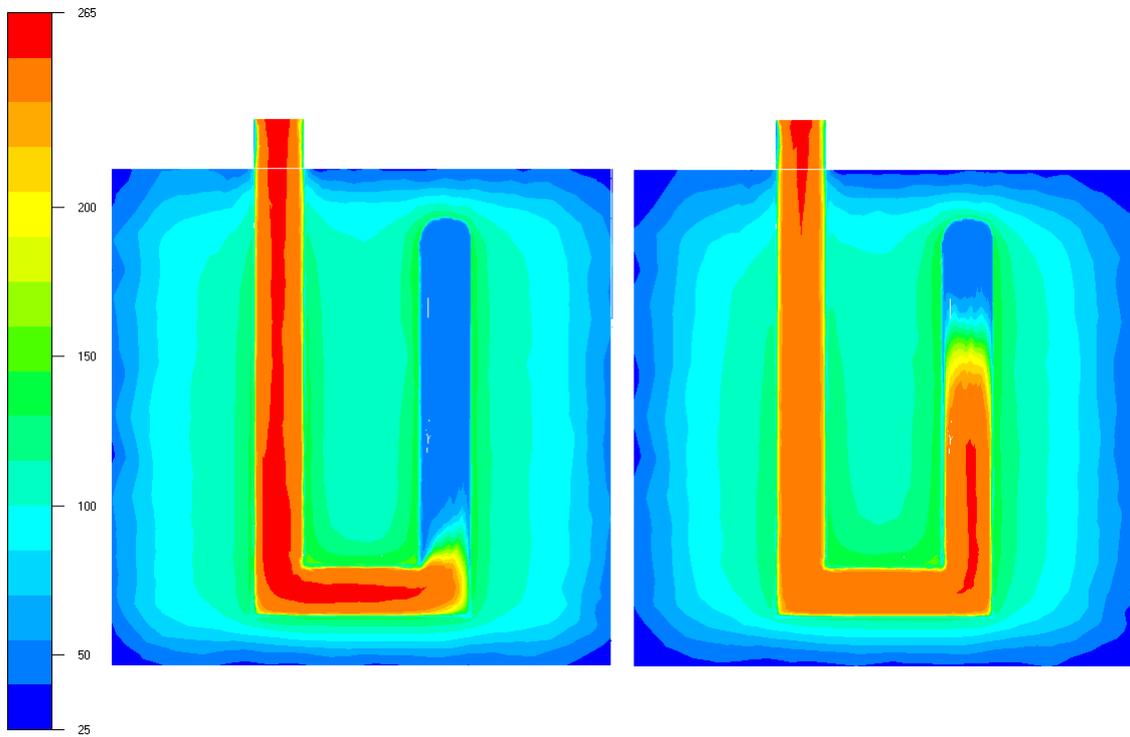
(a) fuzzy interface



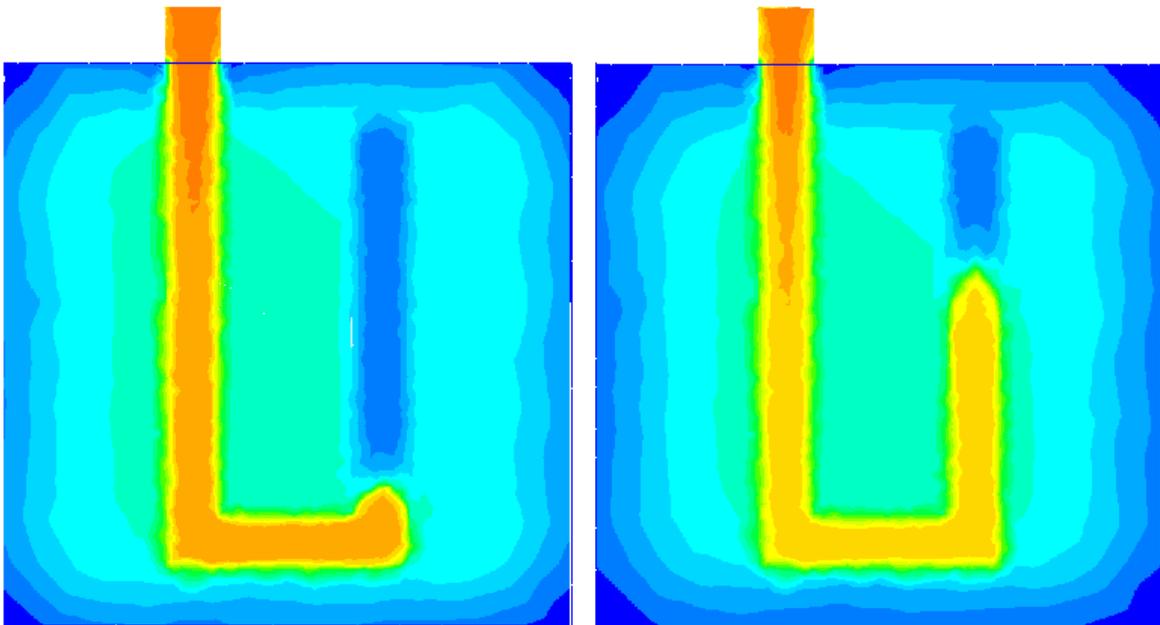
(b) exact interface (where the filling ratio is converted in a  $P^1$  field, losing the exactness)

Figure 7.3: Flow front (surface of the 50% filling ratio levelset) at timesteps  $t=1.5$  s, 3 s, 4.5 s et 6 s

These results are obtained on very different meshes, but the same CAD and the same natural metric have been used for both subdomains. The only difference is the interface treatment, for which the fuzzy approach leads to much more elements.



(a) fuzzy interface



(b) exact interface (same scale)

Figure 7.4: Temperature field ( $^{\circ}\text{C}$ ) on the same cutting plane at  $t=3$  s and 6 s

It can be seen on figure 7.4(a) that the air layer at  $50^{\circ}\text{C}$  is not fully eliminated between the mold and the fluid. Further investigations on the mixing law are required.

### 7.1.2 Injection molding with a complex geometry and thermal coupling

Simulating the injection molding process, with a good accuracy, requires thermal computations not only inside the mold cavity, but also inside the mold itself, sometimes composed of heterogeneous parts, with thermal regulation channels. Here, we study a classical industrial test case (introduced on figure 5.8 page 98), with the multidomain mesh (158 778 nodes) of figure 6.1(c) page 99.

The simulation concerns only the filling stage, other stages (cooling and packing) being less demanding about the mesh. The fluid (with the same rheology than in the previous section) is injected with an imposed rate of  $1000 \text{ mm}^3 \cdot \text{s}^{-1}$ , at a temperature of  $250^\circ\text{C}$ . The mould consists in three (quasi) rigid parts, all initially at  $50^\circ\text{C}$ , but with different densities, specific heat and conductivity. The ambient temperature is still  $25^\circ\text{C}$ , which is also the cooling temperature imposed in the thermal regulation channels.

Rem3D can perfectly deal with a multidomain mesh and fuzzy interfaces, so, the simulation has succeeded and the flow front evolution is given on figure 7.5. For the first time, this kind of anisotropic mesh has proved to be compatible with the mechanical solver (in what concerns the thermal solver, it was already known).

The final temperature field on the symmetry plane is given on figure 7.6. The thermal computation has also succeeded, in particular, there is no artificial self heating of partially filled elements (between the mold and the fluid). This kind of artificial self heating could have happened, since in those elements the averaged viscosity is rather high and the velocity field is not strictly zero.

These results are qualitatively close to those obtained with exact interfaces (not mentioned here). However for this example, treated by fuzzy interfaces, the end user only provides the CAD of the three mold parts and the CAD of the global computational domain independently (and the desired number of element layers inside each sub-mesh).

### 7.1.3 Over-molding

Other polymer forming processes can be treated by a multidomain technique, like multi-fluid injections (co-injection, gas or water assisted injection, for instance). Unfortunately in these cases, positions of the fluids are not known before the computation is run. A static a priori multidomain mesh could not capture such moving interfaces: a dynamic adaptive meshing is required.

A R-adaption technique, like [Bigot 2001], could be invoked during the computation so as to tighten the mesh (assumed to be fine enough) around moving interfaces. A R-adaption consists only in node position changing, not in mesh topology changing (contrary to remeshing). In [Bigot 2001], this node position changing is used to shrink the partially filled elements (in the vicinity of an interface between several subdomains). This mesh operation introduces a mesh velocity, which implies the use of an ALE (Arbitrary Lagrangian-Eulerian) formulation of equilibrium equations.

However, in what concerns the injection process on solid insert (over-molding), the position of the insert subdomain is known at the beginning (even if it can move during the process). An a priori multidomain mesh can then be built, so as to capture the interface between the insert and the cavity around. An example has already been given on figure 6.8 (page 109), where the insert and the cavity are depicted on figure 6.6 (page 108).

Here, we consider a second example, inspired by [Coupez *et al.* 2004], which deals with injection on deformable blades. The simple geometry of interest (courtesy of Snecma Propulsion Solide) involves a unique rectangular blade, considered as the insert (figure 7.7). A bidomain mesh is generated with very fine elements around the interface between the insert and the cavity (figure 7.7(a)).

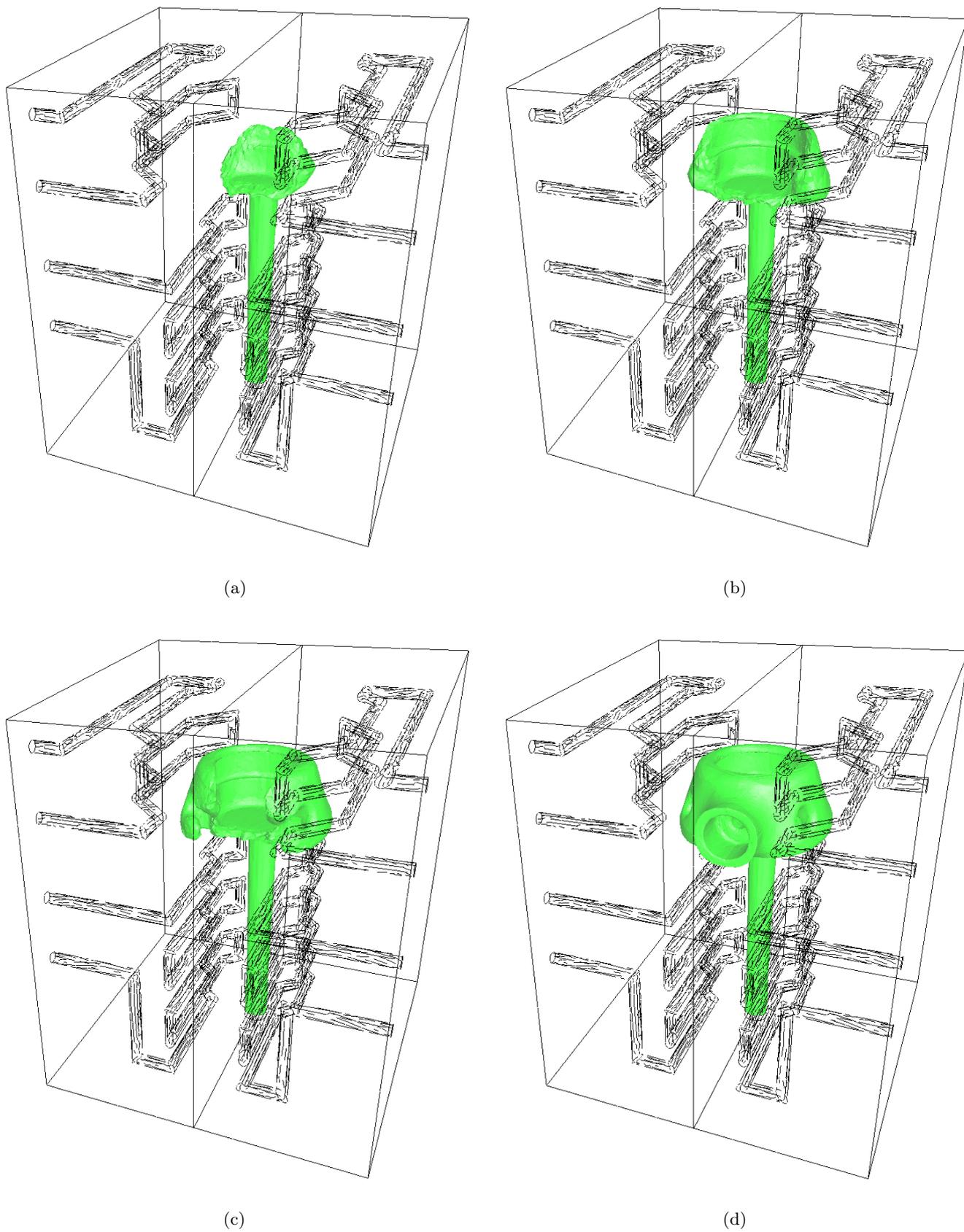


Figure 7.5: Filling ratio (50% levelset after  $P^1$  averaging) during the filling stage

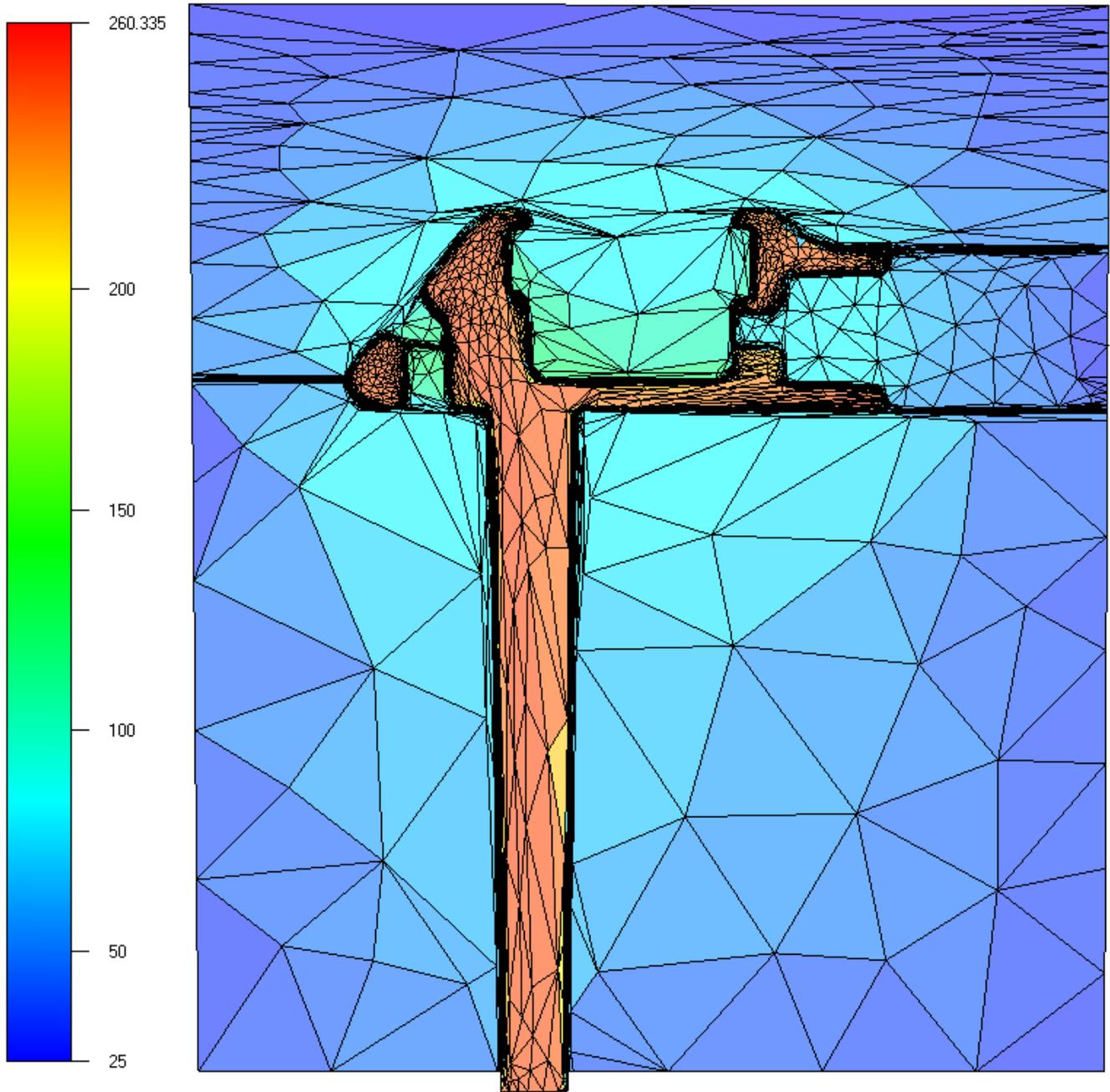


Figure 7.6: Temperature at the end of the filling stage, on the symmetry plane

Furthermore, the interface, between the injected fluid and the air trapped inside the cavity, is not captured by the multidomain metric before the computation. Instead, this interface is captured by the aforementioned R-adaption dynamically, during the computation. Thanks to this R-adaption, the diffusion of the filling ratio interpolation is highly reduced (figures 7.7(b) and 7.7(c)).

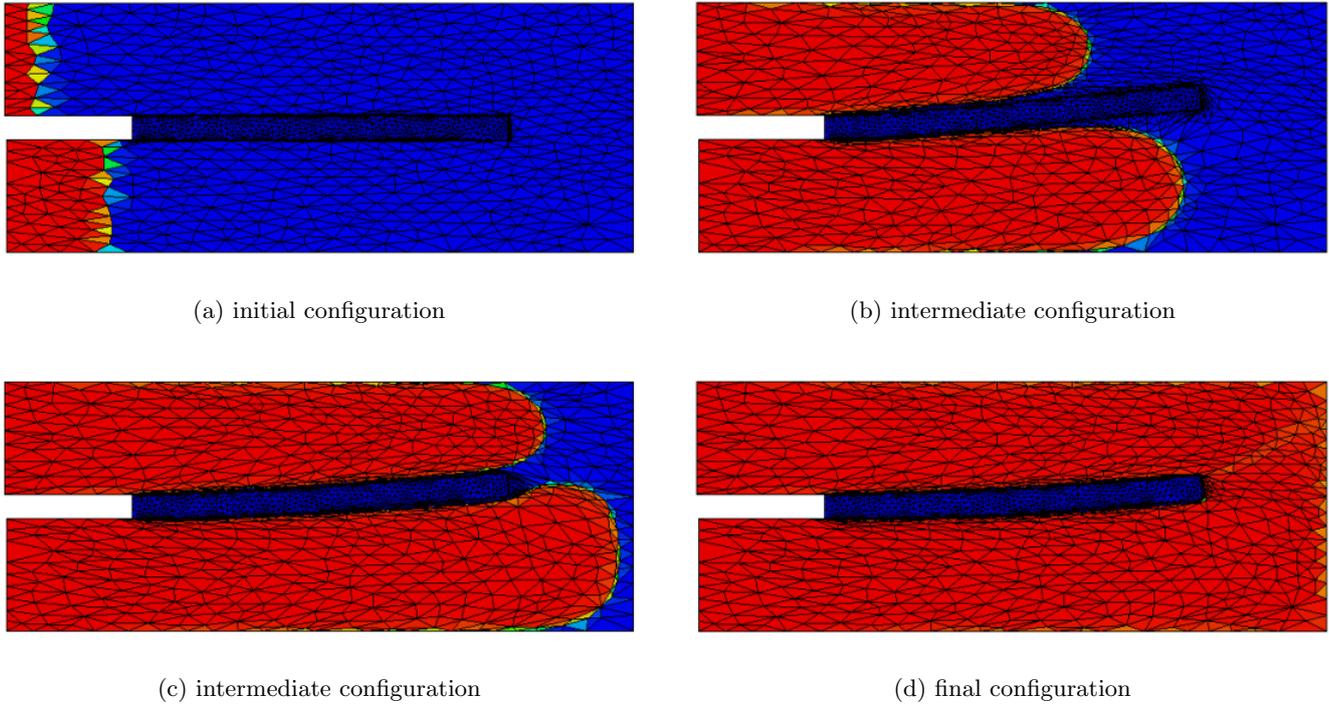


Figure 7.7: Over-molding: evolution of the fluid filling ratio (red elements are full, blue ones are empty)

In this simulation, the fluid has not the same initial position above and under the blade, which prevents the blade from remaining at the same position during the process. The fluid front above the blade is delayed, which induces a raising of the insert, until its final position (figure 7.7(d)). This raising makes the fluid even more delayed above the blade.

In fact, the moving interface between the insert and others subdomains (the fluid and the air) is followed dynamically by the same R-adaption technique. Since the mesh is initially very refined around this interface, this capturing remains accurate during the (quite small) movement. In other words, the R-adaption technique is improved by the use of a mesh generated by a multidomain metric.

This example shows also that an ALE framework (where the mesh velocity is given by the node position changing of a R-adaption) can be used in conjunction with a mesh generated by a multidomain metric (only Eulerian computations has been presented, up to now).

## 7.2 Computations at microscopic scale

Multidomain meshes are not limited to the macroscopic scale; they can also be useful for computations at finer scales. Numerical studies of material behaviors can take the microstructure into account, with many heterogeneities between the components (macroscopic laws can then be deduced by homogenization techniques).

In such simulations, each component can be seen as a subdomain and the microstructure involves interactions between many subdomains. In general, the microstructure mesh needs to be very fine in the vicinity of interfaces between components (and coarser inside each subdomain). The multidomain metric is a good candidate for that purpose.

Here, we present two applications coming from our research center: a rigid inclusion movement induced by a pure shearing flow (section 7.2.1) and a compression test on a polycrystalline sample (section 7.2.2).

### 7.2.1 Rigid inclusion movement in a pure shearing flow

We consider a rigid and spherical inclusion in a viscous fluid, subjected to a pure shearing flow (figure 7.8). The aim of this study [Ménard 2003] is to compare the results obtained by a 3D model to those

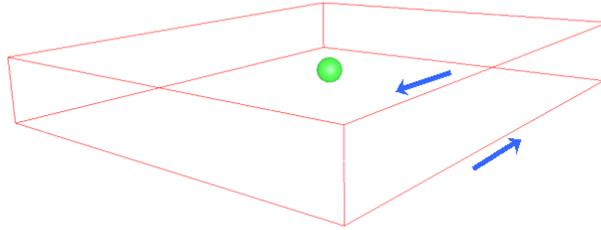


Figure 7.8: Parallelepiped computational domain, with a spherical inclusion and imposed shearing boundary conditions (arrows)

obtained by a 2D model (hopefully, the multidomain metric works, whatever the spatial dimension is).

Focusing our attention on the 3D case, the computational domain is a simple plate and its subdomains are the rigid sphere (representing a gel inclusion), located at the center, and its complement (a polymeric matrix). The mesh has been highly refined around the interface between both subdomains, by the multidomain metric method with a fuzzy interface (figure 7.9), so as to interpolate accurately the strong gradients that take place in this region (figure 7.10). Without interface capturing and refinement, the  $P^0$  discontinuous interpolation of the inclusion characteristic function would present a high numerical diffusion, which would lead to non accurate results.

One of the results obtained in this study is given figure 7.11, where some vortices can be seen around the sphere (the sphere appears in green, as the 50% levelset of the approximated filling ratio, after a continuous  $P^1$  averaging).

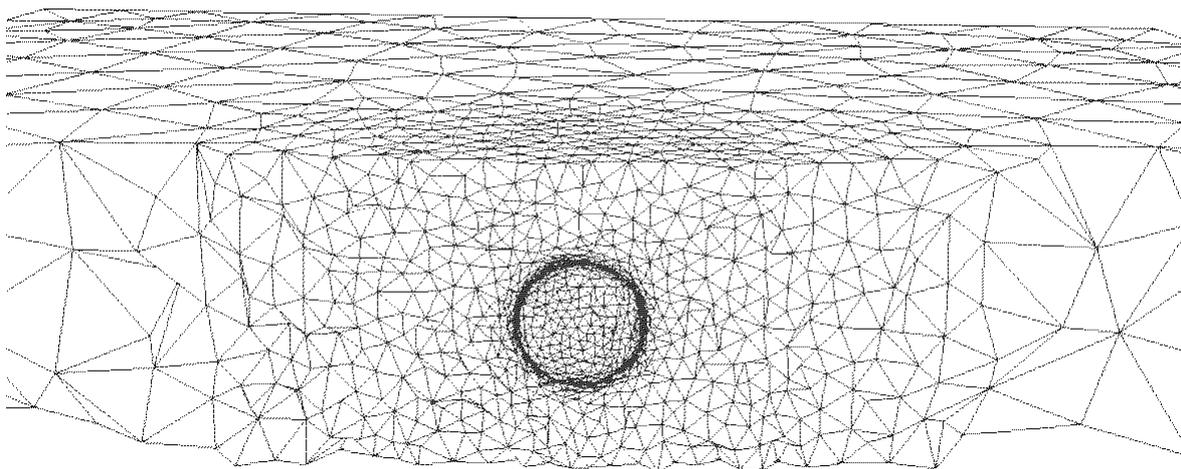


Figure 7.9: Sectional view of the mesh, after interface capturing by a multidomain metric

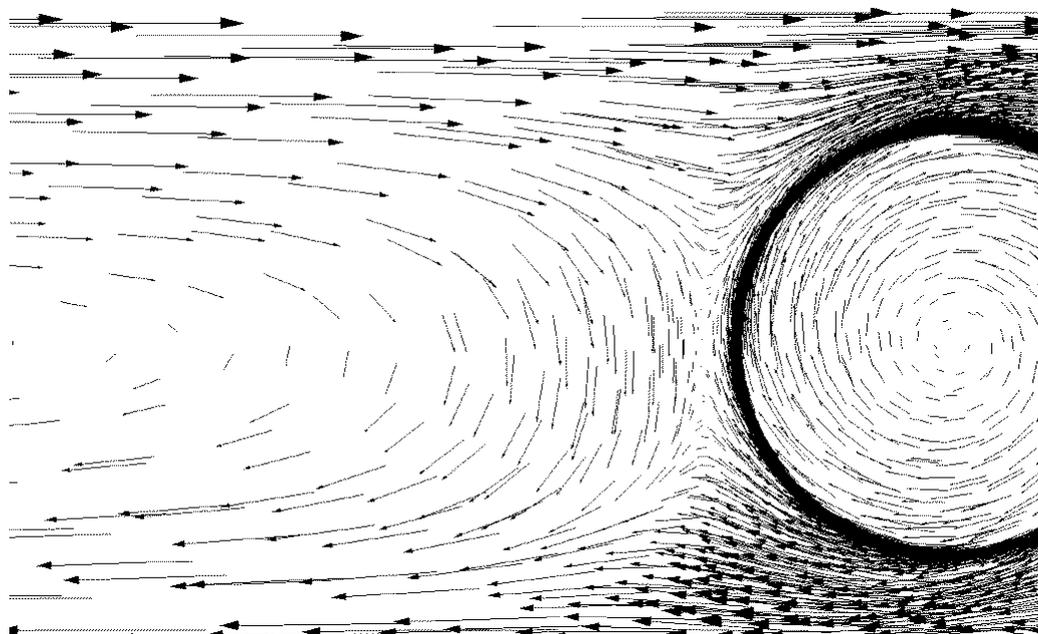


Figure 7.10: Velocity field on a cutting plane  
(the inclusion behaves like a rigid body, but its movement is not imposed, instead, it results from the shearing flow)

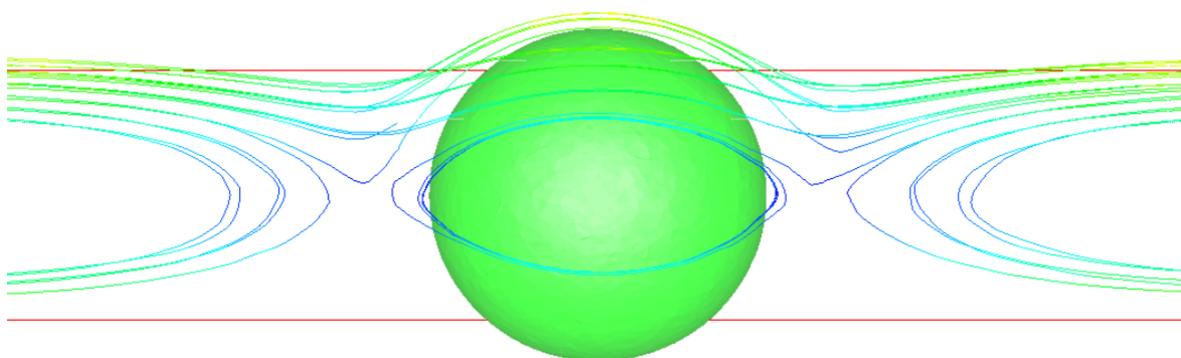


Figure 7.11: Some particle tracing around the inclusion

### 7.2.2 Polycrystalline deformation

Other finite elements computations than Rem3D could benefit from the multidomain metric. Still in material forming, some results obtained by [Rey 2003] with Forge3 are presented here, involving meshes driven by a multidomain metric.

In this section, computational compression tests are performed on metal samples, consisting of several polyhedral grains, as a model of experimental tests on polycrystalline aggregate. Each grain is a Voronoï cell (indeed, a homogeneous crystallization from several random germs leads to Voronoï cells) and the sample is a 3D Voronoï diagram (figure 7.12(a)). Each Voronoï cell is a subdomain (the global

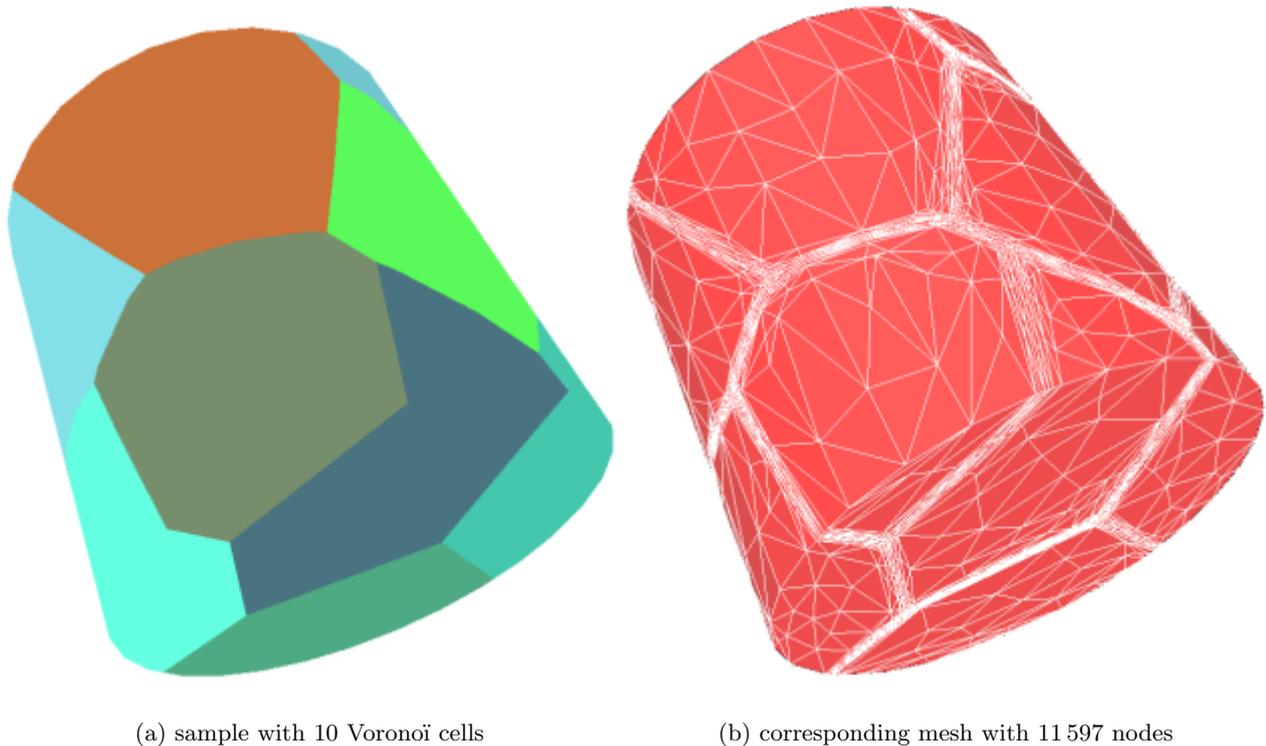


Figure 7.12: Cylindrical sample for compression tests

computational domain being a cylinder). The sample is meshed with an interface capturing between grains (figure 7.12(b)). The final mesh has a reasonable size (about 1 000 nodes for each grain) and is refined enough around each grain, so that an accurate simulation can be run.

First simulations with Forge3 are concerned with an uniaxial compression, involving a simple viscous modelling of the material, with a different viscosity for each grain (heterogeneity is taken into account). The multidomain mesh is compatible with the computation, since 70% of compression are attained without remeshing. It is thus proved that a mesh generated by a multidomain metric can be useful, even in a pure Lagrangian formulation (Forge3).

Further simulations has been performed, involving a more representative material law (with crystalline orientation taken into account) for each grain. Again, the finite strain is tolerated by the multidomain mesh [Rey 2003].

This polycrystalline study shows also that there is no limit about the number of subdomains: on figure 7.14, a cylindrical sample with 50 subdomains has been meshed.

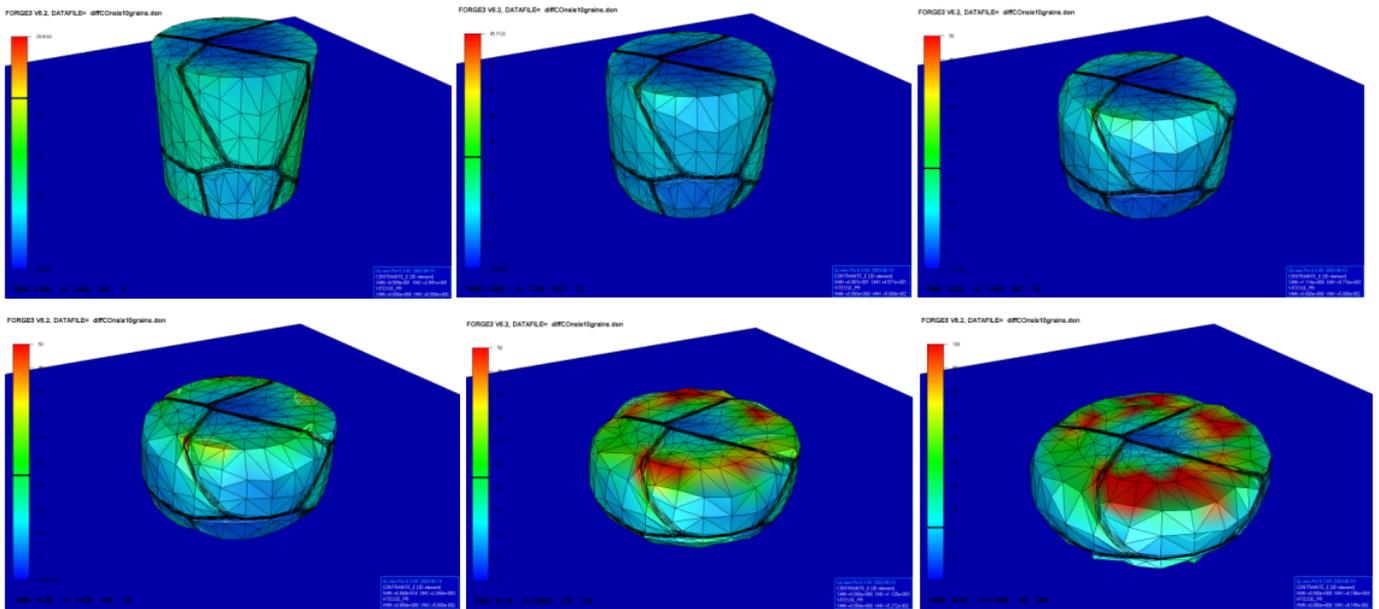


Figure 7.13: Compression test between two rigid plates (the color scale plots the equivalent von Mises stress and the bottom plate appears is in blue)

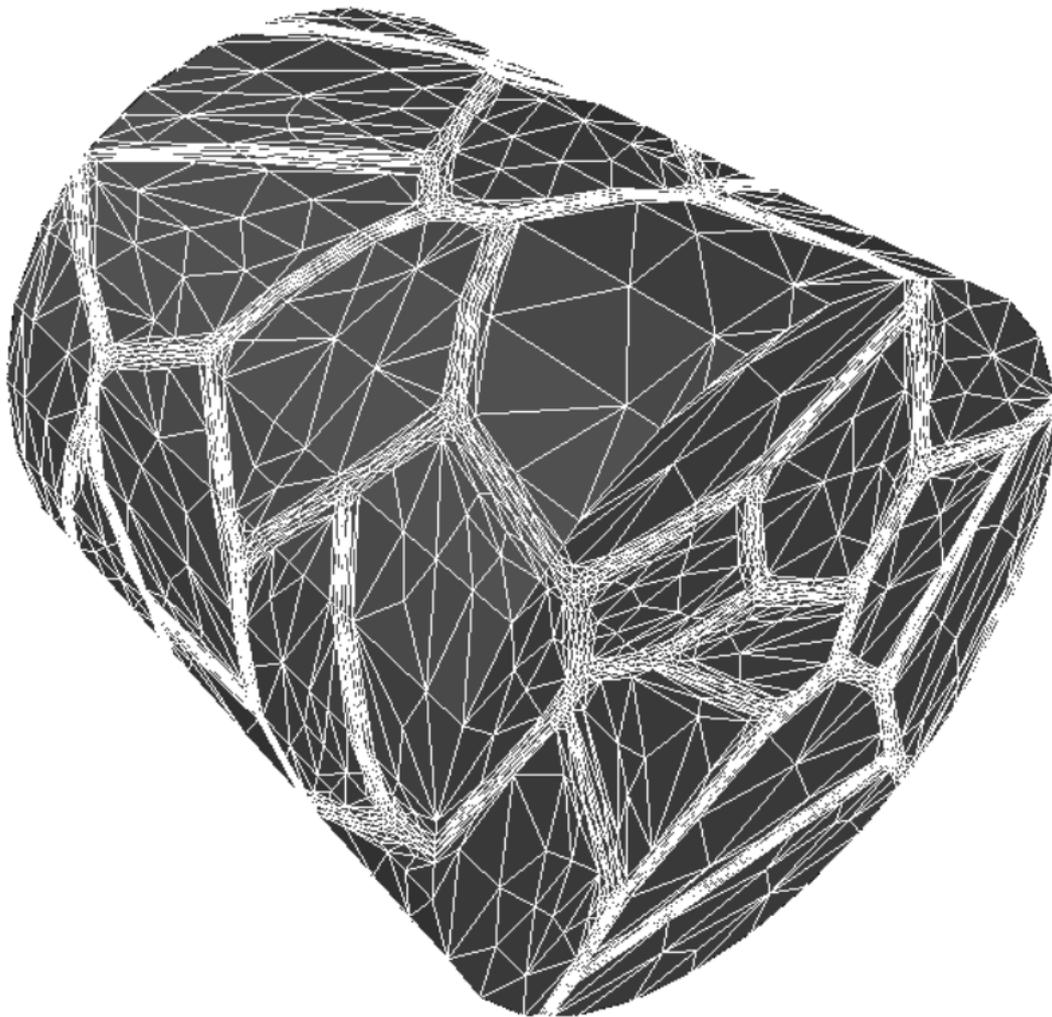


Figure 7.14: Another sample with 50 grains

## 7.3 Boundary layers

In many simulations, usually in computational fluid dynamics, a boundary layer is required for the mesh. More precisely, several elements layers aligned with the boundary are needed, within a specific thickness [Jansen and Shephard 2001]. In what concerns Rem3D, a mesh with a boundary layer (even if this layer is not accurate) would be interesting, since it would lead to a better approximation of:

- the temperature gradient between the injected fluid and the mold
- the final polymer layer after a gas-assisted injection process
- the presence of a thin lubricant layer.

So as to build such a boundary layer, several techniques can be considered: without metric (as a mesh extrusion from the boundary [Garimella and Shephard 1998]) or with a metric (for a mesh size to be imposed orthogonally to the boundary [Castro-Díaz *et al.* 1997]). In the Rem3D framework, a simple (and undocumented) node moving technique towards the boundary has been widely used.

In fact, the multidomain metric leads straightaway to a totally new alternative (section 7.3.1). However, for the thickness of the boundary layer to be accurately controlled, a more powerful technique is proposed (section 7.3.2).

### 7.3.1 Monodomain mesh improvement

As evoked in section 6.2.3 (page 106), the  $P^0$  gradient computation can use non adiabatic boundary conditions (adiabatic means no jump across the boundary faces). If an infinite jump of the characteristic function is considered across the boundary faces, then the multidomain metric tends to adapt the mesh in the vicinity of the (global) boundary. It is not always required, since this boundary is already triangulated by the mesh boundary. However, several element layers introduced in this region (figure 7.15), as if it were an interface between subdomains, can be interesting.

Some adiabatic boundary conditions can still be used, where such a boundary layer is not useful (on entrance or exit gates and on symmetry planes, for example). This is the case on figure 7.16, where the symmetry plane and the entrance gate have no boundary layers.

### 7.3.2 Boundary layer with a fixed thickness size

Unfortunately, the preceding method does not ensure a constant thickness size for the boundary layer and the number of element layers is not controlled. For specific applications, a thickness size and a number of element layers are prescribed.

This is the case for the clay extrusion process [Abbad 2003]. Rheological and frictional tests show that, above a particular stress threshold, the contact between the clay and the extrusion die becomes frictionless. In fact, the water of the clay moves outwards, under pressure, which lubricates the extrusion.

Computations of such a process could employ classical frictionless contact conditions, as for the casting process [Saez 2003]. However, a new treatment of the frictionless contact is proposed in this section. In this approach, the extrusion concerns not only the clay, but also a thin water layer, whose viscosity  $\eta$  is related to the thickness size  $e$ , so as to compute an equivalent friction coefficient  $\alpha$  between the shear stress  $\tau$  and the wall velocity  $v_{\text{wall}}$ . We obtain the following equivalent friction law

$$\tau = \alpha v_{\text{wall}} \quad (7.1)$$

$$\text{where } \alpha = \frac{\eta}{e} \quad (7.2)$$

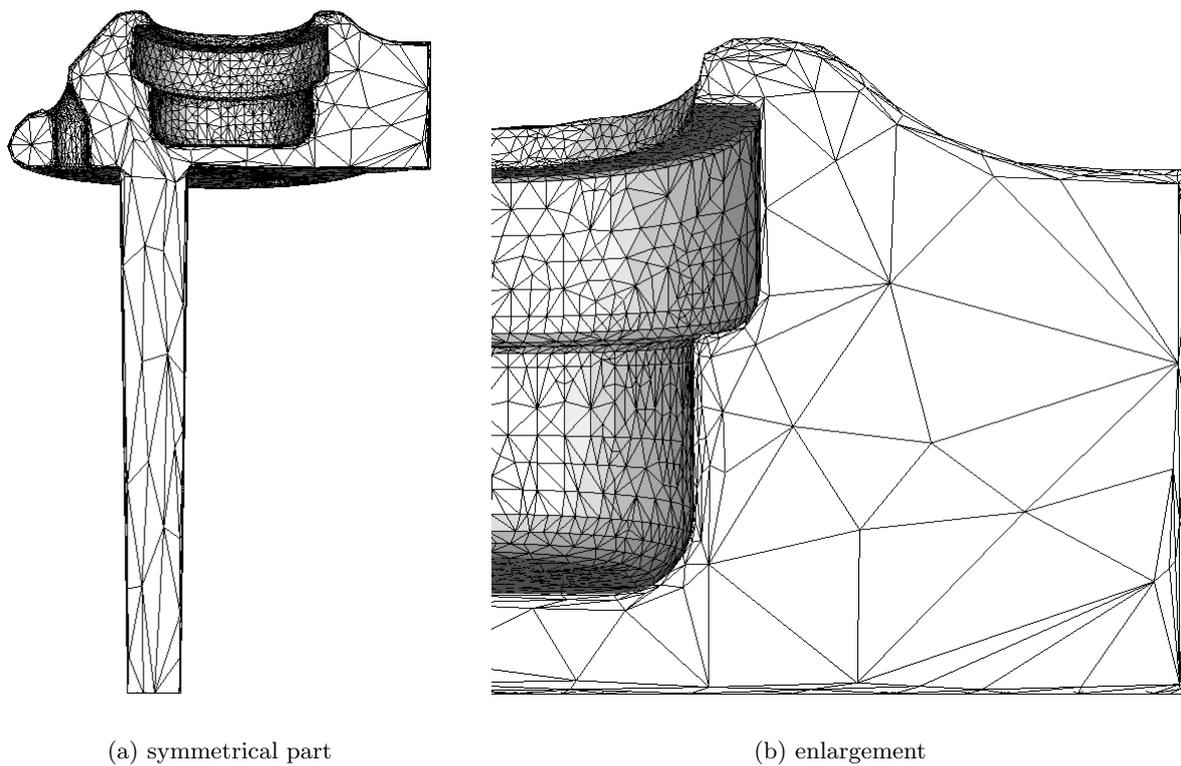


Figure 7.15: Natural and monodomain mesh, with a boundary layer generated by a multidomain metric

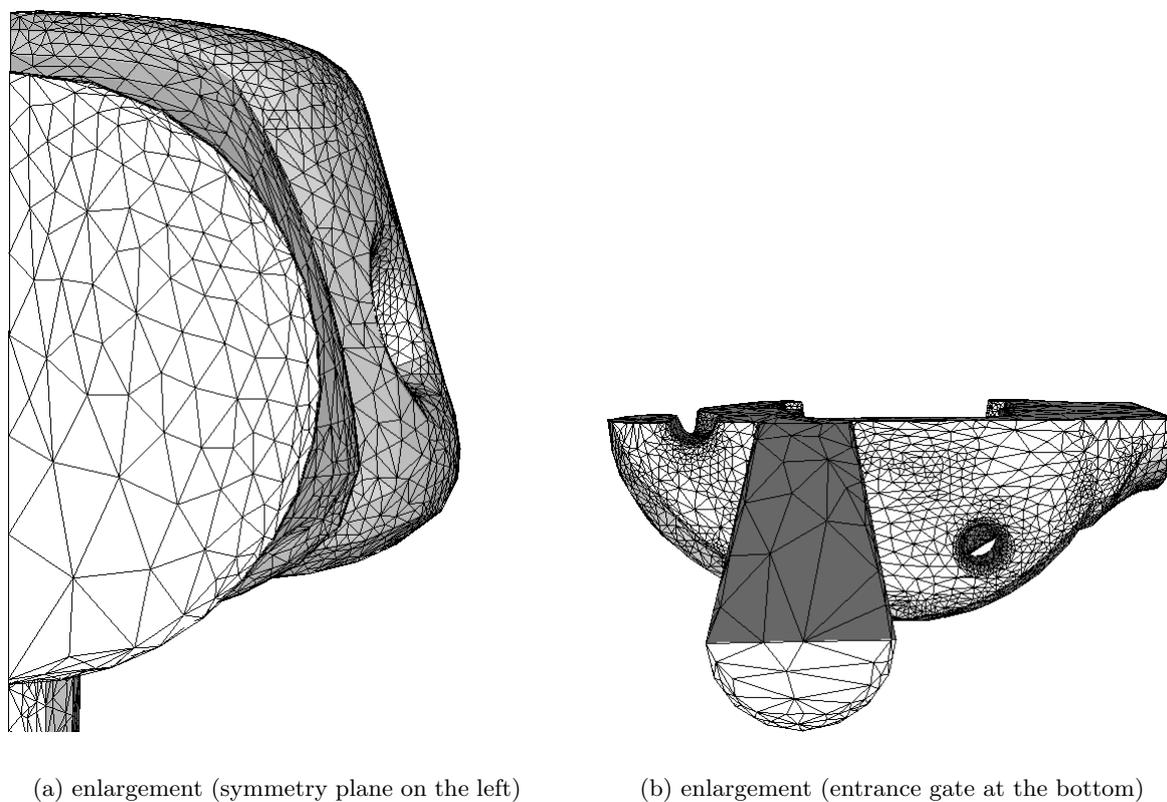


Figure 7.16: The same mesh with new enlargements, so as to see the adiabatic effect

The computation involves a subdomain for the clay and another subdomain for the water. Since the extrusion die is initially full, the geometries of those subdomains are known a priori. The multidomain metric can then be used, so as to build an anisotropic mesh (figure 7.17(a)), with a natural refinement in

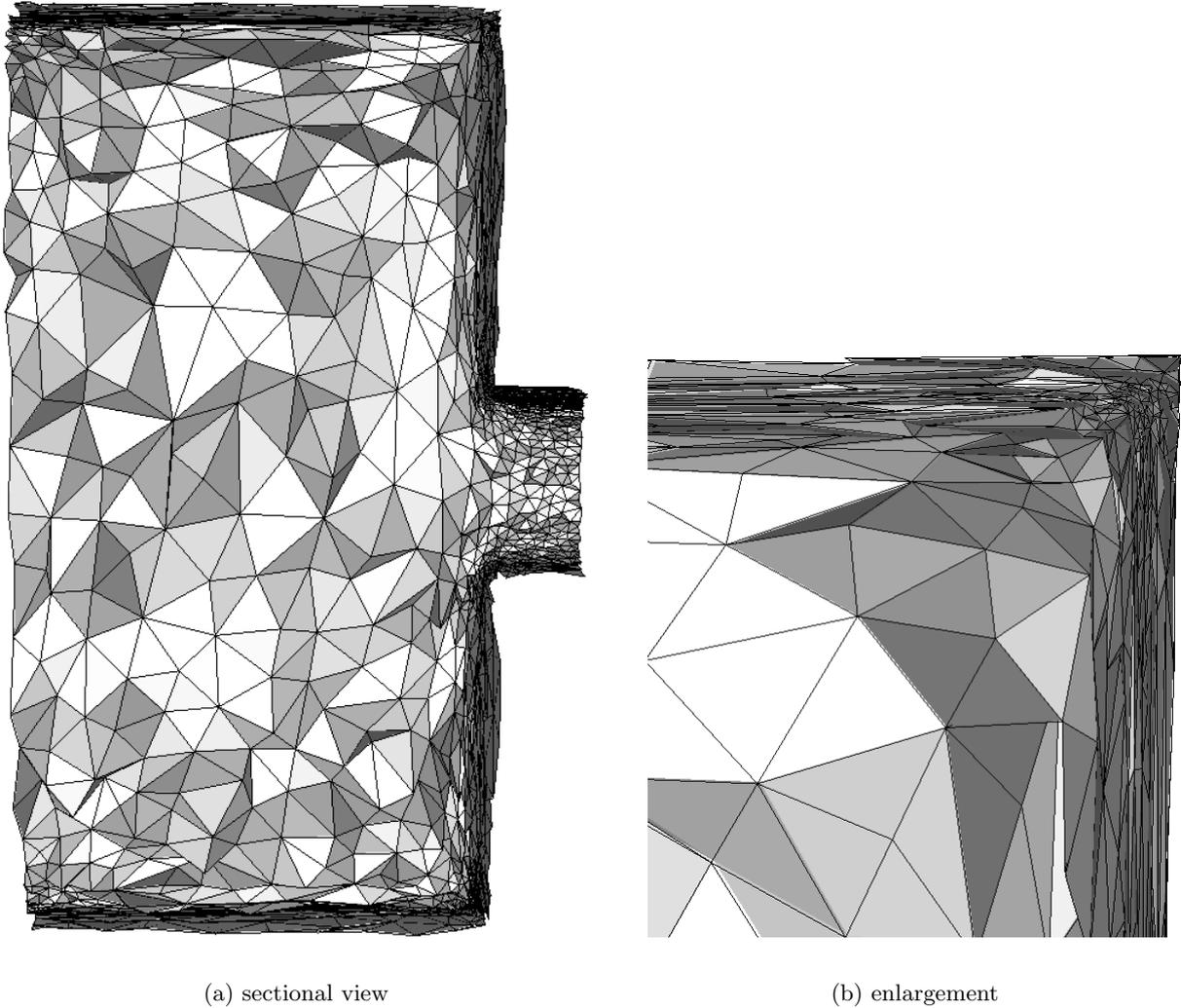
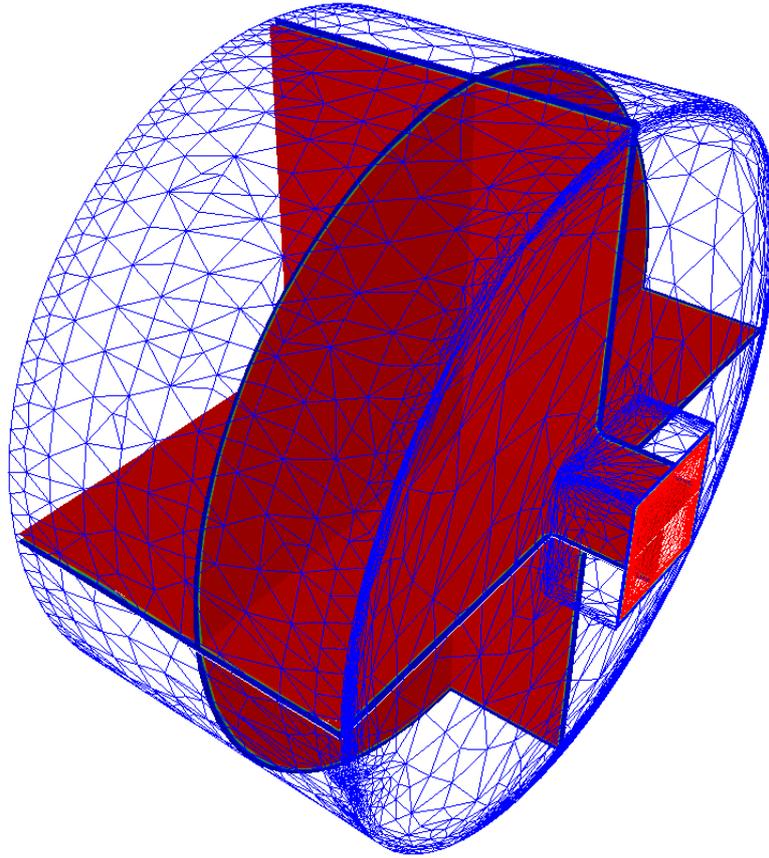


Figure 7.17: Extrusion die (100 mm diameter), with a 20 mm large exit gate and a 1 mm thick boundary layer (83 863 nodes)

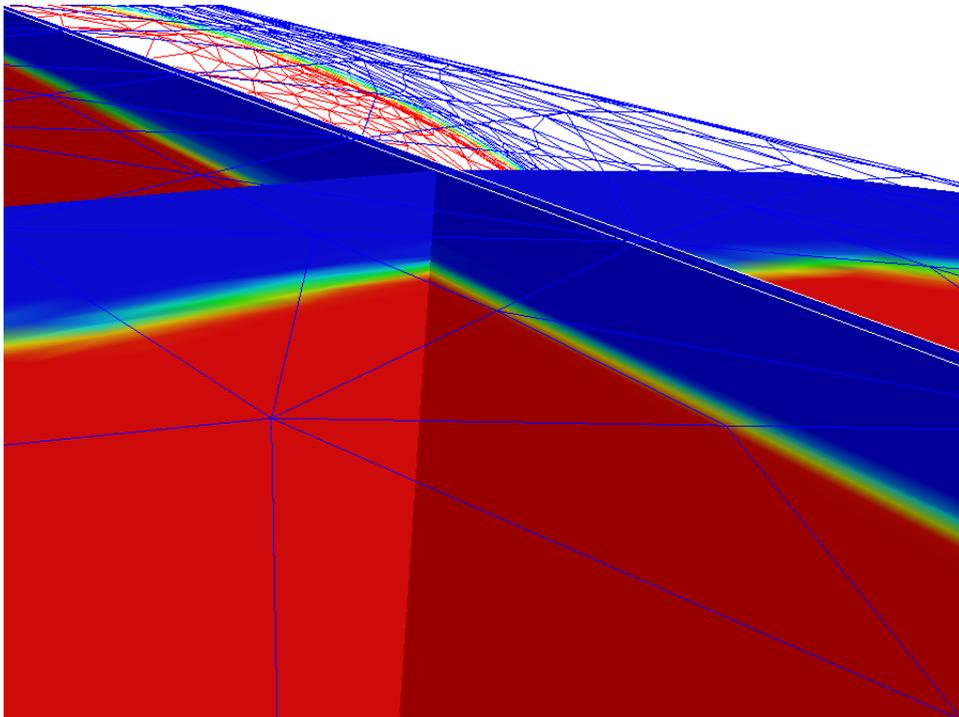
the clay subdomain, a natural refinement in the (thin) water subdomain and a high refinement between both (figure 7.17(b)).

So as to define the clay subdomain geometry, CAD operations can be avoided by skinning progressively the voxelization of the extrusion die, with the desired thickness size (except at the entrance and exit gates). Such a skinning consists in removing external lit voxels. The resulting voxelization is then used in the computation of the approximated characteristic function (filling ratio) of the clay (figure 7.18), on the adapted mesh (the filling ratio of the water layer being its complement). Such a bidomain mesh has a subdomain dedicated to the boundary layer.

The flow of both fluids can then be simulated, with an imposed pressure at the entrance gate (on the left), while the exit gate has free boundary conditions (figure 7.19). In fact, for this computation, no stress threshold has been taken into account, that is the reason why, in the dead zone of the die (on the upper right corner), a back flow is observed (figure 7.20).



(a) global view (boundary mesh and three cutting plane by transparency)



(b) enlargement

Figure 7.18: Clay filling ratio: red elements are full, blue one are empty (full of water)

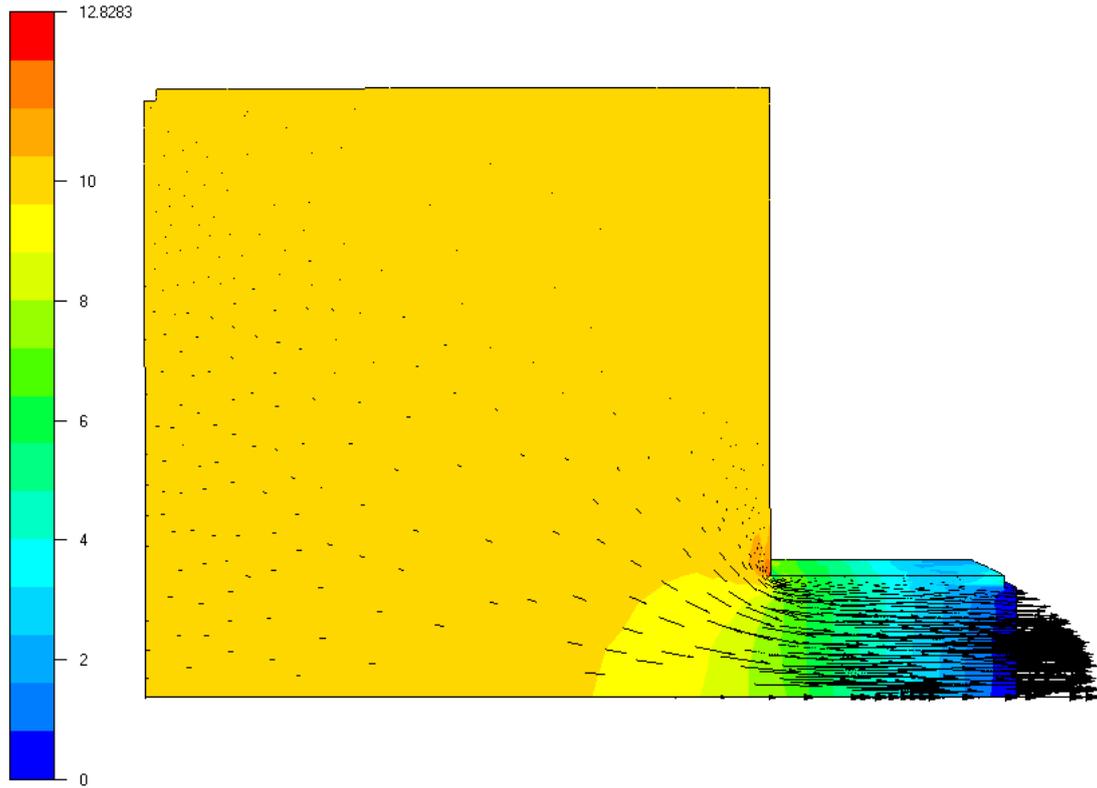


Figure 7.19: Velocity field (arrows) and pressure field (color scale) on a symmetry plane

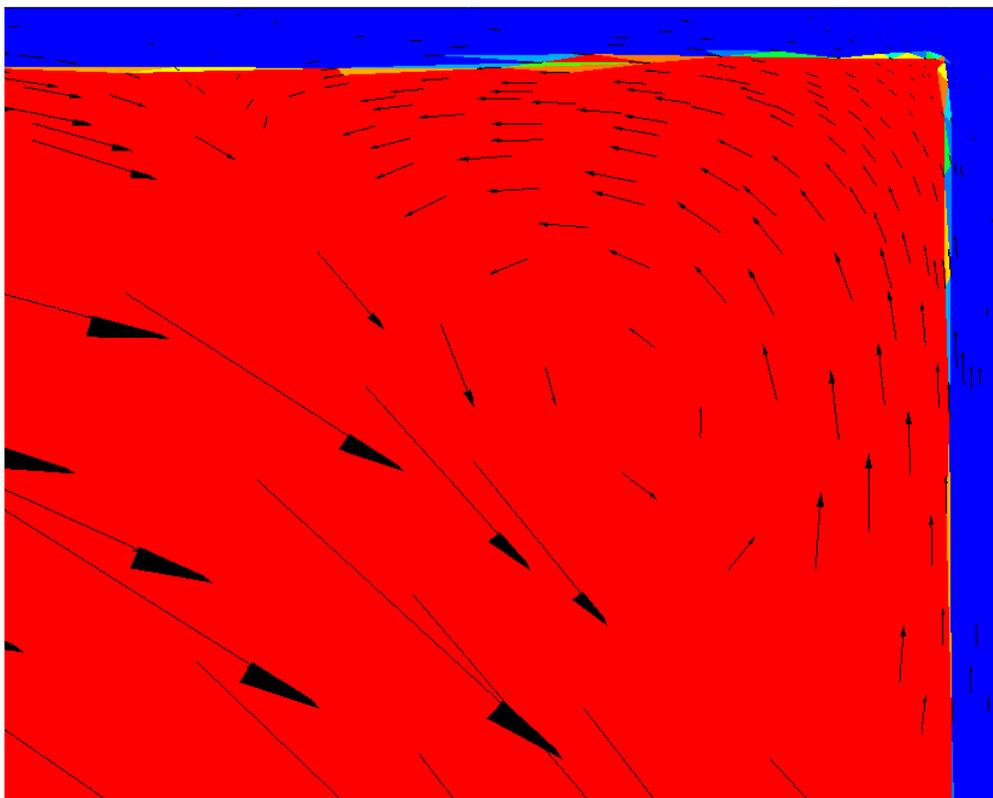


Figure 7.20: Velocity field (arrows) and clay filling ratio (color scale) with enlargement of the dead zone

## 7.4 Mesh generation

Lastly, the multidomain metric can be used as a mesh generation technique, in other words, as an alternative of chapter 2 (page 19). Indeed, a mesh of a subdomain  $\omega$  can be extracted from a global mesh of a bounding box, adapted with a bidomain metric.

Let us take, for example, a subdomain  $\omega$  representing the geometry of a biplane (figure 7.21), which is

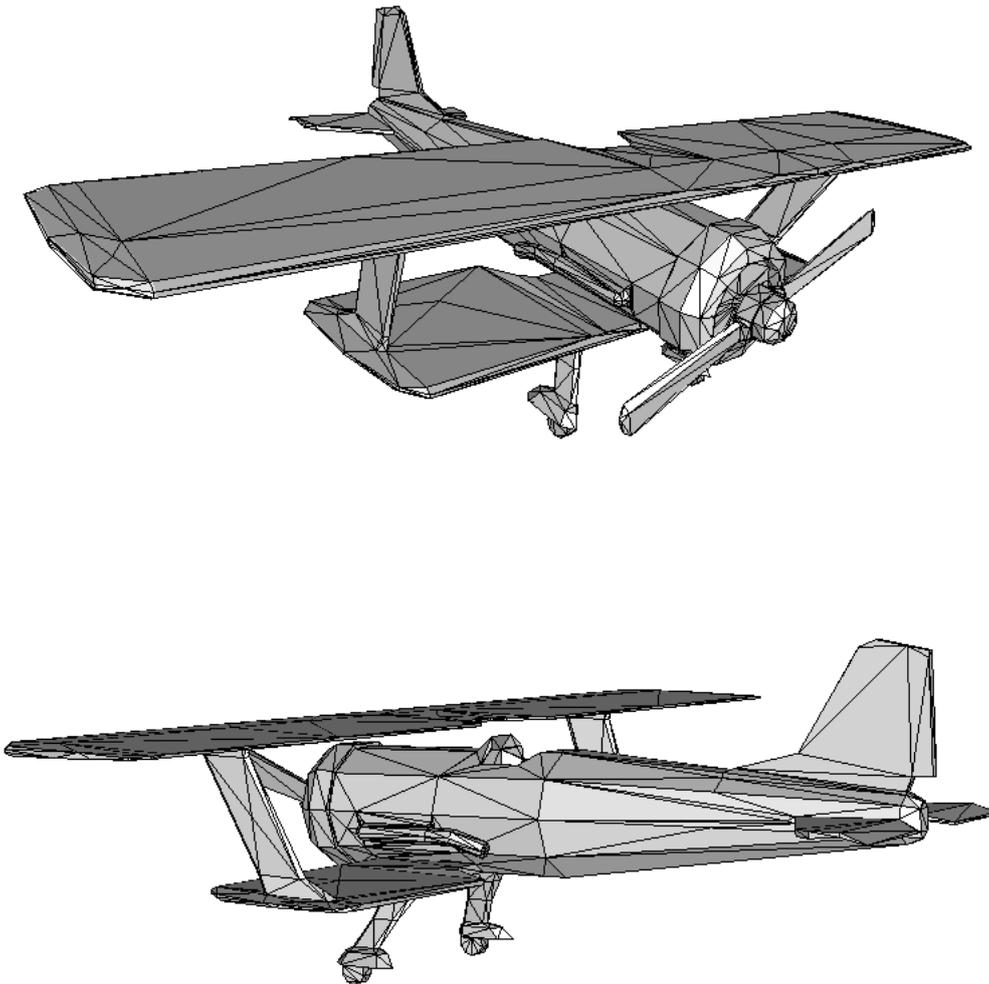
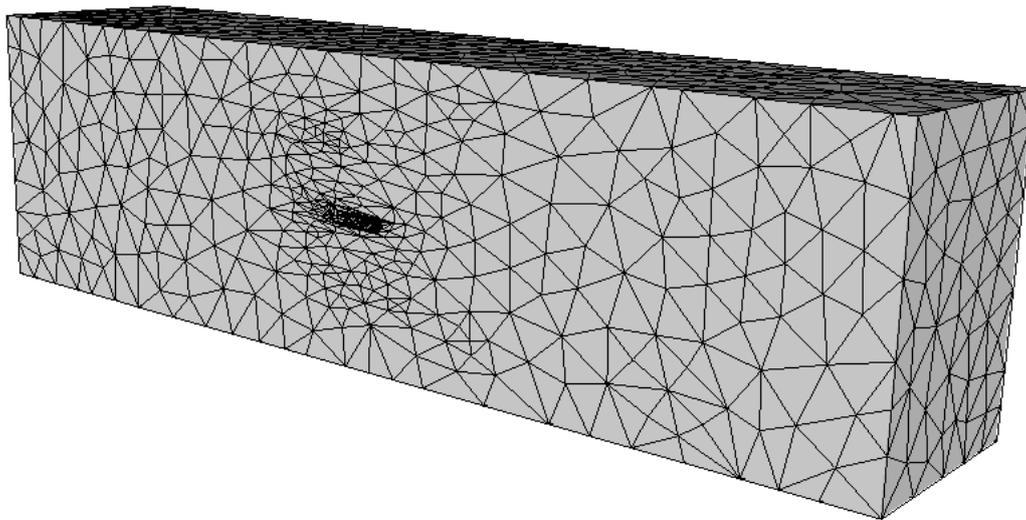


Figure 7.21: A coarse mesh defining the subdomain geometry (this mesh is used for the voxelization)

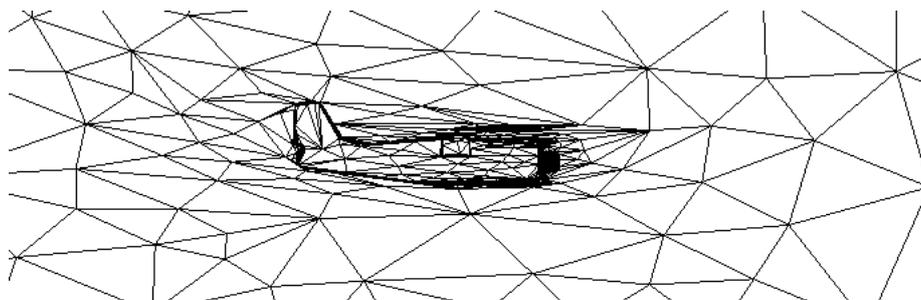
symmetrical, if the airscrew is removed. This subdomain is then considered in a computational domain  $\Omega$ , whose geometry is simple and sufficiently large to contain the symmetrical part of  $\omega$  (figure 7.22(a)). The multidomain metric leads to a mesh of  $\Omega$  that captures well the details of  $\omega$ 's geometry (figure 7.22(c)). The adaptive process is robust, since the initial mesh of  $\Omega$  contains only 6 elements and the final mesh has succeeded in refining around very small details (small with respect to  $\Omega$ 's dimension).

The boundary of  $\omega$  has been captured by the bidomain mesh of  $\Omega$ . If we extract the sub-mesh of elements whose filling ratio  $g_\omega$  is greater (respectively lower) than 50%, we obtain a suitable mesh of  $\omega$  (respectively of  $\Omega \setminus \omega$ , as we can see on figure 7.23). Unfortunately, the boundary of those sub-meshes is quite noisy (figure 7.24(a)).

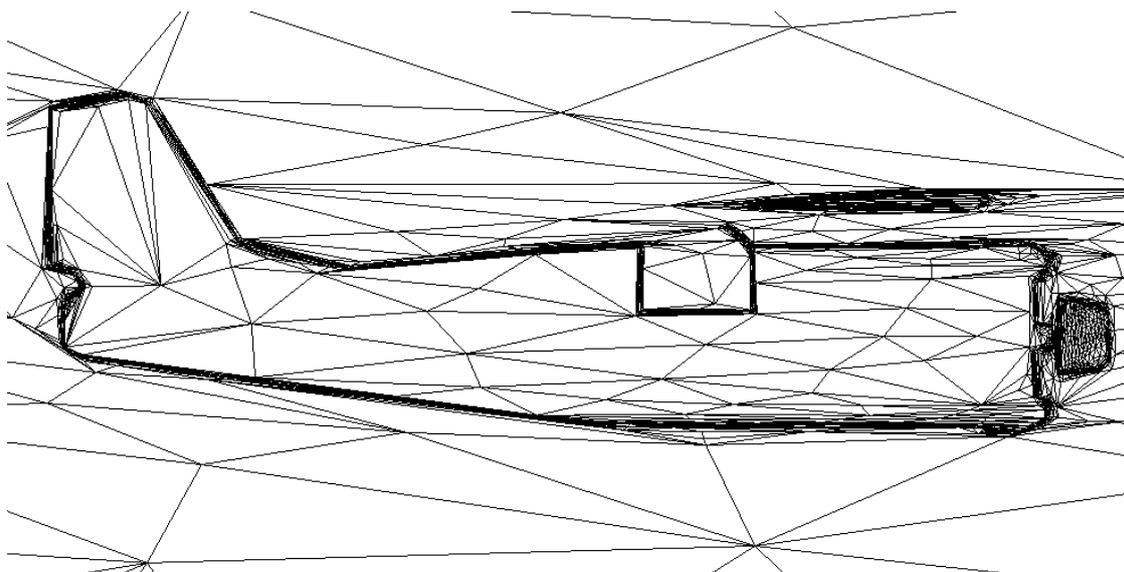
This noise can be reduced by cutting the partially filled elements, by the 50% levelset of  $g_\omega$  after a  $P^1$  averaging. Cut elements should then be triangulated, so as to get a simplicial mesh. It would lead to a smoother boundary (figure 7.24(b)), but element quality is sure to be deteriorated.



(a) global view

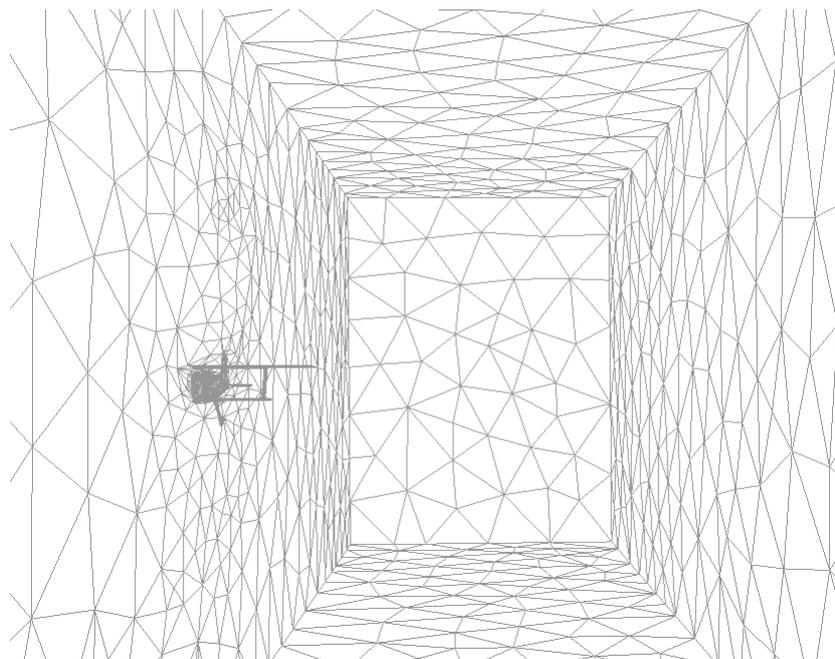


(b) enlargement of the symmetry plane

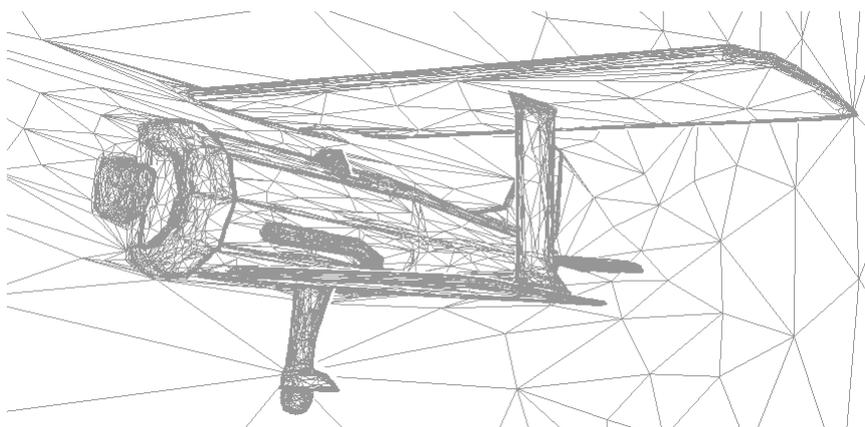


(c) enlargement of the biplane

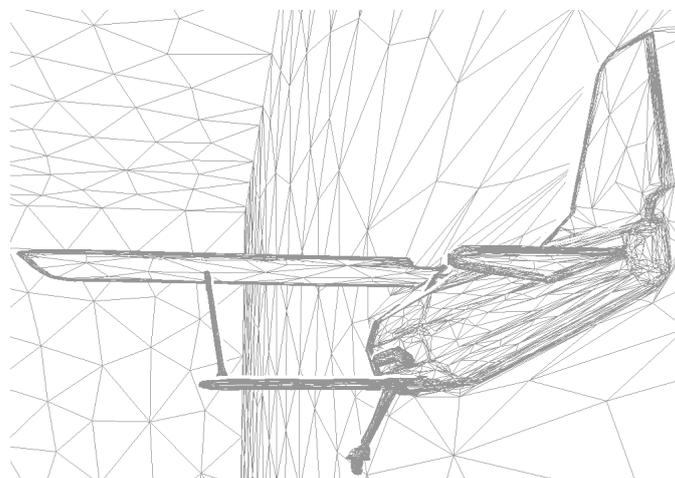
Figure 7.22: Bidomain mesh obtained for the biplane



(a) global view

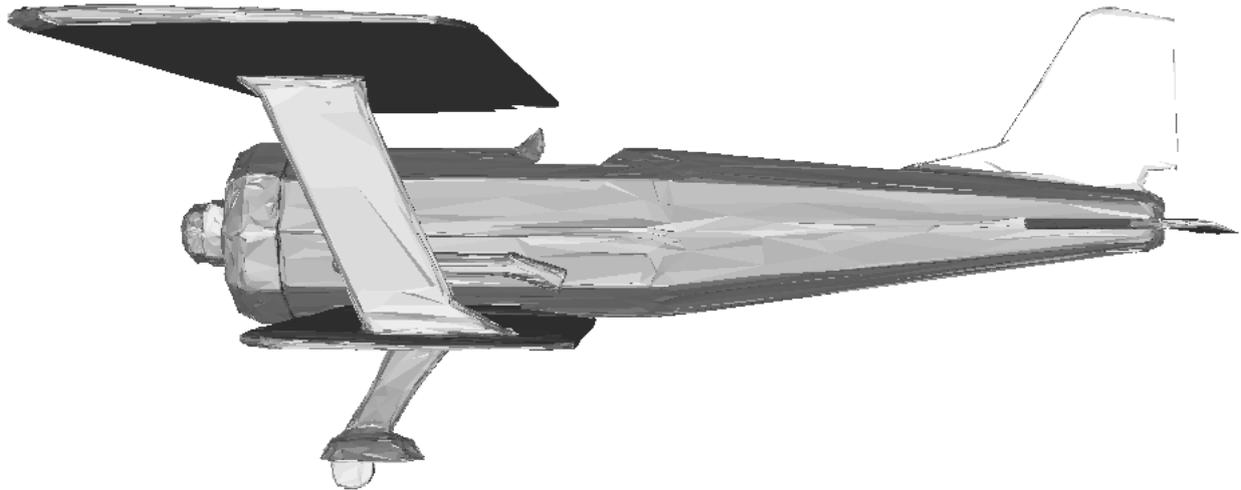


(b) enlargement of the biplane



(c) enlargement with another point of view

Figure 7.23: Boundary mesh of  $\Omega \setminus \omega$ , seen from the inside



(a) Extracted sub-mesh

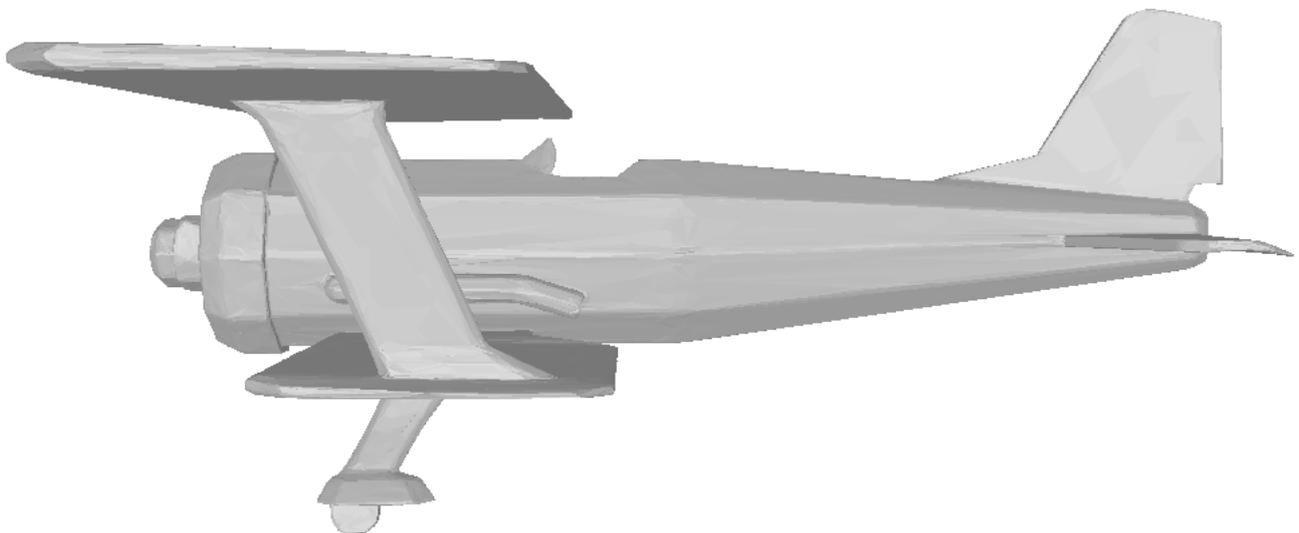
(b) 50% levelset of the filling ratio after a  $P^1$  averaging

Figure 7.24: A noisy boundary obtained by element extraction and a smoother levelset

Furthermore, if the voxelization of  $\omega$  relies only on a mesh of  $\omega$ 's boundary (currently, this is not the case, a mesh of  $\omega$  itself is required), then a new technique of mesh generation could be derived from the anisotropic adaption to a multidomain metric. Such a technique would present several advantages:

- the boundary  $\partial\omega$  could be connective or not (currently, the mesh generation by topological optimization needs the connectivity of the boundary mesh)
- a mesh of  $\Omega \setminus \omega$  could be built from separated CAD of  $\Omega$  and  $\omega$
- a partial mesh of  $\omega$  (a symmetry part for example) could be built by choosing  $\Omega$  adequately
- small details (under a certain mesh size threshold) of  $\omega$  could be removed
- depending on the voxelization algorithm, some errors in the mesh of  $\partial\omega$  could be repaired.

In fact, many CAD operations could be saved, with such a technique.



# Main conclusion



At the beginning of this work, in 2001, anisotropic meshes were very appealing for their computational time saving. However, they were merely used in 2D, because of several reasons:

- in 3D, adaption techniques were recent and not widespread
- in 3D, only analytical metrics or metrics defined by the anisotropy of already computed fields, were available, which was not convenient for all mesh requirements.

Nowadays, anisotropic mesh adaption in 3D is operational. In particular, thanks to the work presented here:

- anisotropic adaption by topological optimization has been established in all spatial dimensions (including 2D, 3D and 4D)
- automatic metric building, before finite element computation, has made Rem3D end users enthusiast about 3D anisotropic meshes.

After a summarize of the results exposed in this document (next section), the future works arising from these progresses are listed in the last section.

## Summarize

In the framework of injection molding simulation, industrial plastic parts are usually thin, with some massive zones. For that reason, a 3D simulation tool has been developed, so as to fill the gap of Hele-Shaw approaches. However, treating thin zones with a 3D solver requires the use of anisotropic meshes (isotropic ones would be too much expensive), with enough element layers through the thickness.

### Anisotropic adaption

So as to treat complex industrial parts, a robust mesh adaption method, driven by a metric field, has been established. This method is based on topological optimization, whose goal is to improve an anisotropic quality criterion.

The robustness of this anisotropic adaption comes from the optimization strategy adopted. On one hand, mesh operations are purely topological, so, they are less exposed to numerical roundoff errors than geometrical operations. On the other hand, the use of an anisotropic size criterion, in conjunction with an anisotropic shape criterion, and the use of an averaged mesh size (contrary to an element diameter) are important ingredients of the method efficiency.

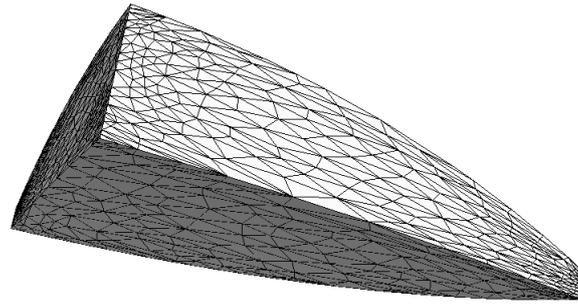
The overall algorithm gives good results, both in 2D and 3D, in what concerns the conformity between an adapted mesh and the metric field used in its adaption process.

### Natural and multidomain metrics

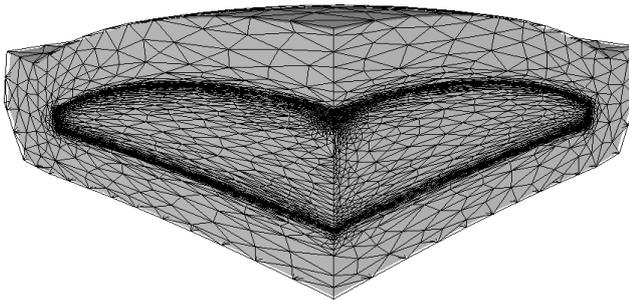
Efforts have then mainly been spent on the elaboration of an a priori metric field that is able, not only to introduce several element layers through the thickness for a thin and curved geometry (figure 7.25(a)), but also to refine a mesh around internal discontinuity surfaces, in the multidomain framework (figure 7.25(b)).

In order to introduce several element layers through the thickness of a given geometry, whatever its complexity may be, a natural metric field is proposed. The natural metric detects automatically the local anisotropy and the curvature of a geometry, by proceeding as follows on each node of its mesh:

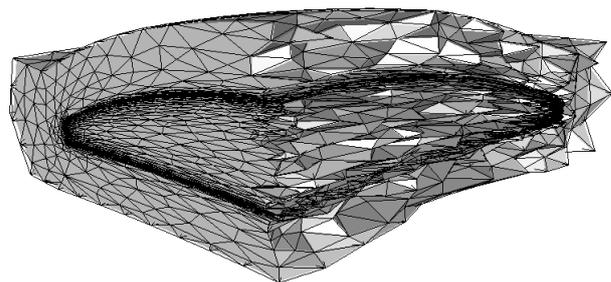
- aggregation of mesh elements around the node, so as to select the concerned zone



(a) natural mesh of the lens



(b) bidomain mesh of the lens and its mold



(c) sectional view of the bidomain mesh

Figure 7.25: Quarter of an optical lens and its surrounding mold (courtesy of Essilor)

- elliptic interpolation of this aggregation, via a new technique based on orientation tensors.

The obtained metric is then selectively divided, so as to impose a relevant mesh size through the thickness. The coupling between metric generation and mesh adaption is iterated and leads to a suitable mesh for the Rem3D computation (monodomain case).

Besides, for a simulation involving several bodies in interaction, the mesh must correctly represent the computational subdomains (multidomain case). For that purpose, the natural metric is enriched by an additional tensor dedicated to refining the mesh around interfaces between subdomains. This additional tensor is computed by:

- voxelization of each subdomain, so as to interpolate its characteristic function
- gradient computation of this interpolation and tensorial product, in order to obtain a tensor.

This multidomain metric leads to an adapted mesh that reduces deeply the diffusion of characteristic function interpolations. In fact, the adapted mesh is refined where strong gradients take place. Thus, the multidomain metric is useful for an Eulerian, a Lagrangian or even an ALE formulation.

This metric can also be employed as a new boundary layer treatment, which could help the natural metric around crossing zones or where the thickness changes suddenly. Another boundary layer treatment (with a prescribed thickness) can be derived by considering a lubricant subdomain, whose material behavior can model the macroscopic friction behavior.

As far as we know, the natural metric and the multidomain metric are thought to be new approaches. In the literature, only 2D metrics are computed a priori, for boundary layer treatment [Castro-Díaz *et al.* 1996, Castro-Díaz *et al.* 1997].

## A posteriori metrics

Unfortunately, a mesh generated by adaption to a priori metrics is not always suitable during all the simulation. Some developments have thus been devoted to a feasibility study of an a posteriori mesh adaption, still driven by a metric field.

The natural metric can be improved a posteriori, after a first computation, with a relevant field output. Such a simulation can be the solving of a stationary and linear Stokes problem, for a filled cavity, under an imposed entrance pressure and a zero exit pressure. Then, the anisotropy of a relevant scalar field output (the acceleration norm, for instance) is taken into account, not through its Hessian matrix but rather through its gradient vector, squared by a tensorial product.

Another way to improve the natural metric consists in smoothing an error indication on the mesh. Such a smoothing leads to a local multiplicative factor for the natural metric, so that the initial anisotropy is preserved, the mesh being locally coarsened or refined. Furthermore, an a posteriori error estimator on an anisotropic mesh, which is beyond the scope of this work, would help in the local anisotropy improvement.

## Industrial applications

In practice, this work has been implemented as a new pre-treatment tool, devoted to the metric generation for anisotropic mesh adaption. Major functionalities of this tool are already used in an industrial framework. Mesh generation for Rem3D is really simplified for the end users: the meshing stage is less time consuming (with respect to the equation solving stage) and multidomain tasks, like the crucial thermal coupling for injection molding, are easier to use.

## Mathematical contributions

From a mathematical point of view, some results have been established. The most important are:

- the minimal volume theorem, which is now proved without the orientation assumption (page 28)
- the elliptic interpolation raises a semi-definite programming problem that can be expressed by a Fourier series on the unit sphere (page 63)
- the Hessian matrix can be non elliptic, which could be advantageously replaced by the gradient operator, squared by a tensorial product (page 104).

Other minor results have been proved for all spatial dimensions, which enables the implementation for all spatial dimensions and its verification in 2D, 3D and 4D. 4D tests are thought to be new, in what concerns unstructured meshes.

Furthermore, some results rely on Riemannian geometry notions, employed in a discrete framework. However, this approach should be distinguished from the, so called, discrete Riemannian geometry [Dimakis and Müller-Hoissen 1999].

## Publications

Mathematical developments have been published in [Gruau and Coupez 2003] (without proofs), in [Gruau and Coupez 2005] (with proofs) and in [Gruau and Coupez 2004] (French translation). Rem3D applications have been evoked in [Silva *et al.* 2003] and further detailed in [Silva *et al.* 2004] and in [Coupez *et al.* 2004].

## Future work

In this section, a brief review of the remaining technical questions and research fields is drawn.

### Technical developments

The natural metric implementation is quite complex and a question arises of whether the results presented here would be reproducible within another framework (involving another mesh technology). The answer is not straightforward, since many parameters have been tuned, according to our specific framework.

Besides, parallelization of the metric computation is required, since meshes with 10 or 100 million nodes have to be treated on computer clusters. In what concerns the multidomain metric and a posteriori metrics, there is no theoretical limitations for the parallelization. However, in what concerns the natural metric, the local anisotropy detection around each node would lead to many expensive communications between mesh partitions.

Furthermore, the metric generation tool and the mesh generator (used in the adaption mode) could be more strongly coupled during the iterative anisotropic adaption process. It would solve the memory allocation problem for the future adapted mesh and a convergence criterion could be used, in order to automatically stop the process when a suitable mesh is attained.

### Natural metric

The natural metric computation is quite expensive, since all the work of local anisotropy detection is lost from one iteration to another. A good speedup could be reached by a metric transport technique from one mesh to another.

Besides, other techniques could be tested for the local anisotropy detection. In particular, the use of a signed distance to the boundary could give good results.

### Voxelization

Our uniform and isotropic voxelization technique is also quite expensive (in memory and in time). A tree approach could save a lot of memory and could enable a better accuracy (around the boundary, for example). Preliminary tests in 3D by [Henry 2004] (octree) show promising results in this field. But many implementation difficulties arise, especially when a spatial dimension independence is required.

Another expected improvement would be the possibility to perform the voxelization of a subdomain, only from a mesh of its boundary. After the boundary voxels are lit, other voxels could be scan so as to lighten those inside the region delimited by previously lit voxels. It would be time saving and would lead to a new mesh generation process for the subdomain (based on a bidomain anisotropic adaption in a bounding box).

Such a voxelization technique could also be used with an analytical description of the subdomain.

### Dynamic adaptivity

The work presented here is used to build a suitable mesh at the beginning of a simulation. However, remeshing can be compulsory, especially with a Lagrangian formulation in finite strains. The remeshing during the simulation needs a transport of the natural and multidomain metrics:

- from one deformed configuration to another: a tensorial transport solver is required (ALE formulation)
- and from one mesh to another: a conservative metric re-interpolation method is required.

## 4D meshes

Our implementation is dimension independent (including its use in 4D), but a 4D mesh asks some questions. Firstly, how could it be visualized? Animated cutting hyperplanes are a solution, but a pentatope cut by a 3D hyperplane gives numerous configurations.

Secondly, how the boundary mesh can be built? (such a boundary is composed of tetrahedra). With an Eulerian (or ALE) formulation, the boundary of the mesh at the initial time step can be extruded (in time), until the final time step. This mesh could then be completed by the 3D initial mesh and the 3D final mesh (usually, there are identical for Rem3D).

Once the boundary mesh is built, the topological mesh generator can produce a first 4D mesh. Furthermore, this mesh can be adapted, according to a 4D metric, so as to obtain a suitable mesh. The most critical stage in this process concerns the natural metric: the thickness size in the time direction (with is already known) can disturb the local anisotropy detection in the space directions. In fact, the computation of a 3D natural metric in a 4D mesh is required (this metric being completed in the 4-th dimension by a time mesh size, which can even be heterogeneous).

In what concerns the multidomain metric, its use in 4D is straightforward. Such a metric could delimitates a subdomain whose position evolution in time is known. This is the case, for instance, of an extrusion screw in its tube. The 4D mesh can then be automatically refined around moving interfaces.



# Appendices



# Appendix A

## Complements on meshes and metrics

Several properties about metrics, simplices and meshes, although useful for implementing the techniques described previously, are not required at the first reading. They are grouped in this appendix.

### A.1 Metric

In what concerns metrics, few properties have not been evoked yet. In particular, plotting the unit ball of a metric, obtaining a metric from a tensor  $x \otimes x$  and operating arithmetics on metrics.

#### A.1.1 Conic and parametrization

The unit ball of a metric  $M$  of  $\mathbb{R}^d$  (which is defined by: all points  $x \in \mathbb{R}^d$  such that  $\|x\|_M = 1$ ) is a non degenerated elliptic conic.

In 2D, the unit ball of a metric  $M$  is an ellipse, which can be plotted by considering the following parametrization

$$\theta \rightarrow M^{-1/2} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \quad \text{where } \theta \in ]-\pi; \pi] \quad (\text{A.1})$$

In 3D, the unit ball of a metric  $M$  is an ellipsoid, which can be plotted by considering the following parametrization

$$(\theta, \phi) \rightarrow M^{-1/2} \begin{pmatrix} \cos(\theta) \cos(\phi) \\ \sin(\theta) \cos(\phi) \\ \sin(\phi) \end{pmatrix} \quad \text{where } \theta \in ]-\pi; \pi] \quad \text{and } \phi \in ]-\pi/2; \pi/2] \quad (\text{A.2})$$

#### A.1.2 Tensorial product

So as to generate metrics, we intensively use tensors like  $x \otimes x$ . However, such a square matrix, although real and symmetrical, is not a metric.

**Proposition A.1** let  $d \geq 1$  and  $x \neq 0 \in \mathbb{R}^d$ , then eigenvalues of  $x \otimes x$  are 0, with multiplicity  $d-1$ , and  $\|x\|^2$ , whose eigenvector is  $x$ .

Thus,  $x \otimes x$  is not definite positive, but only semi-definite positive.

**Proof :** since  $x \otimes x$  is symmetrical, it can diagonalized. Firstly, let us prove that  $\|x\|^2$  is an eigenvalue of  $x \otimes x$ , with eigenvector  $x$ : we have  $x \otimes x = xx^\top$ , so,  $(x \otimes x)x = (xx^\top)x = x(x^\top x) = x\|x\|^2$ , since  $x^\top x = \|x\|^2$ .

Secondly, let  $\lambda \neq 0$  be an eigenvalue of  $x \otimes x$  and let us prove that  $\lambda = \|x\|^2$ : we denote by  $u \neq 0 \in \mathbb{R}^d$  a vector such that  $(x \otimes x)u = \lambda u$ ; since  $(x \otimes x)u = x(x^\top u)$ , it comes  $x(x^\top u) = \lambda u$  and since  $\lambda \neq 0$  and

$x^\top u$  is a scalar, we have  $x^\top u \neq 0$ . Thus,  $x$  and  $u$  are collinear. But,  $x$  and  $u$  are eigenvectors of  $x \otimes x$ , so, the only possibility for  $x$  and  $u$  to be collinear, is to be equal. In other words,  $\lambda = \|x\|_2^2$ . Thus, when  $d \geq 2$ , 0 is the only eigenvalue of  $x \otimes x$  distinct from  $\|x\|_2^2$ .

Thirdly, when  $d \geq 2$ , let us prove that eigenvectors of the eigenvalue 0 are orthogonal to  $x$ : let  $u \neq 0$  be such that  $x^\top u = 0$ , we have  $(x \otimes x)u = x(x^\top u) = 0$ , so,  $u$  is an eigenvector of 0. For dimensional reasons, this inclusion is enough to prove that the orthogonal space of  $x$  (which has a dimension of  $d-1$ ) is the space of all eigenvectors of 0.  $\square$

In fact, if we add a metric to the tensor  $x \otimes x$ , then we still have a metric. Furthermore, if we make a sum of tensors  $\sum_i x_i \otimes x_i$ , where  $d$  vectors  $x_i$  are linearly independent, then there is no more eigenvector for the eigenvalue 0, so,  $\sum_i x_i \otimes x_i$  is a metric.

### A.1.3 Operators between metrics

So as to illustrate arithmetical operation between two metrics, we consider the following metrics (there unit balls are in dotted lines on the following figures)

$$M_1 = R\left(-\frac{\pi}{32}\right) \begin{pmatrix} 1 & 0 \\ 0 & 1000 \end{pmatrix} R\left(-\frac{\pi}{32}\right)^\top \quad (\text{A.3})$$

$$\text{and } M_2 = R\left(\frac{\pi}{16}\right) \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1000 \end{pmatrix} R\left(\frac{\pi}{16}\right)^\top \quad (\text{A.4})$$

$$\text{where } R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (\text{A.5})$$

#### A.1.3.1 Several averaged values

When several metrics are given at the same point, we often need to compute one metric that has averaged characteristics of the initial metrics. Those initial metrics may have different weights, but this case has not been encountered, in this work. That is the reason why we focus our attention on non weighted averages. Formula are given for two metrics, the generalization to  $n$  metrics can be done straight away.

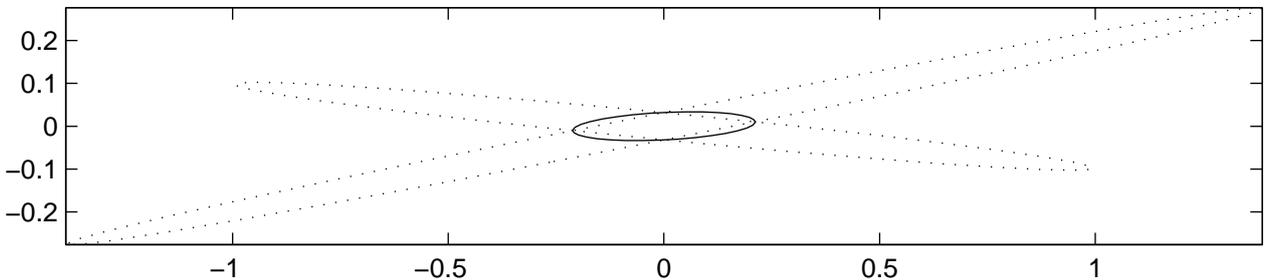


Figure A.1: Classical averages between two metrics

The averaged metric  $\frac{M_1 + M_2}{2}$  is easy to compute (no diagonalization, no matrix inversion). Unfortunately, this average between two metric loses the stretching of initial metrics (figure A.1).

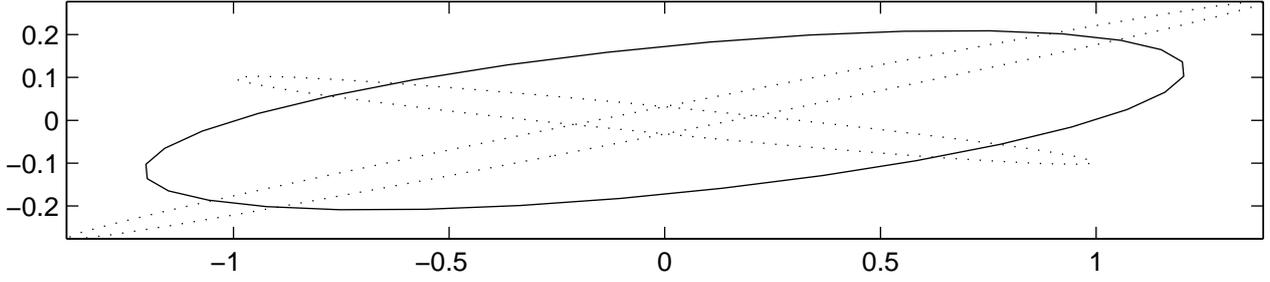
In fact, the unit ball of  $\frac{M_1 + M_2}{2}$  passes through the intersection points of of initial unit balls of  $M_1$

and  $M_2$ . Indeed, if we have  $\|x\|_{M_1} = 1 = \|x\|_{M_2}$ , then

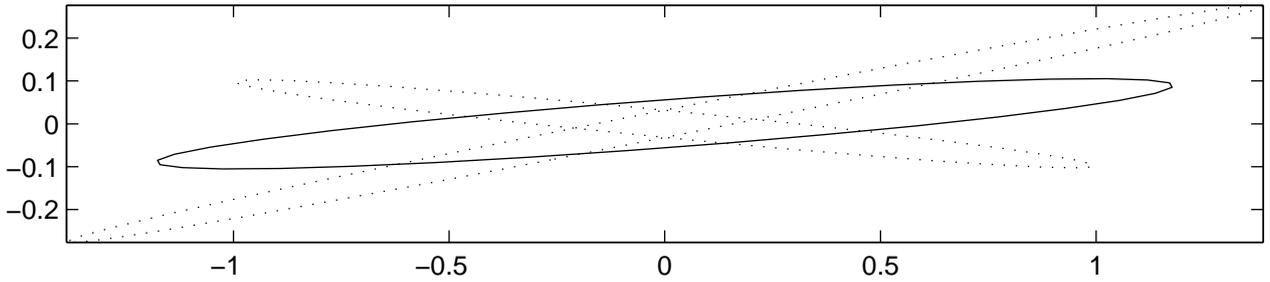
$$\|x\|_{\frac{M_1+M_2}{2}} = x^\top \left( \frac{M_1 + M_2}{2} \right) x = x^\top \left( \frac{M_1 x + M_2 x}{2} \right) \tag{A.6}$$

$$= \left( \frac{x^\top M_1 x + x^\top M_2 x}{2} \right) = \frac{\|x\|_{M_1} + \|x\|_{M_2}}{2} = 1 \tag{A.7}$$

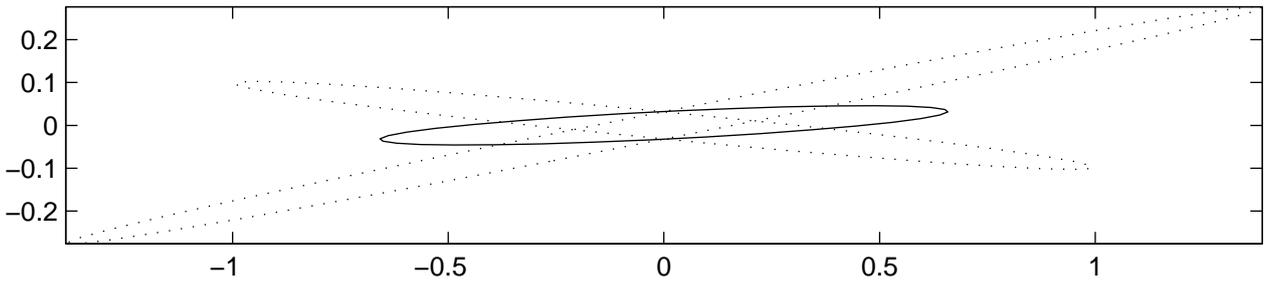
However, the topological mesh generator uses this operator between metrics, for convergence reasons (see section 3.2.2 page 42).



(a)  $p = -1$



(b)  $p = -1/2$



(c)  $p = 1/2$

Figure A.2:  $L^p$  average of two metrics

Among all averages defined by

$$\left( \frac{M_1^p + M_2^p}{2} \right)^{1/p} \tag{A.8}$$

where  $p \neq 0 \in \mathbb{R}$ , the particular values  $p = -1$ ,  $p = -1/2$  and  $p = 1/2$  can be tested (figure A.2).

The metric obtained with  $p = -1/2$  preserves quite well the stretching of initial metrics and its eigenvectors are in the middle of initial eigenvectors (figure A.2(b)). This is the most interesting average operator for interpolation. Furthermore,  $p = -1/2$  corresponds to the classical average operation between mesh sizes, since  $M^{-1/2}$  has the dimension of a length. Unfortunately, the computation of  $M^{-1/2}$  involves the diagonalization of  $M$ . Besides, numerical tests performed with the topological mesh generator (section 3.2.2 page 42) show surprisingly that  $p = -1/2$  give worse results than  $p = 1$ .

### A.1.3.2 Intersection

Sometimes, a metric with small mesh sizes is needed, so as to refine a mesh at locations where different metrics are provided. This is the case for the multidomain metric around triple points (section 6.2.3 page 106).

Intersection of two metrics has been investigated by [Castro-Díaz *et al.* 1996, Castro-Díaz *et al.* 1997, Mohammadi *et al.* 2000]. The (difficult) problem is to find the metric whose unit ball is the greatest one included in the unit ball of  $M_1 = V_1 D_1 V_1^\top$  (in this diagonalization, the diagonal  $D_1$  may not be composed of eigenvalues) and included in the unit ball of  $M_2 = V_2 D_2 V_2^\top$  (in this diagonalization, the diagonal  $D_2$  may not be composed of eigenvalues).

Simultaneous diagonalization consists in diagonalizing  $V_1^\top M_2 V_1 = V D_2 V^\top$  (then we have  $M_2 = V_1 V D_2 V^\top V_1^\top$ , that is,  $V_2 = V_1 V$ ) and we may have also  $M_1 = V_1 V D_1 V^\top V_1^\top$ . In this case, we could take  $M_1 \cap M_2 = V_1 V \max(D_1, D_2) V^\top V_1^\top$ .

An approximated intersection to obtain a metric whose unit ball is more or less included in those of  $M_1$  and those of  $M_2$  (figure A.3) can be computed by

$$M_1 \cap M_2 = \frac{V_1 \hat{D}_1 V_1^\top + V_2 \hat{D}_2 V_2^\top}{2} \quad (\text{A.9})$$

where  $\hat{D}_1$  is the diagonal matrix whose coefficients are

$$\hat{\lambda}_1^j = \max\left(\lambda_1^j, V_1^{j\top} M_2 V_1^j\right) \quad (\text{A.10})$$

and for  $\hat{D}_2$

$$\hat{\lambda}_2^j = \max\left(\lambda_2^j, V_2^{j\top} M_1 V_2^j\right) \quad (\text{A.11})$$

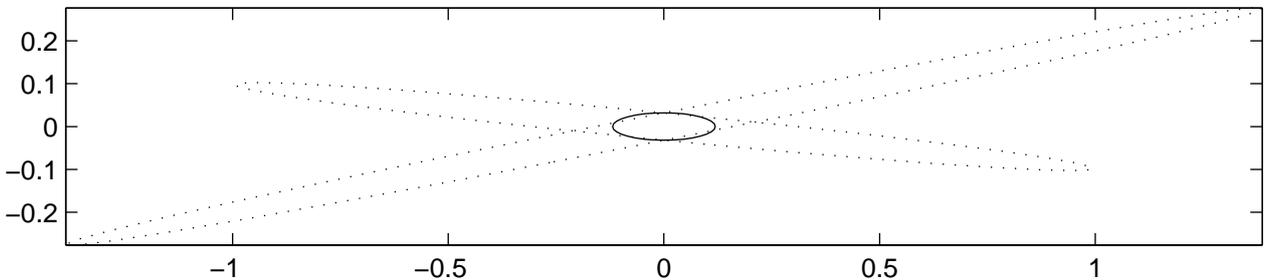


Figure A.3: Approximated intersection between two metrics

The metric (A.9) provides a fair approximation of the intersection. In practice, it can be used when the following relationship has to be enforced

$$\forall x \in \mathbb{R} \quad \|x\|_{M_1 \cap M_2} \geq \max(\|x\|_{M_1}, \|x\|_{M_2}) \quad (\text{A.12})$$

Unfortunately, this is not an associative operator (in other words,  $(M_1 \cap M_2) \cap M_3 \neq M_1 \cap (M_2 \cap M_3)$ ) and formula (A.9-A.11) are much more expensive than the following operator, which gives sufficiently good results, for our purpose.

The (coefficient by coefficient) maximal metric is defined by

$$M_{\max}^{ij} = \max(M_1^{ij}, M_2^{ij}) \quad (\text{A.13})$$

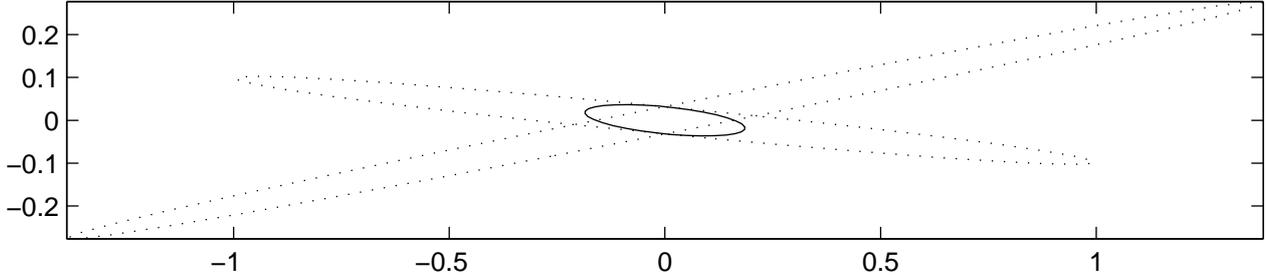


Figure A.4: Maximal metric

According to this operator, stretching of initial metrics is rapidly lost and the result tends to be isotropic (figure A.4). Since then, bad eigenvectors do not matter. In practice, this (mathematically bad) operator is very useful when an isotropic transition zone is needed between two contradictory metrics.

#### A.1.4 Analytical metrics

When the part geometry is simple, that is, Cartesian, cylindrical or spherical, then a continuous metric field can be computed analytically, so as to introduce several element layers through the thickness.

##### A.1.4.1 Cartesian metric

Let  $\Omega$  be an *orthogonal parallelootope* of  $\mathbb{R}^d$  (a rectangle is an orthogonal parallelootope of  $\mathbb{R}^2$  and a parallelepiped is an orthogonal parallelootope of  $\mathbb{R}^3$ ), which can be not aligned with Cartesian axes. In other words

$$\Omega = R(I_1 \times \cdots \times I_d) \quad (\text{A.14})$$

where  $R$  is a rotation matrix (whose columns are the axes of  $\Omega$ ),  $\times$  is the Cartesian product and  $(I_i)_{1 \leq i \leq d}$  is a set of segments of  $\mathbb{R}$  (closed and non degenerated).

So as to impose the mesh size  $h_i$  in the  $i$ -th axis of  $\Omega$ , the following constant metric is employed

$$M(x) = M = R \begin{pmatrix} \frac{1}{h_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{h_d^2} \end{pmatrix} R^\top \quad (\text{A.15})$$

This metric can be used to triangulate a plate with anisotropic elements (figure A.5). With such a Cartesian metric, the metric field is homogeneous, so, we are in the Euclidian framework.

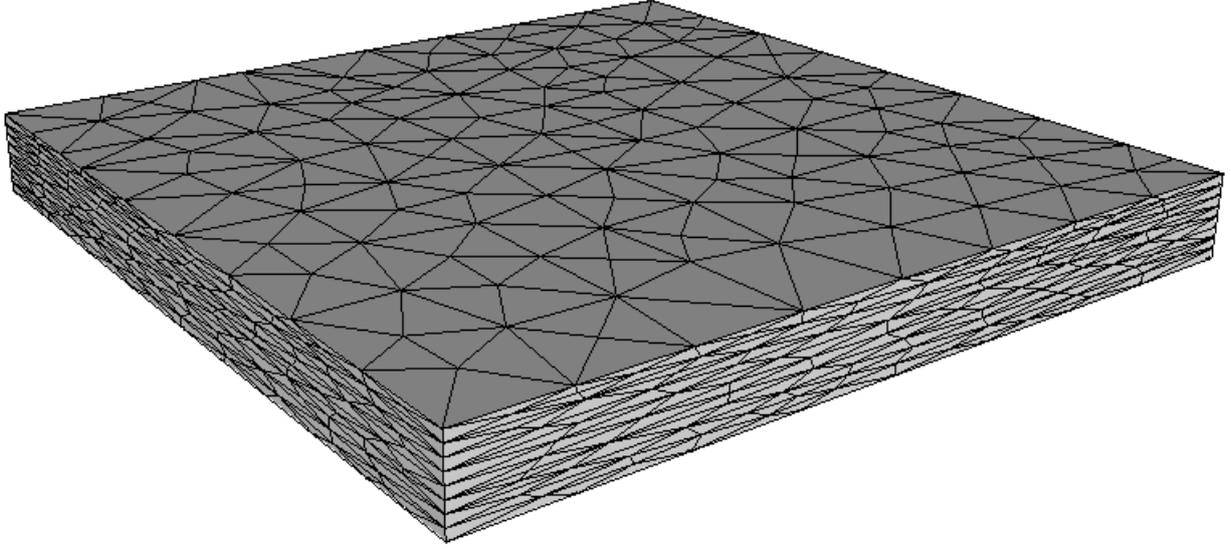


Figure A.5: Mesh of a plate with a Cartesian metric

#### A.1.4.2 Cylindrical metric

This section is restricted to the 3D case. Let  $\Omega$  be a cylindrical domain of  $\mathbb{R}^3$ , with axis  $(O, \vec{w})$ , where  $O$  has coordinates  $(x_0, y_0, z_0)$ . Let  $\vec{u}$  and  $\vec{v}$  be vector of  $\mathbb{R}^3$ , such that the base  $(\vec{u}, \vec{v}, \vec{w})$  is orthogonal, unitary and directly oriented (such a base exists). In this base, cylindrical coordinates  $(r, \theta, z)$  can be used. We denote by  $h_r, h_\theta$  and  $h_z$  the mesh sizes to be imposed, respectively in directions  $r, \theta$  and  $z$ .

Let  $R_1$  be the rotation whose columns are  $(\vec{u}, \vec{v}, \vec{w})$ . This rotation transforms the domain to a cylinder with axis  $(Oz)$ . At any point  $P \in \mathbb{R}^3$ , whose coordinates are  $(x, y, z)$ , is associated a point  $P' = R_1^\top(P - O)$ , with coordinates  $(x', y', z')$ . Eigenvectors of the desired metric in  $P'$  are  $(\vec{e}_r, \vec{e}_\theta, \vec{e}_z)$ , so, the cylindrical metric in  $P$  is

$$M(x, y, z) = R_1 R_2(x, y, z) \begin{pmatrix} \frac{1}{h_r^2} & 0 & 0 \\ 0 & \frac{1}{h_\theta^2} & 0 \\ 0 & 0 & \frac{1}{h_z^2} \end{pmatrix} R_2^\top(x, y, z) R_1^\top \quad (\text{A.16})$$

$$\text{where } R_2(x, y, z) = (\vec{e}_r \vec{e}_\theta \vec{e}_z) = \begin{pmatrix} \frac{x'}{\sqrt{x'^2 + y'^2}} & \frac{-y'}{\sqrt{x'^2 + y'^2}} & 0 \\ \frac{y'}{\sqrt{x'^2 + y'^2}} & \frac{x'}{\sqrt{x'^2 + y'^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.17})$$

$$\text{and } (x', y', z') = R_1^\top(x - x_0, y - y_0, z - z_0) \quad (\text{A.18})$$

This cylindrical metric makes it possible to triangulate et full cylinder or a thick tube by anisotropic elements (figure A.6). We remark that along axis  $(O, \vec{w})$ , the cylindrical metric is not defined. However, it can be extended by

$$M(x, y, z) = R_1 \begin{pmatrix} \frac{1}{h_r^2} & 0 & 0 \\ 0 & \frac{1}{h_r^2} & 0 \\ 0 & 0 & \frac{1}{h_z^2} \end{pmatrix} R_1^\top \quad (\text{A.19})$$

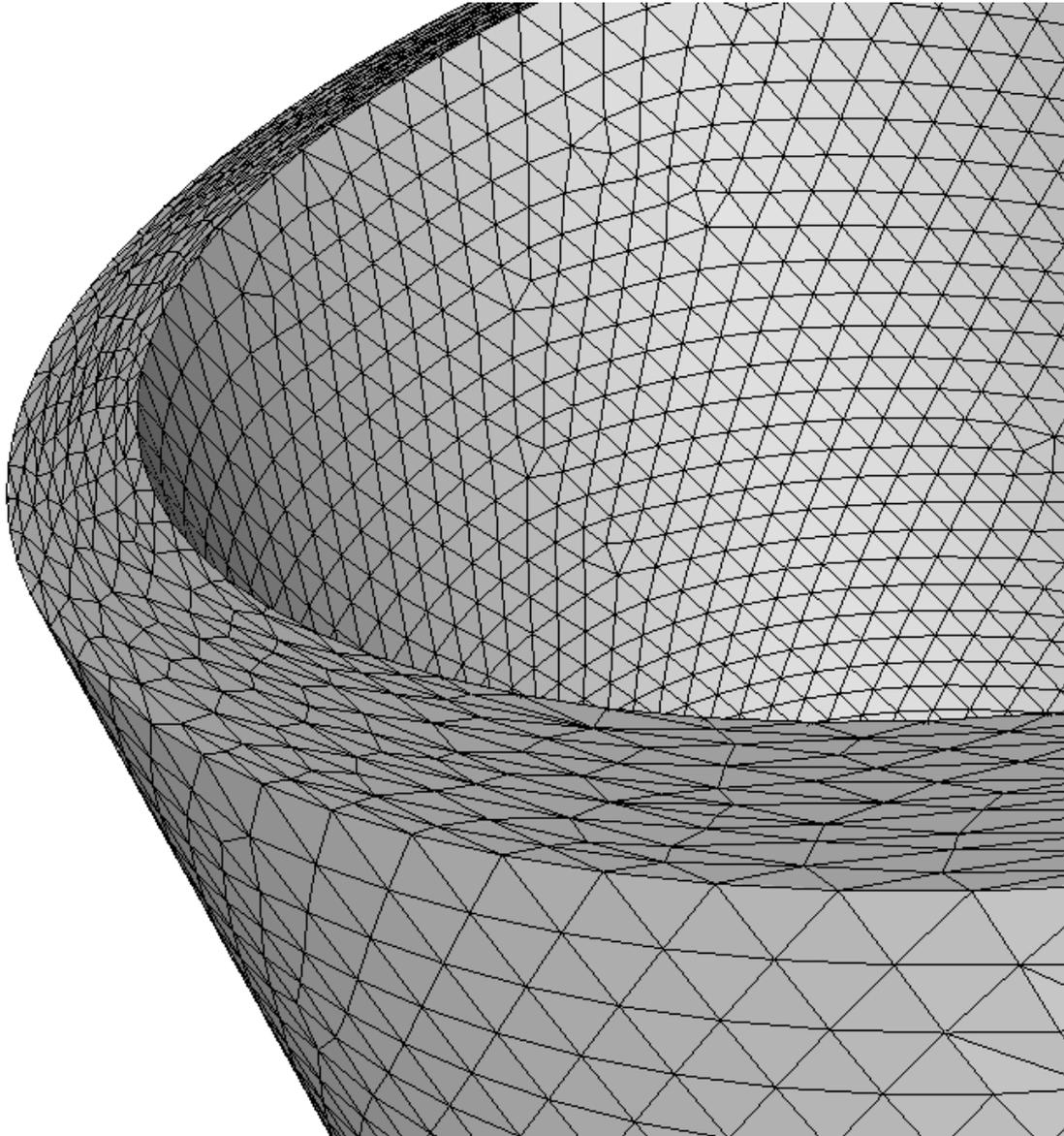


Figure A.6: Thick tube mesh obtained with a cylindrical metric

where we have  $\sqrt{x'^2 + y'^2} = 0$ . This cylindrical metric is heterogeneous and sufficiently regular to be Riemannian (except along the axis  $(O, \vec{w})$ ).

### A.1.4.3 Spherical metric

Still in the 3D case, we consider a spherical domain  $\Omega$  of  $\mathbb{R}^3$ , with center  $O$ , whose coordinates are  $(x_0, y_0, z_0)$ . Contrary to the cylindrical case, we can stay in the canonical axes (no preliminary rotation is required). We work in spherical coordinates  $(r, \theta, \phi)$ , with  $O$  as the origin (a translation is needed). We denote by  $h_r, h_\theta$  and  $h_\phi$  the mesh sizes to be imposed, respectively in the directions  $r, \theta$  and  $\phi$ .

Eigenvectors of the desired metric at a point  $(x, y, z)$  are  $(\vec{e}_r, \vec{e}_\theta, \vec{e}_\phi)$ , so, the spherical metric is

$$M(x, y, z) = R(x, y, z) \begin{pmatrix} \frac{1}{h_r^2} & 0 & 0 \\ 0 & \frac{1}{h_\theta^2} & 0 \\ 0 & 0 & \frac{1}{h_\phi^2} \end{pmatrix} R^\top(x, y, z) \quad (\text{A.20})$$

$$\text{where } R(x, y, z) = (\vec{e}_r \vec{e}_\theta \vec{e}_\phi) \quad (\text{A.21})$$

$$= \begin{pmatrix} \frac{x'}{\sqrt{x'^2 + y'^2 + z'^2}} & \frac{-y'}{\sqrt{x'^2 + y'^2}} & \frac{-x'z'}{\sqrt{x'^2 + y'^2}\sqrt{x'^2 + y'^2 + z'^2}} \\ \frac{y'}{\sqrt{x'^2 + y'^2 + z'^2}} & \frac{x'}{\sqrt{x'^2 + y'^2}} & \frac{-y'z'}{\sqrt{x'^2 + y'^2}\sqrt{x'^2 + y'^2 + z'^2}} \\ \frac{z'}{\sqrt{x'^2 + y'^2 + z'^2}} & 0 & \frac{\sqrt{x'^2 + y'^2}}{\sqrt{x'^2 + y'^2 + z'^2}} \end{pmatrix} \quad (\text{A.22})$$

$$\text{and } (x', y', z') = (x - x_0, y - y_0, z - z_0) \quad (\text{A.23})$$

This spherical metric can drive the triangulation of any spherical part (figure A.7). We remark that at point  $O$ , the spherical metric is not defined. However, it can be easily extended by

$$M(O) = \begin{pmatrix} \frac{1}{h_r^2} & 0 & 0 \\ 0 & \frac{1}{h_r^2} & 0 \\ 0 & 0 & \frac{1}{h_r^2} \end{pmatrix} \quad (\text{A.24})$$

In 2D, we can defined the polar metric

$$M(x, y) = R(x, y) \begin{pmatrix} \frac{1}{h_r^2} & 0 \\ 0 & \frac{1}{h_\theta^2} \end{pmatrix} R^\top(x, y) \quad (\text{A.25})$$

$$\text{where } R(x, y) = (\vec{e}_r, \vec{e}_\theta) = \begin{pmatrix} \frac{x'}{\sqrt{x'^2 + y'^2}} & \frac{-y'}{\sqrt{x'^2 + y'^2}} \\ \frac{y'}{\sqrt{x'^2 + y'^2}} & \frac{x'}{\sqrt{x'^2 + y'^2}} \end{pmatrix} \quad (\text{A.26})$$

$$\text{and } (x', y') = (x - x_0, y - y_0) \quad (\text{A.27})$$

with is extended in the same way at point  $O$ . Again, the spherical metric and the polar metric are heterogeneous and sufficiently regular to be Riemannian (except at point  $O$ ).

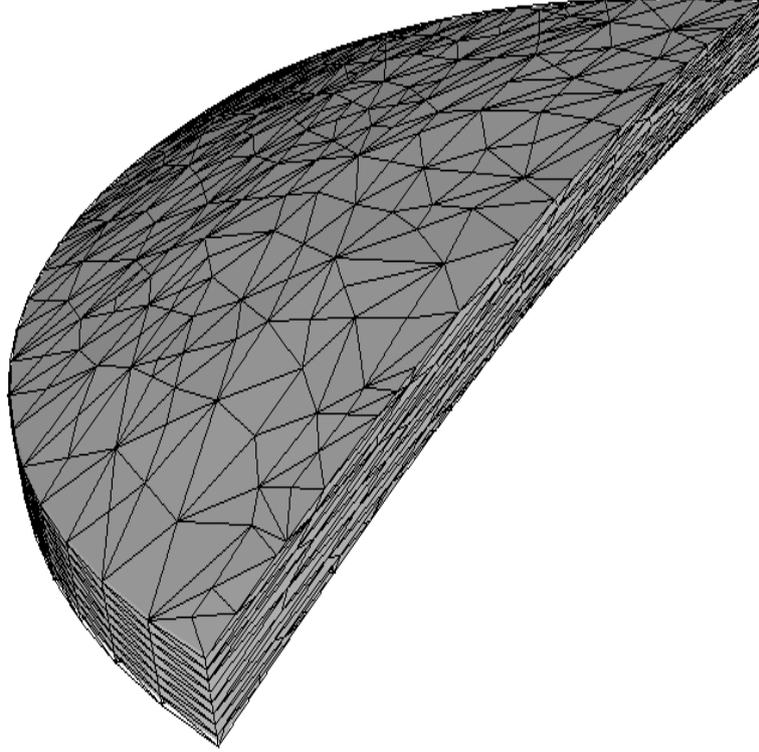


Figure A.7: Mesh of a lens with a spherical metric (courtesy of Essilor)

## A.2 Simplex

In what concerns a  $d$ -simplex in  $\mathbb{R}^d$  ( $k = d$ ), we need to compute its outgoing normals from the coordinates of its vertices. For that purpose, a study on the effect of a circular vertex permutation on orientation is necessary. Besides, we need an algorithm to construct an equilateral  $d$ -simplex, whatever dimension  $d$ .

### A.2.1 Orientation

**Definition A.1** The numbering  $S_0, \dots, S_d$  of a  $d$ -simplex vertices is said to be *well oriented* when

$$\det(S_0S_1, \dots, S_0S_d) \geq 0 \tag{A.28}$$

and ill oriented, otherwise.

Let us now have a look on the effect of a circular vertex permutation on the orientation.

**Proposition A.2** let  $T$  be a non degenerated  $d$ -simplex of  $\mathbb{R}^d$  (where  $d \geq 2$ ) and let  $S_0, \dots, S_d$  be a well oriented numbering of its vertices. A new numbering  $S_i, \dots, S_d, S_0, \dots, S_{i-1}$  obtained by a circular permutation is

- always well oriented, when  $d$  is even
- alternatively well and ill oriented, when  $d$  is odd.

because

$$\det(S_iS_{i+1}, \dots, S_iS_d, S_iS_0, \dots, S_iS_{i-1}) = (-1)^{d*i} \det(S_0S_1, \dots, S_0S_d) \tag{A.29}$$

In other words, in 2D and in 4D, a circular permutation on vertices has no effect on orientation. Conversely, in 3D, the orientation changes after each single ( $i = 1$ ) circular permutation on vertices.



**Proposition A.3** let  $T$  be a non degenerated  $d$ -simplex of  $\mathbb{R}^d$  (where  $d \geq 2$ ) and let  $S_0, \dots, S_d$  be a well oriented numbering of its vertices. Let  $F$  be a face of  $T$  and let  $i$  be such that  $F$  is the opposite face of vertex  $S_i$ . Then  $S_{i+1}, \dots, S_d, S_0, \dots, S_{i-1}$  is another numbering of vertices of  $F$  and the outgoing normal of  $F$  for  $T$  is

$$\vec{n} = \frac{(-1)^{d*i}}{|F|} S_{i+1}S_{i+2} \wedge \dots \wedge S_{i+1}S_d \wedge S_{i+1}S_0 \wedge \dots \wedge S_{i+1}S_{i-1} \quad (\text{A.33})$$

In other words, the outgoing normal computation can be achieved by vectorial products and face measurement. In addition, when  $d$  and  $i$  are odd, the opposite result has to be considered.

**Proof :** firstly, using the vectorial product definition, we know that the following vector

$$S_{i+1}S_{i+2} \wedge \dots \wedge S_{i+1}S_d \wedge S_{i+1}S_0 \wedge \dots \wedge S_{i+1}S_{i-1} \quad (\text{A.34})$$

is orthogonal to all vectors  $S_{i+1}S_{i+2}, \dots, S_{i+1}S_d, S_{i+1}S_0, \dots, S_{i+1}S_{i-1}$  (so, this vector is orthogonal to  $F$ ). Besides, property 1.2 page 13 says that  $|F|$  is the norm of this vector. So,

$$\frac{1}{|F|} S_{i+1}S_{i+2} \wedge \dots \wedge S_{i+1}S_d \wedge S_{i+1}S_0 \wedge \dots \wedge S_{i+1}S_{i-1} \quad (\text{A.35})$$

is a unitary normal vector.

Secondly, the vectorial product has the following property

$$\det \left( \begin{array}{c} S_{i+1}S_{i+2}, \dots, S_{i+1}S_d, S_{i+1}S_0, \dots, S_{i+1}S_{i-1}, \\ S_{i+1}S_{i+2} \wedge \dots \wedge S_{i+1}S_d \wedge S_{i+1}S_0 \wedge \dots \wedge S_{i+1}S_{i-1} \end{array} \right) \geq 0 \quad (\text{A.36})$$

in other words, when the numbering  $S_{i+1}, \dots, S_d, S_0, \dots, S_i$  of vertices of  $T$  is well oriented, the following normal

$$\frac{1}{|F|} S_{i+1}S_{i+2} \wedge \dots \wedge S_{i+1}S_d \wedge S_{i+1}S_0 \wedge \dots \wedge S_{i+1}S_{i-1} \quad (\text{A.37})$$

is ingoing. Using formulae (A.29), and multiplying this normal by  $(-1)^{d*i}$  is enough to always get an outgoing normal.  $\square$

### A.2.3 Equilateral simplex

Before we describe an algorithm to built an equilateral  $d$ -simplex of  $\mathbb{R}^d$ , we need the following result about its outer radius and is height.

**Proposition A.4** the outer radius of an equilateral  $d$ -simplex whose size is  $h$  and its height are

$$\text{radius}(d, h) = \sqrt{\frac{d}{2(d+1)}} h \quad (\text{A.38})$$

$$\text{height}(d, h) = \sqrt{\frac{d+1}{2d}} h \quad (\text{A.39})$$

For construction of an equilateral simplex, only (A.39) is needed. However, the proof of (A.39) requires (A.38).

**Proof :** let  $S_0 \dots S_d$  be an equilateral simplex with size  $h$ . We denote by  $G$  its center and by  $H$  the center of the face  $S_1 \dots S_d$  (figure A.9).

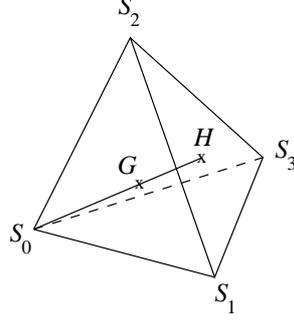


Figure A.9: Equilateral tetrahedron ( $d = 3$ )

On one hand, we have

$$GS_0 + \dots + GS_d = \vec{0} \quad (\text{A.40})$$

on the other hand, we have

$$GS_1 + \dots + GS_d = d(GH) \quad (\text{A.41})$$

so

$$GS_0 + d(GH) = \vec{0} \quad (\text{A.42})$$

which can be re-written as

$$(d+1)(GS_0) = d(HS_0) \quad (\text{A.43})$$

thus, we obtain

$$\|GS_0\| = \frac{d}{d+1} \|HS_0\| \quad (\text{A.44})$$

Since  $\|GS_0\|$  is the outer radius and  $\|HS_0\|$  is the height, it comes

$$\text{radius}(d, h) = \frac{d}{d+1} \text{height}(d, h) \quad (\text{A.45})$$

We still have to prove recursively on  $d > 0$  that

$$\text{height}(d, h) = \sqrt{\frac{d+1}{2d}} h \quad (\text{A.46})$$

- for  $d = 1$ , we have indeed  $\text{height}(1, h) = h$
- for  $d > 1$ , in the triangle  $S_0S_1H$  that is orthogonal on  $H$  we have

$$S_0H^2 + S_1H^2 = S_0S_1^2 \quad (\text{A.47})$$

in other words, we have

$$\text{height}(d, h)^2 + \text{radius}(d-1, h)^2 = h^2 \quad (\text{A.48})$$

because the face  $S_1 \dots S_d$  is an equilateral  $(d-1)$ -simplex with size  $h$  and with center  $H$ ; according to (A.45), we obtain

$$\text{height}(d, h)^2 + \left(\frac{d-1}{d}\right)^2 \text{height}(d-1, h)^2 = h^2 \tag{A.49}$$

By assumption, we have

$$\text{height}(d-1, h)^2 = \frac{d}{2(d-1)} h^2 \tag{A.50}$$

So, it comes

$$\text{height}(d, h)^2 = h^2 \left(1 - \left(\frac{d-1}{d}\right)^2 \frac{d}{2(d-1)}\right) \tag{A.51}$$

$$= h^2 \left(1 - \frac{d-1}{2d}\right) \tag{A.52}$$

$$= h^2 \left(\frac{d+1}{2d}\right) \tag{A.53}$$

Conclusion: (A.46) is true for all  $d > 0$ . □

We are now able to construct recursively an equilateral  $d$ -simplex of  $\mathbb{R}^d$  with size  $h$ , denoted by  $T^0$ . The vertices of  $T^0$  are:

- if  $d = 0$ , then  $S_0 = O$ , the origin
- if  $d > 0$ , then:
  - we build an equilateral  $(d-1)$ -simplex  $S_0 \dots S_{d-1}$ , with size  $h$  in the hyperplane  $x_d = 0$
  - then, the  $d$  coordinates of  $S_d$  corresponds to those of the center of  $S_0, \dots, S_{d-1}$  in what concerns the first  $(d-1)$  coordinates and by

$$\text{height}(d, h) = \sqrt{\frac{d+1}{2d}} h \tag{A.54}$$

for the last coordinate.

Next, so as to normalize the shape criterion (3.1) page 39, the measure of a equilateral simplex is required.

**Proposition A.5** the measure of an equilateral  $d$ -simplex, with size  $h$ , is

$$\text{measure}(d, h) = \frac{\sqrt{d+1}}{d! 2^{d/2}} h^d \tag{A.55}$$

**Proof :** thanks to the previous construction algorithm, we can build  $T^0$  an equilateral  $d$ -simplex with size  $h$ , whose first vertex is the origin  $O$  and whose edge vectors from  $O$  compose an upper triangular matrix. The  $i$ -th diagonal coefficient of this matrix is

$$\sqrt{\frac{i+1}{2i}} h \tag{A.56}$$

Since the determinant of this upper triangular matrix is equal to the product of its diagonal coefficients, it comes, with property 1.2 page 13, that the measure of  $T^0$  is

$$\text{measure}(d, h) = \frac{1}{d!} \prod_{i=1}^d \left( \sqrt{\frac{i+1}{2i}} h \right) \quad (\text{A.57})$$

$$= \frac{1}{d! \sqrt{2^d}} \prod_{i=1}^d \left( \sqrt{\frac{i+1}{i}} \right) h^d \quad (\text{A.58})$$

$$(\text{A.59})$$

and we remark that

$$\prod_{i=1}^d \left( \sqrt{\frac{i+1}{i}} \right) = \sqrt{d+1} \quad (\text{A.60})$$

which leads to (A.55). □

Since the shape criterion (3.1) must be equal to 1 for any equilateral simplex, the normalization factor  $c_0$  (3.5) is

$$c_0 = \text{measure}(d, 1)^{-1} = \frac{d!}{\sqrt{d+1}} 2^{d/2} \quad (\text{A.61})$$

## A.3 Mesh

In what concerns meshes, major properties have already been enunciated in chapter 1. However, two essential algorithms are depicted in this section. We consider a simplicial mesh  $(\mathcal{T}, \mathcal{S})$ , whose topological dimension is  $D$  (that is to say that each element has  $D$  vertices) and we assume that boundary faces are connected to a fictitious node 0 (see page 33).

### A.3.1 Reverse topology

An indispensable information to know about a mesh is its reverse topology, which is, for each node  $S$  of the mesh, the set of elements whose  $S$  is a vertex.

**Algorithm A.1** an efficient way to compute the reverse topology is as follows:

- to allocate an array  $\mathcal{T}'$ , with the same size of the array  $\mathcal{T}$  (the latter describing the vertices of each elements)
- to scan a first time the array  $\mathcal{T}$ , so as to increment, for each node  $S$ , the number of elements, whose  $S$  is a vertex
- to space out the array  $\mathcal{T}'$  between all nodes (by pointers), so that each node has enough space for its elements
- to scan again  $\mathcal{T}$ , so as to fill  $\mathcal{T}'$  like that: for each element  $T$ , add the numbering of  $T$  in the array  $\mathcal{T}'$  for each vertex of  $T$ .

The array  $\mathcal{T}'$  is accompanied by two other arrays:

- $\mathcal{T}''$  indicating for each node the first empty space in  $\mathcal{T}'$  ready for that node
- and  $\mathcal{T}'''$  specifying for each node the number of its elements.

In fact, the array  $\mathcal{T}'''$  is not necessary, if we fill  $\mathcal{T}'$  like that:

- each element appears  $D$  times in  $\mathcal{T}'$ , so, we could decide that the  $i$ -th entry in  $\mathcal{T}'$  is tacitely attributed to element  $i/D$  (integer division)
- the we could fill  $\mathcal{T}'$  by a compact chained list for each node describing successively the elements of that node.

The array  $\mathcal{T}''$  is still useful, so as to know where each chain list starts.

### A.3.2 Neighborhood

When the reverse topology is known, the neighborhood can be easily computed. The neighborhood is the description, for each node  $S$ , of the other nodes connected to  $S$  by an edge.

**Algorithm A.2** an acceptable way to compute the neighborhood is as follows:

```

for each node  $S$  do
  – allocate an empty node set  $\mathcal{V}(S)$  (whose size is unknown)
  for each element whose  $S$  is a vertex (thanks to the reverse topology) do
    – add to  $\mathcal{V}(S)$  the other vertices of this element (except node 0 and node  $S$  itself),
      without considering if those vertices are already present in  $\mathcal{V}(S)$ 
  done
  – sort  $\mathcal{V}(S)$  and remove multiple items, so as to compact  $\mathcal{V}(S)$ 
done

```

Again two additional arrays are needed to access  $\mathcal{V}(S)$  and to know the size of  $\mathcal{V}(S)$ .

### A.3.3 4D mesh

As announced in section 2.3 page 34, the topological mesh generator has been tested successfully in 4D. Contrary to [Takahashi *et al.* 2001], this is not a tetrahedral morphing test in 4D, but instead, the generated mesh is directly composed of pentatopes (simplices with 5 vertices) in 4D. Unfortunately, no visualization tool is available, at that time, for such 4D meshes. That is the reason why we reproduce the node coordinates and the element vertices in two tables.

The geometry for this test case is simple: the unitary hypercube  $[0, 1]^4$  of  $\mathbb{R}^4$ . The mesh of its boundary is given table A.1, in the first columns on the left. This boundary mesh is composed of 48 tetrahedrons in  $\mathbb{R}^4$ .

The initial 4D mesh generated by topological optimization is given on table A.1, in the last columns on the right. This mesh is composed of 36 pentatopes in  $\mathbb{R}^4$  (virtual elements connected to node 0 have been omitted).

Then, the topological mesh generator has been asked to adapt the initial mesh to a uniform mesh size of  $h = 1/3$ . The obtained result is given on table A.2 (organized in two columns), in what concerns the node coordinates and on the (long) table A.3 (organized in four columns), for the element vertices.

boundary mesh of the hypercube					mesh of the hypercube													
coordinates				vertices				coordinates				vertices						
n <sup>o</sup>	x	y	z	t	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	n <sup>o</sup>	x	y	z	t	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
1	0	0	0	0	8	4	2	1	1	0	0	0	0	12	16	10	9	8
2	1	0	0	0	8	2	6	1	2	1	0	0	0	16	14	10	9	8
3	0	1	0	0	8	3	4	1	3	0	1	0	0	16	12	11	9	8
4	1	1	0	0	8	7	3	1	4	1	1	0	0	15	16	11	9	8
5	0	0	1	0	8	6	5	1	5	0	0	1	0	14	16	13	9	8
6	1	0	1	0	8	5	7	1	6	1	0	1	0	16	15	13	9	8
7	0	1	1	0	16	12	10	9	7	0	1	1	0	1	14	10	2	8
8	1	1	1	0	16	10	14	9	8	1	1	1	0	14	1	6	2	8
9	0	0	0	1	16	11	12	9	9	0	0	0	1	14	1	10	9	8
10	1	0	0	1	16	15	11	9	10	1	0	0	1	1	14	13	9	8
11	0	1	0	1	16	14	13	9	11	0	1	0	1	1	14	6	5	8
12	1	1	0	1	16	13	15	9	12	1	1	0	1	14	1	13	5	8
13	0	0	1	1	14	10	2	1	13	0	0	1	1	16	12	4	3	8
14	1	0	1	1	14	2	6	1	14	1	0	1	1	12	16	11	3	8
15	0	1	1	1	14	9	10	1	15	0	1	1	1	16	15	11	3	8
16	1	1	1	1	14	13	9	1	16	1	1	1	1	15	16	7	3	8
					14	6	5	1						1	12	4	2	8
					14	5	13	1						12	1	10	2	8
					16	12	4	3						12	1	4	3	8
					16	4	8	3						1	12	11	3	8
					16	11	12	3						1	12	10	9	8
					16	15	11	3						12	1	11	9	8
					16	8	7	3						14	16	6	5	8
					16	7	15	3						16	15	7	5	8
					12	4	2	1						16	14	13	5	8
					12	2	10	1						15	16	13	5	8
					12	3	4	1						1	15	11	9	8
					12	11	3	1						15	1	13	9	8
					12	10	9	1						15	1	11	3	8
					12	9	11	1						1	15	7	3	8
					16	8	6	5						1	15	13	5	8
					16	6	14	5						15	1	7	5	8
					16	7	8	5						16	12	10	2	8
					16	15	7	5						14	16	10	2	8
					16	14	13	5						12	16	4	2	8
					16	13	15	5						16	14	6	2	8
					15	11	9	1										
					15	9	13	1										
					15	3	11	1										
					15	7	3	1										
					15	13	5	1										
					15	5	7	1										
					16	12	10	2										
					16	10	14	2										
					16	4	12	2										
					16	8	4	2										
					16	14	6	2										
					16	6	8	2										

Table A.1: Mesh of the boundary (left columns) and the interior mesh (right columns) of a 4D hypercube

node coordinates					node coordinates (continued)				
n°	x	y	z	t	n°	x	y	z	t
1	0	0	0	0	64	0.238412764	0.747588156	0.572296788	0
2	1	0	0	0	65	0	0.5	0	1
3	0	1	0	0	66	1	0	0.5	1
4	1	1	0	0	67	1	0.5	0.5	1
5	0	0	1	0	68	0.5	0	0	0.75
6	1	0	1	0	69	0.749794363	0.727129555	0.290628173	0
7	0	1	1	0	70	0.309292119	0.257152317	0	0.2852347
8	1	1	1	0	71	0	1	0.5	1
9	0	0	0	1	72	1	0	0.5	0.5
10	1	0	0	1	73	0.541206858	0.516602146	0.531204268	0.467611401
11	0	1	0	1	74	0.5	1	0.5	1
12	1	1	0	1	75	1	0.25	1	0.25
13	0	0	1	1	76	0.760147031	0.699239584	0.799481968	0.620498157
14	1	0	1	1	77	0.697471817	0.271453177	0.712617094	1
15	0	1	1	1	78	0.76162513	0.319574892	0	0.72646316
16	1	1	1	1	79	0.713045567	0.459806611	0.327487295	1
17	0	0.5	0.5	0	80	1	1	0.5	1
18	0.750276169	0.297773469	0.73043163	0	81	1	0	0	0.5
19	0.28867662	0.712699919	0.580010421	0.433687703	82	0.5	0.5	1	1
20	0	0	0.5	0.5	83	0.5	0	1	1
21	0	0.5	0	0	84	0	0.5	1	1
22	0	0.5	0	0.5	85	0.223439868	0.239374171	0.792222364	0.498672813
23	0	0	0	0.5	86	1	0.701142063	0.375381304	0.218020022
24	0.5	0.5	0	0	87	0	0.231033385	0.233701762	0.695096453
25	0.5	0	0	0	88	0.740625	0.73515625	0.7125	0
26	0.5	0	0.5	0.5	89	0.5	1	1	1
27	0.5	0	0.5	0	90	0	0.707310606	0.80969697	0.454924242
28	0.739581304	0.313101613	0.657165844	0.397729472	91	0.510161497	0.426551756	0.387094158	0
29	1	0.301948958	0.306551199	0.296455287	92	0.5436897	0.537755857	0	0.433347673
30	0	1	0	0.5	93	0	1	1	0.5
31	0.719930251	0.216597401	0.255507862	0.475507326	94	1	0.701553669	0.264718734	0.757980553
32	0.5	0.875	0	0	95	1	0.5	1	0
33	0.629234088	0.292072788	0.703869024	0.748016775	96	0	0.268803177	0.225587612	0.330658079
34	0.250553316	0.752373755	0.284648284	1	97	0	0.75	1	0.5
35	1	0.5	0	0.5	98	1	0.5	0.5	0
36	0.746227674	0	0.704150129	0.292226185	99	1	0.5	0	0
37	0	1	0.5	0	100	0.56	0.22	1	0.34
38	0.418643219	0.44113343	1	0.55368645	101	0.5	1	0.5	0
39	1	0.549168841	0.608694759	0.595066241	102	1	0.5	0	1
40	0	0.75	0	0.25	103	0.5	0.5	1	0
41	0.5	1	0	1	104	0.25	1	0.75	0
42	1	0	0.5	0	105	0	0	1	0.5
43	0.324619951	0.798970558	0	0.497187136	106	0	0.5	0.75	1
44	0	1	0.5	0.5	107	0.260955901	0.460498534	1	0.258938731
45	0.570364201	0.652214425	0.609589062	0.796537949	108	0.610945279	1	0.418992519	0.639935385
46	1	0.307953573	0.269359105	0.714330537	109	0	0.720731061	0.755969697	0.795492424
47	0.5	1	1	0.5	110	0	0.5	1	0
48	0.702845919	0	0.301876606	0.277827313	111	1	1	0	0.5
49	1	1	1	0.5	112	0.280925028	0.253614754	0.768035168	1
50	1	0.5	1	0.5	113	0.5	0	1	0
51	0.25	0.473281831	0.773250717	0.801676322	114	0.5	0	1	0.5
52	1	1	0.5	0	115	0.300653265	0.68790133	0.300966502	0.352410691
53	0.25	0	0.722	0.778	116	0	0	0.5	0
54	1	1	0.5	0.5	117	1	0.350256	0.717008	0.284272
55	0.505975369	0.527090884	0.557146313	1	118	0.708589744	0.261188849	0	0.323575483
56	0.304966	0	0.254114	0.275689	119	0.5	1	0	0.25
57	0.75	1	0.75	0	120	0.5	1	0	0
58	0.74673443	0.768825575	0	0.65159046	121	0.5	1	1	0
59	0.261671607	0.311624497	0.67750696	0	122	0.278400763	0.274049298	0.238548231	0.332893359
60	0.5	0.5	0	1	123	0.490625	0.73515625	0.7125	0.25
61	0.5	0	0	1	124	0.429149923	0.351635154	0.333265388	0.710317801
62	0	0	0.5	1	125	1	0.5	1	1
63	0.5	0	0.5	1	126	0	0.446563661	0.546501434	0.603352645

Table A.2: Node coordinates of the 4D hypercube mesh, adapted to a mesh size of 1/3

Table A.3: Pentatope vertices of the same mesh

element vertices					element vertices (continued)					element vertices (continued)					element vertices (continued)				
$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$
124	39	66	33	31	32	44	115	101	108	54	123	47	76	88	35	99	24	91	69
69	4	98	86	52	105	113	85	26	20	14	28	33	39	66	23	68	122	25	56
45	47	89	16	74	60	45	58	108	12	83	82	33	50	38	68	23	122	26	56
83	63	33	26	14	108	45	74	60	12	85	5	105	113	110	108	41	58	60	12
28	85	33	26	100	104	44	47	101	64	125	14	33	50	39	48	72	28	36	26
124	68	66	61	10	93	97	15	90	47	71	45	126	19	34	54	16	80	108	39
124	43	60	34	108	45	47	108	16	76	45	12	80	79	74	68	31	48	26	56
84	109	15	90	47	28	103	38	50	88	69	4	54	86	35	123	54	69	101	108
18	48	27	91	26	124	78	10	46	31	85	5	105	110	20	66	81	72	46	31
33	14	125	67	39	92	4	69	35	54	88	123	38	76	73	124	10	66	46	31
124	68	60	78	10	124	39	35	46	31	88	28	98	39	91	122	25	68	118	31
18	48	28	91	29	76	47	38	123	88	48	18	27	36	26	79	124	60	78	10
122	59	126	20	85	18	36	113	26	100	78	79	102	46	35	23	20	122	26	56
41	108	74	60	12	4	111	58	108	54	35	92	124	39	58	25	2	31	81	118
20	23	9	87	124	104	103	47	101	121	88	8	54	76	39	62	126	124	65	106
43	124	115	34	108	45	19	115	34	108	91	19	115	101	64	45	126	38	112	73
1	56	27	91	25	1	122	20	17	59	22	122	126	17	96	47	44	71	19	90
19	45	126	115	34	71	93	15	90	47	103	123	47	38	64	31	28	26	91	48
44	19	71	34	108	25	56	31	68	48	49	16	54	76	39	29	69	39	86	35
33	45	125	76	39	29	2	31	81	48	94	102	58	79	35	25	122	31	91	118
19	115	69	91	73	108	41	11	43	119	65	63	55	61	124	103	104	47	7	121
36	18	117	28	100	110	7	64	107	103	56	59	27	91	26	97	93	7	90	47
49	8	50	76	39	92	31	39	91	73	23	122	68	25	70	2	25	31	81	48
17	40	115	21	22	19	59	126	91	38	108	92	124	60	58	59	56	27	91	1
1	122	21	17	96	73	122	92	124	115	78	81	35	118	31	20	23	122	26	124
12	45	80	108	74	28	48	31	91	29	92	115	69	91	32	24	32	115	21	91
31	92	122	91	73	61	55	124	79	63	105	113	85	107	100	17	40	115	44	3
123	19	38	91	64	44	17	126	19	90	69	29	99	86	35	122	23	1	25	70
108	11	44	43	119	88	18	98	95	117	20	87	9	62	124	66	81	10	68	31
115	19	69	108	73	73	85	122	26	124	31	122	26	91	73	35	79	94	39	58
48	18	28	36	29	18	103	95	117	6	84	51	126	38	109	124	65	126	22	34
19	44	115	101	64	7	47	64	107	103	32	40	115	43	22	71	44	126	22	34
107	97	105	90	110	17	32	115	64	91	18	117	28	36	29	45	55	82	125	77
62	126	13	112	106	92	115	122	91	73	71	108	44	11	34	79	35	124	39	58
28	88	38	91	103	122	59	26	91	85	47	45	89	71	74	73	124	115	92	108
116	1	59	56	20	122	73	126	124	115	59	56	27	20	26	59	122	126	91	85
47	45	82	38	109	87	65	22	23	124	44	104	47	7	64	16	45	80	67	55
2	35	31	91	29	38	47	64	107	90	88	8	50	76	47	81	29	72	46	31
88	8	98	86	39	85	28	26	91	73	19	59	126	90	17	88	28	38	91	73
2	35	99	91	118	35	2	99	91	29	40	17	115	44	22	16	54	76	108	39
45	33	125	76	82	92	108	39	124	73	122	92	68	22	124	56	48	31	91	26
44	101	37	3	64	88	69	98	86	52	113	105	85	107	110	32	69	115	92	108
27	116	59	56	20	47	7	64	107	90	32	108	58	4	119	47	45	108	16	74
13	38	82	112	84	122	23	68	22	70	103	59	100	107	38	54	123	47	101	108
85	20	33	112	124	44	47	64	19	90	14	83	33	77	125	4	69	99	86	35
85	28	38	91	100	55	45	124	106	34	33	28	73	76	39	67	79	94	12	80
18	98	42	91	29	123	88	38	91	73	20	63	62	112	53	65	71	126	22	34
88	123	38	91	103	21	122	24	22	70	63	20	33	112	53	31	28	33	26	66
90	84	126	38	109	108	45	124	60	34	123	19	69	91	73	35	92	24	91	118
76	82	33	50	125	55	45	74	60	34	71	65	11	22	34	45	55	77	125	67
103	18	95	117	88	8	49	50	76	47	86	69	39	54	35	32	44	3	101	64
18	98	28	91	88	42	36	117	29	72	122	87	20	23	96	59	38	64	107	90
18	113	27	36	26	68	124	66	31	10	54	111	58	108	12	84	47	82	38	109
4	92	69	108	54	24	122	118	25	91	59	19	126	91	17	18	98	42	117	6
40	17	115	21	3	51	84	82	106	112	43	124	60	22	92	38	105	97	107	90
12	45	60	79	74	13	53	105	112	62	19	71	44	108	47	35	2	31	81	29
31	92	39	91	35	126	19	38	91	73	19	71	45	108	34	124	65	60	55	34
50	33	125	76	39	81	78	10	46	35	126	85	105	112	84	35	2	31	91	118
1	122	59	56	20	18	48	28	36	26	44	115	126	22	34	2	25	31	91	118
126	122	20	17	96	122	126	124	87	20	45	71	126	90	109	16	49	54	76	47
19	45	47	76	38	45	79	124	60	55	29	2	72	42	48	103	18	28	117	100
110	59	64	107	90	48	18	42	91	29	44	71	11	22	34	92	4	69	108	32
65	87	126	62	124	55	45	74	106	89	43	44	11	34	108	92	58	39	54	35
87	122	22	23	96	66	33	124	26	63	92	73	69	108	54	109	84	82	89	106
48	18	28	91	26	38	88	50	76	47	41	108	58	60	43	81	25	31	68	48
72	14	28	36	26	101	88	121	103	47	40	32	115	21	22	59	110	85	107	90
71	109	15	89	47	28	14	50	39	117	101	88	91	69	123	104	44	37	7	64
8	88	98	95	117	82	76	33	50	38	43	124	60	92	108	103	28	38	50	100
109	71	15	90	47	126	85	38	112	73	19	45	73	126	38	32	44	115	3	64
69	4	99	86	98	84	109	82	89	47	32	3	44	101	119	108	45	124	39	58
31	28	72	26	48	88	8	54	57	47	24	92	69	91	32	88	28	98	18	117
69	4	99	24	35	59	122	1	91	17	123	19	73	76	108	78	79	124	35	46
103	123	91	101	64	122	17	1	21	91	44	19	115	17	64	103	88	121	95	47
68	124	31	78	10	45	60	58	79	12	54	94	58	111	12	38	59	85	107	90
20	85	126	112	124	82	83	13	38	112	35	79	94	102	46	39	124	66	33	79
122	59	126	91	17	44	17	115	3	64	13	84	126	106	112	108	32	58	43	119
88	101	54	69	123	59	103	100	91	38	47	38	97	107	90	79	45	124	60	58
88	8	98	39	117	122	92	91	118	31	108	92	39	54	73	103	59	64	91	38
45	73	108	124	115	5	116	59	113	20	18	48	42	36	29	4	69	92	24	35
68	63	9	61	124	36	14	117	6	100	108	71	74	11	34	55	45	82	89	1

element vertices (continued)					element vertices (continued)					element vertices (continued)					element vertices (continued)				
$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$
125	16	50	76	39	8	88	121	57	47	73	88	54	69	123	66	33	77	14	67
45	51	126	106	109	45	55	80	16	74	123	19	47	101	108	97	84	15	90	47
69	29	39	91	35	124	58	60	78	92	8	88	54	86	39	55	45	77	79	67
59	19	64	90	17	28	29	98	39	91	49	16	50	76	47	71	65	126	106	34
19	44	115	34	108	59	85	100	107	38	103	75	117	50	100	85	126	105	90	84
45	33	124	79	63	99	35	24	91	118	56	48	27	91	25	75	103	117	6	100
122	17	115	126	91	55	45	89	16	74	36	18	42	6	27	39	124	35	92	31
11	65	60	22	34	48	28	31	72	29	44	32	115	3	43	67	79	66	46	39
122	92	68	124	31	92	54	58	108	39	45	47	90	38	109	101	115	69	91	19
88	69	54	86	39	28	31	72	26	66	16	45	80	108	39	63	68	66	61	124
63	20	9	62	124	115	19	126	91	73	68	124	60	61	10	69	92	39	54	35
116	27	59	56	1	21	122	1	22	96	88	101	54	57	52	45	124	126	115	34
28	14	72	39	66	98	69	99	91	29	124	68	60	61	65	103	59	100	91	113
124	68	60	22	92	4	92	69	24	32	122	73	126	91	85	92	78	35	118	31
122	92	24	22	70	49	8	54	76	47	45	55	124	60	34	113	59	100	91	26
59	113	27	91	26	122	92	68	118	70	24	122	115	92	91	4	101	69	108	32
19	59	64	91	17	45	94	58	79	39	8	88	50	95	47	78	79	102	60	10
115	101	69	91	32	103	88	50	95	117	65	68	9	61	124	20	85	33	112	53
29	28	72	36	117	19	44	47	101	108	62	126	105	112	13	85	126	59	110	20
2	29	72	81	48	103	18	100	117	6	88	103	50	95	47	92	69	39	91	35
41	108	11	43	60	33	66	39	14	67	44	43	115	34	108	78	79	102	35	58
30	40	44	43	3	69	92	24	91	35	3	32	44	43	119	45	16	80	67	39
110	59	113	107	103	76	82	47	50	38	25	81	68	118	31	65	68	22	23	124
45	94	80	79	12	63	65	55	62	124	92	35	124	78	58	2	99	24	91	118
5	85	105	113	20	56	25	31	91	48	55	45	124	79	63	79	78	102	46	10
124	43	115	92	108	108	41	58	111	119	39	28	66	33	31	110	59	85	126	90
123	103	64	91	38	29	81	35	46	31	122	22	21	17	96	16	82	50	76	47
26	124	66	33	31	17	44	126	19	115	4	92	58	35	54	67	79	102	46	10
59	103	64	107	38	101	88	54	69	52	39	124	73	33	31	126	85	105	90	110
68	31	122	26	124	47	76	38	123	19	60	102	58	79	12	126	85	105	110	20
28	14	72	36	117	122	24	1	25	91	122	21	24	22	115	63	65	9	61	124
14	28	72	39	117	94	102	58	35	12	24	122	1	25	70	2	25	27	91	48
29	69	39	91	98	83	85	105	38	112	18	103	28	91	100	68	31	122	25	56
51	45	82	106	109	85	53	105	112	83	51	45	126	38	109	41	108	58	111	12
123	19	91	101	64	47	103	38	123	88	92	78	68	124	31	23	122	1	25	56
88	8	121	95	47	32	43	115	92	22	45	109	82	89	106	36	18	42	117	6
28	31	39	91	29	101	4	119	108	32	115	40	44	43	22	32	17	115	21	91
51	45	82	38	112	126	62	124	112	106	31	92	39	124	73	51	84	126	106	109
44	17	37	7	64	56	122	1	25	91	32	17	115	64	3	8	88	98	86	52
111	4	58	35	54	44	32	115	101	64	55	61	124	79	60	92	39	69	91	73
94	102	67	79	12	45	94	80	108	39	65	124	126	106	34	101	115	69	108	32
18	48	27	36	42	65	87	9	23	124	108	34	11	43	60	101	88	54	57	47
81	66	72	68	31	13	38	105	112	83	73	88	54	76	39	73	122	126	91	115
68	122	22	23	124	126	62	124	87	20	54	123	73	76	108	92	122	24	91	118
88	101	91	103	123	98	18	42	117	29	25	56	31	91	122	39	29	72	28	31
2	98	99	91	29	101	4	69	108	54	45	39	73	76	108	83	85	105	26	100
28	29	98	18	117	45	33	124	112	73	69	88	98	39	91	94	54	80	108	39
122	1	59	56	91	108	92	58	60	43	19	123	47	101	64	69	4	54	101	52
69	88	98	86	39	122	24	115	21	91	123	103	47	101	64	38	105	85	107	100
122	23	1	22	96	122	87	22	126	96	41	108	58	43	119	109	84	15	89	47
28	31	26	91	73	18	103	113	6	100	31	68	72	26	66	45	55	124	112	63
72	81	31	68	48	45	33	73	76	39	59	38	85	126	90	28	85	38	91	73
42	2	27	91	48	30	3	44	43	119	88	123	69	91	73	85	73	122	126	124
69	88	39	54	73	59	100	113	107	103	45	47	108	71	74	53	13	105	112	83
113	59	85	107	100	85	59	100	113	26	28	88	38	76	73	109	71	15	89	106
45	47	82	16	89	32	43	115	44	108	45	55	89	16	82	45	71	74	106	89
98	2	42	91	29	23	68	9	65	124	20	63	33	26	53	83	33	77	14	63
19	69	115	101	108	122	17	115	21	22	123	54	47	76	108	65	63	9	62	124
72	48	28	36	29	32	40	115	21	3	85	28	33	38	100	87	122	20	126	96
34	55	124	65	106	83	63	33	112	53	44	71	126	19	34	124	65	60	22	68
34	22	126	124	115	82	76	16	50	125	45	47	82	38	76	18	36	113	6	27
48	2	31	91	29	115	19	64	91	17	45	47	82	89	109	48	72	42	36	29
110	105	85	107	90	33	45	124	112	63	65	87	22	126	124	117	18	42	36	29
44	19	47	101	64	111	94	58	35	12	28	85	26	33	73	28	14	33	50	100
45	108	74	60	34	45	55	74	60	79	44	19	115	34	126	28	88	39	91	73
88	8	50	95	117	79	78	124	35	58	55	45	82	125	16	14	75	117	6	100
103	18	28	91	88	69	88	54	86	52	84	85	126	38	112	79	78	102	60	58
111	4	58	108	119	45	33	77	79	67	25	2	31	91	48	14	28	33	50	39
66	33	39	79	67	19	123	38	91	73	29	28	98	39	117	79	67	102	46	94
92	122	68	22	70	45	125	82	16	76	126	122	124	85	20	14	83	33	50	100
32	43	58	92	108	83	85	33	53	112	55	62	124	65	106	26	85	59	113	20
98	18	95	117	6	85	126	124	112	73	41	108	11	60	34	103	123	47	101	88
18	36	117	6	100	19	123	69	91	101	84	109	15	89	106	45	79	94	39	80
63	68	9	26	124	94	45	80	108	12	47	45	82	16	76	123	54	73	76	88
20	23	122	1	56	116	27	59	113	20	53	13	63	112	62	85	73	33	26	124
45	51	82	38	109	124	45	115	34	108	102	94	58	79	12	109	71	45	89	47
122	24	118	25	70	16	45	125	67	39	47	38	64	107	103	45	16	76	108	39
81	78	10	68	31	92	122	24	118	70	43	44	115	34	22	126	85	38	90	84
126	45	90	38	109	122	22	92	124	115	51	84	126	38	112	85	83	33	26	100
4	92	58	108	32	108	41	74	60	34	85	83	105	38	100	45	82	33	77	112
28	10																		

element vertices (continued)					element vertices (continued)					element vertices (continued)					element vertices (continued)				
$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$
43	124	115	92	22	43	11	60	22	34	8	88	54	57	52	45	79	124	39	58
78	92	68	118	31	55	45	77	112	63	109	84	38	90	47	23	20	9	26	124
8	49	54	76	39	85	59	100	91	38	85	38	105	112	84	103	47	38	50	88
24	122	1	21	91	124	79	60	61	10	84	97	38	90	47	62	20	124	112	63
25	122	68	118	70	47	44	64	7	90	51	45	126	106	112	85	5	59	110	113
51	84	82	38	109	63	83	33	26	53	73	31	33	26	124	45	33	73	112	38
110	59	85	107	113	17	44	64	19	90	45	51	126	38	112	55	62	124	112	63
122	21	1	22	70	92	122	24	22	115	38	105	85	84	90	92	4	58	108	54
33	45	73	76	38	8	88	50	39	117	20	59	122	26	56	24	32	115	92	22
1	122	20	23	96	43	32	115	92	108	31	73	122	26	124	3	32	120	101	119
126	17	59	110	20	68	23	9	26	124	87	122	22	126	124	92	115	69	108	73
72	39	66	46	31	40	115	44	43	3	17	44	37	3	64	32	4	120	101	119
92	122	68	118	31	33	66	77	14	63	59	122	126	20	17	31	28	26	33	73
45	124	126	106	112	28	76	88	73	39	82	45	33	38	112	126	59	90	38	19
108	32	44	101	119	114	36	113	6	100	54	94	80	108	12	88	101	121	57	47
104	103	47	7	64	45	94	58	108	12	69	29	39	86	98	55	45	124	112	106
103	50	28	117	88	126	45	124	112	73	79	67	66	46	10	82	45	33	77	125
45	126	90	38	19	44	19	115	101	108	47	38	64	19	90	68	124	60	78	92
113	59	27	20	26	29	39	72	46	31	85	20	105	112	53	79	124	78	46	10
116	5	59	110	20	13	126	105	112	84	108	92	39	124	58	124	79	66	46	10
48	18	27	91	42	113	105	85	26	100	19	45	73	76	108	45	108	124	60	58
14	36	117	28	100	88	8	54	86	52	20	63	9	26	124	85	83	105	26	53
100	18	28	36	26	18	113	100	91	26	122	87	22	23	124	108	45	39	124	73
26	18	28	91	100	65	124	60	22	34	71	19	45	108	47	41	108	74	11	34
17	122	115	21	91	103	104	47	101	64	116	110	59	17	20	107	97	7	90	47
13	53	63	112	83	84	51	126	106	112	63	20	33	26	124	28	88	98	39	117
85	20	33	26	53	19	123	47	76	108	28	14	33	26	66	75	14	117	50	100
122	1	20	17	96	31	28	39	91	73	11	34	44	43	22	39	124	35	46	79
14	36	117	6	72	20	63	33	112	124	123	54	47	101	88	94	111	58	35	54
18	103	28	117	88	123	54	73	69	108	50	83	33	38	100	115	92	69	91	73
7	110	64	107	90	38	13	105	112	84	84	51	82	38	112	105	38	85	107	90
25	2	24	91	118	14	28	33	26	100	103	75	95	117	6	44	104	37	101	64
93	71	44	90	47	36	42	117	6	72	30	11	44	43	22	22	44	126	17	115
45	19	73	115	108	31	28	72	39	66	33	45	79	67	39	55	45	80	79	74
83	105	114	26	100	124	43	115	34	22	45	19	126	115	73	103	28	100	91	38
17	44	90	7	64	17	32	115	21	3	20	85	33	26	124	45	16	125	76	39
105	113	114	26	100	123	19	47	38	64	45	71	89	106	109	54	94	58	108	39
67	79	94	39	46	85	26	59	122	20	45	19	73	76	38	32	91	115	101	64
19	71	45	90	47	23	20	122	87	124	83	85	33	26	53	71	45	126	106	109
115	40	32	43	3	45	71	126	106	34	88	28	50	39	117	126	62	105	112	20
78	35	102	46	10	87	65	9	62	124	73	85	33	38	112	28	76	33	50	39
81	66	10	46	31	33	45	125	67	77	78	81	10	46	31	14	83	114	26	100
23	122	1	22	70	33	66	77	79	67	124	39	66	46	79	83	14	33	26	100
45	71	126	19	90	19	59	64	90	38	73	85	33	112	124	45	51	82	106	112
79	35	94	39	46	17	126	59	110	90	91	31	122	26	56	19	123	73	69	108
36	18	113	6	100	82	83	33	50	125	45	19	47	76	108	38	28	33	50	100
59	19	64	91	38	66	33	77	79	63	45	55	77	79	63	28	14	36	26	100
14	28	117	50	100	45	16	80	108	74	97	84	38	90	105	83	85	33	38	100
48	56	27	91	26	29	39	35	91	31	76	28	88	50	39	36	14	114	26	100
19	123	69	101	108	33	125	77	14	67	32	24	115	21	22	97	107	7	90	110
14	28	72	26	66	92	69	39	54	73	76	28	33	50	38	83	82	33	77	125
108	45	74	71	34	32	108	44	43	119	26	72	31	68	48	83	14	33	50	125
4	69	54	86	52	28	31	39	33	73	85	126	105	112	20	83	82	33	38	112
75	103	117	50	95	29	69	99	91	35	40	30	44	43	22	83	63	33	77	112
98	18	28	91	29	22	122	126	124	115	28	29	72	39	117	39	54	73	76	108
92	35	31	91	118	31	68	66	26	124	124	45	126	106	34	94	45	58	108	39
113	18	27	91	26	88	69	73	39	91	81	78	35	46	31	19	123	73	76	38
94	45	58	79	12	82	83	33	77	112	71	109	45	90	47	8	88	50	76	39
85	5	59	113	20	85	28	38	33	73	38	47	45	19	90	79	67	94	39	80
78	124	35	46	31	45	16	125	67	55	1	116	59	17	20	33	45	77	79	63
39	124	66	46	31	85	20	122	26	124	28	33	73	76	38	55	45	82	112	77
53	105	85	26	20	32	24	115	92	91	45	33	125	67	39	5	85	59	110	20
59	56	122	91	26	45	33	82	76	38	39	29	35	46	31	14	36	114	6	100
11	30	44	43	119	85	126	38	91	73	110	7	64	17	90	113	36	114	26	100
78	79	124	60	58	84	51	82	106	109	31	68	122	26	56	59	110	64	107	103
16	49	50	76	39	122	85	26	91	73	59	85	100	91	26	38	59	126	91	85
										45	55	82	112	106	94	54	58	108	12

This mesh is composed of 4 573 pentatopes in 4D (again, virtual elements connected to node 0 have been omitted).

This kind of adaption has been successfully tested with  $h = 1/3$ ,  $h = 1/4$ ,  $h = 1/5$ . All generated mesh are not reproduced here, however, table A.4 gives major characteristics of these tests. Good final

uniform size	CPU time in seconds on a 220 Mflops workstation	number of nodes	number of elements	final averaged quality	minimal averaged quality
$h = 1/2$	4	126	946	0.486350	0.218303
$h = 1/3$	41	451	4573	0.502368	0.213219
$h = 1/4$	179	1 192	14 887	0.516302	0.198019
$h = 1/5$	547	2 588	35 894	0.518706	0.145331

Table A.4: Results about 4D adaption on a hypercube

qualities are attained, in reasonable computational times.

## Appendix B

# Complements on natural and multidomain metrics

Some useful algorithms for the natural metric and the multidomain metric have not been described in chapters 4 and 6. For the sake of completeness, their are detailed in this appendix.

### B.1 Local computation of the layer number

In the natural metric computation, the local number of element layers through the thickness is required, both for the minimal neighborhood order evaluation (section B.1.1) and for the convergence criterion (section B.1.2).

#### B.1.1 Minimal order evaluation

The following algorithm is employed to evaluate the local number of element layers in the mesh.

**Algorithm B.1** proceeds by two steps:

descending: an incremental propagation according to successive neighborhoods

- every boundary nodes belong to layer 1
- their (not already visited) neighbors belongs to layer 2
- this process is repeated, until no node remains

we obtain  $k_{\max}$ , the maximal layer number

ascending: a retro-propagation from the interior to the boundary

```
for  $k$  from  $k_{\max} - 1$  to 1 do  
  for  $l$  from  $k + 1$  to  $k_{\max}$  do  
    – the nodes belonging to the current layer  $k$   
      that are also neighbors of nodes belonging to the new layer  $l$   
      become nodes of the new layer  $l$   
  done  
done
```

At the end of this algorithm, each node  $S$  is associated with a final layer number, denoted by  $k(S)$ , which is nearly half the local number of element layers. Then, the minimal order  $k_{\min}$ , chosen at node  $S$ , is at least equal to  $k(S)$  and at most equal to  $2k(S) - 1$ . Empirically, this range works well.

Furthermore, we denote by  $k_0$  the first order such that the hull  $\mathcal{C}_{k_0}(S)$  intersects the boundary. We denote by  $S_0$  one of the intersection node (belonging to  $\mathcal{C}_{k_0}(S) \cap \partial\Omega$ ). Then, we determine whether all nodes in  $\mathcal{C}_{k_0}(S) \cap \partial\Omega$  belongs to the same connex compound than  $S_0$  (by boundary edge searching) or not. If it is not the case, then the hull  $\mathcal{C}_{k_0}(S)$  intersects the boundary  $\partial\Omega$  with, at least, two connex compounds and the elliptic interpolation begins to be relevant on  $\mathcal{C}_{k_0}(S)$ .

Finally, the minimal order is

$$k_{\min} = \min \left( \max(k_0, k(S)), 2k(S) - 1 \right) \quad (\text{B.1})$$

### B.1.2 Convergence criterion

Thanks to algorithm B.1,  $2k(S) - 1$  is the local number of element layers (or this number + 1). That is the reason why we consider the following averaged number of element layers as a convergence criterion

$$n_{\text{average}} = \frac{1}{\text{card}(\mathcal{N})} \sum_{S \in \mathcal{N}} \left( 2k(S) - \frac{3}{2} \right) \quad (\text{B.2})$$

In this average, all the node set  $\mathcal{N}$  is used, except for nodes where the natural metric computation is impossible (a node belonging to no element, for instance). Furthermore, when the natural metric is computed for a subdomain of  $\Omega$  (next section), the node set used in this average computation is intentionally reduced to the nodes belonging to that subdomain.

## B.2 Subdomain boundary

In the multidomain framework, we need to determine the boundary of a subdomain  $\omega$  (for the computation of its natural metric), within a mesh of the global computational domain  $\Omega$ . Such a boundary is essential, since boundary nodes play an important role in the aforementioned minimal order evaluation.

### B.2.1 Boundary nodes

The following algorithm is devoted to the boundary node determination of  $\omega$  (among all nodes that belong to  $\omega$ ).

**Algorithm B.2** a loop involving two steps is required:

```

– node 0 is said to be on layer number 1
– the layer number, denoted by  $k$ , is initially 0
while some nodes have not been treated do
  while a node can be found outside  $\omega$  do
    – increment  $k$ 
    for each node  $S$  on the layer number  $k$  do
      – each neighbor of  $S$  is either on the layer number (-1) if it belongs to  $\omega$ ,
        or on the layer number  $k+1$  otherwise
    done
  done
  – all nodes of the layer number (-1) are boundary nodes of  $\omega$ 
  and become nodes on the layer number  $k+1$ 
  while a node can be found inside  $\omega$  do
    – increment  $k$ 
    for each node  $S$  on the layer number  $k$  do
      – each neighbor of  $S$  is either on the layer number  $k+1$  if it belongs to  $\omega$ ,
        or on the layer number (-2) otherwise,
        in the latter case,  $S$  becomes a node of the layer (-1)
    done
  done
  – all nodes of the layer number (-1) are boundary nodes of  $\omega$ 
  and become nodes of the layer number  $k+1$ 
  – all nodes of the layer number (-2) become nodes of the layer number  $k+2$ 
done

```

This algorithm proceeds by coloring from the node 0, following successive neighborhoods and distinguishing between nodes inside and nodes outside the subdomain. After this algorithm is run, all boundary nodes of the subdomain are detected and its natural metric can be computed by hull growing.

Unfortunately, the list of boundary nodes is not enough in the determination of whether a hull intersects the boundary with at least two connex compounds. The connectivity between boundary nodes is also required (following section).

### B.2.2 Boundary faces

Once the nodes that belong to the subdomain  $\omega$  are known, the sub-topology  $\mathcal{T}_\omega$  (that is, elements of  $\Omega$ 's mesh whose all vertices belongs to  $\omega$ ) can be extracted. For a multidomain mesh with exact interfaces, the obtained sub-mesh represents accurately  $\omega$  and boundary nodes of this sub-mesh are the same than those of the preceding section.

However, if interfaces are fuzzy, the sub-mesh does not represents accurately  $\omega$  and boundary nodes of this sub-mesh may not be all the same than boundary nodes determined by the previous section (figure B.1). In particular, some boundary nodes (determined by the previous section) may belong to no

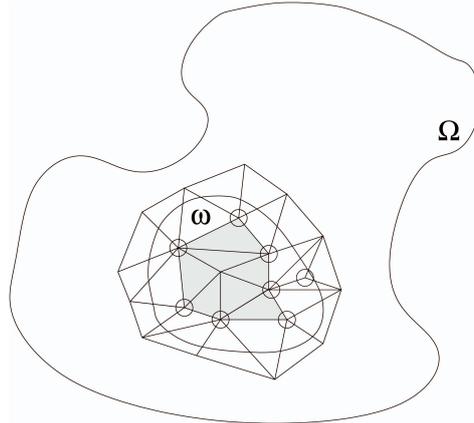


Figure B.1: Sub-mesh of a subdomain  
(boundary nodes of the preceding section are circled, while  $\mathcal{T}_\omega$  is the set of elements in gray)

element of  $\mathcal{T}_\omega$ . That is the reason why  $\mathcal{T}_\omega$  is completed by the elements of the initial mesh, whose at least one vertex is a boundary node of  $\omega$  (elements in white on figure B.1). In this way, no boundary node is isolated.

Thus, some nodes outside  $\omega$  can be used in the final sub-mesh. However, this complementary nodes are not used as boundary nodes. They only serve the connectivity purpose, in the determination of whether a hull intersects the boundary  $\partial\mathcal{T}_\omega$  with at least two connex compounds (as described in section B.1.1).

## B.3 Voxelization

Still considering a subdomain  $\omega$ , the discontinuous  $P^0$  interpolation of its characteristic function relies on the filling ratio computation of each mesh element by  $\omega$ . In our case, this filling ratio is evaluated by sampling the element in many points and by counting points that belong to  $\omega$  (section B.3.1). For that purpose, an efficient technique, to determine whether a point belong to  $\omega$  or not, is required. Here, we use a black and white voxelization of  $\omega$  (section B.3.2).

### B.3.1 Sampling

We denote by  $\epsilon$  the minimal thickness size of  $\omega$ . To ensure that no geometrical detail of  $\omega$  is ignored by the sampling, the distance between two consecutive point of the sample is imposed to be always lower than  $\epsilon/\sqrt{2}$ .

Let  $T$  be a mesh element whose vertices are  $S_0, \dots, S_d$ . The sample used to evaluate the filling ratio of  $T$  is composed by the nodes:

$$S_0 + \mu_1 S_0 S_1 + \dots + \mu_d S_0 S_d \quad (\text{B.3})$$

where scalars  $\mu_j$  are

$$\mu_j = \frac{i_j}{I_j} \left(1 - \frac{i_1}{I_1}\right) \dots \left(1 - \frac{i_{j-1}}{I_{j-1}}\right) \quad (\text{B.4})$$

and where indices  $i_j$  are

$$0 \leq i_j \leq I_j = \text{ceil} \left( \left(1 - \frac{i_1}{I_1}\right) \dots \left(1 - \frac{i_{j-1}}{I_{j-1}}\right) \frac{\|S_0 S_j\|}{\epsilon/\sqrt{2}} \right) \quad (\text{B.5})$$

the ceil function being the integer part +1.

Such a sampling is not uniform over the mesh, but sufficiently accurate that no geometrical detail with a size greater or equal than  $\epsilon$  misses all points.

### B.3.2 Mesh screening

When  $\omega$  is defined by simple geometrical zones (like a paralleloptope, a cylinder, a sphere, maybe transformed by an affinity), it is easy to localize a point  $x \in \mathbb{R}^d$  inside or outside  $\omega$ , with a constant number of operations. In this case, the filling ratio evaluation is not expensive.

Unfortunately,  $\omega$  is, in general, a complex geometry, modelled by a CAD tool. After meshing operations, a mesh of  $\omega$  can be used for the localization of a point  $x$ . But the cost of such a localization is linear with the number of elements in the mesh (and numerical roundoff errors can be encountered). The localization cost based on a mesh can be reduced by the use of an appropriate tree (section 2.1.3.2 page 24).

Here, a voxelization (in 2D, it would be called a pixelization) of  $\omega$  is preferred (figure B.2). In practice, a bounding box of  $\omega$  (the screen) is divided in small parallelotopes (the pixels in 2D, the voxels otherwise) and the geometrical zone corresponding to  $\omega$  colors the concerned pixels or voxels. Many 3D rendering software use this technique [Schroeder *et al.* 2002, Kaufman 2004].

For our purpose, only monochromatic screens are considered, were lit pixels or voxels are those belonging to  $\omega$ . Pixels or voxels could be anisotropic, but we only use isotropic ones, whose size is imposed by the required accuracy. For the sake of simplicity, we also consider only Cartesian principal directions for the screen. Presently, no tree organization (section 2.1.1 page 19) is used. So, all pixels or voxels have the same size  $\epsilon'$ , which should be small enough that no geometrical detail of  $\omega$  is missed.

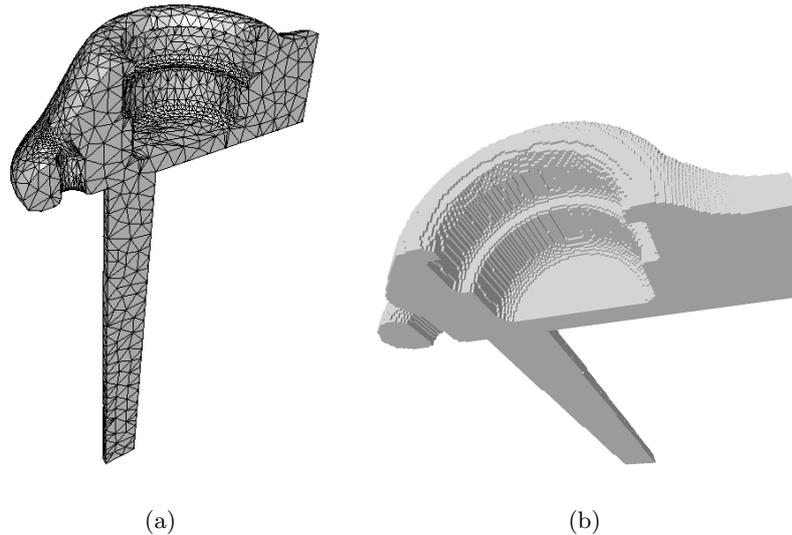


Figure B.2: Coarse mesh of  $\omega$  and its voxelization.

So as to lighten voxels belonging to  $\omega$  (which is defined by a coarse mesh), several techniques could be employed:

- for each voxel of the screen, determine if it belongs or not to one element of the mesh (this is the most expensive way)
- for each element  $T$  of the mesh and for each voxel  $V$  of a bounding box of  $T$ , compute the distance vector between  $T$  and  $V$ 's center; if all coordinates of this vector are lower than  $\epsilon'/2$ , then  $V$  is lit (this way, no voxel is forgotten, but the distance vector is not easy to compute for a simplex  $T$ )
- for each element  $T$  of the mesh, sample  $T$  as in the preceding section and light the closest voxel of each point in the sample (this way, some voxels can be forgotten, but only in the vicinity of the boundary)
- the previous technique can be used to lighten the boundary of  $\omega$ , then, each connex compound of the screen can be scan (unfortunately, many pathological cases arise in 3D).

The second technique is employed by the Vtk library [Schroeder *et al.* 2002]. However, we need a technique easy to implement for all spatial dimensions. We focus our attention on the third technique.

So as to forget no voxel in the vicinity of the boundary, boundary faces are also sampled and the corresponding voxels are lit. The overall technique is not very expensive, since a screen involving 31 million voxels has been lit with the coarse mesh of figure B.2, within 24 secondes on a 220 Mflops workstation.

## Appendix C

# Rem3D simulation example

Here we add the report of a Rem3D simulation, relying on an anisotropic mesh, generated by a natural metric. The aim of this appendix is to show that such a mesh is suitable for a finite element computation of the filling process, together with a viscoplastic behavior and thermal coupling.

The geometry of interest has been proposed by Plastic Omnium (figure C.1). Simulation of the filling

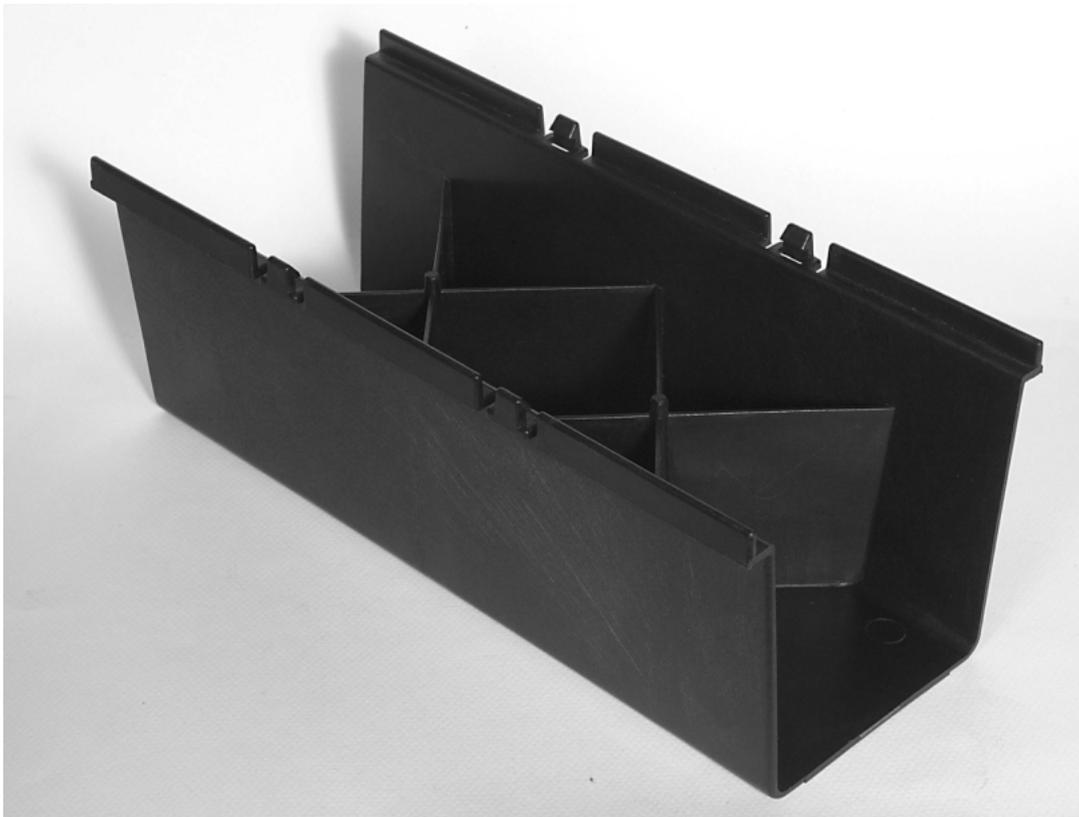


Figure C.1: Picture of the real part

stage has been performed with Rem3D version 2.1.

## C.1 Pre-treatment

In what concerns modelling, some characteristics of the mesh generation are described. Rheological parameters, thermal parameters and boundary conditions are also mentioned in this section.

### C.1.1 Mesh

This simulation relies on a mesh generated by topological optimization (section 2.2 page 25), driven by a natural metric field. The metric field is built with 8 element layers required through the thickness (chapter 4 page 55). Blocking faces (that is to say, faces whose vertices are all boundary nodes) have then been treated. For symmetry reasons, only half of the geometry is meshed (figure C.2).

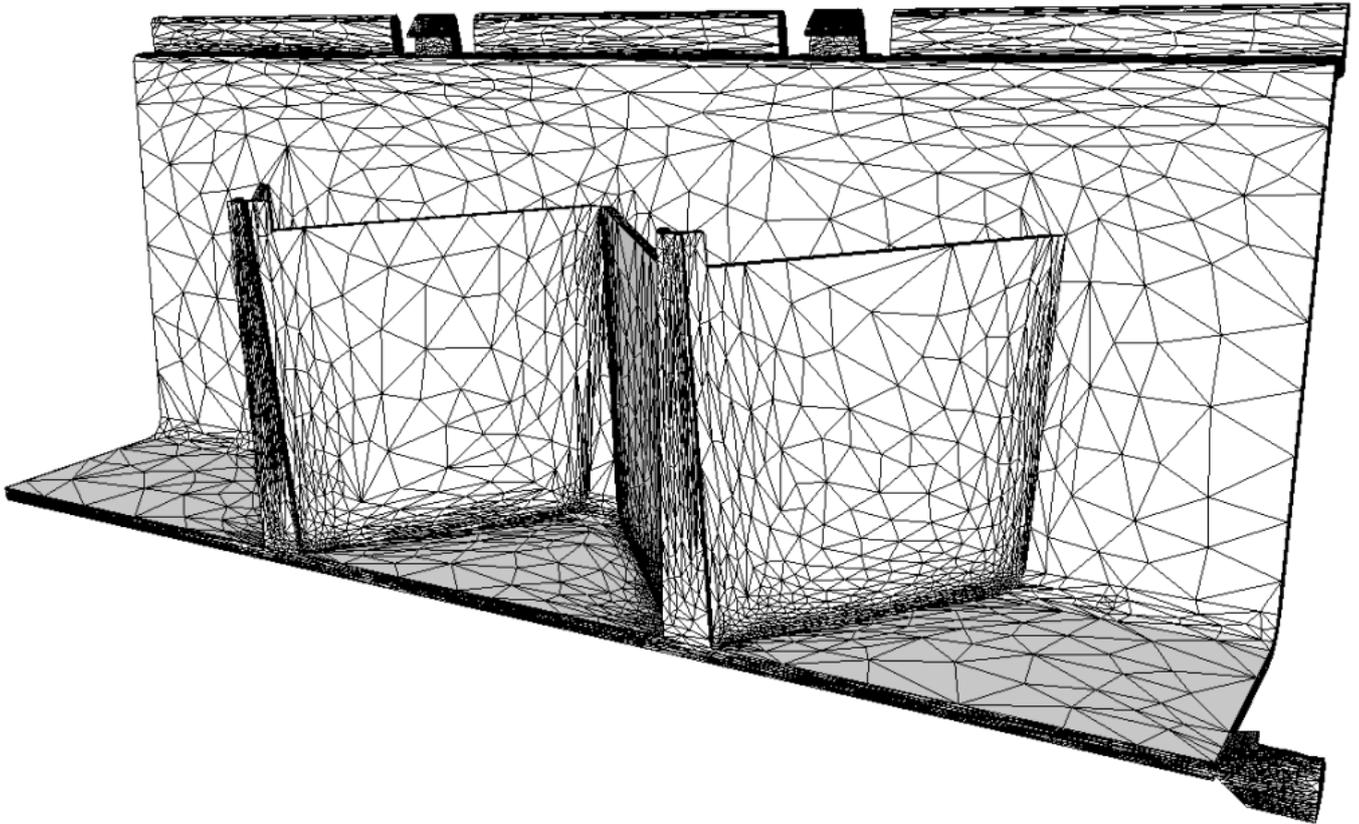


Figure C.2: Natural mesh of the Plastic Omnium geometry

This meshing phase leads to a mesh with about 63 000 nodes and 320 000 elements. This order of magnitude is compatible with an implicit simulation running on a workstation equipped with 750 Mo RAM (tableau C.1).

number of nodes	62 937
number of elements	320 540
workstation	Pentium IV 1.5 GHz 750 Mo RAM
elapsed time	31 hours

Table C.1: Summary of simulation characteristics

Furthermore, a R-adaption (without remeshing) has been applied on the mesh during the simulation. It allows a numerical diffusion reduction around the flow front. Without R-adaption, this kind of diffusion

makes the flow front position very noisy, because of the mesh anisotropy.

### C.1.2 Materials

The injected material is a polypropylene, called Stamax P30YM240. Its fiber reinforcement is not modelled: a Cross-WLF behavior law is chosen. This kind of law is traditionally employed for amorphous polymers, between their glass transition temperature and 100 K above. Here, the Cross-WLF is chosen for a semi-crystalline polymer, because parameters are available in a material database. This materials database (embedded with GLPre 2.2) gives the following values:

- a consistency of 2 175.99 Pa
- a shear rate exponent of 0.35
- a density of  $0.93 \text{ g.cm}^{-3}$
- a specific heat of  $2.25 \text{ J.g}^{-1}.\text{K}^{-1}$
- and a conductivity of  $0.19 \text{ W.m}^{-1}.\text{K}^{-1}$ .

For the air trapped inside the mold cavity, the default parameters are considered (a low density and a low viscosity).

### C.1.3 Initial conditions and boundary conditions

In what concerns the initial temperature, the fluid polymer is injected at  $250^\circ\text{C}$  and the mold is assumed to be perfectly regulated at  $40^\circ\text{C}$ . Initially, the air inside the cavity is considered at a temperature of  $40^\circ\text{C}$ .

At the injection gate, the fluid is injected with an imposed rate. This rate is computed so that the filling stage is achieved in 3.7 secondes (which is the injection duration given by Plastic Omnium). Volume of the part being  $189\,357.9 \text{ mm}^3$ , we imposed a rate of  $51\,177.8 \text{ mm}^3.\text{s}^{-1}$ .

The symmetry plane is adiabatic and no normal velocity is admitted on it. All other boundary zones (except the injection gate and the symmetry plane) have an imposed zero velocity (no slip contact).

## C.2 Post-treatment

Here, global results are presented and we also focus our attention on the local behavior around a weld line.

### C.2.1 Flow during injection

In what concerns the flow front evolution during the filling stage, some numerical short shots are plotted on figure C.3. The flow front is plotted via the 50% level set surface of the fluid filling ratio.

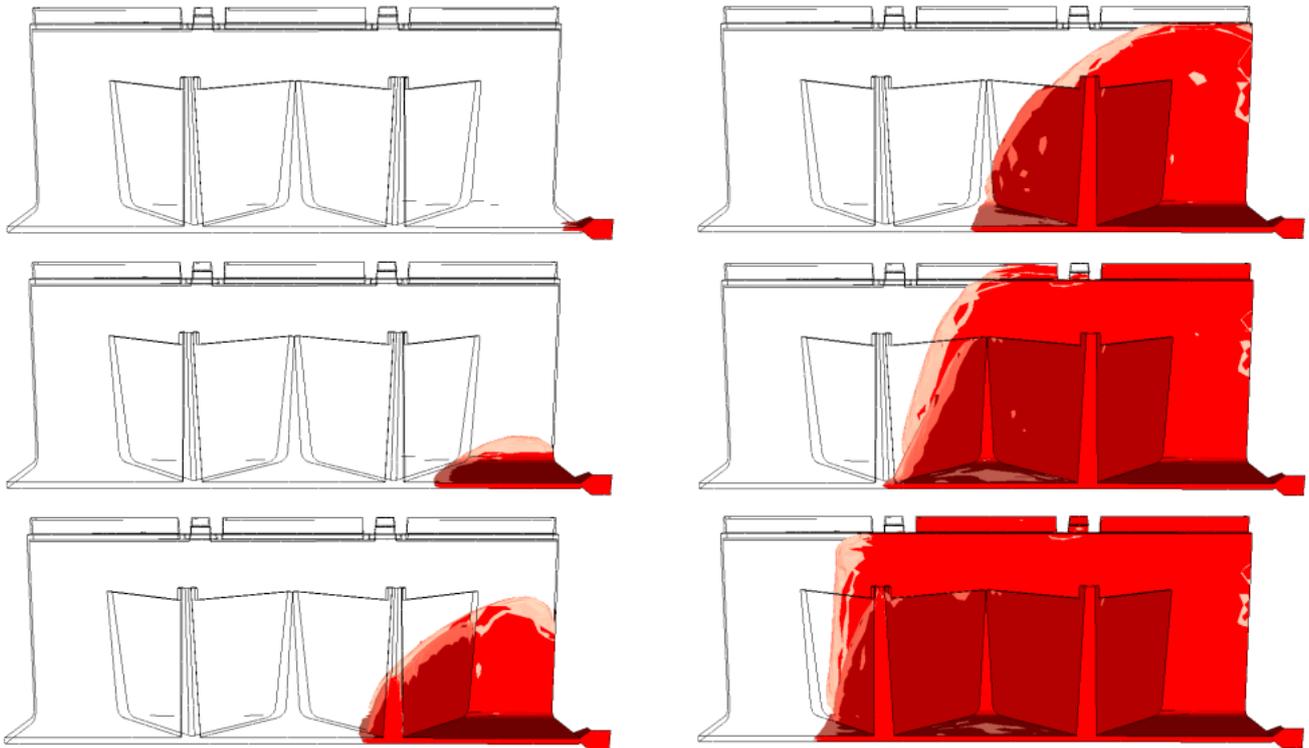


Figure C.3: Numerical short shots (organized in column)

With the injection gate on the right, the front moves from the right to the left and fills successively the floor, the side and the internal cross plates. This filling scenario is realistic, however some elements close to the boundary are not easily filled. It may be caused by a slight incompatibility between the convection solver and some anisotropic elements.

The pressure history at the injection gate (figure C.4) shows a sudden inflection around 0.2 s, when

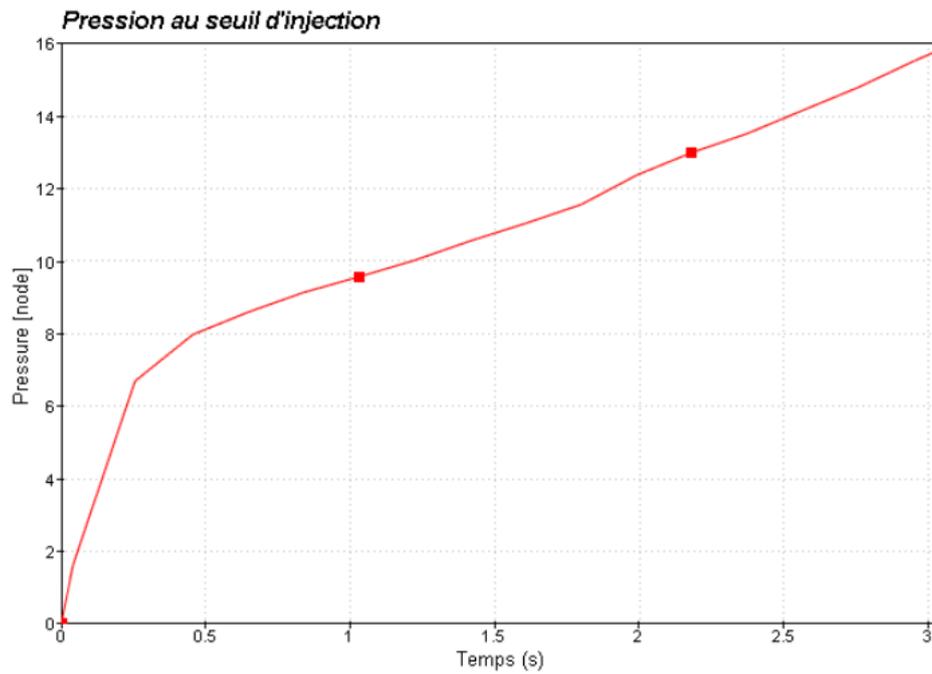


Figure C.4: Pressure history at the injection gate

the fluid reaches the floor plate. After 0.2 s, the flow front can extend in a larger space, thus, a constant injection rate leads to a lower injection pressure. At 2.75 s, the pressure field is given on figure C.5.

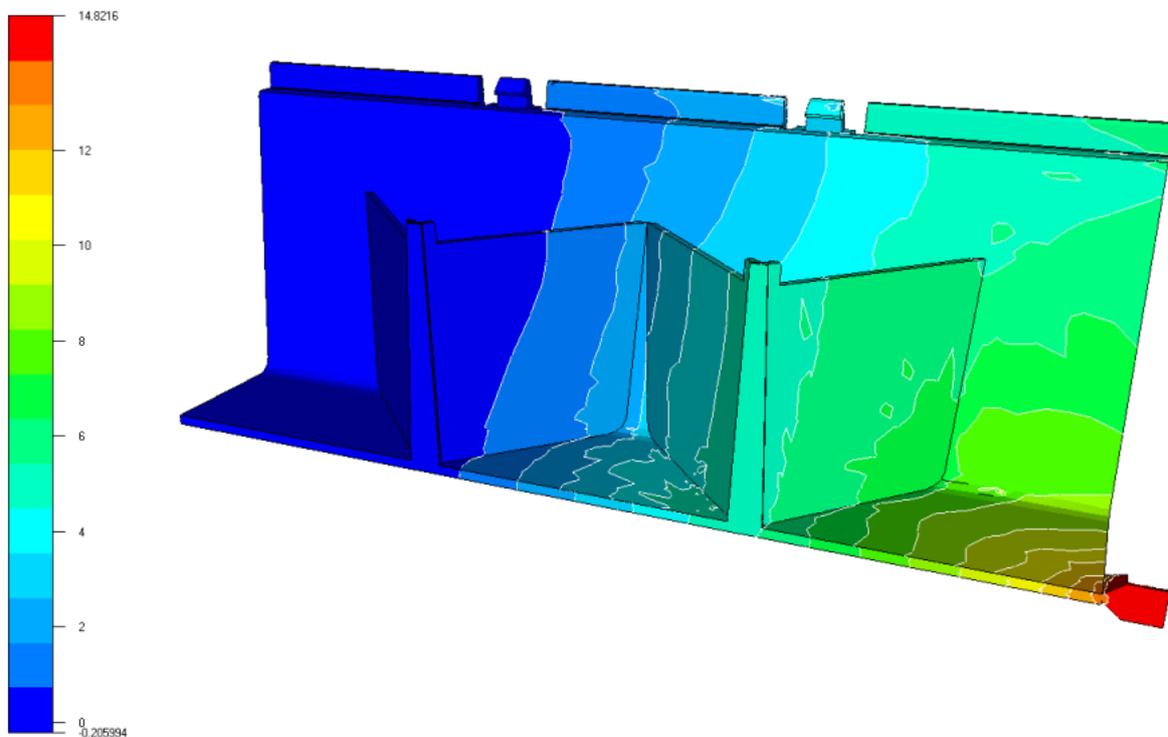


Figure C.5: Pressure field (in MPa) at 2.75 s

The boundary velocity field is not very interesting, since the boundary condition is homogeneous, except at the injection gate and on the symmetry plane. That is the reason why we prefer to plot the interior velocity profile (figure C.6). Beyond the flow front, we can observe a non zero velocity field in

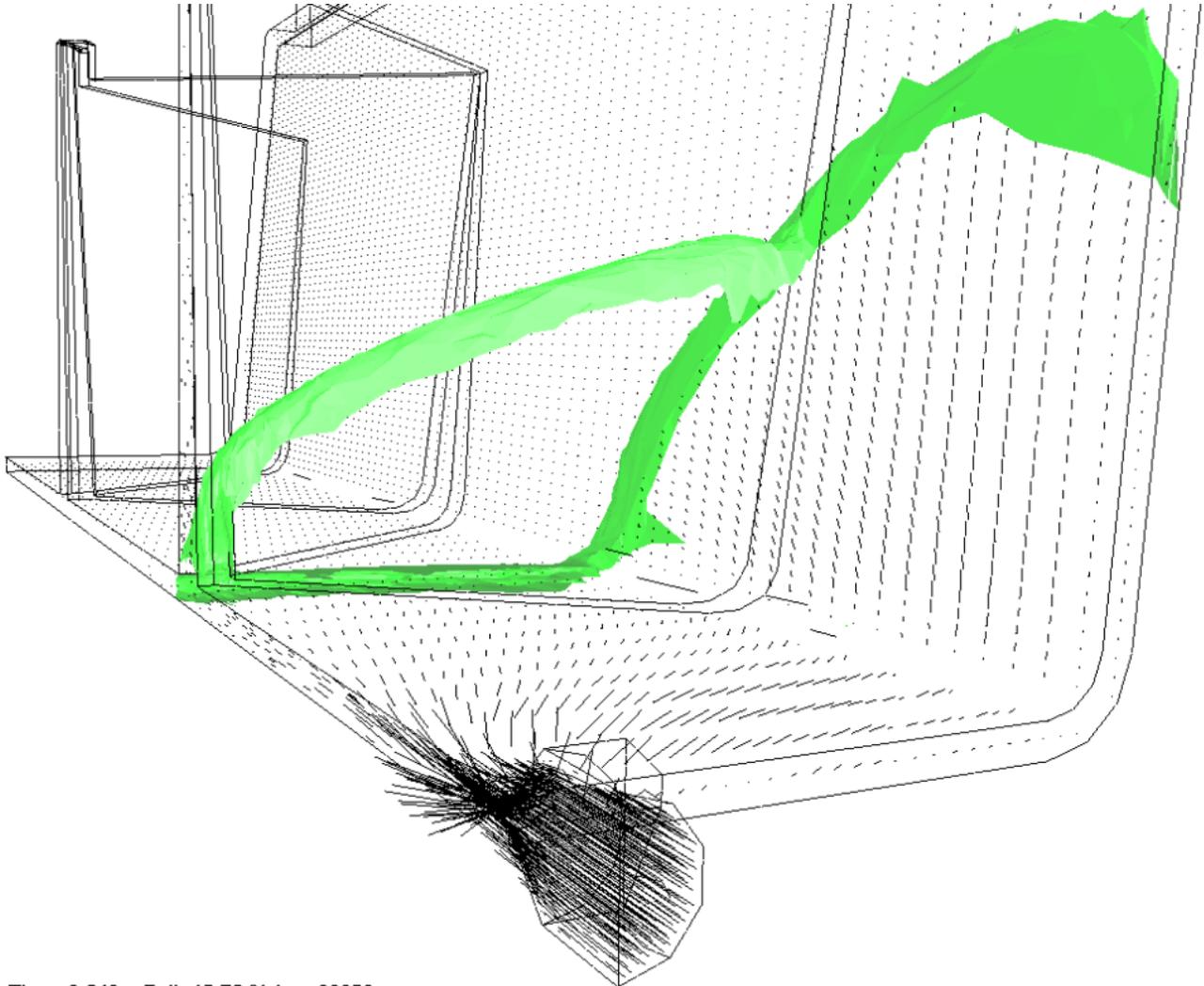


Figure C.6: Velocity vector field during the filling process (the flow front is in green)

the air. This is due to the fact that the air is modelled by a compressible and almost inviscid material.

In what concerns the temperature field (figure C.7), mechanical dissipation leads to a 19°C tempera-

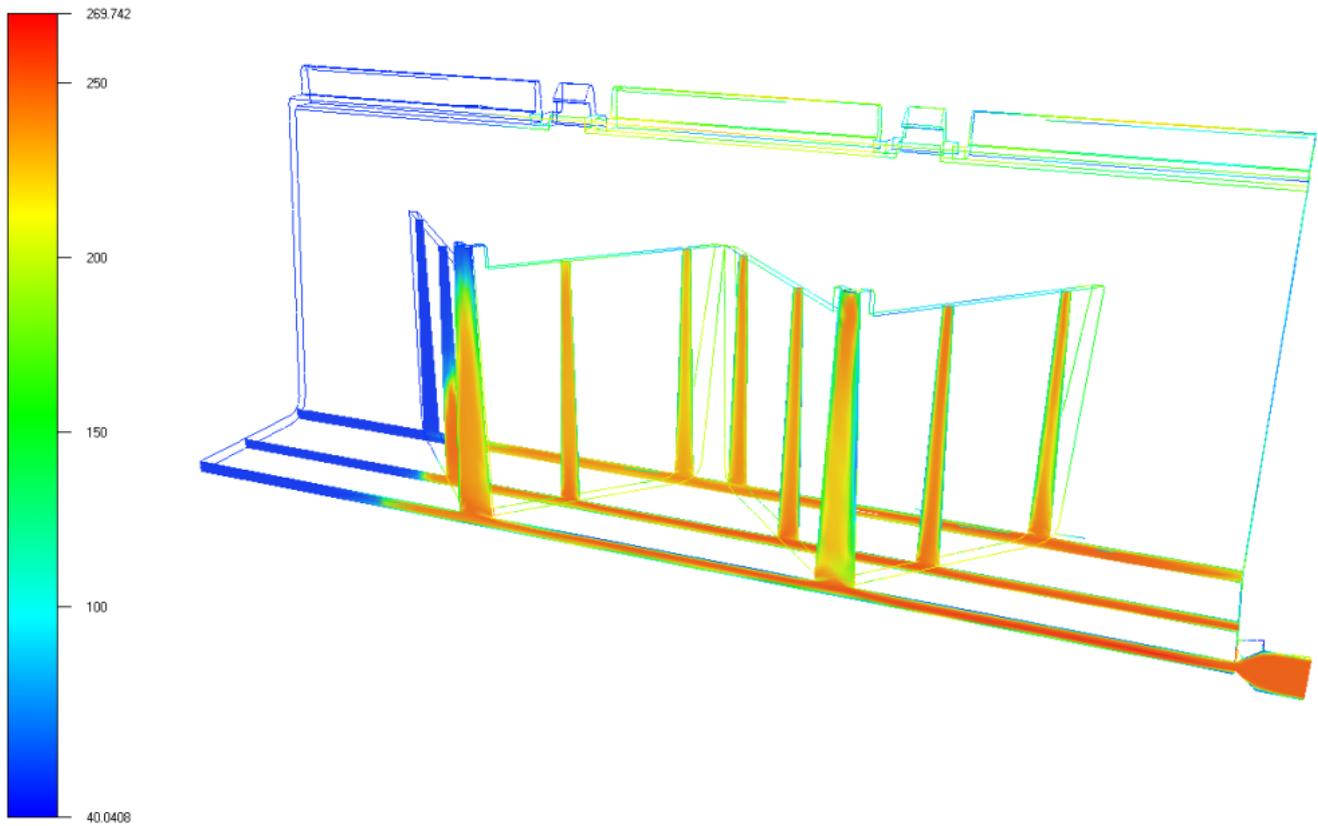


Figure C.7: Temperature (in °C) at 3 s

ture rising, which might be overestimated. It militates in favor of taking the element shape into account, in the thermal solver (which relies on a mixed formulation, P0 in temperature and P0+ in flux), so that anisotropic meshes become more compatible with.

This global results show that a Rem3D computation has succeeded with a natural mesh. However, some improvements are possible for the Rem3D convection-diffusion solvers, so that anisotropic meshes can be used without any limitation.

### C.2.2 Weld line study

In fact, the real part has a default: a weld line takes place in the middle of its side (figure C.8). In



Figure C.8: Picture of one part side (in this rear view, the injection gate is on the left)

this picture, we can observe the local orientation of the fiber reinforcement. We focus our attention on the undesired weld line (this curved line appears darker than its surrounding). For symmetry reasons, the same weld line can be observed on the other side plate.

This weld line is located before the crossing zone between the side plate and an interior plate (figure C.9, this crossing zone appears brighter).

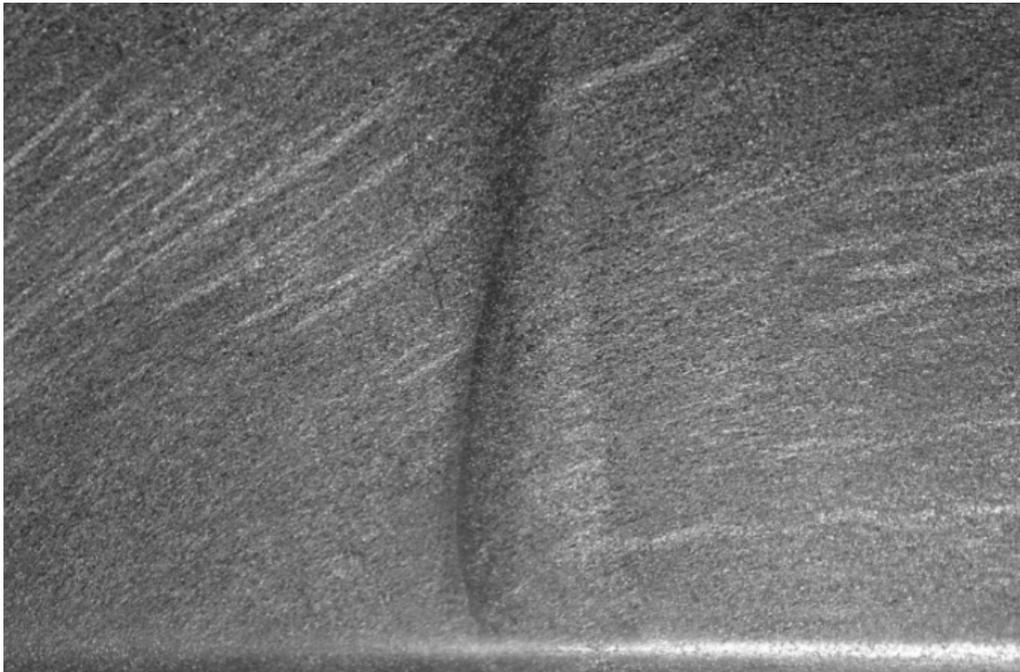


Figure C.9: Partial view around the weld line

It is important for Plastic Omnium to determine whether such a weld line formation can be predicted by Rem3D or not, because a weld line is a weakness of the final part, which must be taken into account in further structure simulations. Here, we present some results around this local zone.

During the filling stage, flow fronts of the side plate and of the interior plate meet between  $t = 1.5$  s and  $t = 2.0$  s (figure C.10). The simulation result indicates a weld line formation, located at the crossing

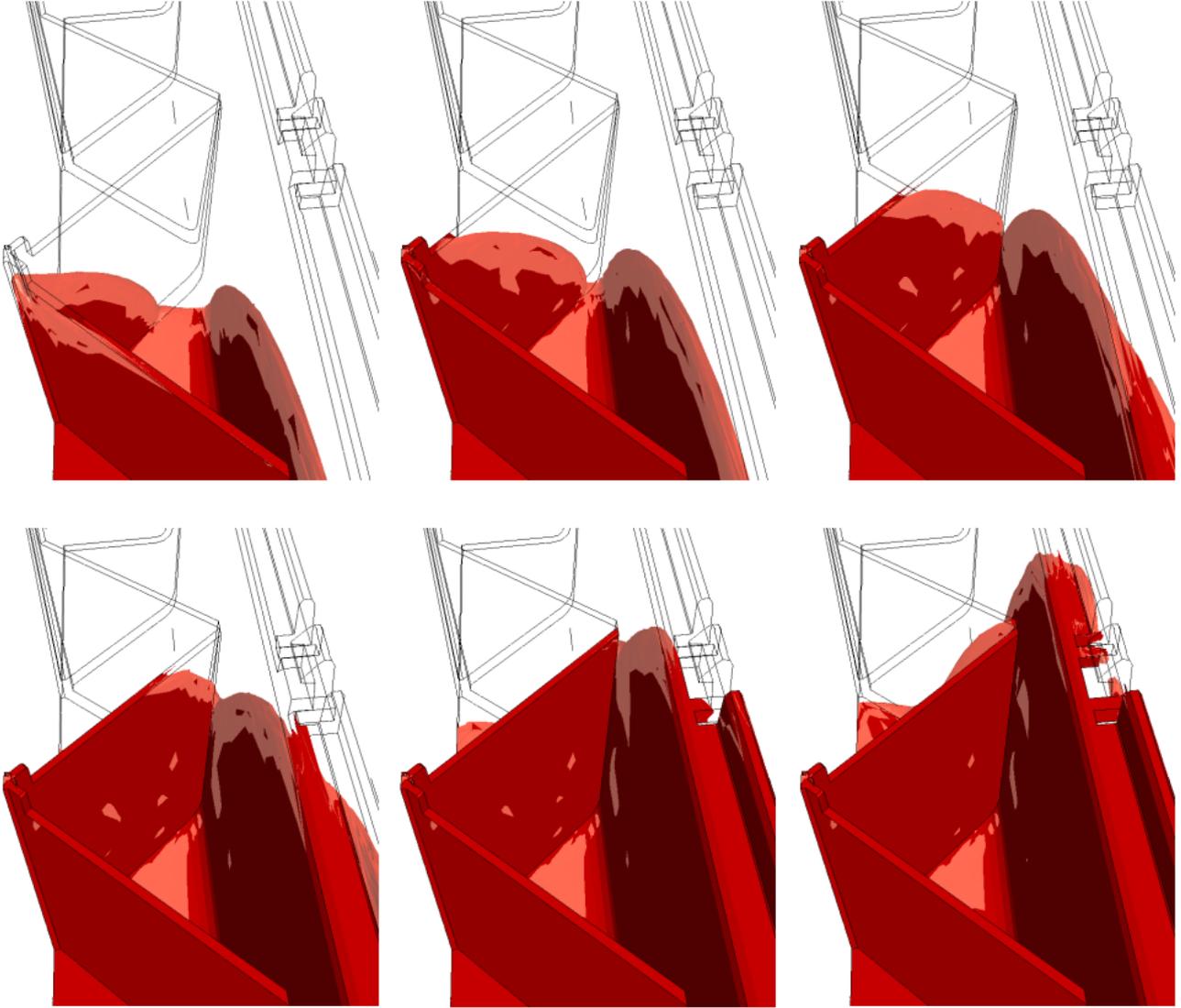


Figure C.10: Numerical short shots around the weld line

zone between the side plate and the interior plate (which is a little different place from the observation of figure C.8). However, some numerical diffusion remains around the flow front (despite the R-adaption), which prevents us from accurately locating the weld line.

Unfortunately, the post-treatment tool (GLView) cannot keep the weld line information. However, its formation can be detected by the temperature field observation on a cutting plane inside the side plate (figure C.11).

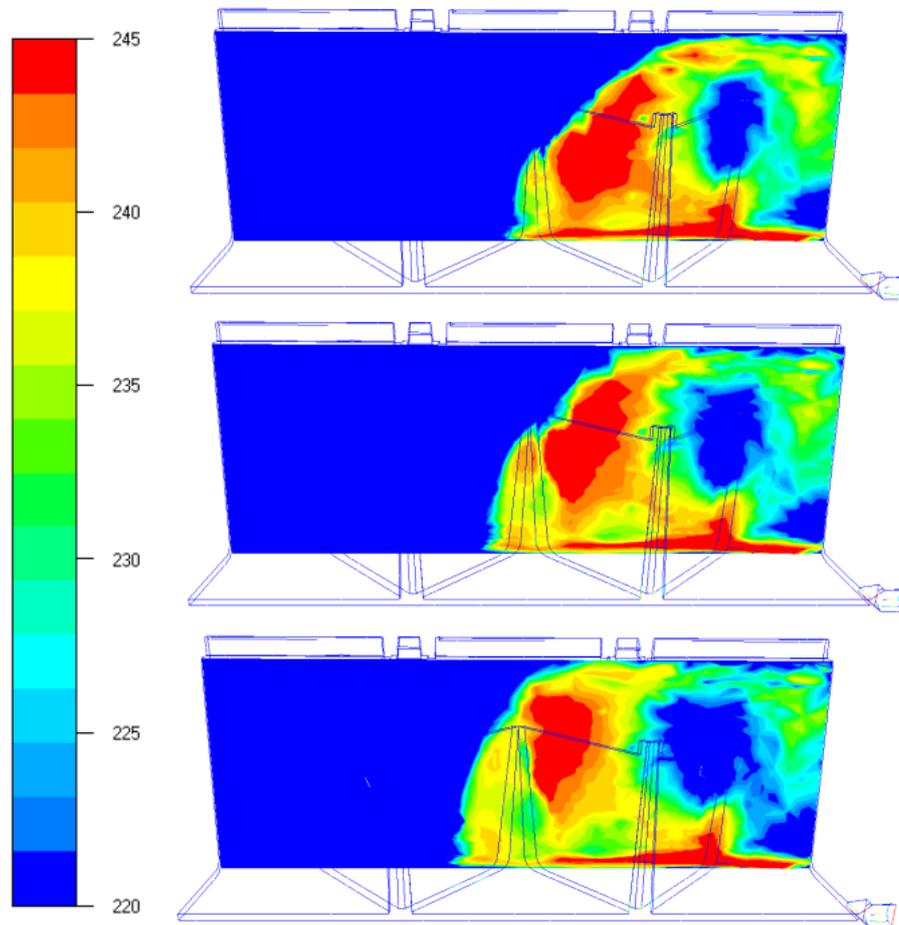


Figure C.11: Temperature (in  $^{\circ}\text{C}$ ) on a cutting plane (inside the side plate) at different time steps (truncated scale)

On figure C.11, the scale has been intentionally tighten around  $220^{\circ}\text{C}$ , so that we can observe a cooler line formation at the weld line place. This cooler zone is a weld line manifestation. However, it seems difficult to derive an automatic weld line detection tool from such a temperature field observation.

As a conclusion, Rem3D and GLView can be used to locate manually the formation of a weld line, but this is not an automatic tool for weld line prediction. Further investigations are needed so as to detect and to store a weld line information, dedicated to structural simulations on final parts.

# Appendix D

## Notations

$\mathbb{1}_\omega$	characteristic function of a subdomain $\omega$ .....	103
$AB$	vector $B - A$ .....	
$\alpha$	friction coefficient .....	127
$\alpha_h$	security factor dedicated to size selection for division .....	71
$\alpha_{\text{imposed}}$	imposed error level .....	91
$\alpha_S$	error indicator at node $S$ .....	92
$\alpha_{S'}$	weight of a node $S'$ in the computation of center $C_k(S)$ .....	60
$\alpha_T$	error indicator on element $T$ .....	91
$A_T, C_T^\top$	transformation associated to a simplex $T$ .....	14, 94
$\beta$	reduction factor computed from the indicated error .....	92
$\partial\Omega$	(geometrical) boundary of $\Omega$ .....	15
$\partial\mathcal{T}$	topological boundary of $\mathcal{T}$ .....	16
$\partial\mathcal{T}_k^{\text{ext}}(S, S')$	boundary faces of $\mathcal{T}_k^{\text{ext}}(S)$ around a node $S'$ .....	60
$\partial_{\frac{1}{2}}\Omega$	median surface of a domain $\Omega$ .....	58
$c_0$	normalization factor of the shape criterion .....	39
$c_{\text{allocation}}$	memory allocation of the next mesh .....	18
$\text{card}(\mathcal{T})$	number of elements in the mesh .....	92
$\text{card}(\mathcal{T})_{\text{imposed}}$	imposed number of elements .....	91
$C_k(S)$	center of the $k$ -th order hull around node $S$ .....	60
$\mathcal{C}_k(S)$	$k$ -th order hull around node $S$ .....	59
$c(T)$	quality criterion of a simplex $T$ .....	14
$d$	spatial dimension, number of coordinates .....	11
$D$	topological dimension, number of element vertices .....	15
$\vec{d}$	unitary direction .....	94
$\nabla \cdot, \text{div}$	divergence .....	84
$\mathcal{E}(M, x)$	unit ball of the metric $M$ , centered on $x$ .....	58
$\epsilon$	voxelization precision .....	171
$\epsilon_T$	non conformity coefficient of a simplex $T$ .....	18
$\epsilon_{\mathcal{T}}$	non conformity coefficient of a topology $\mathcal{T}$ .....	18
$\epsilon(v)$	strain rate tensor .....	84
$\eta$	viscosity .....	84
$\eta_T$	residual error indicator on element $T$ .....	93
$\mathcal{F}$	face set (faces of elements of $\mathcal{T}$ ) .....	15
$f(M)$	elliptic interpolation function .....	60

$f_\omega$	regularized characteristic function of a subdomain $\omega$ .....	103
$F_\omega$	graph of $f_\omega$ .....	103
$\bar{\gamma}$	acceleration norm .....	86
$g_\omega$	$P^0$ discontinuous characteristic function of a subdomain $\omega$ .....	105
$\nabla g$	$P^0$ gradient, where $g$ is a $P^0$ discontinuous field .....	106
$H$	Hessian tensor	
$\overline{\overline{H}}$	$H$ tensor coefficients reorganized in matrix form .....	61
$H_{2i}(x)$	$2i$ -th order tensor with zero trace .....	64
$h_i$	mesh size in $i$ -th direction .....	12
$h_{\min}(T)$	minimal height of a simplex $T$ .....	94
$h_T$	size of a $T$ .....	15
$h_{T,\vec{d}}$	simplex $T$ size in direction $\vec{d}$ .....	94
$h_T^{\text{optimal}}$	optimal mesh size .....	91
$h(x)$	length distribution function .....	63
$h_M$	$L^2$ averaged size computed with metric $M$ .....	39
$i, j$	integer indices	
$I$	identity matrix with $d$ columns and $d$ rows	
$[0, T]$	time interval .....	34
$[\epsilon(v)]_T^F$	jump of $\epsilon(v)$ across a face $F$ from the element $T$ .....	13
$k$	order (of neighborhood, for instance)	
$K$	final order .....	70
$k_{\min}, k_{\max}$	minimal and maximal growing order .....	68
$k(S)$	number of local element layers .....	168
$\lambda$	eigenvalue .....	12
$\lambda_{\max}$	maximal eigenvalue .....	108
$\lambda_T^F$	weight of a face $F$ in the computation of the residual on element $T$ .....	93
$m$	number of desired element layers at the interface between subdomains ...	103
$M$	metric or metric field .....	11
$\vec{M}$	vector of $\mathbb{R}^{d^2}$ whose components are coefficients of $M$ 's rows .....	61
$m_1(e, T)$	matching function .....	94
$M_1, M_2, M_3, M_4$	elliptic interpolation metrics .....	67, .....
$M_1 \cap M_2$	(approximated) intersection of two metrics $M_1$ and $M_2$ .....	65, .....
$M_1 < M_2$	relationship order between metrics $M_1$ and $M_2$ .....	66, .....
$M_{\text{a posteriori}}$	a posteriori metric .....	150
$M^\beta$	metric $M$ to the power $\beta$ .....	58
$\max(M_1, M_2)$	(coefficient by coefficient) maximum of two metrics $M_1$ and $M_2$ .....	86
$ T $	measure of a simplex $T$ .....	12
$ T _M$	simplex $T$ measure with the metric $M$ .....	151
$ \Omega $	measure of a computational domain $\Omega$ .....	13
$M_{\text{multidomain}}$	multidomain metric .....	17
$M_{\text{natural}}$	natural metric .....	107
$M_{\text{natural}}^n$	partially divided natural metric .....	70
$M_\omega$	metric associated to a subdomain $\omega$ .....	71
$M_{\text{optimal}}$	optimal metric .....	104
$M(S, k)$	$k$ -th order elliptic interpolation metric at node $S$ .....	92
$M_T$	metric associated to a simplex $T$ .....	68
$M(T)$	averaged metric on a simplex $T$ .....	14
$M'_T$	averaged value of metric $M$ on a simplex $T$ .....	39
		18

$\mu_k$	optimal constant for $M_2$ .....	65
$n$	number of desired element layers through the thickness .....	71
$N$	number of nodes in the mesh .....	
$\vec{n}$	outgoing unitary normal .....	156
$n_{\text{average}}$	averaged number of element layers through the thickness .....	168
$\mathcal{N}_k(S)$	$k$ -th order neighborhood of a node $S$ .....	59
$\tilde{\mathcal{N}}_k(S)$	filtered $k$ -th order neighborhood of a node $S$ .....	59
$\  \cdot \ $	Euclidian canonical norm .....	
$\  \cdot \ _M$	norm associated to metric $M$ .....	11
$\  \epsilon(v) \ $	equivalent strain rate .....	86
$\mathcal{N}, \mathcal{S}$	node set .....	15
$\mathcal{N}(S)$	node $S$ neighborhood .....	30
$\mathcal{N}(S, S')$	common neighborhood of both nodes $S$ and $S'$ .....	31
$N_T$	size matrix of an element $T$ .....	94
$\mathcal{O}$	variety (in fact $\Omega$ or $\partial\Omega$ ) .....	15
$\omega$	subdomain .....	97
$\Omega$	computational domain .....	11
$\overline{P}$	transposition matrix .....	62
$p$	pressure .....	84
$R$	rotation formed by eigenvectors of metric $M$ .....	12
$\rho(T)$	inner radius of a simplex $T$ .....	15
$R_T$	residual on element $T$ .....	18
$S$	node .....	
$S_0, \dots, S_d$	vertices of simplex $T$ .....	13
$\langle \cdot, \cdot \rangle_M$	scalar product associated with a metric $M$ .....	11
$S^{d-1}$	unitary ball of $\mathbb{R}^d$ .....	63
$T$	simplex, element .....	13
$T^0$	unitary equilateral simplex .....	14
$\hat{T}$	reference simplex .....	13
$\tau$	shear stress .....	127
$\mathcal{T}$	element set, topology .....	15
$\mathcal{T}_1 < \mathcal{T}_2$	relationship order between topologies .....	40
$\mathcal{T}_c$	local topology to be cut .....	30
$\mathcal{T}_e$	candidate topology to be paste .....	30
$\mathcal{T}_f$	exterior topology around $\mathcal{T}_c$ .....	31
$\mathcal{T}_k^{\text{ext}}(S)$	$k$ -th order exterior topology around a node $S$ .....	59
$\mathcal{T}_k(S)$	$k$ -th order topology built by star operation from a node $S$ .....	66
$\mathcal{T}_\omega$	subtopology of a subdomain $\omega$ .....	170
$\mathcal{T}(S)$	topology around a node $S$ (elements whose $S$ is a vertex) .....	92
$\mathcal{T}^*(S, \mathcal{F})$	star operator from a node $S$ to a face set $\mathcal{F}$ .....	25
$u$	scalar field .....	84
$U, D, L$	upper triangular part, diagonal part and lower triangular part .....	61
$v$	velocity .....	84
$\mathcal{V}(S)$	neighborhood of a node set $\mathcal{S}$ .....	59
$v_{\text{wall}}$	wall velocity .....	127
$x, y$	vectors from $\mathbb{R}^d$ .....	
$x \otimes y$	tensorial product $(x_i y_j)_{1 \leq i, j \leq d}$ .....	



# List of Figures

1	Comparisons between experimental short shots (in white on black) and Rem3D computation	4
2	The overall mesh generation and adaption process	5
1.1	Forbidden situations for a mesh	16
1.2	$\{S_1, \dots, S_6\}$ are the neighbors of $S$ (a) and a mesh whose boundary is not a mesh (b), because of condition <b>(i)</b> violation (a point, which is a face of the 1D boundary mesh, shares 4 elements)	16
2.1	Quadtree (from [Frey and George 1999])	20
2.2	Advancing front in 2D (from [Frey and George 1999])	21
2.3	Voronoi diagram of a set of points and corresponding Delaunay triangulation in 2D (from [Frey and George 1999])	22
2.4	Delaunay method in 2D: triangulation of the convex hull (a) and recovering of the boundary (b) (from [Ern and Guermond 2002])	24
2.5	Mesh generation by topological optimization in 2D	30
2.6	The local topology $\mathcal{T}_c$ around a node $S$ (a) and two candidates for $\mathcal{T}_e$ (b-c).	30
2.7	The local topology $\mathcal{T}_c$ around an edge $[S, S']$ (a) and three candidates for $\mathcal{T}_e$ (b-d).	31
2.8	$\mathcal{T}_c$ is in normal lines, $\partial\mathcal{T}_c$ in bold lines, $\mathcal{T}_e$ in dotted lines and $\mathcal{T}_f$ in dashed lines	32
2.9	A mesh topology without boundary, obtained by connecting old boundary faces to the node 0, so as to form virtual elements (in dashed lines)	33
3.1	Constrained advancing front in 2D (from [Ern and Guermond 2002])	36
3.2	Anisotropic mesh adaption by an hybrid Delaunay / advancing front method, the initial mesh on the left and the final mesh (after 8 iterations) on the right, from [Alauzet 2003] (bottom figures are sectional views)	38
3.3	Iterative adaption process on a 2D anisotropic test case	43
3.4	Convergence history for the 2D anisotropic test case	44
3.5	Enlargement of the convergence curves of figure 3.4	44
3.6	Iterative adaption process on a 2D isotropic test case	45
3.7	Convergence history for the 2D isotropic test case	46
3.8	Enlargement of the convergence curves of figure 3.7	46
3.9	Iterative adaption process on a 3D anisotropic test case	48
3.10	Convergence history for the 3D anisotropic test case	49
3.11	Enlargement of the convergent curves of figure 3.10	49
3.12	Comparison between different averages for the 2D anisotropic test case	50
3.13	Complex geometry that could not be decomposed in simple patterns (courtesy of Schneider Electric)	53
4.1	Thickness refining (from [Garimella 1998])	55
4.2	A geometry that can be divided in plates and cylinders (courtesy of Plastic Omnium)	56
4.3	Mesh obtained with an analytical metric field	56
4.4	Curved industrial geometry (courtesy of Atofina)	57
4.5	Notations for the natural metric definition	58
4.6	Three successive neighborhoods around a node $S$ .	59

4.7	Notations for the length distribution function . . . . .	63
4.8	Vectors $S' - C_3(S)$ . . . . .	65
4.9	Star operating with the hull of order $k = 3$ . . . . .	66
4.10	Test case mesh containing 96 033 nodes (courtesy of Schneider Electric) . . . . .	67
4.11	2D and 3D meshes for determining a lower bound of the elliptic interpolation error threshold	68
4.12	2D and 3D meshes for determining an upper bound of the elliptic interpolation error threshold . . . . .	69
4.13	Initial mesh (a) and first iterations (courtesy of Sophysa), with a sectional view (f) of step (e) . . . . .	72
4.14	Convergence history of the iterative process introducing several element layers through the thickness . . . . .	73
4.15	Initial mesh (a) and first iterations (academic geometry) . . . . .	74
4.16	Spiral mold (courtesy of Atofina) . . . . .	75
4.17	Mesh with several element layers through the thickness, but with few element layers through the width . . . . .	76
4.18	Mesh with several element layers through the thickness and through the width . . . . .	76
4.19	Natural mesh of a Plastic Omnium part (87 054 nodes) . . . . .	77
4.20	The same natural mesh (sectional view through the thickness) . . . . .	78
4.21	Correct transition between a vertical and a $45^\circ$ oriented anisotropy . . . . .	79
4.22	Natural mesh of a curved and thin geometry (sectional view) . . . . .	79
4.23	Manual correction of the natural metric in the vicinity of an injection gate (sectional view)	80
4.24	Natural mesh (174 759 nodes) of figure 3.13's part . . . . .	81
4.25	Problems encountered by the a priori natural metric (each mesh is obtained after 16 iterations) . . . . .	82
5.1	Boundary conditions . . . . .	85
5.2	Velocity (vectors) and pressure (color scale) . . . . .	85
5.3	A natural mesh to be improved (1 327 nodes) . . . . .	87
5.4	Acceleration values $\bar{\gamma}$ on several cutting planes . . . . .	88
5.5	Equivalent shear rate values $\ \epsilon(v)\ $ on the same cutting planes . . . . .	88
5.6	Mesh obtained with the acceleration heterogeneity (920 nodes) . . . . .	89
5.7	Mesh obtained with a natural metric correction by acceleration heterogeneity (5 686 nodes)	90
5.8	A multidomain configuration: mold coupling with thermal regulation (courtesy of Sophysa and Transvalor) . . . . .	98
6.1	Three meshing strategies for a multidomain simulation . . . . .	99
6.2	A bidomain mesh (18 714 nodes) with an exact interface (sectional view) . . . . .	102
6.3	Function $f_\omega$ with $d(x, \partial\omega) = \ x - c\ _2 - r$ . . . . .	103
6.4	Element sampling and notations for the $P^0$ gradient . . . . .	106
6.5	Two overlapping subdomains (equivalent to three subdomains), involving two triple points	107
6.6	Over-molding case, including a pierced plate . . . . .	108
6.7	Ill interface capturing . . . . .	108
6.8	Good interface capturing after that coefficient $m$ has been tuned . . . . .	109
6.9	Iterative process of the anisotropic adaption to a multidomain metric . . . . .	110
6.10	Convergence history of the multidomain metric iterative process . . . . .	111
6.11	Convergence of the number of element layers, for the same case . . . . .	112
6.12	24-th iteration mesh . . . . .	112
6.13	Approximated characteristic function $g_\omega$ , interpolated on the 24-th iteration mesh . . . . .	113
6.14	Sub-meshes of the 24-th iteration (extraction is performed by filtering $g_\omega$ ) . . . . .	114
7.1	Validation test case for the fuzzy interface approach . . . . .	116
7.2	Sectional view of the bidomain mesh (69 201 nodes) . . . . .	116
7.3	Flow front (surface of the 50% filling ratio levelset) at timesteps $t=1.5$ s, 3 s, 4.5 s et 6 s .	117
7.4	Temperature field ( $^\circ\text{C}$ ) on the same cutting plane at $t=3$ s and 6 s . . . . .	118

7.5	Filling ratio (50% levelset after $P^1$ averaging) during the filling stage . . . . .	120
7.6	Temperature at the end of the filling stage, on the symmetry plane . . . . .	121
7.7	Over-molding: evolution of the fluid filling ratio (red elements are full, blue ones are empty) . . . . .	122
7.8	Parallelepiped computational domain, with a spherical inclusion and imposed shearing boundary conditions (arrows) . . . . .	123
7.9	Sectional view of the mesh, after interface capturing by a multidomain metric . . . . .	124
7.10	Velocity field on a cutting plane . . . . .	124
7.11	Some particle tracing around the inclusion . . . . .	124
7.12	Cylindrical sample for compression tests . . . . .	125
7.13	Compression test between two rigid plates . . . . .	126
7.14	Another sample with 50 grains . . . . .	126
7.15	Natural and monodomain mesh, with a boundary layer generated by a multidomain metric . . . . .	128
7.16	The same mesh with new enlargements, so as to see the adiabatic effect . . . . .	128
7.17	Extrusion die (100 mm diameter), with a 20 mm large exit gate and a 1 mm thick boundary layer (83 863 nodes) . . . . .	129
7.18	Clay filling ratio: red elements are full, blue one are empty (full of water) . . . . .	130
7.19	Velocity field (arrows) and pressure field (color scale) on a symmetry plane . . . . .	131
7.20	Velocity field (arrows) and clay filling ratio (color scale) with enlargement of the dead zone . . . . .	131
7.21	A coarse mesh defining the subdomain geometry (this mesh is used for the voxelization) . . . . .	132
7.22	Bidomain mesh obtained for the biplane . . . . .	133
7.23	Boundary mesh of $\Omega \setminus \omega$ , seen from the inside . . . . .	134
7.24	A noisy boundary obtained by element extraction and a smoother levelset . . . . .	135
7.25	Quarter of an optical lens and its surrounding mold (courtesy of Essilor) . . . . .	140
A.1	Classical averages between two metrics . . . . .	148
A.2	$L^p$ average of two metrics . . . . .	149
A.3	Approximated intersection between two metrics . . . . .	150
A.4	Maximal metric . . . . .	151
A.5	Mesh of a plate with a Cartesian metric . . . . .	152
A.6	Thick tube mesh obtained with a cylindrical metric . . . . .	153
A.7	Mesh of a lens with a spherical metric (courtesy of Essilor) . . . . .	155
A.8	Outgoing normal . . . . .	156
A.9	Equilateral tetrahedron ( $d = 3$ ) . . . . .	158
B.1	Sub-mesh of a subdomain . . . . .	170
B.2	Coarse mesh of $\omega$ and its voxelization. . . . .	172
C.1	Picture of the real part . . . . .	173
C.2	Natural mesh of the Plastic Omnium geometry . . . . .	174
C.3	Numerical short shots (organized in column) . . . . .	176
C.4	Pressure history at the injection gate . . . . .	177
C.5	Pressure field (in MPa) at 2.75 s . . . . .	177
C.6	Velocity vector field during the filling process (the flow front is in green) . . . . .	178
C.7	Temperature (in °C) at 3 s . . . . .	179
C.8	Picture of one part side (in this rear view, the injection gate is on the left) . . . . .	180
C.9	Partial view around the weld line . . . . .	180
C.10	Numerical short shots around the weld line . . . . .	181
C.11	Temperature (in °C) on a cutting plane (inside the side plate) at different time steps (truncated scale) . . . . .	182



# List of Tables

2.1	Comparison between mesh generation techniques . . . . .	34
4.1	Comparisons between four elliptic interpolation methods . . . . .	67
4.2	Elliptic interpolation error at different neighborhood orders . . . . .	69
A.1	Mesh of the boundary (left columns) and the interior mesh (right columns) of a 4D hypercube	162
A.2	Node coordinates of the 4D hypercube mesh, adapted to a mesh size of $1/3$ . . . . .	163
A.3	Pentatope vertices of the same mesh . . . . .	164
A.4	Results about 4D adaption on a hypercube . . . . .	166
C.1	Summary of simulation characteristics . . . . .	174



# Bibliography

- [Abbad 2003] H. Abbad. Simulation numérique d'un procédé d'extrusion de tuiles et briques. CEMEF report, 2003.  
 .....127
- [Advani and Tucker 1987] S. Advani and C. Tucker. The use of tensors to describe and predict fiber orientation in short fiber composites. *Journal of Rheology*, 1987, 31, p. 751-784.  
 .....65
- [Aguilar-Villegas 2000] J.-C. Aguilar-Villegas. *Anisotropic adaptive refinement algorithms for finite element methods*. Ph. D. Thesis, New York University, 2000.  
 .....114
- [Alauzet 2003] F. Alauzet. *Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en Mécanique de Fluides*. Ph. D. Thesis, Université des Sciences et Techniques du Languedoc, 2003.  
 .....38, 187
- [Alauzet et al. 2003] F. Alauzet, P. Frey and B. Mohammadi. *Anisotropic mesh adaptation. Application to transient CFD problem*. In: Bathe, K. J. eds. Proceedings of the Second MIT Conference on Computational Fluid and Solid Mechanics, Boston, 17-20 juin 2003.  
 .....53, 63, 83
- [Barboza et al. 2002] J. Barboza, L. Fourment and J.-L. Chenot. *Contact algorithm for 3D multi-bodies problems: application to forming of multi-material parts and tool deflection*. In: 5th World Congress on Computational Mechanics, Vienne, 2002.  
 .....97, 99
- [Batkam 2002] S. Batkam. *Thermique multidomaines en simulation numérique du remplissage 3D*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2002.  
 .....3, 4, 97, 100, 115
- [Batkam and Coupez 2001] S. Batkam and T. Coupez. *A new multi-domain approach for the thermal problem solution in 3D mold filling*. In: ESAFORM Conference on Material Forming, Liège, 2001, p. 31-34.  
 .....97
- [Baker 2000] T. Baker. *Deformation and quality measures for tetrahedral meshes*. In: European Congress on Computational Methods in Applied Sciences and Engineering, Barcelone, septembre 2000.  
 .....15
- [Belhamadia et al. 2004] Y. Belhamadia, A. Fortin and É. Chamberland. Anisotropic mesh adaptation for the solution of the Stefan problem. *Journal of Computational Physics*, 2004, 194, p. 233-255.  
 .....83

- [Ben Belgacem *et al.* 1995] F. Ben Belgacem, P. Hild and P. Laborde. The mortar finite element method for contact problems. *Journal Math. Comput. Modelling*.  
 ..... 100
- [Bigot 2001] E. Bigot. *Simulation tridimensionnelle du remplissage de corps mince par injection*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2001.  
 ..... 3, 93, 94, 119
- [Bonet and Peraire 1991] J. Bonet and J. Peraire. An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *International Journal for Numerical Methods in Engineering*, 1991, 31, p. 1-17.  
 ..... 20
- [Borouchaki and George 2000] H. Borouchaki and P.-L. George. *Adaptive quality mesh generation in three dimension*. In: European Congress on Computational Methods in Applied Sciences and Engineering, Barcelone, 11-14 septembre 2000.  
 ..... 37
- [Bouchard *et al.* 2000] P.-O. Bouchard, F. Bay, Y. Chastel and I. Tovenar. Crack propagation modelling using an advanced remeshing technique. *Computer Methods in Applied Mechanics and Engineering*, 2000, 189, p. 723-742.  
 ..... 97
- [Boussetta 2005] R. Boussetta. *Remaillage adaptatif tridimensionnel piloté par estimation d'erreur a posteriori*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, prévue pour 2005.  
 ..... 91
- [Bottasso 2003] C. Bottasso. On the choice of the optimization strategy and of the objective function for metric-driven mesh adaptation. In: Wiberg, N.-E. and Díez, P. eds. *Adaptive Modeling and Simulation*, Göteborg, 29 septembre - 1<sup>er</sup> octobre 2003. Barcelone : International Center for Numerical Methods in Engineering (CIMNE).  
 ..... 47
- [Bottasso 2004] C. Bottasso. Anisotropic mesh adaption by metric-driven optimization. *International Journal for Numerical Methods in Engineering*, 2004, 60 (3), p. 597-639.  
 ..... 39, 40, 41, 47
- [Bruchon 2004] J. Bruchon. *Étude de la formation d'une structure de mousse par simulation directe de l'expansion de bulles dans une matrice liquide polymère*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2004.  
 ..... 3, 4
- [Bruchon and Coupez 2003] J. Bruchon and T. Coupez. A study of the 3D polymer foam formation based on the simulation of anisothermal bubble expansion. *Mécanique & Industries*, 2003, 4, p. 331-338.  
 ..... 97
- [Castro-Díaz and Hecht 1995] M. Castro-Díaz and F. Hecht. Anisotropic surface mesh generation. INRIA report, n° 2672, 1995.  
 ..... 37

- [Castro-Díaz *et al.* 1996] M. Castro-Díaz, H. Borouchaki, P. L. George, F. Hecht and B. Mohammadi. *Anisotropic adaptive mesh generation in two dimensions for CFD*. In: Computational Fluid dynamics, 1996, p. 181-186.  
..... 140, 150
- [Castro-Díaz *et al.* 1997] M. Castro-Díaz, F. Hecht, B. Mohammadi and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids*, 1997, 25, p. 475-491.  
..... 127, 140, 150
- [Coupez 1991] T. Coupez. *Grandes transformations et remaillage automatique*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 1991.  
..... 9, 19, 25, 34
- [Coupez 1994] T. Coupez. A mesh improvement method for 3D automatic remeshing. *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, 1994, p. 615-625.  
..... 9
- [Coupez 2000] T. Coupez. Génération de maillage et adaptation de maillage par optimisation locale. *Revue européenne des éléments finis*, 2000, 9, p. 403-423.  
..... 25, 28, 31, 39
- [Coupez and Bigot 2000] T. Coupez and E. Bigot. *3D anisotropic mesh generation and adaptation with applications*. In: European Congress on Computational Methods in Applied Sciences and Engineering, Barcelone, 11-14 septembre 2000.  
..... 9, 25, 93, 94
- [Coupez and Baranger] T. Coupez and J. Baranger. Convection et diffusion pour des éléments finis discontinus P0. CEMEF report.  
..... 106
- [Coupez *et al.* 2000] T. Coupez, H. Dignonnet and R. Ducloux. Parallel meshing and remeshing. *Applied Mathematical Modelling*, 2000, 25, p. 153-175.  
..... 34
- [Coupez *et al.* 2004] T. Coupez, C. Gruau, C. Pequet and J. Bruchon. *Metric map and anisotropic mesh adaptation for static and moving surfaces*. In: World Congress on Computational Mechanics IV, Beijing, 5-10 septembre 2004. Tsinghua University Press & Springer-Verlag.  
..... 97, 119, 141
- [Creusé *et al.* 2003] E. Creusé, G. Kunert and S. Nicaise. A posteriori error estimation for the Stokes problem: Anisotropic and isotropic discretizations. *SFB 393 Preprint*, Technische Universität Chemnitz, 2003.  
..... 93
- [Daboussy 2000] D. Daboussy. *Calculs 3D multi-fluides appliqués à la co-injection et à l'injection assistée gaz*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2000.  
..... 4
- [Dimakis and Müller-Hoissen 1999] A. Dimakis and F. Müller-Hoissen. Discrete Riemannian geometry. *Journal of Mathematical Physics*, 1999, 40 (3).  
..... 141

- [do Carmo 1988] do Carmo, M. P. *Riemannian geometry*. Birkhäuser, 1988.  
 ..... 13, 104
- [Dompierre *et al.* 2002] J. Dompierre, M.-G. Vallet, Y. Bourgault, M. Fortin and W. Habashi. Anisotropic Mesh Adaptation: Towards User-Independent, Mesh-Independent and Solver-Independent CFD. Part III: Unstructured Meshes. *International Journal for Numerical Methods in Fluids*, 2002, 39, p. 675-702.  
 ..... 39
- [Dompierre *et al.* 2003] J. Dompierre, M.-G. Vallet, P. Labbé and F. Guibault. *On simplex shape measures with extension for anisotropic meshes*. In: Proceedings of Workshop on Mesh Quality and Dynamic Meshing, Livermore, janvier 2003, p. 46-71.  
 ..... 14, 15, 18, 40
- [Dobrzynski *et al.* 2004] C. Dobrzynski, O. Pironneau and P. Frey. *Numerical coupling for air flow computations in complex architectures*. In: European Congress on Computational Methods in Applied Sciences and Engineering, Jyväskylä, juillet 2004.  
 ..... 53
- [Ern and Guermond 2002] A. Ern J.-L. Guermond. *Éléments finis: théorie, applications, mise en œuvre*. Springer, 2002.  
 ..... 24, 36, 187
- [Fournier 2003] R. Fournier. *Optimisation et modélisation du procédé R.T.M.* Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2003.  
 ..... 3, 97
- [Frey and Field 1991] W. Frey and D. Field. Mesh relaxation: a new technique for improving triangulations. *International Journal for Numerical Methods in Engineering*, 1991, 31, p. 1121-1133.  
 ..... 25
- [Frey and George 1999] P. Frey and P.-L. George. *Maillages - applications aux éléments finis*. Paris : Hermès, 1999.  
 ..... 20, 21, 22, 187
- [Garimella 1998] R. Garimella, *Anisotropic tetrahedral mesh generation*. Thèse Doct., Rensselaer Polytechnic Institute, 1998.  
 ..... 55, 187
- [Garimella and Shephard 1998] R. Garimella and M. Shephard. *Boundary layer meshing for viscous flows in complex domains*. In: 7th International Meshing Roundtable, Dearborn, 1998, p. 107-118.  
 ..... 55, 127
- [George and Borouchaki 1997] P.-L. George and H. Borouchaki. *Triangulation de Delaunay et maillage - applications aux éléments finis*. Paris : Hermès, 1997.  
 ..... 22, 43, 45
- [George and Borouchaki 2002] P.-L. George and H. Borouchaki. Premières expériences de maillage automatique par une méthode de Delaunay anisotrope en trois dimensions. INRIA report, n° 272, 2002.  
 ..... 36, 37

- [Gobeau 1996] J.-F. Gobeau. *Étude expérimentale et simulation numérique 3D par éléments finis de l'écoulement dans les filières d'extrusion de profilé PVC*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 1996.  
..... 56
- [Gruau and Coupez 2003] C. Gruau and T. Coupez. *Anisotropic and multidomain mesh automatic generation for viscous flow finite element method*. In: Wiberg, N.-E. and Díez, P. eds. *Adaptive Modeling and Simulation*, Göteborg, 29 septembre - 1<sup>er</sup> octobre 2003. Barcelone : International Center for Numerical Methods in Engineering (CIMNE).  
..... 141
- [Gruau and Coupez 2005] C. Gruau and T. Coupez. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. To appear in *Computer Methods in Applied Mechanics and Engineering*.  
..... 141
- [Gruau and Coupez 2004] C. Gruau and T. Coupez. *Génération de maillages 3D non structurés et anisotropes avec adaptation aux métriques naturelle et multidomaine*. In: 36<sup>ème</sup> Congrès National d'Analyse Numérique, Obernai, 31 mai - 4 juin 2004.  
..... 141
- [Halír and Flusser 1998] R. Halír and J. Flusser. *Numerically Stable Direct Least Squares Fitting of Ellipses*. In: Winter School of Computer Graphics, Plzen-Bory, 1998.  
..... 60
- [Henry 2004] F. Henry. Classe écran. CEMEF report, 2004.  
..... 142
- [Hoch and Rasclé 2003] P. Hoch and M. Rasclé. Hamilton-Jacobi equations on a manifold and applications to grid generation or refinement. *SIAM Journal on Scientific Computing*, 2001, 23 (6), p. 2056-2074.  
..... 104
- [Jacquotte 1989] O. Jacquotte. Génération, optimisation et adaptation de maillages multidomaines autour de configurations complexes. In: AGARD 64th FDP, Loen, 24-25 mai 1989.  
..... 100
- [Janka 2002] A. Janka. Development of multigrid methods for convergence acceleration of solvers for Navier-Stokes equations on non-structured meshes. *Second work-in-progress report*, INRIA, Dassault contract no. 42G01441, ref. AW/MIT, 2002.  
..... 41, 66, 77
- [Jansen and Shephard 2001] K. Jansen and M. Shephard. On Anisotropic Mesh Generation and Quality Control in Complex Flow Problems. In: Proceedings of 10th International Meshing Roundtable, Sandia National Laboratories, 7-10 octobre 2001, p. 341-349.  
..... 127
- [Jonas and Louis 2001] P. Jonas and A. Louis. A Sobolev Space Analysis of Linear Regularization Methods for Ill-Posed Problem. *Journal of Inverse Ill-Poses Problems*, 2001, 9 (1), p. 59-74.  
..... 61
- [Jost 1998] J. Jost. *Riemannian geometry and geometric analysis* Springer-Verlag, 1998.  
..... 13, 104

- [Kaufman 2004] A. Kaufman. *Voxels as computational representation of geometry*. In: The Computational Representation of Geometry. SIGGRAPH '94 Course Notes, 1994.  
.....171
- [Knockaert 2001] R. Knockaert. *Etude expérimentale et numérique de la localisation de la déformation lors de la mise en forme de produits minces*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2001.  
.....55
- [Knupp 2001] P. Knupp. Algebraic mesh quality metrics. *SIAM Journal on Scientific Computing*, 2001, 23 (1), p. 193-218.  
.....15
- [Kunert 1998] G. Kunert. *A posteriori error estimation for anisotropic tetrahedral and triangular finite element meshes*. Thèse Doct. rerum naturalium, Fakultät für Mathematik der Technischen Universität Chemnitz, 1998.  
.....93
- [Kunert 1999] G. Kunert. *A posteriori error estimation for anisotropic tetrahedral and triangular finite element meshes*. Ph. D. Thesis, Fakultät für Mathematik der Technischen Universität Chemnitz, 1999.  
.....93
- [Kunert 2001] G. Kunert. A local problem error estimator for anisotropic tetrahedral finite element meshes. *SIAM Journal on Numerical Analysis*, 2001, 39 (2), p. 668-689.  
.....93
- [Labbé *et al.* 2000] P. Labbé, J. Dompierre, M.-G. Vallet, F. Guilbault and J.-Y. Trépanier. *Measuring the metric conformity of a mesh*. In: 7th International Conference on Numerical Grid Generation in Computational Field Simulations, Whistler, 25-28 septembre 2000.  
.....18
- [Labbé *et al.* 2001] P. Labbé, J. Dompierre, M.-G. Vallet, F. Guilbault and J.-Y. Trépanier. *A measure of the conformity of a mesh to an anisotropic metric*. In: 10th International Meshing Roundtable, Newport Beach, 7-10 octobre 2001.  
.....18
- [Leservoisier *et al.* 2001] D. Leservoisier, P.-L. George and A. Dervieux. Métrique continue et optimisation de maillage. INRIA report, n° 4172, 2001.  
.....63, 83
- [Lesoinne and Farhat 1996] M. Lesoinne and C. Farhat. Geometric conservation law for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 1996, 134, p. 71-90.  
.....99
- [Liu 2004] W. Liu. Mesh adaptation in R2SOL. CEMEF report, 2004.  
.....84
- [Lo 1985] S. Lo. A new mesh generation scheme for arbitrary planar domains. *International Journal For Numerical Methods In Engineering*, 1985, 21, p. 1403-1426.  
.....20

- [Löhner 1988] R. Löhner. An adaptive finite element solver for transient problems with moving bodies. *Computers & Structures*, 1988, 30 (1), p. 303-317.  
..... 9
- [Löhner 1989] R. Löhner. Adaptive remeshing for transient problems. *Computer Methods in Applied Mechanics and Engineering*, 1989, 75, p. 195-214.  
..... 9
- [Löhner *et al.* 1999] R. Löhner, C. Yang, E. Oñate and S. Idelsohn. An unstructured grid-based, parallel free surface solver. *Applied Numerical Mathematics*, 1999, 31, p. 271-293.  
..... 97
- [Magnin 1994] B. Magnin. *Modélisation du remplissage des moules d'injection pour polymères thermoplastiques par une méthode eulérienne-lagrangienne arbitraire*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 1994.  
..... 3
- [Maman and Farhat 1995] N. Maman and C. Farhat. Matching fluid and structure meshes for aeroelastic computations: a parallel approach. *Computers & Structures*, 1995, 54 (4), p. 779-785.  
..... 100
- [Megally *et al.* 2004] A. Megally, P. Laure, T. Coupez and M. Vincent. *Direct calculation of the motion of rigid fibres in a viscous fluid*. In: 8th International Conference on Numerical Methods in Industrial Forming Processes, Columbus, 13-17 juin 2004.  
..... 4, 97
- [Ménard 2003] P. Ménard. Étude numérique avec REM3D® du comportement d'un corps plongé dans un liquide cisailé. CEMEF report, 2003.  
..... 123
- [Meskini 2004] N. Meskini. Le calcul de microstructures - développement du logiciel Digimat. CEMEF report, 2004.  
..... 77
- [Mohammadi *et al.* 2000] B. Mohammadi, P.-L. George, F. Hecht and E. Saltel. 3D mesh adaptation by metric control for CFD. *Revue européenne des éléments finis*, 2000, 9, p. 439-449.  
..... 150
- [Nouatin 2000] O. Nouatin. *Méthode et analyse de simulation numérique d'écoulements 3D des polymères fondus - identification de paramètres rhéologiques viscoélastiques par analyse inverse*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2000.  
..... 3
- [Onat 1982] E. Onat. *Representation of inelastic behavior in the presence of anisotropy and of finite deformations*. In: B. Wilshire and D. Owen, eds., Pineridge Press, Swansea, p. 231-264.  
..... 64
- [Onat and Leckie 1988] E. Onat and F. Leckie. Representation of mechanical behavior in the presence of changing internal structure. *Journal of Applied Mechanics*, 1988, 55.  
..... 64
- [Pain *et al.* 2001] C. Pain, A. Umpheby, C. de Oliveira and A. Goddard. Tetrahedral mesh optimisation and adaptativity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 2001, 190, p. 3771-3796.  
..... 39, 41, 53, 63, 83

- [Pébay and Baker 2001] P. Pébay and T. Baker. *A comparison of triangle quality measures*. In: 10th International Meshing RoundTable, Sandia National Laboratory, 2001, p. 327-340.  
..... 15
- [Peraire and Morgan 1997] J. Peraire and K. Morgan. Unstructured mesh generation including directional refinement for aerodynamic flow simulation. *Finite Elements in Analysis and Design*, 1997, 25, p. 343-355.  
..... 55
- [Peraire *et al.* 1987] J. Peraire, M. Vahdati, K. Morgan and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 1987, 72, p. 449-466.  
..... 9
- [Pichelin 1998] É. Pichelin. *Calcul par éléments finis du remplissage 3D pour des fluides visqueux incompressible. Application à l'injection*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 1998.  
..... 3
- [Redjeb 2003] A. Redjeb. Simulation numérique de l'orientation des fibres en injection de thermoplastique renforcé. CEMEF report, 2003.  
..... 3
- [Remacle *et al.* 2002] J.-F. Remacle, X. Li, N. Chevaugéon and M. Shephard. *Transient mesh adaptation using conforming and non conforming mesh modifications*. In: 11<sup>th</sup> International Meshing Roundtable, Ithaca, 2002.  
..... 39, 41, 53, 63, 83
- [Rey 2003] B. Rey. Simulation numérique du comportement mécanique de matériaux hétérogènes : prise en compte explicite de la topologie. CEMEF report, 2003.  
..... 125
- [Rinaldi-Mareel 2004] D. Rinaldi-Mareel. *Amélioration des propriétés optiques des fibres de verre utilisées en télécommunication*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2004.  
..... 4
- [Saez 2003] E. Saez. *Étude numérique du remplissage 3D en fonderie*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2003.  
..... 3, 4, 127
- [Schroeder *et al.* 2002] W. Schroeder, K. Martin and B. Lorensen. *The visualization toolkit: an object-oriented approach to 3D graphics*. Troisième édition. Kitware, 2002.  
..... 171, 172
- [Silva 2004] L. Silva. *History-dependent fluid flow simulation applied to polymer injection moulding*. Ph. D. Thesis, École Nationale Supérieure des Mines de Paris, 2004.  
..... 3, 78
- [Silva and Coupez 2002] L. Silva and T. Coupez. *A unified model of the filling and post-filling stages in 3D injection moulding simulation*. In: Polymer Processing Society 18, Guimarães, 16-20 juin 2002.  
..... 71

- [Silva *et al.* 2003] L. Silva, M. Barré, H. Digonnet, C. Gruau, A. Rodriguez-Villa, J.-F. Agassant and T. Coupez. *Avancement et perspectives dans la simulation numérique tri-dimensionnelle de l'injection des polymères*. In: 16<sup>ème</sup> Congrès Français de Mécanique, Nice, 1-5 septembre 2003.  
.....141
- [Silva *et al.* 2004] L. Silva, C. Gruau, J.-F. Agassant, T. Coupez and J. Mauffrey. Rem3D : Advanced finite element model for 3D injection molding. Submitted at *International Polymer Processing*.  
.....141
- [Stalický and Roos 1999] T. Stalický and H.-G. Roos. Anisotropic mesh refinement for problems with internal and boundary layers. *International Journal For Numerical Methods In Engineering*, 1999, 46, p. 1933-1953.  
.....114
- [Takahashi *et al.* 2001] S. Takahashi, Y. Kokojima and R. Ohbuchi. *Explicit Controls of Topological Transitions in Morphing Shape of 3D Meshes*. In: Proceedings of the 9th Pacific Conference on Computer Graphics and Applications, Tokyo, octobre 2001, p. 70-79.  
.....161
- [Tam *et al.* 2000] A. Tam, D. Ait-Ali-Yahia, M. Robichaud, M. Moore, V. Kozel and W. Habashi. Anisotropic mesh adaptation for 3D flows on structured and unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 2001, 189, p. 1205-1230.  
..... 39, 41, 53, 63, 83
- [Tchon *et al.* 2003] K.-F. Tchon, M. Khachan, F. Guibault and R. Camarero. *Constructing anisotropic geometric metrics using octrees and skeletons*. In: 12th International Meshing Roundtable, Santa Fe, 2003, p. 293-304.  
..... 19, 71
- [Tezduyar and Osawa 2001] T. Tezduyar and Y. Osawa. The multi-domain method for computation of the aerodynamics of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 2001, 191, p. 705-716.  
.....115
- [Tezduyar *et al.* 1998] T. Tezduyar, S. Aliabadi and M. Behr. Enhanced-Discretisation Interface-Capturing Technique (EDICT) for computation of unsteady flows with interfaces. *Computer Methods in Applied Mechanics and Engineering*, 1998, 155, p. 235-248.  
.....114
- [Vallet 1992] M.-G. Vallet. *Génération de maillages éléments finis anisotropes and adaptatifs*. Ph. D. Thesis, Université Paris 6, 1992.  
..... 9
- [Weisstein *et al.*] E. Weisstein *et al.* *Stokes' Theorem*. MathWorld—A Wolfram Web Resource, [mathworld.wolfram.com/StokesTheorem.html](http://mathworld.wolfram.com/StokesTheorem.html).  
.....17
- [Yerry and Shephard 1983] M. Yerry and M. Shephard. A modified-quadtrees approach to finite element mesh generation. *IEEE Computer Graphics and Applications*, 1983, 3, p. 39-46.  
.....19

- [Zavattieri *et al.* 1996] P. Zavattieri, E. Dari and G. Buscaglia. Optimization strategies in unstructured mesh generation. *International Journal For Numerical Methods In Engineering*, 1996, 39, p. 2055-2071.

.....9

# Index

## Symboles

4D ..... 34

## A

acceleration ..... 86  
 advancing front ..... 20, 36  
   constrained ..... 36  
 air ..... 115  
 ALE ..... 119  
 algorithmic complexity ..... 70, 109  
 approximation  
   error ..... 105  
   tensorial ..... 63  
 averaged  
   metric ..... 148  
   number of element layers ..... 168

## B

ball ..... 147  
 bilinear application ..... 11  
 biplane ..... 132  
 boundary  
   conditions  
     adiabatic ..... 106, 127  
     thermal ..... 97  
   geometrical ..... 17  
   layer ..... 127  
   subdomain ..... 169  
   topological ..... 16  
 bubble ..... 84

## C

candidate ..... 37  
 capillary number ..... 108  
 cavity ..... 29  
   Delaunay ..... 23  
     anisotropic ..... 37  
 cell of Voronoï ..... 22  
 characteristic function ..... 103, 105  
 clay ..... 127  
 coarsening ..... 18, 77  
 cocyclicity ..... 22  
 coefficient  
   non conformity ..... 18  
   normalization ..... 15

  penalty ..... 108  
 coloring ..... 23  
 contraction ..... 64  
 criterion  
   Delaunay ..... 22  
   minimal volume ..... 28  
   quality ..... 40  
   shape ..... 14  
     anisotropic ..... 39  
   size ..... 40  
 curvature ..... 73

## D

decomposition (tensorial) ..... 63  
 detail ..... 80, 132  
 diagonalization ..... 11  
   simultaneous ..... 150  
 diameter ..... 15, 93  
 dimension  
   spatial ..... 15  
   topological ..... 15  
 distance (signed) ..... 103  
 divergence ..... 84  
 division ..... 71  
 domain ..... 11

## E

edge ..... 12  
   swapping ..... 23  
 eigenvalue ..... 12  
 eigenvector ..... 12  
 element ..... 15  
   virtual ..... 33  
 ellipse ..... 147  
 ellipsoid ..... 147  
 equivalent shear rate ..... 86  
 error  
   approximation ..... 105  
   estimator ..... 91  
   indicator ..... 91  
     anisotropic ..... 94  
   interpolation ..... 68  
 extrusion ..... 127  
   die ..... 129  
   mesh ..... 55

**F**

face .....	12
blocking .....	174
boundary .....	170
filling ratio .....	106
levelset .....	117
filter .....	114
friction law .....	127
frictionless contact .....	127
front .....	20, 36

**G**

geometrical detail .....	171
gradient $P^0$ discontinuous .....	106
grain .....	125

**H**

Hessian .....	86
hull .....	59
hypercube .....	161

**I**

ILU(0) .....	86
incremental propagation .....	167
injection gate .....	78
insert .....	119
interface	
capture .....	100
fuzzy .....	99
precise .....	99
interpolation .....	37, 41
discrete elliptic .....	60
discontinuous $P^0$ .....	105
intersection .....	150
iteration process .....	71, 110

**J**

jump .....	93
------------	----

**K**

kernel (Delaunay) .....	23
anisotropic .....	37

**L**

layer	
boundary .....	127
through the thickness .....	71
least-square .....	60
length distribution .....	63
levelset surface .....	117
lexicographical order .....	29, 40, 58
local anisotropy .....	58

localization .....	24, 100
loss of ellipticity .....	69
lubricant .....	127

**M**

measure	
mesh .....	17
simplex .....	13
equilateral .....	159
memory consumption .....	109
mesh .....	15
generation	
Delaunay .....	23
topological .....	9
isotropic .....	35
natural .....	77
size .....	12
minimal .....	42
optimal .....	91
volume .....	17
metric .....	11
a priori .....	53
analytical .....	56
associated .....	14
averaged value .....	148
Cartesian .....	151
cylindrical .....	152
Euclidian .....	35
induced .....	13, 104
intersection .....	150
maximal .....	151
natural .....	58
$P^0$ .....	41
polar .....	154
Riemannian .....	35
spherical .....	154
unit ball .....	58, 147
microscopic scale .....	123
minimization .....	63
mixing law .....	115
multidomain .....	97

**N**

neighborhood .....	16, 161
order $k$ .....	59
Newtonian velocity profile .....	86
node .....	15
0 .....	33
boundary .....	169
fictitious .....	33
isolated .....	170
locked .....	101
norm .....	11

- Frobenius ..... 18  
 normal ..... 156  
 notch ..... 80  
 number of layer ..... 167
- O**
- octree ..... 19  
 order  
   lexicographical ..... 29, 40, 58  
   minimal ..... 167  
 orientation ..... 77, 155  
 orthogonal  
   parallelootope ..... 151  
   projection ..... 104  
 outer ball  
   center and radius ..... 23  
   elliptic ..... 37  
 over-molding ..... 119  
 overlapping ..... 107
- P**
- parallelization ..... 34  
 parallelootope ..... 171  
 peak ..... 72  
 penalization ..... 108  
   diagonal part ..... 61  
 pentatope ..... 12  
 pixel ..... 171  
 polycrystalline ..... 123, 125  
 polygon ..... 11  
 polyhedron ..... 11  
   grain ..... 125  
 polytope ..... 11  
   convex ..... 22  
 pressure ..... 84  
 principal directions ..... 12  
 product  
   scalar ..... 11  
   tensorial ..... 147
- Q**
- quadtree ..... 19
- R**
- R-adaption ..... 119  
 refinement ..... 18  
   systematical ..... 55  
 regularization ..... 107  
   Tikhonov ..... 61  
 relationship order ..... 29, 40, 58  
 residual error ..... 93  
 retro-propagation ..... 167  
 rigid inclusion ..... 123
- robustness ..... 78, 132  
 rule 2 : 1 ..... 20
- S**
- sampling ..... 171  
 security factor ..... 71  
 semi-definite programming ..... 63  
 series  
   Fourier ..... 64  
   orthogonal ..... 64  
 shear stress ..... 127  
 shearing flow ..... 123  
 sigmoid ..... 103  
 signed distance ..... 103  
 simplex ..... 12  
   equilateral ..... 12  
   construction ..... 159  
   height ..... 157  
   measure ..... 159  
   outer radius ..... 157  
   unitary ..... 12  
   measure ..... 13  
   not degenerated ..... 12  
   reference ..... 12  
 size ..... 12  
 skinning ..... 129  
 sliver ..... 23  
 star operator ..... 23, 25, 66  
 state variable ..... 83  
 Stokes equations ..... 84  
 strain rate ..... 84  
   jump ..... 93  
 sub-mesh ..... 101, 170  
 subdomain ..... 97  
   boundary ..... 169  
 surface  
   discontinuity ..... 103  
   median ..... 58
- T**
- temperature gradient ..... 127  
 tensor ..... 147  
 tetrahedron ..... 12  
 thickness ..... 71  
   multiple ..... 75  
 topology ..... 16  
   candidate ..... 30  
   local ..... 30  
   mesh ..... 25  
   reverse ..... 160  
 trace ..... 41  
 transformation ..... 42  
   inverse ..... 42

transposition ..... 62  
tree ..... 19  
triangle ..... 12  
triangulation ..... 22  
truncation ..... 110

## U

unit ball ..... 58  
update ..... 41

## V

validation ..... 42, 115  
variety ..... 15, 103  
velocity ..... 84  
vertex ..... 12  
viscosity ..... 84  
volume  
    mesh ..... 17  
    simplex ..... 13  
Voronoi cell ..... 22, 125  
vortex ..... 123  
voxelization ..... 171  
    precision ..... 110  
    skinning ..... 129

## W

wall velocity ..... 127  
without boundary ..... 25

compiled on 16th June 2008

## Abstract

Computational mechanics can usually encounter three major problems in mesh generation: thin and curved geometry treatment, meshing for a simulation involving several objects and boundary layer improvement.

In this work, we introduce the notion of natural metric in order to automatically place several element layers through the thickness of thin and curved geometries. A suitable anisotropic mesh for numerical simulation is thus obtained by a topological optimization strategy.

When the mesh needs to be refined around interfaces between the different objects involved in a multidomain simulation, we propose a second metric field that tightens the elements around such surfaces, in an anisotropic way.

Furthermore, this multidomain metric can be used in boundary layer generation, which improves meshes obtained with the natural metric. Finally, the anisotropy of a computed field and the uniformization of an a posteriori error indicator can be exploited to correct the natural metric.

All these developments are illustrated on 3D complex parts, coming from material forming, especially from polymer injection molding, using Rem3D.

**Keywords:** unstructured mesh, anisotropic adaption, natural metric, elliptic interpolation, multidomain metric, tensorial product, voxelization, error estimation, Rem3D simulation