

# Generation de maillages par une methode de type Voronoi : partie 2 : le cas tridimensionnel

Paul-Louis George

► **To cite this version:**

Paul-Louis George. Generation de maillages par une methode de type Voronoi : partie 2 : le cas tridimensionnel. [Rapport de recherche] RR-1664, INRIA. 1992. inria-00074893

**HAL Id: inria-00074893**

**<https://hal.inria.fr/inria-00074893>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# INRIA

UNITÉ DE RECHERCHE  
INRIA-ROCQUENCOURT

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P.105  
78153 Le Chesnay Cedex  
France  
Tél.: (1) 39 63 55 11

## Rapports de Recherche

1992



25<sup>ème</sup>

anniversaire

N° 1664

*Programme 6*  
*Calcul Scientifique, Modélisation et*  
*Logiciels numériques*

### GÉNÉRATION DE MAILLAGES PAR UNE MÉTHODE DE TYPE VORONOÏ. PARTIE 2 : LE CAS TRIDIMENSIONNEL

Paul-Louis GEORGE

Avril 1992



\* R R - 1 6 6 4 \*

Rapport - Génération de maillages

Génération de maillages par une méthode de type Voronoï  
Partie 2: Le cas tridimensionnel

Paul Louis George

12 Février 1992

# Génération de maillages par une méthode de type Voronoï Partie 2: Le cas tridimensionnel

Paul Louis George

INRIA Domaine de Voluceau- Rocquencourt BP 105 -78153 LE CHESNAY FRANCE

## Résumé

*Ce papier se propose de faire le point sur les différentes phases nécessaires à la mise en œuvre d'un mailleur automatique de type Voronoï. Il rappelle cette méthode (au moins, pour une des manières de la présenter) et décrit une variante numériquement plus robuste en mentionnant à la fois les problèmes à résoudre et les différentes solutions possibles. Seul le cas tridimensionnel est traité.*

## Voronoi type mesh generation Part 2: The 3D situation

## Abstract

*This paper aims to outline the different phases necessary to implement a Voronoi type automatic mesh generator. It, firstly, summarizes this method (for one of the possible ways of presenting it), and describes a variant which is numerically more robust by mentioning at the same time the problems to solve and the different solutions possible. The description concerns the three-dimensional case.*

## 1 Introduction

Ce papier sur la génération automatique de maillages par une méthode de type Voronoï fait suite au rapport n° 1398 (cf. [7]) traitant du cas de la dimension 2. Il étend les résultats présentés au cas de la dimension 3 et s'attache à trouver une manière numériquement fiable de traduire sur ordinateur les différentes phases utilisées par la méthode de génération.

Le problème que l'on veut résoudre est le suivant (on se place délibérément en dimension 3): *Trouver un maillage acceptable d'un domaine tridimensionnel quelconque à partir de la donnée de son contour décrit par l'ensemble de ses faces triangulaires.*

Le maillage obtenu sera naturellement le support d'un calcul qui pourra indiquer s'il est adapté ou s'il doit l'être par quelque méthode que ce soit (ce n'est pas l'objectif de ce travail). Il est néanmoins important de débiter avec un bon maillage de départ sachant que cette requête de qualité ne peut être que basée sur des considérations géométriques puisque les seules informations supposées connues sont de nature géométrique. La forme et la taille des éléments devront donc être conformes vis-à-vis de ces données.

Avant de présenter la méthode, on va d'abord définir quelques outils qui seront constamment nécessaires dans la suite.

## 2 Quelques outils

On se place dans le cas de maillages (notés  $T$ ) composés de tétraèdres seulement. On note par  $K$  les tétraèdres. Un tétraèdre est défini par la liste orientée de ses sommets, ses faces et ses arêtes sont également orientées, son volume est positif. On associe à chaque tétraèdre  $K$  l'ensemble de ses voisins (par ses faces)  $V_i(K)$ ,  $i = 1, 4$ ; le premier étant le tétraèdre vu par le sommet 1 du tétraèdre donné (avec la convention suivante:  $V_i(K) = 0$  signifie que la face  $i$  (opposée au sommet  $i$ ) de l'élément  $K$  est frontière). De plus, on notera par  $B(K)$  la sphère circonscrite au tétraèdre  $K$ .

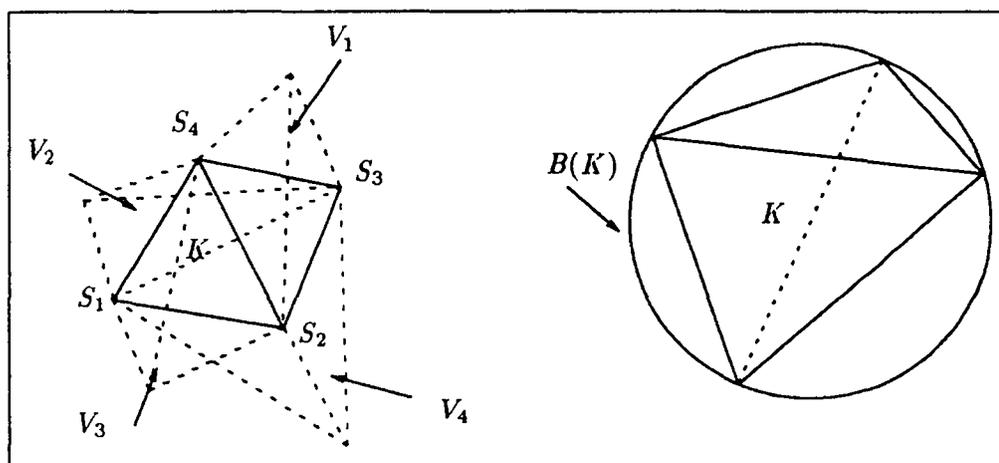


Figure 1: Définitions élémentaires.

## 2.1 Normalisation préalable

Pour éviter les erreurs d'arrondis, les données (i.e. les coordonnées des points fournis) sont normalisées entre 0 et 65535 puis converties en entiers.

Cette valeur représente la résolution ou pouvoir de séparation de la méthode. Elle est choisie de façon à permettre le calcul exact des volumes (i.e., de longueurs au cube) et incidemment celui des surfaces; elle est fonction du nombre de bytes du mot de l'ordinateur employé ( $65535 = 2^{16} - 1$ ).

## 2.2 Position d'un point par rapport à un élément

Une opération usuelle est de répondre à la question suivante: *Trouver la situation d'un point P donné par rapport à un élément K donné ?*

On considère  $K_j^* = \{F_j, P\}$ ,  $1 \leq j \leq 4$ , les 4 tétraèdres virtuels construits en joignant la face  $F_j$  de  $K$  au point  $P$ . Si  $ABCD$ , dans cet ordre, est le tétraèdre  $K$ , alors les quatre tétraèdres associés sont successivement  $ABCP$ ,  $ABPD$ ,  $APCD$  et  $PBCD$ . On définit par  $Det(K)$  le déterminant suivant:

$$Det(K) = \begin{vmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{vmatrix}.$$

où  $x_1, x_2, x_3, x_4$  sont les abscisses des sommets de  $K$ ,  $y_i$  leurs ordonnées et  $z_i$  leurs cotes. Grâce à la normalisation précédente, le calcul de  $Det(K)$  est numériquement exact. Notons que cette valeur est six fois le volume de  $K$  (par la suite, de manière impropre, on fera la confusion entre  $Det(K)$  et le volume de  $K$ ).

Alors:

**Algorithme 1 :**

- a) Si  $\forall j, 1 \leq j \leq 4, Det(K_j^*) > 0 \leftrightarrow P \in K$ , Fin;
- b) Si  $\exists j_1, \dots, j_q, 1 \leq j_1 \neq \dots \neq j_q \leq 4$  tel que  $Det(K_{j_1}^*) = \dots = Det(K_{j_q}^*) = 0 \leftrightarrow \{P \text{ est dans la région intersection des } q \text{ demi-espaces déterminés par les faces } F_{j_1} \dots F_{j_q}\}$ , Fin.
- c) Si  $\exists j_1, \dots, j_q, 1 \leq j_1 \neq \dots \neq j_q \leq 4$  tel que  $Det(K_{j_1}^*) = \dots = Det(K_{j_q}^*) < 0 \leftrightarrow \{P \text{ est dans la région intersection des } q \text{ demi-espaces définis par les faces } F_{j_1} \dots F_{j_q}\}$  ne contenant pas leur sommet opposé, Fin.  $\square$

$P$  est déterminé par rapport aux 15 régions définies par  $K$ .

Démonstration: évidente.

Une application de cet algorithme permet de trouver à quel élément du maillage d'un domaine convexe appartient un point  $P$  interne à ce domaine. Ceci donne l'algorithme suivant:

**Algorithme 2 :**

- a) Initialisation:  $K = K_0$  où  $K_0$  est un tétraèdre quelconque du maillage;
- b) Si  $\forall j, j = 1, 4 \quad Det(K_j^*) > 0$  alors  $P \in K$ , alors  $C = K$  Fin;
- c) Ou si  $\exists j \quad Det(K_j^*) < 0$ , trouver le tétraèdre  $K'$  tel que  $K \cap K' = F_j$  où  $F_j$  est la face  $j$  de  $K$ , faire  $K = K'$  et aller à b);

- d) Ou si  $\exists j_1, \dots, j_q$  tel que  $Det(K_{j_1}^*) = \dots = Det(K_{j_q}^*) = 0$  alors  $P \in (\bigcap_{k=1,q}(F_{j_k}))$  donc  $P \in \mathcal{C}$  l'ensemble des éléments du maillage incluant  $\bigcap F_{j_k}$  (qui est une face si  $q = 1$ , une arête si  $q = 2$  et un sommet si  $q = 3$ ), Fin; □

$\mathcal{C}$  est l'ensemble des éléments contenant  $P$ .

Démonstration: évidente (d) n'est visité que si c) n'est pas vrai).

Ici, en dimension 3, seules 4 situations sont possibles:

1.  $P \in K$ ,
2.  $P$  est sur une face,
3.  $P$  est sur une arête,
4.  $P$  est un sommet (en pratique on ne rencontrera pas ce cas, cf. plus bas).

Donc la solution au problème (en excluant ce dernier cas) est:

- $P \in K$
- ou  $P \in \mathcal{C}$ , les 2 éléments contenant la face  $\bigcap F_{j_k}$ .
- ou  $P \in \mathcal{C}$ , les éléments contenant l'arête  $\bigcap F_{j_k}$ .

L'ensemble des éléments contenant une arête est l'ensemble des éléments s'enroulant autour de cette arête, on appellera un tel ensemble **coquille locale** (cf. plus bas).

### 2.3 Eléments coupés par un segment

On se place dans le cas d'un domaine convexe dont on connaît un maillage et on cherche à répondre à la question suivante: *Trouver tous les éléments du maillage coupés par un segment donné.* En pratique et dans la suite, le but est de trouver tous les éléments coupés par un segment dont les extrémités  $A$  et  $B$  sont deux sommets du maillage.

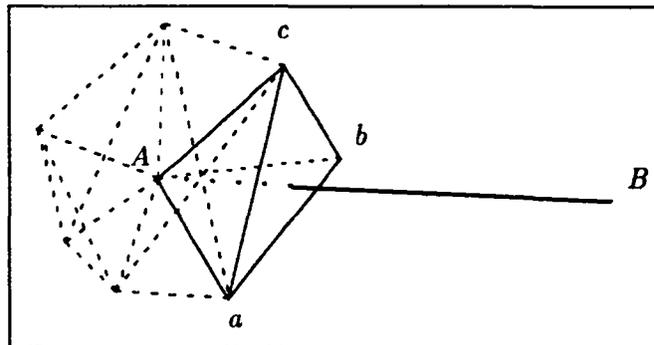


Figure 2: Recherche des tétraèdres coupés par un segment.

L'algorithme suivant donne la solution:

**Algorithme 3 :**

- a) Trouver un élément  $K$  ayant  $A$  comme sommet (si  $B$  est aussi sommet de  $K$ , l'ensemble cherché se réduit à cet élément  $K$ , Fin);
- b) Parmi les éléments ayant  $A$  comme sommet, trouver celui dont la face  $abc$  (cf. dessin) opposée à  $A$  est coupée par  $AB$ 
  - $b_1$ ) un tel élément existe: empiler cet élément;
  - $b_2$ ) un tel élément n'existe pas: alors  $AB$  coupe une arête de la face  $abc$ , soit par exemple l'arête  $ab$ : empiler tous les éléments possédant cette arête;
- c) Tant que l'un de ces éléments n'a pas  $B$  comme sommet, aller en d), sinon Fin.
- d) Parmi les faces ou arêtes des éléments ainsi empilés, trouver celle qui est coupée par  $AB$ ,
  - $d_1$ ) si  $AB$  coupe une face, empiler l'élément ayant cette face commune avec l'élément de départ, aller en c),
  - $d_2$ ) si  $AB$  coupe une arête, empiler tous les éléments possédant cette arête et aller en c),
- e) Aller en c). □

La pile ainsi constituée est l'ensemble cherché.

Démonstration: évidente encore une fois. Les seuls problèmes à résoudre sont les suivants:

- Trouver si un segment  $AB$  coupe une face  $abc$ ,
- Trouver si un segment  $AB$  coupe une arête  $ab$ ,

La solution revient à utiliser l'algorithme 2 ci-dessus et se résume à calculer les  $Det$  incriminés.

### Quelques définitions :

Soient  $AB$  un segment joignant deux sommets d'éléments et  $ab$  une arête du maillage  $T$ , on introduit les définitions suivantes:

**Définition 1:** L'ensemble  $T_{AB}$  est un **tuyau vis-à-vis** de  $AB$  s'il est constitué des tétraèdres suivants:

- un tétraèdre ayant  $A$  comme sommet tel que sa face opposée à  $A$  soit coupée par  $AB$ ,
- un ensemble (éventuellement vide) de tétraèdres dont 2 faces sont coupées par  $AB$ ,
- un tétraèdre ayant  $B$  comme sommet tel que sa face opposée à  $B$  soit coupée par  $AB$ . □

Notons que les éléments de  $T_{AB}$  sont 2 à 2 voisins par la face coupée par  $AB$ .

**Définition 2:** L'ensemble  $T_{AB}$  est une **coquille locale vis-à-vis** de  $AB$  pour l'arête  $ab$  s'il est constitué des tétraèdres s'enroulant autour de l'arête  $ab$  telle que  $AB$  coupe  $ab$ . □

**Définition 3:** L'ensemble  $T_{AB}$  est une **coquille vis-à-vis** de  $AB$  s'il est constitué des tétraèdres suivants:

- un tétraèdre ayant  $A$  comme sommet tel que sa face opposée à  $A$  ou une arête ne contenant par  $A$  soit coupée par  $AB$ ,
- un ensemble (éventuellement vide) de tétraèdres dont 2 faces sont coupées par  $AB$ ,
- au moins une coquille locale (éventuellement au départ ou à l'arrivée seulement)
- un tétraèdre ayant  $B$  comme sommet tel que sa face opposée à  $B$  ou une arête ne contenant pas  $B$  soit coupée par  $AB$ .  $\square$

Ces trois types d'ensemble nous serviront par la suite dans la section consacrée au respect de la frontière donnée.

## 2.4 Retournement d'une face

Soient 2 tétraèdres, adjacents par une face, dont la réunion est un polyèdre convexe, on peut alors remailler ce polyèdre en créant l'arête formée par les 2 sommets opposés à la face initialement commune en supprimant cette dernière.

Cette configuration est en fait un tuyau (cf. définition 1) à deux élément seulement. La transformation des 2 éléments initiaux  $Aabc$  et  $Bacb$  consiste à créer les 3 éléments suivants:  $aABb$ ,  $bABc$  et  $cABa$ . Cette opération, notée  $MOD_{2 \rightarrow 3}$ , utilisée dans la suite, n'est que la plus simple des transformations locales que nous allons introduire; elle ne s'applique que si le tuyau est convexe.

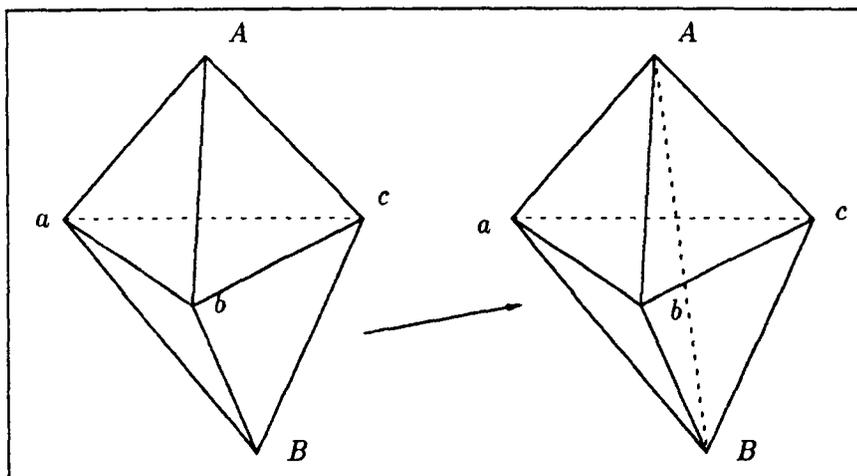


Figure 3: Transformation  $MOD_{2 \rightarrow 3}$ .

## 3 Insertion de points dans un maillage existant

Dans le cas d'un domaine convexe, on se donne un maillage  $T$  et un point  $P$  interne (dans le cas de plusieurs points, on appliquera la méthode à chacun). Le problème posé est le suivant: Construire un nouveau maillage du domaine de telle sorte que  $P$  soit un sommet d'éléments.

### 3.1 La méthode de Voronoï

On suppose ici que  $T$  est un maillage de Delaunay, i.e., la sphère circonscrite  $B(K)$  à chacun de ses éléments  $K$  ne contient pas d'autres points (sommets de tétraèdres du maillage) que les 4 sommets<sup>1</sup> de  $K$ . La méthode de Voronoï consiste à construire un nouveau maillage possédant  $P$  comme sommet de tétraèdres, elle peut se schématiser de la façon suivante:

**Algorithme 4 :**

- a) Trouver l'élément, les 2 éléments ou la coquille locale de  $T$  contenant  $P$ ;
- b) Mettre ce(ces) élément(s) dans une pile  $S$ ;
- c) Compléter récursivement (en prenant en compte les voisins des tétraèdres déjà empilés)  $S$  en y ajoutant tout élément  $K$  de  $T$  tel que  $P \in B(K)$ ;
- d) Trouver  $F_1 \dots F_p$  les faces externes de  $S$  c'est-à-dire les faces non communes à 2 éléments de  $S$ ;
- e) Construire le nouveau maillage comme  $T = (T - S) \cup \{F_j, P\}_j \quad 1 \leq j \leq p$ , Fin.  $\square$

$T$  est une triangulation de Delaunay incluant  $P$  comme sommet de tétraèdres.

Démonstration: voir [13] ou [14] (entre autres).

Remarque:

L'étape a) utilise l'algorithme 2 décrit ci-dessus.

**Remarque :** Cette approche est incrémentale au sens où on traite un seul point. D'autres approches existent pour résoudre le problème. Par exemple, on peut penser aux méthodes de type divide-and-conquer qui sont applicables quand tous les points sont connus. Ce n'est pas, en général, le cas dans notre situation où seuls les points de la frontière sont donnés et où il faudra créer des points internes (cf. ci-dessous).

### 3.2 Insertion locale et optimisation

Sans hypothèse sur la nature de  $T$ , le même problème peut être résolu de la manière suivante:

**Algorithme 5 :**

- a) Trouver l'élément, les 2 éléments ou la coquille locale (de  $n$  éléments) de  $T$  contenant  $P$ ;
- b) Mettre ce(ces) élément(s) dans une pile  $S$ ;
- c) Trouver  $F_1 \dots F_p$  les faces externes de  $S$  et joindre  $P$  à ces faces; on crée de la sorte 4, 6 ou  $2n$  éléments;
- d) Vider  $S$  et mettre ces éléments dans  $S$ ;
- e) Chercher récursivement parmi les voisins des éléments de  $S$  ceux dont la sphère circonscrite contient  $P$  et retourner la face externe de  $S$  commune à ces éléments, mettre à jour  $S$  et tant que  $S$  est non vide aller en e), sinon Fin.  $\square$

<sup>1</sup>Il s'agit ici de l'une des caractérisations des maillages de Delaunay.

Cet algorithme est constructif.

Démonstration: évidente car la méthode est en elle-même constructive.

Remarques:

L'étape a) utilise l'algorithme 2 décrit ci-dessus.

L'étape e) utilise en particulier la procédure de retournement de face  $MOD_{2 \rightarrow 3}$  vue ci-dessus et s'interprète comme une optimisation locale.

Le nouveau maillage est localement de Delaunay.

### 3.3 Insertion directe

La méthode de Voronoï est numériquement peu fiable car savoir situer un point par rapport à une sphère (comparer  $P$  et  $B(K)$ ) ne conduit pas à une réponse claire dès lors que  $P$  est très voisin de  $B(K)$ . Une erreur minime (par exemple de 0.01 pour cent) sur le calcul correspondant entraîne généralement une réponse fautive (i.e. une erreur de 100 pour cent) à la question et la méthode produit un maillage généralement faux.

Pour surmonter ce handicap, on propose la méthode suivante:

**Algorithme 6 :**

- a) Trouver l'élément, les 2 éléments ou la coquille de  $T$  contenant  $P$ ;
- b) Mettre ce(ces) élément(s) dans une pile  $S$ ;
- c) Compléter récursivement  $S$  en y ajoutant tout élément  $K$  de  $T$  tel que  $P \in B(K)$  aux erreurs de calcul près;
- d) Trouver  $F_1 \dots F_p$  les faces externes de  $S$  c'est-à-dire les faces non communes à 2 éléments de  $S$ ;  
 Définir les  $p$  tétraèdres virtuels  $K_j^* = (F_j, P)$ ,  $1 \leq j \leq p$  ;  
 Définir  $V_F$  l'ensemble des sommets des faces  $F_j$ ;  
 Définir  $V_S$  l'ensemble des sommets des éléments de  $S$ ;
- e) Initialiser  $S' = S$ ;
- f) Pour  $j$  de 1 à  $p$ :
  - Si  $Det(K_j^*) > 0$ , ne rien faire;
  - Si  $Det(K_j^*) < 0$ ,  $S' = S' - K_j$  où  $K_j$  est le tétraèdre de  $S'$  de face  $F_j$ ;
  - Si  $Det(K_j^*) = 0$ ,  $S' = S' \cup K_j$  où  $K_j$  est le tétraèdre de  $T$  non contenu dans  $S'$  ayant la face  $F_j$ .
- g) Si  $S'$  n'a pas été modifié par l'étape f), aller en h); sinon mettre à jour l'ensemble des  $F_j$ ,  $V_F$  et  $V_{S'}$  puis aller en f);
- h) Dénombrer  $V_F$  et  $V_{S'}$ ;  
 Si ces 2 ensembles ont le même cardinal faire  $S = S'$  et aller en i);  
 Sinon faire  $S' = S' - K$  et aller en f) ( $K$  est un tétraèdre possible<sup>2</sup> de  $T$  possédant un sommet, différent de  $P$ , contenu dans  $V_{S'}$  sans être dans  $V_F$ );

<sup>2</sup>Oltre le fait de contenir un point interne "en trop", un élément est possible si, d'une part l'ensemble reste connexe après son élimination et si, d'autre part il ne fait pas partie de l'ensemble des éléments obtenus à l'étape a) de l'algorithme.

- i) Construire le nouveau maillage comme  $T = (T - S) \cup \{F_j, P\}_j \quad 1 \leq j \leq p$ , Fin.  $\square$

Cet algorithme est constructif.

Démonstration: évidente car la méthode est elle-même constructive.

Remarques:

On doit vérifier que  $S$  est connexe ce qui revient à dire que tout élément de  $S$  (dés que  $S$  a plus d'un élément) possède au moins un voisin lui-même dans  $S$ . Si tel n'est pas le cas, l'élément est supprimé de la pile.

L'étape a) utilise l'algorithme 2 décrit ci-dessus.

L'étape f) ne repose que sur le calcul des  $Det$  correspondants.

L'étape h) revient à effectuer un dénombrement.

Le nouveau maillage est correct (seuls des calculs entiers (donc exacts) ont été utilisés) et est presque de Delaunay.

Un problème fréquemment rencontré en dimension 3 est celui lié à la création d'éléments très plats bien que corrects du point de vue des propriétés requises (*sliver* en anglais). Ces éléments ne pouvant qu'être source de difficultés numériques par la suite doivent donc être évités autant que faire ce peut. On note, en particulier, que ce cas se présente quand de nombreux points sont coplanaires, situation fréquente dans la réalité<sup>3</sup>

### 3.4 Insertion contrôlée par un critère

Une écriture plus abstraite des schémas présentés est la suivante:

On se donne:

1.  $T$  un maillage d'un domaine contenant le point  $P$ ,
2.  $C_r$  un critère (ou un ensemble de critères),
3.  $C_o$  une contrainte (ou un ensemble de contraintes).

On isole dans  $T$  l'ensemble  $S$  des éléments tels que:

(H1)  $S$  vérifie le critère  $C_r$  pour le point  $P$

(H2)  $S$  satisfait la contrainte  $C_o$

Sous la **seule condition** que l'on sache remailler  $S$  en prenant en compte le point  $P$  (soit  $S'$  l'ensemble des éléments de ce remaillage local) on obtient alors le nouveau maillage  $T$  par:

$$T = T - S \cup S'$$

Un choix adéquat de  $C_r$  permet d'assurer la validité du processus d'insertion du point  $P$ ; le choix de  $C_o$  permet d'ajouter des propriétés supplémentaires au maillage ainsi construit.

Dans le cas standard, on a le choix suivant:

$$C_r = \{S \text{ vérifie } C_r \text{ pour le point } P$$

$$\Leftrightarrow P \text{ est dans la sphère circonscrite de tout élément de } S\}$$

<sup>3</sup>En ce sens, traiter un nuage de points aléatoirement distribués apparaît comme une source de simplification plutôt que comme un cas pratiquement réaliste.

et on n'impose pas de contrainte.

Le cas 2 revient au même. Le cas 3 est en fait une interprétation constructive du critère qui s'écrit simplement comme:

$$C_r = \{S \text{ vérifie } C_r \text{ pour } P \Leftrightarrow \text{ses faces externes sont visibles par } P\}$$

ou, dit autrement:

$$C_r = \{S \text{ vérifie } C_r \text{ pour le point } P \Leftrightarrow P \text{ étoile l'ensemble } S\}$$

toujours sans imposer de contrainte.

Il est donc clair qu'un choix judicieux du couple  $\{C_r, C_o\}$  va permettre d'obtenir une triangulation possédant déjà des propriétés intéressantes, en particulier de façon à respecter une frontière donnée ou pour contrôler la qualité des éléments générés, par exemple pour éviter la création d'éléments très plats (cf. ci-dessus).

## 4 Respect d'une frontière donnée

La section 3 a montré comment insérer un point dans un maillage (pour un domaine convexe), le problème posé initialement est *sensiblement* différent. Il s'agit de mailler un domaine, non nécessairement convexe, à partir de la donnée de sa frontière, celle-ci étant connue comme l'ensemble des faces triangulaires la constituant.

Une première idée consiste à:

- Construire l'ensemble des points  $P$  ci-dessus comme l'ensemble des sommets des faces de la frontière donnée;
- Construire 8 points supplémentaires tels que tout point donné soit interne au parallépipède formé par ces 8 points;
- Mailler cette boîte avec 5 tétraèdres (de Delaunay);
- Appliquer l'un quelconque des processus d'insertion de points pour construire, pas à pas, un maillage de la boîte composé de tétraèdres dont les sommets sont, d'une part les 8 points ajoutés et d'autre part tous les points de la frontière.

Ce maillage *ne contient pas*, en général, dans la liste des faces de ses éléments les faces données (il ne contient même pas, toujours en général, toutes les arêtes des faces données). Par suite, on ne peut pas définir correctement le domaine donné (les notions d'intérieur et d'extérieur ne peuvent, en effet, être précisées que par rapport à la frontière fournie donc par rapport aux faces de celle-ci).

La définition du domaine nécessite donc la présence de la frontière donnée dans le maillage issu de l'insertion, un par un, des points de cette frontière. Il existe plusieurs façons de poser ce problème et, par suite, plusieurs réponses possibles.

#### 4.1 Différentes définitions du problème

La première idée consiste à trouver la régularité que doivent satisfaire les données de telle sorte que le problème ne se pose pas c'est-à-dire que le maillage contenant les extrémités des faces données contienne ces faces. On trouvera dans [11] l'introduction de la notion de faces Delaunay admissibles qui implique le résultat. Pratiquement cette notion est délicate, voire impossible, à traduire, par suite, elle est d'un intérêt numérique faible.

Une deuxième idée consiste à recenser les faces et arêtes manquantes et à les découper par création de points internes. Le nouvel ensemble de points ainsi formé sert alors de donnée au processus d'insertion de points avec le secret espoir que le maillage obtenu contienne toutes les faces données (faces initiales ou faces initiales découpées). Intuitivement, une application récursive du procédé donne le résultat car elle revient à former numériquement un ensemble de données Delaunay admissible. On trouvera dans [15] la mise en application de cette méthode et dans [21] une extension astucieuse (au cas de la dimension 2).

Une autre idée consiste à transformer localement le maillage initial pour régénérer les faces manquantes (et supprimer les faces et arêtes gênantes). C'est cette idée que nous allons développer ci-dessous.

#### 4.2 Retour au cas bidimensionnel

On renvoie à [7] en rappelant que dans ce cas le problème revient à recréer dans  $T$ , le maillage construit en insérant les points extrémités des arêtes données, les arêtes frontières manquantes. Dans ce cas l'ensemble  $T_{AB}$ , ensemble des triangles dont une arête est coupée par l'arête manquante  $AB$ , est nécessairement un tuyau (au sens de la définition 1 adaptée au cas 2D) et on a montré que l'on pouvait toujours modifier  $T_{AB}$  de telle sorte que  $AB$  existe, sans autres opérations que des retournement d'arêtes. Par suite, appliquant le procédé à chaque arête manquante, on modifie  $T$  en régénérant toutes les arêtes frontières.

Il est important de remarquer que la modification locale appliquée ne nécessite pas de créer de points supplémentaires dans ce cas (la dimension 2).

#### 4.3 Introduction au cas tridimensionnel

Le problème de la recréation des arêtes et des faces manquantes dans le cas tridimensionnel n'est pas aussi simple. En particulier, il existe des exemples simples pour lesquels il est impossible de forcer les arêtes et faces imposées sans création de points internes (cf. figure 4 où le phénomène est illustré dans le cas d'un simple prisme).

Par suite, il est a priori évident que l'emploi de transformations locales comme en dimension 2 ne donnera pas, dans le cas général, la solution au problème posé.

#### 4.4 Modifications locales

Néanmoins, on propose comme méthode de régénération des items manquants d'utiliser, tant que cela peut se faire (i.e. tant que des items sont recréés), des transformations locales. Parmi celles-ci on considère la transformation déjà vue  $MOD_{2 \rightarrow 3}$  qui permet de casser une face et de construire une arête et son application en sens inverse, notée  $MOD_{3 \rightarrow 2}$  qui permet donc de casser une arête et de construire une face. On définit de plus des transformations locales intéressantes des ensembles plus importants de tétraèdres:

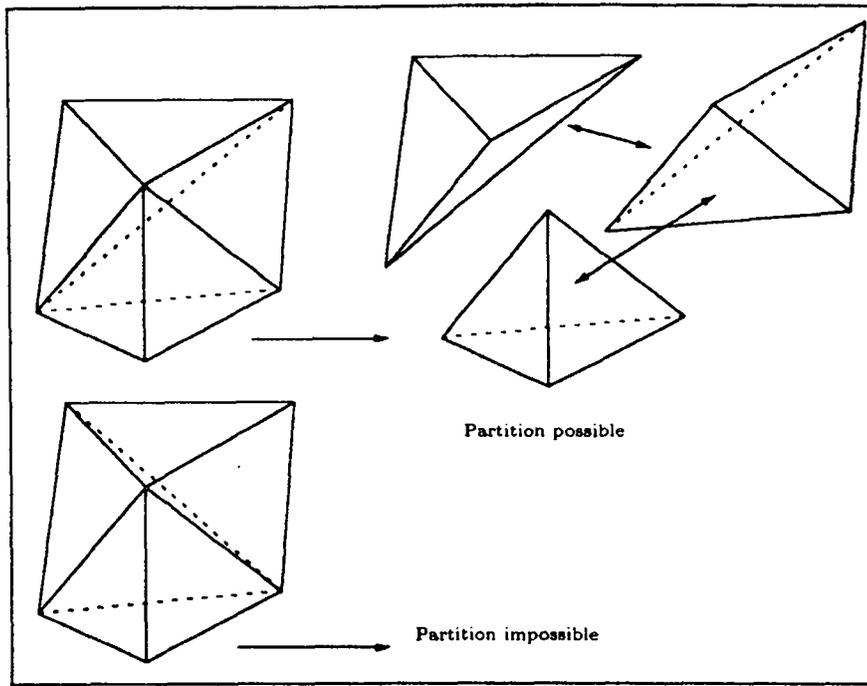


Figure 4: *Maillage réalisable et maillage impossible.*

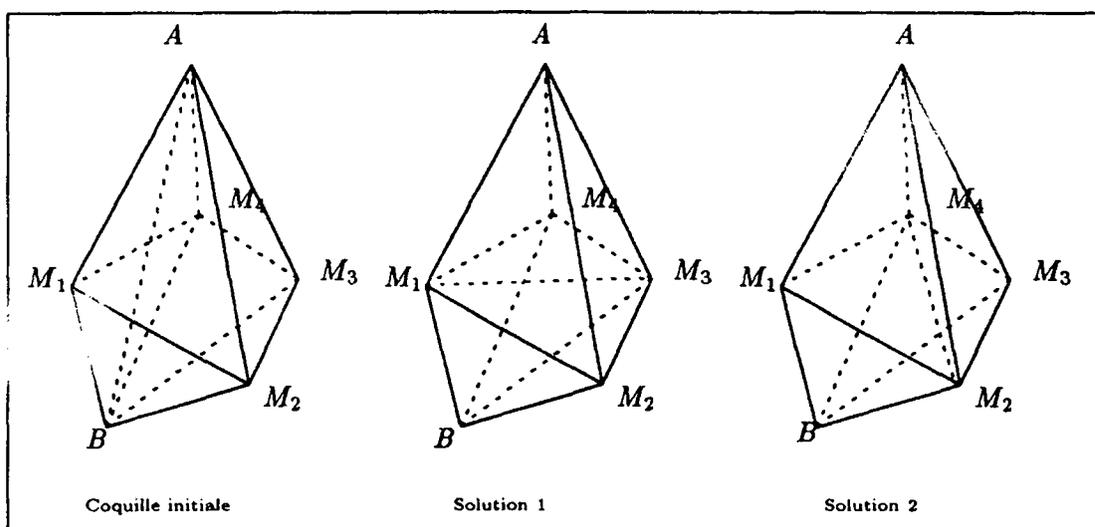


Figure 5: *Les maillages possibles d'une coquille de 4 éléments.*

- Coquille locale (cf. définition) de 4 éléments: un tel ensemble, s'il est convexe, peut être remaillé de 2 manières différentes (cf. figure 5).
- Coquille locale de 5 éléments: dans ce cas, il existe 5 possibilités de remaillage.
- Coquille locale de plus de 5 éléments: l'application de transformations locales de type  $MOD_{2 \rightarrow 3}$  permet, dans certains cas, de diminuer le nombre d'éléments d'une telle coquille. On aurait pu, comme dans l'exemple précédent, énumérer toutes les possibilités de remaillage, ceci se révèle fastidieux et gourmand en temps calculs: il y a en effet de nombreuses simulations à faire. En effet,  $N_n$  le nombre de combinaisons possibles pour remailler une coquille de  $n$  éléments est défini par la formule suivante:

$$N_n = \sum_{i=3}^n N_{i-1} N_{n+2-i}$$

formule initialisée par  $N_2 = 1$ . On obtient donc successivement (et, au moins pour les premières valeurs, on vérifie aisément à la main):

- $N_3 = 1$  : il n'y a qu' 1 manière de remailler une coquille convexe de 3 éléments, la solution comprend  $n_3 = 2$  tétraèdres
- $N_4 = 2$  : il y a 2 manières de remailler une coquille convexe de 4 éléments, chaque solution comprend  $n_4 = 4$  tétraèdres
- $N_5 = 5$  : il y a 5 manières de remailler une coquille convexe de 5 éléments, chaque solution comprend  $n_5 = 6$  tétraèdres
- $N_6 = 14$  : il y a 14 manières de remailler une coquille convexe de 6 éléments, chaque solution comprend  $n_6 = 8$  tétraèdres
- $N_7 = 43$  : il y a 43 manières de remailler une coquille convexe de 7 éléments ! Le remaillage comprend  $n_7$  tétraèdres (la formule générale donnant le nombre de tétraèdres étant simplement  $n_i = n_{i-1} + 2$ ).
- ...

On vérifiera le bilan, en terme de création et suppression d'arêtes et de faces, de ces transformations.

#### 4.5 Applications aux arêtes et faces manquantes

Il suffit maintenant d'appliquer, à chaque item manquant, la ou les transformations voulues pour modifier localement le maillage. En pratique, on traite tout les éléments proches de l'item cherché en considérant dans l'ensemble des tétraèdres contenant un sommet de l'arête (de la face) manquante les coquilles partant de ce sommet.

#### 4.6 Création de points internes

On trouvera dans [9] la description complète d'une méthode de régénération des arêtes et des faces manquantes en créant le cas échéant des points internes quand les différentes modifications locales ne sont plus possibles.

Il suffit alors d'appliquer cette méthode aux quelques arêtes et faces qui manqueraient après l'utilisation de la méthode indiquée ci-dessus.

L'intérêt de procéder de la sorte est que, au vue des essais effectués, il n'y a que peu d'exemples pour lesquels il manque encore un item de la frontière et dans le cas inverse, il ne manque que très peu de tels items, par suite, peu de points internes seront introduits et le traitement de la suite du processus ne sera pas gêné outre mesure par ces points.

#### 4.7 Applications aux arêtes et faces manquantes

Le même papier ([9]) décrit l'application de la méthode précédente pour générer, un à un, les items manquants (arêtes puis faces de la donnée).

#### 4.8 Définition de l'intérieur du domaine

Toutes les arêtes puis toutes les faces de la frontière du domaine existent maintenant dans  $T$ . Ce maillage permet donc de définir l'intérieur du domaine en construisant le graphe des composantes connexes du domaine.

**Algorithme 8 :**

- a) Prendre un élément quelconque de  $T$ , le marquer par un entier  $n$ ;
- b) Examiner ses voisins:
  - Tant que la face commune n'est pas une face frontière et que le voisin n'est pas déjà marqué, le marquer avec le nombre  $n$  et empiler le tétraèdre si ses 4 faces ont été examinées;
- c) Tant que la pile d'indice  $n$  peut évoluer, aller en b)
- d) Sinon, tant qu'il reste des tétraèdres non marqués, prendre un tétraèdre non marqué, faire  $n = n + 1$ , le marquer avec  $n$  et aller en b) □

Ensuite, il suffira de retenir tous les éléments marqués par  $n$  pour avoir tous les éléments appartenant à cette composante du domaine (ceci permettra d'éliminer l'extérieur du domaine et sera effectué plus tard).

## 5 Génération des points internes

A ce stade, on dispose d'un maillage  $T$  dont les éléments sont marqués. Ce maillage ne contient comme sommets que les points de la frontière (et éventuellement quelques points internes) donc est généralement impropre aux calculs. Cette phase va donc consister à créer des points internes dans les éléments marqués et à les insérer dans  $T$ .

Pour procéder à cette phase, il faut répondre aux deux questions suivantes:

- Faut-il créer un (des) point(s) dans un tétraèdre de  $T$  ?
- Où mettre ce (ces) points ?

La réponse à ces questions dépend étroitement de ce que l'on veut faire du maillage final en terme de calculs. Il est clair que la nature de celui-ci (mécanique du solide, simulation thermique, mécanique des fluides, ...) et que la nature des données physiques (caractéristiques des matériaux, vitesse, pression, nature des efforts, températures imposées, etc.) influent sur la nature attendue du maillage.

### 5.1 Qualité attendue

Que peut t-on faire ? Quel est le but visé et quels sont les paramètres sur lesquels on peut influencer ?

Le but visé est d'obtenir un maillage adapté aux calculs. Les paramètres sur lesquels on peut agir sont:

- La taille des tétraèdres;
- La forme des tétraèdres

La seule donnée étant la description discrète de la frontière du domaine (et éventuellement la donnée de quelques points internes), on ne peut utiliser, pour créer  $T$ , que des informations relatives à cette donnée. Ainsi le but visé ne peut être *raisonnablement* que d'obtenir un maillage isotrope reflétant les propriétés des données.  $T$  sera donc adapté aux données et non, a priori au problème physique à résoudre. On ne peut pas faire mieux, sans choix discutables, en l'absence d'informations supplémentaires. Par suite, la seule ambition de la méthode proposée est de construire un maillage  $T$  raisonnable qui permettra d'effectuer un premier calcul du problème posé; le résultat de celui-ci et l'analyse des champs de solutions permettra alors d'adapter  $T$  au problème (voir par exemple [5] sur les méthodes d'adaptation et [18] pour le cas anisotrope ou directionnel en dimension 2).

### 5.2 Méthode de création d'un nuage de points internes

A chaque point donné, on associe un  $h$  local qui est la distance souhaitable entre ce point et ses voisins. Si le point est un point interne spécifié,  $h$  est fourni par l'utilisateur; si le point est un point de la frontière,  $h$  est une moyenne:

- la moyenne entre les longueurs des arêtes ayant ce point comme extrémité.
- la moyenne entre les surfaces des 2 faces ayant le point comme sommet (à une correction près, genre racine carrée et coefficient de normalisation).

On propose le processus suivant:

**Algorithme 9 :**

- a) Faire, pour tout les éléments  $K$  de  $T$ :
  - b) Examiner les 6 arêtes de  $K$  en comparant leurs longueurs  $l_i$  avec les  $h$  de leurs extrémités (soient  $h_{i1}$  et  $h_{i2}$  les  $h$  des extrémités de l'arête  $i$ );
    - \* c) calculer  $n_i$  le nombre de points à créer sur l'arête de telle sorte que ces points suivent une répartition donnée (voir ci-dessous);
  - d) Prendre tous les points ainsi créés et empiler ceux qui:
    - \*  $d_1$ ) ne sont pas sur une frontière donnée;
    - \*  $d_2$ ) ne sont pas trop proches (au sens de  $h$ ) d'un point existant;
- e) Tant que la pile est non vide, insérer les points de la pile puis aller en a), sinon Fin  $\square$

Le point c) de cet algorithme peut être mené de deux façons.

### 5.2.1 Découpe arithmétique d'une arête

Soit  $F_k$  l'arête examinée et  $d_k$  sa longueur, on note par  $P_k(0)$  et  $P_k(n)$  ses 2 extrémités et par  $h(0)$  and  $h(n)$  les  $h$  correspondants, on définit la suite:

$$\alpha_i = d(P_k(i), P_k(i+1)) \text{ pour } i = 0, n$$

avec

$$\alpha_0 = h(0) + r \text{ et } \alpha_n = h(n) - r$$

Le problème est de calculer  $n$  et  $r$  donc les  $\alpha_i$  tels que la suite des points ainsi définie soit arithmétique, c'est-à-dire que:

$$\alpha_{i+1} = \alpha_i + r$$

où  $r$  est la raison cherchée. Le système à résoudre est:

$$\begin{cases} \sum_{i=0}^n \alpha_i = d_k \\ (n+2)r = \Delta H = \|h(0) - h(n)\| \end{cases} \quad (1)$$

Ce système traduit que  $d_k$  est la somme des longueurs des sous-segments que l'on cherche à créer pour découper l'arête  $F_k$  et que la variation de  $h(0)$  à  $h(n)$  est équirépartie entre tous ces segments.

On en déduit:

$$\begin{cases} (n+1) \cdot (h(0) + \frac{(n+2)}{2} \cdot r) = d_k \\ (n+2)r = \Delta H \end{cases} \quad (2)$$

Pour trouver  $n$ , on exprime  $r$  en fonction de  $n$  et on obtient:

$$n = \frac{2d_k}{h(0) + h(n)} - 1$$

Toutefois,  $n$  doit être entier ce que ne donne pas ce calcul, par suite, on choisit pour  $n$  la valeur entière la plus proche de la valeur calculée.  $n$  étant connu, on calcule  $r$ :

$$r = \frac{2d_k(h(n) - h(0))}{(n+1)(n+2)(h(n) + h(0))}$$

Cette valeur est utilisée pour remettre à l'échelle  $h(0)$  et  $h(n)$  avec un facteur  $\alpha$  ( $\beta$ ) calculé comme suit:

$$\alpha = \frac{2 \cdot d_k - (n+1)(n+2)r}{2 \cdot (n+1)h(0)}$$

$$\beta = \frac{2 \cdot d_k + (n+1)(n+2)r}{2 \cdot (n+1)h(n)}$$

Ces nouvelles valeurs,  $n$  connu, permettent de recalculer  $r$  et de définir correctement la suite  $\alpha_i$  donc les points intermédiaires.

### 5.2.2 Découpe géométrique d'une arête

La même analyse est appliquée dans le cas où une progression géométrique est souhaitée. Dans ce cas, on a :

$$\alpha_{i+1} = r \alpha_i$$

avec

$$\alpha_0 = r.h(0) \text{ et } \alpha_n = r^{-1}.h(n)$$

le système à résoudre est, dans ce cas :

$$\begin{cases} \sum_{i=0}^n \alpha_i = d_k \\ h(n) = r^{n+2}.h(0) \end{cases} \quad (3)$$

qui conduit à :

$$\begin{cases} h(0).r \frac{1-r^{n+1}}{1-r} = d_k \\ r^{n+2} = \frac{h(n)}{h(0)} \end{cases} \quad (4)$$

$n$  est calculé comme ci-dessus puis  $r$  est obtenu. On obtient successivement :

$$n = \frac{\ln(h(n)) - \ln(h(0))}{\ln(d_k + h(n)) - \ln(d_k + h(0))} - 2.$$

$$r = \frac{d_k + h(n)}{d_k + h(0)}$$

Les différents  $\alpha_i$  sont alors connus et les points intermédiaires sont parfaitement déterminés.

Remarque: le cas  $r = 1$ . est à traiter séparément.

### 5.2.3 Rejet des points

Appliquant l'opération à tous les tétraèdres marqués de  $T$ , on dispose, à ce stade, d'une série de points placés sur les arêtes de  $T$ .

Le résultat est un ensemble de points bien situés localement (sur chaque arête) mais, à cause de la localité du procédé, il est en général possible d'avoir généré trop de points et en particulier des points très voisins. Pour éviter ce problème, on examine maintenant tous les points afin de supprimer ceux qui ne sont pas nécessaires.

Pour ce faire, on introduit une grille régulière (on peut également utiliser une structure de type octree) et on code dans ses cases les points présents :

- chaque case est analysée: si elle contient plusieurs points, ceux-ci sont comparés et supprimés le cas échéant (en fonction de leur  $h$ ); sinon on regarde les cases voisines tant que les points présents sont trop proches au regard des  $h$  des points examinés;
- les points des frontières sont enlevés de la liste (cf. respect de la frontière).

### 5.2.4 Insertion des points

Les points retenus sont insérés (phase e) de l'algorithme 9) dans  $T$  en utilisant les algorithmes décrits au début du papier:

- Faire, pour tous les points de la pile:
  - Insérer le point dans  $T$
- Fin

## 6 Schéma du mailleur

Il reste à regrouper les résultats des sections précédentes pour obtenir un algorithme de maillage capable de traiter tout domaine, convexe ou non, de frontière non croisée.

Le schéma proposé est le suivant:

- a) Création du nuage de points associé aux données, c'est-à-dire les points des faces du contour du domaine et les points spécifiés;
- b) Détermination de la position de huit points supplémentaires tels que le parallélépipède formé par ceux-ci englobe tous les points du nuage;
- c) Maillage de cette boîte à l'aide de cinq tétraèdres;
- d) Insertion, un à un, des points du nuage pour construire un maillage comprenant comme sommets d'éléments ces points;
- e) Recherche dans  $T$ , le maillage précédent, des arêtes du contour donné:
  - e1)  $T$  contient les arêtes données, aller en f);
  - e2) Sinon, modifier localement  $T$  en recréant, une à une, les arêtes manquantes;
- f) Recherche dans  $T$ , le maillage précédent, des faces du contour donné:
  - f1)  $T$  contient les faces données, aller en g);
  - f2) Sinon, modifier localement  $T$  en recréant, une à une, les faces manquantes;
- g) Marquage des tétraèdres de  $T$  en fonction de leur position vis-à-vis du contour;
- h) Création des points internes et insertion de ceux-ci;
- i) Elimination des tétraèdres extérieurs;
- j) Régularisation éventuelle.

Remarques: Les étapes a) b) et c) (tout comme le fait de n'éliminer l'extérieur qu'en i)) permettent d'être constamment dans un cas convexe.

L'insertion des points (étapes d) et h)) est faite via l'algorithme 6, ceci donne le nom de la méthode (et le titre du papier).

## 7 Exemples d'application

Sont donnés maintenant quelques exemples de maillages obtenus en employant cette méthode. Pour chaque exemple, on donnera le nombre de tétraèdres, le nombre de points, le temps CPU (les tests numériques ont été menés sur un APOLLO-HP 425, un APOLLO DN10000 et un IBM 6000), la qualité du résultat c'est-à-dire la répartition des éléments en fonction de leur forme mesurée via la quantité  $Q = \alpha \frac{h}{\rho}$  où  $\alpha$  est un coefficient de normalisation tel que  $Q = 1$  pour un tétraèdre équilatéral,  $h$  est le diamètre de l'élément mesuré,  $\rho$  est le rayon de sa sphère inscrite: notons que  $Q$  augmente quand l'élément s'aplatit. On note également, pour apprécier le résultat obtenu, la qualité théoriquement attendue qui est mesurée à partir de la plus mauvaise face de la donnée.

Par ailleurs, un paramètre  $\omega$  a été introduit pour diluer l'influence des  $h$  frontières au fur et à mesure que l'on s'éloigne de celle-ci ( $\omega \geq 1.$ , une valeur 1. donne un résultat neutre, une valeur 1.2 indique une grande dilution et diminue le temps de calcul nécessaire).

La méthode s'applique naturellement pour construire le maillage de domaines de forme quelconque de frontières à plusieurs composantes connexes. Elle s'applique également pour créer le maillage autour de domaine, par exemple en particulier dans le cas de la mécanique des fluides (maillage autour d'un avion, d'une navette, etc) et dans le cas de problème de simulation de refroidissement (maillage de moule enrobant un objet).

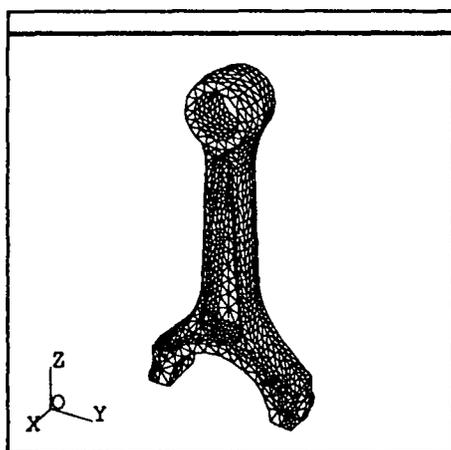


Figure 6: Exemple 1.

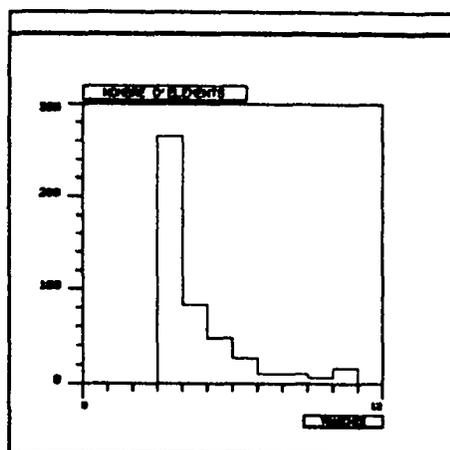


Figure 7: Histogramme.

Les 4 exemples présentés (figures 6, 8, 10 et 12) sont extraits du jeu de tests dont nous disposons et ont déjà été utilisés (cf. en particulier [8]) avec les versions antérieures du mailleur. A chaque exemple est associé son histogramme de qualité (figures 7, 9, 11 et 13) c'est-à-dire le tracé de la répartition de ses éléments en fonction de leur qualité<sup>4</sup>; le but visé est, bien évidemment, d'obtenir un histogramme à la fois le plus tassé possible vers la gauche (les 'très bons' éléments) et ne dépassant pas (les 'mauvais' éléments) la qualité théorique recherchée (cf. plus bas).

- l'exemple 1 correspond au maillage d'une bielle (géométrie fournie par Matra Datavision).
- l'exemple 2 correspond au maillage d'un moule entourant la bielle ci-dessus (en fait, seule la moitié de cette bielle est prise en compte pour tenir compte de la symétrie existante).

<sup>4</sup>Pour une meilleure visibilité, seule la répartition des éléments plus mauvais que 3 est montrée.

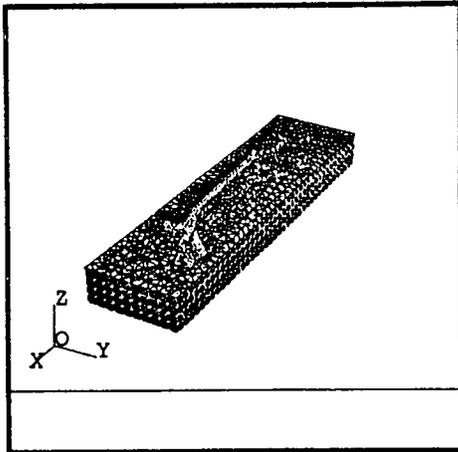


Figure 8: *Exemple2.*

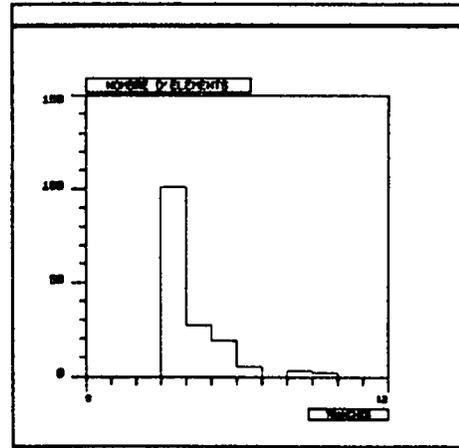


Figure 9: *Histogramme.*

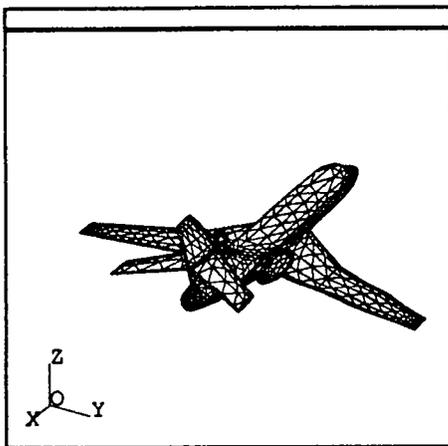


Figure 10: *Exemple3.*

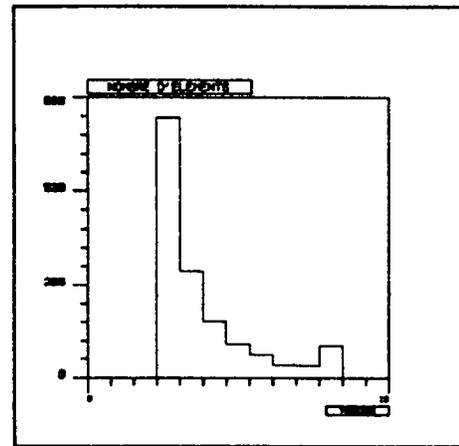


Figure 11: *Histogramme.*

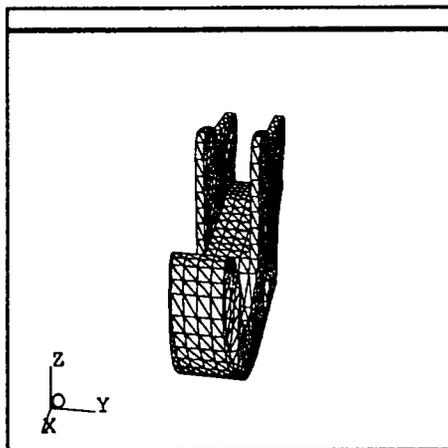


Figure 12: *Exemple4.*

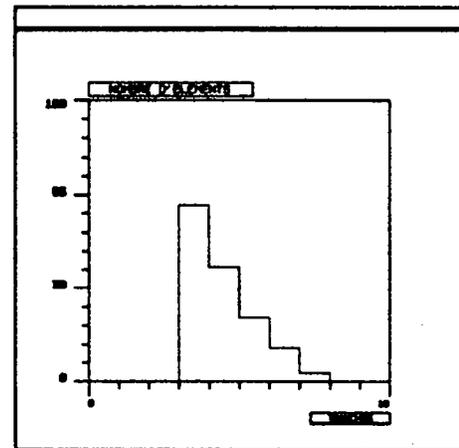


Figure 13: *Histogramme.*

- l'exemple 3 est le maillage de la zone comprise entre un avion (géométrie fournie par Dassault Aviation) et une sphère englobante. Le dessin montre l'avion uniquement.
- l'exemple 4 montre le maillage d'un crochet.

Tous ces maillages ont été obtenus en découpant selon une progression géométrique les arêtes. Le choix d'une progression arithmétique donne des résultats analogues.

Cas de test	1	2	3	4
Nombre de points donnés	1378	2260	1997	1192
Nombre de tétraèdres	6960	22973	49760	5521
Nombre de sommets	1824	5243	.	1455
Temps CPU (en sec.)	869	2798	16694	634
Qualité théorique	9.95	8.01	42.87	7.10
Qualité atteinte	23.94	12.57	55.41	7.77
Nombre de tétraèdres "mauvais"	14	5	2	3
Nombre de tétraèdres "très bons"	6583	22856	48226	5401
Pourcentage	93	99	96	97

Tableau 1 : Résultats relatifs aux maillages ci-dessus.

Le tableau 1 donne, pour les 4 exemples sélectionnés, le nombre d'éléments générés, le nombre de sommets total, le temps nécessaire (IBM 6000), la qualité théoriquement atteignable, la qualité effectivement atteinte et le nombre de tétraèdres plus mauvais que la qualité théorique ainsi, enfin, que celui de ceux meilleurs que 3.

La qualité est celle définie ci-dessus et on donne comme qualité théorique la valeur obtenue en construisant le meilleur tétraèdre possible sur la face de donnée la plus mauvaise.

Cas de test	1	2	3	4	5	6
Nombre de tétraèdres	291	1237	3560	64000	5802	5521
Nombre de sommets	106	375	1009	.	.	1455
Temps partiel 1	2.3	6.5	14.1	210.	54.	27.
Temps partiel 2	6.93	65.66	34.12	2311.	299.	189.
Temps total (en sec.(IBM 6000))	40.8	231.	386.	16200.	1657.	634.

Tableau 2 : Temps requis en fonction de la taille du maillage.

Le tableau 2 donne le temps nécessaire en fonction du nombre de tétraèdres (et de sommets) du maillage final. La première valeur (notée temps partiel 1) représente le temps nécessaire à l'insertion des points initiaux (ceux de la frontière), la deuxième valeur (notée temps partiel 2) donne le temps nécessaire pour régénérer la frontière cumulé avec le temps précédent. La dernière valeur donne le temps total nécessaire pour obtenir le maillage final (la différence représente donc le temps passé pour créer et insérer les points internes et pour effectuer les régularisations).

Calculer la vitesse théorique de la méthode relève, me semble t'il, d'un exercice de style. Il me paraît plus informatif de regarder les résultats obtenus (cf. tableau 2) qui donnent une idée des performances atteintes effectivement.

Notons d'autre part qu'un calcul de complexité ne donnerait qu'une valeur biaisée puisqu'il ne s'appliquerait qu'au processus d'insertion de points (utilisé au départ de l'algorithme et lors de

la phase de création des points internes), sans tenir compte des difficultés numériques (erreurs d'arrondis) et sans prendre en compte la nécessité de préserver les arêtes et faces initialement données.

En effet, l'examen des différentes étapes du schéma proposé pour le mailleur fait apparaître, en terme de temps nécessaires, le découpage suivant:

- une phase de préparation (étapes a), b) et c)) marginale en coût,
- une phase d'insertion de points (étapes d) et h)) dont les coûts se situent essentiellement au niveau de:
  - la localisation du point à insérer: l'algorithme proposé est en  $n^2$  dans le pire des cas, il n'est donc pas optimal dans l'absolu (cf. [2] et [4] pour des études théoriques de complexité) mais se révèle rapide car on connaît a priori l'élément de départ ou un élément très voisin.
  - l'insertion proprement dite: sa complexité est celle des méthodes standards (voir en particulier les mêmes références).
- une phase de régénération de la frontière fournie (étapes e) et f)) relativement couteuse,
- une phase de marquage (étape g)) et une phase d'élimination de l'extérieur (étape i)) marginale,
- une phase de régularisation (étape j)) prépondérante en terme de coût.

En conclusion, le mailleur décrit semble numériquement fiable, raisonnablement rapide et donne des maillages de bonne qualité (au sens introduit). Il est à remarquer que les points internes sont constitués par paquets en suivant les arêtes. Ceci est une différence notable avec la méthode développée dans [13] et, nous semble-t-il, constitue un point fort de la méthode.

On n'a pas repris la méthode algébrique décrite dans [7] (construire un seul paquet de points internes et l'insérer en une seule fois) pour des raisons de vitesse; il est néanmoins clair qu'elle s'applique sans difficulté.

Bien que donnant des résultats satisfaisants en des temps de calcul raisonnable, la méthode peut probablement être encore améliorée. A cet égard il serait précieux (au regard des temps montrés dans le tableau 2) de réussir à limiter les régularisations finales comme en dimension 2 où elles sont dans la plupart des cas parfaitement inutiles.

Notons enfin un joli problème ouvert concernant la possibilité ou non de régénérer la frontière sans création de points ou dans le cas contraire la question de savoir combien de points faut-il ajouter au minimum.

**Remerciements :** Je tiens à remercier C. Aicardi qui dans le cadre de son stage de DESS effectué à la société Simulog a programmé la méthode de création des points internes et a réalisé de très nombreux tests pour la valider. Il faut aussi remercier E. Saltel pour les nombreuses discussions que nous avons eu tout au long de cette étude et pour son aide dans la programmation de plusieurs des algorithmes employés dans la méthode décrite. Enfin, je remercie J.M. Moreau de l'E.N.S.M.S.T. pour une discussion passionnée et passionnante dont un résultat est la formule permettant le calcul du nombre de combinaisons possibles pour remailler une coquille convexe de  $n$  éléments.

## 8 Bibliographie

- [1] M. BERNADOU ET ALL, *Modulef: une bibliothèque modulaire d'éléments finis*, INRIA, 1988.
- [2] J.D. BOISSONNAT, M. DEVILLERS-TEILLAUD, *On the randomized construction of the Delaunay tree*, Rapport de Recherches n° 1140, INRIA, 1991.
- [3] C. CHERFILS, F. HERMELINE, *Diagonal swap procedures and characterizations of 2D-Delaunay triangulations*, *M<sup>2</sup>AN*, vol 24, n° 5, pp. 613-625, 1990.
- [4] O. DEVILLERS, S. MEISER, M. TEILLAUD, *Fully dynamic Delaunay triangulation in logarithm expected time per operation*, Rapport de Recherches n° 1349, INRIA, 1990.
- [5] P.L. GEORGE, *Génération automatique de maillages. Applications aux méthodes d'éléments finis*, collection RMA n° 16, Masson, Paris, 1990.
- [6] P.L. GEORGE, *Modulef : Construction et modification de maillages*, Guide Modulef n° 3, INRIA, 1991.
- [7] P.L. GEORGE, *Génération de maillages par une méthode de type Voronoï*, Rapport de Recherches n° 1398, INRIA, 1991.
- [8] P.L. GEORGE, F. HECHT, E. SALTEL, *Fully automatic mesh generator for 3D domains of any shape*, *Impact of Computing in Science and Engineering*, vol 2, pp. 187-218, 1990.
- [9] P.L. GEORGE, F. HECHT, E. SALTEL, *Automatic mesh generator with specified boundary*, *Comp. Meth. in Appl. Mech and Eng.*, 92, pp. 269-288, 1991.
- [10] P.L. GEORGE, F. HECHT, M.G. VALLET, *Creation of internal points in Voronoi's type method. Control adaptation*, *Adv. Eng. Software*, vol 13, n° 5/6, pp. 303-313, 1991.
- [11] P.L. GEORGE, F. HERMELINE, *Delaunay's mesh of a convex polyhedron in dimension d. Application for arbitrary polyhedra*, *Int. J. Num. Meth. Ing.*, vol 33, pp. 975-995, 1992.
- [12] F. HECHT, E. SALTEL, *Emc2 : Editeur de maillages et de contours bidimensionnels*, Manuel d'utilisation, Rapport Technique n° 118, INRIA, 1990.
- [13] F. HERMELINE, *Une méthode automatique de maillage en dimension n*, Thèse , Université Paris 6, Paris , 1980.
- [14] F. HERMELINE, *Triangulation automatique d'un polyèdre en dimension N*, *Rairo, Analyse numérique*, vol 16, n° 3, pp. 211-242, 1982.
- [15] A. PERRONNET, *A generator of tetrahedral finite elements for multi-material object and fluids*, *Numerical grid generation in computational fluid mechanics'88*, Miami, 1988.
- [16] F.P. PREPARATA, M.I. SHAMOS, *Computational geometry, an introduction*, Springer-Verlag, 1985.
- [17] J.Y. TALON, *Algorithmes de génération et d'amélioration de maillages en 2D*, Rapport Technique n° 20, Artemis-Imag, 1987.
- [18] M.G. VALLET, *Génération de maillages anisotropes adaptés. Application à la capture de couches limites*, Rapport de Recherches INRIA n° 1360, 1991.
- [19] D.F. WATSON, *Computing the n-dimensional Delaunay Tesselation with applications to Voronoi polytopes*, *Computer Journal* 24, n° 2, pp. 167-172, 1981.
- [20] N.P. WEATHERILL, *The generation of unstructured grids using Dirichlet tessellation*, MAE report n° 1715, Princeton Univ., 1985.
- [21] N.P. WEATHERILL, *The integrity of geometrical boundaries in the 2-dimensional Delaunay triangulation*, *Comm. in Appl. Num. Meth.*, vol 6, pp. 101-109, 1990.

**ISSN 0249-6399**