



Generation de maillages par une methode de type Voronoi

Paul-Louis George

► To cite this version:

Paul-Louis George. Generation de maillages par une methode de type Voronoi. [Rapport de recherche] RR-1398, INRIA. 1991. inria-00075162

HAL Id: inria-00075162

<https://inria.hal.science/inria-00075162>

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.



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

N°1398

*Programme XG
Calcul Scientifique, Modélisation et
Logiciel numérique par Ordinateur*

GÉNÉRATION DE MAILLAGES PAR UNE MÉTHODE DE TYPE VORONOÏ

Paul Louis GEORGE

Mars 1991



Rapport - Génération de maillages

Génération de maillages par une méthode de type Voronoï

Paul Louis George

21 Février 1991

Génération de maillages par une méthode de type Voronoï

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. Le cas bidimensionnel est entièrement décrit et l'extension des résultats au cas tridimensionnel est indiquée.

Voronoi type mesh generation

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 two-dimensional case is described fully and the extension of the results to the three-dimensional case is indicated.

1 Introduction

Encore un papier sur la génération automatique de maillages par une méthode de type Voronoï ! de plus, essentiellement en dimension 2 ! En fait, pour répondre à la deuxième partie de cette remarque, il suffit de voir (comme cela sera précisé) que les résultats présentés ne dépendent pas de la dimension de l'espace. Pour tenter de balayer la première partie de l'objection, on ne peut que remarquer que, s'il existe de nombreux papiers relatifs au sujet, il reste beaucoup à faire pour trouver des méthodes numériquement complètement fiables, ce qui donne de la pertinence à cette étude.

Le problème que l'on veut résoudre est le suivant (on se place délibérément en dimension 2): *Trouver un maillage acceptable d'un domaine quelconque à partir de la donnée de son contour sous la forme d'un ensemble de segments.*

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 donc 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 triangles seulement. On note par K les triangles. Un triangle est défini par la liste orientée de ses sommets, ses arêtes sont également orientées, sa surface est positive. On associe à chaque triangle K l'ensemble de ses voisins (par ses arêtes) $V_i(K)$, $i = 1, 3$; le premier étant le triangle vu par le sommet 1 du triangle donné (avec la convention suivante: $V_i(K) = 0$ signifie que l'arête i (opposée au sommet i) de l'élément K est frontière). De plus, on notera par $B(K)$ le cercle circonscrit au triangle K .

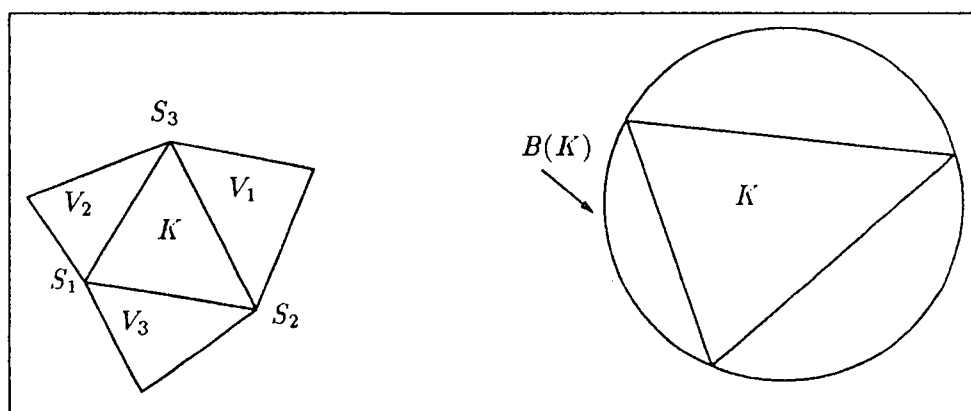


Figure 1: Définitions élémentaires.

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 dernière valeur représente la résolution ou pouvoir de séparation de la méthode. Elle est choisie de façon à permettre le calcul exact des surfaces (i.e., de longueurs au carré) et est donc 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 ?*

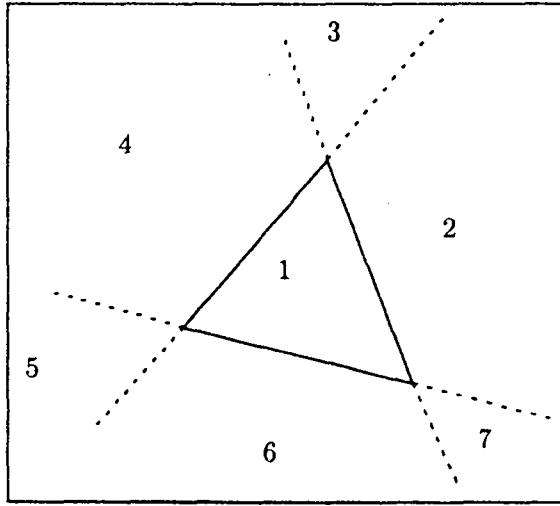


Figure 2: *Position d'un point.*

On considère $K_j^* = \{F_j, P\}$, $1 \leq j \leq 3$, les 3 triangles virtuels construits en joignant l'arête F_j de K au point P . Si ABC , dans cet ordre, est le triangle K , alors les trois triangles associés sont successivement ABP , APC et PBC . On définit par $Det(K)$ le déterminant suivant:

$$Det(K) = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix}.$$

où x_1, x_2, x_3 sont les abscisses des sommets de K et y_1, y_2, y_3 leurs ordonnées. Grâce à la normalisation précédente, le calcul de $Det(K)$ est numériquement exact. Notons que cette valeur est le double de la surface de K (par la suite, de manière impropre, on fera la confusion entre $Det(K)$ et la surface de K).

Alors:

Algorithme 1 :

- a) Si $\forall j, 1 \leq j \leq 3, Det(K_j^*) > 0 \leftrightarrow P \in K$, Fin;
- b) Si $\exists j_1, \dots, j_q, 1 \leq j_1 \neq \dots \neq j_q \leq 3$ tel que $Det(K_{j_1}^*) = \dots = Det(K_{j_q}^*) = 0 \leftrightarrow \{P \text{ est dans la région intersection des } q \text{ droites déterminées par les arêtes } F_{j_1} \dots F_{j_q}\}$, Fin.

- c) Si $\exists j_1, \dots, j_q, 1 \leq j_1 \neq \dots \neq j_q \leq 3$ tel que $Det(K_{j_1}^*) = \dots = Det(K_{j_q}^*) < 0 \leftrightarrow \{P \text{ est dans la région intersection des } q \text{ demi-plans définis par les arêtes } F_{j_1} \dots F_{j_q}\} \text{ ne contenant pas leur sommet opposé, Fin.}$ $\square$

P est déterminé par rapport aux 7 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 triangle quelconque du maillage;
- b) Si $\forall j, j = 1, 3 \quad Det(K_j^*) > 0$ alors $P \in K$, alors $\mathcal{C} = K$ Fin;
- c) Ou si $\exists j \quad Det(K_j^*) < 0$, trouver le triangle K' tel que $K \cap K' = F_j$ où F_j est l'arête 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 arête si $q = 1$ et un sommet si $q = 2$), Fin; $\square$

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

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

Ici, en dimension 2, seules 3 situations sont possibles:

1. $P \in K$,
2. P est sur une arête,
3. 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 l'arête $\bigcap F_{j_k}$.

2.3 Éléments 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.

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);

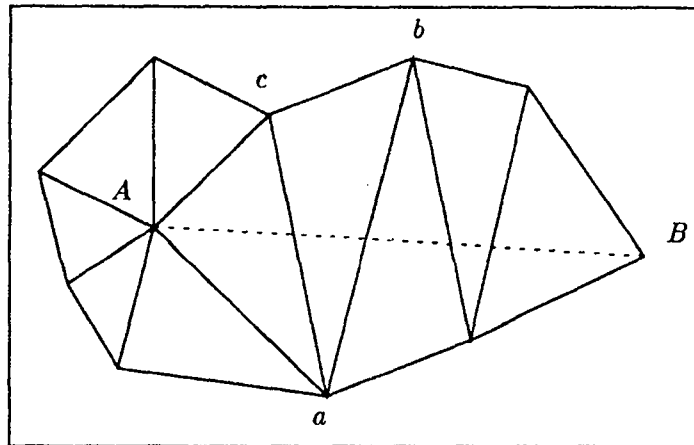


Figure 3: Triangles coupés par un segment.

- b) Parmi les éléments ayant A comme sommet, trouver celui dont l'arête ac (cf. dessin) opposée à A est coupée par AB , empiler cet élément;
- c) Prendre le voisin de K par l'arête ac , empiler cet élément;
- d) Soient abc les trois sommets de l'élément tête de pile, trouver parmi ab et bc (l'arête ac , coupée par AB est déjà connue) le couple tel que les puissances de a et b (respectivement b et c) par rapport à AB soient de signe opposé et prendre le voisin par l'arête ainsi repérée, empiler cet élément;
- e) Tant que l'élément tête de pile n'a pas B comme sommet, aller en d), sinon Fin. $\square$

La pile ainsi constituée est l'ensemble cherché.

Démonstration: évidente encore une fois.

Remarquons que le calcul des puissances utilisées ci-dessus est équivalent au calcul des *Det* incriminés.

2.4 Retournement d'arêtes

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

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., le cercle circonscrit $B(K)$ à chacun de ses éléments K ne contient pas d'autres points (sommets de triangles du maillage) que les 3 sommets¹ de K . La méthode de Voronoï consiste à construire un nouveau maillage possédant P comme sommet de triangles, elle peut se schématiser de la façon suivante:

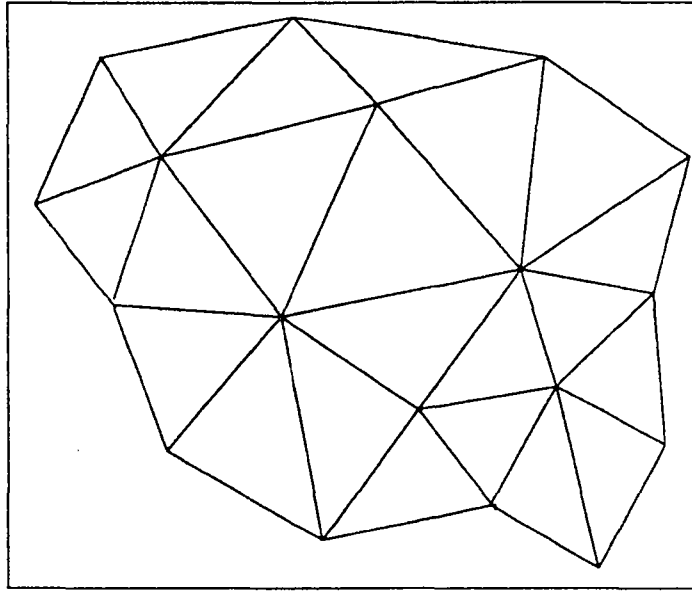


Figure 4: *Maillage initial et point à insérer.*

Algorithme 4 :

- a) Trouver l'élément ou les 2 éléments de T contenant P ;
- b) Mettre ce(ces) élément(s) dans une pile S ;
- c) Compléter récursivement (en prenant en compte les voisins des triangles 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 arêtes externes de S c'est-à-dire les arêtes 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. □

T est une triangulation de Delaunay incluant P comme sommet de triangles.

Démonstration: voir [9] ou [10] (entre autres).

Remarque:

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

¹Il s'agit ici de l'une des caractérisations des maillages de Delaunay.

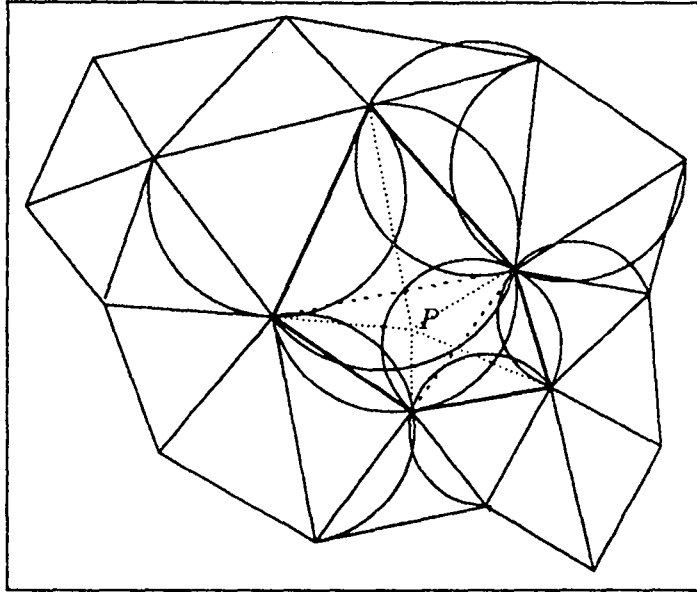


Figure 5: Insertion d'un point (méthode 1).

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 ou les 2 éléments de T contenant P ;
- b) Mettre ce(ces) élément(s) dans une pile S ;
- c) Trouver $F_1 \dots F_p$ les arêtes externes de S et joindre P à ces arêtes; on crée de la sorte 3 ou 4 éléments;
- d) Vider S et mettre ces éléments dans S ;
- e) Chercher récursivement parmi les voisins des éléments de S ceux dont le cercle circonscrit contient P et retourner l'arête externe de S commune à ces éléments, mettre à jour S et tant que S est non vide aller en e), sinon Fin. $\square$

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 la procédure de retournement d'arêtes vue ci-dessus et s'interprète comme une optimisation locale.

Le nouveau maillage est localement de Delaunay.

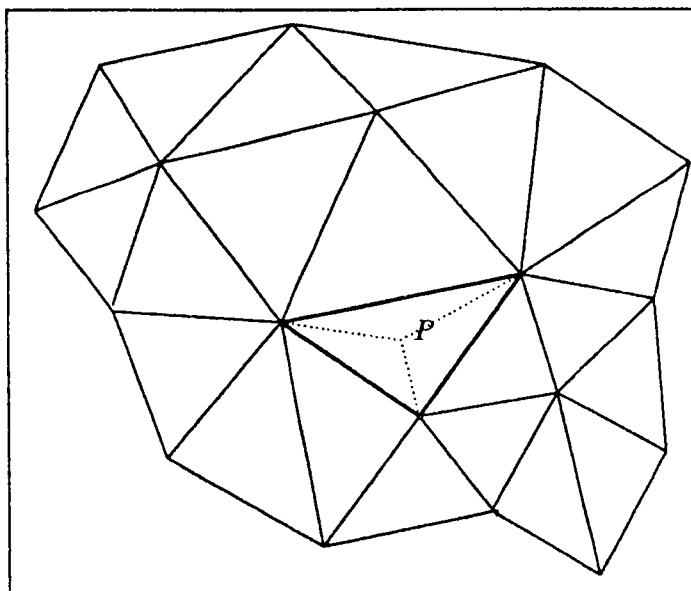


Figure 6: *Insertion d'un point (méthode 2).*

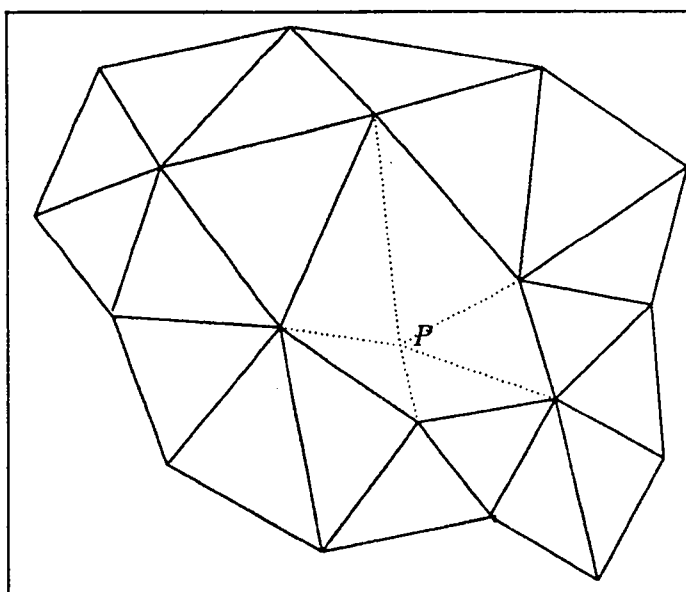


Figure 7: *Optimisation (méthode 2).*

3.3 Insertion directe

La méthode de Voronoï est numériquement peu fiable car savoir situer un point par rapport à un cercle (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 fausse (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 ou les 2 éléments 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 arêtes externes de S c'est-à-dire les arêtes non communes à 2 éléments de S ;
 Définir les p triangles virtuels $K_j^* = (F_j, P)$, $1 \leq j \leq p$;
 Définir V_F l'ensemble des sommets des arêtes 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 triangle de S' d'arête F_j ;
 - Si $Det(K_j^*) = 0$, $S' = S' \cup K_j$ où K_j est le triangle de T non contenu dans S' ayant l'arête 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 triangle quelconque de T possédant un sommet, différent de P , contenu dans $V_{S'}$ sans être dans V_F);
- i) Construire le nouveau maillage comme $T = (T - S) \cup \{F_j, P\}_j \quad 1 \leq j \leq p$, Fin. □

Cet algorithme est constructif.

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

Remarques:

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.

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:

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

$$\Leftrightarrow P \text{ est dans le cercle circonscrit de tout élément de } S\}$$

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:

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

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.

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 segments la constituant.

Une première idée consiste à:

- Construire l'ensemble des points P ci-dessus comme l'ensemble des points extrémités des arêtes de la frontière donnée;
- Construire 4 points supplémentaires tels que tout point donné soit interne au quadrilatère formé par ces 4 points;
- Mailler ce quadrilatère avec 2 triangles (de Delaunay);
- Appliquer l'un quelconque des processus d'insertion de points pour construire, pas à pas, un maillage du quadrilatère composé de triangles dont les sommets sont, d'une part les 4 points ajoutés et d'autre part tous les points de la frontière.

Ce maillage *ne contient pas*, en général, dans la liste des arêtes de ses éléments les arêtes 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 arêtes 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 arêtes données contienne ces arêtes. On trouvera dans [7] l'introduction de la notion d'arêtes 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 arêtes manquantes et à les couper en deux par création du point milieu. 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 arêtes données (arêtes initiales ou arêtes initiales coupées en deux). Intuitivement, une application récursive du procédé donne le résultat car elle revient à former numériquement un ensemble de données Delaunay admissible. On trouvera dans [11] la mise en application de cette méthode et dans [17] une extension astucieuse.

Une autre idée consiste à transformer localement le maillage initial pour régénérer les arêtes manquantes (et supprimer les arêtes gênantes). Plusieurs manières de procéder existent, parmi lesquelles nous choisissons les deux méthodes exposées ci-dessous.

4.2 Une solution naturelle

Soit T le maillage précédent construit en insérant les points extrémités des arêtes données. Soit AB une arête de donnée non présente dans T , on définit T_{AB} l'ensemble des triangles dont une

arête est coupée par AB . Ainsi $T = T' \cup T_{AB}$ où T' est l'ensemble des triangles non intéressés par AB .

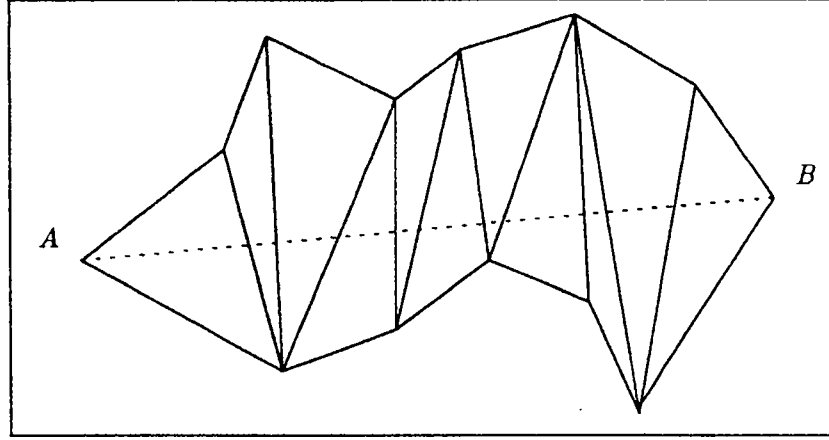


Figure 8: *Ensemble de départ.*

T_{AB} définit un polygone (non convexe) dont on note les sommets par $a_k, k = 1, n_1$ avec $a_1 = A$ et $a_{n_1} = B$ pour la partie située d'un côté de AB et $b_k, k = 1, n_2$ avec $b_1 = A$ et $b_{n_2} = B$ pour la partie située dans le demi-plan délimité par AB ne contenant pas les a_k .

On note par Pb^0 le problème à résoudre i.e., le remaillage de T_{AB} . Il est clair que, si on note Pb_1^0 le problème du remaillage du polygone constitué par les a_k et Pb_2^0 celui du remaillage du polygone formé par les b_k , le problème initial revient d'une part, à traiter indépendamment les deux sous-problèmes Pb_i^0 dont la complexité est moindre, et d'autre part, en fait à ne traiter que l'un de ces sous-problèmes. Considérons donc le problème Pb_1^0 .

On trace AB et on cherche dans a_k le point le plus proche de AB , soit a_l ce point, on trace Aa_l et a_lB et on forme le triangle ABa_l . La solution du problème Pb_1^0 est donc la suivante:

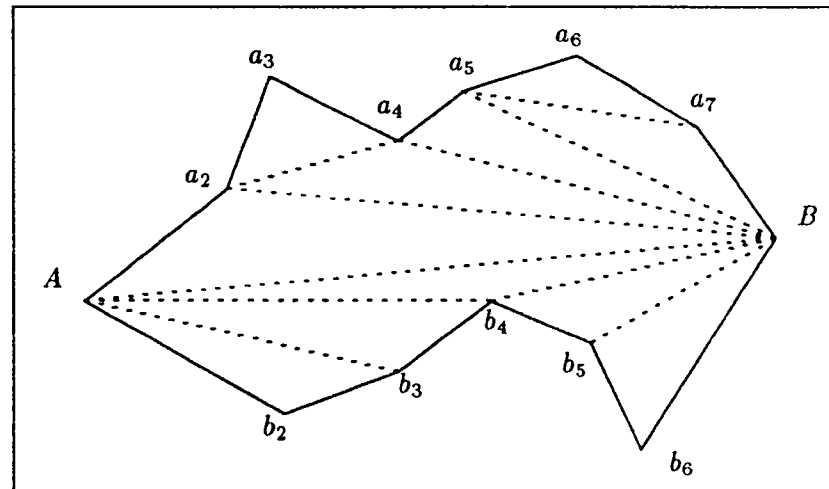


Figure 9: *Forçage naturel.*

- construire le triangle a_1, a_{n_1}, a_l ;

- définir le problème Pb_1^1 qui est celui du maillage du polygone $a_1, a_2, \dots, a_l$;
- définir le problème Pb_1^2 qui est celui de maillage du polygone $a_l, a_{l+1}, \dots, a_{n_1}$;

Le problème est donc décomposé en sous-problèmes de complexité inférieure, appliquant récursivement le procédé, la solution est obtenue (un polygone réduit à trois points donne la solution et arrête la décomposition).

La description complète de cette méthode figure dans [6].

4.3 Une solution par retournement d'arêtes

La méthode ci-dessus donne le résultat mais celui-ci est peu équilibré, par suite une autre solution est proposée, basée sur le procédé de retournement d'arête.

L'idée de base est la suivante: si le premier triangle de T_{AB} (celui contenant A comme sommet) et le second triangle de cet ensemble (le voisin d'arête commune celle opposée à A) forment un polygone convexe, on peut appliquer le retournement d'arête: d'une part, l'arête commune disparaît et d'autre part, l'arête créée ne coupe pas AB (i.e. on a avancé) (on a le même résultat pour le dernier élément (de sommet B) et l'avant dernier).

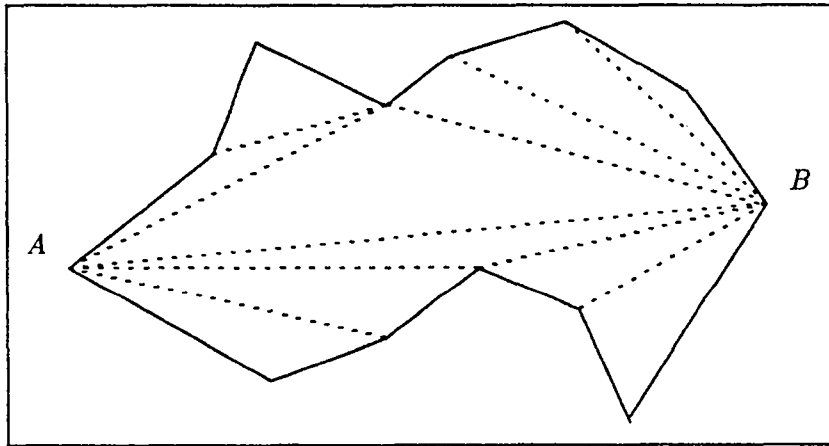


Figure 10: Forçage par retournement.

La solution proposée est donc la suivante:

Algorithme 7 :

- a) définir T_{AB} ;
- b) essayer de retourner l'arête commune entre le premier triangle et le second (le dernier et l'avant dernier)
 - b1) le retournement marche, aller en c);
 - b2) le retournement ne marche pas, parcourir T_{AB} jusqu'à ce qu'un retournement marche, aller en c)
- c) mettre à jour T_{AB}
- d) si $Card(T_{AB}) = 2$ Fin, sinon, aller en b)

□

Le polygone T_{AB} initial est remaillé et ce maillage contient AB comme arête de triangles.

Démonstration: on utilise un théorème, démontré par exemple dans [2], qui montre que l'on peut passer d'une triangulation quelconque à une triangulation de Delaunay. En fait on l'utilise à l'envers en constatant qu'il n'utilise que le processus de retournement d'arêtes, processus hautement réversible.

Notons que cette méthode, ou au moins une de ses variantes, est utilisée dans [8]

4.4 Définition de l'intérieur du domaine

Comme $T = T' \cup T_{AB}$ et que T_{AB} a été remaillé de telle sorte que AB existe, AB existe dans T . Sous la seule hypothèse que la frontière du domaine n'est pas croisée (ce qui est trivialement vérifié), on peut appliquer le processus à toutes les arêtes manquantes de façon à obtenir un maillage T contenant exactement toutes les arêtes données. Ce maillage permet 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 l'arête commune n'est pas une arête frontière et que le voisin n'est pas déjà marqué, le marquer avec le nombre n et empiler le triangle si ses 3 arêtes ont été examinées;
- c) Tant que la pile d'indice n peut évoluer, aller en b)
- d) Sinon, tant qu'il reste des triangles non marqués, prendre un triangle non marqué, faire $n = n + 1$, le marquer avec n et aller en b) □

Ensuite, il suffira de retenir tous les triangles 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 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 triangle 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 triangles;
- La forme des triangles

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 [3] sur les méthodes d'adaptation et [14] pour le cas anisotrope ou directionnel).

5.2 Méthode algébrique par éléments

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 la moyenne entre les longueurs des 2 arêtes ayant ce point comme extrémité.

On propose le processus suivant:

Algorithme 9 :

- Faire, pour tout les éléments K de T :
 - Examiner les 3 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);
 - Si l_i est plus grand que $h_{i1} + h_{i2}$ 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);
 - Si les 3 n_i sont égaux, générer les 3 ensembles de n_i points sur les arêtes de K ;
 - Sinon, poser $n1 = \max(n_i)$ et $n2 = \min(n_i)$ et générer $n1$ points sur les 2 arêtes dont le n_i vaut $n1$ ou est proche de $n1$ et $n2$ points sur la troisième arête;
 - Appliquer une méthode algébrique (voir ci-dessous) pour générer les $n_i(n_i - 1)/2$ ou les $n2.n1$ points internes correspondant;
 - Prendre tous les points ainsi créés et empiler ceux qui:
 - * ne sont pas sur une frontière donnée;
 - * ne sont pas trop proches (au sens de h) d'un point existant;
- Fin □

Dans la suite de cette section, nous allons donner quelques indications relatives aux différentes étapes de cet algorithme.

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

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

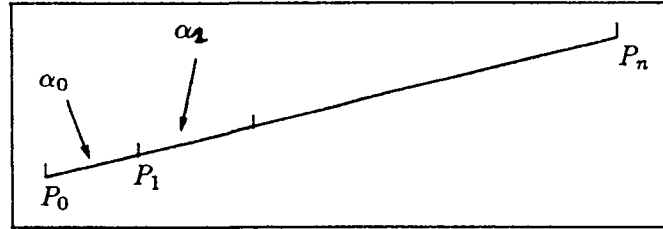


Figure 11: Découpe d'une arête.

Le problème est de calculer n et r donc les α_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:

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

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 α (β) calculé comme suit:

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

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

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

Les autres arêtes de K sont traitées de la même manière vis-à-vis des valeurs qui leur sont attribuées. En pratique, seul un paramètre ou deux paramètres n sont possibles, on calcule donc a priori les 3 valeurs réelles et on en déduit la ou les valeurs possibles.

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 α_i sont alors connus et les points intermédiaires sont parfaitement déterminés.

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

5.2.3 Découpe de l'élément (cas 1)

La méthode algébrique [3] permet de mailler l'intérieur d'un domaine analogue à un triangle à partir d'une discrétisation polygonale donnée de son contour supposé formé de 3 côtés (topologiquement parlant) et comportant un nombre égal de points sur chacun de ceux-ci. Dans notre cas, le domaine est un simple triangle et son contour se réduit à ses 3 arêtes, on suppose que chacune possède n points intermédiaires et on applique la méthode algébrique générale dans

une version simplifiée adaptée à la particularité du cas traité (domaine convexe de côtés droits). Cette méthode n'est pas utilisée pour mailler le triangle mais seulement pour créer des points internes.

On classe les 3 côtés en fonction de leur longueur (de manière à isoler le "grand" côté des deux autres dit "petits").

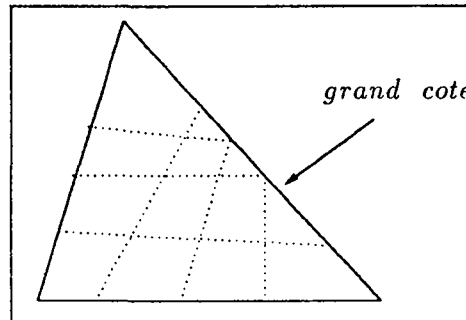


Figure 12: *Le motif à un seul paramètre.*

Création des points dans K . On connecte les points se correspondant d'un côté à l'autre du triangle. Pour ce faire, on construit deux droites déterminées par les points des arêtes logiquement reliés (cf. figure 12); l'intersection de ces droites donne le point. La première droite passe par un point intermédiaire d'un "petit" côté et par son correspondant sur le "grand" côté; la seconde utilise l'autre "petit" côté.

5.2.4 Découpe de l'élément (cas 2)

Pour traiter le cas, fréquent dans la pratique, où on a deux paramètres de découpe sensiblement différents (n_1 et n_2), on définit une méthode algébrique adaptée.

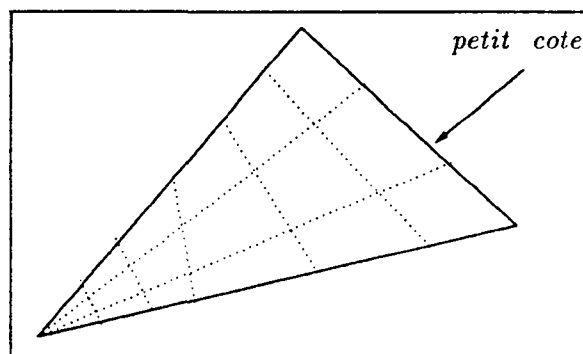


Figure 13: *Le motif à 2 paramètres.*

On classe, de même, les 3 côtés en fonction de leur longueur (de manière à isoler le "petit" côté des deux autres dit "grands").

Création des points dans K . On connecte les points se correspondant d'un côté à l'autre du triangle. Pour ce faire, on construit deux droites déterminées par les points des arêtes

logiquement reliés (cf. figure 13); l'intersection de ces droites donne le point. La première droite passe par un point intermédiaire d'un "grand" côté et par son correspondant sur l'autre "grand" côté; la seconde passe par le sommet commun aux 2 "grands" côtés et le point voulu du "petit" côté.

5.2.5 Rejet des points

Appliquant l'opération à tous les triangles marqués de T , on dispose, à ce stade, d'une série de points placés sur les arêtes de T et à l'intérieur de ses éléments.

Le résultat est un ensemble de points bien situés localement (dans chaque triangle) mais, à cause de la localité du procédé, il est en général possible d'avoir généré trop de points et en particulier, le même point plusieurs fois ou 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 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 proche au regard des h des points examinés;
- les points des frontières sont enlevés de la liste (cf. respect de la frontière).

5.2.6 Insertion des points

Les points retenus sont insérés 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 arêtes du contour du domaine et les points spécifiés;
- b) Détermination de la position de quatre points supplémentaires tels que le quadrilatère formé par ceux-ci englobe tous les points du nuage;
- c) Maillage de ce quadrilatère à l'aide de deux triangles;
- 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) Marquage des triangles de T en fonction de leur position vis-à-vis du contour;
- g) Création des points internes et insertion de ceux-ci;
- h) Elimination des triangles extérieurs;
- i) Régularisation éventuelle.

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

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

Sont donnés maintenant quelques exemples de maillages obtenus en employant cette méthode. Pour chaque exemple, on donnera le nombre de triangles, le nombre de points, le temps CPU (APOLLO DN 4000), la qualité du résultat c'est-à-dire la répartition des triangles en fonction de leur forme mesurée via la quantité $Q = \alpha \frac{h}{\rho}$ où α est un coefficient de normalisation tel que $Q = 1$ pour un triangle équilatéral, h est le diamètre de l'élément mesuré, ρ est le rayon de son cercle inscrit: notons que Q augmente quand le triangle s'applatit.

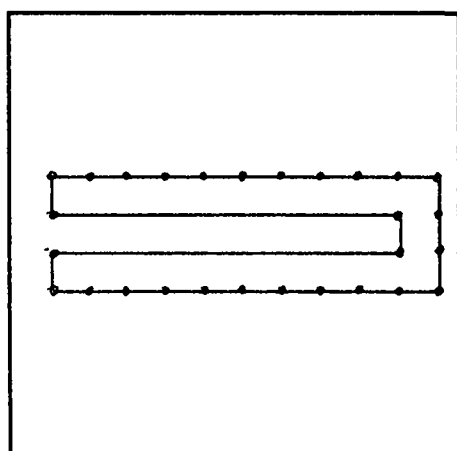


Figure 14: "porc-épic": la donnée.

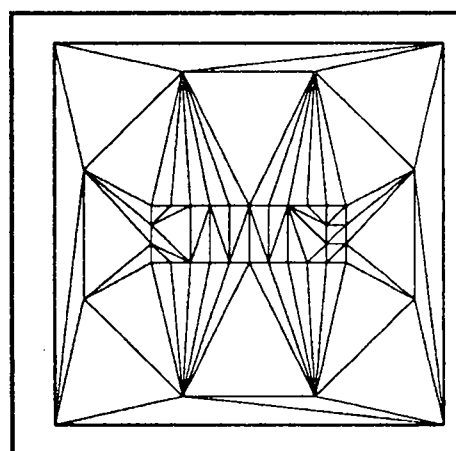
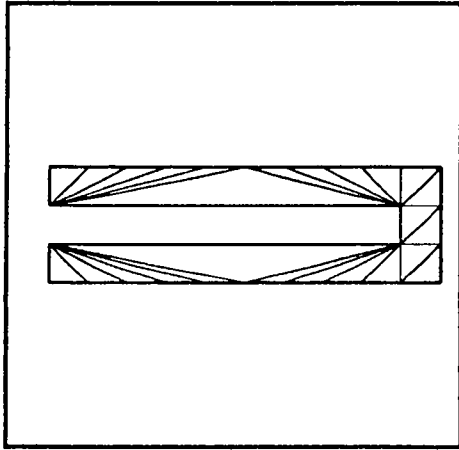
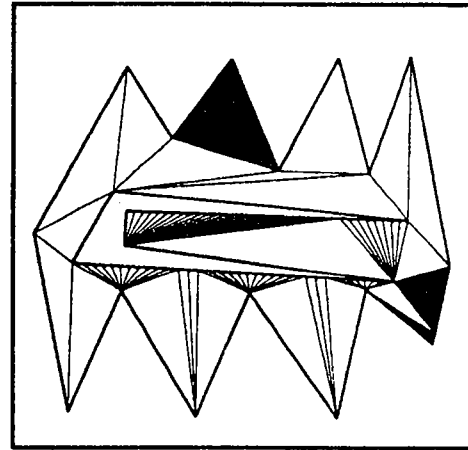
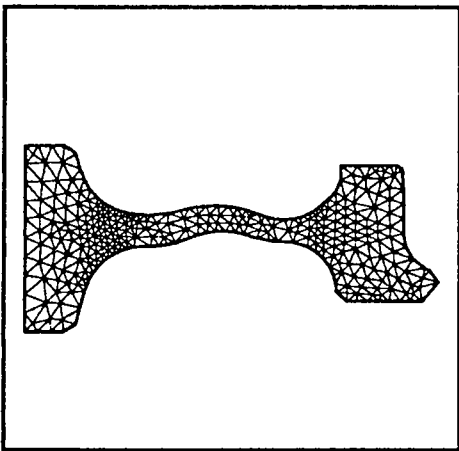
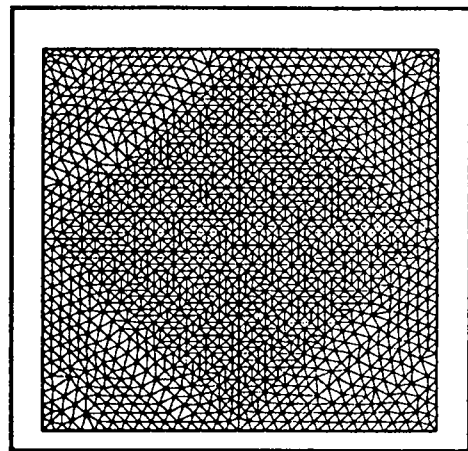
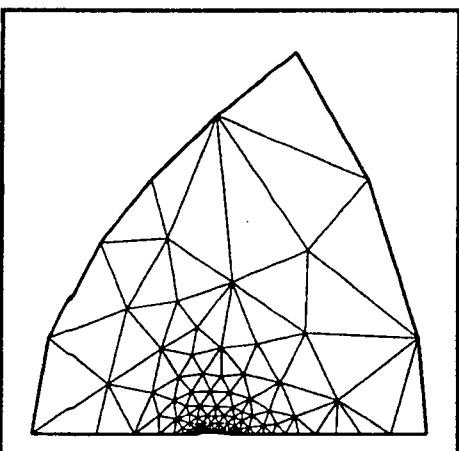
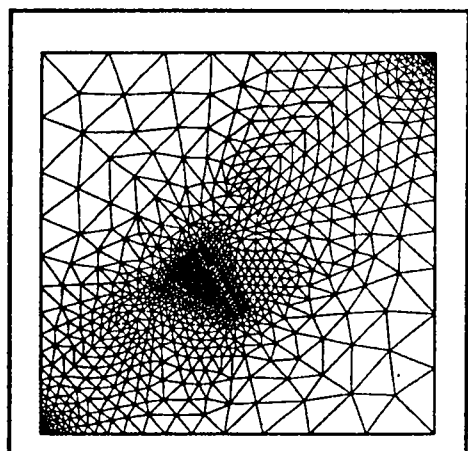


Figure 15: "porc-épic": le maillage englobant.

La figure 14 (cas 1 du tableau 1) montre un exemple de contour "mal" discrétisé produisant la non création de ses arêtes dans le maillage du quadrilatère englobant (figure 15). La phase de régénération de la frontière est donc utilisée pour construire les arêtes manquantes et obtenir le résultat (figure 16). L'exemple montré sur la figure 17 (cas 2) a été construit pour tester cette phase de recréation d'arêtes manquantes, le test n'est en aucun cas réaliste, mais il illustre la robustesse de la méthode.

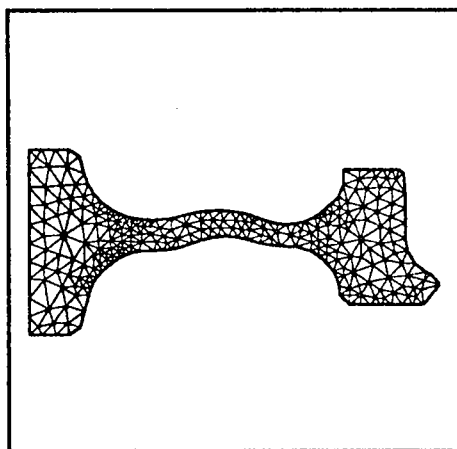
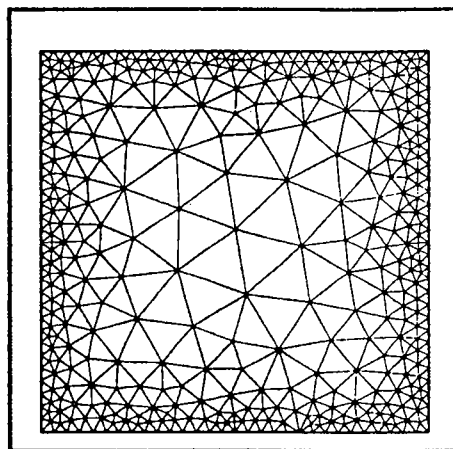
La figure 18 (cas 3 du tableau 1) montre l'exemple "habituel" utilisé dans Modulef qui est celui d'une coupe d'une roue. La figure 19 (cas 4) illustre le cas d'un carré et, contrairement à ce que l'on pourrait croire a priori, nous paraît plus significative du comportement de la méthode au sens où une géométrie simple n'est pas généralement correctement maillée par un mailleur automatique.

La figure 20 (cas 5 du tableau 1) montre l'exemple classique d'un profil et la figure 21 (cas 6) le cas d'un carré défini par son contour, discrétisé à l'aide de progressions variables, et quelques points internes spécifiés (avec des h petits).

Figure 16: *Résultat.*Figure 17: *Autre exemple de test.*Figure 18: *La coupe d'une roue.*Figure 19: *Le cas d'un carré.*Figure 20: *Exemple de profil.*Figure 21: *Cas de points internes spécifiés.*

Par ailleurs, un paramètre ω 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 figure 22 reprend le cas de la roue déjà vue pour un paramètre $\omega = 1.2$, dans ce cas la différence est faible car la dilution est une fonction de la distance entre un point et la frontière et de ce fait n'est jamais importante. La figure 23 illustre le cas d'un carré, cas dans lequel, en diminuant la taille des segments discrétisant le contour, on augmente la distance relative entre un point interne (par exemple le centre de gravité) et la frontière.

Tous ces maillages ont été obtenus en découpant selon une progression géométrique les arêtes. Le choix d'une progression arithmétique donne des résultats satisfaisants bien qu'un peu moins bons (en terme de qualité).

Figure 22: Influence du paramètre ω .Figure 23: Influence du paramètre ω .

Cas de test	1	2	3	4	5	6
Nombre de triangles	26	244	536	2976	199	2290
Nombre de sommets	28	244	325	1549	126	1186
Temps CPU (en sec.)	6.2	12.3	27.4	26.7	15.	100.
Qualité	15.04	89.33	3.37	1.84	2.49	2.73
Nombre de triangles "mauvais"	16	231	2	0	15	13
Qualité après régularisation	-	-	1.68	-	2.11	1.84
Nombre de triangles "mauvais"	-	-	0	-	5	0

Tableau 1 : Résultats relatifs aux maillages ci-dessus.

Le tableau 1 donne comme qualité du maillage la plus mauvaise valeur atteinte et indique combien de triangles sont de mesure plus mauvaise que 2. Dans l'implémentation, on effectue automatiquement une régularisation dans le cas où le résultat brut possède au moins un triangle dont la qualité dépasse la valeur 2 (ceci augmente légèrement le temps CPU et correspond aux deux dernières lignes du tableau).

Cas de test	1	2	3	4	5	6	7
Nombre de triangles	1398	2976	5484	12346	21832	33852	121674
Nombre de sommets	740	1549	2823	6294	11077	17127	61218
Temps CPU (en sec.)	14.4	26.7	47.4	112.8	220.3	380.6	2170.

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

Le tableau 2 donne le temps nécessaire en fonction du nombre de triangles créés. L'exemple pris est le carré déjà utilisé pour lequel on fait varier le nombre de points sur le contour (notons que pour cet exemple, la qualité est toujours de l'ordre de 2).

Calculer la vitesse théorique de la méthode relève d'un exercice de style. Il nous 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, sans tenir compte des difficultés numériques (erreurs d'arrondis) et sans prendre en compte la nécessité de préserver les arêtes initialement données.

En conclusion, le mailleur décrit semble numériquement fiable, rapide et donne des maillages de bonne qualité. Il est à remarquer que les points internes sont constitués en une seule étape et insérés en un seul paquet. Ceci est une différence notable avec la méthode développée dans [9] et, nous semble-t-il, constitue un point fort de la méthode. Il reste à étendre les résultats à la dimension 3 (cf. ci-dessous pour les principales idées relatives à ce cas²)

7 Application à la dimension 3

Nous allons, dans cette courte section, et avant d'écrire un papier directement consacré à la dimension 3, tenter de recenser les extensions possibles des résultats présents au cas de la dimension 3. Nous reprenons l'ordre dans lequel ils apparaissent.

- Outils : la notion de triangle est remplacée par celle de tétraèdres définis par la liste orientée de 4 sommets, possédant 4 voisins par face (la notion de face remplace la notion d'arête). $B(K)$ est la sphère circonscrite à l'élément K .
- Normalisation : le souci est de tenir exact non plus des surfaces mais des volumes (i.e. une longueur au cube).
- Position d'un point : l'algorithme 1 se généralise sans difficulté.
- Recherche de l'élément contenant un point P donné : l'algorithme 2 se généralise sans difficulté.
- Éléments coupés par un segment : l'algorithme 3 se généralise.
- Éléments coupés par un triangle : point nouveau à développer.
- Retournement d'arêtes : sa généralisation conduit à casser 2 tétraèdres pour en créer 3 en cassant une face ou à réaliser l'opération inverse.

²En fait, la présente étude, vise, en particulier, à comprendre *numériquement* la méthode avant de l'appliquer à la dimension 3.

- Insertion de points : les 3 méthodes s'étendent sans problème.
- Respect d'une frontière : l'extension au cas $3D$ est loin d'être triviale (voir [5] et [6]).
- Marquage de l'extérieur : l'algorithme 8 convient dans son esprit.
- Points internes : la méthode algébrique peut être étendue.

Annexe technique

Le mailleur décrit dans ce papier est en cours d'incorporation dans la version 91 du code Modulef [1] pour remplacer le mailleur existant à ce jour. Le module d'entrée est la subroutine TRIHER dont suivent les premières lignes:

```

SUBROUTINE TRIHER(M1,M2,XM1,XM2,COOR,IACOOR,NREFPC,NREFS,NREFA,
+               NPMAX,NBRNOE,NCOMP,NTACO,IOPTIN,NPROPA,
+               NUMPIN,NBRINT,POIDS,NDSDE,NFNOPO,NINPOPO)
C *****
C BUT : MAILLAGE PAR LA METHODE DE VORONOI
C --- CAS 2D , CONTOUR QUELCONQUE , DONNEES DU CONTOUR .
C *****
C IN :
C --
C M1,XM1,M2,XM2 : LE(S) SUPER(S) TABLEAU(X) ( EN GENERAL M1 = M2 )
C COOR : LES COORDONNEES DES POINTS FOURNIS
C IACOOR : L'ADRESSE DE COOR DANS M
C NREFPC : LA REFERENCE DES POINTS CARACTERISTIQUES
C NREFS : LA REFERENCE DES POINTS DU CONTOUR
C NREFA : LA REFERENCE DES ARETES FRONTALIERES
C NPMAX : UN MAJORANT DU NOMBRE DE SOMMETS A GENERER
C NBRNOE : NOMBRE DE NOEUDS DU CONTOUR DE LA PARTIE A MAILLER
C NCOMP : NOMBRE DE COMPOSANTES CONNEXES DE LA FRONTIERE
C NTACO : NOMBRE DE NOEUDS DE CHACUNE DES COMPOSANTES
C IOPTIN : OPTION DE CREATION OU DE FOURNITURE DES POINTS INTERNES
C NPROPA : COEFFICIENT DE PROPAGATION ET TYPE DE CELLE-CI
C NBRINT : NOMBRE DE POINTS INTERNES FOURNIS
C POIDS : POIDS DES POINTS INTERNES FOURNIS
C NDSDE : NUMERO DU SOUS-DOMAIN
C NF(NI)NOPO : FICHIER ET NIVEAU DE LA S.D.S. NOPO
C *****

```

Pour appeler ce mailleur, on utilise le processeur APNOXX [4] qui permet de construire un fichier de données contenant les données nécessaires. Celles-ci sont constituées des points et des lignes constitutives du contour du domaine à mailler puis des paramètres utiles à l'appel du module TRIHER, cet appel étant déclenché par la présence du mot-clé TRIH. Le résultat obtenu est une Structure de données de type NOPO, i.e., la structure de maillage du code Modulef.

8 Bibliographie

- [1] M. BERNADOU ET ALL, Modulef: une bibliothèque modulaire d'éléments finis, INRIA, 1988.

- [2] C. CHERFILS, F. HERMELINE, Diagonal swap procedures and characterizations of 2D-Delaunay triangulations, *M²AN*, vol 24, n° 5, pp. 613-625, 1990.
- [3] P.L. GEORGE, Génération automatique de maillages. Applications aux méthodes d'éléments finis, collection RMA n° 16, Masson, Paris, 1990.
- [4] P.L. GEORGE, Modulef : Construction et modification de maillages, Rapport Technique n° 104, INRIA, 1989.
- [5] 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.
- [6] P.L. GEORGE, F. HECHT, E. SALTEL, Automatic mesh generator with specified boundary, à paraître dans *Comp. Meth. in Appl. Mech and Eng.*, 1991.
- [7] P.L. GEORGE, F. HERMELINE, Delaunay's mesh of a convex polyhedron in dimension d. Application for arbitrary polyhedra, soumis à *Int. J. Num. Meth. Ing.*, 1990.
- [8] F. HECHT, E. SALTEL, Emc2 : Editeur de maillages et de contours bidimensionnels, Manuel d'utilisation, Rapport Technique n° 118, INRIA, 1990.
- [9] F. HERMELINE, Une méthode automatique de maillage en dimension n, Thèse , Université Paris 6, Paris , 1980.
- [10] F. HERMELINE, Triangulation automatique d'un polyèdre en dimension N, *Rairo, Analyse numérique*, vol 16, n° 3, pp. 211-242, 1982.
- [11] A. PERRONNET, A generator of tetrahedral finite elements for multi-material object and fluids, *Numerical grid generation in computational fluid mechanics'88*, Miami, 1988.
- [12] F.P. PREPARATA, M.I. SHAMOS, *Computational geometry, an introduction*, Springer-Verlag, 1985.
- [13] J.Y. TALON, Algorithmes de génération et d'amélioration de maillages en 2D, Rapport Technique n° 20, Artemis-Imag, 1987.
- [14] M.G. VALLET, Génération de maillages anisotropes adaptés. Application à la capture de couches limites, Rapport de Recherches INRIA, à paraître, 1991.
- [15] D.F. WATSON, Computing the n-dimensional Delaunay Tessellation with applications to Voronoi polytopes, *Computer Journal* 24, n° 2, pp. 167-172, 1981.
- [16] N.P. WEATHERILL, The generation of unstructured grids using Dirichlet tessellation, MAE report n° 1715, Princeton Univ., 1985.
- [17] 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