Encore Delaunay

Houman Borouchaki, Paul-Louis George

▶ **To cite this version:**

Houman Borouchaki, Paul-Louis George. Encore Delaunay. [Rapport de recherche] RR-2461, INRIA. 1995. inria-00074215

HAL Id: inria-00074215
https://inria.hal.science/inria-00074215

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

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Encore Delaunay

Houman BOROUCHAKI - Paul-Louis GEORGE

N° 2461
Janvier 1995

PROGRAMME 6

Encore Delaunay

Houman BOROUCHAKI et Paul Louis GEORGE

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France.

Résumé.

Nous proposons un algorithme efficace et rapide pour la construction incrémentale de la triangulation de Delaunay d'un nuage de points dans R^d. Plusieurs procédures d'accélérations ont été conçues : localisation du point inséré, formules d'évaluation des quantités géométriques importantes des nouveaux simplexes à partir de celles des anciens et mise à jour des relations d'adjacence. Le schéma résultant semble l'un des plus rapides en temps cpu, à notre connaissance; en particulier, pour un nuage de points aléatoires de R^3, il est capable de générer la triangulation à une vitesse constante de 15000 tétraèdres par seconde sur une station de travail raisonnablement rapide.

Mots-clés

Triangulation, Triangulation de Delaunay, Diagrammes de Voronoï, Enveloppe Convexe.

Still Delaunay

Abstract.

An efficient algorithm for Delaunay triangulation of a given set of points in d dimensions based on the point insertion technique is presented. Various steps of the triangulation algorithm are reviewed and many acceleration procedures are devised to speed up the triangulation process. New features include the search of a neighbouring point by the layering scheme, locating the containing simplex by random walk, formulas of important geometrical quantities of a new simplex based on those of an existing one, a novel approach in establishing the adjacency relationship and the use of connexion matrix. The resulting scheme seems to be one of the fastest triangulation algorithms, which enables us to generate tetrahedra in R^3 with a linear generation rate of 15000 tetrahedra per second for randomly generated points on a HP 735 machine.

Key-Words

Triangulation, Delaunay Triangulation, Voronoï Tesselations, Convex Hull.

Table des matières

1 Introduction

La triangulation de Delaunay d'un ensemble S de points de R^d est telle que les boules ouvertes circonscrites aux simplexes ne contiennent aucun point de S. Cette triangulation est duale à l'ensemble des polyèdres de Voronoï [13] associés à S. Le polyèdre de Voronoï associé à un point de S est l'ensemble des points plus proches de ce point que des autres points de S. La dualité fait correspondre les k-faces des simplexes de la triangulation aux $(d-k)$-faces des polyèdres de Voronoï. Ainsi chaque simplexe de la triangulation est associé à un sommet d'un des polyèdres de Voronoï qui est en effet le centre de la sphère circonscrite à ce simplexe. Les polyèdres de Voronoï de S constituent un graphe dont les nœuds sont les sommets et les arêtes sont les arêtes des polyèdres de Voronoï. Par dualité ce graphe représente aussi la triangulation de Delaunay de S. On l'appelle graphe d'adjacence de cette triangulation; il est composé de simplexes et de couples de simplexes adjacents.

Delaunay [5] a montré que si chaque arête de ce graphe vérifie le critère de la "sphère vide", alors ce critère est vérifié pour un couple quelconque de nœuds de ce graphe. Une arête du graphe, qui représente un couple (K, K') de simplexes adjacents contenant une face commune f, vérifie le critère de la sphère vide si et seulement si la boule ouverte circonscrite à K (resp. K') ne contient pas le sommet $P_f^{K'}$ de K' (resp. P_f^K de K) non contenu dans f; une telle arête est alors appelée arête de Delaunay. Le *"Lemme général"* de Delaunay permettra de construire, à partir d'une triangulation quelconque de S, la triangulation de Delaunay de S. Ceci en considérant toutes les possibilités de bascules d'arêtes dans le graphe d'adjacence pour obtenir les arêtes de Delaunay. Dans R^2 la bascule d'arête est la bascule entre les diagonales du quadrilatère formé par le couple de triangles constituant l'arête dans le graphe [12] [9]; la détermination des arêtes de Delaunay est alors immédiate. Dans R^d où $d > 2$ et dans le cas où $|S| = d+2$, une étude détaillée a été faite par Lawson [11] sur les possibilités de bascules d'arêtes dans le graphe d'adjacence. La généralisation pour $|S|$ quelconque est délicate et exige la connaissance préalable d'un graphe d'adjacence.

Green et Sibson [8] démontrent dans le plan un théorème, basé sur le *Lemme général* de Delaunay, qui évite les bascules d'arêtes dans le graphe d'adjacence et qui permet une construction incrémentale de ce graphe. Le graphe possède alors, à chaque étape d'insertion de points, des arêtes de Delaunay. Ce résultat a été généralisé indépendamment par Bowyer [4], Watson [14] et Hermeline [10] en dimension quelconque :

La réunion des simplexes dont la boule ouverte circonscrite contient un point inséré interne à la triangulation est un polyèdre étoilé par rapport à ce point; les simplexes définis par ce point et les faces frontalières du polyèdre constituent alors la nouvelle triangulation de celui-ci; la triangulation du complémentaire de ce polyèdre reste inchangée.

La mise à jour incrémentale du graphe est donc divisée en deux parties : recherche et destruction des nœuds dont la boule ouverte circonscrite est non vide et création des nouveaux nœuds prenant en compte le point inséré ainsi que les nouvelles arêtes correspondantes.

Appelons tâche associée à un point inséré, le polyèdre formé par les simplexes à sphères non vides et boule la triangulation de la tâche prenant en compte le point inséré. La mise à jour du graphe est alors la détermination de la tâche et la construction de la boule.

Cet algorithme déjà ancien est malgré tout le plus efficace. Nous proposons une version moderne de cet algorithme, utilisant la notion de transport de structures, qui permet une construction inductive de la boule à partir de la tâche; nous dirons que la structure de la boule hérite de celle de la tâche.

Chaque nœud du graphe qui représente un simplexe est alors enrichi par le rayon et le centre de sa sphère circonscrite, $d+1$ vecteurs normales à ses faces et une matrice $(d+1)\times(d+1)$ d'entiers compris entre 0 et d. Ce nouveau graphe est appelé graphe de connexion de S [3]. En remarquant que si $s=[P_1,\cdots,P_d,X]$ est un simplexe de la boule alors il existe un simplexe t de la tâche qui est défini par $t=[P_1,\cdots,P_d,Y]$, on peut définir s à partir de t en remplaçant Y par X. Nous montrons alors que l'on peut transporter la struture de t pour définir la struture de s utilisant des opérations élémentaires bien adaptées au critère de Delaunay. Nous évitons ainsi les calculs de déterminant pour les vecteurs normales et les résolutions de systèmes linéaires pour les centres des sphères circonscrites. Nous montrons aussi comment on peut diriger la construction des simplexes de la boule pour éviter les tris pour les mises à jour des relations d'adjacence.

L'algorithme a été implémenté en dimension deux et trois et des résultats numériques montrent l'efficacité de notre méthode: pour la plupart des exemples testés la vitesse d'insertion de points est constante; en dimension deux elle est de l'ordre de 12500 points ($\sim$ 25000 triangles) par seconde et en dimension trois 2500 points ($\sim$ 15000 tétraèdres) par seconde sur un HP 735.

Dans la seconde section, on définit le graphe de connexion ainsi que la notion de transport de structures. Dans la troisième on développe une méthode classique pour l'évaluation de la tâche. La quatrième est consacrée à la construction inductive de la boule à partir de la tâche. Dans la cinquième on propose une extension au cas des points externes. Les résultats numériques font l'objet de la sixième section. Enfin nous concluons sur quelques problèmes ouverts dans la dernière section.

2 Structure de données et notion de transport

2.1 Graphe de connexion

La performance des algorithmes de triangulation dépend fortement de la définition de la structure de données de la triangulation. En particulier pour les algorithmes incrémentaux, que nous définissons: à chaque étape d'insertion une modification locale est apportée à la triangulation; pour cela on considère le graphe d'adjacence dont les nœuds sont les simplexes et les arêtes sont les couples de simplexes adjacents.

2.1.1 Définitions

Un nœud K représente un simplexe défini par $d+1$ points $P_0^K, P_1^K, \ldots, P_d^K$ de R^d. Ainsi chaque sommet est repéré par son indice dans la structure nodale. Une arête issue d'un nœud représente le couple constitué par le simplexe correspondant à ce nœud et un des simplexes adjacents dans la triangulation; On définit le simplexe voisin A_i^K d'indice i d'un simplexe K comme étant le simplexe adjacent contenant la face f_i^K de K opposée au sommet P_i^K de K; ainsi l'ordre des simplexes voisins dans la structure est celui des

sommets; donc a priori aléatoire puisque les sommets ne sont pas ordonnés dans R^d. A K on associe en plus le centre O^K et le carré du rayon r^K de sa sphère circonscrite (qui seront utiles pour valider ou non le critère de Delaunay) et à chacune de ses faces f_i^K, le vecteur unitaire n_i^K normale à l'hyperplan support de la face orientée vers P_i^K (permettant ainsi une réponse rapide à un critère de visibilité); enfin pour faciliter l'établissement des relations d'adjacence, on affine la structure par une matrice dite de connexion qui repére les indices des sommets des simplexes voisins: l'élément $c_{i,j}^K$ (si $i \neq j$) de la matrice de connexion C^K du simplexe K est l'indice de P_j^K dans le simplexe voisin A_i^K; l'élément $c_{i,i}^K$ est l'indice du sommet *opposé* à P_i^K dans A_i^K. Comme K et A_i^K ont une face commune, ils ont d sommets en commun; on a:

$$K = [P_0^K, P_1^K, \ldots, P_d^K] \quad \text{et} \quad A_i^K = [P_0^{A_i^K}, P_1^{A_i^K}, \ldots, P_d^{A_i^K}] ;$$

alors si $j \neq i$, $P_j^K = P_{c_{i,j}^K}^{A_i^K}$ et P_i^K dans K est *opposé* à $P_{c_{i,i}^K}^{A_i^K}$ dans A_i^K.

2.1.2 Exemple de matrice de connexion

Soient le triangle $K = [P_0, P_1, P_2]$ dans R^2 et ses triangles voisins A_0^K, A_1^K, A_2^K; Supposons que les indices des sommets dans K et A_i^K soient donnés par le schéma suivant, où les points A, B, C complètent les simplexes voisins A_0^K, A_1^K, A_2^K (figure 1).

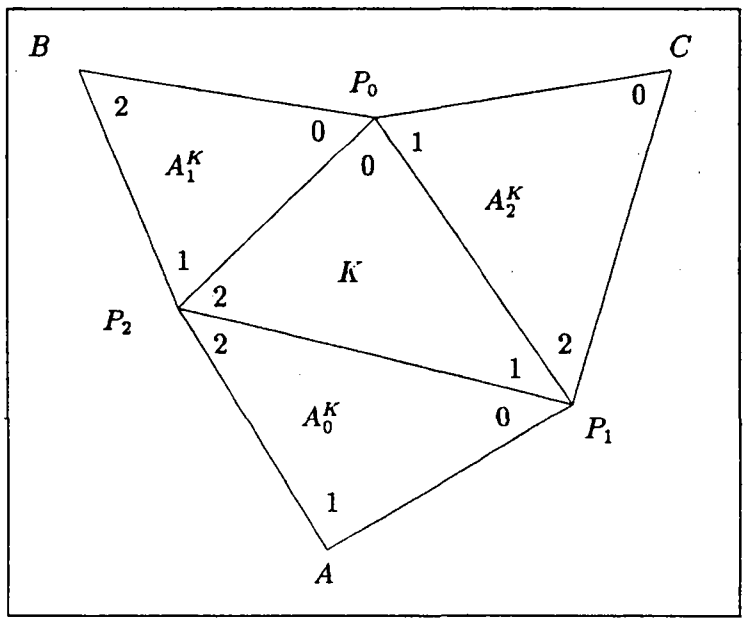

FIG. 1 - exemple de matrice de connexion

Sommet d'indice	0	1	2
K	P_0	P_1	P_2
A_0^K	P_1	A	P_2
A_1^K	P_0	P_2	B
A_2^K	C	P_0	P_1

$c^K_{0,0}$ est l'indice du sommet opposé au sommet d'indice 0 de K, P_0, dans le simplexe voisin d'indice 0, A^K_0; $c^K_{0,0} = 1$. $c^K_{0,1}$ est l'indice du sommet d'indice 1 de K, P_1, dans le simplexe voisin d'indice 0, A^K_0; $c^K_{0,1} = 0$. De même $c^K_{0,2} = 2$; On complète ainsi la matrice de connexion de K :

$$C^K = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 1 \\ 1 & 2 & 0 \end{pmatrix}$$

2.2 Transport de structure

Soit K un simplexe, X l'un de ses sommets et R le simplexe déduit de K en remplaçant X par un point Y donné. Nous allons montrer que l'on peut calculer O^R, $(r^R)^2$ et n^R_i, $0 \leq i \leq d$, à partir de O^K, $(r^K)^2$ et n^K_i, $0 \leq i \leq d$, grâce à des opérations élémentaires adaptées au critère de Delaunay et à des critères de visibilités.

2.2.1 Transport des centres

Soient f^K_X la face de K opposée au sommet X et n^K_X le vecteur unitaire normale à l'hyperplan support de f^K_X orientée vers X; on a: $K = [f^K_X, X]$ et $R = [f^K_X, Y]$, où les crochets désignent l'enveloppe convexe. Par définition, O^R appartient à la droite de vecteur directeur n^K_X passant par O^K. Ainsi il existe un réel t vérifiant :

$$(1) \qquad O^R = O^K + t * n^K_X.$$

Soit A l'un des sommets de f^K_X, on a : $||\overrightarrow{O^RY}|| = ||\overrightarrow{O^RA}||$. Si M est le milieu du segment $[A, Y]$, alors O^R appartient à l'hyperplan de vecteur normale $\overrightarrow{AY}$ passant par M, ou :

$$(2) \qquad \overrightarrow{AY} \cdot \overrightarrow{MO^R} = 0.$$

On peut déterminer t à partir des équations (1) et (2); en effet, $MO^R = MO^K + t * n^K_X$ et $\overrightarrow{AY} \cdot (\overrightarrow{MO^K} + t * n^K_X) = 0$; on a $\overrightarrow{AY} = \overrightarrow{O^KY} - \overrightarrow{O^KA}$, $\overrightarrow{O^KM} = \dfrac{\overrightarrow{O^KY} + \overrightarrow{O^KA}}{2}$ et $H_{f^K_X}(Y) = \overrightarrow{AY} \cdot n^K_X$, où $H_{f^K_X}(P)$ désigne la puissance du point P par rapport à l'hyperplan support de f^K_X. Comme $||\overrightarrow{O^KA}|| = r^K$, on en déduit :

$$t = \frac{||\overrightarrow{O^KY}||^2 - (r^K)^2}{2 * H_{f^K_X}(Y)};$$

en reportant t dans (1) on obtient O^R et on a : $(r^R)^2 = ||\overrightarrow{O^RY}||^2$.

Nous remarquons que la quantité $C_d(K, Y) = ||\overrightarrow{O^KY}||^2 - (r^K)^2$ est associée au critère de Delaunay; en particulier si $C_d(K, Y) < 0$, la boule ouverte circonscrite à K contient Y; de même la quantité $C_v(f^K_X, Y) = H_{f^K_X}(Y)$ est associée à un critère de visibilité; en particulier si $C_v(f^K_X, Y) > 0$, la face f^K_X est visible du point Y.

Cette formule permet d'éviter la résolution du système linéaire qui est défini par $||\overrightarrow{O^RP^R_0}|| = ||\overrightarrow{O^RP^R_i}||$, $0 < i \leq d$, pour le calcul de O^R.

2.2.2 Transport des normales

Soit i un indice tel que $f_i^R \neq f_X^K$. Par définition, il existe une $(d-2)$-face g de f_X^K telle que $f_i^R = [g, Y]$ (figure 2); en effet g ne contient pas le point $P_i^R = P_i^K$.

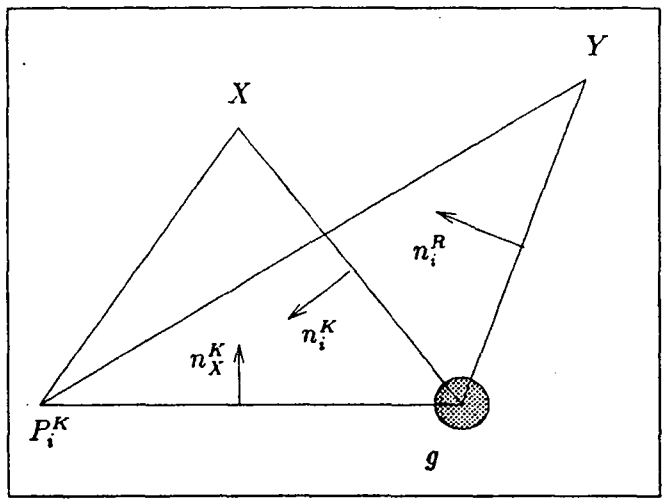

FIG. 2 - transport des normales

Comme g est de dimension $d-2$, le sous espace orthogonal à g, $g^\perp$, est de dimension 2. Les vecteurs n_X^K et n_i^K appartiennent à $g^\perp$ et sont indépendants. Ainsi $g^\perp$ est engendré par les vecteurs n_X^K et n_i^K. On a $n_i^R \in g^\perp$, donc il existe un réel λ tel que:

$$(3) \qquad n_i^R = n_i^K + \lambda * n_X^K.$$

Soit A un sommet quelconque de g; on a $n_i^R \cdot \overrightarrow{AY} = n_i^K \cdot \overrightarrow{AY} + \lambda * n_X^K \cdot \overrightarrow{AY}$. Comme $n_i^R \cdot \overrightarrow{AY} = 0$, $H_{n_i^K}(Y) = n_i^K \cdot \overrightarrow{AY}$ et $H_{n_X^K}(Y) = n_X^K \cdot \overrightarrow{AY}$, on en déduit:

$$\lambda = -\frac{H_{n_i^K}(Y)}{H_{n_X^K}(Y)}.$$

En reportant λ dans (3) on obtient un vecteur normale à l'hyperplan support de f_i^R qu'il suffit de normaliser pour obtenir n_i^R. Ainsi on peut déterminer les vecteurs unitaires normales aux hyperplans supports des faces de R, à partir des puissances du point Y par rapport aux hyperplans supports des faces de K.

3 Evaluation de la tâche

La tâche $\mathcal{T}$ est constituée des simplexes dont la boule ouverte circonscrite contient le point P à insérer. Il suffit alors de rechercher parmi tous les simplexes ceux dont la boule ouverte circonscrite est non vide; mais cette recherche est coûteuse. Par définition $\mathcal{T}$ est une partie connexe de R^d; ainsi ayant l'un de ses éléments, on peut la déterminer

par une recherche par adjacence (ou par profondeur dans le graphe). Un simplexe K est à sphère non vide si le critère de Delaunay correspondant n'est pas valide: la quantité $C_d(K,P) = ||\overrightarrow{O^K P}||^2 - (r^K)^2$ est négative. Comme les coordonnées de O^K sont en général réelles, le critère est basé sur un calcul inexact et peut conduire à des résultats non conformes comme le montre l'exemple qui suit.

3.1 Exemple d'une configuration ambiguë

Soient les triangles $[A,B,C]$, $[A,C,D]$, $[A,D,E]$, $[A,E,G]$, $A,G,F]$, $[G,E,F]$ et le point P définis par la configuration de la figure 3.

On suppose dans cet exemple que:

- les disques ouverts circonscrits de $[A,B,C]$ et $A,D,E]$ contiennent le point P et celui de $[A,C,D]$ ne contient pas P, à cause de la cosphéricité des points A, B, C, D et E.
- les disques ouverts circonscrits de $[A,E,G]$, $[A,G,F]$ et $[G,E,F]$ contiennent P car P est proche de G.

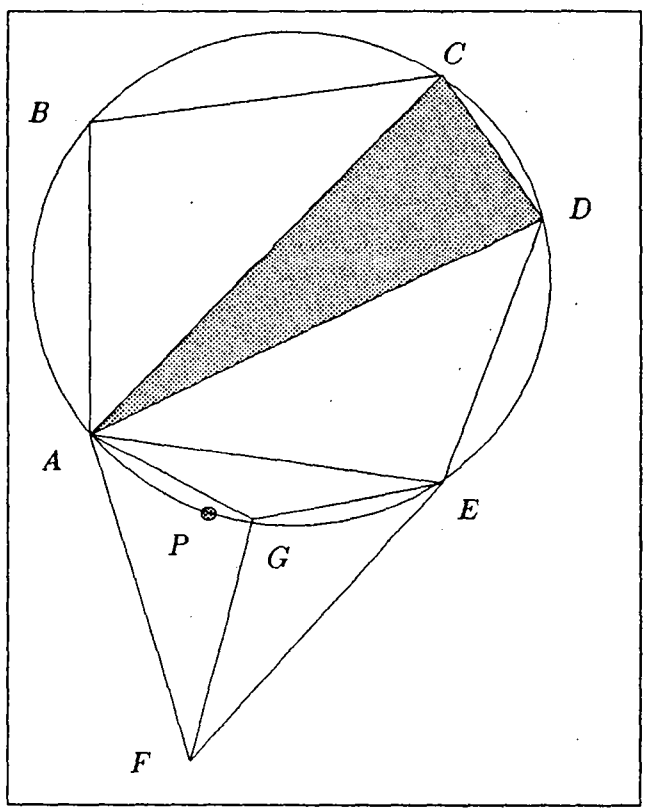

FIG. 3 - configuration ambiguë

Nous remarquons alors que :

- la tâche est constituée de tous les triangles sauf le triangle $[A, C, D]$ si une recherche exhaustive parmi tous les triangles est appliquée; la tâche n'est pas étoilée par rapport à P car l'arête $[A, C]$ de $[A, B, C]$ n'est pas visible de P et elle contient le point G dans son intérieur; mais une triangulation de la tâche prenant en compte P est possible;

- la tâche est réduite au triangle $[A, B, C]$ si la recherche des éléments de la tâche est effectuée par adjacence à partir de $[A, B, C]$; dans ce cas la tâche ne contient pas P et aucune triangulation de la tâche prenant en compte P n'est possible.

A ce stade, pour résoudre ces problèmes, on fera une hypothèse que l'on validera à la fin de la section.

Hypothèse. *Si f est une face quelconque d'un simplexe K, alors l'évaluation de la quantité $C_v(f, P) = H_f(P)$ (puissance de P par rapport à l'hyperplan support de f) est basée sur un calcul exact ou l'on peut répondre d'une manière robuste à la requête $C_v(f, P) > 0$.*

Cette hypothèse nous permet une construction de simplexes à volume positif; en effet c'est une condition nécéssaire pour garantir une triangulation correcte. On montre par la suite qu'elle est suffisante.

Nous définissons la base $\mathcal{B}$ comme étant l'ensemble des simplexes qui contiennent P. Par définition $\mathcal{B}$ est : - une partie connexe de R^d, - étoilé par rapport à P. La base $\mathcal{B}$ est inclue dans la tâche $\mathcal{T}$; désignons son complémentaire dans $\mathcal{T}$ par $\mathcal{D}$. Par la suite, nous proposons un algorithme robuste pour l'évaluation de la tâche qui n'est qu'une version légérement modifiée de celui de George dans [6], garantissant l'étoilement de la tâche par rapport à P; il est divisé en deux parties : - initialisation, - correction.

3.2 Initialisation de la tâche

A partir d'un élément de $\mathcal{B}$, par une recherche par adjacence on peut déterminer $\mathcal{B}$; à ce niveau on utilise des critères de visibilités, donc basés sur un calcul exact (un simplexe K contient P si toutes ses faces sont visibles du point P ou $\forall f \in K$, $C_v(f, P) > 0$). La base $\mathcal{B}$ vérifie ainsi l'étoilement par rapport à P et ne contient, dans son intérieur, aucun point d'autre que P.

De même dans une recherche par adjacence, à partir des éléments de $\mathcal{B}$, on détermine $\mathcal{D}$; ainsi on obtient $\mathcal{T} = \mathcal{B} \cup \mathcal{D}$. A ce niveau on utilise le critère de Delaunay, donc un calcul inexact. On en déduit que la tâche peut éventuellement ne pas être étoilée par rapport à P et contenir des points dans son intérieur. Pour cela on apporte une correction à la tâche.

3.3 Correction de la tâche

La correction consiste à enlever des éléments de $\mathcal{D}$ pour retrouver les propriétés voulues de $\mathcal{T}$. On obtient l'algorithme suivant :

- s'il existe un point interne à $\mathcal{T}$, repérer l'un des simplexes de $\mathcal{D}$ le possédant comme sommet et retirer ce simplexe de $\mathcal{D}$.
- s'il existe une face frontalière de $\mathcal{T}$ non visible de P, repérer le simplexe de $\mathcal{D}$ qui la contient et retirer ce dernier de $\mathcal{D}$.
- itérer si le nombre d'éléments de $\mathcal{D}$ est modifié.

On constate que dans le cas pire, la tâche $\mathcal{T}$ s'identifie à la base $\mathcal{B}$ si $\mathcal{D}$ est réduit à l'ensemble vide; on en déduit la convergence de l'algorithme.

Les tests de visibilités assurent la connexité de la tâche; comme $\mathcal{B}$ est toujours inclue dans la tâche, cette dernière contient le point P. Ainsi la méthode proposée garantit une construction correcte de la triangulation de la tâche prenant en compte P, à savoir: chaque simplexe de cette triangulation est défini par P et une face frontalière de la tâche. Ceci a été notre premier souci; pour cela on est amené à violer par moment le critère de Delaunay. Donc la triangulation résultante n'est pas 100% Delaunay; ce qui n'est pas gênant pour le type d'application envisagé (maillage éléments finis).

Dans cet algorithme, on a supposé que l'un des éléments de la base est connu; par la suite on développe un méthode classique pour la recherche d'un tel élément.

3.4 Recherche d'un élément de la base

On suppose que les seules informations accessibles à ce niveau sont les simplexes constituant le graphe de connexion et le point P. On considère un simplexe quelconque K_0 de la triangulation (en particulier le dernier construit) et on se propose de parcourir la triangulation, par adjacence à partir de K_0, pour aboutir à un élément de la base (figure 4).

Nous allons définir deux parcours possibles.

3.4.1 Parcours suivant un critère d'intersection

Soit G le barycentre de K_0. Deux éléments consécutifs dans ce parcours sont définis par une intersection non vide de leur face commune avec le segment $[G, P]$. Par définition ce parcours est bien défini et le dernier élément de ce parcours est un élément de la base. Dans ce cas les requêtes sont des tests d'intersection d'une face et d'un segment. Chaque test est équivalent à d critères de visibilité: Une face quelconque f intersecte $[G, P]$ si toutes les faces du simplexe virtuel $[f, P]$ exceptée f sont visibles de G.

3.4.2 Parcours suivant un critère de visibilité

Deux éléments consécutifs dans ce parcours ont en commun une face dont l'hyperplan support sépare le premier élément du point P. Dans ce cas le parcours n'est pas unique,

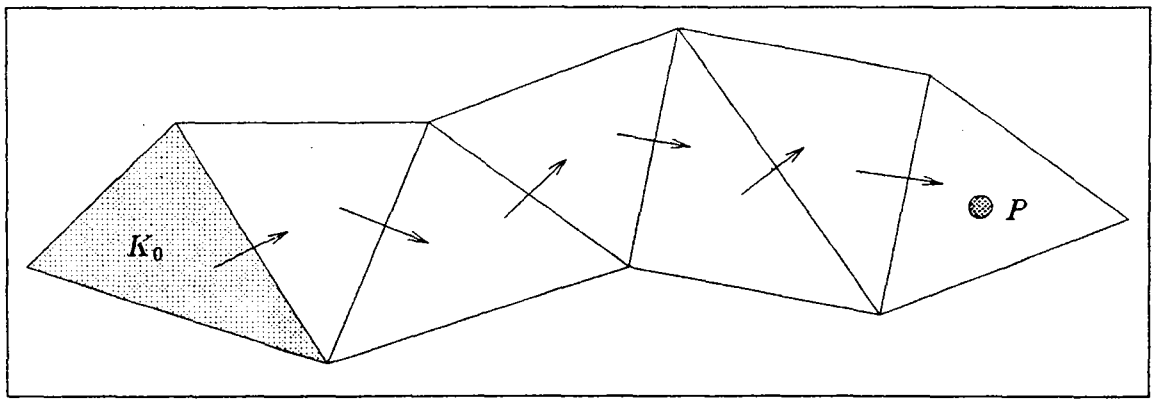

FIG. 4 - parcours par adjacence

il n'est pas bien défini et peut être cyclique sans jamais atteindre un élément de la base comme le montre l'exemple suivant (figure 5).

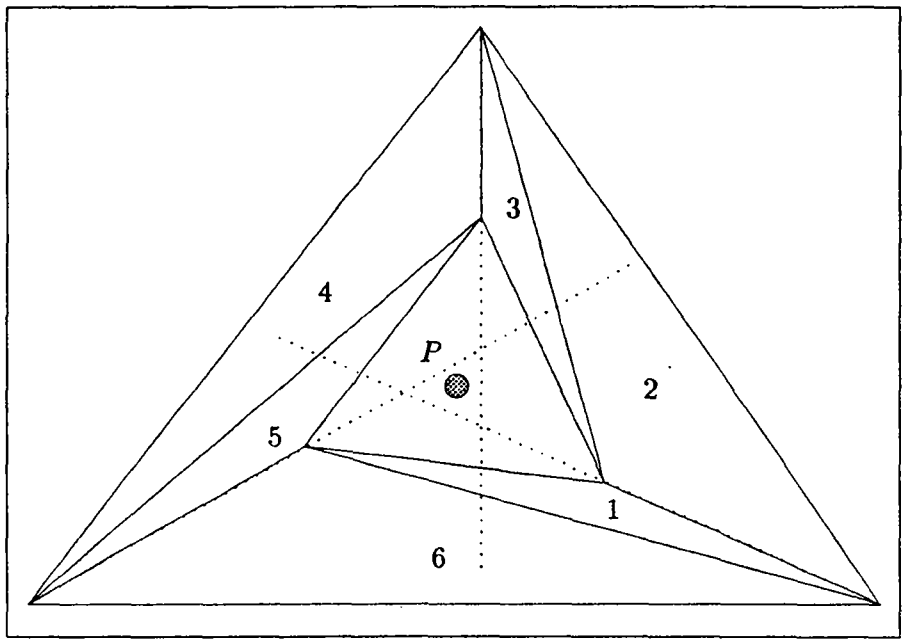

FIG. 5 - configuration cyclique

A partir du triangle 1, les triangles 2, 3, 4, 5, 6 sont visités pour aboutir de nouveau au triangle 1. De même la situation est bloquée si on décide de ne pas parcourir les triangles déjà visités.

Pour éviter les cycles dans ce parcours, on propose de considérer d'une manière aléatoire l'une des faces du simplexe courant, dont l'hyperplan support sépare le simplexe du point P, si un choix est possible. Comme il existe toujours un tel parcours, l'algorithme

converge. Cette méthode, basée sur des critères de visibilité, est moins coûteuse que celle précédemment introduite et s'avère assez efficace.

L'efficacité des deux méthodes repose sur le choix du simplexe initial du parcours, K_0. En particulier en choisissant K_0 assez proche de P, on réduit la longueur du chemin reliant K_0 à P, donc le nombre de simplexes dans le parcours; il suffit alors que l'un des sommets de K_0 soit assez voisin de P. Ainsi pour chaque point, il faut définir ses points voisins. Pour cela, nous introduisons une grille de voisinage.

3.4.3 Grille de voisinage

La grille est définie par un ensemble de parallèlépipèdes uniformes, de côtés parallèles aux axes de coordonnées, que l'on appelera cases. La longueur de chaque côté est déterminée par la densité du nuage des points à trianguler. Dans le cas où la variation de la densité des points n'est pas très importante, on peut imposer le nombre maximal de points par case.

A chaque case on associe les points déjà insérés dans la triangulation et à chaque point déjà inséré, l'un des simplexes de la triangulation possédant ce dernier comme sommet. Une case c est identifiée par un vecteur $(c_{i_1}, \cdots, c_{i_d})$ d'entiers. L'ensemble $E_k(c)$ des cases voisins de degré k de c est défini par

$$E_k(c) = \{c' \text{ case}; \quad \exists \alpha, 1 \leq \alpha \leq d; \quad |c'_{i_\alpha} - c_{i_\alpha}| \leq k\}$$

Pour rechercher un point Q voisin de P, il suffit alors de trouver la case contenant P, puis de considérer l'un des points contenu dans cette case si elle est non vide et sinon, exploiter les cases voisines dans l'ordre croissant du degré. Le simplexe K_0 est alors celui associé à Q.

Notons que l'utilisation de la grille n'est efficace que si le nombre de points est assez grand; dans ce cas la recherche via la grille sera declanchée qu'à partir d'un seuil donné qui représente le nombre de points déjà insérés.

Il existe d'autres méthodes pour la recherche d'un élément de la tâche, qui considérent des structures plus complexes que la grille. Par exemple, en conservant la hiérarchie de la construction du graphe d'adjacence, Boissonnat [2] montre que cette recherche s'effectue en un temps optimal si les points sont insérés dans un ordre aléatoire.

3.5 Une validation possible de l'hypothèse

Pour justifier l'hypothèse, on propose un calcul exact du critère de visibilité qui n'est efficace qu'en petites dimensions, en particulier si $d = 2$ ou 3. Les formules de transports des normales, qui utilisent une division de deux quantités, ne sont plus valables.

3.5.1 Cas de R^2

Une face est un segment de droite $[A, B]$ du plan; si x_A et y_A (resp. x_B et y_B) sont les coordonnées de A (resp. B) alors un vecteur normale à la droite support de $[A, B]$ est défini par $n = (y_A - y_B, x_A - x_B)$ et l'équation de cette droite par

$$H(x, y) = (y_A - y_B) * x + (x_B - x_A) * y + (y_B - y_A) * x_A + (x_A - x_B) * y_A = 0.$$

En supposant que les coordonnées de A et de B sont des entiers compris entre 0 et une valeur N donnée, on peut calculer $H(x,y)$ d'une manière exacte si le calcul de la quantité N^2 est exact. On peut supposer que la quantité N^2 ne dépasse pas la valeur 2^m, pour un m fixé; on en déduit $N < 2^{m/2}$. On considère alors une normalisation des coordonnées entre 0 et N, puis une conversion en entiers. La valeur de m est fixé par les moyens de calcul.

3.5.2 Cas de R^3

Dans ce cas, une face représente un triangle $[A,B,C]$ dans l'espace; Un vecteur normale au plan support de $[A,B,C]$ est défini par $n = \overrightarrow{AB} \times \overrightarrow{AC}$ et l'équation de ce plan par

$$H(X) = \overrightarrow{AX} \cdot n,$$

où X est un point de R^3.

De même, en supposant que toutes les coordonnées sont des entiers compris entre 0 et une valeur N donnée, on peut calculer $H(X)$ d'une manière exacte si le calcul de la quantité N^3 est exact. En considérant un majorant 2^m pour cette quantité, on obtient $N < 2^{m/3}$.

On trouve une formule analogue en dimension d qui nous indique la borne supérieure de N, à savoir $N < 2^{m/d}$. En effet le calcul de $H(X) = \text{Volume}([f,X])/d!$ se ramène à celui de la puissance d-ième d'une distance. Dans la pratique m est de l'ordre de 51 (type réel double précision); ainsi le procédé n'est plus efficace pour $d > 4$.

La méthode proposée n'est pas optimale, mais sa mise en œuvre s'avère assez simple. Avnaim et al. [1] ont proposé un autre procédé qui limite la valeur de N à 2^m dans R^2 et à 2^{m-1} dans R^3.

4 Construction de la boule

Chaque simplexe R de la boule est défini par le point P et une face frontalière f de la tâche. La face f appartient à un simplexe K de la tâche. On dit alors que les simplexes K et R sont homologues. Comme R est K partagent la face f, la structure nodale de R est complètement définie à partir de celle de K par transport de structure. Ainsi en parcourant toutes les faces frontalières de la tâche, on peut construire les simplexes de la boule et mettre à jour leurs structures nodales par transport de structure de leurs homologues. Il reste à définir pour chaque simplexe de la boule, $d+1$ relations d'adjacence, soient 1 externe et d internes.

Les relations d'adjacence externes sont définies par les couples de simplexes dont le premier appartient à la boule et le second n'appartient pas à la boule. Si (R,L) est un tel couple, alors L est adjacent à l'homologue de R et la relation d'adjacence entre R et L peut s'établir automatiquement par transport de structure.

Les relations d'adjacence internes sont définies par les couples de simplexes appartenant à la boule. Pour établir ces relations il suffit d'identifier les couples de simplexes possédant $d-1$ sommets identiques, tous différents de P. Pour cela on peut appliquer des

différentes méthodes de tri ou de hachage. Ces recherches étant coûteuses, on propose de procéder autrement.

Soient R et R' deux simplexes adjacents de la boule. il existent par définition, deux faces f et f' frontalières à la tâche telles que $R = [f, P]$ et $R' = [f', P]$. Comme R et R' sont adjacents, f et f' partagent une $(d-2)$-face ($d-1$ sommets) et elles sont donc adjacentes. On peut alors, dans la construction de la boule, au lieu de parcourir les faces frontalières de la tâche dans un ordre aléatoire, les parcourir par adjacence et ainsi établir automatiquement les relations d'adjacence internes. Par la suite on développe une méthode qui permet pour une face frontalière de la tâche, d'identifier ses faces adjacentes frontalières à la tâche et ainsi pour un simplexe de la boule établir les relations d'adjacence internes.

4.1 Parcours conservant une $(d-2)$-face

Soient $R = [f, P]$ et $R' = [f', P]$ deux simplexes adjacents de la boule, où f et f' sont deux faces adjacentes frontalières de la tâche possédant la $(d-2)$-face commun g et soient K et K' appartenant à la tâche homologues de R et de R'. Pour établir la relation d'adjacence entre R et R', il suffit d'identifier K' à partir de K. On peut ordonner d'une manière cyclique les simplexes de la tâche contenant g tels que K soit le premier et K', le dernier suivant cette ordre; en effet :

Soient la suite des simplexes (K_j) de la tâche, où $K_0 = K$, et la suite des indices (m_j) et (m'_j) où $P^{K_0}_{m_0}$ est le sommet de f non contenu dans g et $P^{K_0}_{m'_0}$, le sommet de K non contenu dans f, vérifiant les relations de récurrence suivantes (figure 6) :

$$K_{j+1} = A^{K_j}_{m_j}, \quad m_{j+1} = c^{K_j}_{m_j,m'_j} \text{ et } \quad m'_{j+1} = c^{K_j}_{m_j,m_j};$$

alors $K' = K_l$ où l est l'indice qui vérifie $A^{K_l}_{m_l} = \emptyset$ ou $A^{K_l}_{m_l} \notin \mathcal{T}$.

On peut supposer que les indices des sommets de f dans R (resp. R') soient identiques à ceux de f dans K (resp. K') (figure 7); on en déduit :

$$A^R_{m_0} = R' \quad \text{et} \quad A^{R'}_{m'_l} = R.$$

De même on pose $c^{K_0}_{m_0,m_0} = m'_0$, $c^{K_0}_{m_0,m'_0} = m_0$ et $c^{K_0}_{m_0,k} = k$ pour tout k, indice d'un sommet de g; et on considère les suites n^j_k definies par les relations de récurrence :

$$n^j_0 = j \quad \text{et} \quad n^j_{k+1} = c^{K_k}_{m_k,n^j_k}.$$

Alors on a :

$$c^R_{m_0,j} = \begin{cases} m'_l & si\ j = m_0 \\ m_l & si\ j = m'_0 \\ n^j_{l-1} & sinon \end{cases}$$

et donc $c^{R'}_{m'_l,c^R_{m_0,j}} = j$ pour $0 \leq j \leq d$.

Ces relations traduisent entre autre le transport de la structure des indices de K_0 à travers les simplexes $(K_j)_{1 \leq j \leq l}$. Pour clarifier le procédé on propose un exemple dans R^2. Considérons la configuration suivante (figure 8).

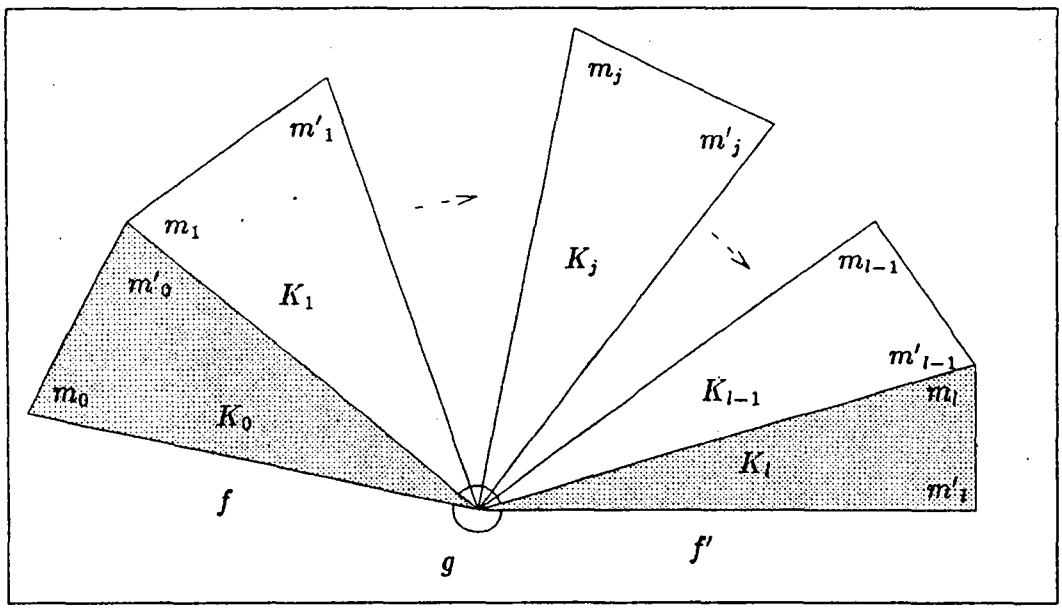

FIG. 6 - recherche de K' à partir de K

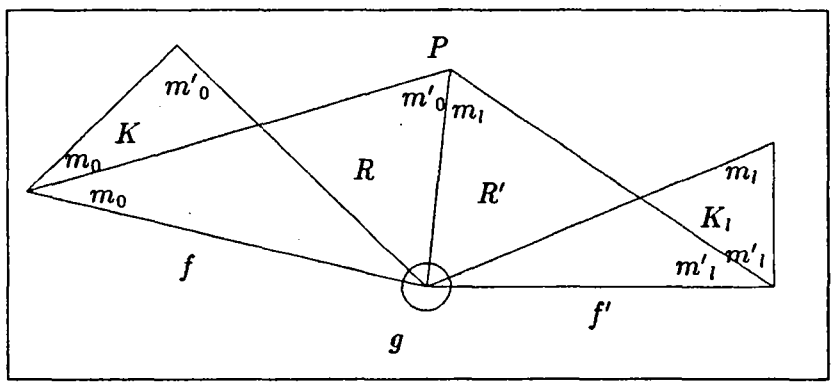

FIG. 7 - Transport des indices

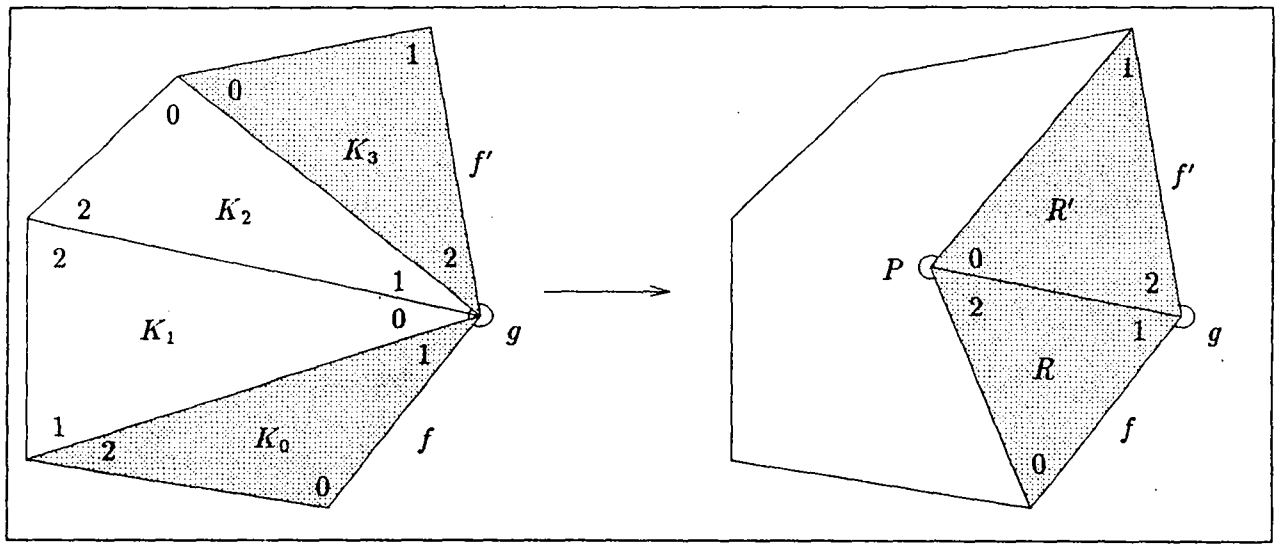

FIG. 8 - transport de structures

Dans ce cas g est le sommet d'indice 1 de K_0; R hérite de la structure des indices des sommets de K_0, c.a.d. les indices des sommets de f dans K_0 et R sont identiques (ici 0 et 1); l'indice du sommet P de R est alors égal à 2. De même R' hérite de la structure des indices des sommets de K_3.

Nous montrons que R' est le voisin d'indice 0 de R et nous déterminons les éléments $c^R_{0,0}$, $c^R_{0,1}$ et $c^R_{0,2}$ de la matrice de connexion C^R de R. Pour cela on transporte la structure des indices de K_0 à travers les triangles $(K_j)_{1\leq j\leq 3}$.

On a le schéma suivant :

$$\begin{array}{ccccccc}
\langle K_0,0,2\rangle & \rightarrow & \langle K_1,1,2\rangle & \rightarrow & \langle K_2,2,0\rangle & \rightarrow & \langle K_3,0,1\rangle \\
\begin{pmatrix} 2 & 0 & 1 \\ * & * & * \\ 0 & 1 & 2 \end{pmatrix} & & \begin{pmatrix} * & * & * \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 2 \\ * & * & * \\ 0 & 2 & 1 \end{pmatrix} & & \begin{pmatrix} 0 & 1 & 2 \\ 0 & 2 & 1 \\ * & * & * \end{pmatrix} \\
m_0 = 0 & & m_1 = 1 & & m_2 = 2 & & m_3 = 0 \\
m'_0 = 2 & & m'_1 = 2 & & m'_2 = 0 & & m'_3 = 1 \\
m_1 = c^{(0)}_{0,2} = 1 & & m_2 = c^{(1)}_{1,2} = 2 & & m_3 = c^{(2)}_{2,0} = 0 & & \\
m'_1 = c^{(0)}_{0,0} = 2 & & m'_2 = c^{(1)}_{1,1} = 0 & & m'_3 = c^{(2)}_{2,2} = 1 & & \\
c^{(0)}_{m_0,1} = c^{(0)}_{0,1} = 0 & & c^{(1)}_{m_1,0} = c^{(1)}_{1,0} = 1 & & c^{(2)}_{m_2,1} = c^{(2)}_{2,1} = 2 & &
\end{array}$$

où $c^{(k)}_{i,j}$ est lélément i,j de la matrice de connexion de K_k.

Comme K_0 est homologue à R et K_3 à R', on obtient donc l'adjacence de R et de R' et on a :

$$\begin{aligned}
c^R_{0,0} &= m'_3 = 1 \\
c^R_{0,1} &= c^{(2)}_{m_2,c^{(1)}_{m_1,c^{(0)}_{m_0,1}}} = c^{(2)}_{2,1} = 2 \\
c^R_{0,2} &= m_3 = 0 .
\end{aligned}$$

4.2 Algorithme de construction

On peut appliquer une procédure récursive pour construire les simplexes de la boule et mettre à jour les relations d'adjacence. Soit un couple (K, f) où K est un simplexe de la tâche et f une face frontalière de la tâche contenue dans K. On construit alors, par transport de structure, l'homologue R de K et on établit ses relations d'adjacence externe. Au triplet (K, f, R), on fait subir les opérations suivantes :

- à partir de (K, f) on identifie les couples (K', f') où f' est une face frontalière de la tâche adjacente à f et K' le simplexe de la tâche qui contient f'.
- pour chaque couple on vérifie si l'homologue correspondant R' est déjà construit :
  - si c'est le cas, on établit les relations d'adjacence internes entre R et R',
  - sinon on construit R' par transport de structure, on établit ses relations d'adjacence externe ainsi que les relations d'adjacence internes entre R et R' et on applique récurssivement le procédé au triplet (K', f', R').

5 Extension aux points externes

Dans le cas d'un point P externe à la triangulation, un simplexe de la base $\mathcal{B}$ est défini par P et une face frontalière de la triangulation dont l'hyperplan support sépare P du polyèdre convexe résultant de la triangulation. Ainsi ces simplexes devront être construits dans l'initialisation de la tâche. Pour cela il suffit : – d'identifier l'une des faces frontalière de la triangulation visible de P, en généralisant la procédure de recherche décrite dans 3.2. au cas des points externes, – d'appliquer les techniques de 4.1. pour rechercher les autres faces frontalières de la triangulation, visibles de P. Dans la correction de la tâche, les faces frontalières à la fois à la tâche et à la triangulation seront considérées comme visibles du point P, mais n'interviennent pas dans la construction de la boule.

6 Résultats numériques

6.1 Maquette d'expérimentation

Les algorithmes ont été testés sur des données diverses en dimension deux et trois sur un HP 735. Les codes ont été écrits en langage C. Le choix de ce langage a permis une manipulation aisée des structures de données (ici les graphes) et la programmation des procédures récursives. Pour chaque exemple testé, c'est l'enveloppe convexe qui à été triangulée. Pour analyser les résultats, nous avons jugés utile de définir quelques diagrammes. Ainsi chaque exemple est accompagné de ces diagrammes. Dans certain cas on montre aussi le maillage résultant ou l'enveloppe convexe. Enfin pour conclure on évalue les courbes de complexité en temps cpu qui indique le temps de la construction du maillage, en secondes en fonction du nombre de points insérés.

Pour chacun des cas R^2 et R^3 on considère en plus deux exemples dont le premier est constitué des points coplanaires et cosphériques et le second, d'un nombre important de

points aléatoires. ces deux exemples nous paraissent significatifs dans la mesure où il est plus délicat, pour cette méthode de traiter des nuages de points présentant des propriétés de coplanarité et de cosphéricité que des nuages quelconques.

6.1.1 Quelques diagrammes utiles

Le premier diagramme, D1, représente la taille des tâche; il indique le nombre de points ayant eu à l'insertion une tâche de taille donnée. Pour minimiser en moyenne cette taille, l'insertion des points est faite de manière aléatoire.

Le deuxième diagramme, D2, représente la taille des chemins; il indique le nombre de points ayant eu à l'insertion un chemin de taille donnée. Dans ce cas, pour minimiser en moyenne cette taille, la grille de voisinage a été utilisée.

Le troisième diagramme, D3, montre l'encombrement des cases de la grille de voisinage. le nombre maximal de points par case est alors défini en fonction de la densité des points et le nombre de cases imposées.

Enfin le dernier diagramme, D4, montre le nombre de cases visitées lors de l'analyse d'un point. Ce nombre dépend d'une part du nombre de cases de la grille et d'autre part de l'ordre d'insertion des points; ainsi il dépend des choix ci-dessus.

6.1.2 Exemple dans R^2

On considère 3 exemples dans R^2 avec dans les deux premiers, une grille régulière de taille (100×100) et dans le troisième, une grille de taille (200×200). Le premier (hypn2d) est constitué de 17132 points pris d'une façon aléatoire sur des droites et des cercles cocentriques ou non, construits eux-mêmes d'une façon aléatoire dans un rectangle; le deuxième (rotor8) comprend 570 points d'un contour donné et le troisième, 250000 points aléatoires dans un carré.

Le maillage de hypn2d (figure 9) montre une complexité (figure 10) pratiquement linéaire malgré les propriétés de dégénérescence et de cocyclicité des points. Sur la figure 11, les diagrammes indiquent une moyenne de 4 triangles dans les tâches (D1), un cheminement de longueur 4 en moyenne (D2), l'encombrement de la grille (D3) et une moyenne de 4 cases visitées pour l'analyse d'un point: D1, D2 et D4 montrent le faible nombre d'opérations à effectuer en moyenne.

Pour rotor8, on trouve les mêmes caractéristiques pour les diagrammes D1 et D2; cependant la moyenne des cases visitées est plus élevée puisque la grille n'est pas adaptée aux nombre de points du nuage. Malgré tout la complexité reste pratiquement linéaire.

Le dernier exemple montre des résultats analogues sur un cas de taille important.

Le tableau suivant montre pour chaque nuage le nombre de points, le nombre de triangles de la triangulation du nuage, le temps en secondes pour la construction du maillage et la vitesse exprimée en triangles par seconde.

nuage	nb. points	nb. triangles	temps en sec.	t./s.
Hypn2d	17132	34258	1.15	29680
Rotor8	570	1134	0.04	27000
Random2d	250000	499994	20.8	24040

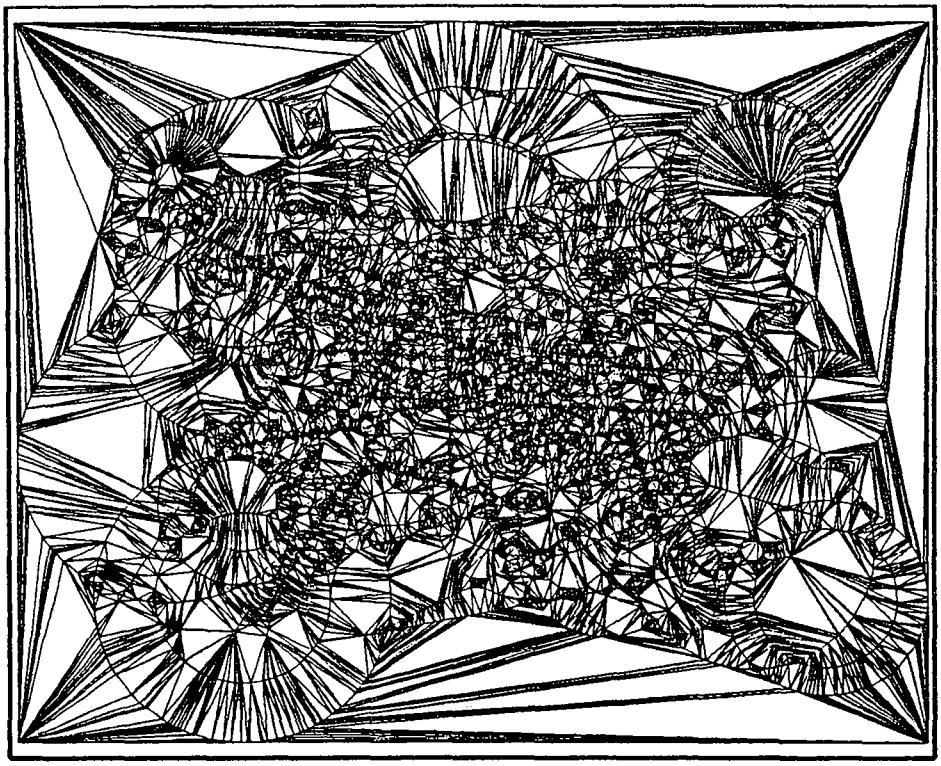

FIG. 9 - *maillage Hypn2d*

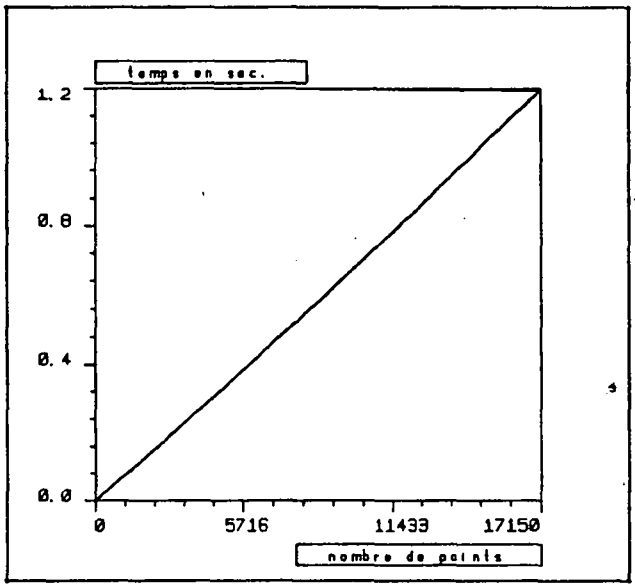

FIG. 10 - *complexité Hypn2d*

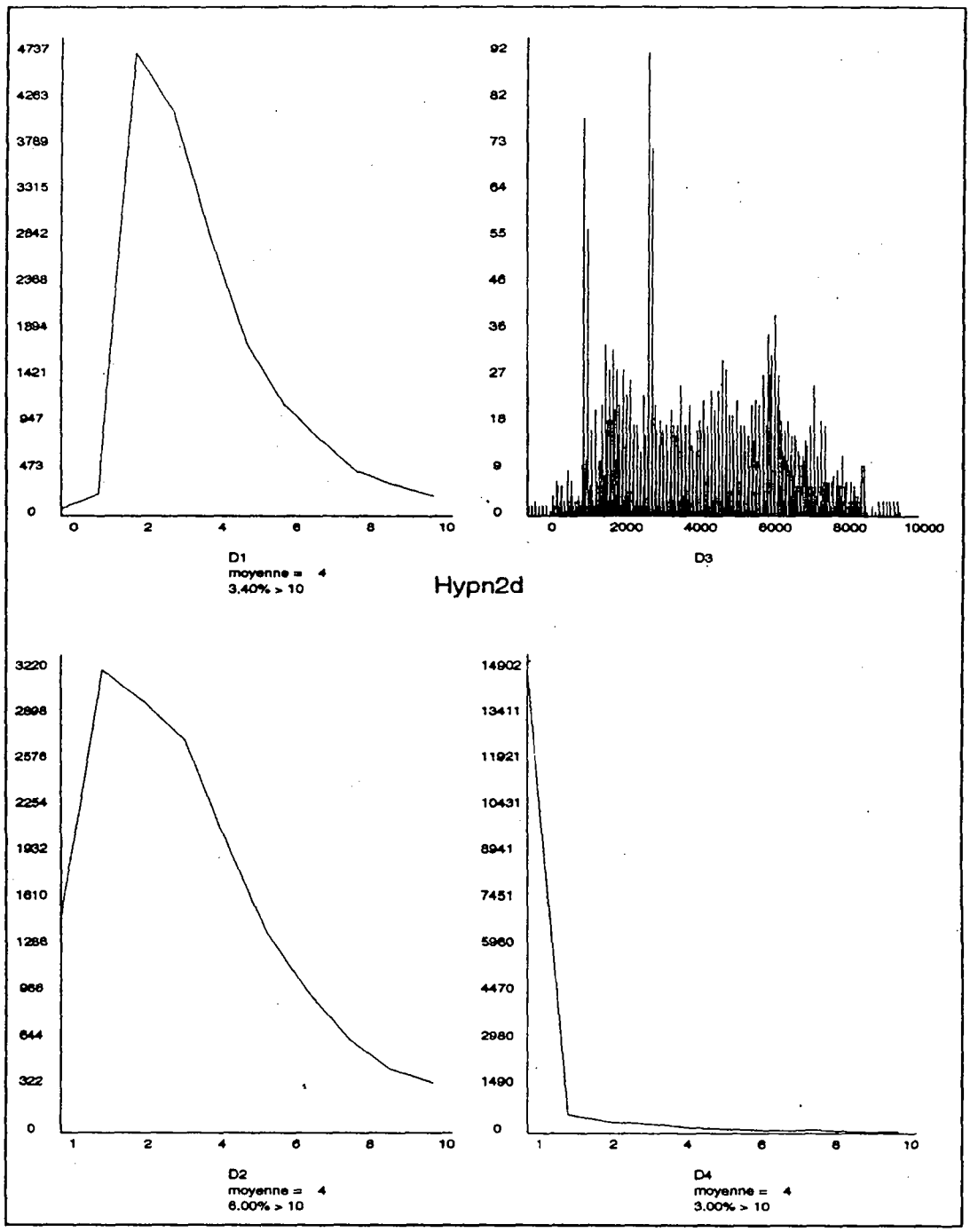

FIG. 11 - *diagrammes Hypn2d*

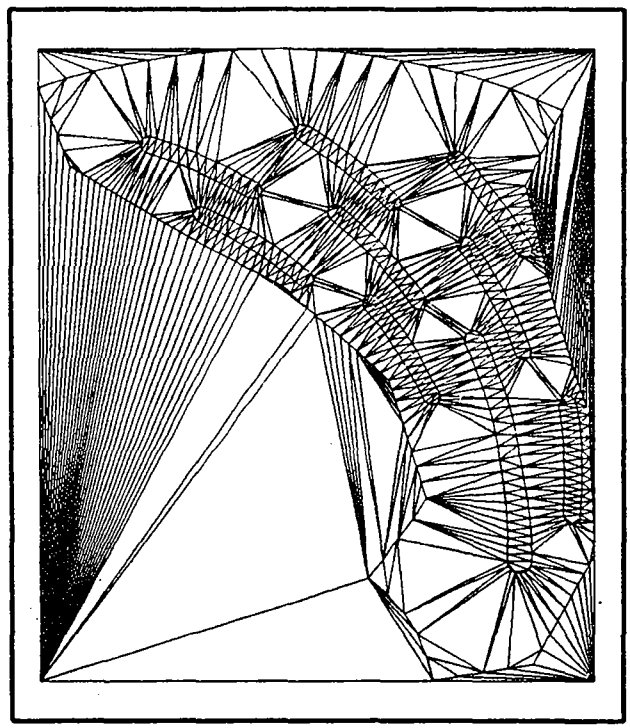

FIG. 12 - *maillage Rotor8 (Telma-Simulog)*

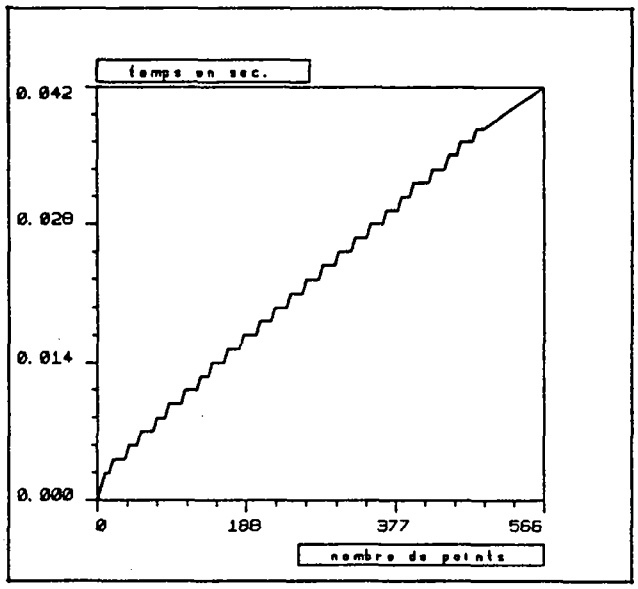

FIG. 13 - *complexité Rotor8*

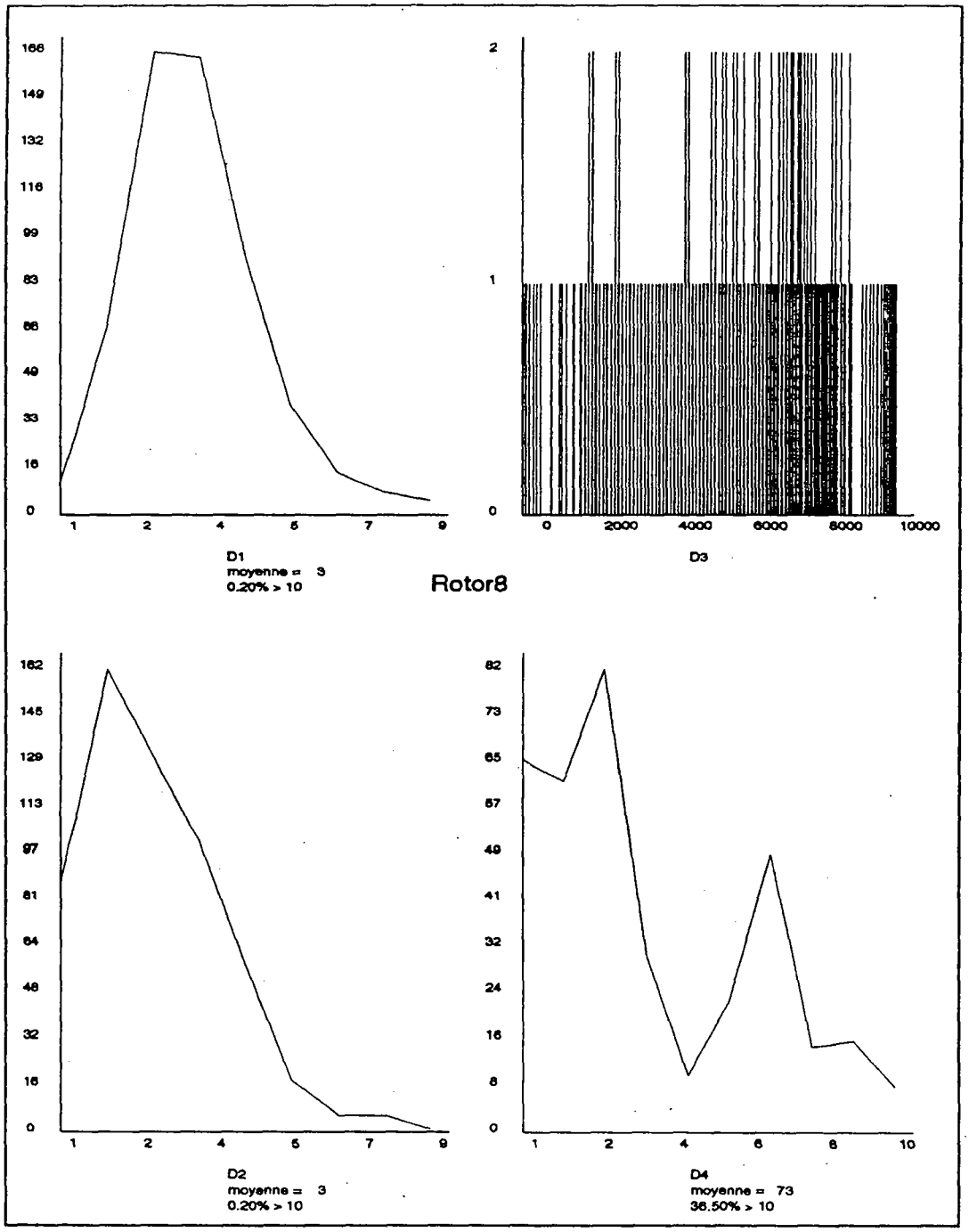

FIG. 14 - *diagrammes Rotor8*

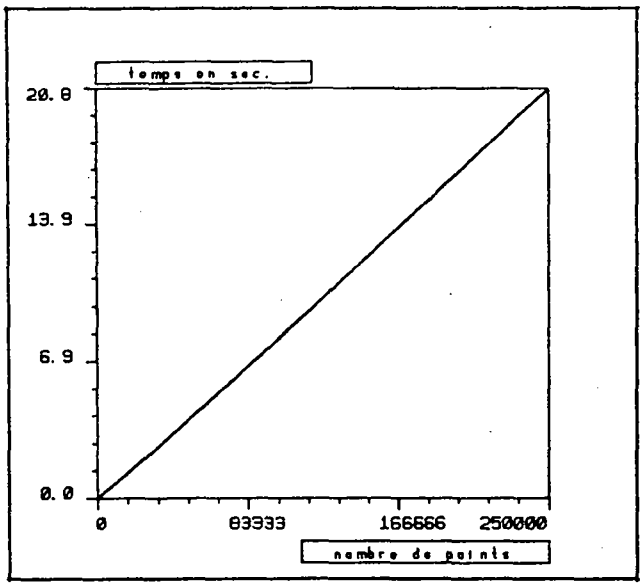

FIG. 15 - *complexité Random2d*

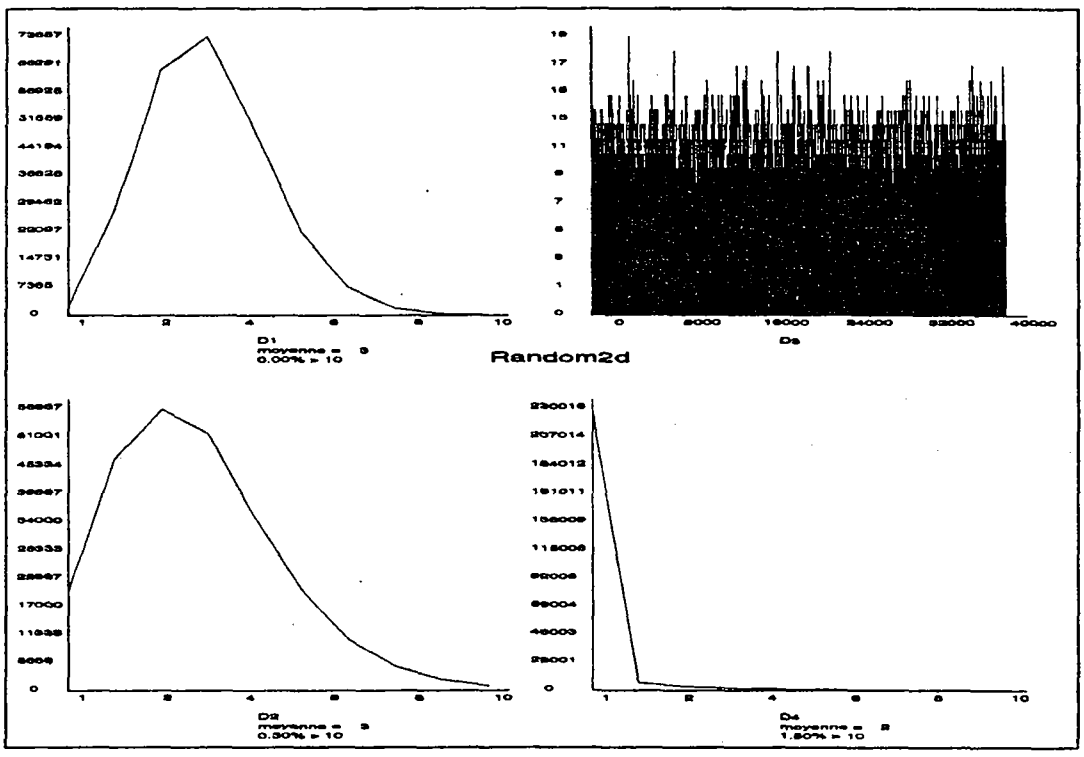

FIG. 16 - *diagrammes Random2d*

6.1.3 Exemple dans R^3

On considère 6 exemples dans R^3 et pour tous ces exemples une grille régulière de taille $(40 \times 40 \times 40)$. Le premier est constitué de 5582 points pris d'une façon alátoire sur des plans et des sphères cocentriques ou non, construits eux-mêmes d'une façon aléatoire dans l'espace. Pour les exemples 2 à 5 les points sont pris sur la surface des objets définis via un système de C.A.O. Enfin le sixième exemple représente 160000 points aléatoires dans un cube.

On a le tableau suivant :

nuage	nb. points	nb. tétraèdres	temps en sec.	t./s.
Hypn3d	5582	33635	2.24	15015
Hugo	2026	12762	0.78	16360
Peugeot	4557	27124	1.67	16270
Biellette	1559	9584	0.6	15710
Ourag	16609	100077	6.7	14937
Random3d	160000	1077045	68	15840

Ce tableau indique un effort (exprimé en nombre de simplexes construits) de l'ordre de 1.7 fois plus grand entre le maillage 2d et le maillage 3d.

La vitesse semble pratiquement constante d'un exemple à l'autre, quelques soient, en particulier, les propriétés de dégénérescence et de cocyclicité présentes dans le cas : la complexité apparaît linéaire.

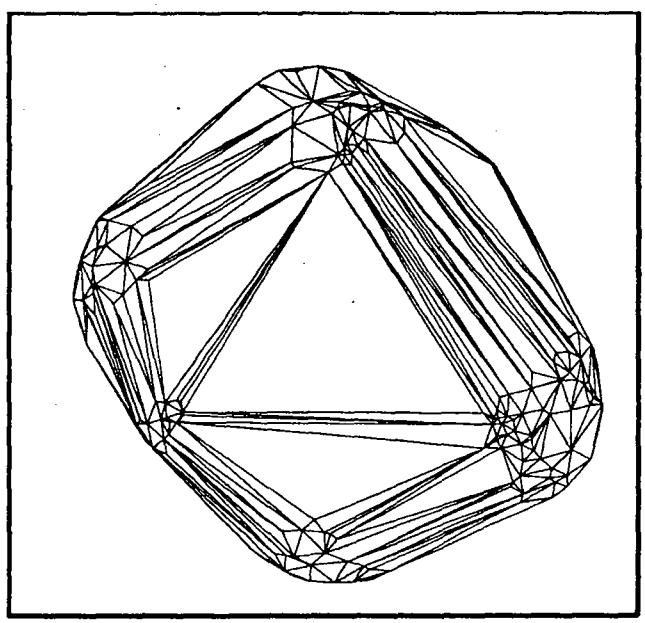

FIG. 17 - *enveloppe convexe, Hypn3d*

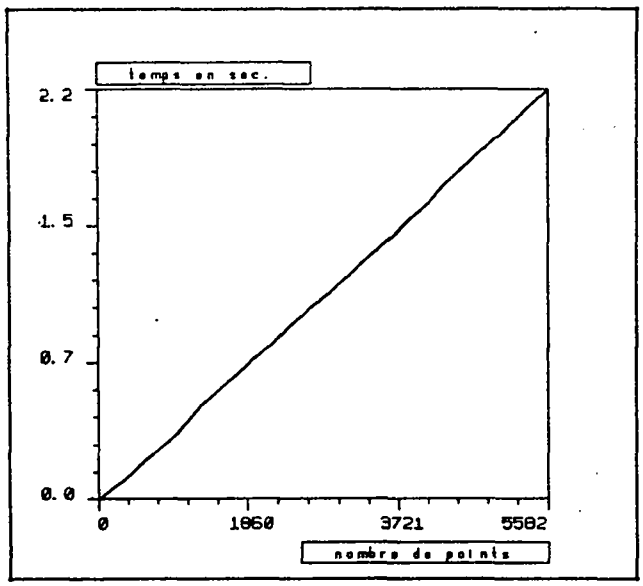

FIG. 18 - *complexité Hypn3d*

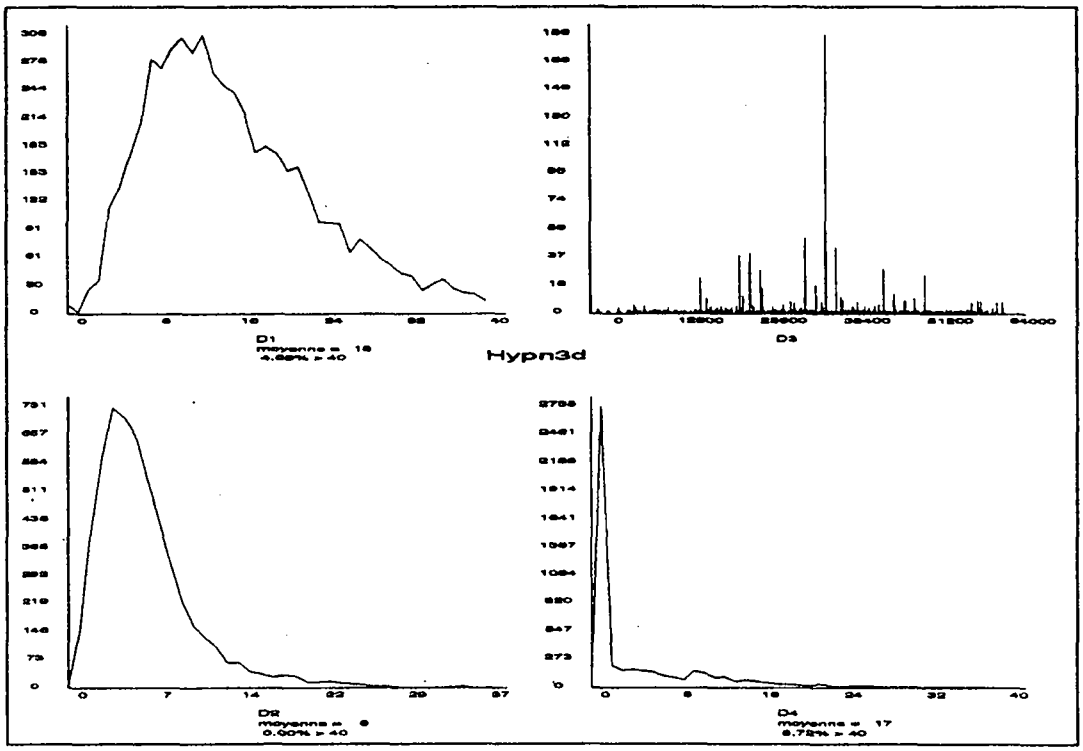

FIG. 19 - *diagrammes Hypn3d*

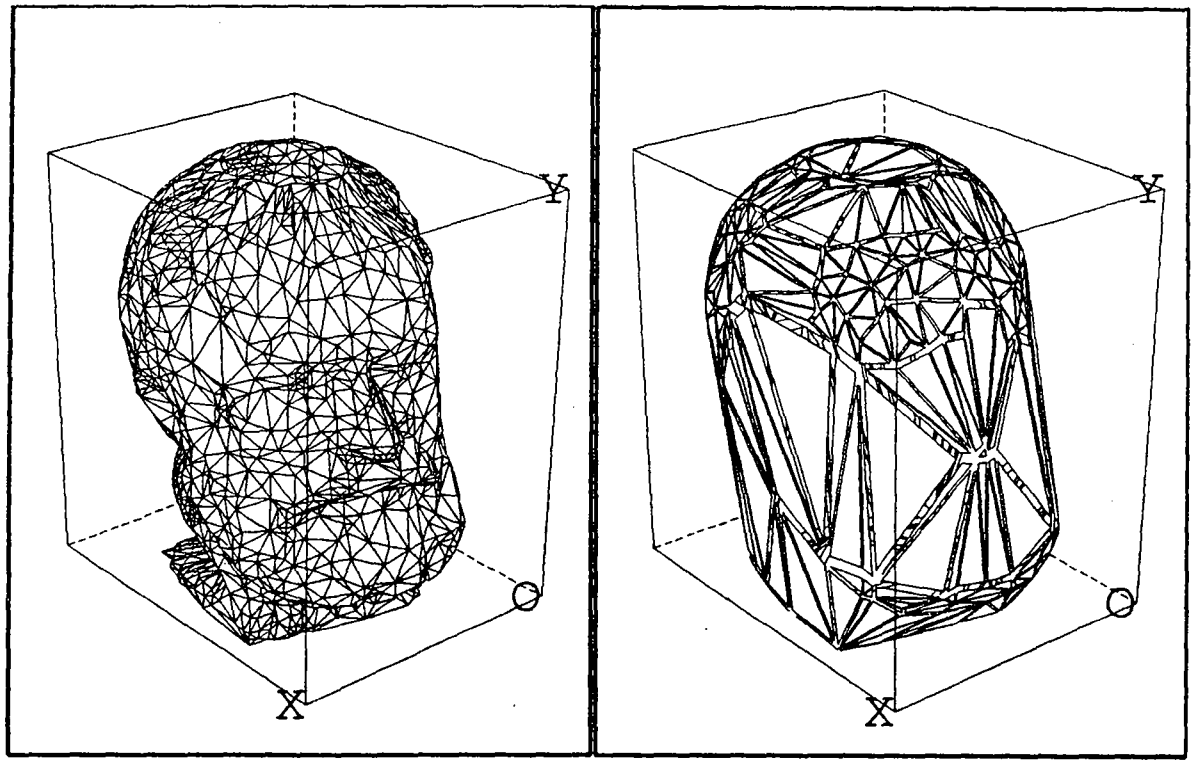

FIG. 20 - *objet + enveloppe convexe, Hugo*

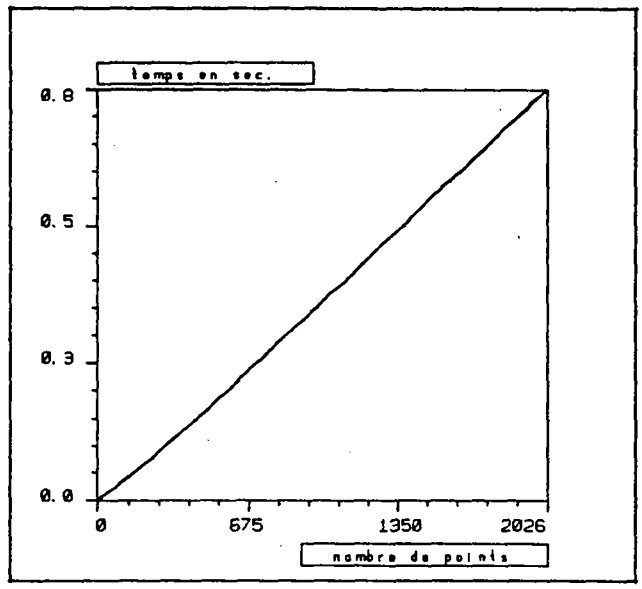

FIG. 21 - *complexité Hugo*

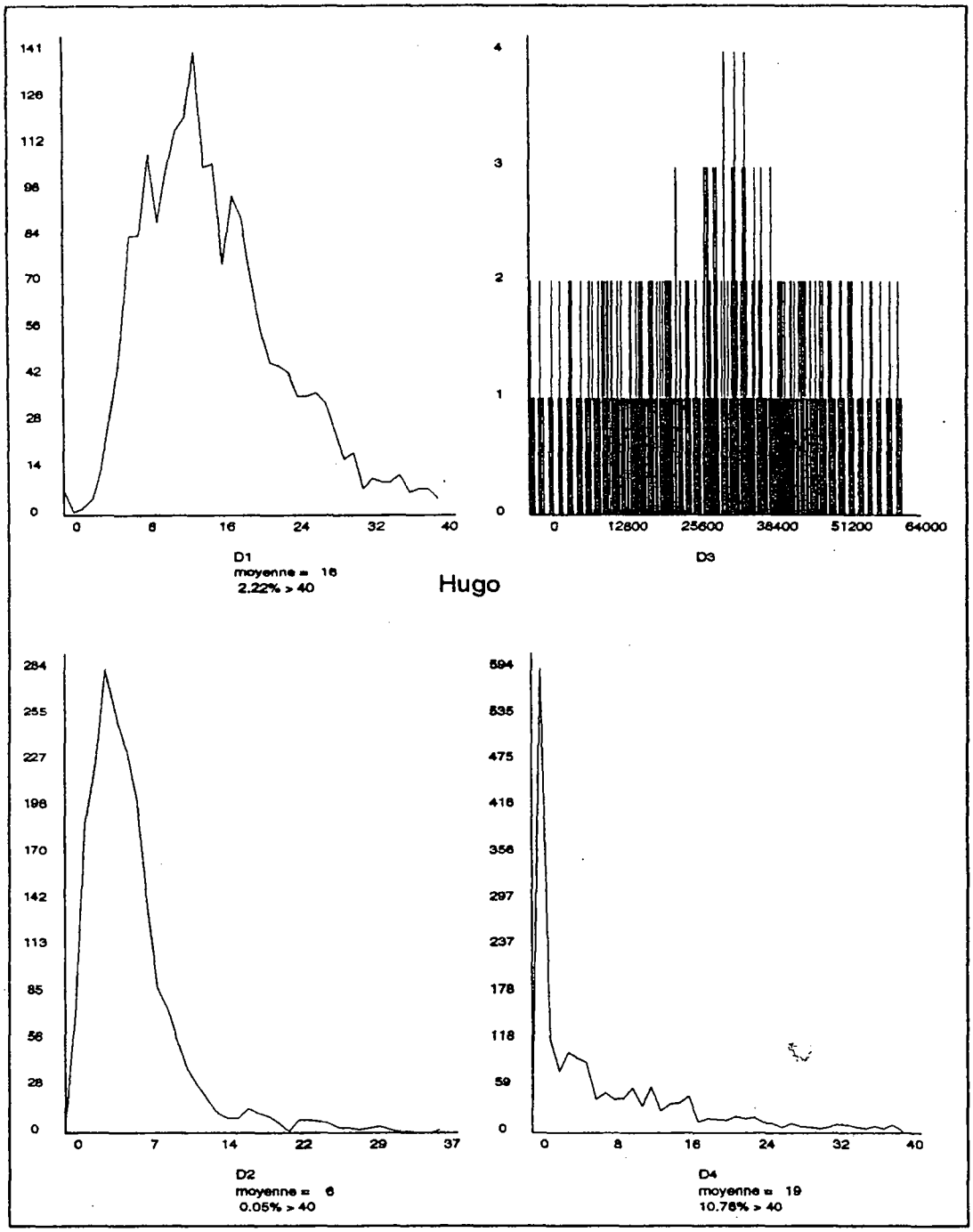

FIG. 22 - *diagrammes Hugo (ENST)*

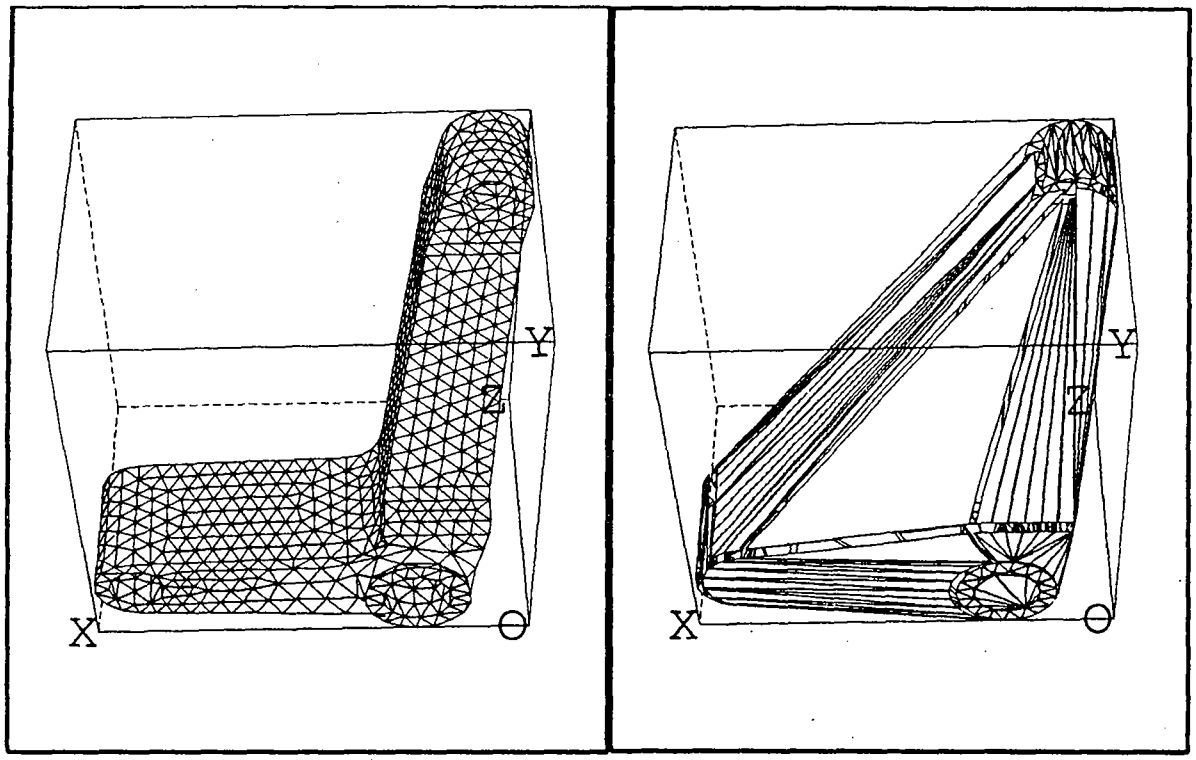

FIG. 26 - *objet + enveloppe convexe, Biellette (SDRC)*

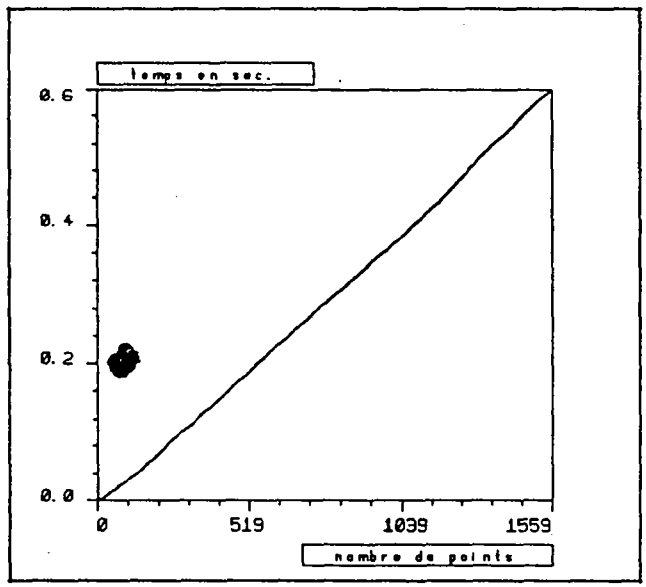

FIG. 27 - *complexité Biellette*

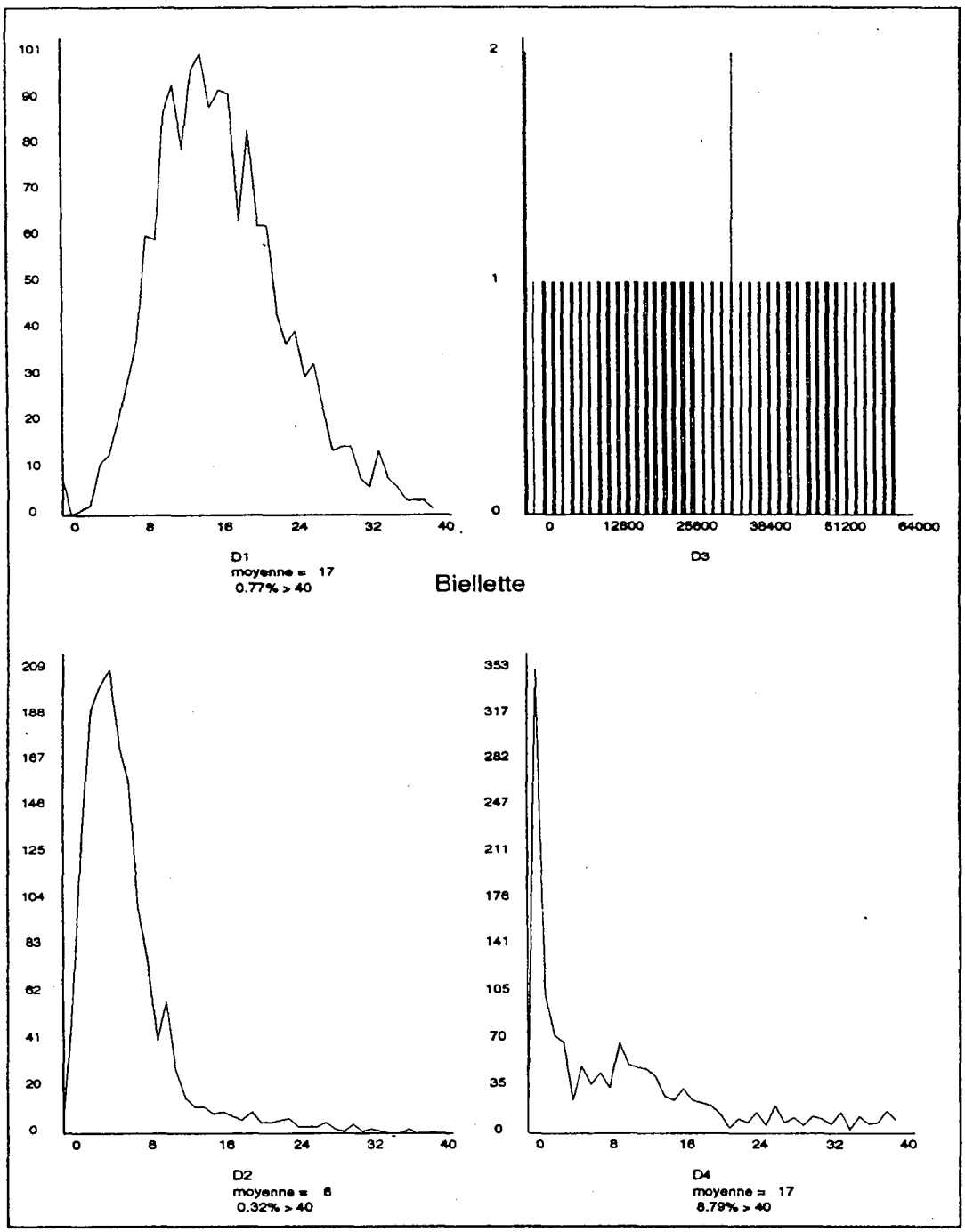

FIG. 28 - *diagrammes Biellette*

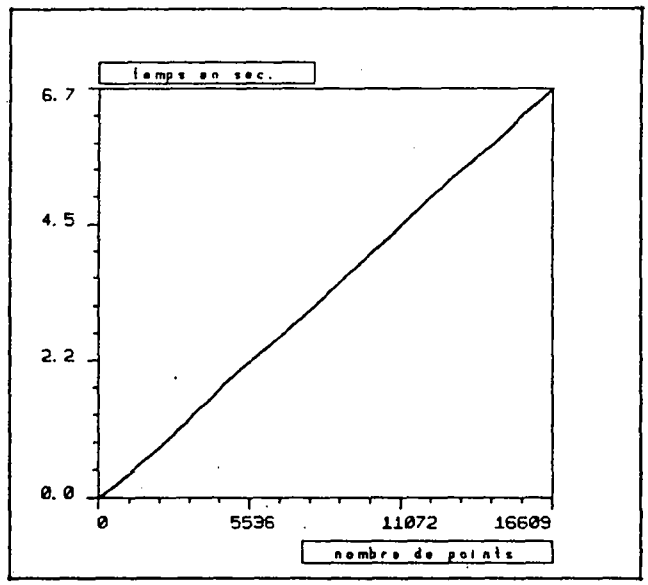

FIG. 29 - *complexité Ourag*

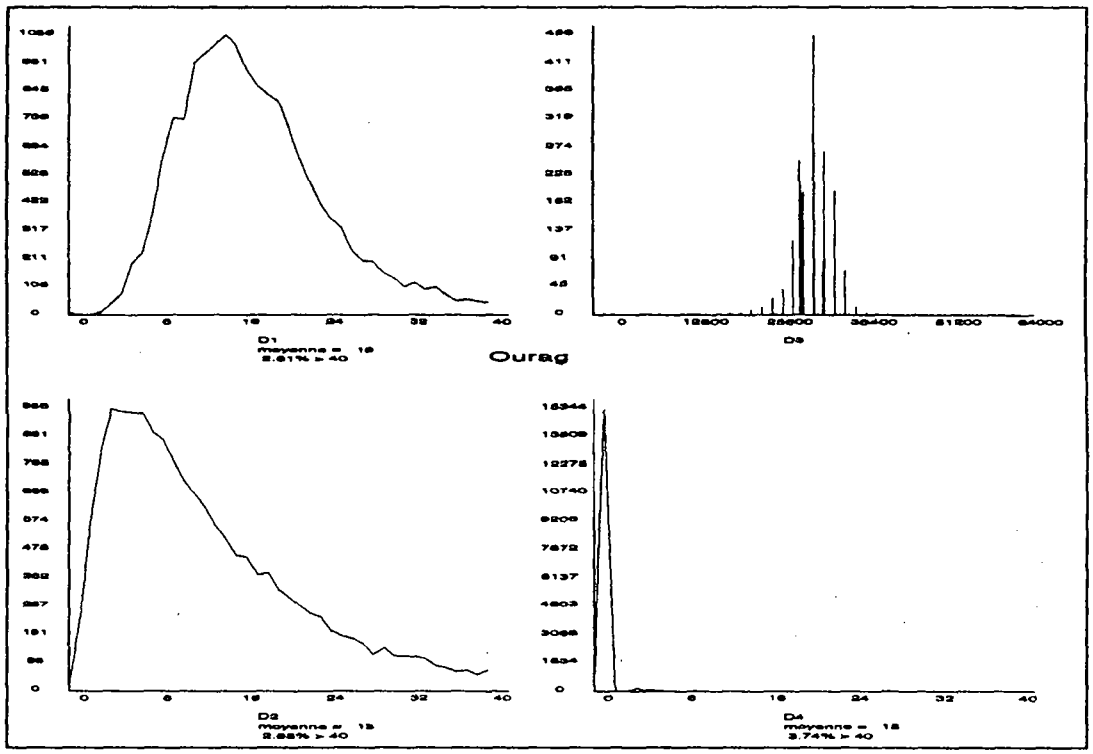

FIG. 30 - *diagrammes Ourag*

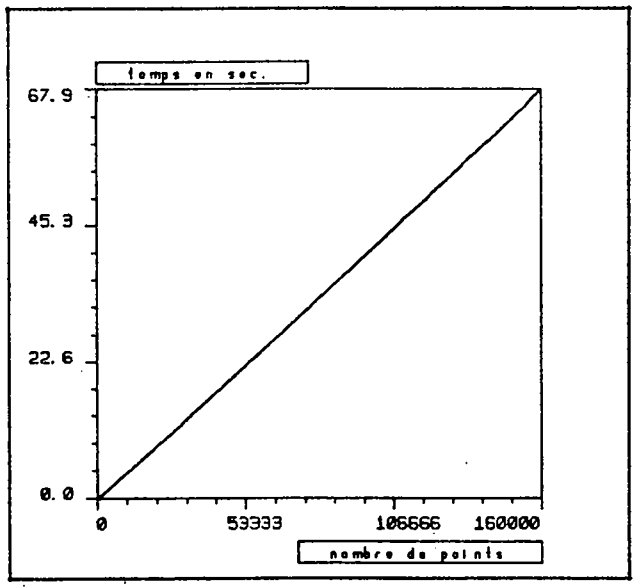

FIG. 31 - *complexité Random3d*

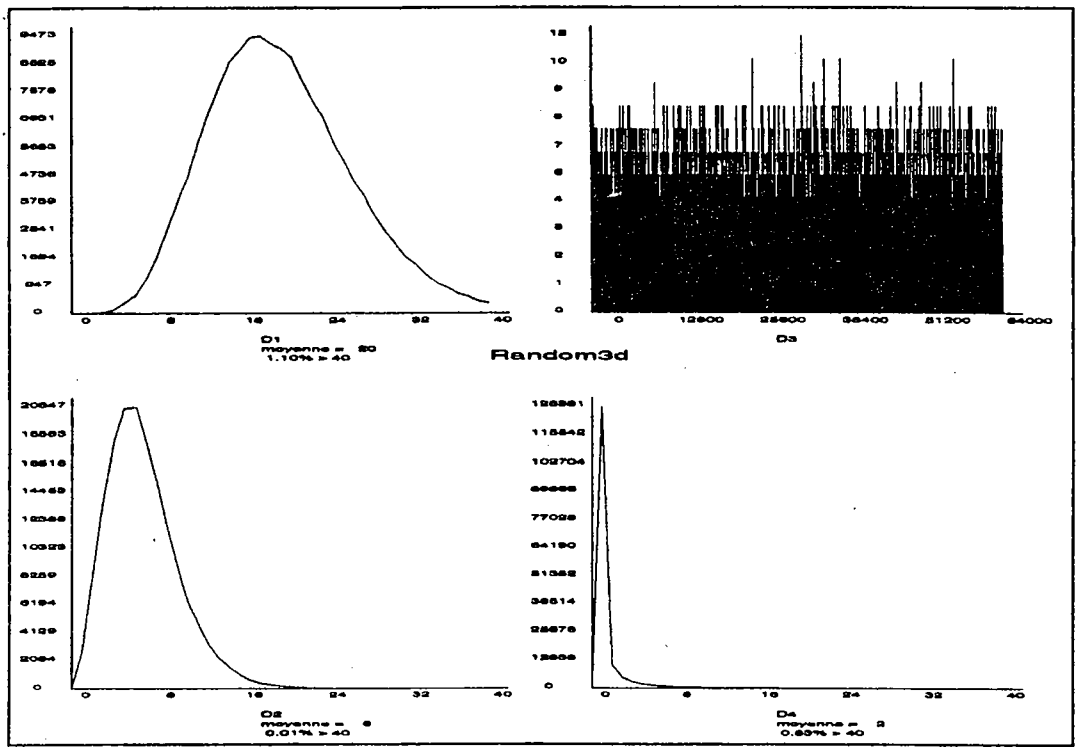

FIG. 32 - *diagrammes Random3d*

6.2 Application éléments finis

Le développement d'un mailleur automatique pour les applications de type éléments finis utilisant la méthode de Delaunay comprend différentes phases parmi lesquelles intervient de manière importante l'algorithme d'insertion de points décrit dans ce papier.

Il est nécessaire qu'un tel algorithme soit :

- robuste et fiable,
- rapide et
- optimal (au sens de la qualité du maillage à obtenir).

Nous avons appliqué les idées ci-dessus dans un logiciel 2D et dans le logiciel GHS3D [7] afin d'en améliorer les performances.

A titre d'illustration des résultats obtenus, nous donnons trois exemples qui semblent significatifs. Le premier exemple (figure 33) correspond à une simulation technique sur un frein, le deuxième plutôt orienté mécanique de la structure, conduit à trianguler un domaine (figure 26) de volume très "petit" et produit un maillage de petite taille (en termes de nombres de tétraèdres). Le dernier exemple (figure 34), de type mécanique des fluides, impose de mailler un domaine de "grand" volume et produit un maillage de taille importante.

objet	nb. points	nb. simplexes	temps en sec.	t./s.
Rotor8	2921	5682	0.6	9470
Biellette	1843	6152	2.35	2600
Ourag	257749	1510519	439	3400

La comparaison des vitesses doit tenir compte du fait que l'on mesure ici le temps nécessaire à la création d'un maillage éléments finis et non pas à celle du maillage de l'enveloppe convexe du domaine. En effet dans le cas réel, la contribution au temps de la phase insertion de points est comprise entre 20% et 80% selon la taille de l'objet. Par ailleurs, l'algorithme utilisé est une version sous contrainte de l'algorithme basique.

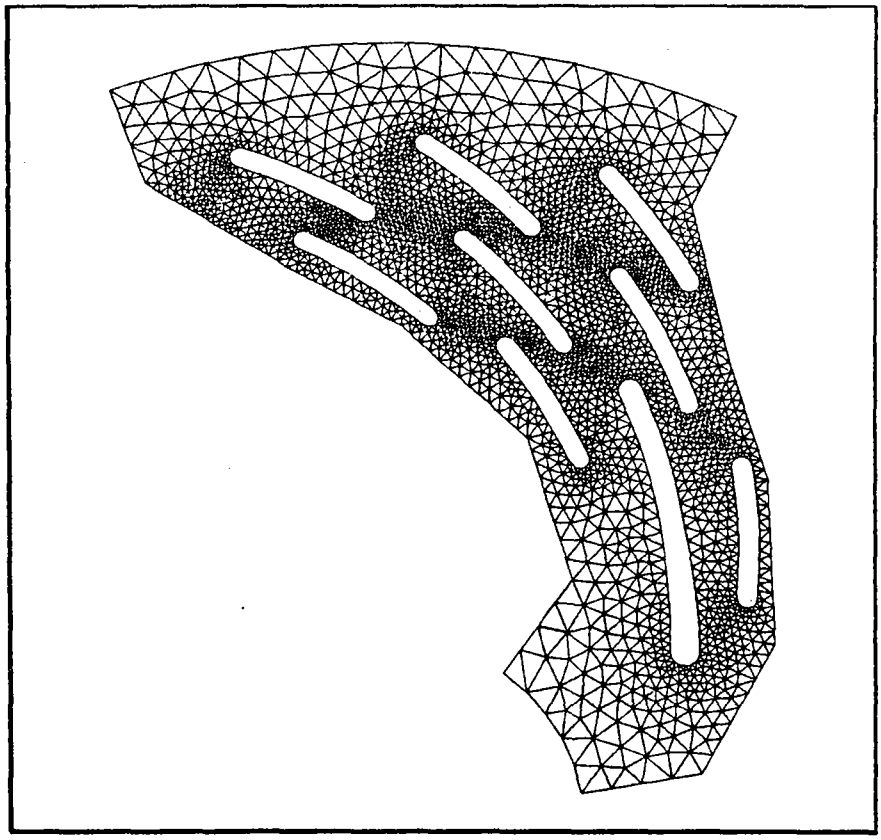

FIG. 33 - *Rotor8 maillé*

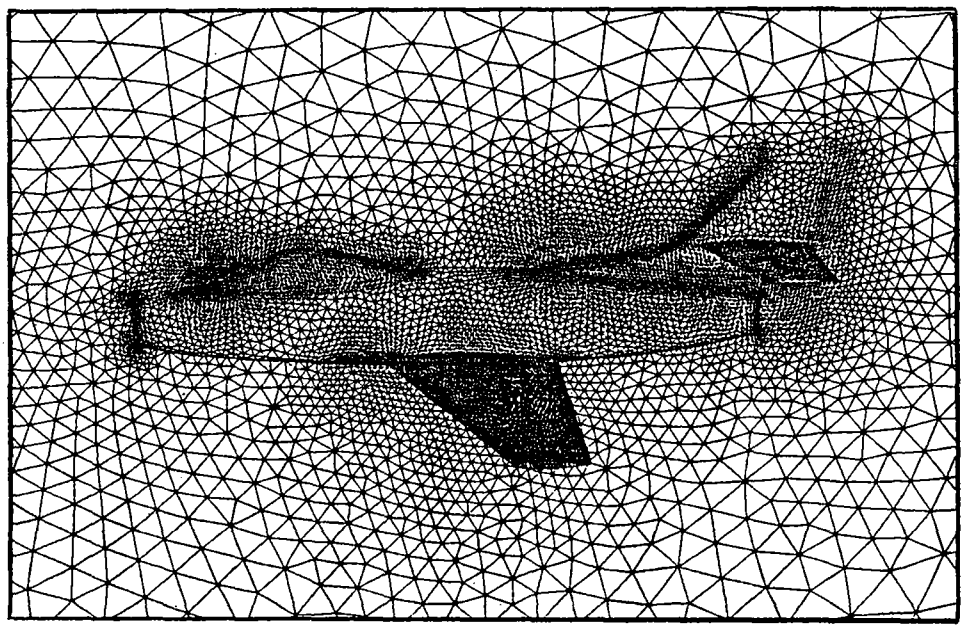

FIG. 34 - *Ourag (Dassault Aviation)*

7 Conclusion

Des nouvelles procédures pour la construction incrémentale de la triangulation d'un nuage de points dans R^d ont été développées; en bref:

- l'utilisation d'une grille de voisinage pour repérer un des points voisins du point inséré,
- le cheminement aléatoire pour la recherche d'un élément de la tâche,
- la redéfinition de la tâche,
- le calcul des quantités géométriques liées aux nouveaux simplexes à partir de celles des anciens, évitant ainsi les calculs de déterminants d'ordre d et la résolution des systèmes linéaires $d \times d$ et
- la construction de la boule en parcourant par adjacence les faces frontalières de la tâche.

L'implantation de l'algorithme en dimension deux et trois nous a permis de vérifier l'efficacité de la méthode; en particulier la vitesse d'insertion de points semble être pratiquement constante pour un nombre raisonnable de points.

Une recherche future consiste à

- adapter aux mieux la grille aux données pour minimiser le nombre de simplexes parcourus dans la recherche d'un élément de la base,
- trouver un ordre d'insertion de points qui minimise en moyenne le nombre de simplexes de la tâche et (ou) de la boule.
- valider l'hypothèse dans R^d, $d > 4$ et trouver une solution qui permet l'utilisation des formules de transport des normales.

Références

[1] F.Avnaim, J-D.Boissonnat, O Devillers, F.P.Preparata and M. Yvinec; "Evaluating signs of determinants using single-precision arithmetic", *INRIA Rapport de recherche*, 2306, (1994).

[2] J-D.Boissonnat and M.Teillaud; "A hirearchical representation of objects: The Delaunay Tree", *Second ACM Symp. on Comput. Geom. in Yorktown Heights*, (1986).

[3] H.Borouchaki, F.Norgue et F.Schmitt; "Algorithmes généraux d'enveloppe convexe en dimension quelconque et randomisation", *Prépub. Maths. de l'U.R.A.212, Univ. PARIS 7 CNRS-U.R.A.212*, No 50 (1992).

[4] A.Bowyer; "Computing Dirichlet tessellations", *Comput. J.*, 24, 162–166 (1981).

[5] B.DELAUNAY; "Sur la sphère vide", *Izv. Akad. Nauk SSSR Otdelenie Matemat. Estestvennyka Nauk*, Vol 7, pp.793–800 (1934).

[6] P.L.GEORGE AND 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).

[7] P.L.GEORGE, F.HECHT AND 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).

[8] P.J.GREEN AND R.SIBSON; "Computing Dirichlet tessellations in the plane", *Comput. J.*, 21, 168–173 (1978).

[9] L.J.GUIBAS AND J.STOLFI; "Primitives for the manipulation of general subdivisions and the computation of Voronoi diagram", *Proc. 15th ACM Conf. on the Theory of Comput.*, pp.221–234 (1983).

[10] F.HERMELINE; "Triangulation automatique d'un polyèdre en dimension N", *R.A.I.R.O. Analyse num.*, 16, no 3, 211–242 (1982).

[11] C.L.LAWSON; "Properties of n-dimensional triangulations", *Comput. Aided Geom. Design 3*, pp.231–246 (1986).

[12] D.T.LEE AND B.J.SCHACHTER; "Two algorithms for constructing a Delaunay triangulation", *Internat.J.Comput.Inform.Sci.*, 9, 219–242 (1980).

[13] G.VORONOÏ; "Nouvelles application des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire : Recherches sur les parallélоèdres primitifs", *J. Angew. Math.*, Vol 134, 167–171 (1908).

[14] D.F.WATSON; "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes", *Comput.J.*, 24, 167–172 (1981).

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine - Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes - IRISA, Campus universitaire de Beaulieu 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 Grenoble Cedex 1 (France)
Unité de recherche INRIA Sophia Antipolis - 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

ISSN 0249 - 6399