



HAL
open science

Méthodes d'approximation et de géométrie algorithmique pour la reconstruction de courbes et surfaces

Jean-Christophe Roux

► **To cite this version:**

Jean-Christophe Roux. Méthodes d'approximation et de géométrie algorithmique pour la reconstruction de courbes et surfaces. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1994. Français. NNT: . tel-00344528

HAL Id: tel-00344528

<https://theses.hal.science/tel-00344528>

Submitted on 5 Dec 2008

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.

THESE

présentée par

Jean-Christophe ROUX

pour obtenir le titre de

DOCTEUR

DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

(arrêtés ministériels du 5 Juillet 1984 et du 30 Mars 1992)

Spécialité : Mathématiques Appliquées

Méthodes d'approximation et de géométrie algorithmique pour la reconstruction de courbes et surfaces

Soutenue le 17 Février 1994.

Composition du jury :

Président	:	M. P.-J. LAURENT
Rapporteurs	:	M. J.- P. DEDIEU M. A. LE MEHAUTE
Examineurs	:	M. P. CHENIN M. P. CINQUIN M. B. LACOLLE

Thèse préparée au sein du laboratoire LMC-IMAG

à Isa

à mes parents

à mes grands-parents

avec une pensée pour Chantal Favardin

Table des matières

Première partie

Ordonnancement de points appartenant à une courbe . . .	1
Chapitre I Approximation d'un nuage de points par un cercle aux moindres carrés	7
1 Problème de minimisation	9
1.1 Première méthode : résolution sans contraintes	9
1.2 Deuxième méthode : problème avec contraintes	11
2 Interprétation géométrique du problème de minimisation pour le cercle	19
3 Comparaison des deux méthodes	21
3.1 Tests numériques	21
3.2 Exemples d'approximations	27
4 Géométrie des courbes planes	33
4.1 Définitions, notations, rappels	33
4.2 Définition "limite" du cercle osculateur	36
4.3 Etude expérimentale de la convergence du cercle approximant	40
4.4 Projection de points et conservation de l'ordre	48
4.4.1 Fonctions angulaires associées à un arc plan	48
4.4.2 Conservation de l'ordre par projection radiale	50
Chapitre II Ordonnancement de points sur une courbe	55
1 Méthode : situation dans le contexte existant et présentation	56

2	Algorithmes de reconstruction de courbes.	62
2.1	Découpe de l'ensemble des points	62
2.1.1	Le quadtree	62
2.1.2	Critères d'arrêt, méthode de découpe	63
2.2	Approximation par des cercles dans les cellules du quadtree	65
2.3	Ordonnancement des points dans les cellules du quadtree	66
2.4	Raccord des arcs de courbe	67
3	Complexité des algorithmes	75
4	Temps d'exécutions et exemples	80
4.1	Etude de temps	80
4.2	Exemples de données bruitées	90
4.3	Auto-intersections et points de rebroussement	93
5	Application de l'ordonnancement des points : calcul d'une paramétrisation	97

Deuxième partie

	Reconstruction de surfaces	105
	Chapitre III Triangulations : définitions, propriétés et exemples	115
1	Triangulations planes	116
2	Triangulation de Delaunay	118
2.1	Diagramme de Voronoï ([PS 85])	118
2.2	Triangulation de Delaunay dans le plan	120
2.3	Propriétés de la triangulation de Delaunay plane	120
2.4	Triangulation de Delaunay : algorithmes	123
3	Triangulations Dépendantes des Données	126
3.1	Présentation du problème	126
3.2	Principe de calcul des D.D.T.	127

Table des matières	iii
3.3 Coûts associés aux arêtes	128
3.4 Coût associé aux sommets	130
3.5 Algorithmes et complexité du calcul d'une DDT	130
Chapitre IV Triangulations de surfaces	135
1 Triangulation de la sphère	136
1.1 Sphère et métrique associée	136
1.2 Triangles sphériques (définition géométrique)	137
1.3 Triangulation sur la surface d'une sphère	139
2 Triangulation de Delaunay sur la sphère	140
2.1 Diagramme de Voronoï	140
2.2 Triangulation de Delaunay	142
2.3 Propriétés des triangulations	144
2.4 Algorithmes pour la triangulation de la surface de la sphère . .	148
2.4.1 Triangulation de la surface totale de la sphère	150
2.4.2 Triangulation "ouverte" sur la sphère	151
2.4.3 Insertion d'un point dans la triangulation	153
2.5 Complexité des algorithmes	156
3 Méthode de reconstruction, exemples et comparaisons . . .	159
3.1 Description de la méthode de reconstruction	159
3.2 Expérimentations et comparaisons avec les triangulations dépendantes des données	160
Conclusion	187
Bibliographie	191

Remerciements

Ce travail a été réalisé au sein de l'équipe de Géométrie Algorithmique du Laboratoire de Modélisation et Calcul (LMC-IMAG), à Grenoble, sous la responsabilité de Monsieur Bernard Lacolle.

Je tiens à exprimer ma sincère et profonde reconnaissance à Bernard Lacolle, qui a su me guider, me soutenir et me conseiller tout au long de ces années de thèse.

Merci donc pour ta compétence, ton aide et ta disponibilité, sans lesquelles il est sûr que ce manuscrit n'aurait jamais vu le jour.

Je souhaite exprimer également ma gratitude à Messieurs Jean-Pierre Dedieu et Alain Le Méhauté, qui ont accepté d'être les rapporteurs de cette thèse.

Je vous remercie pour l'intérêt que vous avez porté à ce travail, et pour vos judicieux et précieux conseils.

Je remercie très vivement Monsieur Pierre-Jean Laurent d'avoir accepté de présider ce jury.

Je remercie également Messieurs Patrick Chenin et Philippe Cinquin d'avoir bien voulu faire partie du jury.

Je vous remercie tous pour l'honneur que vous me faites par votre présence aujourd'hui.

Je remercie tous les membres de l'équipe de Géométrie Algorithmique pour avoir toujours été prêts à m'apporter leur aide, ainsi que pour la bonne ambiance qu'ils ont su maintenir au cours de ces années de thèse. Merci particulièrement aux "locataires" du bureau 56, Rémi, Mohammed "Tazer", Pierre-Olivier et Bruno pour l'ambiance de franche convivialité qu'ils ont su instaurer.

Je remercie mes parents et mes grands-parents qui m'ont soutenu et encouragé durant mes années d'études.

Enfin, merci à Isabelle, qui m'a supporté et soutenu durant ces années de thèse, et qui a toujours cru en moi, même durant les moments difficiles.

Première partie

Ordonnancement de points appartenant à une courbe

Le problème que nous proposons d'examiner dans cette partie peut s'énoncer en les termes suivants.

Etant donné un ensemble $\{M_i, i = 1, \dots, n\}$ de n points dans le plan, et n'ayant pas d'autre information que les coordonnées de ces points, comment pouvons-nous ordonner cet ensemble ? C'est à dire, en ce qui nous concerne, si ces points proviennent d'un échantillonnage quelconque d'une courbe, comment établir un ordre sur les points qui corresponde à un ordre induit par une paramétrisation naturelle de la courbe, comme la paramétrisation par abscisse curviligne, par exemple.

Plusieurs approches et méthodes pour résoudre ce problème de "reconstruction" de courbes ont été développées par différents auteurs dans différents contextes.

Lorsque l'on s'impose un ordre lié à des valeurs de paramètres : $M_i = M(t_i)$ ($i = 1, \dots, n$) et $t_1 \leq \dots \leq t_n$, on peut alors calculer une courbe de Bézier ou une courbe B-spline passant par les points, ou près de ces points.

Il est alors possible d'appliquer sur les données $M(t_i)$, $t_1 \leq \dots \leq t_n$ les méthodes d'interpolation, d'approximation ou de lissage classiques, et largement développées par de nombreux auteurs ([Far 88],[SL 89], [BFK 84] ou [LMS 91] par exemple).

Cependant, même lorsque l'on ne connaît que l'ordre sur les points $M(t_i)$, la détermination des paramètres t_i associés à chaque point reste un problème difficile et fondamental. Les travaux d'Epstein ([Eps 76]) ou de Foley et Nielson ([FN 89]) traitent de ce problème et font apparaître les propriétés et l'influence que peuvent avoir, sur la qualité de l'approximation, les différentes méthodes de calcul de ces t_i .

Toutefois, dans le travail présenté ici, ni l'ordre ni aucune information sur d'éventuels paramètres ne sont connus.

Dans un autre contexte, où l'on suppose alors que la courbe est simple et fermée, plusieurs approches ont été développées.

Il est possible, par exemple, d'adapter la méthode proposée par Boissonnat pour les surfaces ([Boi 84]). Il faut alors procéder en deux grandes étapes. La première consiste à calculer la triangulation de Delaunay de l'ensemble des n points

du plan supposés appartenir à la courbe. Selon un résultat bien connu, il obtient ainsi une triangulation de l'enveloppe convexe de l'ensemble de points, dont un ensemble d'arêtes fournit une "première approximation" de la courbe.

Le seconde étape est alors une étape de sculpture de ce contour, en ôtant successivement des triangles ayant une arête sur le contour provisoire, et ce jusqu'à ce que tous les points du nuage se retrouvent sur la frontière de la triangulation, formant ainsi le contour recherché. Le temps de calcul est ainsi en $O(n \log n)$, pour une place mémoire en $O(n)$.

Lorsque les points sont obtenus sur un contour fermé à partir de rayons (courbes semi-infinies qui représentent par exemple le bras d'un robot venant toucher l'objet), Alevizos et *al.* ([ABY 90]) montrent que la connaissance de ces rayons et l'information qu'ils fournissent permettent d'instaurer un ordre sur les points et d'obtenir alors une ligne polygonale représentant le contour recherché ; ceci en temps et taille mémoire optimaux.

Edelsbrunner [Ede 87] développe une autre méthode originale et propose d'utiliser les α -enveloppes pour calculer un contour à partir de points donnés.

Ces α -enveloppes sont en fait une généralisation de la notion d'enveloppe convexe, et sont calculés, pour une certaine valeur α réelle fixée, comme l'intersection de tous les disques de rayon $1/\alpha$ (pour $\alpha > 0$) et contenant tous les points de l'ensemble donné. Si $\alpha = 0$, ces disques deviennent alors des demi-plans (et on retrouve à ce moment là l'enveloppe convexe classique) ; on considère le complémentaire des disques de rayon $-1/\alpha$ dans le cas où α est négatif.

Dans un contexte un peu différent, connaissant l'équation d'une courbe algébrique, Johnstone et Bajaj [JB 90] présentent une méthode permettant d'ordonner, au sens énoncé ci-dessus, un ensemble de points de cette courbe par rapport à l'abscisse curviligne, en procédant par une décomposition en parties convexes de la courbe.

Enfin, à partir de la seule donnée des points, Dedieu et Favardin [DF 91] ont développé une méthode consistant à approximer l'ensemble des points par une courbe algébrique plane de degré fixé, calculée sous sa forme implicite.

Puis, toujours à partir de la seule donnée des points, Favardin [Fav 93], a

développé deux algorithmes permettant de reconstruire des courbes simples et non-simples, avec une ou plusieurs composantes connexes dans le cas des courbes simples.

La première méthode fonctionne sur un principe de proximité entre les points, et le contour est successivement reconstruit en recherchant, à une étape donnée, le point le plus proche de l'une des extrémités de l'arc de courbe déjà calculé.

La deuxième, dans le but de traiter les auto-intersections, est basée sur un critère de variation de la tangente à la courbe.

Si les résultats obtenus par de telles méthodes sont probants, il semble néanmoins qu'une certaine fragilité des algorithmes apparaisse lorsque les points sont bruités, et qu'ainsi les distances entre points ou la variation de la tangente sont perturbées.

La principale idée que nous allons développer ici, et dont va dépendre la méthode que nous proposons, est de calculer un ensemble de courbes algébriques simples, données sous leur forme implicite, et qui soient des approximations d'arcs de cette courbe. Il faut noter que cette stratégie, appliquée globalement (ajuster une courbe sous forme implicite), présente de nombreuses difficultés. Ceci nous a amené à une approche que nous qualifions de "mixte", basée sur les deux principes de proximité et d'approximation locale par une courbe algébrique. Nous recherchons de même une méthode peu coûteuse en temps et en structure de données, ayant pour but de traiter un grand nombre de points.

Notre choix des courbes implicites s'est porté sur des arcs de cercle.

Pourquoi ce choix ?

Les principales raisons sont, comme nous le verrons par la suite, que le calcul d'un cercle approximant amène à la minimisation d'une fonctionnelle quadratique simple, et à laquelle nous donnerons un sens géométrique : la minimisation d'une distance par rapport à un ensemble de points.

Nous voulons aussi prendre en compte le problème du bruit sur les données, et nous recherchons une méthode qui présente une bonne stabilité lorsque les points sont perturbés.

D'autre part, le cercle est naturellement lié à une courbe par l'intermédiaire de la courbure, et nous étudierons quelques résultats en relation avec cette

dernière, comme la convergence du cercle approximant vers le cercle osculateur ou la projection sur celui-ci des points d'un arc de courbe. Nous obtenons alors localement une approximation d'ordre 2 de la courbe.

Nous allons donc tout d'abord examiner le problème de l'approximation d'un nuage de points par un cercle, et nous donnerons quelques exemples avant d'évoquer la convergence vers le cercle osculateur en un point, et le problème de la conservation de l'ordre des points par le paramétrage sur les points de la courbe, lorsque ceux-ci sont projetés localement sur un cercle.

Nous nous intéresserons ensuite au problème de reconstruction proprement dit. Nous décrirons les méthodes et algorithmes que nous avons mis en place, avant d'examiner la complexité et d'étudier le comportement des algorithmes sur quelques exemples bruités ou non.

Chapitre I

Approximation d'un nuage de points par un cercle aux moindres carrés

Nous allons nous intéresser, dans ce chapitre, au calcul d'un cercle qui approxime le mieux possible un nuage de points du plan. Dans la pratique, ce nuage sera censé représenter un ensemble de points situés approximativement sur un arc de courbe.

Qu'entendons nous par "le mieux possible" ?

"Le mieux possible" sera par rapport à un critère d'erreur qui tiendra compte de la moyenne des distances de chaque point au cercle calculé. En effet, notre principale préoccupation, comme nous le verrons par la suite, est de calculer un cercle sur lequel nous projetterons les points, et qui permettra de conserver l'ordre de ces points une fois ceux-ci projetés. Il est alors naturel de penser que, plus le cercle sera proche des points en moyenne, plus l'ordre aura de chances d'être respecté après projection.

Notre seconde préoccupation est d'obtenir une méthode de calcul stable, qui réagisse correctement dans le cas de données perturbées ou dans le cas de points (quasi-)alignés.

Nous avons mis au point et testé deux méthodes qui vont être décrites dans les paragraphes suivants.

La différence essentielle, comme nous allons le voir, réside dans le fait que l'un des problèmes de minimisation engendré est contraint et l'autre non.

Nous donnerons ensuite un sens géométrique à la deuxième méthode présentée — celle que nous retiendrons — avant de comparer les résultats obtenus par

8 _____ Chapitre I Approximation d'un nuage de points par un cercle aux moindres carrés
chacune d'elles.

Après quelques rappels sur les courbes paramétriques, nous étudierons quelques propriétés du cercle osculateur et du cercle approximant, qui donnent une interprétation pertinente de la méthode de reconstruction décrite dans le chapitre II.

1 Problème de minimisation

Nous nous placerons, dans toute la suite, dans le plan \mathbb{R}^2 , et l'on désignera indifféremment par $u_i = {}^t(x_i, y_i)$, $i = 1 \dots n$, les points du nuage ou les vecteurs associés.

1.1 Première méthode : résolution sans contraintes

Pour un cercle (C) , de centre ω et de rayon r , on peut écrire l'équation vectorielle :

$$\begin{aligned} {}^t(u - \omega)(u - \omega) &= r^2 \\ \iff {}^tu u - 2{}^tu\omega + {}^t\omega\omega - r^2 &= 0 \end{aligned}$$

pour un point u lui appartenant .

On va alors minimiser la quantité suivante, que certains auteurs appellent distance algébrique, par rapport à ω et r :

$$\begin{aligned} & \text{Min}_{\omega, r} \sum_{i=1}^n \left[({}^tu_i u_i - 2{}^tu_i \omega + {}^t\omega\omega - r^2)^2 \right] \\ &= \text{Min}_{\omega, r} \sum_{i=1}^n \left[{}^tu_i u_i {}^tu_i u_i - 4{}^tu_i u_i {}^tu_i \omega + 4{}^tu_i \omega {}^tu_i \omega - 4{}^tu_i \omega d + 2{}^tu_i u_i d + d^2 \right] \\ &= \text{Min}_{\omega, r} \sum_{i=1}^n \left[(4{}^tu_i \omega {}^tu_i \omega + d^2 - 4{}^tu_i \omega d) + (-4{}^tu_i u_i {}^tu_i \omega + 2{}^tu_i u_i d) \right. \\ & \quad \left. + ({}^tu_i u_i {}^tu_i u_i) \right] \quad \text{où } d = {}^t\omega\omega - r^2. \end{aligned}$$

Soit ${}^t p = ({}^t\omega, d) = (\omega_1, \omega_2, d)$ un vecteur de \mathbb{R}^3 , l'expression s'écrit alors sous la forme :

$$\text{Min}_p \sum_{i=1}^n ({}^t p A_i p + {}^t v_i p + c_i)$$

$$\begin{aligned} \text{avec : } \quad c_i &= x_i^4 + y_i^4 + 2x_i^2 y_i^2 \\ {}^t v_i &= (-4x_i^3 - 4x_i y_i^2, -4y_i^3 - 4x_i^2 y_i, 2(x_i^2 + y_i^2)) \\ A_i &= \begin{pmatrix} 4x_i^2 & 4x_i y_i & -2x_i \\ 4x_i y_i & 4y_i^2 & -2y_i \\ -2x_i & -2y_i & 1 \end{pmatrix} \end{aligned}$$

Finalement, le problème de minimisation est le suivant :

(P)

$$\boxed{\text{Min}_p \quad {}^t p A p + {}^t v p + c}$$

$$\text{avec} \quad A = \sum_{i=1}^n A_i, \quad {}^t v = \sum_{i=1}^n {}^t v_i, \quad c = \sum_{i=1}^n c_i.$$

En ce qui concerne les solutions du problème (P), nous avons la propriété suivante sur la matrice A :

Propriété 1.1 *La matrice A associée au problème (P) ci-dessus est définie positive si et seulement si les points u_i données du problème ne sont pas alignés.*

Preuve : Soit ${}^t p = ({}^t \omega, d) = (\omega_1, \omega_2, d)$ un vecteur de \mathbb{R}^3 . Nous avons alors :

$$\begin{aligned} {}^t p A p &= \sum_{i=1}^n {}^t p A_i p \\ &= \sum_{i=1}^n 4x_i^2 \omega_1^2 + 4y_i^2 \omega_2^2 + d^2 + 8x_i y_i \omega_1 \omega_2 - 4x_i \omega_1 d - 4y_i \omega_2 d \\ &= \sum_{i=1}^n (2x_i \omega_1 + 2y_i \omega_2 - d)^2 \end{aligned}$$

d'où :

$$\forall p \in \mathbb{R}^3, \quad {}^t p A p \geq 0$$

et, si ${}^t p \neq 0$:

$${}^t p A p = 0 \iff 2x_i \omega_1 + 2y_i \omega_2 - d = 0, \text{ pour tout } (x_i, y_i) \in \mathbb{R}^3.$$

Ce qui équivaut à dire que tous les points u_i données du problème appartiennent à la même droite. □

Ceci nous assure donc l'unicité de la solution du problème (P).

Il est alors possible de résoudre numériquement (P) soit par une méthode de type gradient conjugué, soit en calculant directement la solution.

En effet, comme A est définie positive dans le cas où les points ne sont pas alignés, la solution \bar{p} de (P) (reformulé comme : $\text{Min}_p J(p)$) doit vérifier :

$$\begin{aligned} J'(p) = 0 &\iff 2A\bar{p} + v = 0 \\ &\iff A\bar{p} = -(1/2)v. \end{aligned}$$

1.2 Deuxième méthode : problème avec contraintes

Pratt [Pra 87] présente une méthode avec contrainte pour le calcul d'une sphère qui a l'avantage de ne pas souffrir d'instabilité lorsque les points sont quasiment alignés. La conséquence de ceci est que l'on n'a plus à ce moment un problème de moindres carrés pouvant de résoudre de manière directe.

Soit (\mathcal{C}) un cercle dont l'équation peut s'écrire sous la forme :

$$F(x, y) = a_1(x^2 + y^2) + a_2x + a_3y + a_4 = 0$$

En posant ${}^t u = (x, y)$, on peut réécrire cette équation sous la forme :

$$F(u) = a_1 {}^t u u + {}^t u \omega + a_4 = 0 \quad \text{avec } {}^t \omega = (a_2, a_3)$$

Nous allons introduire une contrainte qui donne une interprétation géométrique intéressante au problème énoncé en ces termes : "Trouver le cercle (\mathcal{C}) *approchant le mieux possible un ensemble de n points du plan*". Nous verrons au paragraphe 2 la signification mathématique et géométrique de ce que nous entendons par là, ainsi que le pourquoi de la contrainte que nous allons présenter.

Nous allons donc calculer la quantité :

$$\text{Min}_{a_1, \omega, a_4} \sum_{i=1}^n (F(u_i))^2$$

en imposant la contrainte

$$\|\nabla F(u)\|^2 = 1 \quad (u \in (\mathcal{C})).$$

Exprimons la contrainte explicitement :

$$\frac{\partial F(x, y)}{\partial x} = 2a_1x + a_2 \text{ et } \frac{\partial F(x, y)}{\partial y} = 2a_2y + a_3$$

d'où :

$$\begin{aligned} \|\nabla F(u)\|^2 &= 4a_1^2x^2 + a_2^2 + 4a_1a_2x + 4a_1^2y^2 + a_3^2 + 4a_1a_3y \\ &= a_2^2 + a_3^2 + 4a_1(a_1(x^2 + y^2) + a_2x + a_3y) \end{aligned}$$

ce qui est égal à

$$a_2^2 + a_3^2 - 4a_1a_4$$

si $F(x, y) = 0$.

D'autre part, en posant toujours $u_i = {}^t(x_i, y_i) (i = 1, \dots, n)$,

$$\begin{aligned} & \text{Min}_{a_1, \omega, a_4} \sum_{i=1}^n (a_1 {}^t u_i u_i + {}^t u_i \omega + a_4)^2 \\ &= \text{Min}_{a_1, \omega, a_4} \sum_{i=1}^n \left((a_1)^2 {}^t u_i u_i {}^t u_i u_i + {}^t u_i \omega {}^t u_i \omega + (a_4)^2 + 2a_1 {}^t u_i u_i {}^t u_i \omega \right. \\ & \quad \left. + 2a_1 a_4 {}^t u_i u_i + 2a_4 {}^t u_i \omega \right) \\ &= \text{Min}_p \sum_{i=1}^n {}^t_p M_i p \end{aligned}$$

avec : ${}^t_p = (a_1, {}^t\omega, a_4) = (a_1, a_2, a_3, a_4)$,

$$M_i = \begin{pmatrix} ({}^t u_i u_i)^2 & x_i ({}^t u_i u_i) & y_i ({}^t u_i u_i) & {}^t u_i u_i \\ x_i ({}^t u_i u_i) & x_i^2 & x_i y_i & x_i \\ y_i ({}^t u_i u_i) & x_i y_i & y_i^2 & y_i \\ {}^t u_i u_i & x_i & y_i & 1 \end{pmatrix}.$$

Si l'on pose : $M = \sum_{i=1}^n M_i$, le problème de minimisation est alors le suivant :

(P_{cont})

$\text{Min}_p {}^t_p M p \quad \text{sous la contrainte : } a_2^2 + a_3^2 - 4a_1 a_4 = 1$

Remarque 1.2 : La méthode est indépendante de la dimension de l'espace et s'applique de manière similaire pour le calcul d'une sphère aux moindres carrés.

Dans ce cas là, la matrice M_i sera de la forme :

$$M_i = \begin{pmatrix} ({}^t u_i u_i)^2 & x_i ({}^t u_i u_i) & y_i ({}^t u_i u_i) & z_i ({}^t u_i u_i) & {}^t u_i u_i \\ x_i ({}^t u_i u_i) & x_i^2 & x_i y_i & x_i z_i & x_i \\ y_i ({}^t u_i u_i) & x_i y_i & y_i^2 & y_i z_i & y_i \\ z_i ({}^t u_i u_i) & x_i z_i & y_i z_i & z_i^2 & z_i \\ {}^t u_i u_i & x_i & y_i & z_i & 1 \end{pmatrix}$$

et le vecteur ${}^t p = (a_1, a_2, a_3, a_4, a_5)$ impliquera la contrainte :

$$a_2^2 + a_3^2 + a_4^2 - 4a_1 a_5 = 1.$$

Nous pouvons alors énoncer la propriété suivante :

Propriété 1.3 Soit un ensemble de n points du plan comportant au moins trois points non-confondus. Soit d'autre part M la matrice du problème (P_{cont}) ci-dessus, calculée à partir de cet ensemble.

(i) M est symétrique et vérifie ${}^t p M p \geq 0$, pour tout $p \in \mathbb{R}^4$.

(ii) M est définie positive si et seulement si les n points de l'ensemble donné ne sont pas alignés ou cocycliques.

(iii) Si les n points donnés sont alignés ou cocycliques, alors il existe un unique vecteur p_0 (à une constante multiplicative près) tel que $M p_0 = 0$.

Preuve : Soit $p = (a_1, a_2, a_3, a_4)$ un vecteur quelconque donné de \mathbb{R}^4 . Alors :

$$\begin{aligned} {}^t p M p &= \sum_{i=1}^n {}^t p M_i p \\ &= \sum_{i=1}^n \left(a_1 (x_i^2 + y_i^2) + a_2 x_i + a_3 y_i + a_4 \right)^2 \\ &\geq 0 \end{aligned}$$

ce qui prouve (i).

D'autre part,

$$\begin{aligned} {}^t p M p &= 0 \\ \iff a_1 (x_i^2 + y_i^2) + a_2 x_i + a_3 y_i + a_4 &= 0 \quad (\forall i = 1, \dots, n) \\ \iff (x_i, y_i) \in \left\{ (x, y) \in \mathbb{R}^2 \text{ tels que } a_1 (x^2 + y^2) + a_2 x + a_3 y + a_4 &= 0 \right\} = E \end{aligned}$$

L'ensemble E est, si $p \neq 0$, soit un cercle, soit une droite. Il en résulte donc (ii).

Supposons maintenant que tous les points u_i donnés appartiennent à un cercle d'équation $\alpha_1(x^2 + y^2) + \alpha_2x + \alpha_3y + \alpha_4 = 0$.

Considérons la matrice M_i associée à l'un des points u_i , alors :

$$M_i p = 0,$$

d'où, si $p = {}^t(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = p_0$, alors

$$\forall i = 1, \dots, n, M_i p = 0 \iff M p = 0.$$

Réciproquement, $M p_0 = 0 \implies {}^t p_0 M p_0 = 0$ et ceci entraîne donc que :

$$\alpha_1(x_i^2 + y_i^2) + \alpha_2x_i + \alpha_3y_i + \alpha_4 = 0, \text{ pour tout } i = 1, \dots, n,$$

et ceci définit un cercle unique à une constante multiplicative près, puisqu'il passe par au moins trois points, et donc p_0 est unique. \square

Notation : $\mathcal{M}_{n,n}(\mathbb{R})$ désignera dans la suite l'ensemble des matrices carrées de taille n et à coefficients réels.

Lemme 1.4 Soient $M \in \mathcal{M}_{n,n}(\mathbb{R})$ symétrique définie positive, $A \in \mathcal{M}_{n,n}(\mathbb{R})$ inversible et $k \in \mathbb{R}^*$

Soient d'autre part :

$$\begin{aligned} J : \mathbb{R}^n &\longrightarrow \mathbb{R} \\ p &\longmapsto {}^t p M p \end{aligned}$$

et $\mathcal{U} = \{p \in \mathbb{R}^n, {}^t p A p = k\}$, supposé non-vide.

Une condition nécessaire pour que J admette un extremum relatif en p par rapport à l'ensemble \mathcal{U} est que p soit un vecteur propre de la matrice $A^{-1}M$.

Preuve : J est définie positive et dérivable sur \mathbb{R}^n , de dérivée $J' : \mathbb{R}^n \rightarrow \mathbb{R}, p \mapsto 2Mp$.

Si \mathcal{U} est défini par : $\{p : \varphi(p) = {}^t p A p = k\}$, alors $\varphi'(p) = 2Ap \neq 0$ pour tout $p \in \mathcal{U}$.

En effet, $2Ap = 0 \implies p = 0$ car A est inversible, et ${}^t p A p = 0 \neq k$ est en contradiction avec le fait que $p \in \mathcal{U}$.

En appliquant alors la théorie des extrema liés (cf. [Cia 82] page 148), une condition nécessaire d'existence d'un extremum relatif est :

$$\begin{aligned} \exists \lambda \in \mathbb{R} \text{ tel que } J'(p) + \lambda \varphi'(p) &= 0 \\ \iff A^{-1}Mp &= \lambda p. \end{aligned}$$

Ce qui revient à dire que p est un vecteur propre de la matrice $A^{-1}M$. \square

Après avoir énoncé une condition nécessaire d'existence d'un extremum pour la fonctionnelle J du lemme 1.4, nous allons maintenant examiner le problème de l'existence d'un minimum.

Théorème 1.5 Soient $M \in \mathcal{M}_{n,n}(\mathbb{R})$ symétrique définie positive, $A \in \mathcal{M}_{n,n}(\mathbb{R})$ inversible et $k \in \mathbb{R}^*$

Soient :

$$\begin{aligned} J : \mathbb{R}^n &\longrightarrow \mathbb{R} \\ p &\longmapsto {}^t p M p \end{aligned}$$

et $\mathcal{U} = \{p \in \mathbb{R}^n, {}^t p A p = k\}$, supposé non-vidé.

Alors :

- (i) Il existe toujours un minimum relatif de J par rapport à \mathcal{U} .
- (ii) Si $k > 0$, $A^{-1}M$ admet au moins une valeur propre positive et $p \in \mathcal{U}$ est un minimum relatif de J si et seulement si p est le vecteur propre associé à la plus petite valeur propre positive de $A^{-1}M$, qui vérifie ${}^t p A p = k$.
- (iii) Si $k < 0$, $A^{-1}M$ admet au moins une valeur propre négative et $p \in \mathcal{U}$ est un minimum relatif de J si et seulement si p est le vecteur propre associé à la valeur propre négative de $A^{-1}M$ de plus petit module, qui vérifie ${}^t p A p = k$.

Preuve : (i) L'existence résulte du fait que l'on cherche le minimum d'une norme $\|p\|_M = ({}^t p M p)^{1/2}$ sur l'ensemble \mathcal{U} .

Si l'on suppose que cet ensemble est non-vidé, alors il existe $p_0 \in \mathcal{U}$ tel que l'on puisse restreindre la recherche du minimum à $B \cap \mathcal{U}$ où B désigne la boule centrée en l'origine et de rayon $\|p_0\|_M$.

La recherche d'un minimum revient alors à trouver le minimum d'une fonction continue sur un compact $K = B \cap \mathcal{U}$, et ce minimum existe et est atteint.

(ii) et (iii) D'après le lemme 1.4, il suffit de considérer les valeurs propres réelles de $A^{-1}M$ pour obtenir un extremum relatif. Et il nous faut d'abord restreindre notre choix d'extremum aux vecteurs propres qui vérifient ${}^t pAp = k$.

Or si p vérifie ${}^t pAp = k$ et $A^{-1}Mp = \lambda p$, alors

$${}^t pMp = \lambda {}^t pAp = \lambda k.$$

Le choix revient donc alors à retenir les vecteurs propres relatifs à la plus petite valeur propre positive, dans le cas $k > 0$, qui rendront minimum $\lambda k = {}^t pAp$.

L'existence étant assurée, nous pouvons donc affirmer : si p est associé à la plus petite valeur propre positive de $A^{-1}M$, notée λ_{min}^+ , on a alors :

$${}^t pMp = \lambda_{min}^+ {}^t pAp > 0,$$

et l'on peut toujours multiplier p par une constante pour obtenir ${}^t pAp = k$.

De même, si $k < 0$, le raisonnement est identique. Si λ_{max}^- désigne la valeur propre négative de plus petit module,

$${}^t pMp = \lambda_{max}^- {}^t pAp > 0$$

et on peut choisir une constante $\mu \in \mathbb{R}$ telle que :

$${}^t(\mu p)A(\mu p) = \mu^2 {}^t pAp = k.$$

Retrouvons l'existence d'une telle valeur propre pour $A^{-1}M$ par un calcul matriciel.

M étant symétrique et définie positive, on peut écrire M sous la forme¹ :

$$M = {}^t SS \text{ avec } S \in \mathcal{M}_{n,n}(\mathbb{R}) \text{ inversible.}$$

S'il existe $\lambda \in \mathbb{R}$ et $p \in \mathbb{R}^4$ non-nul tels que :

$$Mp = \lambda Ap,$$

¹ par une factorisation de type Choleski, par exemple.

alors :

$$\begin{aligned} Mp &= {}^tSSp = \lambda Ap \\ \iff Sp &= \lambda {}^t(S^{-1})Ap \\ \iff Sp &= \lambda {}^t(S^{-1})AS^{-1}Sp, \end{aligned}$$

ce qui est équivalent à dire qu'il existe $q \neq 0$ tel que :

$$q = \lambda {}^t(S^{-1})AS^{-1}q.$$

La matrice ${}^t(S^{-1})AS^{-1}$ est inversible et symétrique, car

$${}^t({}^t(S^{-1})AS^{-1}) = {}^t(S^{-1}) {}^tA {}^t({}^t(S^{-1})) = {}^t(S^{-1})AS^{-1}.$$

Elle est donc diagonalisable dans une base de vecteurs propres, et elle admet des valeurs propres $1/\lambda_i$ toutes réelles.

D'autre part, si $\forall i, \lambda_i > 0$, alors

$$\begin{aligned} \forall i, 1/\lambda_i > 0 \text{ et } , \forall q \neq 0, {}^tq {}^t(S^{-1})AS^{-1}q &> 0 \\ \iff \forall r \neq 0, {}^tr Ar &> 0. \end{aligned}$$

De même,

$$\forall i, 1/\lambda_i < 0 \implies \forall r \neq 0, {}^tr Ar < 0.$$

En conséquence, si $k > 0$ et $\mathcal{U} = \{ {}^t pAp = k \} \neq \emptyset$, alors il existe λ_i valeur propre de A strictement positive et la proposition équivalente si k est négatif.

Ces valeurs étant atteintes, les conditions énoncées en (ii) et (iii) deviennent alors nécessaires et suffisantes. \square

Nous pouvons alors énoncer, en ce qui concerne la résolution du problème (P_{cont}), le théorème suivant :

Théorème 1.6 *Le problème de minimisation :*

$$\text{Min}_p \quad {}^t p M p \quad \text{sous la contrainte } (a_2)^2 + (a_3)^2 - 4a_1 a_4 = 1$$

$$\text{où } {}^t p = (a_1, a_2, a_3, a_4)$$

admet toujours une solution.

De plus, un vecteur \bar{p} est solution si et seulement si :

(i) \bar{p} est un vecteur propre associé à la plus petite valeur propre positive

de la matrice $A^{-1}M$, avec $A = \begin{pmatrix} 0 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 \end{pmatrix}$.

(ii) ${}^t p A p = 1$.

Preuve : Tout d'abord, si tous les points u_i données du problème sont alignés ou cocycliques, la valeur du minimum se déduit de la propriété 1.3 et vaut 0. On en déduit aussi que \bar{p} normalisé par ${}^t \bar{p} A \bar{p} = 1$ est l'unique vecteur vérifiant $A^{-1}M\bar{p} = 0$.

En effet, s'il existe $\tilde{p} (\neq \mu \bar{p}, \mu \in \mathbb{R})$ tel que $A^{-1}M\tilde{p} = 0$, alors on a également $M\tilde{p} = 0$, ce qui est en contradiction avec la propriété 1.3 .

Ensuite, si tous les points u_i sont ni alignés ni cocycliques, alors M est définie positive d'après la propriété 1.3 . En conséquence, le problème $\text{Min}_{{}^t p A p = 1} {}^t p M p$ vérifie les hypothèses du théorème 1.5 , avec $k > 0$, puisque $\mathcal{U} = \left\{ p = {}^t(a_1, a_2, a_3, a_4), (a_2)^2 + (a_3)^2 - 4a_1 a_4 = 1 \right\}$ est évidemment non-vide.

On peut alors en déduire que \bar{p} est solution et existe si et seulement si \bar{p} est le vecteur propre associé à la plus petite valeur propre positive de $A^{-1}M$ et tel que ${}^t \bar{p} A \bar{p} = 1$. □

2 Interprétation géométrique du problème de minimisation pour le cercle

Nous allons étudier dans cette partie la signification géométrique du problème de minimisation tel que nous l'avons posé dans le paragraphe 1.2 . Nous allons pour ceci nous intéresser à la signification de la contrainte du problème telle que nous l'avons définie .

$$\text{Soit } F : \begin{cases} \mathbb{R}^2 \longrightarrow \mathbb{R} \\ M \longmapsto F(M) \end{cases}$$

Considérons le cercle (\mathcal{C}) défini par $F(M) = 0$, avec l'équation normalisée $a_2^2 + a_3^2 - 4a_1a_4$, ce qui signifie :

$$\|\nabla F(M)\|^2 = 1, \forall M \in (\mathcal{C}).$$

Quel que soit M , on peut définir M_0 comme la projection orthogonale de M sur (\mathcal{C}) , et écrire :

$$M = M_0 + t\nabla F(M_0), \quad t \in \mathbb{R}$$

où $|t|$ représente la distance de M à \mathcal{C} , et M_0 est unique dans le cas général.

Ainsi,

$$F(M) = F(M_0 + t\nabla F(M_0)),$$

et dans un voisinage du point M_0 (*i.e.* près du cercle), $|t|$ est petit. Nous pouvons donc écrire :

$$\begin{aligned} F(M) &\simeq F(M_0) + t \nabla F(M_0) \cdot \nabla F(M_0), \\ &= F(M_0) + t \|\nabla F(M_0)\|^2. \end{aligned}$$

Par conséquent,

$$F(M) - F(M_0) \simeq t \|\nabla F(M_0)\|^2 = t,$$

et $F(M) - F(M_0)$ constitue une approximation à l'ordre 1 de la distance algébrique de M à \mathcal{C} .

Donc, si on impose $\|\nabla F(M_0)\| = 1$, $\forall M_0 \in \mathcal{C}$ dans un voisinage de \mathcal{C} , nous obtenons :

$$|F(M)| \simeq d(M, M_0) = \underset{\bar{M} \in (\mathcal{C})}{\text{Min}} d(M, \bar{M}) = d(M, \mathcal{C}) .$$

D'où, pour M proche de \mathcal{C} :

$$\|\nabla F(u)\|^2 = 1 \quad (u \in \mathcal{C}) \quad \implies \quad F(M) \simeq d(M, \mathcal{C})$$

Remarque 2.1 : La démonstration qui est faite ici dans le cas du cercle se généralise aisément à \mathbb{R}^n , en considérant alors au lieu du cercle une sphère \mathcal{S}^{n-1} de cet espace.

Remarque 2.2 : La contrainte telle qu'elle est formulée est un invariant géométrique en translation et rotation, et ce résultat reste donc valable dans notre cas quel que soit le nuage de points donné.

Ceci nous fournit alors en pratique une méthode efficace et robuste pour calculer un cercle qui approxime le mieux possible un ensemble de points, même si les données sont perturbées, ou si les points sont pratiquement sur une droite.

Ce dernier cas, d'ailleurs, est un exemple typique qui fait échouer la méthode sans contraintes, puisque en effet la matrice du problème de minimisation devient alors singulière, ce qui n'est pas le cas pour le problème contraint.

3 Comparaison des deux méthodes

Si la première méthode conduit à un algorithme de résolution numérique plus simple à mettre en œuvre et peu coûteux en temps de calcul ($O(n)$ pour l'initialisation des calculs si n représente le nombre de points du nuage ; la convergence de l'algorithme du gradient est assurée en d itérations, ce qui correspond à la dimension de la matrice A dans l'expression de (P)), elle présente néanmoins un manque de robustesse dans le cas où les points sont approximativement situés sur une droite, et plus généralement lorsque la forme du nuage est assez "aplatie" et que de ce fait les points sont situés sur une partie restreinte du cercle seulement. Dans ce cas en effet, le centre du cercle approximant devrait être assez éloigné du nuage et la valeur du rayon assez importante.

3.1 Tests numériques

Nous avons testé les deux méthodes décrites au paragraphe 1.1 et au paragraphe 1.2 pour l'approximation de points bruités (ou non) situés sur un arc de cercle.

Perturbation des données

Précisons que le bruit correspond, selon une variable aléatoire Gaussienne, à augmenter ou diminuer la valeur du rayon d'au plus le pourcentage correspondant.

Erreur calculée

D'autre part, l'erreur calculée correspond à la moyenne des écarts en distance entre les points et le cercle calculé.

Nous avons de même fait figurer ce que représente l'erreur commise en pourcentage du rayon calculé.

Ce résultat est important de notre point de vue, si nous n'oublions pas que nous recherchons un cercle approximant qui puisse nous garantir ensuite la conservation de l'ordre par projection. Il est alors évident que plus ce pourcentage sera faible, plus les points seront proches du cercle en moyenne ; et meilleures seront les chances de conserver effectivement l'ordre.

Nous avons donc extrait des différents tests et mesures que nous avons

effectués les quatre exemples dont les résultats figurent dans les tableaux ci-dessous.

Nous présentons ainsi tout d'abord des jeux d'essais portant sur 100 points (ce qui correspond à une densité importante de points par rapport à la longueur des arcs), puis sur 10 points, pour illustrer le comportement des deux méthodes dans des cas de haute et faible densité, et qui correspondront en pratique aux configurations de points que nous aurons à approximer sur des arcs de courbes.

Dans les tableaux 1 à 4 ci-dessous figurent donc, selon différentes valeurs de bruit appliquées aux points, les coordonnées du centre du cercle approximant calculé par les deux méthodes, ainsi que la valeur de son rayon et l'erreur commise. Les coordonnées du centre et le rayon du cercle initial sur lequel ont été bruités (ou non) les points figurent en légende de chaque tableau.

Nombre de points : 100

Bruit	Centre et rayon calculés par la méthode sans contraintes	Centre et rayon calculés par la méthode avec contraintes
0	<i>Centre et rayon exacts</i>	<i>Centre et rayon exacts</i>
2%	Centre : 1,161346 1,074977 Rayon : 0,828786 Erreur : 0,010319 % Rayon : 1,2 %	Centre : 0,987490 0,986733 Rayon : 1,015531 Erreur : 0,009547 % Rayon : 0,9 %
3,3%	Centre : 1,322239 1,142100 Rayon : 0,662800 Erreur : 0,017067 % Rayon : 2,6 %	Centre : 0,988262 0,979268 Rayon : 1,024649 Erreur : 0,014470 % Rayon : 1,4 %
10%	Centre : 1,322284 1,345633 Rayon : 0,288224 Erreur : 0,081791 % Rayon : 28,4 %	Centre : 0,899687 0,905994 Rayon : 1,126682 Erreur : 0,047769 % Rayon : 4,2 %

Tableau 1 : Résultats de l'approximation d'un arc de cercle de longueur $\pi/2$, de centre (1,1) et de rayon 1

Nombre de points : 100

Bruit	Centre et rayon calculés par la méthode sans contraintes	Centre et rayon calculés par la méthode avec contraintes
0	<i>Centre et rayon exacts</i>	<i>Centre et rayon exacts</i>
2%	Centre: 1,805590 1,160818 Rayon: 0,205255 Erreur: 0,020136 % Rayon: 9,8 %	Centre: 0,929496 0,969963 Rayon: 1,073955 Erreur: 0,009544 % Rayon: 0,8 %
3,3%	Centre: 1,908065 1,179531 Rayon: 0,130397 Erreur: 0,035944 % Rayon: 27,5 %	Centre: 0,872135 0,948168 Rayon: 1,134087 Erreur: 0,014451 % Rayon: 0,8 %
10%	Centre: 1,988548 1,193954 Rayon: 0,122620 Erreur: 0,042660 % Rayon: 33,8 %	Centre: 1,991962 1,205314 Rayon: 0,138530 Erreur: 0,049630 % Rayon: 35,8 %

Tableau 2: Résultats de l'approximation pour un arc de cercle de longueur $\pi/4$, de centre (1,1) et de rayon 1

Nombre de points : 10

Bruit	Centre et rayon calculés par la méthode sans contraintes	Centre et rayon calculés par la méthode avec contraintes
0	<i>Centre et rayon exacts</i>	<i>Centre et rayon exacts</i>
2%	Centre : 0,059340 0,056307 Rayon : 0,925769 Erreur : 0,010411 % Rayon : 1,1 %	Centre : -0,049803 -0,078991 Rayon : 1,082772 Erreur : 0,006464 % Rayon : 0,6 %
3,3%	Centre : 0,110481 0,101676 Rayon : 0,863407 Erreur : 0,015754 % Rayon : 1,8 %	Centre : -0,078807 -0,125013 Rayon : 1,131776 Erreur : 0,009786 % Rayon : 0,9 %
10%	Centre : 0,444747 0,351460 Rayon : 0,492321 Erreur : 0,081791 % Rayon : 10,7 %	Centre : -0,392831 -0,618400 Rayon : 1,680125 Erreur : 0,032050 % Rayon : 1,9 %

Tableau 3 : Résultats de l'approximation d'un arc de cercle de longueur $\pi/2$, de centre $(0,0)$ et de rayon 1

Pour une longueur d'arc égale à la moitié de l'arc précédent :

Nombre de points : 10

Bruit	Centre et rayon calculés par la méthode sans contraintes	Centre et rayon calculés par la méthode avec contraintes
0	<i>Centre et rayon exacts</i>	<i>Centre et rayon exacts</i>
2%	Centre : 0,454950 0,169914 Rayon : 0,529296 Erreur : 0,010692 % Rayon : 9,8 %	Centre : -0,438458 -0,240978 Rayon : 1,489756 Erreur : 0,006475 % Rayon : 0,4 %
3,3%	Centre : 0,623044 0,230598 Rayon : 0,364715 Erreur : 0,018329 % Rayon : 5 %	Centre : -0,902220 -0,495230 Rayon : 2,013159 Erreur : 0,009797 % Rayon : 0,5 %
10%	Centre : 0,886700 0,328621 Rayon : 0,181464 Erreur : 0,057811 % Rayon : 31,8 %	Centre : 1,807406 0,964683 Rayon : 1,108162 Erreur : 0,031302 % Rayon : 2,8 %

Tableau 4 : Résultats de l'approximation pour un arc de cercle de longueur $\pi/4$, de centre $(0,0)$ et de rayon 1

Il ressort évidemment de ces tests (et d'autres effectués ...) que la méthode avec contrainte s'avère plus robuste dans la majorité des cas que la première méthode envisagée.

En effet, pour une longueur d'arc égale à $\pi/2$, il apparaît clairement que la deuxième méthode fournit un cercle qui assurera, une fois les points projetés, l'ordre qu'il serait possible d'instaurer sur les points en les classant selon leur angle en coordonnées polaires, et ceci quelle que soit la valeur du bruit. La méthode sans contraintes est défailante dans les cas où les points sont fortement bruités.

Il apparaît d'autre part que, plus la valeur de la longueur de l'arc diminue, plus l'erreur augmente en ce qui concerne la première méthode, alors que la résolution du problème avec contrainte se montre très stable par rapport à cette variation.

De grandes différences apparaissent aussi dans les pourcentages d'erreurs commises par rapport au rayon calculé, et font ainsi ressortir la plus grande fiabilité de la méthode avec contraintes pour ce type d'approximation.

Et ces remarques restent encore valables quel que soit le nombre de points.

Nous citerons enfin pour anecdote certains jeux d'essais qui ont été accomplis pour simuler des ensembles de points quasi-alignés. Nous avons pris pour cela des cercles de "très grands" rayons ($\simeq 10^6$), et des points distribués sur de "très petits" arcs (longueur $\simeq 10^{-5}, 10^{-6}$).

Nous avons alors constaté une très grande instabilité numérique en ce qui concerne la première méthode, voire des échecs répétés (non-convergence, division par 0, etc...) et dénotant du mauvais conditionnement de la matrice à ce moment là.

Dans le même temps, nous avons obtenu avec la méthode contrainte des résultats de la même qualité que celle d'expérimentations moins "extrêmes".

Pour un cercle de centre $(0,0)$, et de rayon 10^6 , et 100 points situés sur un arc de longueur 10^{-6} , nous avons obtenu par exemple pour la méthode avec contraintes :

- centre = $(4,5.10^5, 0,68)$,
- rayon = $5,5.10^5$,
- erreur = $2,5.10^{-7}$,

alors que dans le même temps l'algorithme de gradient conjugué pour résoudre le problème de minimisation sans contraintes ne convergeait pas.

3.2 Exemples d'approximations

Nous illustrons dans ce paragraphe certains des résultats numériques obtenus précédemment. Sur la première série de figures, sont représentées les approximations par la méthode sans contrainte, et sur la seconde des cas similaires traités par la méthode avec contrainte.

Ces résultats graphiques sont représentés à partir d'échantillons aléatoires suivant une loi gaussienne centrée réduite, et selon des échelles différentes.

Méthode sans contrainte :

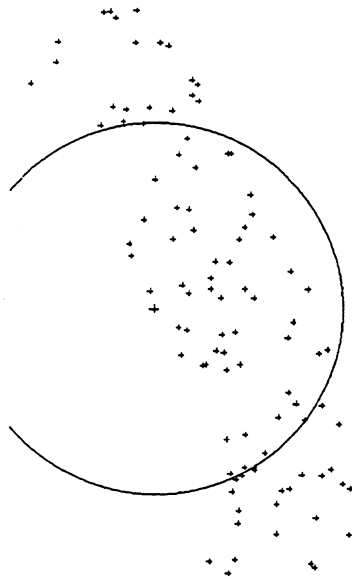


Figure 3.1 Cas n° 1, méthode sans contrainte, points bruités (10%) sur un arc de cercle de longueur $\pi/2$.

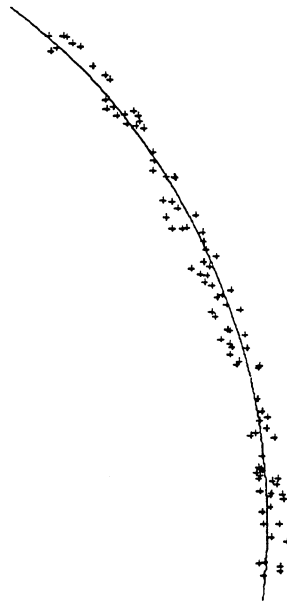


Figure 3.2 Cas n° 2, méthode sans contrainte, points bruités (2%) sur un arc de cercle de longueur $\pi/2$.

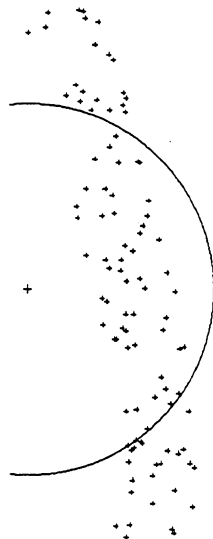


Figure 3.3 Cas n° 3, méthode sans contrainte, points bruités (3.3%) sur un arc de cercle de longueur $\pi/4$.

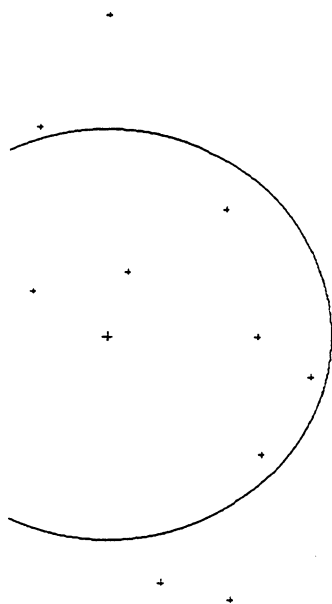


Figure 3.4 Cas n° 4, méthode sans contrainte, points bruités (10%) sur un arc de cercle de longueur $\pi/4$.

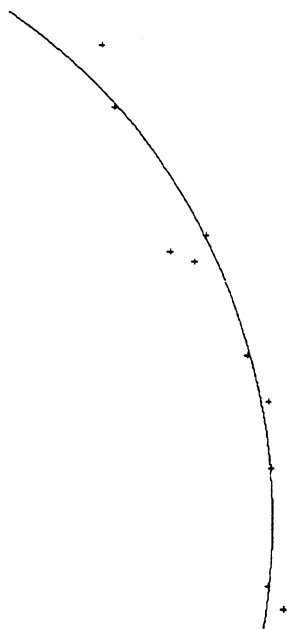


Figure 3.5 Cas n° 5, méthode sans contrainte, points bruités (2%) sur un arc de cercle de longueur $\pi/4$.

Méthode avec contrainte :

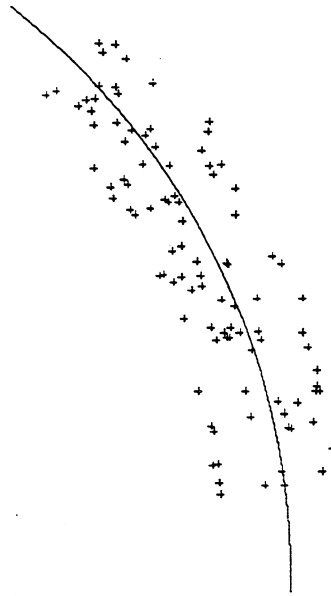


Figure 3.6 Cas n° 1, méthode avec contrainte, points bruités (10%) sur un arc de cercle de longueur $\pi/2$.

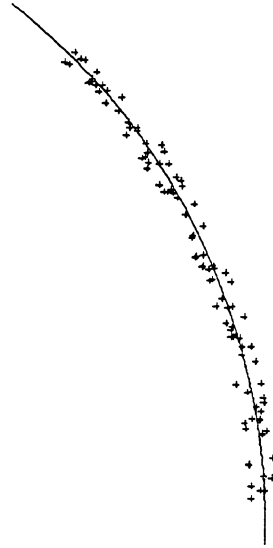


Figure 3.7 Cas n° 2, méthode avec contrainte, points bruités (2%) sur un arc de cercle de longueur $\pi/2$.

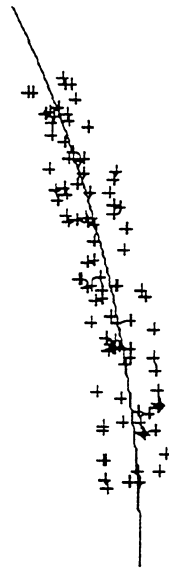


Figure 3.8 Cas n° 3, méthode avec contrainte, points bruités (3.3%) sur un arc de cercle de longueur $\pi/4$.

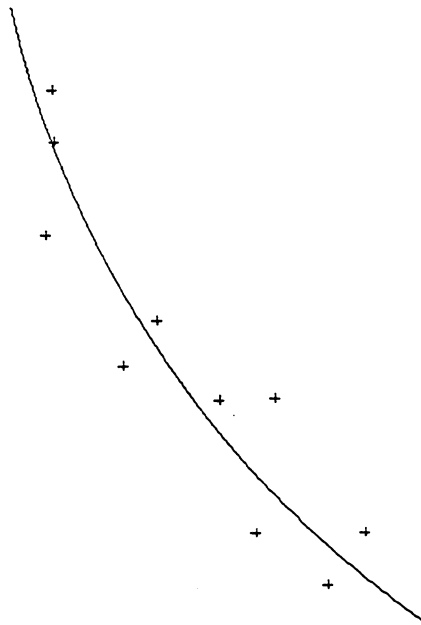


Figure 3.9 Cas n° 4, méthode avec contrainte, points bruités (10%) sur un arc de cercle de longueur $\pi/4$.

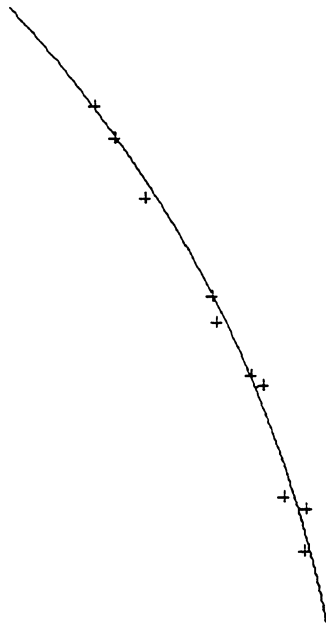


Figure 3.10 Cas n° 5, méthode avec contrainte, points bruités (2%) sur un arc de cercle de longueur $\pi/4$.

4 Géométrie des courbes planes

Nous allons, dans ce paragraphe, rappeler tout d'abord quelques résultats classiques portant sur la géométrie des courbes planes.

Nous examinerons ensuite quelques résultats qui définissent le cercle osculateur à une courbe en un point par un passage à la limite, avant d'étudier expérimentalement la convergence du meilleur cercle approximant vers un tel cercle.

Enfin, nous nous pencherons sur des problèmes liés à la notion d'ordre sur une courbe, et à la conservation de cet ordre par projection sur un cercle.

4.1 Définitions, notations, rappels

Dans toute la suite de ce paragraphe, E désignera \mathbb{R}^2 ou \mathbb{R}^3 considéré en temps qu'espace vectoriel normé.

Définition 4.1 [Maz 91] Représentation paramétrique C^p

On appelle *représentation paramétrique de classe C^p* , ($p \geq 1$), d'un arc de courbe γ , toute application f d'un intervalle d'intérieur non-vide I de \mathbb{R} dans E telle que :

- 1°. f est bijective de I sur $\gamma = f(I)$.
- 2°. f est de classe C^p sur I .

Définition 4.2 [Maz 91] Représentation paramétrique régulière

Soit γ un arc C^k , $k \geq 1$, défini par une paramétrisation $f : I \rightarrow E$. f est dite *régulière* si :

$$f'(t) \neq 0, \forall t \in I.$$

On dira alors que γ est *régulier*.

Définition 4.3 [LFA 75] Une représentation paramétrique f de classe C^p d'un arc de courbe γ est normale si :

$$\|f'(t)\| = 1, \forall t \in I.$$

La paramétrisation f est alors régulière.

Remarque 4.4 : inversement on peut montrer ([LFA 75]) que tout arc régulier admet des paramétrisations normales.

Remarque 4.5 : les représentations paramétriques normales sont celles définies à partir de l'abscisse curviligne, le sens de parcours de la courbe et l'origine étant quelconques.

On pourra donc dire de façon équivalente paramétrisation normale ou paramétrisation suivant l'abscisse curviligne.

Rappelons maintenant la définition d'un changement de paramétrisation :

Définition 4.6 [LFA 75] Soit une représentation paramétrique c de classe C^p d'une courbe γ :

$$c : I \longrightarrow E.$$

On appelle **changement de paramétrisation admissible de classe C^p** toute application f de J (intervalle d'intérieur non-vide de \mathbb{R}) dans I vérifiant :

- (i) f est une bijection de J sur I .
- (ii) f et f^{-1} sont C^p sur respectivement J et I .

En reprenant les notations de la précédente définition, on a :

Propriété 4.7 Si c est régulière, alors la nouvelle paramétrisation :

$$J \longrightarrow \gamma, \quad t \longmapsto c(f(t))$$

est également régulière

Preuve : [LFA 75].

Nous allons maintenant définir pour nos besoins futurs une relation d'ordre sur les points d'un arc de courbe :

Définition 4.8 Soit γ un arc de courbe régulier et C^p défini par une paramétrisation régulière c par rapport au paramètre θ . Soient M_1 et M_2 deux points de γ de paramètres respectifs θ_1 et θ_2 .

On définit alors la relation **prédécesseur** \preceq sur γ par :

$$M_1 (= c(\theta_1)) \preceq M_2 (= c(\theta_2))$$

$$\text{si et seulement si : } \theta_1 \leq \theta_2$$

De plus, \preceq est une relation d'ordre sur γ .

Il est facile de montrer qu'il s'agit bien d'une relation d'ordre et que deux relations trivialement dépendantes peuvent être déduites selon le sens de parcours de la courbe.

Nous allons enfin définir la courbure pour un arc régulier ([LFA 75]) :

Soit γ un arc de courbe régulier de classe C^p , $p \geq 2$ de représentation paramétrique normale $c : I \rightarrow E$. Alors :

Définition 4.9 La fonction vectorielle

$$\vec{\tau} : I \rightarrow E, s \mapsto c'(s)$$

sera appelée la **fonction vecteur unitaire tangent** à γ .

D'autre part, la fonction numérique ρ , définie sur I par :

$$\rho(s) = \|c''(s)\|$$

sera appelée la **fonction courbure** de γ .

On pourra alors définir un **point d'inflexion** de γ comme un point en lequel la courbure est nulle.

Enfin, si l'on suppose qu'un point $M = c(s)$ de γ n'est pas un point d'inflexion, on peut alors définir, en conservant les notations précédentes :

Définition 4.10 [LFA 75] Si $c''(s) \neq 0$, le nombre $R(s) = 1/\rho(s)$ est appelé le **rayon de courbure** de γ en M .

D'autre part, le **centre de courbure** $\Omega(s)$ à γ en $M = c(s)$ est défini par :

$$\begin{aligned} \langle c'(s), c(s) - \Omega(s) \rangle &= 0 \\ \text{et } \langle c''(s), c(s) - \Omega(s) \rangle &= -1 \end{aligned}$$

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire usuel de E .

$\Omega(s)$ et $R(s)$ sont respectivement le centre et le rayon du **cercle osculateur** à γ en M .

4.2 Définition "limite" du cercle osculateur

Il est possible de définir le cercle osculateur à un arc de courbe en un point par un passage à la "limite" ("limite" restant une notion à définir) sur des cercles passant par des points de cet arc. On peut trouver dans l'ouvrage de Lelong-Ferrand une interprétation géométrique de ce cercle qui va en ce sens.

Interprétation géométrique du cercle osculateur ([LFA 75])

Si l'on se donne un arc régulier γ de classe C^2 et M_0 un point de cet arc, il est alors possible d'énoncer que :

Pour tout point M de γ suffisamment proche de M_0 , il existe un cercle unique passant par M et tangent à γ en M_0 ; et ce cercle admet pour limite², lorsque M tend vers M_0 , le cercle osculateur à γ en M_0 .

Cette interprétation inclue le fait que le cercle précédemment défini est tangent à l'arc de cercle en un point. Cette notion de tangence, que nous ne retrouvons pas dans la formulation de notre problème — nous entendons par là au sens du meilleur cercle approximant —, nous amène à étudier la position "limite" d'un

² ce mot "limite" peut être justifié par l'introduction d'une topologie convenable sur les cercles de \mathbb{R}^2

cercle, ce cercle (unique) passant par trois points distincts qui tendent vers un seul point (que nous pourrions qualifier de point limite).

Nous énonçons alors en ce sens le :

Théorème 4.11 Soit $c : [a, b] \rightarrow \mathbb{R}^2$ une courbe régulière C^2 paramétrée suivant l'abscisse curviligne.

Pour tout s de $]a, b[$ tel que $c''(s) \neq 0$, il existe un voisinage V de s dans $]a, b[$ tel que :

$\forall s_1, s_2, s_3 \in]a, b[$ et tous différents, $c(s_1), c(s_2), c(s_3)$ n'appartiennent pas à une même droite, et l'unique cercle passant par $c(s_1), c(s_2), c(s_3)$ de rayon $R(s_1, s_2, s_3)$ et de centre $\Omega(s_1, s_2, s_3)$ tend vers le cercle osculateur à la courbe c en $c(s)$ au sens où :

$$R(s_1, s_2, s_3) \rightarrow R, \text{ lorsque } s_1, s_2, s_3 \rightarrow s$$

$$\text{et } \Omega(s_1, s_2, s_3) \rightarrow \Omega, \text{ lorsque } s_1, s_2, s_3 \rightarrow s,$$

où R et Ω désignent le rayon et le centre du cercle osculateur à c en $c(s)$.

Preuve : Supposons que pour tout voisinage $V =]s - \varepsilon, s + \varepsilon[$ de s , il existe $s_1 < s_2 < s_3$ différents tels que $c(s_1), c(s_2), c(s_3)$ soient alignés et vérifient alors :

$$\langle c(s_i), u_\varepsilon \rangle = \gamma_\varepsilon, \quad i = 1, 2, 3.$$

Il existe donc $\xi_1 \in]s_1, s_2[$ et $\xi_2 \in]s_2, s_3[$ tels que :

$$\langle c'(\xi_i), u_\varepsilon \rangle = 0, \quad i = 1, 2,$$

et donc $\eta \in]\xi_1, \xi_2[$ tel que :

$$\langle c''(\eta), u_\varepsilon \rangle = 0.$$

Si v_ε est un vecteur unitaire orthogonal à u_ε , on a donc :

$$c'(\xi_1) = \pm v_\varepsilon$$

$$\text{et } c'(\xi_2) = \pm v_\varepsilon$$

$$\text{et } c''(\eta) = \lambda_\varepsilon v_\varepsilon.$$

En prenant $\xi = \xi_1$ ou ξ_2 on a :

$$| \langle c'(\xi), c''(\eta) \rangle | = |\lambda_\varepsilon|,$$

et donc

$$\lim_{\xi, \eta \rightarrow 0} \langle c'(\xi), c''(\eta) \rangle = \langle c'(s), c''(s) \rangle = 0$$

car $\|c'(s)\|^2 = 1$, d'où :

$$\lim_{\eta \rightarrow 0} \lambda_\varepsilon v_\varepsilon = 0$$

et donc $c''(s) = 0$, ce qui est contradictoire.

Le cercle osculateur à c en $c(s)$ passe par $c(s)$ et son centre est défini par :

$$\langle c'(s), c(s) - \Omega \rangle = 0$$

$$\text{et } \langle c''(s), c(s) - \Omega \rangle = -1.$$

Choisissons alors s_1, s_2, s_3 dans le voisinage V introduit ci-dessus et considérons le cercle passant par $c(s_1), c(s_2)$ et $c(s_3)$.

Considérons maintenant la fonction

$$t \longmapsto \varphi(t) = \langle c(t) - \Omega(s_1, s_2, s_3), c(t) - \Omega(s_1, s_2, s_3) \rangle,$$

où $\Omega(s_1, s_2, s_3)$ désigne le centre de ce cercle.

Aux points s_1, s_2 et s_3 , la valeur de cette fonction est égale au carré du rayon du cercle passant par les trois points. Il existe donc $\xi_1 \in]s_1, s_2[$ et $\xi_2 \in]s_2, s_3[$ tels que :

$$\varphi'(\xi_1) = \varphi'(\xi_2) = 0$$

et donc :

$$\begin{cases} \langle c'(\xi_1), c(\xi_1) - \Omega(s_1, s_2, s_3) \rangle = 0 \\ \langle c'(\xi_2), c(\xi_2) - \Omega(s_1, s_2, s_3) \rangle = 0. \end{cases}$$

On en déduit donc qu'il existe $\eta \in]\xi_1, \xi_2[$ tel que :

$$\varphi''(\eta) = 0$$

$$\iff \langle c''(\eta), c(\eta) - \Omega(s_1, s_2, s_3) \rangle + \langle c'(\eta), c'(\eta) \rangle = 0$$

$$\iff \langle c''(\eta), c(\eta) - \Omega(s_1, s_2, s_3) \rangle = - \langle c'(\eta), c'(\eta) \rangle = -1.$$

Etudions maintenant les quantités $\langle c'(s), c(s) - \Omega(s_1, s_2, s_3) \rangle$ et $\langle c''(s), c(s) - \Omega(s_1, s_2, s_3) \rangle$.

Pour $\xi = \xi_1$ ou ξ_2 (indifféremment), nous avons :

$$\begin{aligned} & \langle c'(s), c(s) - \Omega(s_1, s_2, s_3) \rangle \\ &= \langle c'(s) - c'(\xi) + c'(\xi), c(s) - c(\xi) + c(\xi) - \Omega(s_1, s_2, s_3) \rangle \\ &= \langle c'(s) - c'(\xi), c(\xi) - \Omega(s_1, s_2, s_3) \rangle + \langle c'(s) - c'(\xi), c(s) - c(\xi) \rangle \\ & \quad + \langle c'(\xi), c(s) - c(\xi) \rangle + \langle c'(\xi), c(\xi) - \Omega(s_1, s_2, s_3) \rangle . \end{aligned}$$

Il est clair que :

$$\begin{aligned} & \langle c'(\xi), c(\xi) - \Omega(s_1, s_2, s_3) \rangle = 0, \\ & \langle c'(\xi), c(s) - c(\xi) \rangle \longrightarrow 0, \text{ lorsque } s_1, s_2, s_3 \rightarrow s, \\ & \langle c'(s) - c'(\xi), c(s) - c(\xi) \rangle \longrightarrow 0, \text{ lorsque } s_1, s_2, s_3 \rightarrow s. \end{aligned}$$

D'autre part,

$$\langle c'(s) - c'(\xi), c(\xi) - \Omega(s_1, s_2, s_3) \rangle \longrightarrow 0, \text{ lorsque } s_1, s_2, s_3 \rightarrow s$$

si $\Omega(s_1, s_2, s_3)$ reste borné lorsque $s_1, s_2, s_3 \rightarrow s$. Or, cette condition est assurée puisque $\langle c''(\eta), c(\eta) - \Omega(s_1, s_2, s_3) \rangle = -1$ et $c''(s) \neq 0$.

On obtient donc :

$$\langle c'(s), c(s) - \Omega(s_1, s_2, s_3) \rangle \longrightarrow 0, \text{ lorsque } s_1, s_2, s_3 \rightarrow s.$$

De même, étudions :

$$\begin{aligned} & \langle c''(s), c(s) - \Omega(s_1, s_2, s_3) \rangle \\ &= \langle c''(s) - c''(\eta) + c''(\eta), c(s) - c(\eta) + c(\eta) - \Omega(s_1, s_2, s_3) \rangle \\ &= \langle c''(s) - c''(\eta), c(\eta) - \Omega(s_1, s_2, s_3) \rangle + \langle c''(s) - c''(\eta), c(s) - c(\eta) \rangle \\ & \quad + \langle c''(\eta), c(s) - c(\eta) \rangle + \langle c''(\eta), c(\eta) - \Omega(s_1, s_2, s_3) \rangle . \end{aligned}$$

Un raisonnement analogue au précédent montre que :

$$\langle c''(s), c(s) - \Omega(s_1, s_2, s_3) \rangle \longrightarrow 0, \text{ lorsque } s_1, s_2, s_3 \rightarrow s.$$

En se plaçant dans le repère $(c(s), \bar{c}'(s), \bar{c}''(s))$, les deux relations nous prouvent que :

$$\Omega(s_1, s_2, s_3) \longrightarrow \Omega, \text{ lorsque } s_1, s_2, s_3 \rightarrow s.$$

Il est donc facile de montrer que le rayon du cercle de centre $\Omega(s_1, s_2, s_3)$ et passant par $c(s_1), c(s_2)$ et $c(s_3)$ tend vers le rayon du cercle osculateur.

4.3 Etude expérimentale de la convergence du cercle approximant

Les principales motivations du caractère expérimental de cette étude ont été qu'il est difficile d'évaluer le cercle osculateur d'après seulement trois points, voire impossible dans le cas où ceux-ci sont bruités. Il est alors nécessaire d'utiliser un procédé plus adapté, et le cercle approximant nous donnera alors évidemment une meilleure estimation du rayon de courbure.

D'autre part, le résultat précédent est difficile à montrer à partir du meilleur cercle approximant (une expression formelle de la solution du problème de minimisation est très compliquée et inexploitable), et il est difficile numériquement de n'utiliser que trois points.

Il serait intéressant de faire la démonstration mathématique de cette convergence et encore plus, et d'obtenir une preuve apportant des résultats et des ordres de grandeur quantitatifs sur la qualité de l'approximation, mais ceci semble difficile ; il est tout de même possible de constater numériquement une telle convergence.

Nous avons donc effectué de nombreuses expériences pour vérifier numériquement la convergence du cercle approximant vers le cercle osculateur en un point d'un arc de courbe. Ces expériences ont toutes été basées sur le calcul du cercle approximant en prenant des points sur un arc de courbe dans un intervalle de plus en plus réduit.

Nous avons ainsi accompli des tests en prenant une densité de points constante ou non sur l'arc de courbe, et aussi avec des données bruitées.

Exemples

Nous donnons ci-dessous des exemples des ces expérimentations. Les points $M(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 9/20 + (r \cos(t)/6) \\ 1/2 + (r \sin(t)/6) \end{pmatrix}$, avec $r = 2 + 6/10 \cos(5t) + 1/10 \cos(7t) + 2/10 \cos(11t) + 1/10 \cos(17t)$ sont pris autour du point $M(t_0)$ en lequel est calculé le rayon de courbure, qui a pour valeur $0,021270$ à 10^{-6} près en ce point. Le premier graphique présente les résultats de deux tests effectués avec un nombre de points différent, et donne la valeur du rayon de cercle approximant en fonction de l'amplitude de l'intervalle autour du point $M(t_0)$ en lequel est calculé le cercle osculateur. L'erreur commise (à 10^{-6} près) figure aussi aux côtés des

valeurs des rayons.

Le deuxième graphique représente le même test, mais avec des données bruitées.

Enfin, sont représentés sur le troisième graphe les résultats obtenus avec une densité de points constante sur les intervalles considérés autour du point en lequel est calculé le cercle osculateur. Nous avons pris pour cela, par rapport au paramétrage, un point tous les 10^{-4} .

Débordements numériques

L' "explosion" numérique de la valeur du rayon sur les derniers intervalles des deux premiers jeux d'essais est due au fait que les points de la courbe se retrouvent "numériquement" sur une droite ; et ceci explique ce phénomène, si l'on considère une droite comme un cercle de rayon infini.

Nous avons ainsi pu constater au travers des différentes expérimentations que le meilleur cercle approximant convergeait numériquement assez bien vers le cercle osculateur et nous fournissait donc une assez bonne approximation de ce dernier.

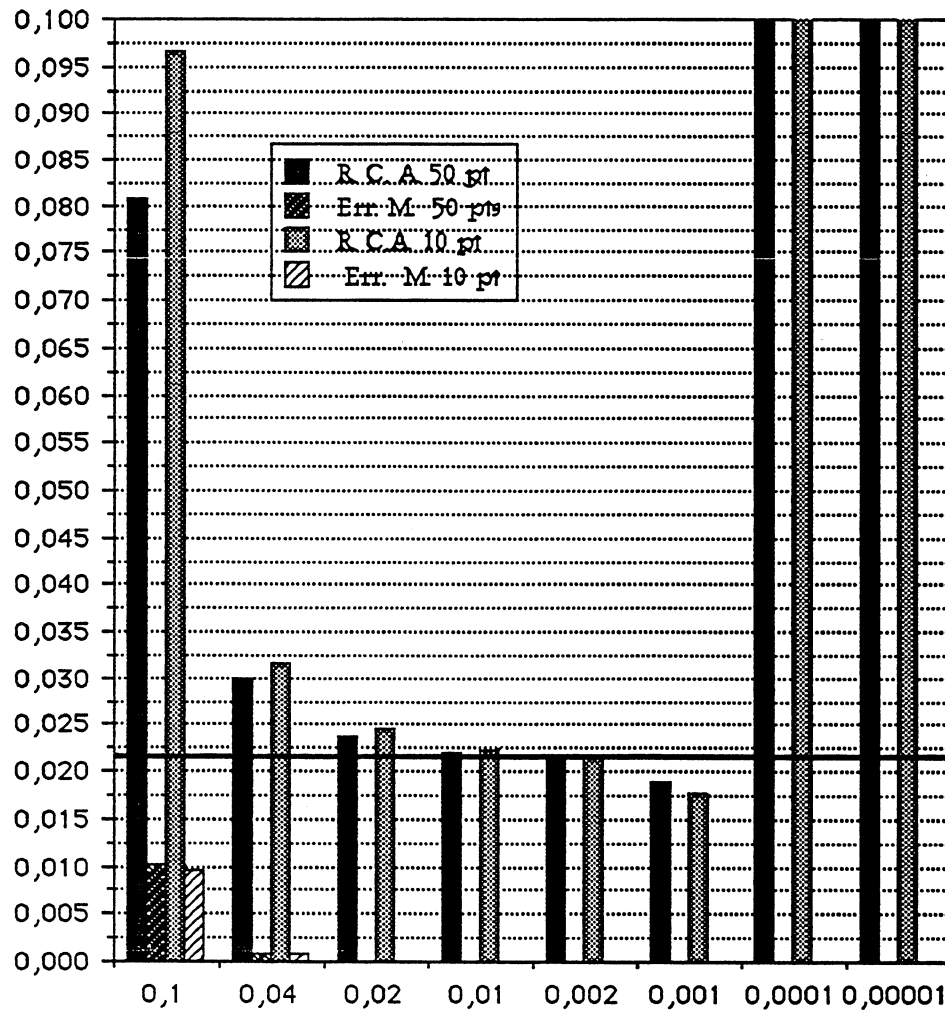


Figure 4.11 Valeurs du rayon du cercle approximant pour 10 et 50 points $M(t)$, valeurs de t uniformément réparties dans $[t_0 - \epsilon, t_0 + \epsilon]$.

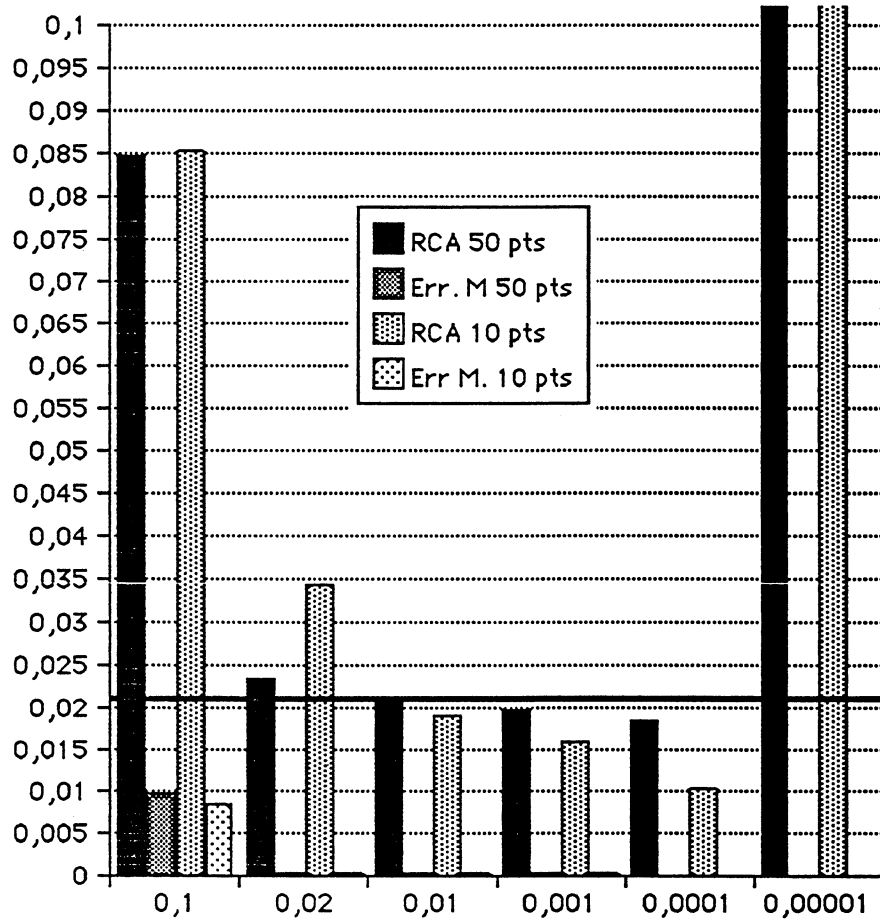


Figure 4.12 Valeurs du rayon du cercle approximant pour 10 et 50 points bruités $M(t)$, valeurs de t uniformément réparties dans $[t_0 - \epsilon, t_0 + \epsilon]$.

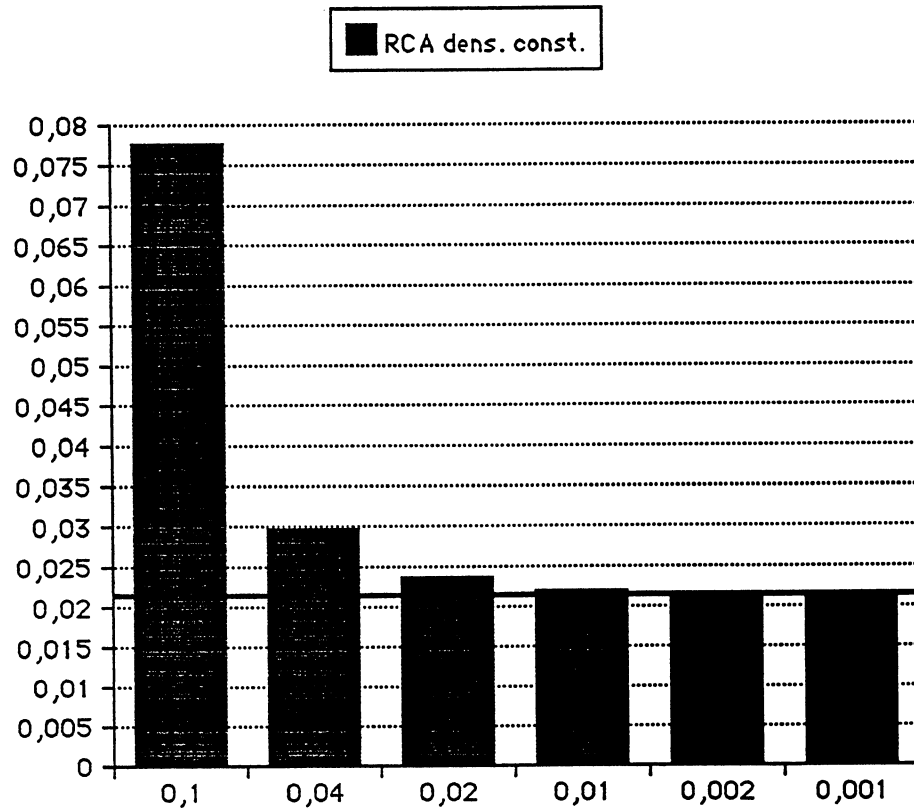


Figure 4.13 Valeurs du rayon du cercle approximant pour un ensemble de points $M(t)$, valeurs de t uniformément réparties à densité constante dans $[t_0 - \epsilon, t_0 + \epsilon]$.

Les figures des pages suivantes représentent les tracés des cercles approximants calculés, correspondants aux intervalles d'amplitude ϵ des graphiques (avant l' "explosion" numérique); le point en lequel est calculé le rayon de courbure est repéré par une flèche.

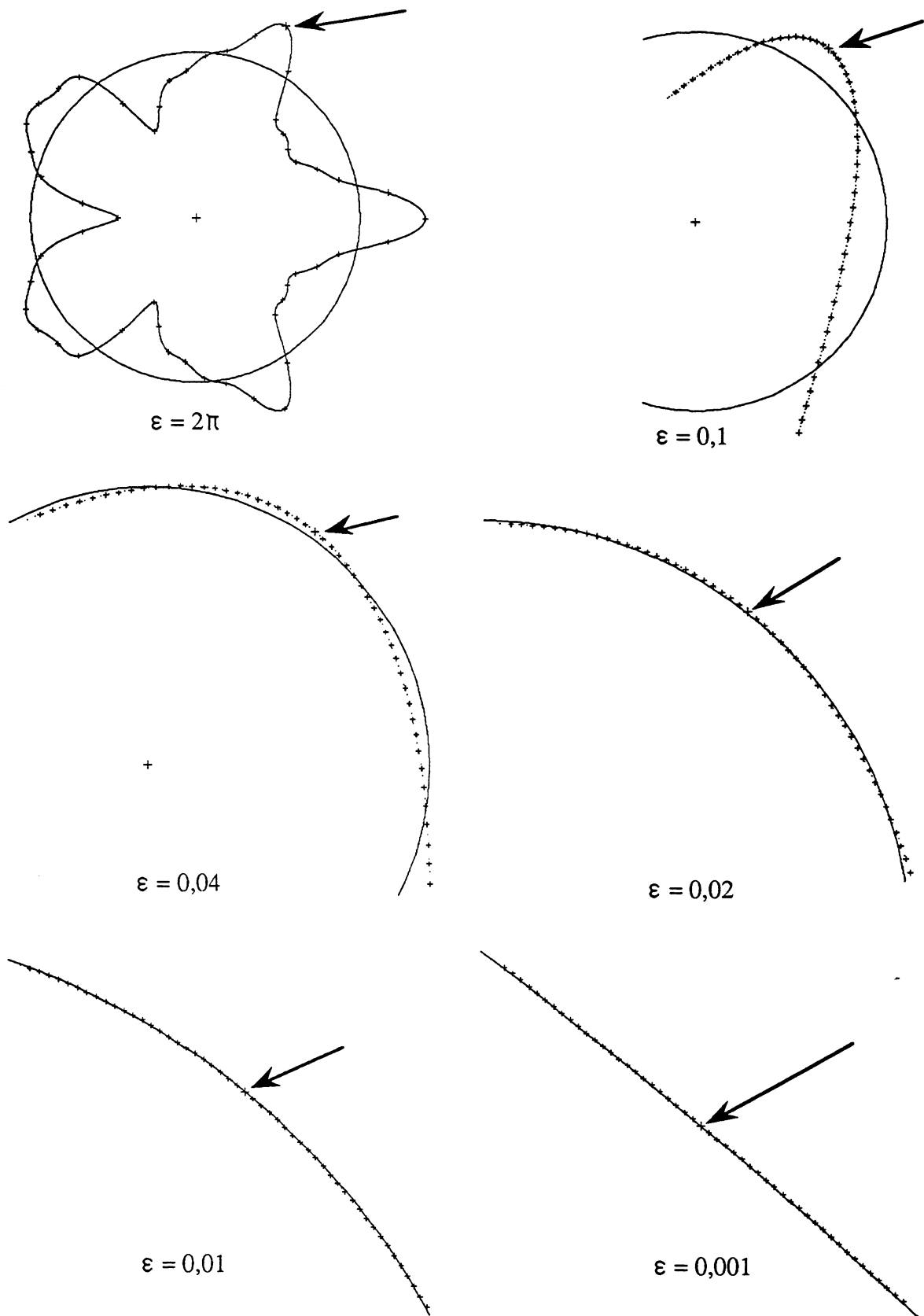


Figure 4.14 Illustration de la convergence pour 50 points $M(t)$, valeurs de t uniformément réparties dans $[t_0 - \varepsilon, t_0 + \varepsilon]$.

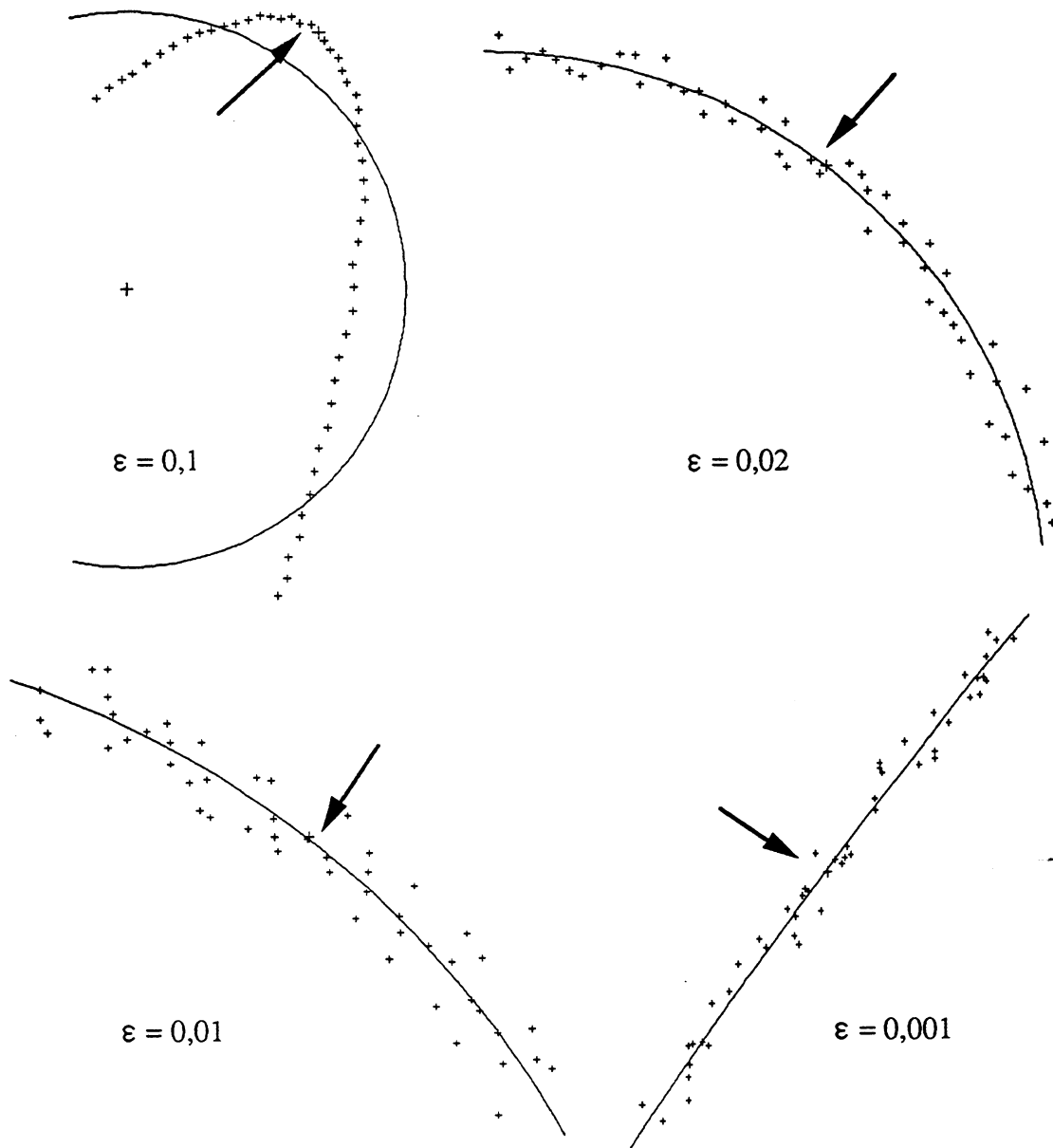


Figure 4.15 Illustration de la convergence pour 50 points bruités $M(t)$, valeurs de t uniformément réparties dans $[t_0 - \epsilon, t_0 + \epsilon]$.

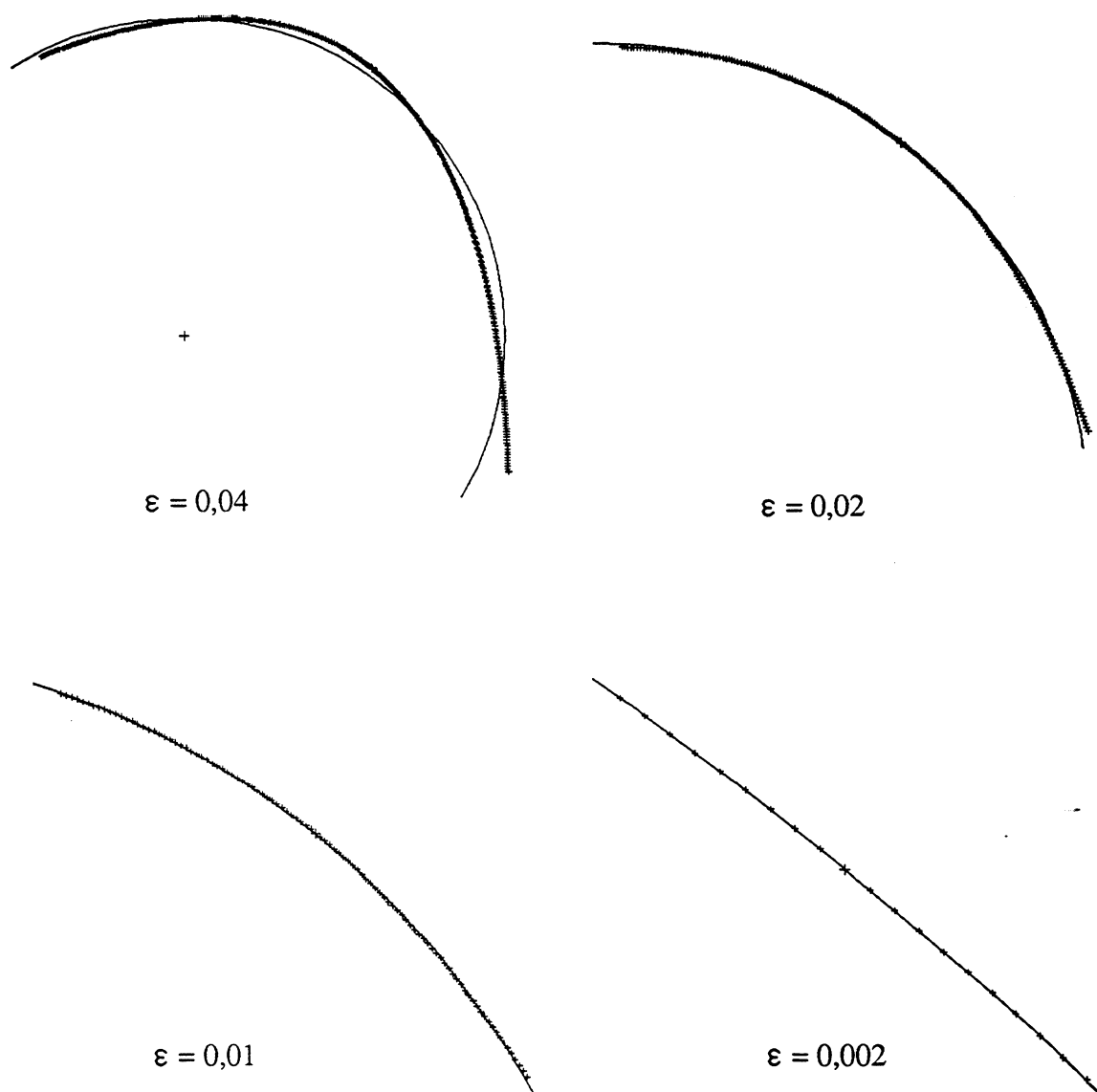


Figure 4.16 Illustration de la convergence pour un ensemble de points $M(t)$, valeurs de t uniformément réparties à densité constante dans $[t_0 - \epsilon, t_0 + \epsilon]$.

4.4 Projection de points et conservation de l'ordre

Comme nous le verrons par la suite, la méthode que nous employons pour reconstruire la courbe nécessite la projection de sous-ensembles de points sur un cercle, dans le but de les ordonner, et d'obtenir ainsi un ensemble d'arcs de cette courbe.

En considérant, dans un premier temps, non pas le meilleur cercle approximant obtenu par la méthode précédemment décrite au chapitre I, mais le cercle osculateur en un point de cette courbe, la question que nous allons examiner dans ce paragraphe est la suivante :

Quelles sont les conditions qui garantissent le fait que, une fois les points d'un arc de courbe projetés sur ce cercle, on puisse retrouver sur le cercle le même ordre des points que celui induit par un paramétrage de la courbe ?

4.4.1 Fonctions angulaires associées à un arc plan

Nous allons étudier ici deux fonctions angulaires associées à un arc plan.

Nous énonçons tout d'abord le théorème suivant (cf. [LFA 75]) :

Théorème 4.12 *Relèvement d'une application continue de module*

1

Soit $I \subset \mathbb{R}, I \neq \emptyset$ et $F : I \rightarrow \mathbb{C}$ une fonction complexe continue sur I vérifiant :

$$(1) \quad \forall t \in I, \quad |F(t)| = 1$$

i) Alors il existe une fonction $\varphi : I \rightarrow \mathbb{R}$ vérifiant :

$$(2) \quad \forall t \in I, \quad \exp(i\varphi(t)) = F(t)$$

ii) Pour que $\psi : I \rightarrow \mathbb{R}$ continue vérifie $F(t) = \exp(i\psi(t)), \forall t \in I$, il faut et il suffit qu'il existe un entier relatif k tel que $\psi(t) = \varphi(t) + 2k\pi, \forall t \in I$.

iii) Si F est C^1 au moins, toute fonction continue φ vérifiant (2) est de même classe que F et vérifie $i\varphi' = \frac{F'}{F}$.

Preuve : [LFA 75].

Soit γ un arc C^1 de E défini par une paramétrisation normale $g : I \rightarrow \mathcal{E}_2$.
Soit $\mathcal{R} = (O, \vec{i}, \vec{j})$ un repère orthonormal direct de \mathcal{E}_2 .

Pour chaque $s \in I$, on peut définir le vecteur tangent $\vec{\tau}(s)$ par une détermination quelconque de l'angle orienté $(\vec{i}, \vec{\tau}(s))$.

Le problème consiste alors à voir si il existe une fonction continue $\varphi : I \rightarrow \mathbb{R}$ telle que $\forall s \in I, (\vec{i}, \vec{\tau}(s)) = \varphi(s) \pmod{2\pi}$.

Si $g(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}$ dans \mathcal{R} , ceci revient à déterminer φ continue sur I vérifiant :

$$\forall s \in I, \cos(\varphi(s)) = x'(s), \sin(\varphi(s)) = y'(s)$$

soit :

$$\forall s \in I, e^{i\varphi(s)} = x'(s) + iy'(s).$$

Par rapport au problème précédent, on peut maintenant énoncer :

Proposition 4.13 [LFA 75] **Angle de la tangente avec un axe fixe**

Soit γ un arc régulier orienté C^k ($k \geq 1$) de \mathcal{E}_2 , défini par une paramétrisation normale $g : I \rightarrow \mathcal{E}_2$; soit (\vec{i}, \vec{j}) une base orthonormale directe de E_2 .

Il existe alors une fonction φ , unique modulo 2π , continue sur I et telle que $\forall s \in I$ on ait :

$$\vec{\tau}(s) : \vec{i} \cos(\varphi(s)) + \vec{j} \sin(\varphi(s))$$

ce qui équivaut à :

$$(\vec{i}, \vec{\tau}(s)) = \varphi(s) \pmod{2\pi}.$$

De plus, φ est C^{k-1} .

Preuve : On choisit une origine O , et soient $x(s)$ et $y(s)$ les coordonnées de $g(s)$ dans le repère (O, \vec{i}, \vec{j}) . Il suffit alors d'appliquer le théorème 4.12 à la fonction $s \mapsto F(s) = x'(s) + iy'(s)$. \square

On dira que φ est la **fonction angulaire** associée à γ , ou encore que c'est une détermination continue de l'angle de la tangente à γ avec l'axe (Ox) .

On peut alors exprimer la courbure algébrique en fonction de φ :

$$\forall s \in I, \rho(s) = \varphi'(s)$$

On en déduit :

Théorème 4.14 [LFA 75] *Soit γ un arc régulier de classe C^k de E défini par une paramétrisation normale g de classe C^{k+1} .*

Pour que la fonction angulaire associée φ définisse un changement de paramètre admissible sur γ , il faut et il suffit que γ soit sans point d'inflexion.

Preuve : Si g est C^{k+1} , φ est C^k et elle définit un changement de paramètre admissible si et seulement si sa dérivée ne s'annule pas. \square

4.4.2 Conservation de l'ordre par projection radiale

La première condition que nous pouvons énoncer pour conserver l'ordre par projection est naturelle :

Condition 4.15 *La variation de l'angle de la tangente aux points de l'arc ne doit pas excéder 2π .*

Nous allons nous placer dans toute la suite sous cette première condition.

Soit γ un arc de courbe C^1 et une paramétrisation régulière $c : I \rightarrow \mathbb{R}$.

Soient d'autre part $M_0 = c(t_0) \in \gamma$ et $\Omega \neq M_0$ tels que

$$\overrightarrow{\Omega_0 M_0} \text{ et } c'(t_0) \text{ ne soient pas colinéaires.}$$

Soit \mathcal{C} un cercle (de rayon non nul) de centre Ω paramétré régulièrement par rapport à l'angle polaire.

Pour tout point M de la courbe γ , on fait correspondre le point $m = \psi(M)$ intersection de $\overrightarrow{\Omega M}$ avec \mathcal{C} .

Nous allons maintenant énoncer la propriété clef de la conservation de l'ordre :

Propriété 4.16 *Il existe un intervalle $I_0 \subset I$, d'intérieur non-vidé et contenant t_0 , tel que $\psi(c(t))$ est bien défini pour tout $t \in I_0$, et est une bijection de $c(I_0) = \gamma_0$ sur $(\psi(c(I_0)))$.*

De plus, on peut choisir un paramétrage par angles polaires de $\psi(I_0)$ tel que l'ordre de deux points $M_1 = c(t_1)$ et $M_2 = c(t_2)$ sur γ_0 soit identique à l'ordre des points $\psi(M_1)$ et $\psi(M_2)$ sur $\psi(\gamma_0)$.

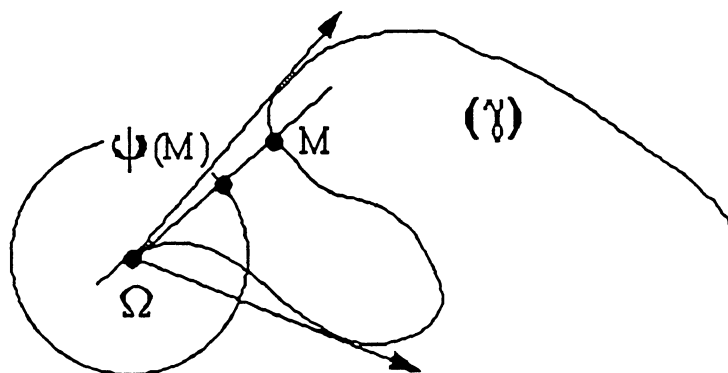


Figure 4.17 Conservation de l'ordre par projection radiale

Preuve : Pour simplifier la démonstration, nous allons prendre Ω comme l'origine de l'espace, et \mathcal{C} le cercle unité.

A $c : I \rightarrow \mathbb{R}$, défini par : $c(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$, on associe la fonction de I dans \mathbb{C} définie par :

$$F(t) = \frac{x(t) + i y(t)}{\sqrt{x^2(t) + y^2(t)}} = \frac{z(t)}{\sqrt{\bar{z}(t)z(t)}}.$$

$c(t_0)$ étant différent de Ω (origine), on peut définir F pour l'intervalle maximum I_0^{max} contenant t_0 et tel que :

$$c(t) \neq 0, t \in I_0^{max}.$$

D'après le théorème 4.12, il existe donc

$$\varphi : I_0^{max} \rightarrow \mathbb{R}$$

telle que :

$$\exp(i \varphi(t)) = F(t), t \in I_0.$$

Comme F est de classe C^1 , il en résulte la même régularité sur φ , et :

$$i \varphi'(t) = \frac{F'(t)}{F(t)}.$$

Calculons donc $\varphi'(t)$:

$$\begin{aligned}
 F'(t) &= \frac{z'(t)}{\sqrt{\bar{z}(t)z(t)}} - \frac{z(t)[z'(t)z(t) + \bar{z}(t)\bar{z}'(t)]}{2(\bar{z}(t)z(t))^{3/2}} \\
 &= \frac{2z'(t)\bar{z}(t)z(t) - z(t)\bar{z}'(t)z(t) - \bar{z}(t) - \bar{z}(t)z'(t)z(t)}{2[\bar{z}(t)z(t)]^{3/2}} \\
 &= \frac{z'(t)\bar{z}(t)z(t) - z(t)\bar{z}'(t)z(t) - z(t)\bar{z}'(t)z(t)}{2[\bar{z}(t)z(t)]^{3/2}} \\
 &= \frac{z(t)[z'(t)\bar{z}'(t) - z(t)\bar{z}'(t)]}{2[\bar{z}(t)z(t)]^{3/2}}
 \end{aligned}$$

d'où

$$1/i \frac{F'(t)}{F(t)} = 1/i \frac{z'(t)\bar{z}(t) - z(t)\bar{z}'(t)}{2\bar{z}(t)z(t)} = \frac{\Im(z'(t)\bar{z}(t))}{\bar{z}(t)z(t)}.$$

Et ainsi,

$$\varphi'(t) = \frac{y'(t)x(t) - x'(t)y(t)}{x^2(t) + y^2(t)}.$$

d'après les hypothèses, $\varphi'(t_0) \neq 0$, et on peut donc trouver un intervalle I_0 autour de t_0 tel que $\psi(c(t))$ soit défini, $\forall t \in I_0$ d'une part et $\varphi'(t) \neq 0$, $\forall t \in I$ d'autre part.

Il est alors clair que le point $\psi(c(t))$ correspond au point $\exp(i\varphi(t))$, et que l'image de $\{c(t), t \in I\}$ est un arc de cercle unité, et dont $\varphi(t), t \in I$ constitue un changement de paramétrage régulier C^1 du paramétrage de l'arc de cercle par l'angle polaire. \square

La conservation de l'ordre tel qu'il a été défini est alors une conséquence directe de cette interprétation.

Remarque 4.17 : Le résultat est intuitivement "évident". En effet, à partir de la valeur t_0 pour laquelle $\overrightarrow{\Omega M_0} \neq \vec{0}$ et $\det(\overrightarrow{\Omega M_0}, c'(t_0)) \neq 0$, on peut par continuité définir correctement la projection sur \mathcal{C} tant que $\overrightarrow{\Omega M} \neq \vec{0}$ et $\det(\overrightarrow{\Omega M}, c'(t_0)) \neq 0$ autour de t_0 .

Ceci peut s'énoncer en les termes suivants :

Condition 4.18 $\forall M \in \gamma, \Omega$ ne doit pas appartenir à la tangente à γ au point M .

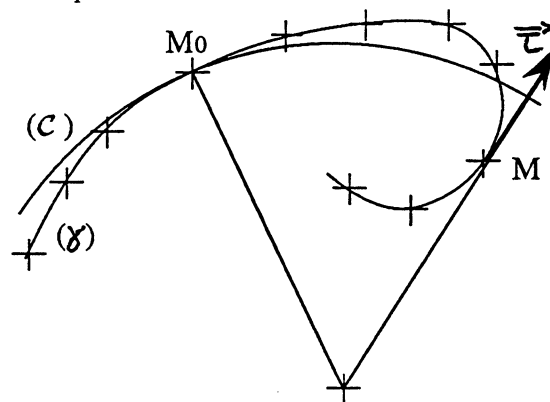


Figure 4.18 Conservation de l'ordre et tangente dans le voisinage d'un point

Enfin, ces propriétés restent évidemment vraies pour le cercle osculateur approché que nous obtenons localement par le meilleur cercle approximant un arc de courbe. C'est à ce cercle que ces résultats vont donc s'appliquer en pratique, pour nous garantir localement la conservation de l'ordre des points lorsque nous projeterons ceux-ci sur le cercle pour les trier.

Chapitre II

Ordonnancement de points sur une courbe

Nous allons aborder dans ce chapitre le problème de la reconstruction de la courbe proprement dit. Après avoir présenté les principes de base de la méthode et l'avoir située par rapport aux travaux existants, nous examinerons les différentes étapes des algorithmes permettant de mener à bien l'ordonnancement de points situés sur des courbes simples et sur des courbes comportant des points multiples ou des points de rebroussement.

La complexité de ces algorithmes sera ensuite étudiée.

Nous proposons enfin quelques jeux d'essais, avec les temps d'exécution, et nous illustrerons la robustesse des algorithmes dans les cas de jeux de données bruitées.

1 Méthode : situation dans le contexte existant et présentation

Comme nous l'avons évoqué dans l'introduction de cette partie, plusieurs méthodes existent pour résoudre le problème posé, et fonctionnent sur différents principes.

Deux grands principes sont principalement utilisés.

Méthodes de proximité

Tout d'abord, les solutions proposées et développées par C. Favardin [Fav 93] utilisent la notion de proximité sur les points ou une direction tangente estimée localement qui permet de venir à bout des points multiples.

En effet, la première méthode, itérative, consiste à chaque étape, à rechercher, dans une boule de rayon ε centrée sur une extrémité, le point le plus proche de cette extrémité du sous-ensemble de points déjà ordonné et à ajouter ce point au sous-ensemble. Ce point devient alors une nouvelle extrémité.

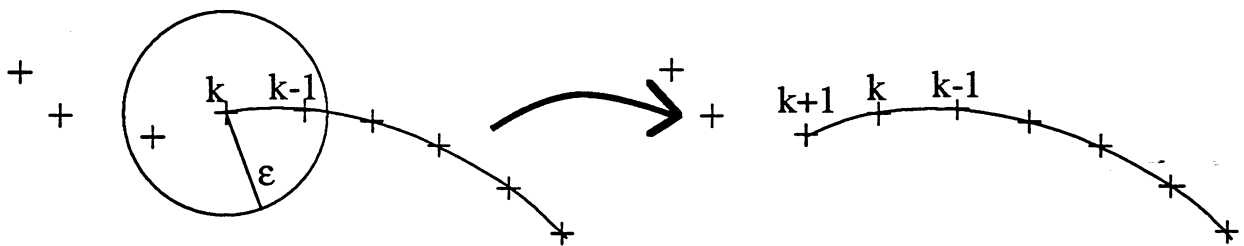


Figure 1.1 $k^{\text{ème}}$ étape de la méthode

Si la courbe comporte des auto-intersections, le seul emploi de la proximité ne suffit plus pour venir à bout de tels cas. En utilisant alors une direction tangente estimée par rapport au sous-ensemble de points déjà traités, Favardin montre qu'il devient possible de privilégier, parmi les points se trouvant à proximité de l'extrémité courante, un point situé le plus près de cette direction tangente.

Dans le cadre de ces méthodes, des résultats théoriques ont été montrés. Par contre, la sensibilité au bruit est forcément assez grande.

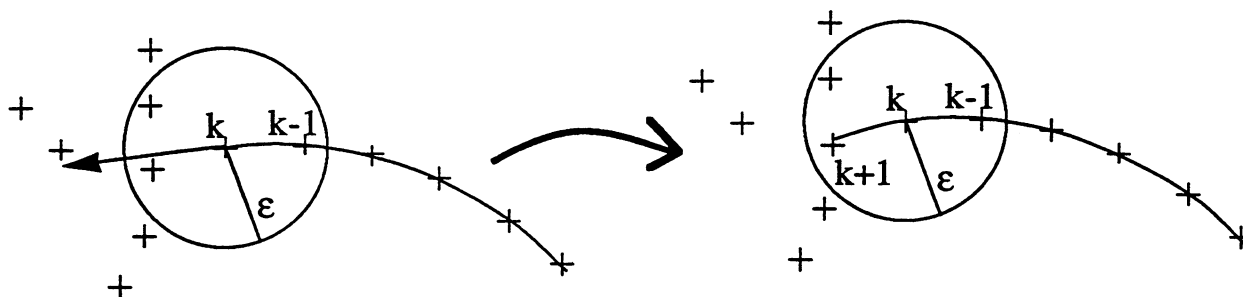


Figure 1.2 $k^{\text{ème}}$ étape de l'utilisation d'une direction tangente

Méthodes globales

D'autres méthodes utilisent une approximation globale de la courbe par une courbe calculée sous sa forme implicite, principalement par une courbe algébrique.

Dedieu et Favardin [DF 91] présentent le problème de la façon suivante.

Pouvons-nous construire une courbe interpolante (ou de lissage) avec seulement la connaissance des coordonnées des points d'un ensemble donné ? Pouvons-nous trouver un "ordre naturel" sur ces points ?

Ils utilisent alors, pour répondre à ces questions, une régression algébrique pour déterminer une courbe algébrique plane de degré fixé, proche des points de l'ensemble selon un sens défini.

Si \mathcal{C} est une courbe algébrique de degré $\leq d$, $\mathcal{C} = \{(x, y) \in \mathbb{R}^2, P_a(x, y) = 0\}$ avec $P_a(x, y) = \sum_{0 \leq \alpha + \beta \leq d} a_{\alpha\beta} x^\alpha y^\beta$, et $a = (a_{\alpha\beta}) \in \mathbb{R}^D$ où $D = \frac{(d+1)(d+2)}{2}$.

En normalisant l'équation par $a \in S_D$, sphère unité de \mathbb{R}^D , ils construisent alors une courbe en utilisant l'un des critères suivants :

- (1) $\min_{a \in S_D} \sum_{k=1}^n P_a(x_k, y_k)^2$
 - (2) $\min_{a \in S_D} \sum_{k=1}^n \left\{ P_a(x_k, y_k)^2 - \mu \left[\frac{\partial P_a(x_k, y_k)^2}{\partial x} + \frac{\partial P_a(x_k, y_k)^2}{\partial y} \right] \right\}$
- avec $\mu > 0$ fixé.

Il a été remarqué que, même en apportant certaines améliorations, l'approximation globale par une courbe implicite présente de nombreuses difficultés.

Une méthode mixte

Nous développons en fait une méthode mixte, basée sur le sous-problème suivant :

l'ordre des points suivant l'abscisse curviligne sur la courbe est le même que celui des points projetés sur le cercle approximant.

Nous utilisons ainsi les deux principes de proximité, et d'approximation locale par des courbes implicites, qui ne présente pas les même inconvénients qu'une approximation globale.

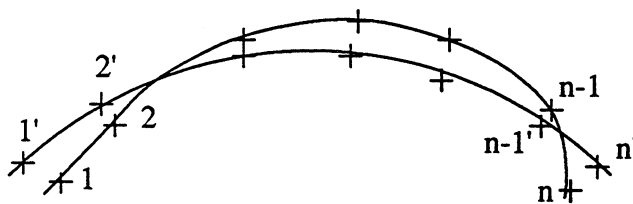


Figure 1.3 conservation de l'ordre par projection

La méthode retenue

Dans le chapitre I , nous avons montré comment obtenir un algorithme robuste pour approcher un ensemble de points par un cercle, et assurer ensuite la conservation de l'ordre par projection sur l'arc de cercle approximant.

L'idée qui vient ensuite en ce qui concerne l'algorithme est donc de subdiviser l'ensemble de points donné (par un quadtree), puis raccorder les sous-ensembles de points ordonnés dans chaque région de la subdivision.

Lorsqu'il existe des points singuliers ou multiples sur la courbe, nous avons

élaboré une stratégie originale d'énumération des différentes morphologies possibles de la courbe dans un voisinage de ces points.

Comme nous le verrons par la suite, c'est cette énumération qui nous permet de venir à bout de tels cas.

Différences par rapport aux autres méthodes

En ce qui concerne la situation de ce travail par rapport aux méthodes existantes, plusieurs différences apparaissent.

C. Favardin avait déjà mis en évidence la nécessité d'avoir recours à une approximation d'ordre 1 pour résoudre le problème posé. Dans la mesure où le cercle approximant converge vers le cercle osculateur, la méthode que nous proposons ici utilise en fait une approximation à l'ordre 2.

D'autre part, l'approximation par des cercles confère à cette méthode un aspect global qui ne se retrouve pas dans d'autres travaux. Nous retrouvons aussi un aspect local, plus proche des méthodes par proximité, provenant du raccordement des morceaux de courbe.

Ce sont ces deux aspects local et global qui induisent le qualificatif "mixte" que nous associons à la méthode présentée ici, et qui contribuent à son originalité.

Inconvénients

Si nous avons pu mettre en évidence certains avantages, la méthode que nous proposons ici possède néanmoins certains inconvénients.

En effet, la justification de la méthode est essentiellement expérimentale, même si nous avons obtenu certains résultats établis au paragraphe 4 du chapitre I qui étayaient l'intuition.

Une justification mathématique semble déjà très difficile dans le cas d'algorithmes utilisant une méthode plus "simple", comme les preuves présentées et développées par C. Favardin.

Il faut d'ailleurs remarquer que les preuves fournies sont des résultats asymptotiques, et l'on constate que les algorithmes sont corrects, alors que, dans la plupart des cas, ces situations limites ne sont pas atteintes.

En ce qui nous concerne, même si nous pouvons prouver, sous certaines hypothèses, la convergence des meilleurs cercles approxima-

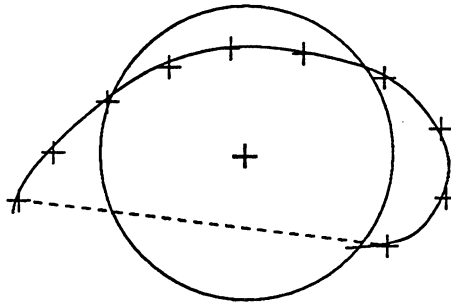


Figure 1.4

lateur en l'un des points de l'ensemble ainsi approché¹, il est clair que l'ordre des points peut être conservé sous des hypothèses beaucoup plus larges et par un résultat non-asymptotique. Si nous sommes en présence, par exemple, d'un arc de courbe convexe, il suffit alors de considérer n'importe quel cercle parmi ceux dont le centre est situé dans l'enveloppe convexe des points de l'arc (cf. figure 1.4). Quant au cas des données perturbées, les démonstrations sont encore plus hypothétiques, voire incertaines.

Pour conclure, des majorations effectives induites par des résultats et des preuves "quantitatives" seraient très intéressantes. En ce sens, le rôle joué par la courbure vis à vis de notre approche semble fondamental. En effet, au niveau de la phase de subdivision, la "profondeur" de découpe paraît en dépendre fortement dans la mesure où nous voulons assurer la conservation de l'ordre ; et ce aussi bien pour la qualité d'approximation par les cercles que pour le morceau de courbe qu'il nous est possible de considérer en se fixant cet objectif.

Avantages

Comme nous le verrons par la suite, cette méthode présente des avantages sur plusieurs points.

Elle nous permet de traiter, en premier lieu, des ensembles comportant de très nombreux points. Nous avons en effet réalisé des jeux d'essais portant sur un million de points en des temps de calcul très raisonnables.

En second lieu, la méthode s'avère robuste si les données sont bruitées, ce qui n'est pas le cas des méthodes de simple proximité. Une certaine robustesse

¹ et assurer ainsi la conservation de l'ordre par projection dans un certain voisinage de ce point.

est également garantie dans les cas de résolution des problèmes liés aux auto-intersections ou aux points de rebroussement, puisque nous traitons de manière globale les différents arcs alors en présence.

2 Algorithmes de reconstruction de courbes.

2.1 Découpe de l'ensemble des points

2.1.1 Le quadtree

Nous utilisons une structure de quadtree pour la découpe de l'ensemble des points en sous-ensembles, à partir desquels nous calculerons des arcs de courbe après l'étape de projection et d'ordonnement.

Rappelons qu'un quadtree est une structure arborescente quaternaire qui décompose un domaine donné en différentes régions (que nous nommerons par la suite cellules).

Chaque cellule représente donc une région du plan définie, dans notre cas, par une frontière fermée carrée. Une cellule contiendra ainsi un certain nombre de points qui seront situés à l'intérieur de cette région.

Une cellule est d'autre part la "fille" d'une autre cellule du quadtree, si sa création provient de la subdivision de cette dernière cellule (appelée cellule "mère"). Enfin, on appelle les "feuilles" du quadtree les cellules qui ne possèdent pas de cellules filles (*i.e.* qui n'ont pas été subdivisées à leur tour).

Le quadtree est ainsi une structure de données hiérarchique (hiérarchie mère/fille), sur laquelle il est possible d'instaurer un ordre sur les cellules feuilles (cf. figure 2.5).

Son principe de construction est basé sur une subdivision récursive.

En effet, une étape de découpe (subdivision) consiste, pour toutes les cellules feuilles de l'arbre existantes à ce moment là, à les subdiviser en quatre. On crée alors, pour chacune de ces feuilles, des cellules filles qui deviennent à leur tour les nouvelles feuilles du quadtree.

La découpe est ainsi calculée sur chaque feuille du quadtree et de manière récursive sur ses cellules filles, tant que l'action de subdivision de la cellule n'est pas stoppée par des conditions préalablement fixées.

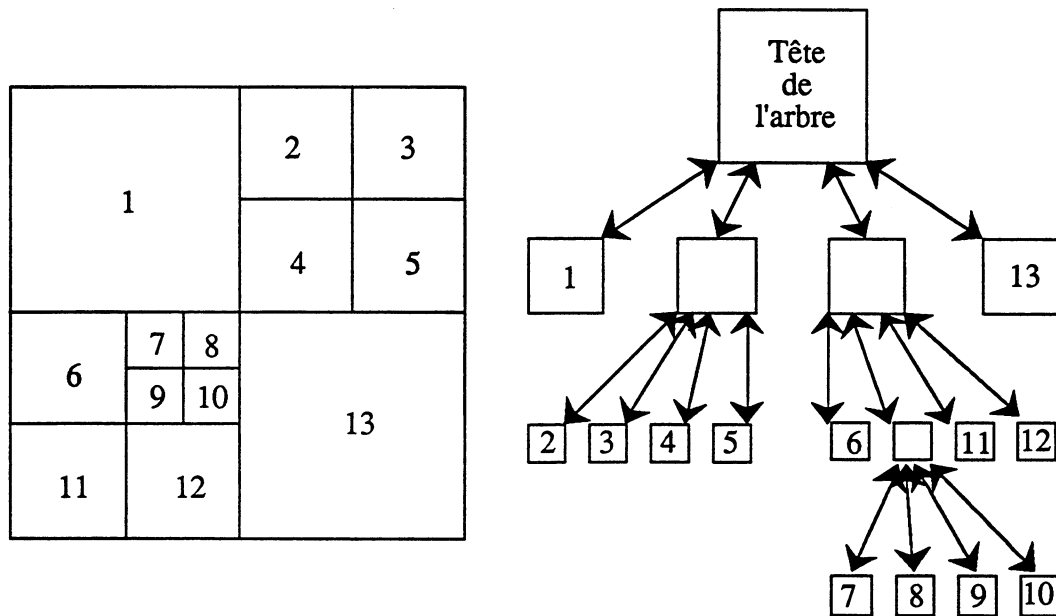


Figure 2.5 Ordre de peano dans le quadtree et son arborescence associée

Pour plus de détails sur les structures de quadtree et les algorithmes s'y rapportant, on pourra consulter par exemple le tour d'horizon de Samet à ce sujet ([Sam 84]).

2.1.2 Critères d'arrêt, méthode de découpe

Nous avons instauré des critères d'arrêt sur la découpe d'une cellule, puisque le principe de la méthode que nous proposons est de traiter des sous-ensembles de points, et contenant le plus de points possible.

Le principe des critères d'arrêt est donc de trouver des conditions qui garantissent le fait d'avoir suffisamment découpé l'ensemble de points initial pour pouvoir mener à bien les opérations de reconstruction locale.

Le premier critère concerne uniquement le nombre de points nécessaire pour calculer un cercle :

critère d'arrêt 2.1 Soit \mathcal{E} un sous-ensemble de points contenu dans une cellule du quadtree.

Alors \mathcal{E} ne sera pas subdivisé si $\text{card}(\mathcal{E}) \leq 3$.

Nous définissons ensuite :

Définition 2.2 Soit \mathcal{E} un sous-ensemble contenant au moins trois points, et \mathcal{C} le cercle approximant \mathcal{E} . On appelle $\Delta\theta$ la variation maximale des angles obtenus en projetant les points de \mathcal{E} sur \mathcal{C} .

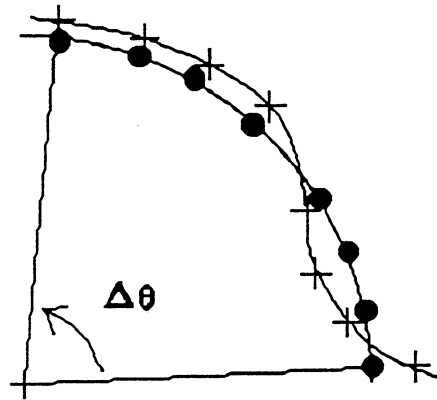


Figure 2.6 variation maximale des angles des points projetés

Définition 2.3 Soit \mathcal{E} un sous-ensemble contenant n points M_i , $3 \leq i \leq n$. On définit l'erreur ε comme la moyenne des distances radiales des points au meilleur cercle approximant, de centre Ω et de rayon R , i.e :

$$\varepsilon = 1/n \sum_{i=1}^n | \|\overline{\Omega M_i}\| - R |.$$

Un deuxième critère concerne alors la qualité d'approximation, en conservant les notations précédentes :

critère d'arrêt 2.4 Soit τ un réel strictement positif fixé.

Alors le processus de découpe sera localement stoppé sur un sous-ensemble de points contenu dans une cellule courante si :

(i) $\varepsilon \leq \tau R$ et $\Delta\theta < \pi$.

ou

(ii) $\varepsilon > \tau R$ pour la cellule considérée et $\varepsilon \leq \tau R$ pour toutes les cellules situées dans le voisinage de la cellule considérée.

Remarque 2.5 : Nous verrons au paragraphe 2.2 la justification de ce critère.

Le quadtree est alors construit récursivement sous ces conditions.

Remarque 2.6 : Nous verrons au paragraphe 2.3 la nécessité d'appliquer la définition 2.2 .

En pratique, la découpe se fait de manière arbitraire quant au positionnement du quadtree par rapport à la courbe. Par des opérations de translation et d'homothétie, il est toujours possible de ramener l'ensemble de points dans le carré unité $[0, 1] \times [0, 1]$; c'est dans ce carré que nous découpons la courbe à cette étape.

Comme nous l'illustrerons par la suite, la position du quadtree par rapport à la courbe n'a, en général, pas d'importance, surtout dans le cas des courbes simples.

Constataion expérimentale

Le point sur lequel nous insisterons est une constatation expérimentale. Il n'y a en effet pas besoin de "beaucoup"² découper la courbe dans la plupart des cas, ce qui est donc intéressant au point de vue complexité et temps de calcul.

Automatisation de la découpe

En fait, le principal problème auquel nous nous sommes heurtés est celui de l'automatisation de cette découpe.

Les natures des courbes sont multiples, et engendrent des configurations de points très diverses. Nous n'avons pas ici conçu une méthode supportant forcément tous les cas avec succès et menant à bien la découpe avec l'objectif de reconstruire correctement la courbe. Nous avons donc seulement aidé ici la découpe en répondant à la question : quand faut-il s'arrêter de découper ?

2.2 Approximation par des cercles dans les cellules du quadtree

Nous allons à cette étape calculer, par la méthode décrite au chapitre I un cercle pour chaque cellule non-vidée et contenant au moins trois points.

² beaucoup s'entend par rapport au nombre de points donnés sur la courbe.

Enfin, chaque cellule pour laquelle un cercle a été calculé va être, selon la qualité de l'approximation, étiquetée selon la propriété suivante (les valeurs ε et τ sont les mêmes que celle du critère d'arrêt 2.4) :

Définition 2.7 Soit ε l'erreur commise en approchant un nuage de points (en contenant au moins trois) par un meilleur cercle approximant. Soient R le rayon du meilleur cercle approximant, et $\tau \in \mathbb{R}$ fixé.

Une cellule sera alors dite **simple** si et seulement si $\varepsilon \leq \tau R$, et **non-simple** dans le cas contraire.

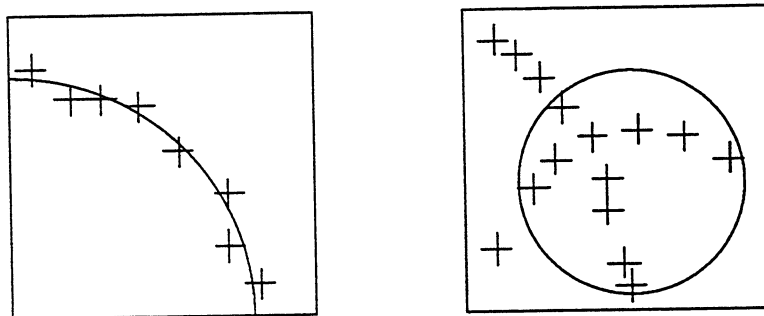


Figure 2.7 Cellules simple et non-simple.

Pratiquement, l'appellation simple signifie que nous sommes en présence d'un arc assez régulier que nous allons pouvoir reconstruire sans ambiguïté.

Dans le cas contraire, la cellule non-simple sera traitée par la suite comme contenant probablement une auto-intersection de la courbe ou un point de rebroussement.

2.3 Ordonnancement des points dans les cellules du quadtree

A cette étape de la reconstruction de la courbe, nous ordonnons, pour toutes les cellules simples, les sous-ensembles de points qu'elles contiennent.

Le calcul se déroule en trois actions principales.

1° Projection : l'ensemble des points de la cellule est projeté sur le meilleur cercle approximant préalablement calculé.

2° **Tri** : l'angle que représente chaque point projeté sur le cercle est calculé. Le tri des points projetés est alors effectué selon les angles par ordre croissant.

3° **Ordre et paramétrage** : la liste des points de la cellule est à ce moment là ordonnée, et nous lui faisons correspondre l'ordre du paramétrage sur l'arc de courbe que nous cherchons à reconstituer.

Il faut remarquer ici que le point formant l'angle le plus petit sur le cercle ne correspond pas forcément au point de plus petit paramètre sur l'arc de courbe. Cela dépend, en effet, de l' "orientation" de l'arc dans le plan et de l'axe pris comme référence pour le calcul des angles (cf. figure 2.8).

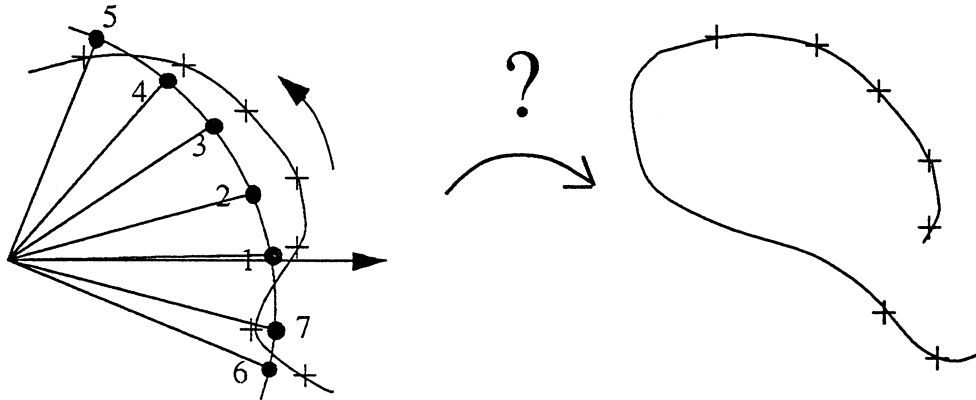


Figure 2.8 Problème de l'ordre par rapport au paramétrage

Ceci explique l'ajout du critère d'arrêt 2.4 pour les opérations de découpe. Comme la variation des angles des points projetés est inférieure à π , nous obtenons ainsi un moyen simple pour retrouver les extrémités de la ligne polygonale formée par les points ordonnés, et correspondant respectivement aux points de paramètres minimum et maximum sur l'arc de courbe ; il nous suffit donc, s'il en est besoin, de couper la ligne polygonale à l'endroit de la plus longue corde et de prendre comme premier et dernier éléments du sous-ensemble ordonné les points extrémités de cette corde (cf. figure 2.9).

2.4 Raccord des arcs de courbe

Nous avons obtenu jusqu'à présent deux catégories de cellules (simples et non-simples) après découpe et approximation, l'une avec des arcs déjà reconstitués et

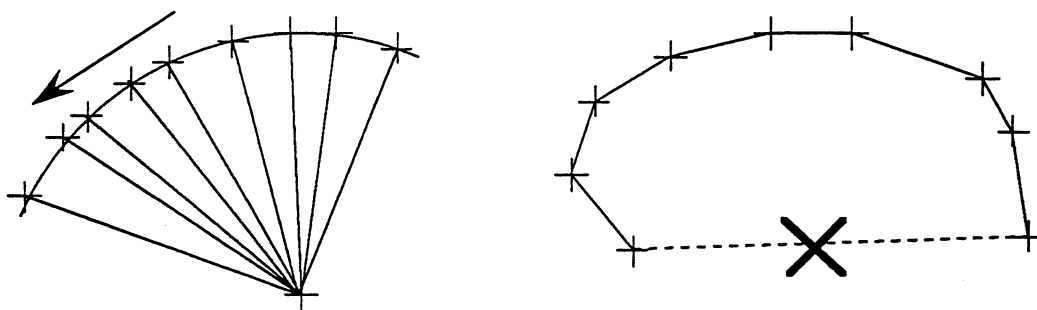


Figure 2.9 Ordonnement des points et calcul de l'arc.

l'autre avec des auto-intersections ou des points de rebroussement éventuels qu'il va nous falloir traiter différemment lors de l'opération de raccord. Nous allons ainsi examiner l'algorithme qui s'applique aux courbes simples, avant de s'intéresser aux courbes non-simples ou comportant des points de rebroussement.

1^{er} cas : pas de cellules non-simples.

Le principe de l'algorithme de calcul des raccords est ici basé sur la structure du quadtree et des relations de voisinage qu'elle implique. Nous allons en effet utiliser ces relations pour décider quelles extrémités d'arcs dans les cellules simples nous allons relier.

Relations de voisinage

Définition 2.8 Soient deux cellules d'un quadtree représentées par les deux domaines de \mathbb{R}^2 suivants :

$\mathcal{A} : [x_{a_{min}}, x_{a_{max}}] \times [y_{a_{min}}, y_{a_{max}}]$ et $\mathcal{B} : [x_{b_{min}}, x_{b_{max}}] \times [y_{b_{min}}, y_{b_{max}}]$.

Alors, on dira que les deux cellules du quadtree représentées par leurs domaines respectifs \mathcal{A} et \mathcal{B} sont **voisines** si et seulement si $\mathcal{A} \cap \mathcal{B} \neq \emptyset$.

Remarque 2.9 : La définition de la relation *voisin* telle qu'elle est énoncée rend comme voisines aussi bien deux cellules admettant uniquement des points communs sur leur frontières, que deux cellules s'intersectant³ ou s'incluant l'une dans l'autre.

L'extension d'une définition "naturelle" à la définition ci-dessus s'avère en effet

³ y compris uniquement en des points de leur frontière.

indispensable dans le déroulement de l'algorithme comme nous le verrons par la suite, à cause de la mise à jour des points minimum et maximum — respectivement les points de la frontière de coordonnées (x_{min}, y_{min}) et (x_{max}, y_{max}) — de la cellule dans laquelle seront fusionnés deux arcs.

Seuillage

Mais la simple relation de voisinage telle que nous l'avons décrite ne suffit pas toujours si l'on veut raccorder de manière "cohérente" les différents arcs de courbe.

Nous utilisons donc aussi un seuil en relation avec la profondeur du quadtree (niveau de découpe) sur les distances entre les extrémités d'arc à raccorder.

Sous une hypothèse de densité de points suffisante sur la courbe, ce seuillage évite ainsi de raccorder entre elles des extrémités trop éloignées et néanmoins voisines dans le quadtree.

Voici alors l'algorithme qui découle de ces considérations :

Oter les cellules vides.

Prendre la première cellule du quadtree comme cellule principale.

Itérer jusqu'à quadtree vide

Sélectionner parmi les cellules voisines simples la cellule courante,
qui contient la plus proche extrémité respectant le seuil.

Raccorder les deux listes de points dans la cellule principale,
et mettre à jour la cellule principale.

Oter la cellule courante.

Fin itérer

Algorithme 1 : Raccord pour des cellules simples

Nous obtenons ainsi à la fin de cette itération la liste des points ordonnés sur la courbe si cette dernière est simple.

Présence de cellules non-simples.

Traitement des auto-intersections ou des points de rebroussement.

Les idées que nous utilisons pour reconstruire la courbe dans les cellules déclarées non-simples sont encore basées sur la notion de voisinage dans le quadtree, mais aussi sur l'utilisation une fois encore de cercles pour reconstruire les différents arcs que va contenir la cellule.

Après avoir traité toutes les cellules simples du quadtree, nous obtenons un ensemble de composantes disjointes. Nous allons alors utiliser l'information donnée par ces composantes dans le voisinage de la cellule non-simple pour calculer un ou plusieurs cercles approximants qui vont traverser celle-ci et nous permettre de reconstituer le ou les arcs qu'elle contient.

En fait, et pour ce faire, nous allons énumérer tous les cas possibles en sélectionnant les extrémités voisines seules ou par paires, et calculer les cercles correspondants.

Morphologie

Il est en effet évident que si nous sommes en présence de plusieurs extrémités voisines de la cellule en cours de traitement, différentes configurations peuvent apparaître quant à la forme ou à la nature de la courbe dans cette cellule.

C'est pourquoi il nous faut à cette étape du raccord dénombrer toutes les possibilités et décider laquelle retenir, puisque nous ne disposons d'aucune information sur la courbe qui puisse nous confirmer la présence d'un point singulier par exemple.

On peut schématiser les différentes possibilités qui s'offrent à nous lors du traitement d'un éventuel point double ou point de rebroussement, comme il est illustré sur la figure 2.9 et la figure 2.10. Les traits gras représentant les configurations de cercles retenues, et donc les morphologies locales qui en découlent, il est alors évident que ces dernières peuvent être très différentes selon les cas.

Nous pourrions alors traiter par exemple la présence d'un point double en prenant deux paires d'extrémités "opposées" par rapport à la cellule, ou encore un point de rebroussement en considérant deux extrémités prises séparément, et non

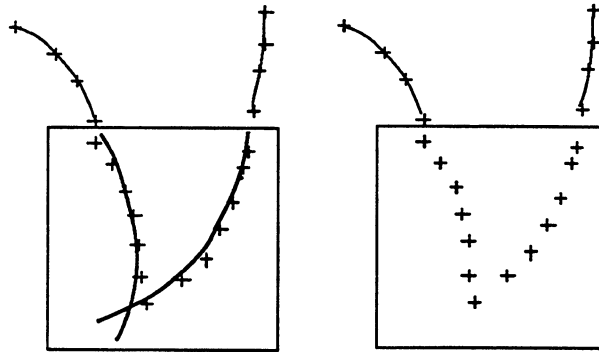


Figure 2.9 Point de rebroussement : énumération des morphologies

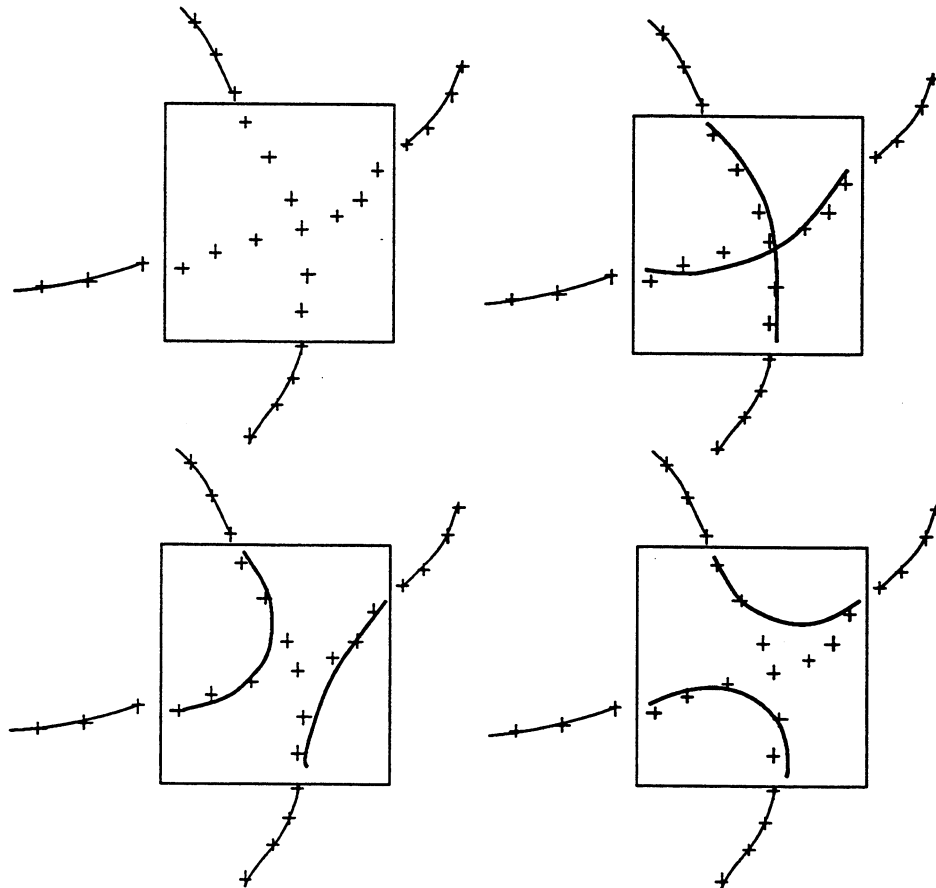


Figure 2.10 Auto-intersection : énumération des morphologies

pas un cercle unique s'appuyant sur ces deux extrémités.

La décision pour retenir une configuration est alors prise par rapport à l'erreur

moyenne commise en projetant les points de la cellule non-simple sur le cercle qui leur est le plus proche.

En résumé, nous recherchons et nous retenons la configuration de cercles qui approxime le mieux possible l'ensemble des arcs de courbe contenus dans la cellule non-simple.

Le point important sur lequel nous voudrions insister ici est que cette énumération est la clef du traitement des éventuels auto-intersections ou points de rebroussement.

C'est uniquement grâce à ce processus de dénombrement qu'il nous est possible de venir à bout de tels cas. Effectivement, le traitement global de la courbe à cet endroit ne peut nous fournir aucune information qualitative, si ce n'est le fait qu'un seul cercle, et plus généralement un seul arc simple ne nous permet pas de mener à bien la reconstruction dans cette cellule.

Calcul des cercles

Le calcul des cercles basés sur les extrémités voisines de la cellule non-simple va se faire différemment selon les cas où celles-ci seront considérées par paires ou isolément.

Dans le cas d'une extrémité seule, nous prolongeons le cercle approximant de l'arc auquel elle correspond à travers la cellule que nous traitons.

Dans le cas d'une paire, plusieurs solutions s'offrent à nous pour le calcul du cercle.

Nous pouvons en effet sélectionner un certain nombre de points voisins des extrémités sur chacun des arcs correspondants et calculer un cercle approximant sur cet ensemble comme nous le faisons dans le cas des cellules simples. Une seconde solution est d'évaluer des tangentes en chaque extrémité, et calculer un cercle passant par celles-ci et tangent à la moyenne des directions tangentes précédemment calculées.

Nous avons testé en pratique ces différentes méthodes, mais les jeux d'essais effectués n'ont pas permis de vraiment distinguer de manière flagrante la meilleure solution.

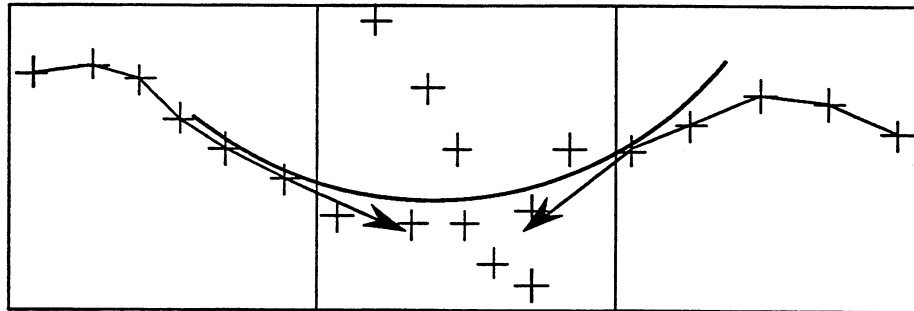


Figure 2.11 cercle tangent

Algorithme

Nous allons donc procéder en deux grandes étapes dans le cas où des cellules non-simples sont présentes dans le quadtree.

La première va consister à traiter toutes les cellules simples du quadtree, de la manière décrite dans la première partie de ce paragraphe.

Nous allons ainsi obtenir un certain nombre de composantes connexes de la courbe formées chacune des points des cellules simples qui auront pu être raccordées, et qui ne sont pas toutes reliées entre elles du fait de la présence des cellules non-simples.

La deuxième étape va concerner les cellules non-simples et va se dérouler de la manière décrite dans l'algorithme ci-dessous.

Initialisation : Prendre la première cellule simple

Itérer jusqu'à (*plus de cellule non-simple*)

Sélectionner une cellule non-simple.

Rechercher les extrémités voisines dans les cellules simples.

Enumérer les cas.

Pour chaque cas :

Prendre chaque combinaison d'extrémités.

Calculer le cercle associé.

Projeter chaque point de la cellule sur le cercle le plus proche.

Calculer l'erreur commise.

Choisir la meilleure configuration.

Calculer les sous-ensembles de points de la cellule correspondant à chaque cercle.

Pour chaque sous-ensemble :

Calculer l'arc.

Raccorder l'arc avec les extrémités voisines.

Stocker la nouvelle composante ainsi créée dans l'une des cellules simples

Supprimer la cellule non-simple.

Fin Itérer

Algorithme 2 : Raccord pour des cellules non-simples

3 Complexité des algorithmes

Dans tout ce paragraphe, nous allons supposer que nous avons un ensemble de n points situés sur une courbe.

Structure de données

Les points P_k , $k = 1, \dots, n$ sont donnés par leurs coordonnées et stockés dans une liste de pointeurs simplement chaînée, chaque pointeur contenant les coordonnées du point et un lien vers le pointeur suivant.

Une cellule du quadtree est une structure contenant :

- les liens de "parenté" (cellules mère et filles).
- un lien vers une cellule suivante.
- Une liste de points qu'elle contient.
- Des informations diverses servant aux algorithmes des différentes étapes : nombre de points, rayon et centre du cercle approximant, points minimum et maximum.

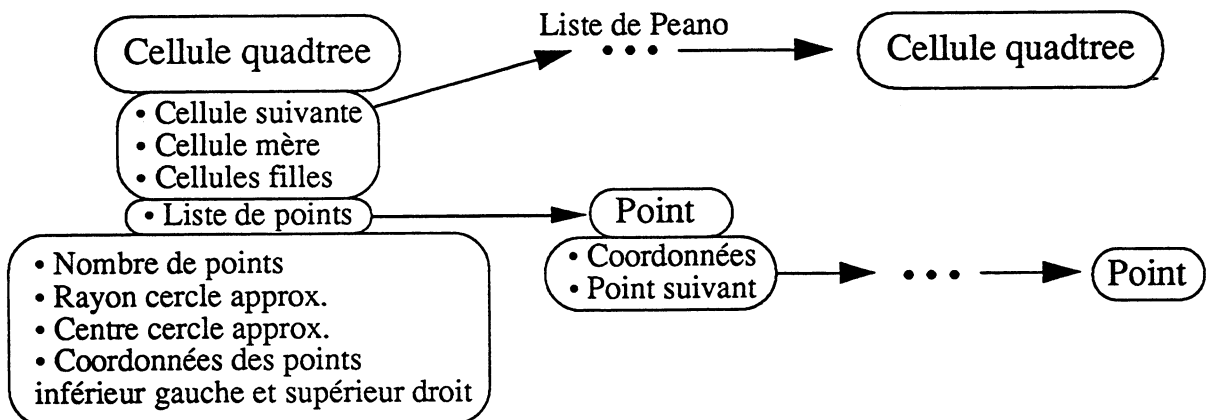


Figure 3.12 Structure de donnée

L'ensemble des arcs de courbes est alors une liste simplement chaînée de cellules feuilles du quadtree.

Nous avons donc :

Proposition 3.1 *Si n est le nombre de points de l'ensemble initial, alors la structure de données occupe une place mémoire de l'ordre de n .*

Preuve : Le nombre m de cellules feuilles du quadtree est au plus égal au nombre de points n , et chaque cellule $c_i, i = 1, \dots, m$ contient k_i points avec $\sum_{i=1}^m k_i = n$, et un nombre constant d'autres informations par rapport au nombre de points. On obtient donc une place mémoire en $O(n)$. \square

Complexité temporelle

Nous allons tout d'abord examiner la complexité de chaque grande étape de la reconstruction de la courbes telles qu'elles ont été décrites auparavant, avant d'énoncer le résultat global du coût de la reconstruction.

Il existe deux grands types de quadtree, comme le font remarquer plusieurs auteurs ([FB 74] par exemple) ; l'un subdivisant un ensemble donné en tenant compte des points, et assurant de façon exacte la bonne répartition des points en nombre ; l'autre subdivisant la région où se trouvent les points, la "bonne" répartition des points étant assurée par des hypothèses de régularité de la distribution des points dans les différentes régions.

Nous utilisons ce second type de subdivision qui ne tient pas compte de la répartition des points dans la région découpée.

Sous l'hypothèse (raisonnable) que les points donnés sont suffisamment dispersés pour être répartis de manière homogène dans les cellules du quadtree lors de la découpe, nous pouvons alors énoncer les résultats, pour un ensemble de n points du domaine $[0, 1] \times [0, 1]$ répartis dans les cellules d'un quadtree subdivisant ce domaine :

Lemme 3.2 [PS 85] *Le coût de la construction d'un quadtree contenant n points du plan est en $O(n \log n)$.*

Lemme 3.3 *Le temps de calcul des cercles approximatifs pour chaque cellule du quadtree contenant au moins 3 points est en $O(n)$.*

Preuve : Comme nous l'avons vu au chapitre I , le calcul d'un cercle approximant nécessite le précalcul d'une matrice de taille $(4, 4)$ à partir de l'ensemble des points de la cellule, puis le calcul d'une valeur propre et d'un vecteur propre de cette matrice.

Il nous faut, pour le calcul de la matrice du problème de minimisation, calculer certaines quantités qui nécessitent pour ce faire de parcourir simplement la liste des points de la cellule.

On aura donc, si chacune des m cellules contient k_i points, un coût en $O(k_i)$ pour chacune d'elles.

D'où un précalcul pour toutes les matrices en $O(n)$.

Quant au calcul des valeurs et vecteurs propres pour chaque cercle, il se fait à partir d'une matrice de taille $(4, 4)$, et on peut faire l'hypothèse qu'il est en temps constant par rapport au nombre de points.

Ainsi, le coût en temps est de l'ordre de n . □

Lemme 3.4 *Soit un ensemble de n points répartis en m sous-ensembles dans les cellules d'un quadtree. Le temps de calcul des m arcs correspondants aux sous-ensembles de points est alors en $O(n \log n)$.*

Preuve : Supposons que chacune des i cellules ($i = 1, \dots, m$) contienne k_i points.

Le calcul d'un arc dans une cellule, comme nous l'avons vu précédemment, nécessite la projection et le tri de l'ensemble de points sur le cercle approximant.

La projection a évidemment une complexité en $O(k_i)$ et il est possible de trier ces k_i points en un temps de l'ordre de $k_i \log k_i$, par un algorithme de type "Divide & Conquer", par exemple.

Nous avons donc pour chacune des cellules un temps en $O(k_i \log k_i)$.

Le coût total pour l'ensemble des arcs est donc en

$$O\left(\sum_{i=1}^m k_i \log k_i\right) \leq O\left(\sum_{i=1}^m k_i \log n\right),$$

et donc $O(n \log n)$. □

Lemme 3.5 Complexité des raccords simples

Soit m cellules contenant chacune k_i point, $i = 1, \dots, m$, qui forment m arcs simples.

Alors, la complexité du raccord de ces m arcs simples est $O(n \log n)$.

Preuve : La recherche de l'extrémité la plus proche parmi les extrémités voisines pour tous les raccords à effectuer va se faire en un temps proportionnel à $O(m \log n)$, puisque $O(\log n)$ est le coût de recherche pour une cellule.

D'autre part, pour chaque cellule que l'on va raccorder, il va falloir soit parcourir la liste de ses points jusqu'à la fin, soit inverser cette liste. Ceci se fait en $O(k_i)$ pour chaque cellule, et donc en $O(n)$ pour toutes les cellules.

D'où un coût en $O(n + m \log n)$ pour effectuer ces deux étapes du raccord, et finalement $O(n \log n)$. □

Nous pouvons alors déduire du lemme 3.2 au lemme 3.5 la :

Proposition 3.6 Complexité dans le cas de raccords simples

La méthode de reconstruction telle que nous l'avons décrite a un coût de l'ordre de $n \log n$ dans le pire des cas, si n est le nombre de points donnés.

Remarque 3.7 : Il apparaît dans cette étude de complexité que le tri des points a un coût important en ce qui concerne toutes les opérations à effectuer (avec la construction du quadtree). Il sera donc judicieux pour une implémentation informatique de choisir un algorithme de tri efficace.

Lemme 3.8 **Complexité dans le cas de raccords de cellules non-simples**

Soient m cellules non-simples, contenant chacune une configuration de points qui va être traitée comme contenant un point multiple ou un point de rebroussement.

Alors la complexité du raccord de ces m cellules est, sous cette hypothèse, $O(n \log n)$.

Preuve : Pour chacune des cellules, la recherche des extrémités voisines est effectuée en $O(\log n)$.

D'autre part, le nombre de morphologies est fini et borné ; en effet, le nombre d'extrémités voisines v est $\ll n$, et engendre C_v^2 combinaisons possibles. Le nombre de cas passés en revue peut alors être considéré comme constant par rapport à n .

Le choix de la meilleure configuration est alors en $O(n)$ (coût dû au calcul du cercle et à la projection).

Quant au tri des points et au raccord des arcs formés dans la cellule, les résultats précédents indiquent que ces opérations se font en $O(n \log n)$.

Nous obtenons donc pour toutes les cellules un ordre de complexité égal à $m \log n + n + n \log n$, ce qui équivaut à $O(n \log n)$. □

Les lemmes précédents induisent donc la :

Proposition 3.9 **Complexité dans le cas de la présence de cellules non-simples**

Le coût de la reconstruction dans le cas présent est $O(n \log n)$.

Enfin, nous pouvons énoncer le résultat de complexité général :

Théorème 3.10 *La complexité de la méthode de reconstruction présentée ici est $O(n \log n)$, quelle que soit la nature de la courbe sur laquelle sont échantillonnés les n points donnés.*

4 Temps d'exécutions et exemples

4.1 Etude de temps

Les quatre premiers graphiques sont en rapport avec les temps d'exécution de l'algorithme sur les deux exemples de la courbe en "U" (cf. figure 4.17) et de la spirale (cf. figure 4.19). Les courbes représentent donc les temps CPU en secondes en fonction du nombre de points. Ces temps correspondent à des calculs effectués sur une station de travail IBM RS 6000.

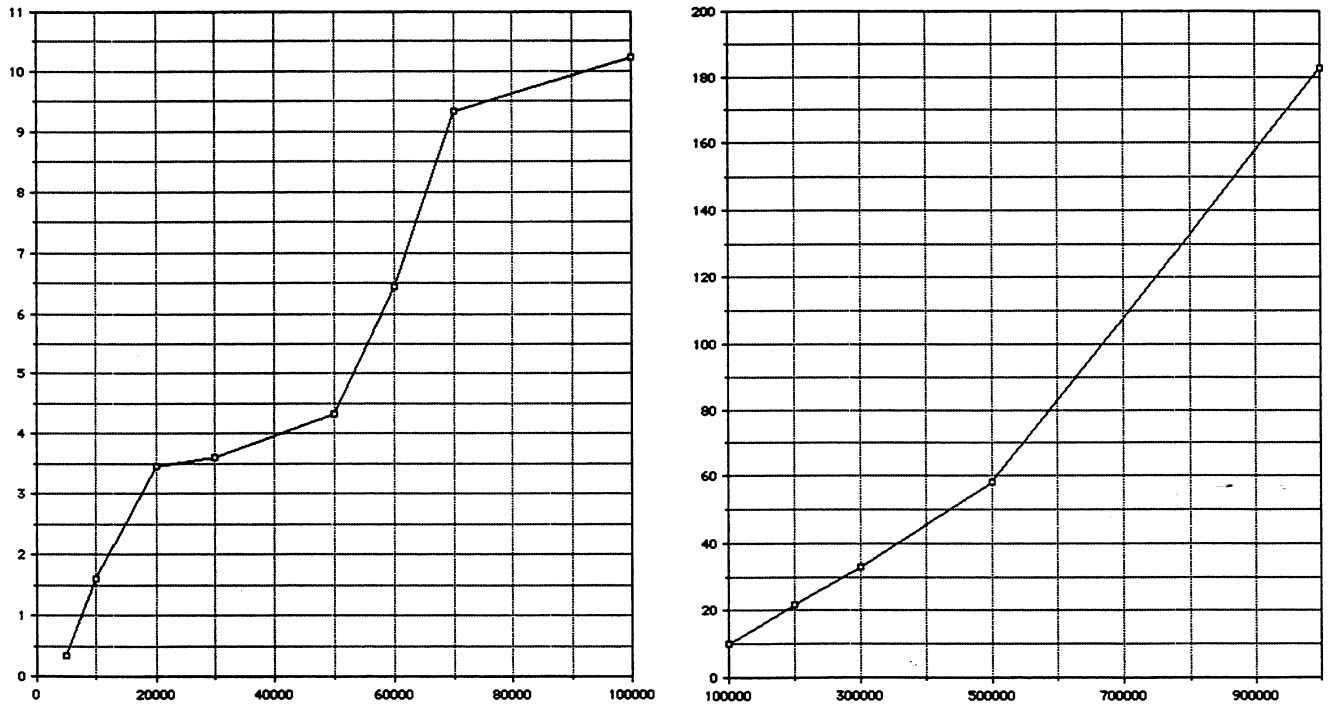


Figure 4.13 courbe en "U" : temps d'exécution en secondes en fonction du nombre de points

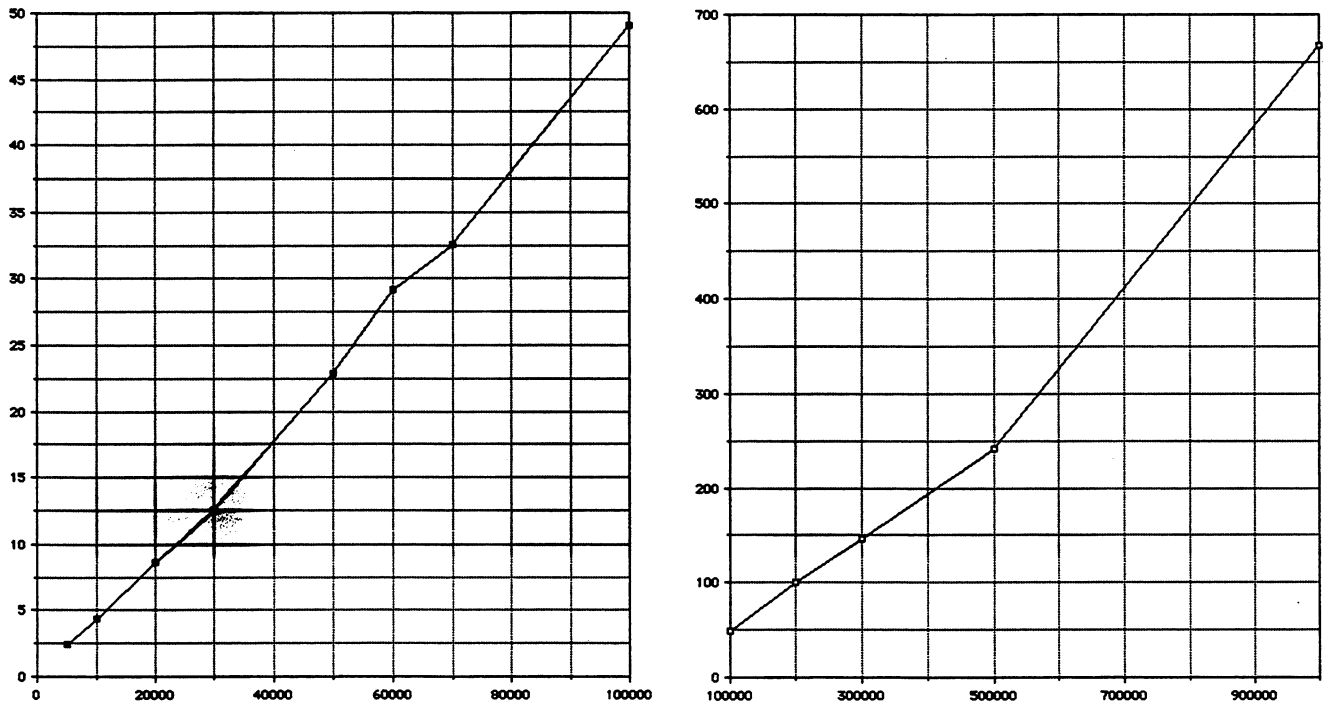


Figure 4.14 spirale : temps d'exécution en secondes en fonction du nombre de points

Nous avons ensuite représenté sur les graphiques suivants la part de temps consacrée à effectuer les quatre grandes étapes de l'algorithme sur ces mêmes exemples d'exécutions.

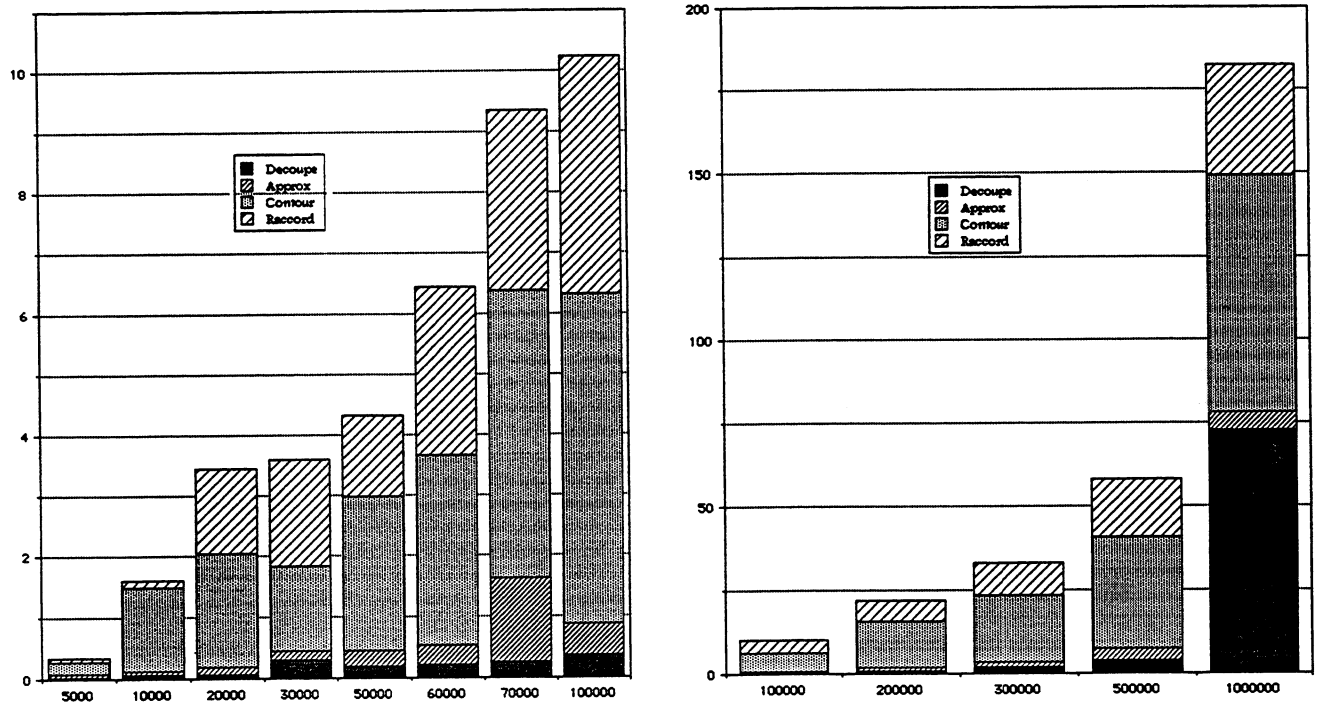


Figure 4.16 courbe en "U" : temps d'exécution cumulés

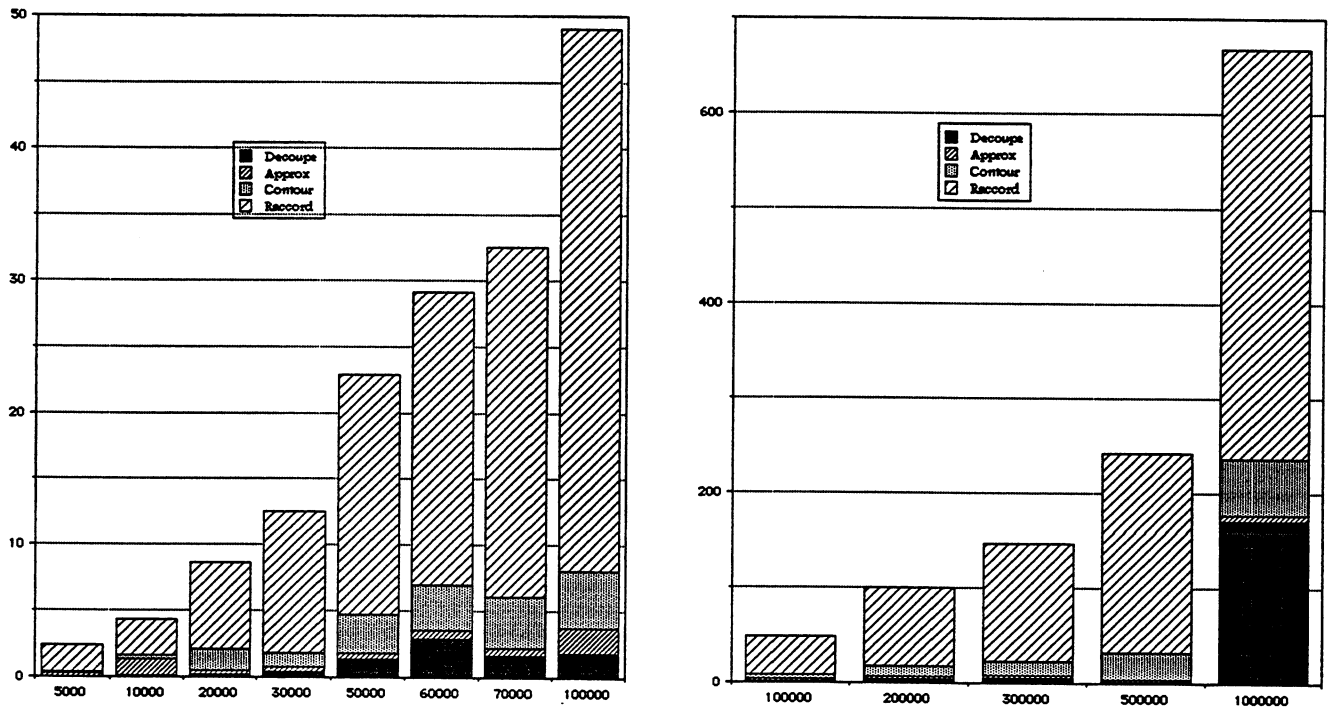


Figure 4.17 spirale : temps d'exécution cumulés

Ces histogrammes cumulés nous amènent à faire deux constatations, qui se sont vérifiées sur d'autres exemples.

La première est que, comme nous l'avions fait remarquer lors de l'étude de complexité, le choix d'un algorithme de tri efficace en temps est essentiel pour assurer de bons temps d'exécutions.

En effet, il ressort clairement sur les graphiques relatifs à la courbe en "U", que l'on peut par ailleurs considérer comme un cas standart de courbe simple sans forte variation de courbure, que la moitié ou plus du temps d'exécution est consacrée au tri des points, et le reste au raccord pour une grande partie.

La deuxième constatation est faite par rapport à l'exemple de la spirale.

On peut en effet s'apercevoir que, dans ce cas, c'est le raccord des différents

arcs calculés dans chaque cellule qui prend nettement le plus de temps.

Ceci s'explique par le fait que nous soyons, dans ce cas là, obligés de beaucoup plus découper la courbe que dans le cas de la courbe en "U". En effet, la morphologie de la spirale nous contraint à un grand niveau de subdivision pour parvenir à séparer effectivement les différents arcs dans différentes cellules.

Nous rejoignons à ce moment là un algorithme de type proximité, avec un ou très peu de points par cellule, et cela montre que nous perdons alors le bénéfice temporel provenant du traitement global des arcs, pour obtenir des temps d'exécutions beaucoup plus importants.

Cet exemple comporte 100 points et une distribution aléatoire de points :

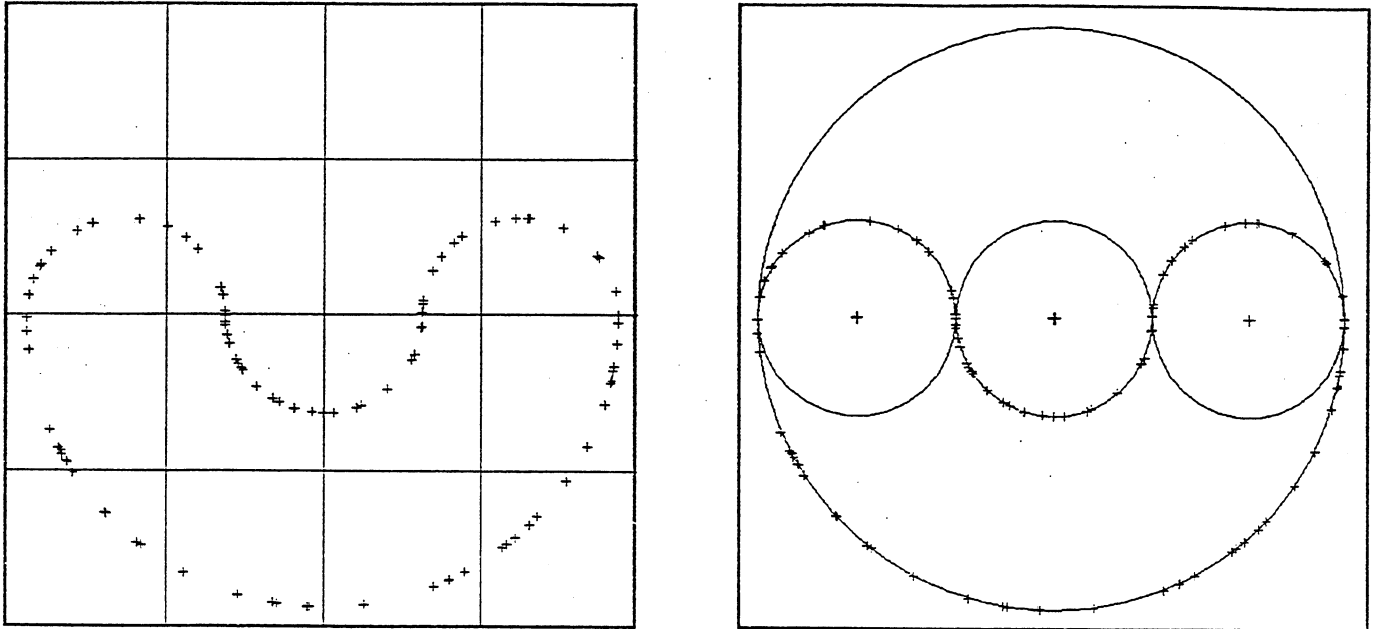


Figure 4.18 courbe en "U" : étapes de subdivision et d'approximation

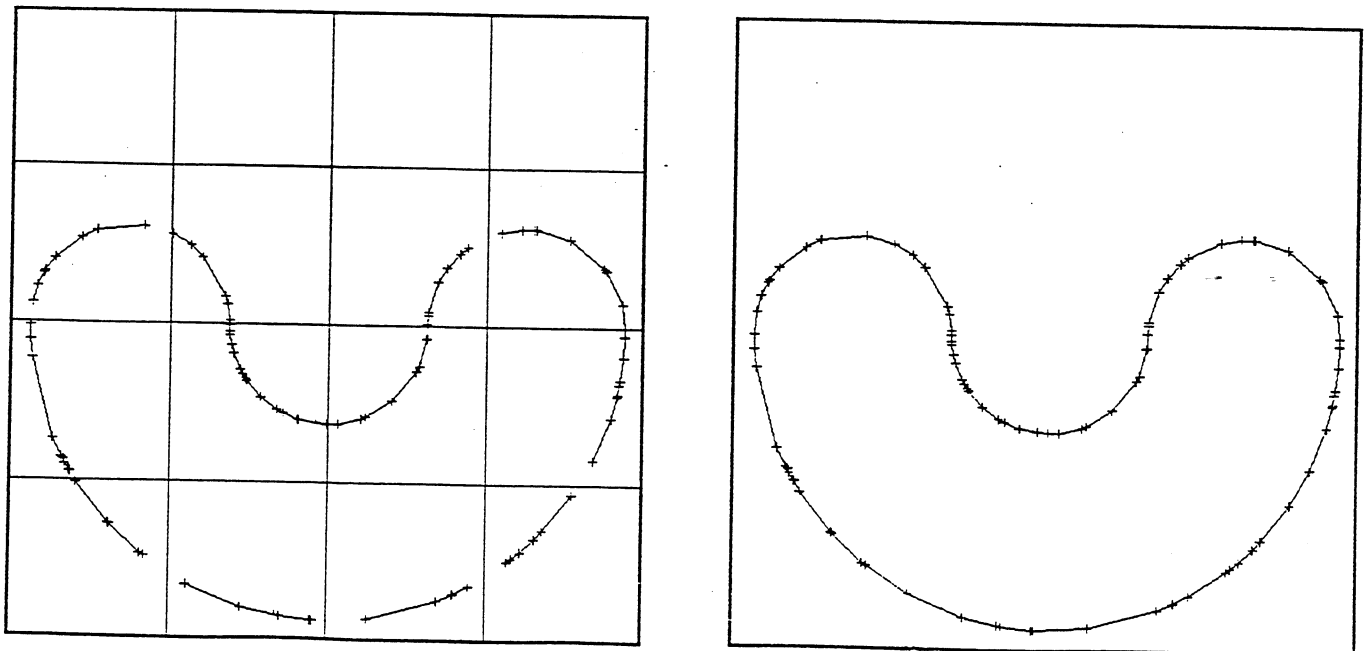


Figure 4.19 courbe en "U" : étapes de calcul des arcs et de raccord

Ces exemples ont été tracés avec 300 points et une distribution régulière :

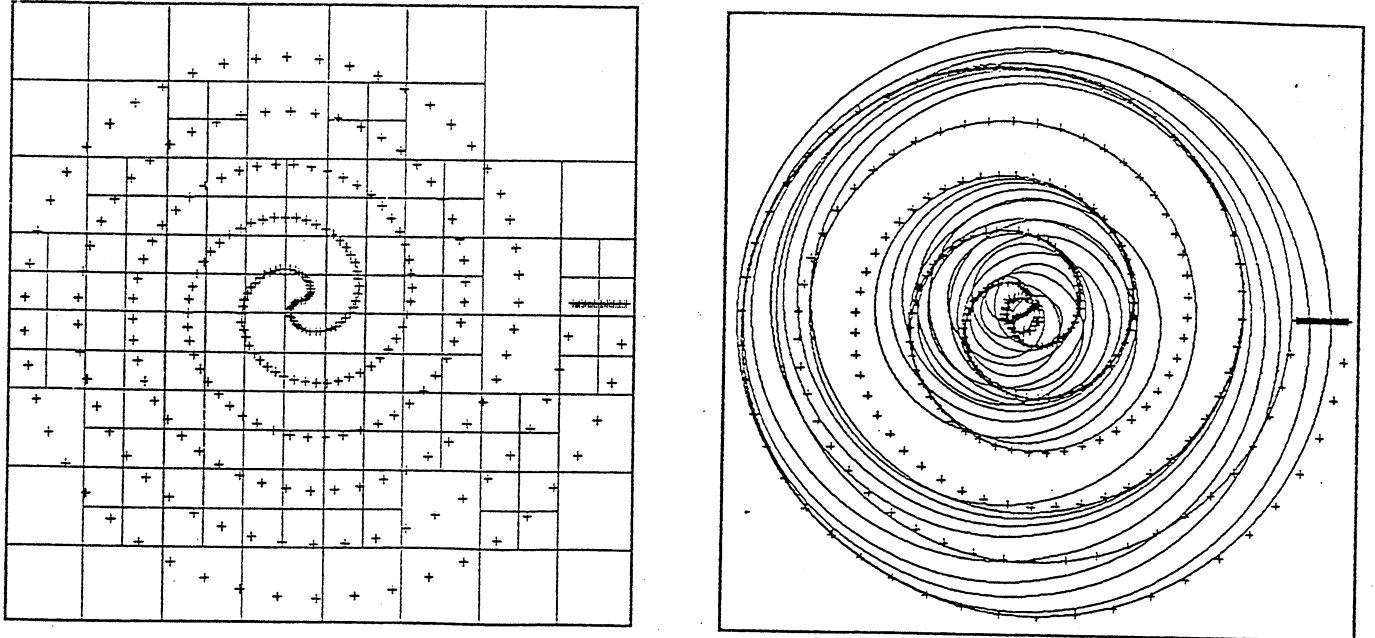


Figure 4.20 spirale : étapes de subdivision et d'approximation

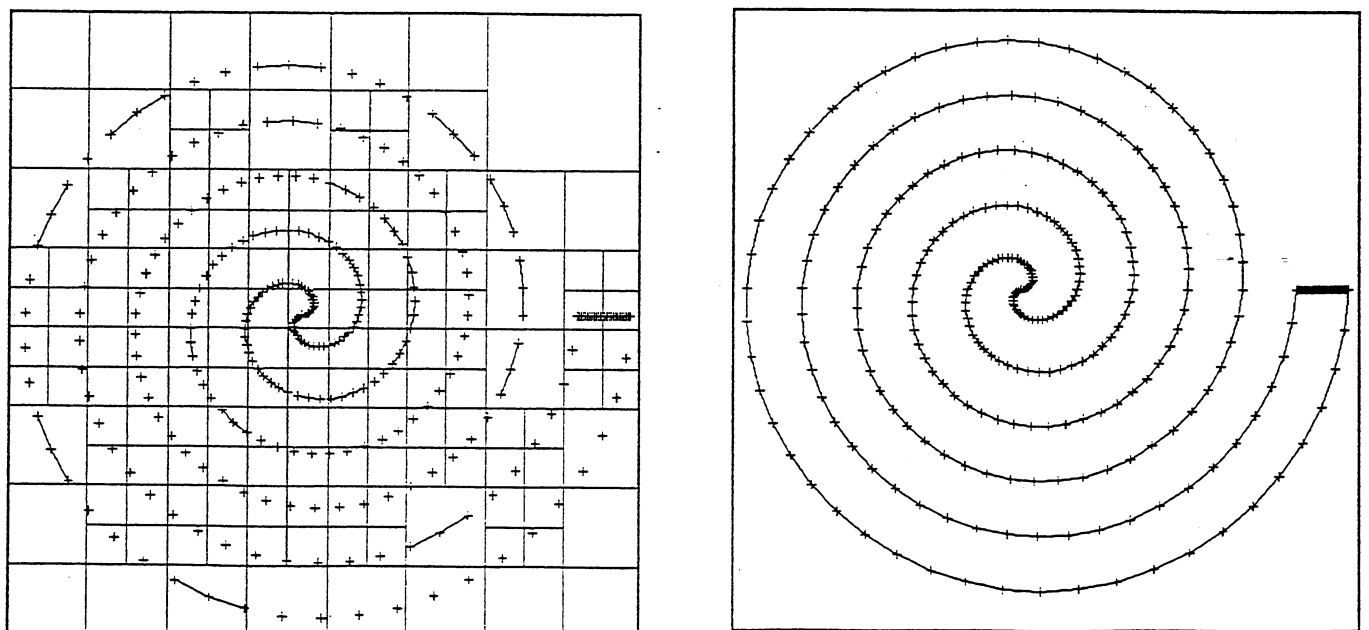


Figure 4.21 spirale : étapes de calcul des arcs et de raccord

Remarquons enfin que l'algorithme de reconstruction n'est pas tributaire d'une distribution régulière des points sur la courbe pour l'obtention du résultat souhaité, comme on peut le constater sur la figure figure 4.21 . Le type de distribution n'a de même aucune influence sur les temps de calcul.

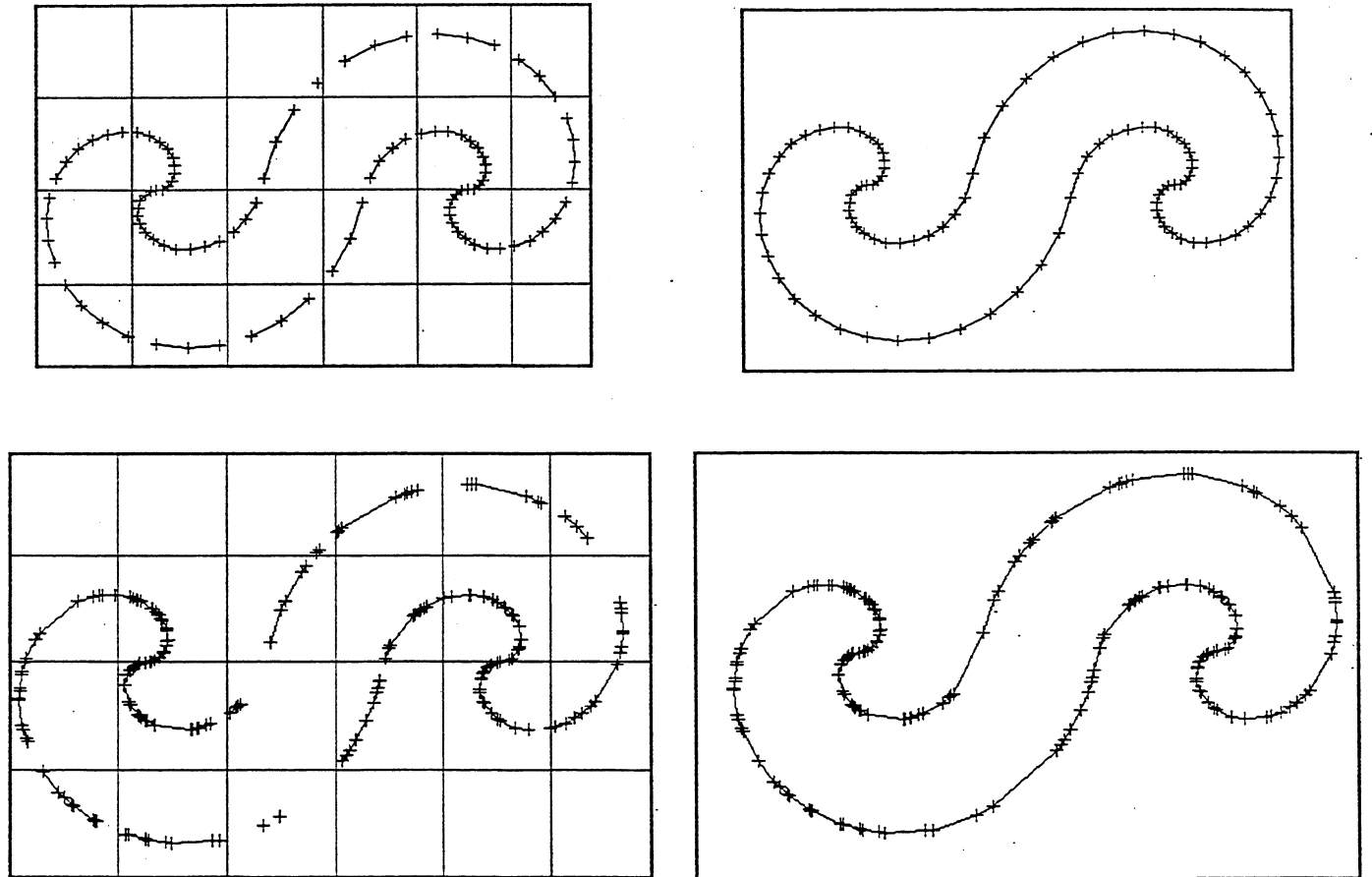


Figure 4.21 200 points, distributions régulière et aléatoire

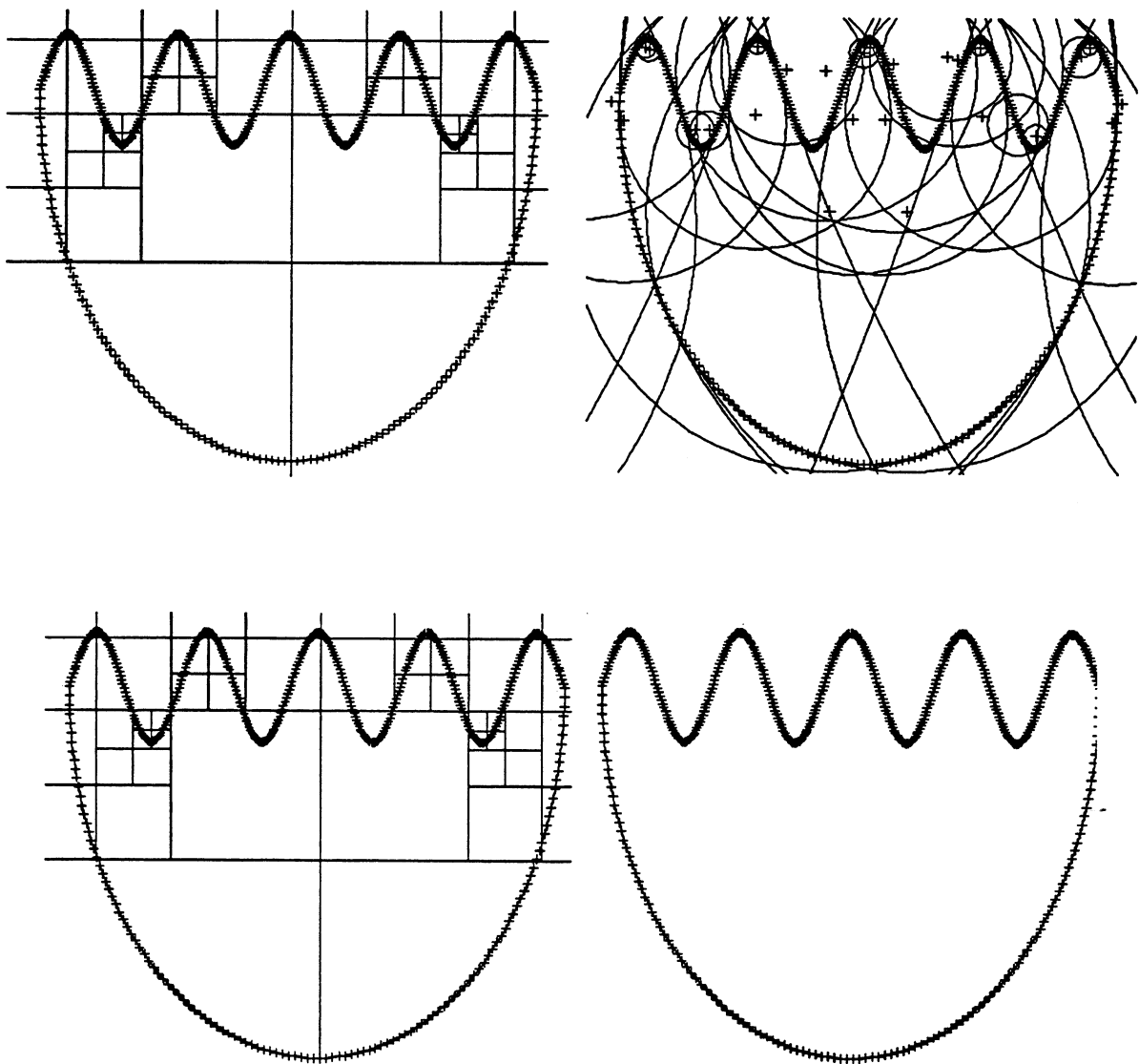


Figure 4.22 300 points, distribution régulière.

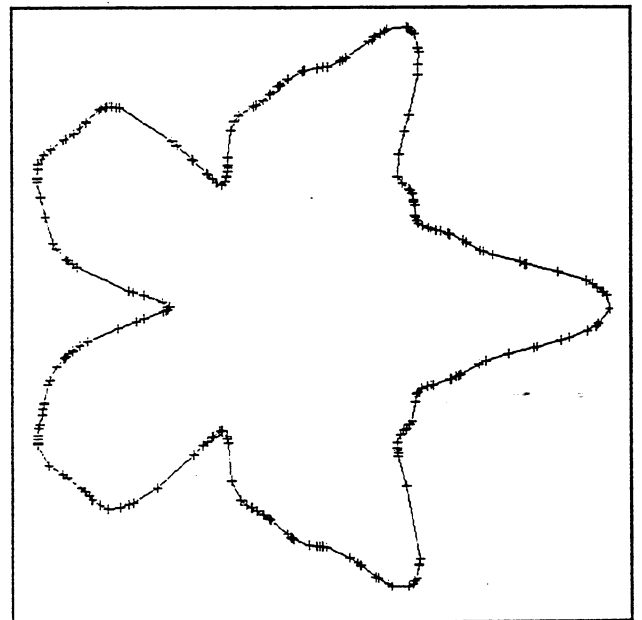
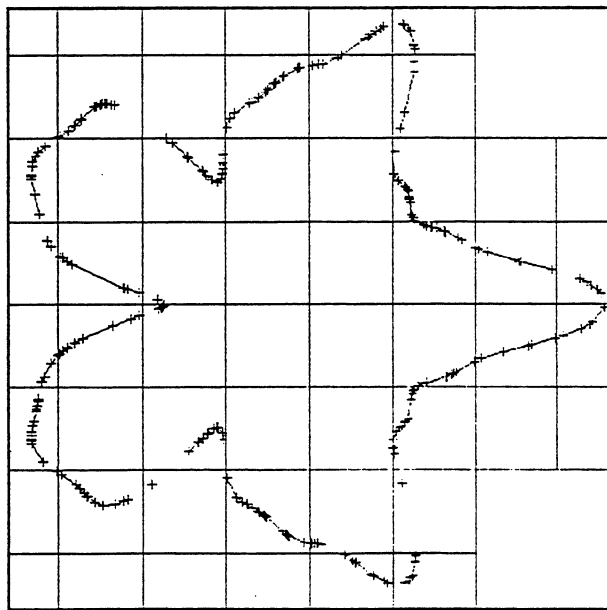
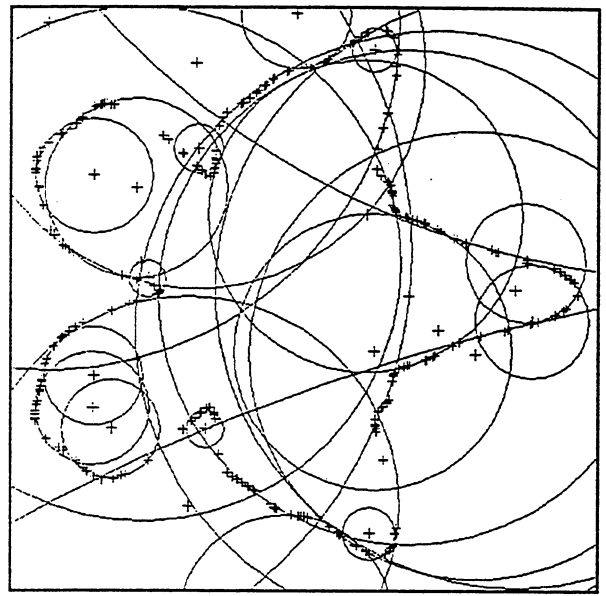
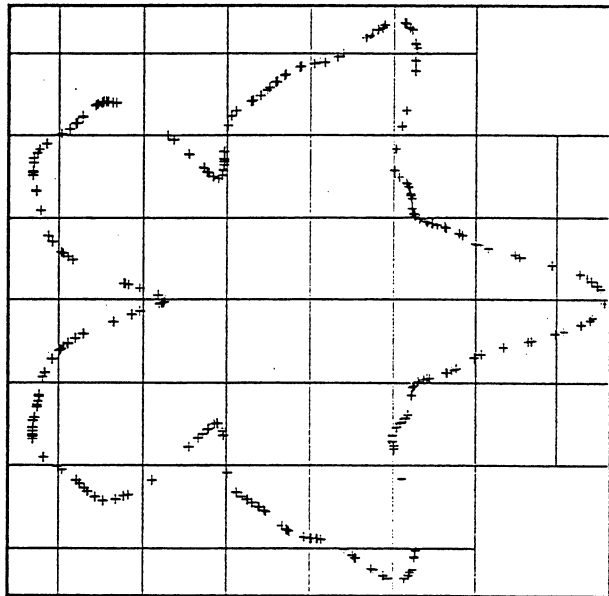


Figure 4.23 300 points, distribution aléatoire.

4.2 Exemples de données bruitées

Nous illustrons par ces exemples la bonne robustesse de la méthode lorsque les données sont perturbées. Nous avons pris pour cela un bruit gaussien et écarté les points selon la normale unitaire à la courbe, d'une valeur d'au plus la valeur du bruit multiplié par la longueur de cette normale.

Les temps d'exécutions sont les mêmes que dans les cas non-bruités.

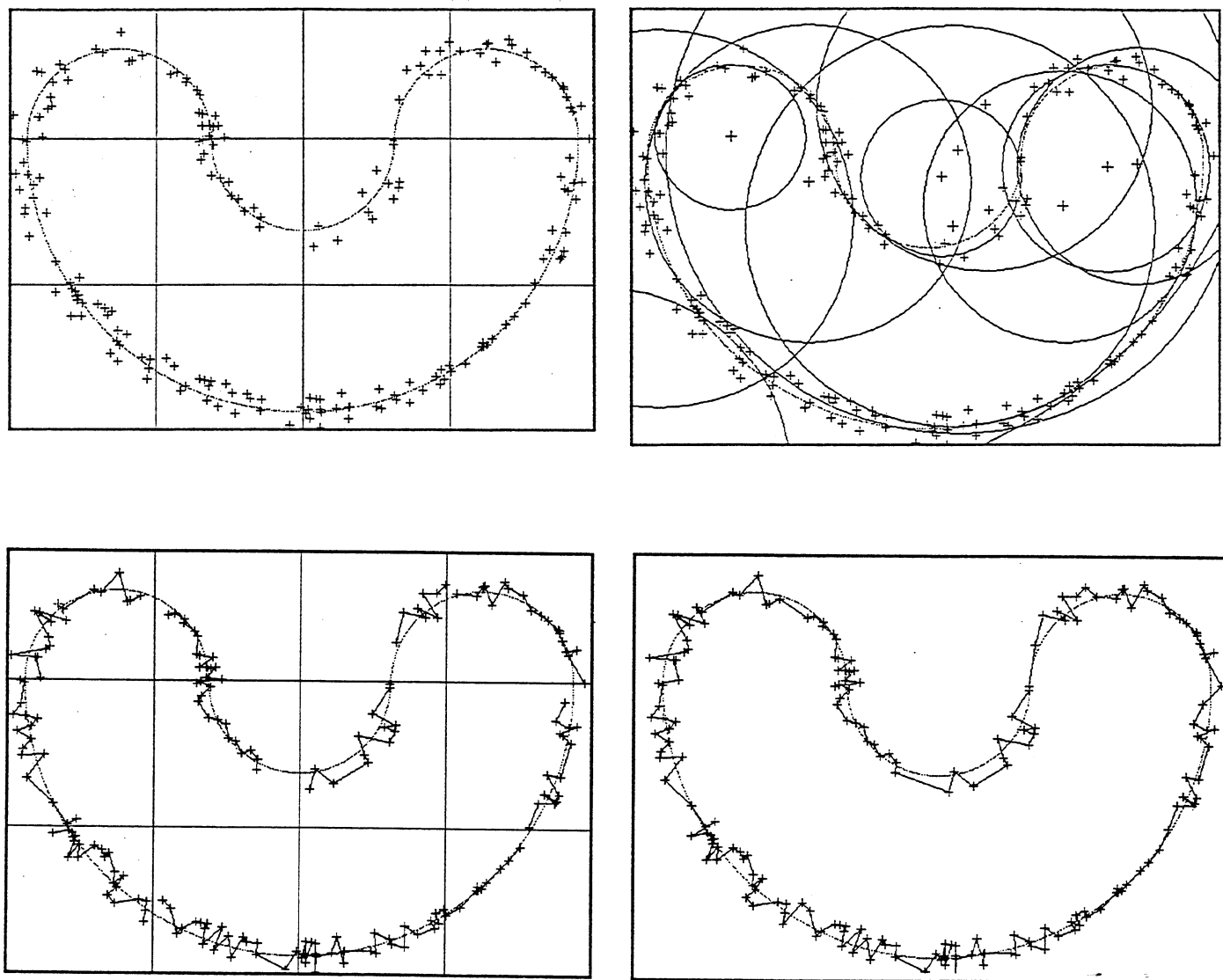


Figure 4.24 200 points, bruit 3%

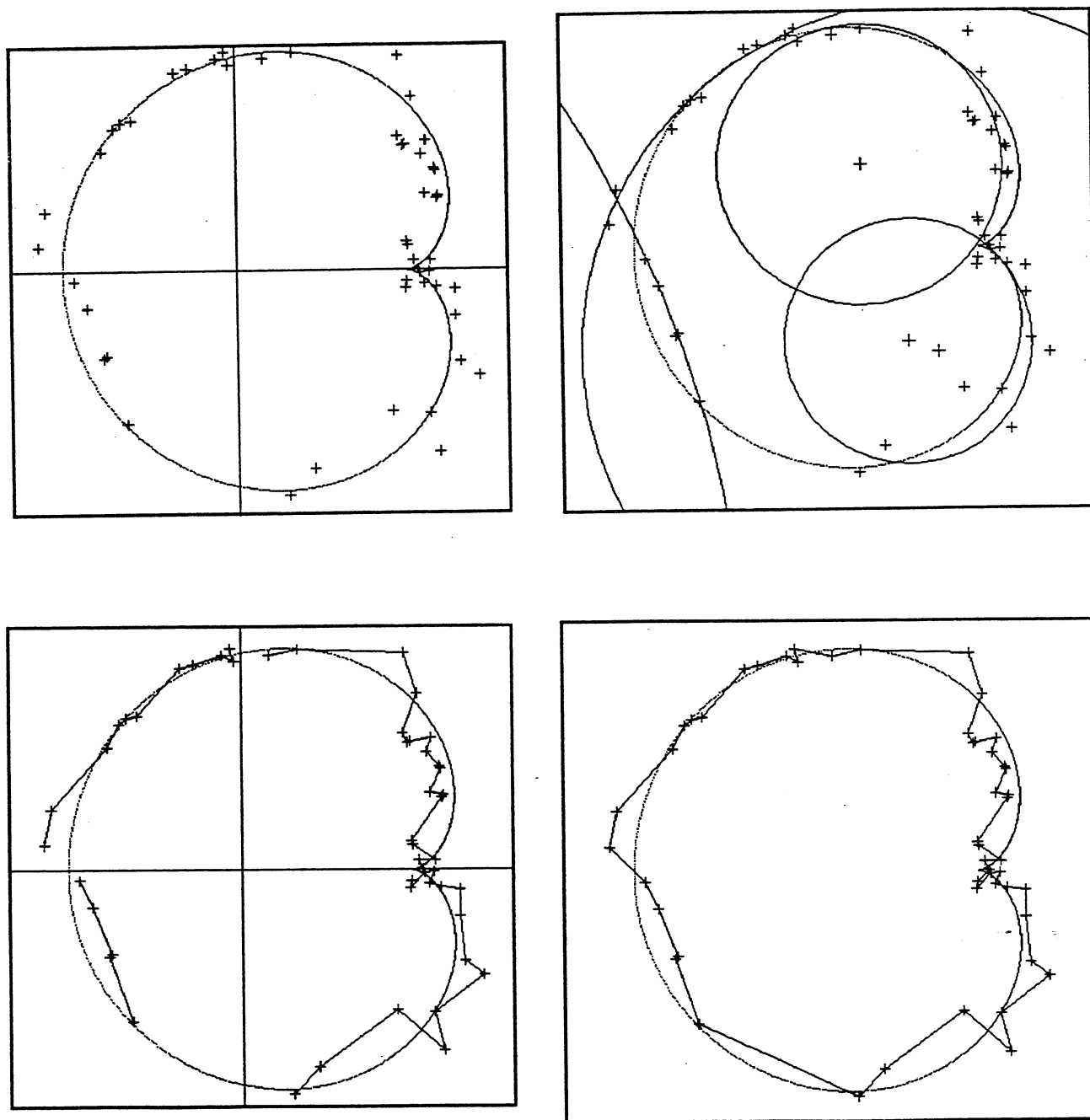


Figure 4.25 50 points, bruit 5%

4.3 Auto-intersections et points de rebroussement

Nous illustrons dans ce paragraphe le comportement de l'algorithme lorsque des cellules non-simples sont présentes.

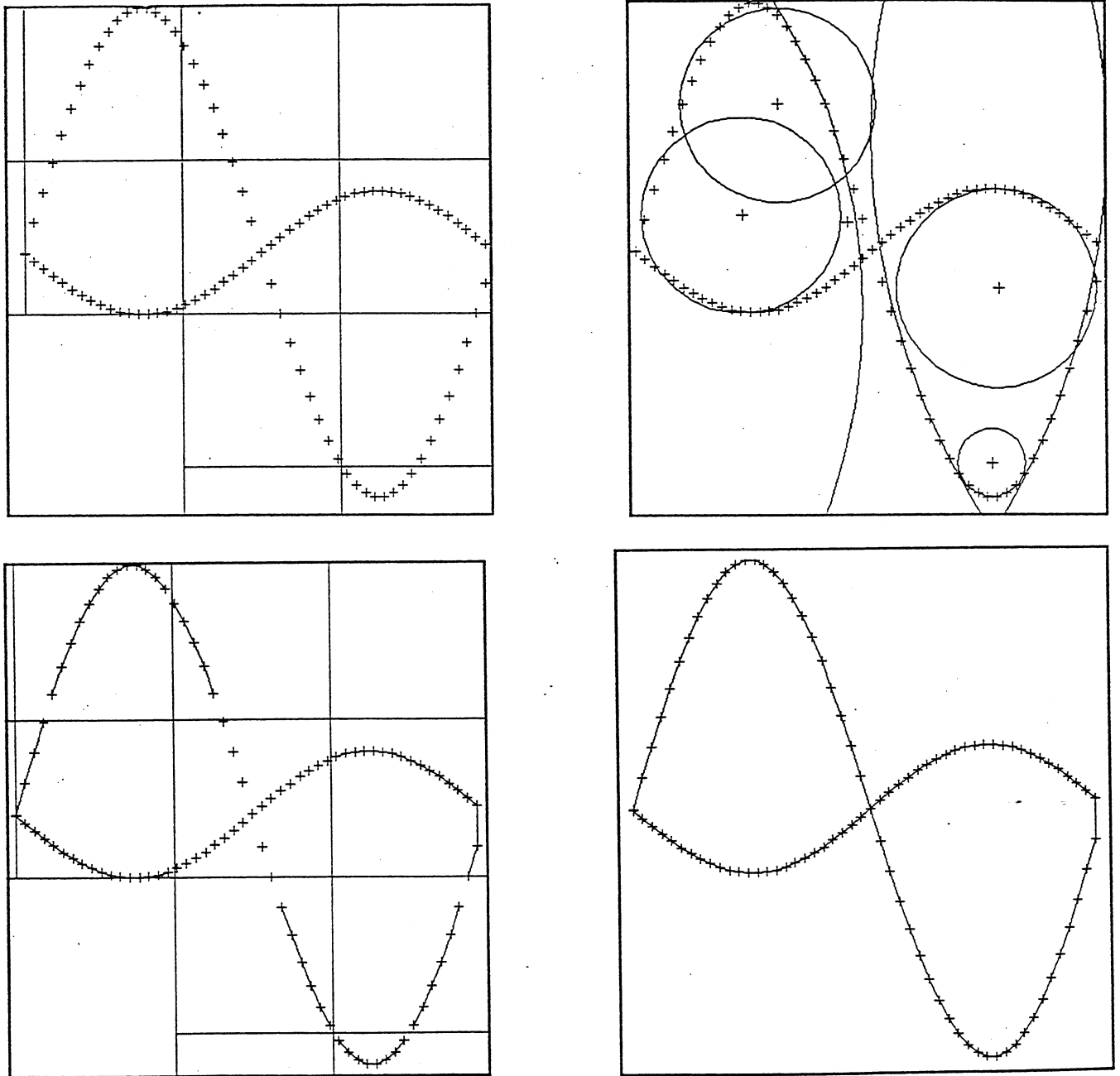


Figure 4.26 Auto-intersection : 100 points, distribution régulière

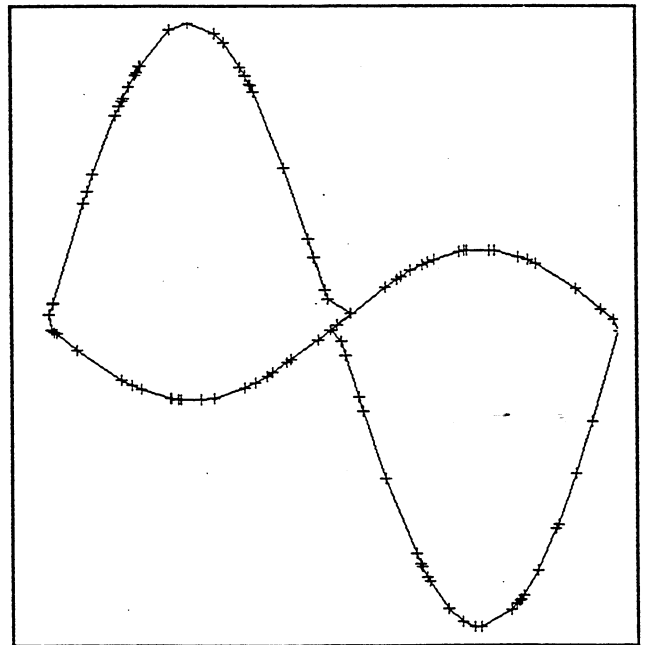
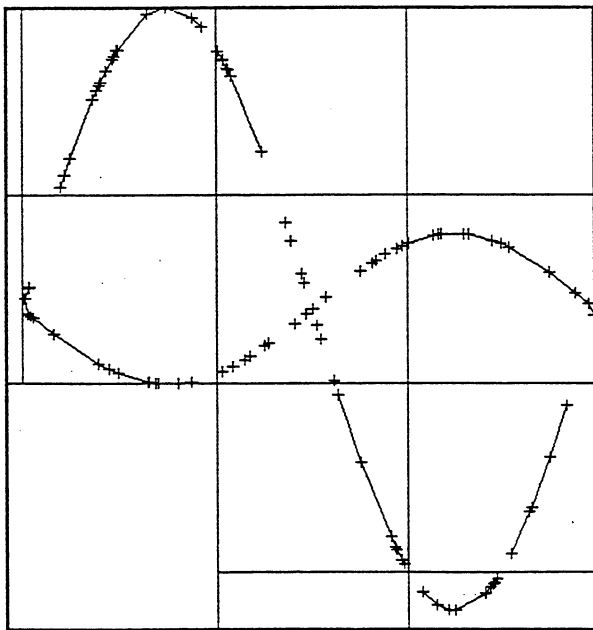
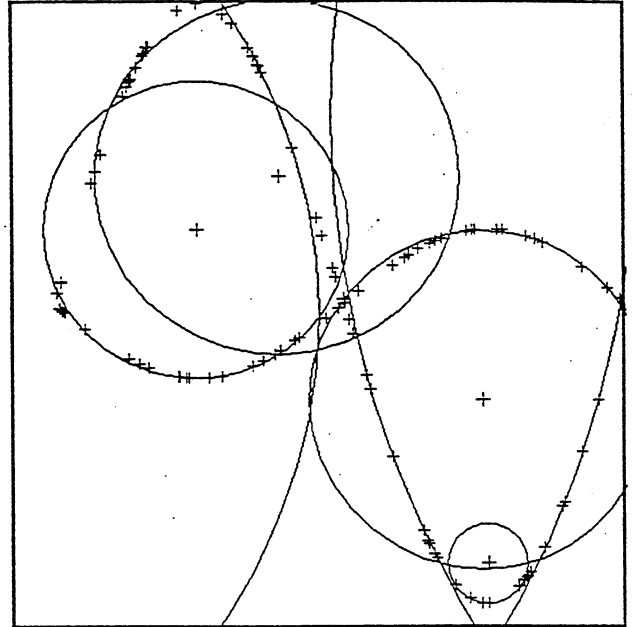
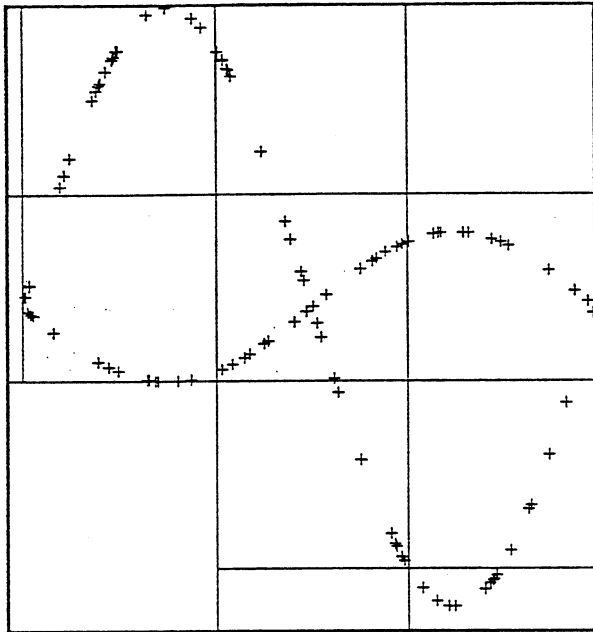


Figure 4.27 Auto-intersection : 100 points, distribution aléatoire

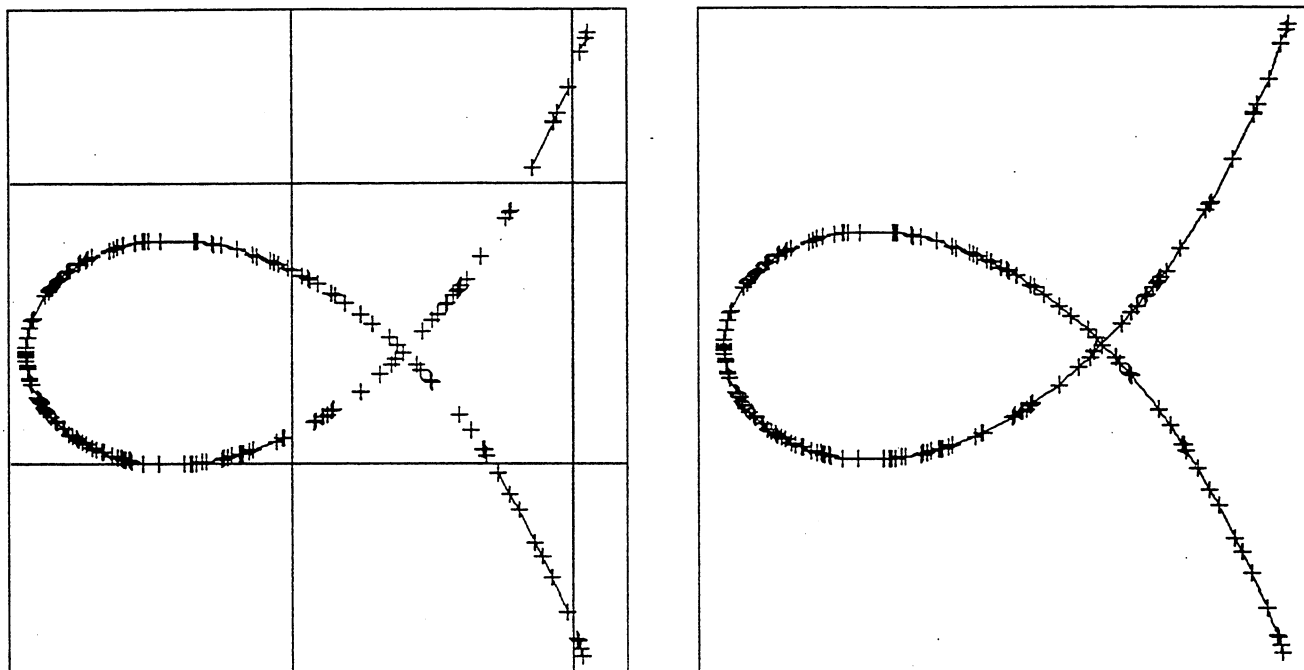


Figure 4.28 Auto-intersection : 200 points, distribution aléatoire

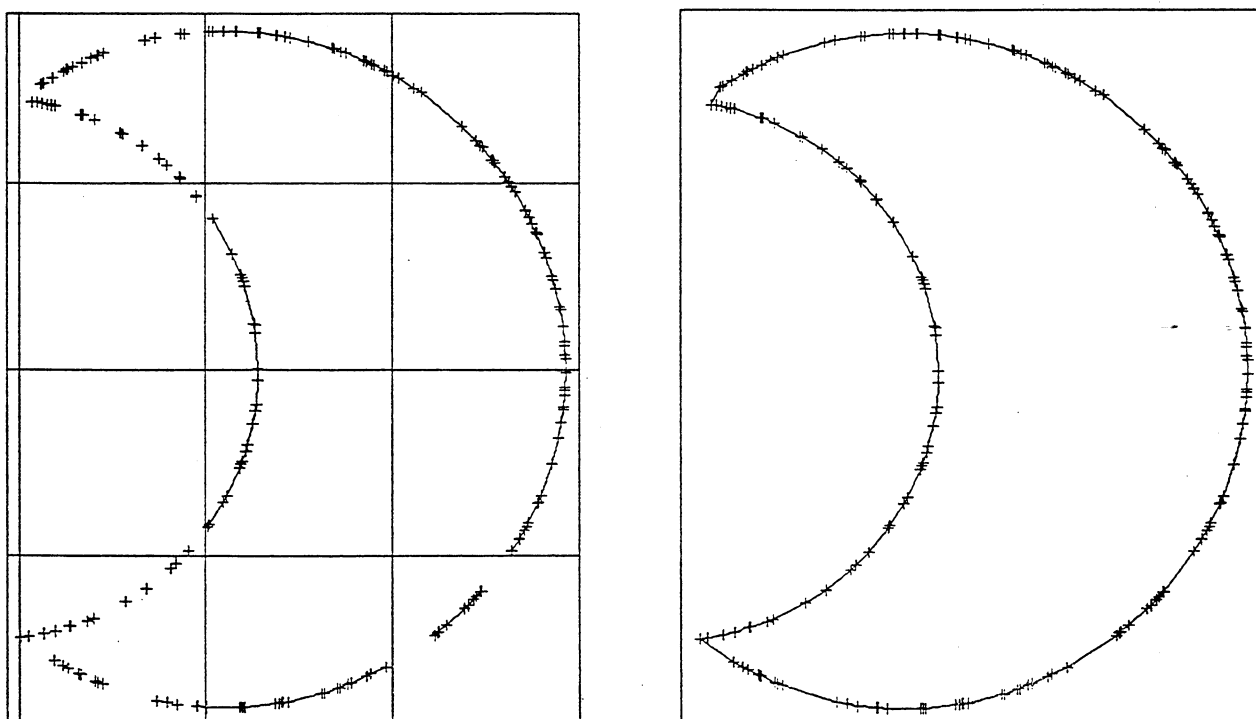


Figure 4.29 Points de rebroussement : 200 points, distribution aléatoire

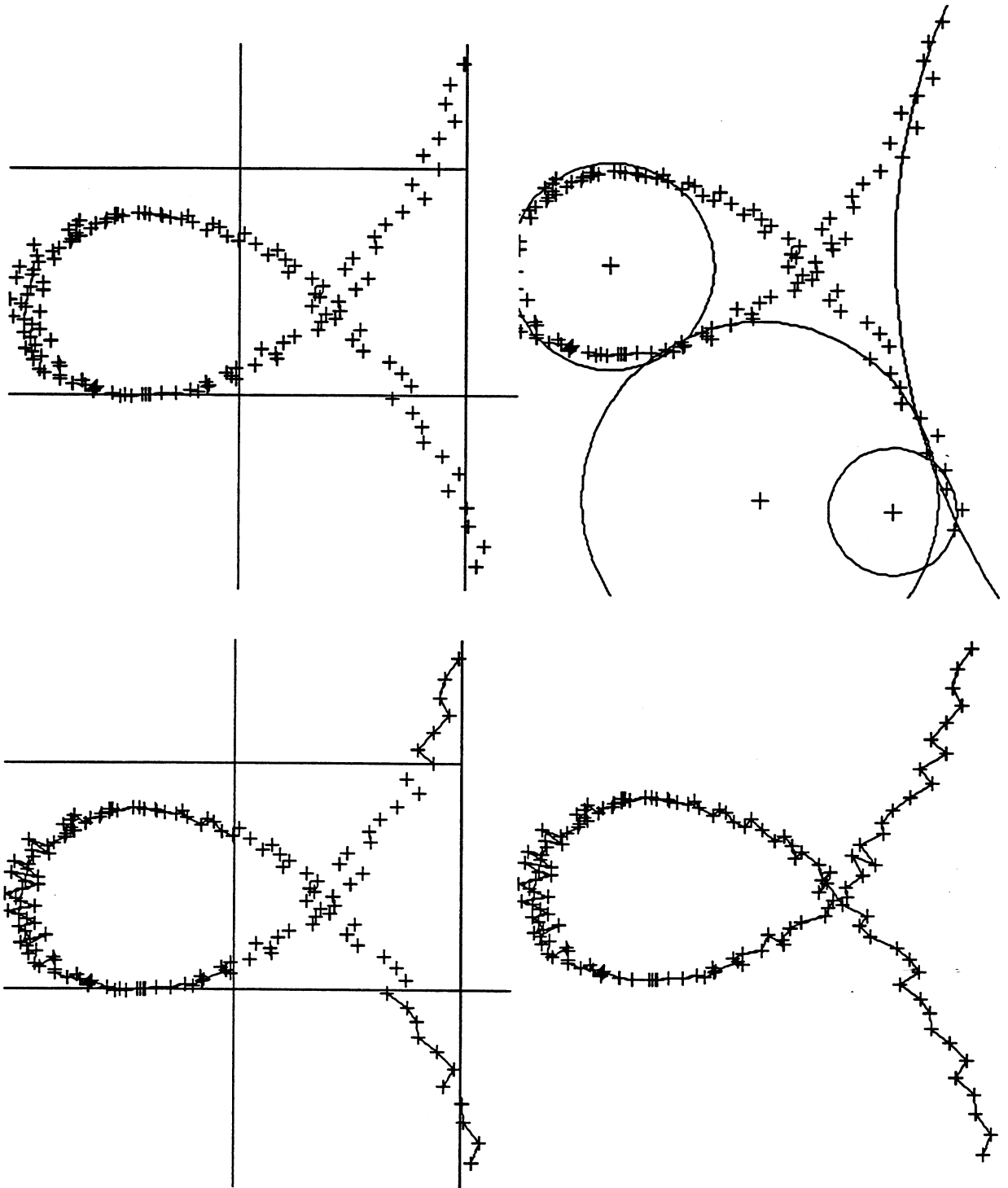


Figure 4.30 Auto-intersection : 200 points, distribution aléatoire et bruitée

5 Application de l'ordonnement des points : calcul d'une paramétrisation

Notre but final, comme nous l'avons déjà précédemment remarqué, est de se servir de l'ordre déterminé sur l'ensemble des points donnés pour associer à ces points un paramétrage.

En effet, comme le souligne Farin ([Far 88]), le problème d'interpolation par une spline est habituellement posé comme "étant donnés des points x_i associés aux valeurs de paramètre u_i, \dots " ; mais en pratique, les valeurs des paramètres sont rarement connues, et doivent donc être calculées.

Nous ne reviendrons pas ici sur les nombreux travaux réalisés dans ce but, et dont quelques uns ont été présentés dans l'introduction.

Notre but n'est pas ici de comparer les différentes méthodes, mais simplement d'examiner si nous pouvons calculer, à partir de notre approche du problème de reconstruction, un paramétrage viable, surtout dans le cas de données perturbées.

L'idée que nous avons appliquée ici est ainsi de se servir des cercles approximaux pour calculer les paramètres dans le cas de points bruités.

Nous associons donc aux points non pas la paramétrisation par longueur d'arc des points sur la courbe, mais la paramétrisation par longueur d'arc sur les cercles approximaux des points projetés correspondants.

Avec le même objectif de ne pas vouloir comparer différents procédés d'interpolation ou de lissage, nous avons fixé notre choix sur une méthode pour calculer la courbe, en l'occurrence celle de calcul de courbes fermées par B-splines régularisées, développée par P. J. Laurent [Lau 87].

B-splines régularisées

Le but est de calculer une fonction approximante, combinaison linéaire de B-splines de degré $d=1, 2$ ou 3 , en introduisant un terme de régularisation basé sur l'intégrale du carré de la dérivée d'ordre d de la spline.

Se donnant des points (x_i, y_i) du plan, correspondants à des valeurs de paramètres t_i d'un intervalle $[a, b]$, on calcule séparément les composantes x et y en fonction de t .

Si S désigne l'espace des fonctions définies sur $[a, b]$ et formées de polynômes de degré d sur chaque intervalle d'une subdivision régulière de $[a, b]$, on peut écrire toute fonction s de S comme :

$$s(t) = \sum_{k=-d}^{n-1} \alpha_k B_k,$$

où B_j désigne la B-spline de degré d dont le support est $[u_k, u_{k+d+1}]$, $k = -d, \dots, n-1$.

La méthode consiste alors, pour les courbes ouvertes, à calculer la fonction $s \in S$ qui minimise :

$$\sum_{k=1}^{nd} (s(t_k) - x_k)^2 + \tau \int_a^b (s^{(d)}(t))^2 dt,$$

où τ est un paramètre positif.

Pour les courbes fermées, il faut ajouter des contraintes de périodicité. Le problème de minimisation est alors :

$$\sum_{k=1}^{nd} (s(t_k) - x_k)^2 + \tau \int_a^b (s^{(d)}(t))^2 dt,$$

avec s vérifiant les conditions :

$$s^{(m-1)}(u_0) = s^{(m-1)}(u_n), \quad m = 1, \dots, d, \quad \text{avec } u_0 = a, u_n = b.$$

Nous avons ainsi effectué des tests dans le cas de données perturbées que nous présentons ci-dessous, et qui se sont avérés plutôt satisfaisants comme on peut le constater sur les exemples.

Les figures représentent donc 4 tests extraits de nos expérimentations. A chaque fois sont dessinés le contour reconstitué et en dessous la courbe réelle et les cent abscisses de calculs en lesquelles la B-spline a été évaluée. Nous donnons en légende le nombre de points de l'ensemble initial, ainsi que la valeur du bruit et la valeur du paramètre τ de régularisation.

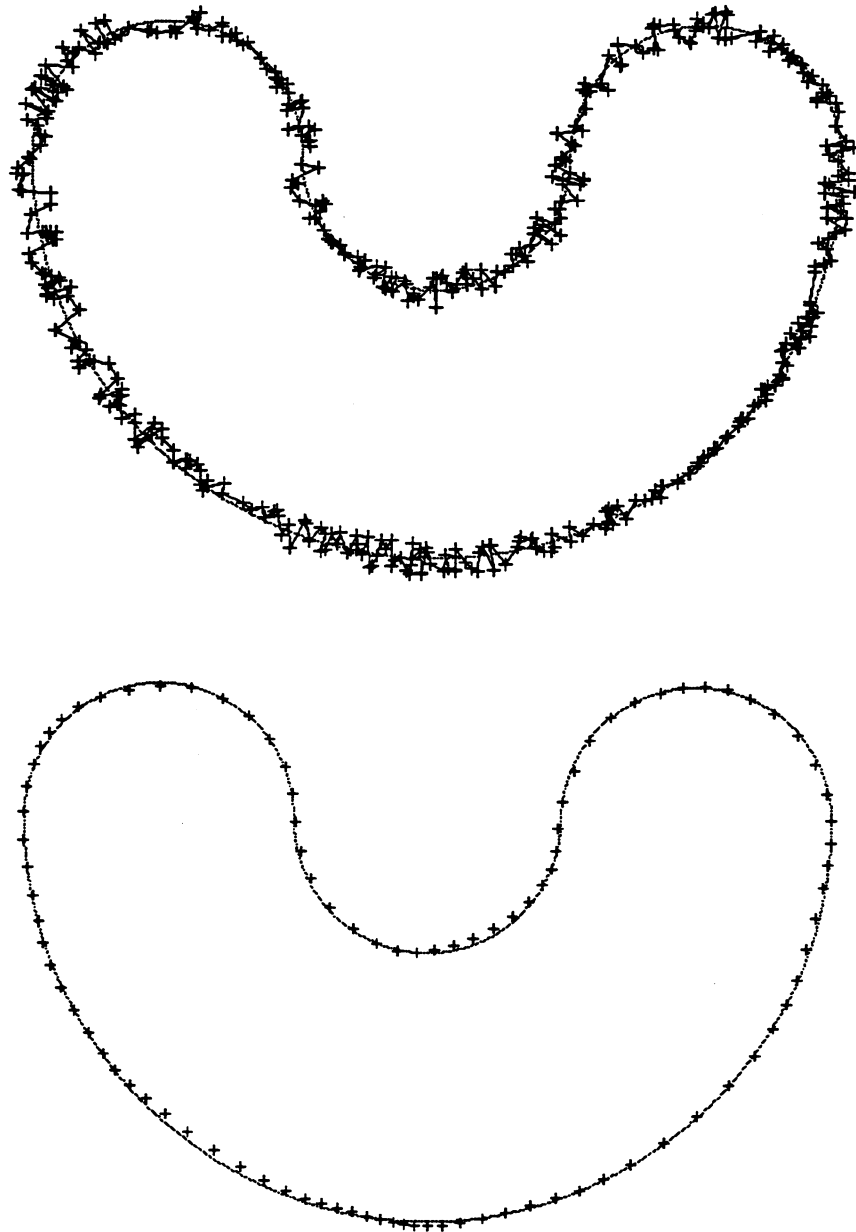


Figure 5.31 500 points, bruit = 2%, $\tau = 0,1$

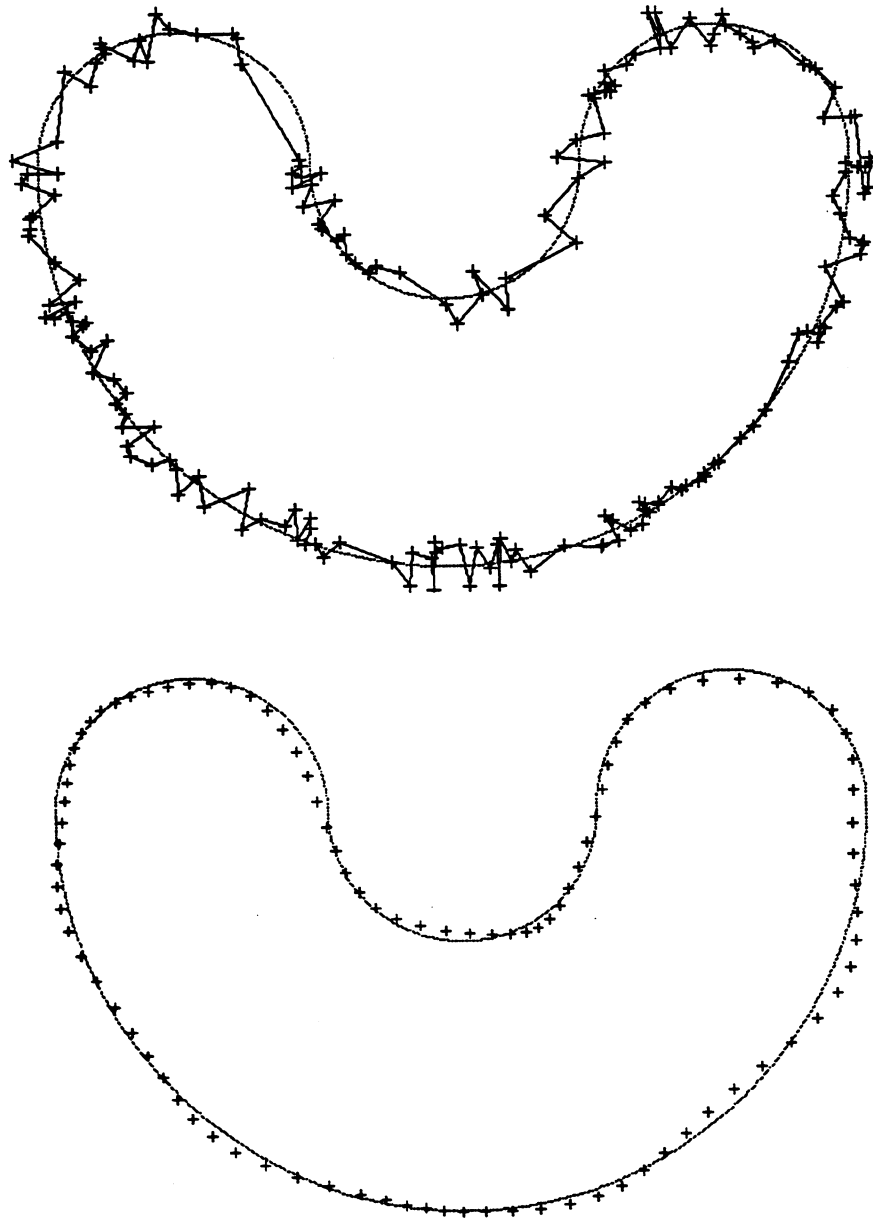


Figure 5.33 500 points, bruit = 3%, $\tau = 0$

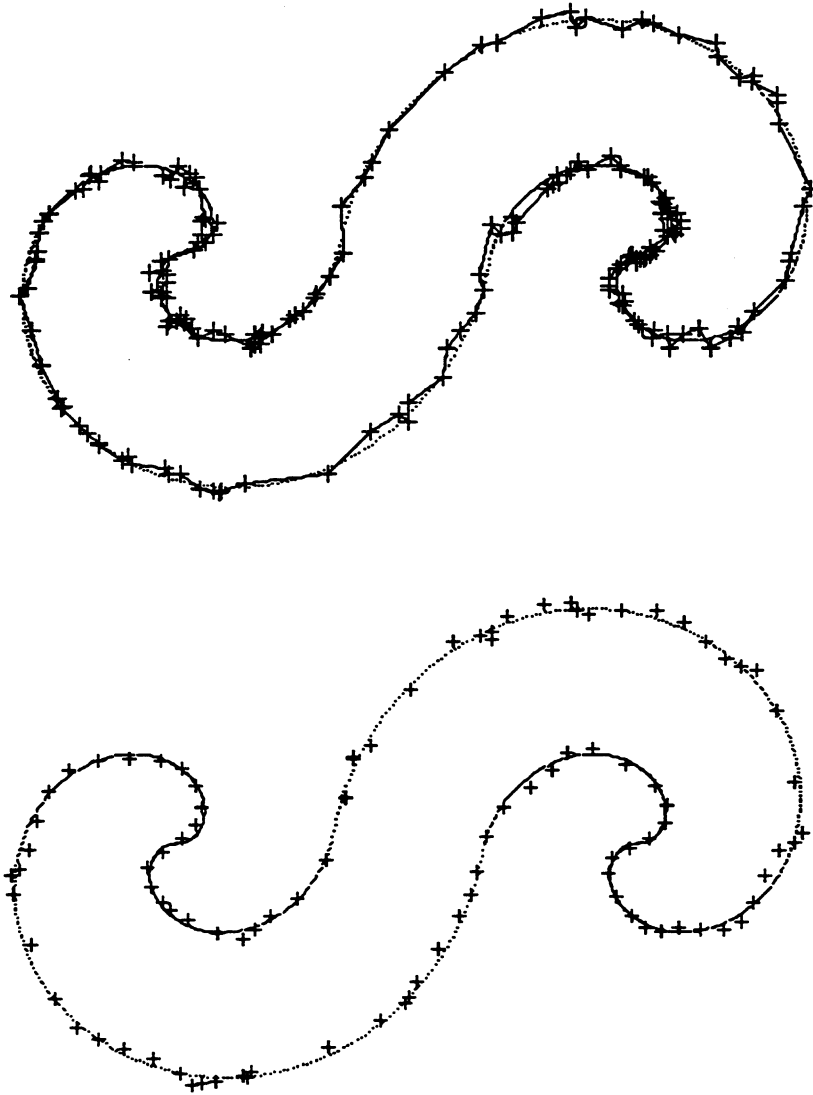


Figure 5.34 200 points, bruit = 1%, $\tau = 0$

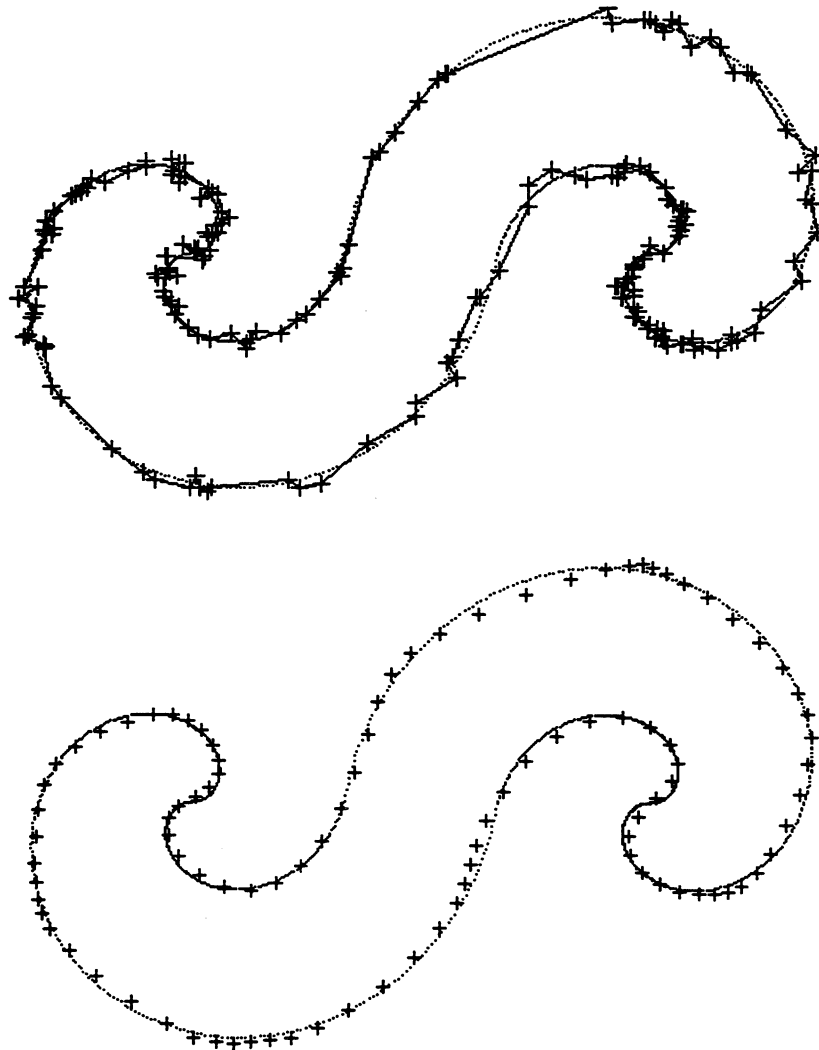


Figure 5.34 200 points, bruit = 1%, $\tau = 0,1$

Deuxième partie

Reconstruction de Surfaces

Après avoir examiné le cas des courbes, nous allons nous intéresser dans cette partie à celui des surfaces.

La problématique semble présenter des analogies avec le cas des courbes, à savoir, étant donné un ensemble de points $M_i, i = 1, \dots, n$ de l'espace \mathbb{R}^3 , connus seulement par leurs coordonnées, comment "lier" ces points entre eux par des relations de proximité naturelles induites par la surface à laquelle ils sont sensés appartenir. En fait, ce passage de \mathbb{R}^2 à \mathbb{R}^3 n'apporte pas uniquement une complexité de calcul supérieure, mais de nombreuses interprétations assez différentes les unes des autres qui peuvent être données.

De façon assez générale, on pourra dire que, dans le cas de \mathbb{R}^3 , il s'agit d'ordonner un ensemble de points de cet espace et en fait de déterminer une structure topologique cohérente sur les points, ce qui signifie en d'autres mots que le problème pourra être résolu en déterminant une triangulation de ces points, triangulation dont le résultat sera une surface polyédrique ayant pour sommets les points initiaux et possédant des propriétés de régularité.

Même lorsque l'on connaît un ordre lié à des valeurs de paramètres sur les points : $M_i = M(u_i, v_i), i = 1, \dots, n$ et $u_1 \leq \dots \leq u_n, v_1 \leq \dots \leq v_n$, le problème n'est pas trivial (cf. [Bar 85], [BFK 84] ou encore [Far 88] par exemple).

En plus, dans cette étude, les seules informations sur les points sont en principe leurs coordonnées dans \mathbb{R}^3 , et les méthodes diffèrent alors, puisqu'il n'est plus possible de calculer les équations paramétriques d'une surface. Nous rentrons alors dans la problématique que nous avons décrite au début de cette introduction.

Dans ce cadre là, le problème de reconstruction de surfaces dans l'espace a été à ce jour examiné par de nombreux auteurs.

Il se dégage de ces travaux deux grandes tendances, souvent liées au fait que les points M_i proviennent de l'échantillonnage d'une surface fermée ou d'une surface correspondant au graphe d'une fonction.

Il s'est ainsi développé des méthodes que nous qualifierons de "volumiques", puisqu'elles impliquent une tétraédisation des points, et des méthodes de type "surfactive" qui procèdent par une triangulation dans l'espace dont les simplexes sont des triangles plans.

Les points sont connus par leurs seules coordonnées

Nous pouvons citer, parmi les méthodes volumiques existantes, celle introduite par Boissonnat ([Boi 82] et [Boi 84]), qui consiste à calculer une tétraédisation de Delaunay de l'ensemble de points donné, puis à "sculpter" cette tétraédisation, c'est à dire ôter des tétraèdres, jusqu'à l'obtention d'un polyèdre qui comporte tous les points initiaux sur sa frontière.

Cette méthode, bien adaptée au cas des surfaces fermées, souffre néanmoins de l'emploi d'une tétraédisation en ce qui concerne la place mémoire nécessaire pour stocker les données et le temps de calcul lorsque l'ensemble initial contient un grand nombre de points (la complexité est $O(n^2 \log n)$, si n est le nombre de points donnés). D'autre part, des difficultés apparaissent lors du calcul (instabilités numériques voire algorithmiques) lorsque des points sont cosphériques ou presque.

Par rapport à un problème de dépouillement de mesures, où une fonction est connue en un nombre fixe de points de \mathbb{R}^3 , Lafranche [Laf 84] développe une méthode mixte pour la reconstruction de cette fonction.

La première étape est le calcul d'une tétraédisation de cet ensemble de points. La méthode consiste ensuite à se ramener dans \mathbb{R}^2 en coupant ce maillage par un plan.

A partir de valeurs associées alors aux points d'intersection des arêtes des tétraèdres dans ce plan, la fonction est reconstituée au moyen d'une méthode d'interpolation par éléments finis triangulaires.

Plus récemment, Hoppe et *al.* [HDD 92] ont développé une autre méthode mixte.

Une fois le partitionnement de la surface dans des cellules cubiques réalisé et la tétraédisation des points contenus dans chaque cube calculée, une fonction est définie comme étant la distance signée à la surface.

La première étape de la méthode consiste alors à calculer l'ensemble des zéros de cette fonction, qui fournit alors une estimation de la surface. Pour ce faire, un plan tangent orienté est associé à chaque point, et sert d'approximation linéaire locale de la surface.

Enfin, la deuxième étape consiste à déterminer une approximation de l'ensemble des zéros de la fonction distance par une surface simplectique, au moyen de la tétraédisation. La complexité de l'algorithme est alors $O(n^2)$ en fonction du

nombre de points n .

Présence d'informations complémentaires

D'autres approches sont envisagées, en particulier lorsque des informations complémentaires sont connues. Par exemple, de nombreux systèmes de mesures fournissent des données réparties sur des sections planes, et une structure initiale sur les points est donc connue. Les méthodes consistent alors à déterminer une triangulation entre chacune des sections planes successives, et traitent aussi des cas plus compliqués comme par exemple celui où la surface correspond à un réseau artériel provenant de techniques d'imagerie médicale, et où certaines sections planes comportent alors plusieurs composantes connexes disjointes qu'il faut reconstruire séparément. On pourra se référer aux travaux de Boissonnat [Boi 86] ou encore au tour d'horizon de Schumaker [Sch 90] à ce sujet. Précisons enfin que Boissonnat annonce une complexité égale à $O(n \log n + t)$, où n est le nombre de points et t le nombre de tétraèdres du résultat final (soit n^2 dans le pire des cas).

Méthodes surfaciques

En ce qui concerne les méthodes de type surfacique, et sous l'hypothèse que seules les coordonnées des points sont connues, on peut tout d'abord citer l'approche développée par Boissonnat [Boi 84]. Celle-ci consiste, après avoir calculé pour chaque point du nuage initial, l'ensemble de ses k plus proches voisins, à propager le contour d'une triangulation initiale sur l'ensemble de points en se servant d'une approximation locale de la surface par un plan tangent pour calculer la triangulation, et des relations de plus proches voisins préalablement calculées.

Si la complexité de l'algorithme est $O(n \log n)$ en fonction du nombre de points, la méthode dépend néanmoins fortement de la finesse de la discrétisation de la surface dans le voisinage d'un point par rapport au plus petit rayon de courbure en ce point. D'autre part, un autre défaut est que la méthode ne donne pas nécessairement le même résultat lorsqu'elle est appliquée à deux points voisins. Enfin, la convexité locale de la surface initiale peut être perdue lors de la reconstruction, puisque sont approximées les quantités du premier ordre relatives à la surface et non pas celles du second ordre.

En utilisant aussi l'arbre des k plus proches voisins associé à un nuage de

points, Favardin [Fav 93] a développé des méthodes permettant de reconstruire une surface, avec les mêmes hypothèses que précédemment.

En insérant les points au fur et à mesure, comme sur les techniques s'appliquant aux courbes, un algorithme, basé sur la maximisation de l'angle dièdre apparaissant entre deux faces adjacentes du polyèdre dont les sommets sont les points donnés, permet de reconstruire des surfaces fermées convexes.

La notion de proximité est rajoutée et prise en compte pour le traitement de surfaces non-convexes ou de genre supérieur à zéro. La complexité de ces méthodes est $n \log n$ dans le pire des cas, avec un prétraitement en $O(n \log n)$ pour construire l'arbre des k plus proches voisins.

Echantillonnage d'une surface qui est le graphe d'une fonction

De multiples méthodes ont été développées pour interpoler (ou approximer) le nuage de points donné, et sont toutes relatives au problème d' "interpolation de données dispersées", qui peut s'énoncer comme suit : étant donné un ensemble de points (x_k, y_k) , $k = 1, \dots, n$, de \mathbb{R}^2 , et un ensemble de valeurs (z_k) , $k = 1, \dots, n$, associées à chaque point (x_k, y_k) , construire une fonction $f = f(x, y)$ telle que $f(x_k, y_k) = z_k$, $k = 1, \dots, n$.

Pour les problèmes consistant à déterminer une telle fonction f qui soit C^1 au moins, les différents auteurs ont traité ce sujet à partir de points de vue très variés. Certains se sont uniquement intéressés à la construction de surfaces satisfaisant un critère de base, d'autres ont essayé de démontrer l'utilité de méthodes variées en termes de temps de calcul, de critères quantitatifs ou qualitatifs. On pourra se référer à Franke et Nielson pour une introduction et un tour d'horizon sur différentes méthodes relatives à ce problème, ou encore à Franke [Fra 82] pour la comparaison de 29 schémas d'interpolation.

En fait, la formulation est celle d'un problème d'interpolation, mais d'autres considérations (comme le très grand nombre de points) amènent à élaborer des méthodes différentes.

Dans le cadre de ce problème, on considère une fonction f linéaire par morceaux, et définie sur une triangulation d'un domaine Ω (de frontière $\partial\Omega$ polygonale) contenant les points (x_k, y_k) , $k = 1, \dots, n$. Comme f est définie de manière unique par ses valeurs aux trois sommets d'un triangle, la seule liberté réside alors dans le choix de la triangulation de l'ensemble de points.

Contrairement aux méthodes de triangulation classiques qui ne tiennent compte que du nuage de points $\{(x_k, y_k), k = 1, \dots, n\}$ dans \mathbb{R}^2 , la triangulation de cet ensemble devrait aussi tenir compte des valeurs $(z_k), k = 1, \dots, n$ associées. Dans cette idée, Dyn, Levin et Rippa ont introduit les triangulations dépendantes des données (data dependant triangulations) [DLR 90,1]. Le principe de ces méthodes est ainsi de modifier localement une triangulation initiale préalablement calculée de l'ensemble de points, selon un critère qualitatif portant sur la surface f interpolant les données. Ces critères de triangulation s'appliquent aux arêtes ou aux sommets de la triangulation. Le résultat final est alors obtenu en relevant la triangulation du plan $z = 0$ (si la fonction à partir de laquelle sont échantillonnés les points est du type $z = f(x, y)$) aux valeurs $(z_k), k = 1, \dots, n$ données.

De très nombreuses méthodes, la plupart heuristiques, ont été élaborées ; nous reviendrons plus en détail sur ce type de méthode au chapitre III.

Enfin, dans le cadre de ces triangulations dépendantes des données, Rippa [Rip 90] a obtenu un résultat surprenant et très intéressant. Si l'on définit en effet le "minimum de rugosité" d'une fonction comme le minimum d'une certaine semi-norme de Sobolev de cette fonction sur la triangulation de l'ensemble de points, Rippa démontre que l'unique triangulation réalisant ce minimum est la triangulation de Delaunay de l'ensemble des points dans le plan. Ce résultat est donc intéressant à double titre, puisque la triangulation de Delaunay d'un ensemble de points est une technique fort employée dans la pratique et se calcule facilement selon des algorithmes bien connus et efficaces, et puisqu'il confère à cette triangulation une propriété jusque là inconnue dans le domaine de l'approximation de surfaces.

Il faut remarquer pour finir sur ce sujet que la justification de la plupart des méthodes que nous avons présentées ici est essentiellement faite par des résultats expérimentaux. Sauf pour le cas du minimum de rugosité dont nous venons de parler et de Favardin [Fav 93], qui démontre, sous certaines hypothèses de régularité sur les données et la surface, que les méthodes qu'elle emploie permettent d'obtenir une suite de surfaces polyédriques qui converge uniformément vers la surface initiale, peu ou pas de preuves mathématiques sont données pour justifier

les méthodes employées.

Au vu des méthodes existantes, et compte tenu du résultat démontré par Rippa sur la triangulation de Delaunay dans le plan, nous avons alors développé une idée nouvelle qui a conduit aux méthodes que nous allons présenter.

Cette idée consiste ainsi à calculer une surface approchante, au lieu d'utiliser le plan $z=0$ ou un plan tangent local (qui ne permet pas de conserver une triangulation de Delaunay par projection), et sur laquelle il soit possible de calculer une triangulation de Delaunay des points initiaux projetés. L'idée directrice est d'obtenir ainsi une triangulation de Delaunay "approchée" de la surface initiale.

Notre prédilection pour une méthode de type "surfactive" par rapport aux approches "volumiques" se justifie par le gain de temps en ce qui concerne la complexité et donc les calculs, et une taille mémoire pour la structure de données moindre, puisque nous aurons à gérer alors des triangles au lieu de tétraèdres.

D'autre part, nous cherchons une surface qui nous permette une approximation plus globale que des plans localement tangents, et qui soit plus fiable quant à la "bonne" conservation (topologique ou qualitative) de la triangulation lors d'une opération de projection. L'emploi d'une méthode globale permettant de traiter l'ensemble des points ou au moins un grand nombre de ceux-ci en même temps est en effet souvent synonyme de gain de temps en ce qui concerne les calculs.

Nous nous sommes donc penchés ici sur le cas des surfaces de \mathbb{R}^3 , et dans le cadre des tests sur celui des surfaces radiales ou du type $z = f(x, y)$, dans un but de comparaison avec les triangulations dépendantes des données. Ces considérations préliminaires nous ont amené à développer une méthode basée sur l'approximation d'un nuage de points supposés appartenir à une telle surface par une portion de sphère. Les points sont ensuite projetés sur celle-ci et triangulés (selon la triangulation de Delaunay sphérique), avant d'être ramenés à leur position initiale.

En conservant la structure topologique qui les ordonnait sur la sphère (*i.e.* la triangulation), nous formons et nous obtenons ainsi une approximation par une surface C^0 , avec une topologie cohérente, de la surface initiale.

Le choix d'une sphère pour approcher l'ensemble de points donné s'est tout d'abord fait par rapport à l'existence d'une méthode d'approximation par un cercle robuste et peu coûteuse que nous avons exposée dans la première partie, et qui se

généralise aisément au cas de la sphère.

Ce choix s'est ensuite justifié par le fait que la sphère s'est avérée comme possédant de "bonnes" propriétés qui permettent de définir la triangulation de Delaunay sphérique, qui est aussi une triangulation géodésique sur cette surface, et de construire d'autre part des méthodes algorithmiques qui, comme nous le verrons, procèdent de manière similaire à celles existant dans le cas du plan pour la triangulation. Nous obtenons alors de ce fait une complexité et une place mémoire optimales pour le calcul de la triangulation.

Nous allons donc, dans cette partie, rappeler tout d'abord quelques définitions et propriétés d'une triangulation, et plus particulièrement de la triangulation de Delaunay dans le plan, avant d'effectuer un bref tour d'horizon des algorithmes et méthodes de calcul de cette triangulation.

Nous présentons ensuite les triangulations dépendantes des données et quelques critères utilisés pour calculer ces triangulations dans le plan, ainsi que l'algorithmique liée à ce type de méthode.

Nous définissons ensuite la notion de triangulation sur une sphère, et nous abordons ensuite la triangulation de Delaunay sphérique proprement dite, ainsi que la triangulation que nous lui associons pour les besoins de l'approximation d'une surface par une surface polyédrique à facettes triangulaires. Nous exposons à ce sujet leurs définitions et propriétés, avant de s'intéresser aux algorithmes qui permettent de les construire.

Nous décrivons ensuite la méthode que nous avons développée et ses domaines d'application sur les surfaces. Nous examinons enfin les résultats que nous avons obtenus par cette méthode, résultats qui sont comparés avec ceux que nous avons obtenus en implémentant les méthodes introduites par Dyn et *al.* dans le cadre des triangulations dépendantes des données.

Chapitre III

Triangulations : définitions, propriétés et exemples

Nous proposons dans ce chapitre préliminaire quelques rappels concernant la définition et les propriétés d'une triangulation d'un ensemble de points en dimension quelconque, ou, le plus souvent, pour nous rapprocher de la dimension qui nous intéressera vraiment par la suite, en dimension 2. Nous rappellerons ensuite les définitions et propriétés d'un diagramme de Voronoï et de la triangulation de Delaunay, avant d'effectuer un tour d'horizon des critères et méthodes algorithmiques de construction de telles triangulations.

Nous exposerons ensuite le problème plus spécifique de la triangulation d'une surface du type $z = f(x, y)$, et les "triangulations dépendantes des données" introduites par Dyn, Levin et Rippa.

Nous rappellerons à cet effet quelques critères de triangulation utilisés par ces derniers, ainsi que les méthodes et algorithmes de calcul.

1 Triangulations planes

Soit \mathcal{S} un ensemble de n points du plan, et \mathcal{E}_C l'enveloppe convexe de \mathcal{S} . Considérant ici les intérieurs des triangles nous pouvons rappeler la :

Définition 1.1 Un ensemble $T = \{T_i, i = 1, \dots, n_T\}$ de triangles est une **triangulation** de \mathcal{E}_C si :

- \mathcal{S} est l'ensemble de tous les sommets des triangles de T .
- Chaque arête d'un triangle de T ne contient que deux points de \mathcal{S} (ses extrémités).
- $\overline{\mathcal{E}_C} = \bigcup_{i=1}^{n_T} \overline{T_i}$, $T_i \cap T_j = \emptyset$, $i \neq j$.

Remarque 1.2 : Cette définition se généralise aisément à un espace de dimension N . Les éléments de la triangulation sont alors dans ce cas des simplexes de dimension N , et la notion d'arête est remplacée par celle de k -face, $k > 1$.

En général, pour un ensemble de points donnés, il n'existe pas de triangulation unique de cet ensemble, ce qui entraîne le besoin de distinguer, autant que possible, les différentes triangulations, et donc de les caractériser par des propriétés qu'elles respecteront. Comme nous le verrons par la suite, ces propriétés nous fourniront des critères de construction de ces triangulations.

Propriété 1.3 Relations d'Euler pour les triangulations.

Soit T une triangulation d'un ensemble \mathcal{S} de n points du plan. Soient t , a et n_b respectivement le nombre de triangles, d'arêtes et de points sur le bord de T (i.e appartenant à la frontière de l'enveloppe convexe de \mathcal{S}).

On a alors :

$$t = 2n - n_b - 2$$

$$a = 3n - n_b - 3 = t + n - 1$$

$$3 \leq n_b \leq n$$

Exemples de triangulations

Il est possible de définir de multiples autres façons de trianguler un ensemble de points que celles qui sont données en exemple ici.

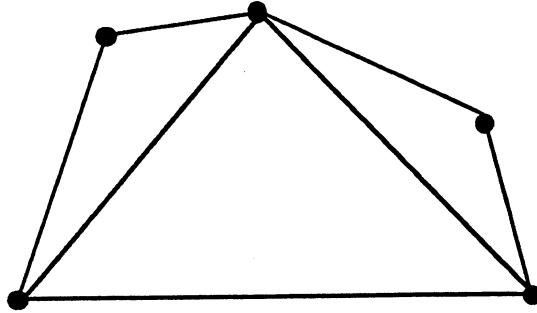


Figure 1.1 Triangulation optimale obtenue en minimisant la somme des longueurs des arêtes.

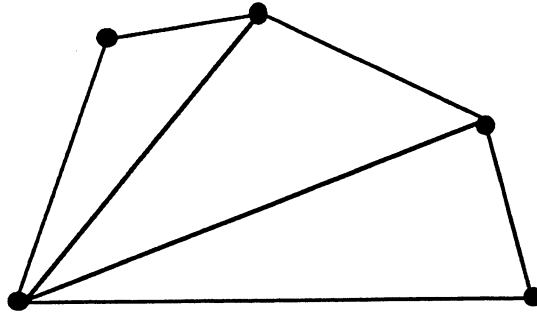


Figure 1.2 Triangulation de Delaunay : aucun cercle circonscrit à un triangle ne contient d'autres points que ses sommets.

2 Triangulation de Delaunay

Nous allons maintenant nous intéresser plus spécifiquement à la triangulation de Delaunay, puisque c'est en effet cette triangulation que nous allons calculer sur la surface de la sphère.

Avant de rappeler sa définition et ses propriétés, nous allons tout d'abord donner quelques définitions portant sur son dual, qui nous permettra de définir la triangulation de Delaunay elle-même.

2.1 Diagramme de Voronoï ([PS 85])

Soit \mathcal{S} un ensemble de n points d'un espace euclidien E^d de dimension d . On munit cet espace de la distance usuelle, et on note $\text{biss}[P_i, P_j]$ l'hyperplan bissecteur des points P_i et $P_j \in E^d$:

$$\text{biss}[P_i, P_j] = \{P \in E^d, d(P_i, P) = d(P_j, P)\} .$$

Si l'on se donne maintenant deux points P_i et P_j distincts de \mathcal{S} , l'ensemble des points de E^d plus proches de P_i que de P_j est le demi-espace noté H^+ , de frontière $\text{biss}[P_i, P_j]$ et contenant P_i :

$$H^+(P_i, P_j) = \{P \in E^d, d(P_i, P) \leq d(P_j, P)\}.$$

On peut alors définir :

Définition 2.1 On appelle polytope de Voronoï associé à $P_i \in \mathcal{S}$ (noté $PV(i)$) l'ensemble des points de E^d plus proches de P_i que de tout autre point de \mathcal{S} :

$$\begin{aligned} PV(i) &= \bigcap_{i \neq j} H^+(P_i, P_j) \\ &= \left\{ P \in E^d, d(P_i, P) \leq d(P_j, P), \text{ pour tout } P_i \neq P_j, P_j \in \mathcal{S} \right\} \end{aligned}$$

Propriété 2.2 Un tel polytope est non borné si et seulement si le point P_i appartient à l'enveloppe convexe de \mathcal{S} .

Si l'on considère maintenant l'ensemble \mathcal{S} , on peut définir l'ensemble des polytopes de Voronoï associés à chaque P_i :

Définition 2.3 On définit le *Pavage de Dirichlet* ou *Diagramme de Voronoï* d'un ensemble \mathcal{S} de points de E^d comme le partitionnement de E^d obtenu au moyen des polytopes de Voronoï associés à chaque point de \mathcal{S} .

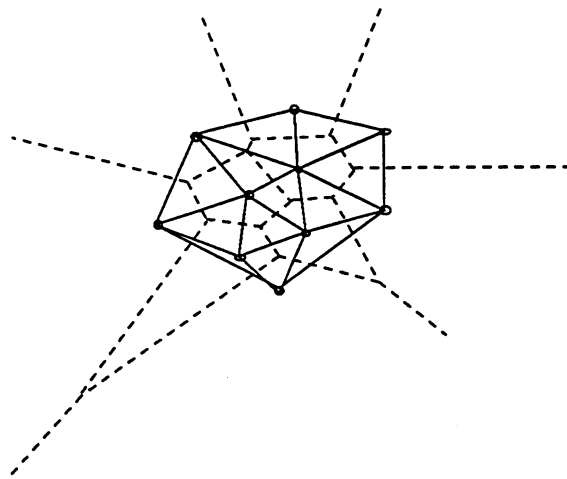


Figure 2.3 diagramme de Voronoï et triangulation de Delaunay

Enfin, on définit dans ce partitionnement la relation topologique "voisin" :

Définition 2.4 Deux points P_i et P_j de \mathcal{S} sont dits *voisins* dans le diagramme de Voronoï si leurs polytopes associés ont une $(d-1)$ -face en commun.

On trouvera, pour plus de détails, un exposé complet portant aussi sur l'algorithme liée à la construction de tels pavages dans [PS 85] pour \mathbb{R}^2 , ou encore dans l'article de [EPW 90] pour les problèmes de diagrammes dans \mathbb{R}^3 .

2.2 Triangulation de Delaunay dans le plan

Le diagramme de Voronoï comme il a été précédemment défini donne une définition de la triangulation de Delaunay :

Définition 2.5 Soit S un ensemble de n points du plan. Soit \mathcal{DV} le diagramme de Voronoï associé à cet ensemble de points.

Sous l'hypothèse qu'il n'existe pas quatre points de S cocycliques, on définit alors la triangulation de Delaunay de S comme le graphe dual de \mathcal{DV} , c'est à dire qu'une arête de Delaunay relie deux points P_i et P_j de S si et seulement si P_i et P_j sont voisins dans \mathcal{DV} (cf. figure 2.4).

Si quatre points sont cocirculaires ou plus, le graphe dual de Voronoï n'est alors pas une triangulation . On obtient au moins, en effet, un polygone dont les sommets sont tous les points appartenant au même cercle. On déduit de ce pavage la triangulation de Delaunay en triangulant le polygone associé aux points cocycliques, ce qui ne se réalise pas de manière unique. On définit alors une **triangulation complétée** de la triangulation de Delaunay, et on peut démontrer que toutes les triangulations complétées ainsi obtenues sont équivalentes, au sens où elles sont localement dans ce polygone toutes de Delaunay bien que différentes (cf. [Ede 87]).

Remarque 2.6 : S'il n'existe pas quatre points de S cocirculaires, alors la triangulation de Delaunay de S est unique, ce qui n'est pas vrai dans le cas contraire.

On supposera pour la suite qu'il n'existe pas quatre points de S cocycliques, assurant ainsi l'unicité de la triangulation.

2.3 Propriétés de la triangulation de Delaunay plane

On peut déduire de la définition de la triangulation de Delaunay une première propriété :

Propriété 2.7 *Propriété du cercle vide.*

Tout cercle circonscrit à un triangle d'une triangulation de Delaunay d'un ensemble de points S du plan ne contient pas d'autres points de S que les sommets de ce triangle.

Preuve : Soient $\Delta(P_1P_2P_3)$ un triangle de Delaunay et (C) son cercle circonscrit. Soit V le centre de (C) , i.e. le sommet de diagramme de Voronoï associé à P_1, P_2 et P_3 . (cf. figure 2.5)

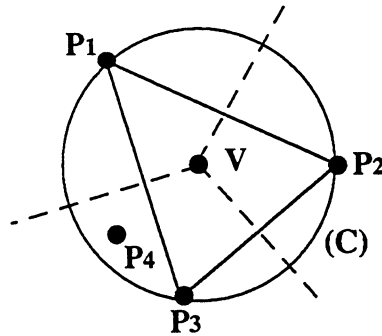


Figure 2.4 propriété du cercle vide

Si un point $P_4 \in (C)$, alors V est plus proche de P_4 que de P_1, P_2 ou P_3 . Dans ce cas, V est contenu dans $PV(4)$, et non dans $PV(1)$, $PV(2)$ ou $PV(3)$ d'après la définition d'un polygone de Voronoï ; on a donc une contradiction si $v \in PV(1) \cap PV(2) \cap PV(3)$. \square

Propriété 2.8 *La triangulation de Delaunay est l'unique triangulation à posséder la propriété précédente.*

Preuve : On trouve la preuve dans l'ouvrage de Edelsbrunner [Ede 87].

Ceci nous permet donc de donner une définition équivalente à la première :

Définition 2.9 *Une triangulation de Delaunay est une triangulation qui possède la propriété du cercle vide, sous l'hypothèse qu'il n'existe pas quatre points cocirculaires.*

Remarque 2.10 : Cette définition se généralise aisément à un espace de dimension quelconque d , en supposant qu'il n'existe pas $d + 2$ points cosphériques et en considérant une sphère de dimension $d - 1$.

Nous allons maintenant énoncer différentes propriétés de la triangulation de Delaunay dans le plan qui vont nous fournir des critères constructifs pour cette triangulation.

Pour ceci, nous allons nous intéresser au problème consistant à trianguler un quadrilatère Q strictement convexe, problème qui revient donc à choisir, pour le trianguler, l'une ou l'autre des diagonales de Q .

Critère 2.11 *Critère du maximum de l'angle minimum.*

Soit Q un quadrilatère strictement convexe, et T_1 et T_2 les deux triangulations possibles de Q . Soient α_i et β_i , $i = 1, \dots, 6$, respectivement les angles intérieurs des triangles formant T_1 et T_2 .

On choisira alors T_1 par rapport à T_2 pour trianguler Q si et seulement si

$$\max \left(\min_{i=1, \dots, 6} \alpha_i \right) > \max \left(\min_{i=1, \dots, 6} \beta_i \right) .$$

Remarque 2.12 : On a égalité entre les deux maxima si et seulement si les quatre sommets de Q sont cocirculaires, i.e. T_1 et T_2 sont deux triangulations de Delaunay équivalentes.

Critère 2.13 *Critère des diagonales*

Soit Q un quadrilatère strictement convexe, de sommets P_1, P_2, P_3 et P_4 . Soit P_0 l'intersection des deux diagonales $[P_1, P_3]$ et $[P_2, P_4]$. On note $r_i = d(P_0, P_i)$, $i = 1, \dots, 4$.

Alors on choisira la diagonale $[P_1, P_3]$ par rapport à $[P_2, P_4]$ pour la triangulation de Delaunay de Q si et seulement si :

$$r_1 \cdot r_3 < r_2 \cdot r_4 .$$

Remarque 2.14 : on aura l'égalité $r_1 \cdot r_3 = r_2 \cdot r_4$ si et seulement si P_1, \dots, P_4 sont cocirculaires.

Critère 2.15 **Critère du cercle vide.**

D'après la propriété du cercle vide, on choisira l'une ou l'autre des diagonales de Q pour la triangulation de Delaunay selon que le cercle circonscrit à chaque triangle contient ou non le quatrième sommet de Q .

Proposition 2.16 *Les 3 critères précédemment énoncés sont équivalents. De plus, si une triangulation d'un quadrilatère strictement convexe satisfait à l'un de ces critères, alors elle est de Delaunay.*

Preuve : Le critère du cercle vide découle directement de la propriété du même nom. Les deux autres critères proviennent de notions de géométrie élémentaire (angles inscrits interceptant le même arc et triangles semblables). \square

Remarque 2.17 : Ces différents critères, sauf celui du cercle vide, ne sont valables que pour le cas planaire.

2.4 Triangulation de Delaunay : algorithmes

Algorithmes de "swapping"

Ces algorithmes procèdent par des échanges d'arêtes locaux et successifs sur une triangulation initiale quelconque.

Comme nous l'avons vu au paragraphe précédent, plusieurs critères "locaux" nous permettent de construire la triangulation de Delaunay d'un quadrilatère strictement convexe.

Le fait remarquable, en ce qui concerne la triangulation de Delaunay, qui est que ces critères, définis au niveau local, sont vérifiés par l'ensemble de la triangulation et deviennent ainsi des critères globaux a pour conséquence que le critère de construction itéré localement à la triangulation donnera bien comme résultat la triangulation de Delaunay de l'ensemble des points.

Lawson [Law 77] à démontré la convergence de tels algorithmes vers la triangulation de Delaunay de l'ensemble des points, en faisant intervenir le principe d'optimisation locale de la triangulation selon un critère choisi.

C'est aussi ce type d'algorithme qui est employé dans le cadre des triangulations dépendantes des données, où, à partir d'une triangulation initiale, des échanges d'arêtes sont réalisés selon un certain critère.

Algorithmes incrémentaux

Une autre classe d'algorithmes, dits *incrémentaux*, procèdent par insertions successives des points. Leur déroulement est le suivant :

Initialisation : triangle construit au moyen de 3 sommets fictifs
tels que ce triangle contienne tous les points à insérer.

Itérer jusqu'à plus de points à insérer

Recherche d'un triangle t dans lequel le point s'insère.

Recherche de tous les triangles parmi les voisins de t dont
dont le cercle circonscrit contient le point courant.

Calcul de la frontière de l'union de ces triangles.

Détruire les triangles en question.

Créer les nouveaux triangles en reliant le point courant
à tous les sommets de la frontière.

Mettre à jour la triangulation.

Fin Itérer

Détruire les triangles ayant au moins un sommet fictif

Algorithme 1 : triangulation de Delaunay incrémentale

La complexité de tels algorithmes, si n est le cardinal de l'ensemble des points, est de l'ordre de $O(n^2)$. Il faut, dans le pire des cas, parcourir en effet tous les triangles de la triangulation à une étape donnée pour déterminer lequel contient le point à insérer.

Etant donné que le calcul de la triangulation de Delaunay se ramène au calcul d'une enveloppe convexe (ce qui peut se faire en $O(n \log n)$ ([PS 85])), cette complexité n'est pas optimale.

”Diviser pour Régner”

L'utilisation du principe de ”Divide and Conquer” permet par contre d'atteindre la borne optimale de $O(n \log n)$. Le principe de ces algorithmes est alors le

suisant :

- tri des points dans l'ordre lexicographique suivant leurs abscisses et leurs ordonnées.
- division de cet ensemble en deux sous-ensembles "droite" et "gauche" contenant chacun la moitié des points.
- construction récursive des triangulations de Delaunay des sous-ensembles "droite" et "gauche".
- fusion des parties droite et gauche triangulées.

On pourra aussi, pour plus de détails, se référer à [CH 90], [CR 84], [GS 85] ou encore [LS 80], ou encore à [Her 82] pour des triangulations en dimension n .

Algorithmes randomisés

Enfin, citons une tendance récente qui consiste à examiner la complexité d'algorithmes en moyenne sur l'ordre dans lequel sont introduites les données.

Des algorithmes dont la complexité n'est pas toujours optimale dans tous les cas, mais seulement en moyenne, sont donc utilisés.

Cette "randomisation" s'effectue principalement sur des algorithmes incrémentaux.

On pourra, à ce sujet, se référer aux travaux préliminaires de Clarkson et Shor ([Cla 87], [CS 89]), et notamment pour les résultats servant à étayer les preuves de complexité de tels algorithmes.

En ce qui concerne la triangulation de Delaunay, on pourra se référer par exemple aux travaux de Boissonnat et Devillers-Teillaud ([BDT 89]), ou de Guibas, Knuth et Sharir ([GKS 90]).

3 Triangulations Dépendantes des Données

DDT : Data Dependant Triangulations

Comme nous l'avons déjà cité dans l'introduction, ce type de triangulation a été étudié par Dyn, Levin et Rippa en 1990 et 1991 ([DLR 90,1],[DLR 90,2],[Rip 90]), de même que par Brown [Bro 91], qui présente d'autres critères (liés aux sommets de la triangulation et non pas aux arêtes) permettant de calculer ce type de triangulation. Plus récemment, Schumaker [Sch 93] construit de telles triangulations à partir de méthodes de recuit simulé, et Rippa [Rip 93] applique sur la triangulation la minimisation d'une fonctionnelle d'énergie basée sur les interpolants de Powell-Sabin.

Il faut noter dès à présent que ces techniques de triangulation des points ne peuvent s'appliquer qu'à des surfaces du type $z = f(x, y)$, puisque ces méthodes nécessitent une projection des points dans le plan $z = 0$, comme nous allons le voir.

3.1 Présentation du problème

Soit \mathcal{P} un ensemble de points du plan ($z = 0$), $\mathcal{P} = \{u_i = (x_i, y_i), i = 1, \dots, n\}$, et $Z = (z_i)$ un vecteur de \mathbb{R}^n , dont chaque composante z_i correspond à la valeur d'une fonction $f(x, y)$ de \mathbb{R}^2 dans \mathbb{R} au point u_i de \mathcal{P} .

Soit maintenant $T = (T_i)_{i=1}^m$ une triangulation de l'enveloppe convexe de \mathcal{P} , notée $\mathcal{E}_C(\mathcal{P})$.

On définit alors ([DLR 90,1]) :

Définition 3.1 Soit $S_1^0(T) = \{g \in C^0(\mathcal{E}_C(\mathcal{P})), g|_{T_i} \text{ soit un polynôme de degré } \leq 1 \text{ de } \mathbb{R}^3\}$.

On définit alors $f_T \in S_1^0(T)$ comme l'unique fonction linéaire par morceaux interpolant Z , i.e. $f_T(x_i, y_i) = z_i, i = 1, \dots, n$.

Remarque 3.2: Une fonction linéaire par morceaux est déterminée de manière unique par les valeurs aux sommets d'un triangle et donc de manière unique par le choix d'une triangulation T spécifique.

Le problème va alors consister à déterminer une triangulation de \mathcal{P} dans le plan qui, une fois ses triangles "remontés" aux valeurs z_i de Z , fournira la meilleure interpolation possible f_T de f (meilleure par rapport à un certain critère d'erreur).

3.2 Principe de calcul des D.D.T.

Le calcul d'une triangulation dépendante des données consiste alors, à partir d'une triangulation préliminaire de l'ensemble des points, à optimiser localement celle-ci selon un critère qui dépend des données (c'est à dire du vecteur Z associé aux points de \mathcal{P}), et ceci successivement jusqu'à l'obtention d'une triangulation optimale au sens de ce critère .

Deux types de critères apparaissent alors, qui tiennent compte soit des sommets soit des arêtes de la triangulation. L'obtention d'une triangulation optimale au sens du critère se déroule par contre de la même manière, et la présentation qui suit concernant les arêtes a_i d'une triangulation serait la même en considérant les sommets s_i de cette triangulation.

Pour ce faire donc, dans le cas d'un critère tenant compte des arêtes, on associe à chaque arête a_i , $i = 1, \dots, m$, située à l'intérieur de l'enveloppe convexe de la triangulation T une fonction coût : $c(f_T, a_i)$.

Si l'on associe à T la fonction coût : $C(f_T) = \sum_{i=1}^m |c(f_T, a_i)|$, on va alors rechercher et calculer la triangulation $T^* \in \mathcal{T}$ (où \mathcal{T} désigne l'ensemble des triangulations de \mathcal{P} , de cardinal fini si le nombre de points est lui aussi fini) telle que :

$$C(f_{T^*}) = \min_{T \in \mathcal{T}} c(f_T).$$

Nous allons donc examiner quelques critères du type NC^1 (Nearly C^1) présentés par Dyn et *all.* . Ces critères du type NC^1 se rapportent ainsi au raccord entre deux triangles adjacents, *i.e.* à l'arête commune à ces deux triangles .

3.3 Coûts associés aux arêtes

Soient T_1 et T_2 deux triangles adjacents d'une triangulation T , d'arête commune a . Soient P_1 et P_2 les deux plans définis par l'interpolation linéaire sur T_1 et T_2 , et soient n_1 et n_2 leurs normales respectives (orientées vers les $z \geq 0$).

Par rapport à la fonction f_T précédente, Dyn et *all.* associent deux critères de coût à l'arête a :

- **ABN (*Angle Between Normals*)**

Ce critère consiste en le calcul de l'angle entre les deux normales de triangles adjacents . Sa définition est la suivante :

$$\text{Coût } (f_T, a) = \theta = \arccos \left(\frac{\langle n_1, n_2 \rangle}{\|n_1\| \cdot \|n_2\|} \right),$$

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire usuel .

- **JND (*Jump in Normal Derivates*)**

On va calculer le saut entre les dérivées normales, ce qui s'exprime sous la forme :

$$\text{Coût } (f_T, a) = | \langle n, n_1 - n_2 \rangle |,$$

où n est un vecteur unitaire du plan $z = 0$ orthogonal à la direction de l'arête a .

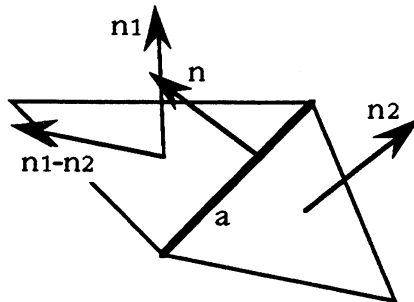


Figure 3.1 Jump in Normal Derivates.

Enfin, un dernier critère de triangulation plus global porte sur l'ensemble des triangles :

• **Minimum de rugosité**

Avec les notations précédentes, on définit la rugosité d'un vecteur de données Z par rapport à une triangulation T formée de m triangles T_i comme :

$$R(Z, T) = |f_T|_{T,1}$$

où $|\cdot|_{T,1}$ est la semi-norme de Sobolev, *i.e.* :

$$|g|_{T,1}^2 = \sum_{i=1}^m |g|_{T_i,1}^2 = \sum_{i=1}^m \int_{T_i} \left(\frac{\partial g}{\partial x} \right)^2 + \left(\frac{\partial g}{\partial y} \right)^2 dx dy.$$

Rippa ([Rip 90]) énonce alors un résultat original et très intéressant :

Théorème 3.3 *Soit \mathcal{P} un ensemble de points du plan distincts et non-colinéaires, et soit T_D la triangulation de Delaunay de cet ensemble de points.*

Si \mathcal{T} désigne l'ensemble des triangulations possibles de l'enveloppe convexe de \mathcal{P} , alors :

$$R(Z, T_D) = \min_{T \in \mathcal{T}} R(Z, T),$$

et ceci pout tout vecteur de données Z .

□

En réexprimant localement la rugosité d'une fonction f sur les deux triangulations possibles d'un quadrilatère strictement convexe Q — expression qui prend en compte alors le critère du cercle vide —, Rippa montre en effet que si T_D est la triangulation de Delaunay de Q , et T la triangulation obtenue en choisissant l'autre diagonale, alors

$$R(Z, T_D) \leq R(Z, T),$$

ce qui revient à dire que pour minimiser le coût considéré, il faut choisir, pour trianguler un quadrilatère, l'arête qui "est de Delaunay".

Et comme nous l'avons vu précédemment, si une triangulation est localement partout de Delaunay, alors elle l'est aussi globalement.

Remarque 3.4 : La propriété de la triangulation de Delaunay démontrée par Rippa est le seul résultat justifié mathématiquement en ce qui concerne les différents coûts présentés dans les travaux se rapportant à ce sujet, les autres critères de coût se justifiant uniquement par une expérimentation qualitative.

3.4 Coût associé aux sommets

Brown [Bro 91] associe à chaque sommet un coût basé sur l'idée de définir un analogue linéaire par morceaux de la courbure.

Soient s un sommet de la surface définie par la fonction f_T précédente, et $\{n_i, i = 1, \dots, m\}$ l'ensemble des normales des triangles de cette surface admettant s parmi leurs sommets. Si on pose $n = \sum_{i=1}^m \frac{n_i}{\|n_i\|}$, alors le coût associé à s est défini de la manière suivante :

$$\text{Coût}(f_T, s) = \sum_{i=1}^m \left(\cos^{-1} \left(\frac{\langle n, n_i \rangle}{\|n\| \cdot \|n_i\|} \right) \right)^2$$

3.5 Algorithmes et complexité du calcul d'une DDT

Les algorithmes de calcul de ces triangulations sont tous basés, sauf dans le cas de la triangulation de Delaunay, sur une méthode de post-optimisation d'une triangulation quelconque.

Nous allons donc citer une première définition qui va nous servir pour établir la notion de triangulation optimale :

Définition 3.5 [Sch 87] *Soit C un critère pour sélectionner la diagonale d'un quadrilatère strictement convexe et qui produit une meilleure triangulation au sens de ce critère.*

Alors on dira qu'une triangulation T est localement optimale par rapport à C si chaque quadrilatère strictement convexe, formé de deux triangles partageant une arête commune dans T est triangulé de manière optimale selon C .

Mais cette "optimalité" locale ne suffit pas pour décider quelle triangulation choisir au demeurant pour l'ensemble des points.

L'idée émise par Lawson ([Law 77]) à ce sujet permet de définir la notion de triangulation optimale.

Soient donc $T = \{T_i, i = 1 \dots, m\}$ une triangulation d'un ensemble de points, et $c(T_i)$ un coût associé au triangle T_i de T .

Si l'on associe à chaque triangulation T de cet ensemble de points un vecteur v_T dont les composantes, triées par ordre croissant, sont les coûts $c(T_i)$ associés à chaque triangle T_i de T , on peut alors ordonner les triangulations au moyen d'un ordre lexicographique sur les vecteurs v_T correspondants, c'est à dire :

$$\text{si } v_T = (c(T_i))_{i=1, \dots, m}, v_{T'} = (c(T'_i))_{i=1, \dots, m} \text{ et si } v_T \leq v_{T'}, \\ \text{alors il existe } k \in [1, m] \text{ tel que : } \forall j \geq k, c(T'_j) \geq c(T_j).$$

On peut alors énoncer, sous les notations précédentes :

Définition 3.6 *Soit \mathcal{T} l'ensemble des triangulations d'un ensemble de points.*

Alors une triangulation T^ est dite **triangulation optimale** si :*

$$v_{T^*} = \underset{T \in \mathcal{T}}{\text{Min}} v_T.$$

Remarque 3.7: Etant donné un ensemble fini de points, il existe au moins une triangulation optimale (le cardinal de \mathcal{T} étant fini), et, de plus, chaque triangulation optimale est localement optimale. Par contre, des triangulations localement optimales ne sont pas forcément optimales (cf. [Sch 87]).

On trouve alors deux types d'algorithmes, comme dans la plupart des problèmes d'optimisation combinatoire : des algorithmes procédant par modifications locales ([DLR 90,2]) et d'autres utilisant les techniques classiques de recuit simulé ([Sch 93]). Leurs descriptions sont les suivantes :

Initialisation : calcul d'une triangulation quelconque.

Tant que *la triangulation n'est pas localement optimale*

 Selectionner une arête a qui n'est pas localement optimale.

 Remplacer a par l'autre diagonale du quadrilatère formé

 par les deux triangles d'arête commune a .

 Mettre à jour la triangulation.

Fin tant que

Algorithme 1 : algorithme de post-optimisation pour le calcul d'une DDT

Dans cet algorithme, *netapes*, *nlimite* et *elimite* sont des entiers fixés.

Pour $k = 1$ jusqu'à *netapes*

Pour $l = 1$ jusqu'à *nlimite*

Tant que *le nombre de "bons" échanges est inférieur à elimite*

 Choisir au hasard une arête e .

Si *e peut être échangée*

Si *le nombre de "bons" échanges est ≥ 0*

 Faire l'échange.

Sinon

 Choisir un nombre aléatoire $0 \leq a \leq 1$.

Si *a est inférieur à une valeur de probabilité courante*

 Faire l'échange.

Fin si

Fin si

Fin si

Fin tant que

Algorithme 2 : algorithme de recuit simulé pour le calcul d'une DDT

Enfin, il a été démontré par Schumaker que la complexité de la phase de post-optimisation locale, si n est le nombre de points donnés, est de l'ordre de n^2 dans le pire des cas ([Sch 87]).

Chapitre IV

Triangulations de surfaces

Nous allons aborder dans ce chapitre le problème de la triangulation de surfaces.

Après avoir reprécisé les notions de triangle et de triangulation sur la surface d'une sphère, nous examinons plus particulièrement la triangulation de Delaunay sphérique. Nous énonçons sa définition et ses propriétés, avant d'introduire une triangulation que nous nommons triangulation à triangles plans, et que nous lui associons pour les besoins de l'approximation d'une surface par une surface polyédrique à faces triangulaires. Nous présentons ensuite les algorithmes de calculs de ces triangulation.

Enfin, nous exposons la méthode que nous employons pour l'approximation de surfaces, et que nous avons testée sur des surfaces radiales et sur des surfaces du type $z = f(x, y)$, avant de présenter quelques exemples et des comparaisons par rapport aux triangulations dépendantes des données, portant sur les résultats numériques obtenus avec certains critères d'erreurs en relation avec la qualité de l'approximation.

1 Triangulation de la sphère

Nous allons nous intéresser, dans cette partie, au problème consistant à trianguler un ensemble de points situés sur la surface d'une sphère, et ce plus particulièrement pour une sphère d'un espace euclidien de dimension 3.

La génération d'une grille triangulaire reliant des points appartenant à la surface d'une sphère a été examinée par Lawson ([Law 84]), Renka ([Ren 84]) ou encore Nielson et Ramaraj ([NR 87]), dans le but d'interpoler par une surface C^1 (définie sur la surface de la sphère) un ensemble de données associées à de tels points. Mais si des critères effectifs de calculs sont présentés pour le calcul du maillage triangulaire, aucun lien n'est établi avec la triangulation de Delaunay sur la surface de la sphère.

Après avoir précisé les notions de sphère et de métrique associée à une telle surface, nous rappellerons ici la définition et quelques propriétés des triangles sphériques. Nous nous intéresserons ensuite à la triangulation proprement dite d'un ensemble de points situés sur la surface d'une sphère, et par analogie au cas planaire, à l'extension des notions de diagramme de Voronoï et de triangulation de Delaunay sur la sphère.

Enfin, nous décrirons et évaluerons les algorithmes liés à la construction de telles triangulations.

1.1 Sphère et métrique associée

Dans tout ce paragraphe, nous nous placerons dans un espace euclidien de dimension 3, et nous appellerons S^2 une sphère de centre O et de rayon R .

Nous allons tout d'abord rappeler les notions qui vont nous permettre de définir une métrique sur S^2 (on pourra, pour plus de détails, consulter [Ber 77,1] et [Ber 77,2]).

On note $C(x, y)$ l'ensemble des courbes d'extrémités x et y d'un espace métrique quelconque M .

On suppose ici que M est connexe par arcs, ce qui équivaut à dire que $C(x, y) \neq \emptyset$, $\forall x, y \in M$.

Définition 1.1 Une courbe $f : [a, b] \rightarrow M$ est appelée un segment si $d(f(t), f(t')) = t' - t, \forall t, t' \in [a, b], t < t'$. On a : $\text{long}(f) = d(f(a), f(b))$.

Définition 1.2 Un espace métrique M est dit intrinsèque s'il est connexe par arcs et si :

$$d(x, y) = \inf \{ \text{long}(f) : f \in C(x, y) \} \quad \forall x, y \in E .$$

Il est dit excellent s'il existe un segment d'extrémités x et $y, \forall x, y \in M$.

Remarque 1.3 : si l'on muni S^2 de la métrique usuelle de \mathbb{R}^3 , S^2 n'est alors pas un espace métrique intrinsèque.

Ceci conduit à dire que la distance induite sur S^2 par \mathbb{R}^3 n'est pas satisfaisante. On va donc poser :

$$\overline{xy} = \arccos \langle (x|y) \rangle \quad \text{pour } x, y \in S^2$$

où $\langle \dots \rangle$ désigne le produit scalaire usuel.

Propriété 1.4 L'application $\overline{\cdot} : S^2 \times S^2 \rightarrow [0, \pi]$ est une métrique intrinsèque et donc excellente sur S^2 ; elle induit sur S^2 la même topologie naturelle que \mathbb{R}^3 .

D'autre part, si $x \neq \pm y$ sont deux points de S^2 , il existe un unique plus court chemin de x à y : l'arc de grand cercle de x à y . Si $x = -y$, les plus courts chemins sont tous les arcs de grands cercles joignant x à y .

Nous ne considérerons pas d'autres métrique sur S^2 pour la suite.

1.2 Triangles sphériques (définition géométrique)

On a, d'après Berger [Ber 77,2], la définition suivante :

Définition 1.5 On appelle triangle de S^2 (ou triangle sphérique) un triplet (x, y, z) de trois points de S^2 , linéairement indépendants en tant que vecteurs de \mathbb{R}^3 .

On notera $\Delta_S(x, y, z)$ un tel triangle.

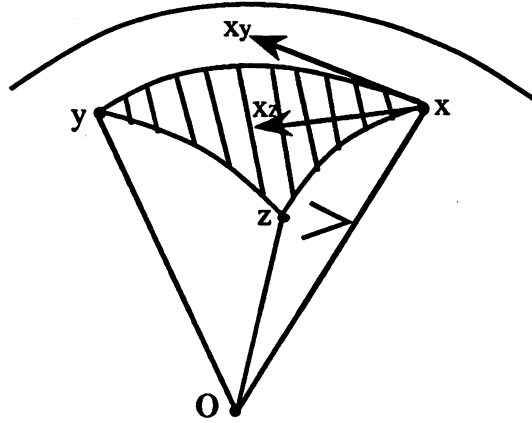


Figure 1.1 triangle sphérique

On peut alors confondre $\Delta_S(x, y, z)$ avec la partie de la sphère qui lui est naturellement associée : l'intersection des trois demi-sphères déterminées par les arcs de grand cercle passant par deux de ces points et contenant le troisième. On appellera cette partie **triangle sphérique** de sommets x, y, z (cf. figure 1.1).

On voit d'autre part sur la figure 1.1 que les angles dièdres du trièdre Ox, Oy, Oz sont égaux aux angles des vecteurs unités tangents aux arcs de grand cercle définis par x, y et z en ces sommets. On introduit donc la notation suivante : si x et y sont deux vecteurs linéairement indépendants de S^2 , on pose :

$$x_y = \frac{\lambda}{\|\lambda\|} \text{ avec } \lambda = y - \langle x, y \rangle x.$$

Définition 1.6 Soit $\Delta_S(x, y, z)$ un triangle sphérique.

- Les points x, y et z sont appelés les sommets de $\Delta_S(x, y, z)$.
- Les côtés de $\Delta_S(x, y, z)$ désignent les arcs de grand cercle \widehat{xy} , \widehat{yz} et \widehat{zx} .
- Les angles de $\Delta_S(x, y, z)$ sont les trois angles non orientés de vecteurs : $\alpha = \overline{x_y x_z} = \arccos(\langle x_y, x_z \rangle)$, $\beta = \overline{y_z y_x} = \arccos(\langle y_z, y_x \rangle)$ et $\gamma = \overline{z_x z_y} = \arccos(\langle z_x, z_y \rangle)$.

• Si les longueurs de côtés appartiennent à l'intervalle $]0, \pi[$, $\Delta_S(x, y, z)$ est dit triangle sphérique **propre**.

Remarque 1.7: On pourrait considérer α, β et $\gamma \notin]0, \pi[$, mais nous ne considérerons par la suite que des triangles sphériques propres.

1.3 Triangulation sur la surface d'une sphère

Après avoir défini géométriquement le triangle sphérique, nous allons donner dans ce paragraphe une définition de la triangulation d'un ensemble de points sur la surface d'une sphère.

Soit \mathcal{P} un ensemble de n points situés sur la surface d'une sphère \mathcal{S}^2 . On définit l'enveloppe convexe sphérique de \mathcal{P} par :

Définition 1.8 [Grü 67]

Un ensemble de points \mathcal{P} sur \mathcal{S}^2 est convexe si et seulement si le cône polyédrique, de sommet O le centre de \mathcal{S}^2 et s'appuyant sur les points M_i , $M_i \in \mathcal{P}$ est convexe.

On notera $\mathcal{E}_{CS}(\mathcal{P})$ l'enveloppe convexe sphérique de \mathcal{P} .

On a alors la définition suivante :

Définition 1.9 Soit \mathcal{P} un ensemble de points de \mathcal{S}^2 . Un ensemble $\mathcal{D} = \{\Delta_{\mathcal{S}}^i, i = 1, \dots, n_{\Delta}\}$ de triangles sphériques est une **triangulation** de $\mathcal{E}_{CS}(\mathcal{P})$ si :

- \mathcal{P} est l'ensemble de tous les sommets de \mathcal{D} .
- Chaque arête (i.e. arc de grand cercle) d'un triangle de \mathcal{D} ne contient que deux points de \mathcal{P} (ses extrémités).

$$\bullet \overline{\mathcal{E}_{CS}} = \bigcup_{i=1}^{n_{\Delta}} \overline{\Delta_{\mathcal{S}}^i} \quad \Delta_{\mathcal{S}}^i \cap \Delta_{\mathcal{S}}^j = \emptyset, \quad i \neq j.$$

Propriété 1.10 *Relations d'Euler*

Soit T_S une triangulation d'un ensemble \mathcal{P} de n points d'une sphère S^2 . Soient t , a et n_b respectivement le nombre de triangles, d'arêtes et de points sur le bord de T (i.e appartenant à la frontière de l'enveloppe convexe sphérique de \mathcal{P}).

On a alors, si les points sont tous situés dans un même hémisphère :

$$t = 2n - n_b - 2$$

$$a = 3n - n_b - 3 = t + n - 1$$

$$3 \leq n_b \leq n.$$

Si, par contre, leur enveloppe convexe est la surface de la sphère toute entière, on a :

$$t = 2n - 4$$

$$a = 3n - 6 = t + n - 2$$

$$n_b = 0.$$

2 Triangulation de Delaunay sur la sphère

Avant de donner la définition d'une telle triangulation, nous allons nous intéresser en premier lieu, comme dans le cas du plan, à la définition de son dual.

2.1 Diagramme de Voronoï

Nous allons élargir la définition du diagramme de Voronoï dans le plan à la sphère S^d , $d \geq 2$.

Soit \mathcal{P} un ensemble de points $P_i, i = 1, \dots, n$ de S^d .

On note $\text{biss}_{S^d}[P_i, P_j]$ la sphère S^{d-1} bissectrice des points P_i et P_j , $i \neq j$ de \mathcal{P} définie par :

$$\text{biss}_{S^d}[P_i, P_j] = \left\{ P \in S^d, \overline{P_i P} = \overline{P_j P} \right\}.$$

On peut alors définir l'ensemble des points plus proches de P_i que de P_j , i.e.

la demi-sphère de frontière biss $S^d [P_i, P_j]$ et contenant P_i par :

$$DS^+(P_i, P_j) = \left\{ P \in S^d, \overline{P_i P} \leq \overline{P_j P}, P_i \neq P_j \right\}.$$

La définition d'une région du pavage de Dirichlet associée à un point sur S^d est alors la suivante :

Définition 2.1 Soit \mathcal{P} un ensemble de points situés sur une sphère S^d . On appelle **région de Voronoï** $\mathcal{RV}_S(i)$ associée à un point P_i de \mathcal{P} l'ensemble des points de S^d plus proches de P_i que de tout autre point de \mathcal{P} :

$$\mathcal{RV}_S(i) = \bigcap_{i \neq j} DS^+(P_i, P_j).$$

On appelle alors P_i le **germe** de la région $\mathcal{RV}_S(i)$.

Ce qui nous permet d'énoncer la :

Définition 2.2 On définit le **pavage de Dirichlet** ou **diagramme de Voronoï** \mathcal{DV}_S d'un ensemble \mathcal{P} de points de S^d comme le partitionnement de S^d obtenu au moyen des régions de Voronoï associées à chaque point de \mathcal{P} .

La définition suivante nous sera utile pour la définition de la triangulation de Delaunay sphérique :

Définition 2.3 Deux points P_i et P_j de \mathcal{P} sont dits voisins dans le diagramme de Voronoï si leurs régions respectives ont une $(d-1)$ -face en commun.

Remarque 2.4 : Si un tel pavage est associé à un ensemble \mathcal{P} contenant au moins deux points, alors chacune de ses régions est convexe au sens donné par Grünbaum . Le théorème suivant est dû à Ash & Bolker ([AB 85]) :

Théorème 2.5 Soit P un sommet de valence n^1 d'un pavage de Dirichlet sur une sphère S^d . Soient P_1, \dots, P_n les germes des régions $\mathcal{RV}_S(1), \dots, \mathcal{RV}_S(n)$ de \mathcal{DV}_S ayant P comme sommet .

Alors les germes $P_i, i = 1, \dots, n$ appartiennent à une sphère S^{d-1} de centre P .

¹ Un sommet P d'un pavage de Dirichlet sur une sphère S^d est de valence n si $n(d-1)$ -faces de ce pavage sont incidentes en P .

Preuve : Si P_i et P_j sont deux voisins par rapport à P , alors $P \in \mathcal{RV}_S(i) \cap \mathcal{RV}_S(j)$, et donc $\overline{P_i P} = \overline{P_j P}$.

Comme chaque couple de régions $\mathcal{RV}_S(i)$ et $\mathcal{RV}_S(j)$ peut être rattaché par une suite de régions dont chacune est une voisine de son prédécesseur, les n distances $\overline{P_i P}$ sont toutes égales à la même constante r , et chaque P_i appartient donc à la même sphère de centre P et de rayon r . \square

2.2 Triangulation de Delaunay

Nous allons considérer dans ce paragraphe la triangulation de Delaunay définie sur une sphère \mathcal{S}^2 d'un espace euclidien de dimension 3.

Définition 2.6 Soit \mathcal{P} un ensemble de n points de \mathcal{S}^2 et soit \mathcal{DV}_S le diagramme de Voronoï associé à \mathcal{P} .

Sous l'hypothèse qu'il n'existe pas quatre points de \mathcal{P} coplanaires, on définit la triangulation de Delaunay sphérique \mathcal{TD}_S de \mathcal{P} comme le graphe dual de \mathcal{DV}_S , i.e. deux points P_i et P_j sont reliés par une arête de \mathcal{TD}_S si et seulement si ils sont voisins dans \mathcal{DV}_S .

Remarque 2.7 : La condition de coplanarité énoncée dans la définition est analogue à la condition de cocircularité du cas planaire, puisque une intersection sphère/plan est égale à un cercle sur \mathcal{S}^2 .

Ceci revient à considérer, par rapport au théorème 2.5, que les sommets de \mathcal{DV}_S sont tous de valence égale à 3.

La triangulation de Delaunay sphérique est donc formée de triangles sphériques, et ses arêtes sont des arcs de grand cercles joignant deux points de \mathcal{S}^2 . Nous allons lui associer, pour les besoins futurs de calcul et de construction, ce que nous allons appeler la triangulation de Delaunay à triangles plans associée à \mathcal{TD}_S .

Définition 2.8 Soit \mathcal{P} un ensemble de points de \mathcal{S}^2 et $\mathcal{TD}_{\mathcal{S}}$ la triangulation de Delaunay sphérique de \mathcal{P} .

On appelle $\mathcal{TD}_{\mathcal{P}}$ la **triangulation de Delaunay à triangles plans** associée à $\mathcal{TD}_{\mathcal{S}}$ la triangulation de \mathcal{P} obtenue en remplaçant chaque arête de $\mathcal{TD}_{\mathcal{S}}$ par la corde de grand cercle joignant les extrémités de l'arête sphérique.

Notation : Si P_i et P_j sont deux points de \mathcal{S}^2 , on notera $\widehat{P_i P_j}$ l'arête sphérique (*i.e.* l'arc de grand cercle d'extrémités ces deux points), et $P_i P_j$ l'arête (segment de droite) de la triangulation à triangles plans associée.

2.3 Propriétés des triangulations

Nous allons examiner dans ce paragraphe le problème consistant à trianguler un quadrilatère sphérique non dégénéré, c'est à dire tel que trois de ses sommets ne soient pas alignés. Ceci revient à choisir l'une ou l'autre de ses diagonales comme arête de la triangulation. On déduit facilement du théorème 2.5 une propriété de \mathcal{TD}_S :

Propriété 2.9 *Soit \mathcal{P} un ensemble de points de S^2 , tel qu'il ne possède pas quatre points coplanaires.*

La triangulation de Delaunay sphérique de \mathcal{P} est la triangulation telle que le cercle circonscrit sur la sphère à chacun de ses triangles ne contienne pas d'autres points de \mathcal{P} que les sommets de ce triangle.

Comme dans le cas des triangulations planes, c'est cette définition équivalente que nous prendrons dans la suite comme définition de la triangulation de Delaunay sphérique, *i.e* :

Définition 2.10 *Soit \mathcal{P} un ensemble de points de S^2 , ne possédant pas quatre points coplanaires.*

La triangulation de Delaunay sphérique de \mathcal{P} est la triangulation telle que le cercle circonscrit sur la sphère à chacun de ses triangles ne contienne pas d'autres points de \mathcal{P} que les sommets de ce triangle.

Nous allons maintenant énoncer la définition suivante qui concerne les arêtes d'une triangulation à triangles plans :

Définition 2.11 *Soit \mathcal{T}_S une triangulation sphérique d'un quadrilatère sphérique non dégénéré Q , sur une sphère S^2 de centre O .*

Soient d'autre part \mathcal{T}_P la triangulation à triangles plans associée à \mathcal{T}_S , et a une arête de \mathcal{T}_P d'extrémités deux sommets $P_i \neq P_j$ de Q .

*Alors on dira que a est **localement convexe** par rapport à Q si a appartient à la frontière de l'enveloppe convexe des sommets de Q et du centre O de la sphère.*

Remarque 2.12 : Ceci revient à dire que les deux triangles de $\mathcal{T}_{\mathcal{P}}$ appartiennent aux faces d'un polyèdre convexe de sommets (O, P_1, \dots, P_4) .

D'après la définition 2.11 , nous avons :

Propriété 2.13 Soit Q un quadrilatère sphérique non dégénéré, et soit $TD_{\mathcal{P}}$ la triangulation à triangles plans associée à la triangulation de Delaunay sphérique de Q .

Alors les arêtes de $TD_{\mathcal{P}}$ sont toutes localement convexes par rapport à Q .

Preuve : Soient P_1, P_2, P_3 trois points d'une sphère S^2 de centre O , formant un triangle sphérique.

Soit P_4 un quatrième point de S^2 . Examinons comment placer P_4 sur la sphère de telle façon que :

- 1°. (P_1, P_2, P_3, P_4) soit un quadrilatère, et :
- 2°. l'arête P_1P_3 n'appartiennent pas à la frontière de l'enveloppe convexe \mathcal{E}_C des points O, P_1, P_2, P_3 et P_4 .

Tout d'abord, soit U le demi-espace défini par le plan (P_1, P_2, P_3) et contenant le centre O de S^2 .

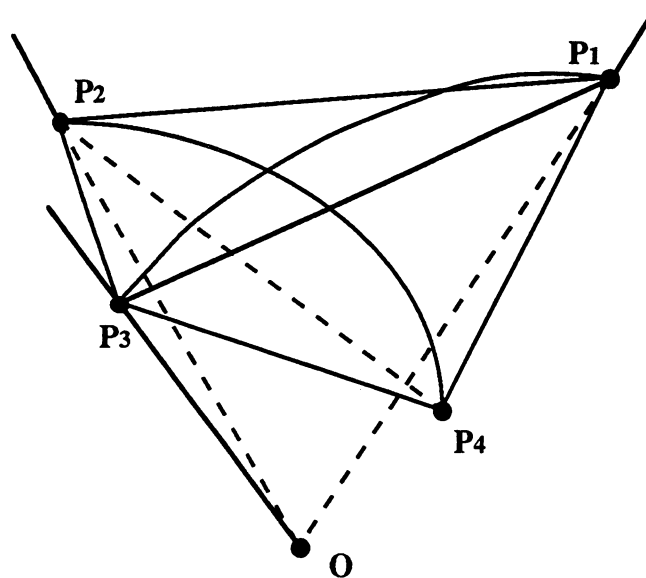


Figure 2.2 $P_4 \in U$

Si $P_4 \in U$, alors P_4 n'appartient pas au cône de sommet O et basé sur les points P_1, P_2 et P_3 (cf. figure 2.2).

P_4 est alors forcément situé à l'extérieur de l'enveloppe convexe des points O, P_1, P_2 et P_3 .

L'arête P_1P_3 appartient donc à la frontière de \mathcal{E}_C .

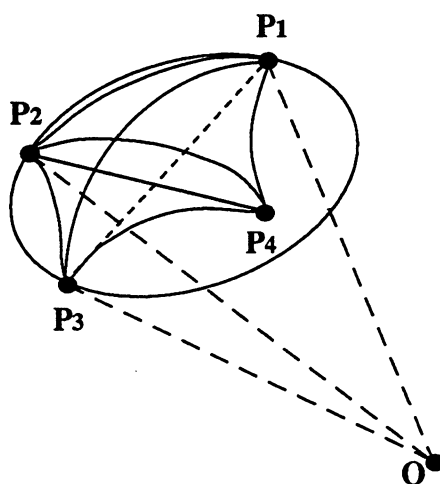


Figure 2.3 $P_4 \notin U$

Si maintenant $P_4 \notin U$, alors P_4 est aussi situé à l'intérieur du dièdre formé par les plans (O, P_1, P_2) et (O, P_2, P_3) , pour former ainsi un quadrilatère sphérique (P_1, P_2, P_3, P_4) .

P_4 est en même temps situé dans le demi-espace défini par le plan (O, P_1, P_3) et ne contenant pas le point P_2 (cf. figure 2.3).

Donc, P_1P_3 n'appartient pas à la frontière de l'enveloppe convexe des points O, P_1, P_2, P_3 et P_4 . De plus, P_4 appartient à l'intérieur du cercle circonscrit au triangle sphérique (P_1, P_2, P_3) .

En résumé, si P_1P_3 n'appartient pas à la frontière de l'enveloppe convexe des points O, P_1, P_2, P_3 et P_4 , alors P_4 appartient à l'intérieur du cercle circonscrit au triangle sphérique (P_1, P_2, P_3) .

Ou, en d'autres termes, si P_4 n'appartient pas à ce cercle, alors P_1P_3 appartient bien à la frontière de l'enveloppe convexe des points O, P_1, P_2, P_3 et

P_4 .

En conclusion, si la triangulation sphérique du quadrilatère (P_1, P_2, P_3, P_4) vérifie la propriété du cercle vide, alors toutes les arêtes de la triangulation à triangles plans associée à cette triangulation de Delaunay sphérique sont localement convexes par rapport à ce quadrilatère. \square

Corollaire 2.14 *Il y a équivalence entre la propriété du cercle vide pour la triangulation de Delaunay sphérique d'un quadrilatère non dégénéré et la propriété de convexité locale par rapport à ce quadrilatère pour les arêtes de la triangulation à triangles plans associée.* \square

Enfin, nous pouvons conclure sur une caractérisation d'une triangulation à triangles plans :

Proposition 2.15 *Soit \mathcal{P} un ensemble de points situés sur une sphère \mathcal{S}^2 .*

Alors les arêtes d'une triangulation à triangles plans associée à une $TD_{\mathcal{S}}$ de \mathcal{P} sont aussi des arêtes de l'enveloppe convexe de \mathcal{P} auquel on rajoute le centre de \mathcal{S}^2 .

Preuve : Au vu de la propriété 2.13 et du corollaire 2.14, ceci est vérifié, que O appartienne ou non à l'enveloppe convexe de \mathcal{P} . \square

2.4 Algorithmes pour la triangulation de la surface de la sphère

Dans le cas où l'on doit calculer une triangulation sur la surface de la sphère, deux choix d'algorithmes sont possibles.

On peut soit appliquer un calcul d'enveloppe convexe sur l'ensemble de points donnés (en utilisant par exemple la méthode développée par Preparata et Hong [PH 77]), soit des algorithmes de triangulation proprement dits, qui sont présentés ci-dessous.

Si les complexités théoriques du calcul d'une enveloppe convexe et d'une triangulation sont équivalentes, notre préférence ira néanmoins vers l'algorithme de triangulation. En effet, celui-ci, plus adapté au résultat final recherché, nous permet de gérer une structure de données plus spécifique et mieux adaptée au problème et à nos besoins (relations topologiques entre triangles, ...).

Nous examinons dans ce paragraphe les algorithmes de calcul des triangulations à triangles plans associées aux triangulations de Delaunay sphériques. Comme nous l'avons remarqué précédemment, un avantage de la triangulation sur la surface d'une sphère est qu'elle présente de très grandes analogies avec la triangulation dans le cas du plan ; de ce fait, les algorithmes employés pour sa construction sont équivalents à ceux utilisés dans le plan, ainsi qu'il va être possible de le constater par la suite.

Nous allons donc pouvoir obtenir des complexités en temps et en place mémoire égales à celle du cas plan, même si nous travaillons à fortiori dans l'espace.

La proposition 2.15 nous fournit un critère de calcul pour la triangulation à triangles plans associée à une triangulation sphérique :

Critère 2.16 Critère de convexité locale

Soit Q un quadrilatère sphérique convexe et non dégénéré, et dont les sommets ne sont pas tous coplanaires. Soient d'autre part T_1 et T_2 les deux triangulations à triangles plans possibles de Q , correspondant respectivement au choix d'une diagonale d_1 ou d_2 de Q .

On choisira alors T_1 par rapport à T_2 pour trianguler Q si d_1 est localement convexe et d_2 ne l'est pas.

En pratique, nous utilisons le critère de calcul suivant :

Critère 2.17 Critère effectif de calcul

Soit Q un quadrilatère sphérique convexe et non dégénéré, de sommets P_1, P_2, P_3 et P_4 numérotés ainsi en parcourant son contour.

Alors on choisira l'arête P_1P_3 pour trianguler Q si et seulement si

$$\det (\overrightarrow{P_1P_2}, \overrightarrow{P_1P_3}, \overrightarrow{P_1P_4}) < 0,$$

et l'arête P_2P_4 si et seulement si ce déterminant est > 0 .

Si le déterminant est nul, alors on choisira indifféremment l'une ou l'autre de ces deux arêtes.

Remarque 2.18 : $\det (\overrightarrow{P_1P_2}, \overrightarrow{P_1P_3}, \overrightarrow{P_1P_4}) = \langle \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}, \overrightarrow{P_1P_4} \rangle$ où \times désigne le produit vectoriel et $\langle \dots \rangle$ le produit scalaire usuel de \mathbb{R}^3 .

Le vecteur $\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}$ possède une direction normale au plan (P_1, P_2, P_3) , et donc l'arête P_1P_3 est choisie si P_4 est situé dans le demi-espace de frontière le plan (P_1, P_2, P_3) et contenant le centre de la sphère. : P_1P_3 est donc localement convexe.

On retrouve d'autre part le cas où les quatre points sont coplanaires, et les deux triangulations possibles sont équivalentes.

Nous séparons ici les deux cas où les points de l'ensemble à trianguler sont situés dans une seule demi-sphère ou non. Comme nous allons le voir, ces deux cas

engendrent des algorithmes légèrement différents.

2.4.1 Triangulation de la surface totale de la sphère

Avant de présenter une méthode de triangulation, nous définissons, pour simplifier la description des algorithmes, ce que nous appelons "zone d'insertion" d'un point dans une triangulation de la manière suivante :

Définition 2.19 Soit C un critère de triangulation et T^k une triangulation d'un ensemble de k points P_1, \dots, P_k qui satisfait à C .

Soit d'autre part P_{k+1} un point que l'on veut insérer dans la triangulation T^k pour former une triangulation T^{k+1} , et soit T_*^k le triangle de T^k contenant P_{k+1} .

On appelle alors **zone d'insertion** de P_{k+1} dans T^k l'union de T_*^k et des triangles T_i^k de T^k tels que le critère C ne soit pas satisfait pour tous les quadrilatères formés de P_{k+1} et des sommets de chaque T_i^k .

Remarque 2.20 : La définition signifie que tous les triangles de la zone d'insertion vont être modifiés par l'introduction du point P_{k+1} , et qu'ils ne seront pas des triangles de la nouvelle triangulation.

Remarque 2.21 : En pratique, pour le calcul de la triangulation à triangles plans associée à la triangulation sphérique, la zone d'insertion d'un point va correspondre à l'union de tous les triangles tels que le point à insérer soit situé dans le demi-espace de frontière le plan du triangle et ne contenant pas le centre de la sphère (d'après le critère de convexité locale), ou, redit de manière plus imagée, tels que le point soit situé "au-dessus" de ces triangles par rapport au centre de la sphère.

Une version incrémentale de l'algorithme de triangulation est alors la suivante :

Initialisation : tétraèdre construit au moyen de 4 points pris au hasard.

Itérer jusqu'à plus de points à insérer

Recherche d'un triangle dans lequel le point s'insère.

Calcul de la zone d'insertion.

Création des nouveaux triangles en reliant le point

aux sommets de la frontière de la zone d'insertion.

Suppression des triangles appartenant à la zone d'insertion.

Mise à jour de la triangulation.

Fin Itérer

Algorithme 1 : Triangulation de la surface totale de la sphère

2.4.2 Triangulation "ouverte" sur la sphère

Dans le cas où les points sont situés dans un seul hémisphère, l'hypothèse de ne considérer que des triangles sphériques propres pour former une triangulation sphérique nous oblige alors à ne pas trianguler toute la surface de la sphère, puisqu'il est possible d'obtenir alors dans le cas général des arêtes de longueur supérieure à $R\pi$, si R est le rayon de ladite sphère.

Plus généralement, dans les cas où nous voudrions appliquer la méthode pour la reconstruction de surfaces du type $z = f(x, y)$, il ne nous faudra pas obtenir une surface polyédrique fermée, mais ouverte.

Nous avons donc développé un algorithme incrémental différent du premier présenté, et qui nous permet de calculer une triangulation "ouverte" sur la sphère, ouverte au sens où il existe des points sur la sphère n'appartenant pas à l'adhérence de la triangulation.

La différence essentielle réside dans le fait qu'à une étape fixée de la triangulation, le nouveau point à insérer peut être situé à l'extérieur ou à l'intérieur de la triangulation. Nous procédons alors dans ces deux cas de la manière suivante :

- **point à l'intérieur** : le traitement est le même que dans le cas de la triangulation de la totalité de la surface de la sphère.

- **point à l'extérieur** : le fait que le point se trouve à l'extérieur de la triangulation signifie que celui-ci se retrouve aussi à l'extérieur de l'enveloppe convexe sphérique du nuage de points déjà triangulé. Il va donc nous falloir modifier l'enveloppe convexe pour obtenir la nouvelle triangulation.

Définition 2.22 *Soit a une arête de la frontière de l'enveloppe convexe sphérique d'un nuage de points appartenant tous à une même demi-sphère. Soit d'autre part un point P de la sphère n'appartenant pas à cette enveloppe convexe.*

*Alors l'arête a est dite **visible** par P si P appartient à la demi-sphère de frontière le grand cercle support de a et qui ne contient pas les autres points du nuage.*

Le calcul des nouveaux triangles de la triangulation va alors consister à relier le point que l'on insère aux sommets des arêtes de l'enveloppe convexe sphérique visibles par ce point, que nous appellerons **arêtes de bord visibles**.

L'algorithme est alors le suivant :

Initialisation : triangle formé au moyen de 3 points pris au hasard.

Itérer jusqu'à plus de points à insérer

Recherche d'un triangle dans lequel le point s'insère.

Si le point est à l'extérieur de la triangulation

Calcul de la ligne polygonale formée d'arêtes de bord visibles.

Création des nouveaux triangles en reliant le point
aux extrémités de ces arêtes.

Mise à jour de la liste des arêtes de bord.

Sinon

Calcul de la zone d'insertion.

Création des nouveaux triangles en reliant le point
aux sommets de la frontière de la zone d'insertion.

Suppression des triangles appartenant à la zone d'insertion.

Fin Si

Mise à jour de la triangulation.

Fin Itérer

Algorithme 2 : Triangulation "ouverte" sur la sphère

2.4.3 Insertion d'un point dans la triangulation

Nous allons prouver dans ce paragraphe que l'algorithme 2 produit à chaque étape d'insertion d'un point dans la triangulation une triangulation à triangles plans qui correspond bien à une triangulation de Delaunay sphérique.

Soient S_{n-1} un ensemble de $n - 1$ points d'une sphère S^2 , \mathcal{TD}_S^{n-1} une triangulation de Delaunay sphérique de S_{n-1} et \mathcal{TD}_P^{n-1} la triangulation à triangles plans associée.

Soit P un point de S^2 et $S_n = S_{n-1} \cup \{P\}$, $P \notin S_{n-1}$.

\mathcal{TD}_P^{n-1} étant donnée, on va construire \mathcal{TD}_P^n en créant des arêtes reliant le point P à certains sommets de S_{n-1} .

On peut tout d'abord construire une triangulation provisoire T_n de \mathcal{S}_n en insérant dans $\mathcal{TD}_{\mathcal{S}}^{n-1}$ les arêtes reliant P aux sommets de \mathcal{S}_{n-1} et ne coupant aucune arête de $\mathcal{TD}_{\mathcal{S}}^{n-1}$.

Définition 2.23 *Une arête de la triangulation $\mathcal{TD}_{\mathcal{P}}^n$ sera dite associée à la triangulation de Delaunay sphérique si elle est localement convexe et si aucune modification appliquée aux autres arêtes de cette triangulation ne peut aboutir à la suppression de cette arête.*

Montrons tout d'abord que toutes les arêtes créées en reliant P aux sommets adéquats de \mathcal{S}_{n-1} pour calculer la triangulation provisoire T_n sont associées à la triangulation de Delaunay sphérique.

Deux cas se présentent alors pour la situation du point P : P appartient ou n'appartient pas à l'enveloppe convexe sphérique $\mathcal{EC}_{\mathcal{S}}(\mathcal{S}_{n-1})$ de \mathcal{S}_{n-1} .

- 1^{er} cas : $P \notin \mathcal{EC}_{\mathcal{S}}(\mathcal{S}_{n-1})$:

Proposition 2.24 *Toutes les arêtes de $\mathcal{TD}_{\mathcal{P}}^n$ reliant P aux sommets des arêtes de bord visibles de $\mathcal{TD}_{\mathcal{S}}^{n-1}$ sont associées à la triangulation de Delaunay sphérique.*

Preuve : Soit PQ l'une de ces arêtes, et supposons que PQ est remplacée par une arête AB lors du calcul de $\mathcal{TD}_{\mathcal{P}}^n$. A et B doivent alors être situés chacun dans l'une des demi-sphères déterminées par l'arc de grand cercle support de l'arête sphérique \widehat{PQ} , et inversement pour P et Q .

Ainsi, AB coupe \widehat{PQ} strictement entre P et Q , ce qui implique que soit A soit B est situé à l'extérieur de l'enveloppe convexe de \mathcal{S}_{n-1} , ce qui est impossible. \square

- 2^{ème} cas : $P \in \mathcal{EC}_{\mathcal{S}}(\mathcal{S}_{n-1})$:

Proposition 2.25 *Toutes les arêtes de $\mathcal{TD}_{\mathcal{P}}^n$ reliant P aux sommets du triangle sphérique dans lequel il est inscrit sont associées à la triangulation de Delaunay sphérique.*

Preuve : P appartient ici à l'intérieur d'un triangle de $\mathcal{TD}_{\mathcal{S}}^{n-1}$; soit (ABC) ce triangle.

$\mathcal{TD}_{\mathcal{P}}^n$ est donc formée des arêtes de $\mathcal{TD}_{\mathcal{S}}^{n-1}$ et des arêtes PA, PB et PC .

Comme $\mathcal{TD}_{\mathcal{S}}^{n-1}$ est une triangulation de Delaunay, le cercle \mathcal{C} circonscrit à (ABC) ne contient pas d'autres points de \mathcal{S}_{n-1} dans son intérieur.

Supposons que \widehat{PA} soit remplacée par une arête \widehat{RS} lors de la triangulation de \mathcal{S}_n . Alors \widehat{PA} et \widehat{RS} doivent se couper strictement, R et S sont à l'extérieur de \mathcal{C} et A doit être à l'extérieur du cercle \mathcal{C}' circonscrit à (PRS) .

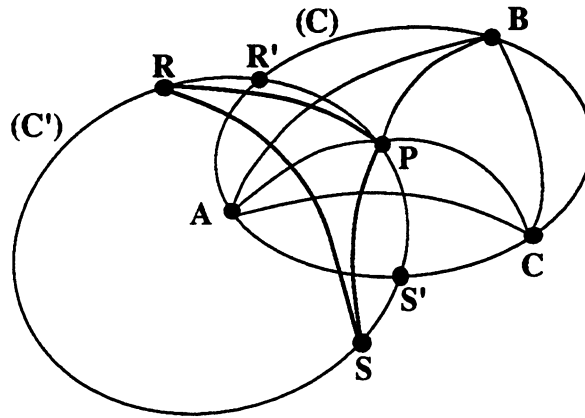


Figure 2.4 intersection \mathcal{C} , \mathcal{C}'

Soient R' et S' les points d'intersection de \mathcal{C} et \mathcal{C}' (Cf. figure 2.4). L'intersection est non-vide, car R et S sont à l'extérieur de \mathcal{C} , et donc dans le même demi-espace de frontière le plan de \mathcal{C} .

L'arc de cercle de \mathcal{C}' $\widehat{R'PS'}$ est alors intérieur à \mathcal{C} car P est à l'intérieur de \mathcal{C} , et R' et S' appartiennent à \mathcal{C}' .

Donc l'arc de cercle de \mathcal{C} compris entre R' et S' et qui est situé à l'intérieur de \mathcal{C}' contient le point A . Ceci signifie donc que A est à l'intérieur de \mathcal{C}' , ce qui est impossible. □

Enfin, le passage de la triangulation provisoire T_n à la triangulation finale $TD_{\mathcal{P}}^n$ nous oblige à tester les triangles de $TD_{\mathcal{P}}^{n-1}$ situés dans le voisinage du triangles dans lequel P est inscrit, ou dans le voisinage des arêtes de bord. Ces tests, comme nous l'avons vu dans la définition de la zone d'insertion, s'appliquent aux quadrilatères convexes formés par les sommets d'un triangles et le point P .

Pour un quadrilatère sphérique sur lequel on applique donc le critère de convexité locale pour le trianguler, nous pouvons énoncer, avec les notations précédentes :

Théorème 2.26 Soient (ABC) un triangle de la triangulation $TD_{\mathcal{P}}^{n-1}$ de S_{n-1} , et $P \in S_n = S_{n-1} \cup \{P\}$, qui forme un triangle (BCP) adjacent à (ABC) sur la sphère.

Si l'arête BC (\widehat{BC}) est remplacée par l'arête PA (\widehat{PA}) lors du calcul de la triangulation $TD_{\mathcal{P}}^n$ (TD_S^n) de S_n , alors PA est associée à la triangulation de Delaunay sphérique.

Preuve : la démonstration du théorème 2.26 est identique à celle de la proposition 2.25 . □

Comme les triangles de la zone d'insertion d'un point P , et qui ne contiennent pas P , sont tous des triangles tels que la diagonale du quadrilatère qu'ils forment avec P sera remplacée par la diagonale d'extrémité P , le théorème précédent et la proposition 2.25 montrent donc que la triangulation obtenue en reliant le point P aux sommets de la frontière de la zone d'insertion et aux sommets du triangle sphérique dans lequel il est inscrit est bien une triangulation de Delaunay. De même, la proposition 2.24 conduit à une conclusion identique lorsque le point P est situé à l'extérieur de la triangulation.

2.5 Complexité des algorithmes

Structure de données

Les points P_k , $k = 1, \dots, n$ sont donnés par leurs coordonnées et stockés dans une liste de pointeurs simplement chaînée, chaque pointeur contenant les coordonnées du point et un lien vers le pointeur suivant.

Un triangle est alors défini comme une structure contenant :

- les liens vers les 3 points qui forment ses sommets, numéroté dans le sens trigonométrique : P_0, P_1, P_2 .
- les liens vers les au plus 3 triangles adjacents T_0, T_1, T_2 (lien nul si l'arête partagée est une arête de bord).
- un lien vers un triangle suivant.

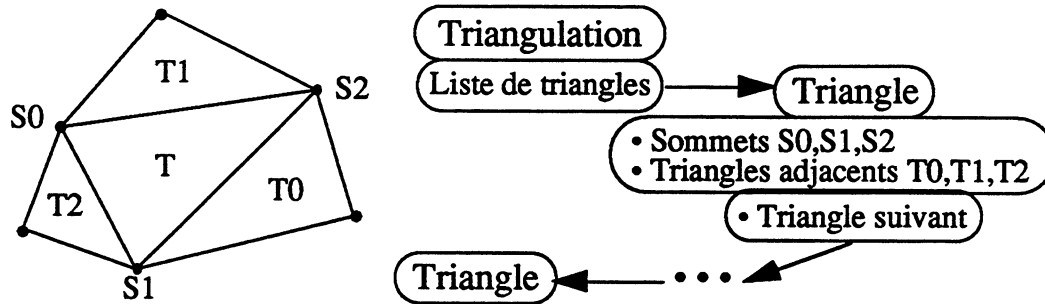


Figure 2.5 Structure de donnée pour la triangulation

La triangulation de l'ensemble de points est alors une liste simplement chaînée de triangles.

Enfin, pour améliorer la rapidité des calculs, nous utilisons une liste simplement chaînée d'arêtes de bord, chaque arête de bord étant une structure contenant :

- un lien vers chacun de ses points extrémités.
- un lien vers le triangle auquel elle appartient.
- un lien vers une arête de bord suivante.

Nous avons donc :

Proposition 2.27 *Si n est le nombre de points de l'ensemble initial à trianguler, alors la structure de données occupe une place mémoire de l'ordre de n .*

Preuve : D'après les relations d'Euler, le nombre d'arêtes de bord a_b est tel que : $3 \leq a_b \leq n$, et le nombre de triangles est égal à $2n - a_b - 2$. \square

Complexité temporelle

Les deux algorithmes présentés ici sont équivalents aux algorithmes employés dans le cas du plan, et procèdent de la même manière. Leur complexité étant bien connue ([GS 85] ou [PS 85] par exemple), nous pouvons alors directement énoncer :

Proposition 2.28 *La complexité des deux algorithmes présentés ici est de l'ordre de n^2 dans le pire des cas.*

Remarque 2.29 : Nous avons présenté ici deux algorithmes qui sont du type incrémental, et dont la complexité n'est pas optimale, puisque comme nous l'avons vu au chapitre précédent, cette borne est de l'ordre de $n \log n$.

Il est néanmoins possible d'atteindre une complexité théorique optimale par une implémentation du type "diviser pour régner". Le déroulement de l'algorithme, pour un nuage de points situés dans le même hémisphère, est alors le même que celui fonctionnant dans le plan :

- Subdivision récursive du nuage en deux parties droite et gauche (on peut pour cela établir un ordre lexicographique sur les points en considérant leurs angles associés en coordonnées sphériques).

- Triangulation sphérique de chaque partie et raccord des triangulations obtenues.

Il semble enfin tout à fait possible d'adapter les algorithmes randomisés développés dans le cas du plan par Boissonnat et Devillers-Teillaud [BDT 89] au cas de la triangulation sur la surface de la sphère.

3 Méthode de reconstruction, exemples et comparaisons

3.1 Description de la méthode de reconstruction

Nous avons utilisé les techniques d'approximation d'un nuage de points par une sphère et la triangulation d'un ensemble de points situés sur une telle surface pour développer une méthode originale dans le cadre de la reconstruction de surfaces.

Bien que cette méthode soit destinée à des surfaces données par les seules coordonnées de points appartenant à leur frontière et sans avoir d'autres informations, nous l'avons testée pour comparaison avec d'autres méthodes sur des surfaces du type $z = f(x, y)$.

Une surface va ainsi être approchée par une surface polyédrique à faces triangulaires, et dont les sommets sont les points échantillonnés sur la surface initiale.

Comme nous l'avons vu au paragraphe 2.2 de ce chapitre, nous allons donc calculer la triangulation à triangles plans associée à la triangulation sphérique, une fois le nuage de points initial approché par une sphère et les points projetés sur cette dernière.

C'est cette triangulation à triangles plans qui nous fournira la surface polyédrique approchant la surface lorsque les points auront été ramenés à leur position initiale.

La méthode se déroule donc de la manière suivante :

Données initiales : nuage de points échantillonnés sur une surface.

Approximation du nuage de points par une sphère.

Projection des points sur la sphère.

Calcul de la triangulation à triangles plans
de l'ensemble des points projetés.

Calcul de la surface approchante en ramenant les sommets
à leur position initiale.

Algorithme 5 : Reconstruction de surfaces

Nous avons enfin pour cette méthode de reconstruction la complexité suivante :

Proposition 3.1 **Complexité optimale en temps et mémoire**

La méthode de reconstruction de surfaces telle qu'elle est décrite ci-dessus possède une complexité de l'ordre de $n \log n$ dans le pire des cas, pour une taille mémoire de l'ordre de n , lorsque elle est appliquée à un nuage de n points échantillonnés sur une surface.

Preuve : Les opérations d'approximation par une sphère et de projection sont linéaires en fonction du nombre de points (nous avons vu dans la première partie que le coût de l'approximation est celui du précalcul de la matrice, c'est à dire $O(n)$), et donc la complexité totale dépend de celle de la phase de triangulation, ce qui peut se faire en temps optimal $O(n \log n)$, et une place mémoire en $O(n)$, comme nous l'avons vu précédemment. □

3.2 Expérimentations et comparaisons avec les triangulations dépendantes des données

Nous avons appliqué la méthode de reconstruction décrite ci-dessus à certaines surfaces, radiales ou non, qui se prêtaient assez bien à l'approximation par une unique sphère.

Nous avons ainsi tout naturellement commencé par examiner les résultats obtenus sur une demi-sphère, avant de bruitez cette dernière. Parmi les tests que nous avons effectués, nous présentons aussi dans ce paragraphe les résultats obtenus sur une surface radiale, et ceux relatifs à une surface plus générale.

Critères de comparaison

Comme nous l'avons précédemment évoqué, nous avons comparé les résultats obtenus par notre méthode de reconstruction avec ceux obtenus par les méthodes de triangulations dépendantes des données, calculées à partir des critères ABN et JND présentés au chapitre III.

Nous avons basé la comparaison sur trois principaux critères qui sont les suivants :

- *Distance entre la surface approchante et la surface initiale*

Ce critère est significatif de la bonne ou de la mauvaise approximation de la surface initiale sur laquelle les points sont échantillonnés par la surface polyédrique approchante.

Il consiste ainsi, à partir d'une grille de points régulière couvrant le domaine plan sur lequel est définie une fonction f engendrant la surface initiale F , à calculer la moyenne des distances entre le point correspondant (mêmes abscisse et ordonnée) et appartenant à F , et le point correspondant appartenant à l'un des triangles de la surface polyédrique approchante F_{approx} , d'équation $z = f_{approx}(x, y)$.

Si l'on se donne les fonctions f et f_{approx} définies sur un domaine plan et une grille de n^2 points (x_i, y_j) , $1 \leq i, j \leq n$ discrétisant ce domaine, l'erreur ε_z est donc définie par la formule :

$$\varepsilon_z = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n | f(x_i, y_j) - f_{approx}(x_i, y_j) |$$

Et donc, plus cette erreur sera faible et plus la surface approchante sera "proche" en moyenne de la surface initiale.

• *Erreur radiale*

Lorsque la surface initiale approchée est une surface radiale, nous avons aussi calculé, d'une manière similaire à l'erreur ε_z l'erreur radiale ε_{rad} commise lors de l'approximation.

Si l'on se donne donc une surface radiale d'équation $r = f(\theta, \varphi)$ en coordonnées sphériques, et une discrétisation $(\theta_i, \varphi_j), 1 \leq i, j \leq n$ des deux angles θ et φ , l'erreur ε_{rad} est donc calculée comme suit :

$$\varepsilon_{rad} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n | f(\theta_i, \varphi_j) - r_{approx}(\theta_i, \varphi_j) |$$

où $r_{approx}(\theta_i, \varphi_j)$ désigne la distance entre le centre de la surface et l'intersection du rayon $f(\theta_i, \varphi_j)$ avec un triangle de la surface approchante.

Cette erreur mesure donc, comme le premier critère, la qualité de l'approximation, mais du point de vue radial.

• *Longueur moyenne des arêtes*

Enfin, nous avons utilisé un critère de comparaison qui consiste à mesurer la longueur moyenne des arêtes de la surface polyédrique approchante obtenue par les différentes méthodes.

Nous avons mis en place cette quantité dans l'idée de mesurer une certaine "régularité" de la triangulation, régularité au sens où, en moyenne, les triangles obtenus seront les moins "allongés" possibles.

Ce critère de comparaison a été motivé par le fait que des triangles les "plus équilatéraux possibles" sont souvent préférés pour l'interpolation par de nombreux auteurs à des triangles plus allongés (cf. par exemple [BZ 70] ou [Gre 75]), puisque les bornes d'erreur de l'interpolation s'accroissent si les triangles sont fins et allongés. Ceci se justifie d'autant plus lorsque les surfaces approchées sont relativement régulières.

Exemple de la demi-sphère

Nous avons comparé notre méthode par approximation sphérique avec les méthodes de triangulations dépendantes des données sur une demi-sphère, cas qui

nous est assez favorable s'il en est (du moins pour la qualité de la sphère approchant le nuage de points initial), mais néanmoins intéressant car représentant une surface régulière selon plusieurs points de vue (courbures,...).

Nous avons ainsi représenté sur un premier graphique l'erreur ε_z pour les quatre méthodes testées, en fonction du nombre de points de l'échantillon initial.

Est représentée ensuite sur un deuxième graphique, l'erreur radiale ε_{rad} commise. Comme il est possible de le remarquer sur ces deux graphiques, l'approximation par la triangulation de Delaunay sphérique donne de meilleurs résultats, surtout en ce qui concerne l'erreur radiale, et lorsque, comme on pouvait s'y attendre, le nombre de points augmente, nous donnant ainsi une approximation de plus en plus précise de cette demi-sphère.

Quant à la longueur moyenne des arêtes, elle s'avère inférieure sur la surface polyédrique générée par notre méthode par rapport aux autres surfaces approchantes, comme on peut le constater sur le troisième graphique. Les figures suivantes représentent les surfaces approchantes obtenues au moyen de la triangulation sphérique et de la triangulation "ABN".

On peut alors remarquer que ces constatations signifient que les triangulations suivant les critères ABN et JND ne respectent pas la convexité de cette surface, puisque la triangulation à triangles plans fournit bien une surface approchante convexe.

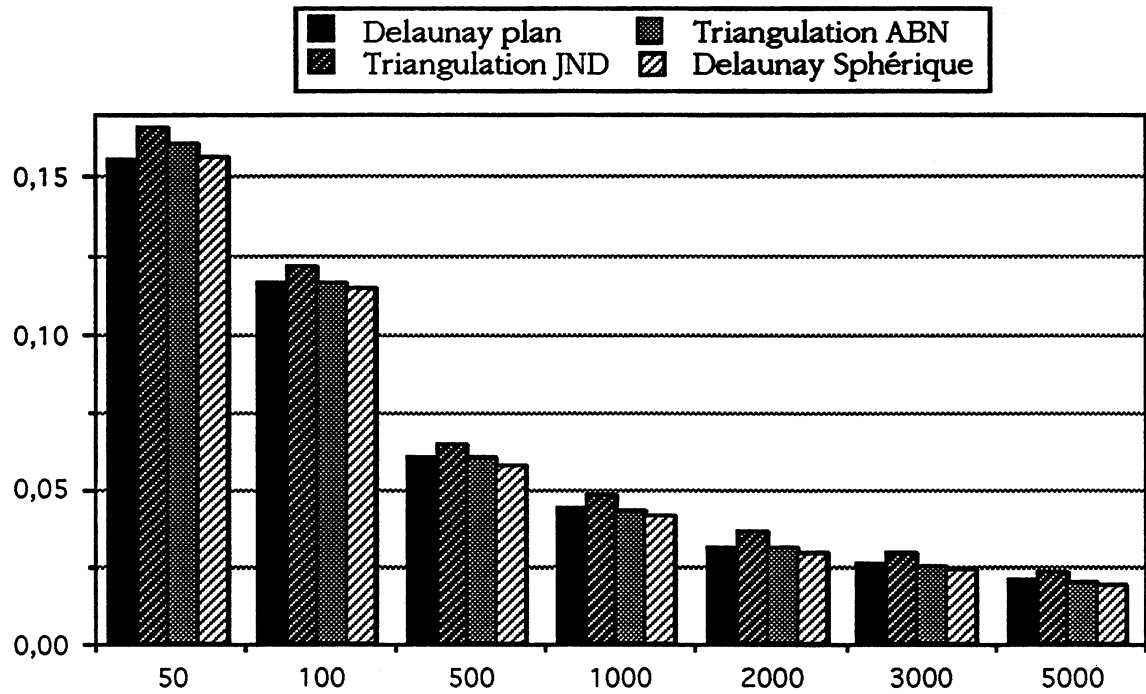


Figure 3.1 Longueur moyenne des arêtes sur la demi-sphère.

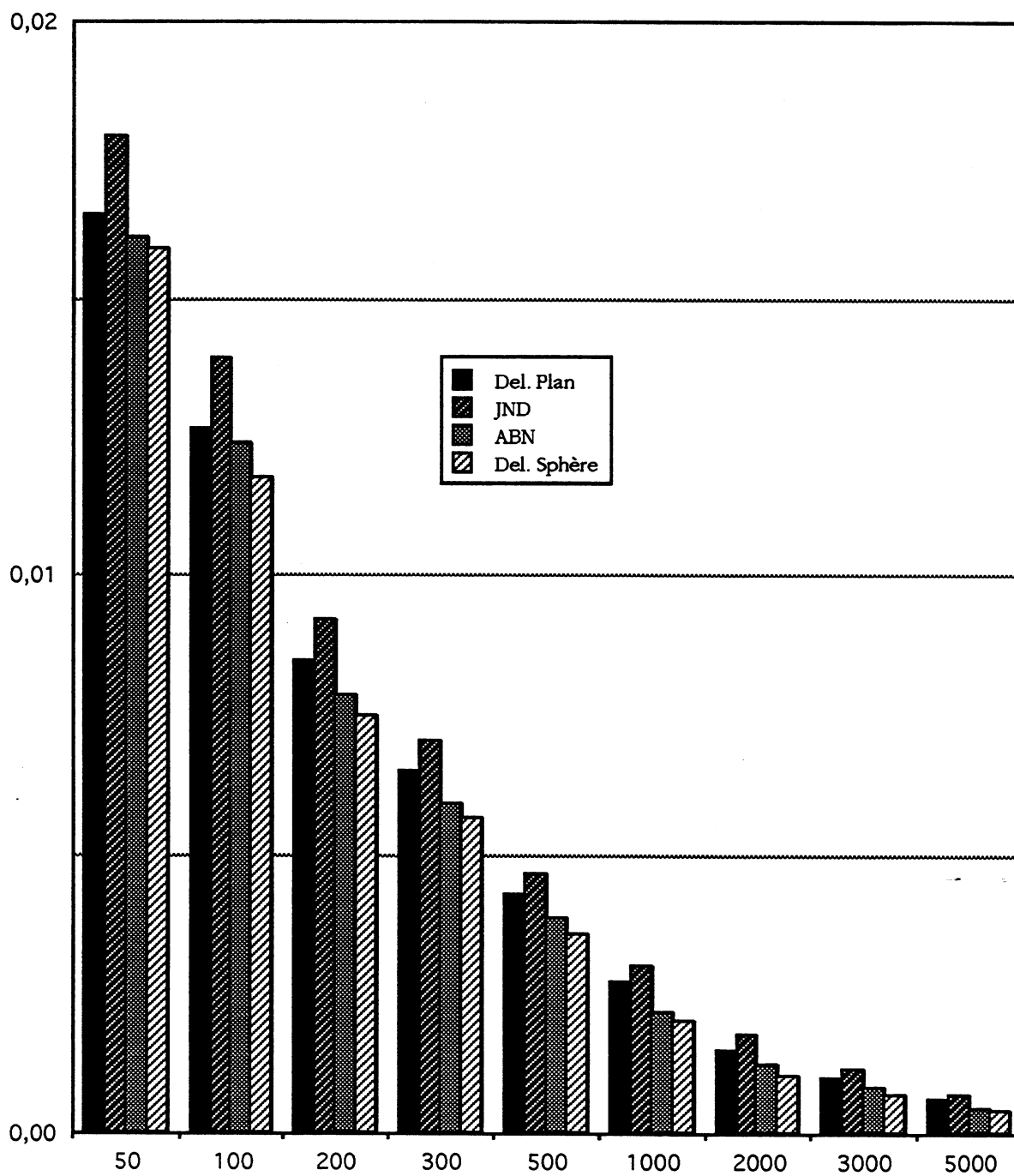


Figure 3.2 Erreur ε_z pour une demi-sphère de centre $(1/2, 1/2, 0)$ et rayon $1/2$

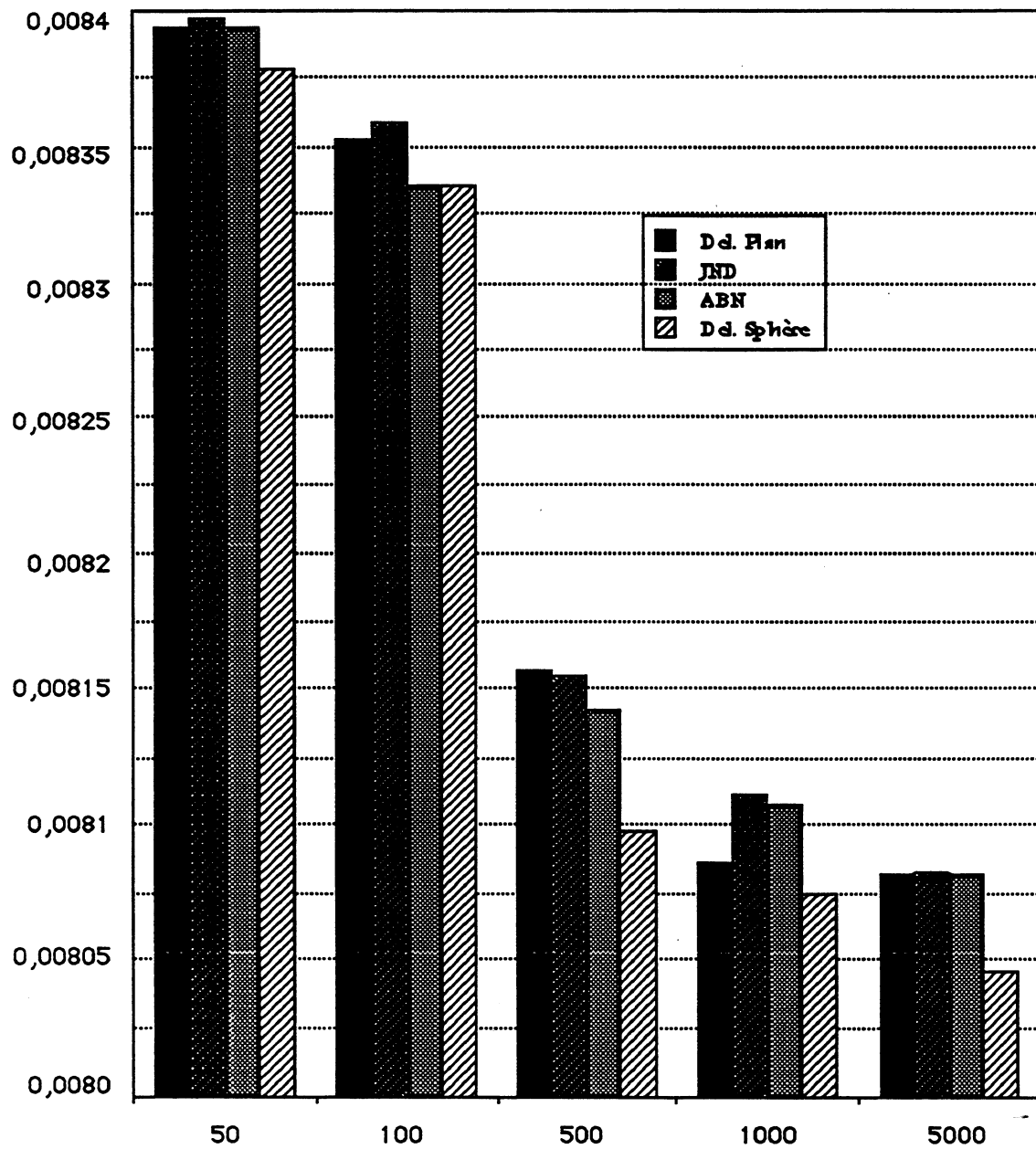


Figure 3.3 Erreur radiale pour une demi-sphère de centre $(1/2, 1/2, 0)$ et rayon $1/2$

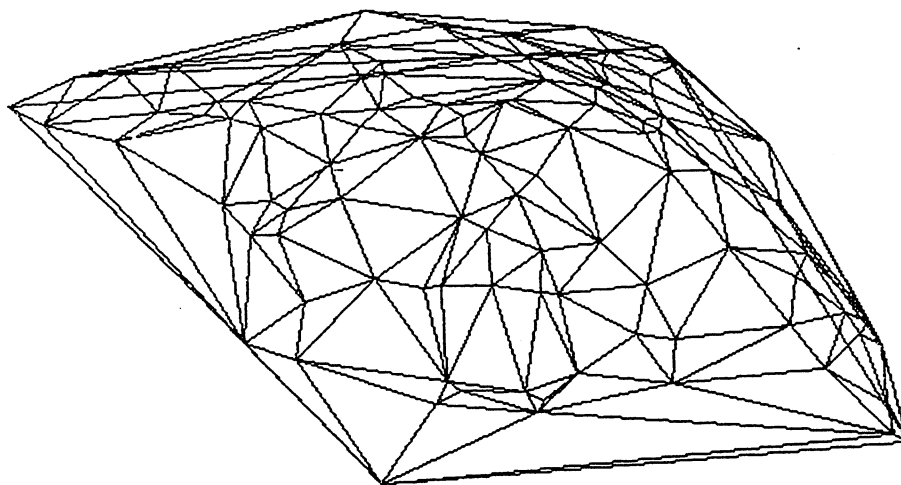


Figure 3.4 Delaunay sphérique , 100 points

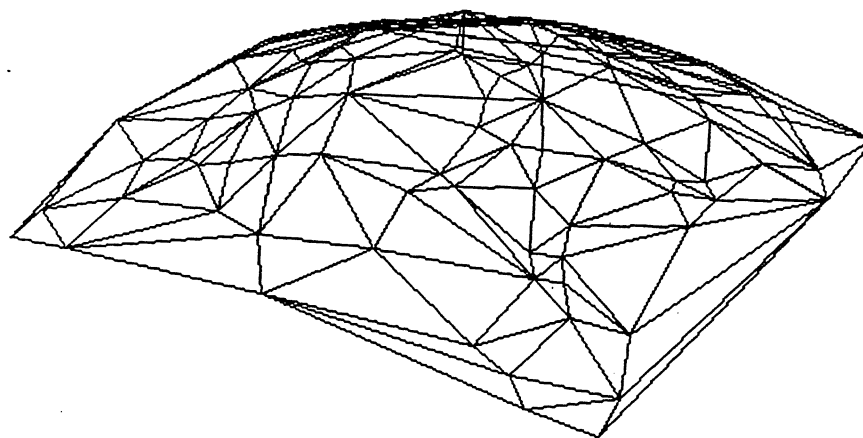


Figure 3.5 Triangulation selon le critère ABN, 100 points

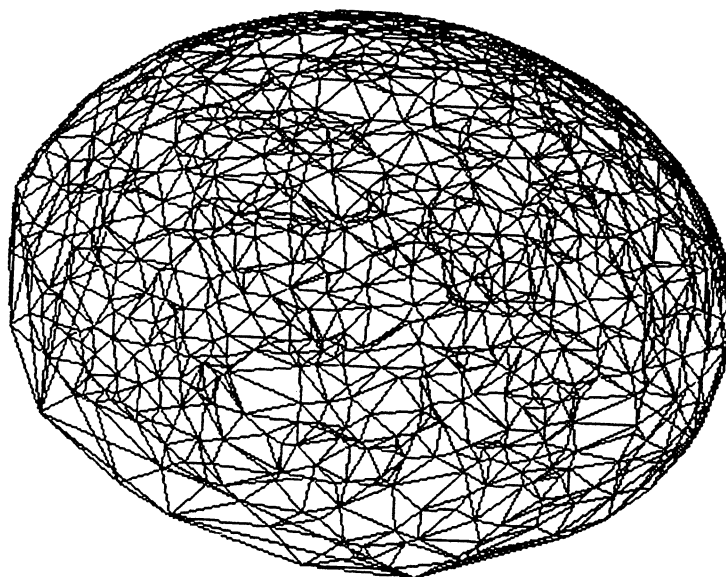


Figure 3.6 Delaunay sphérique, 1000 points

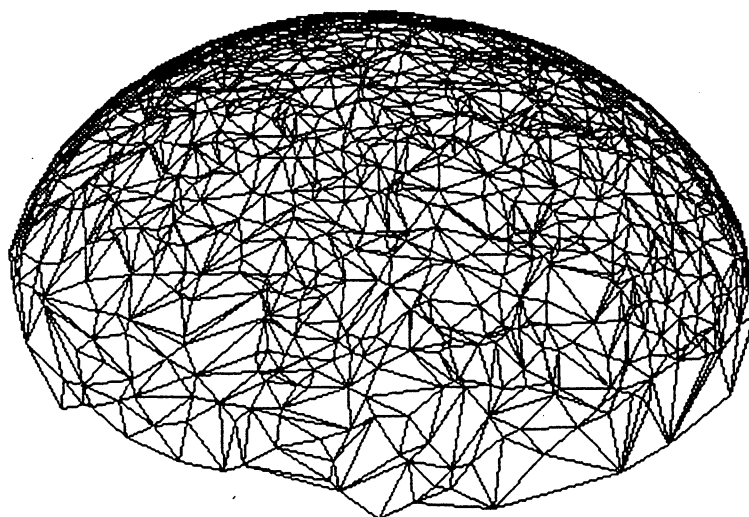


Figure 3.7 Triangulation selon le critère ABN, 1000 points

Exemple de la demi-sphère bruitée

Nous avons étudié ici le comportement des méthodes lorsque l'on introduit du bruit sur les données. Nous avons donc considéré la demi-sphère précédente en bruyant radialement les points échantillonnés selon une loi gaussienne, la valeur du bruit étant au plus égale à un certain pourcentage du rayon.

Nous avons ainsi représenté sur la figure 3.9 les erreurs commises par notre méthode et la triangulation selon le critère ABN, qui s'est révélée être la meilleure approximation parmi les trois critères ABN, JND et Delaunay dans le plan.

Si les erreurs radiales moyennes sont équivalentes pour les deux méthodes (ce qui n'est pas surprenant puisque les points sont bruités radialement selon une loi Gaussienne), l'erreur ε_z s'avère tout de même inférieure pour la triangulation à triangles plans, et ceci de plus en plus nettement avec l'accroissement de la valeur du bruit, ce qui dénote d'une bonne résistance au bruit de la méthode que nous avons présenté.

Les mêmes conclusions que dans le cas non-bruité s'imposent pour la longueur moyenne des arêtes lorsque les points sont perturbés, à savoir que les triangulations dépendantes des données souffrent plus de la présence du bruit en ce qui concerne leur "régularité".

Les figure 3.10 et figure 3.11 représentent les surfaces obtenues par la triangulation à triangles plans (à gauche sur les dessins) et la triangulation selon le critère ABN.

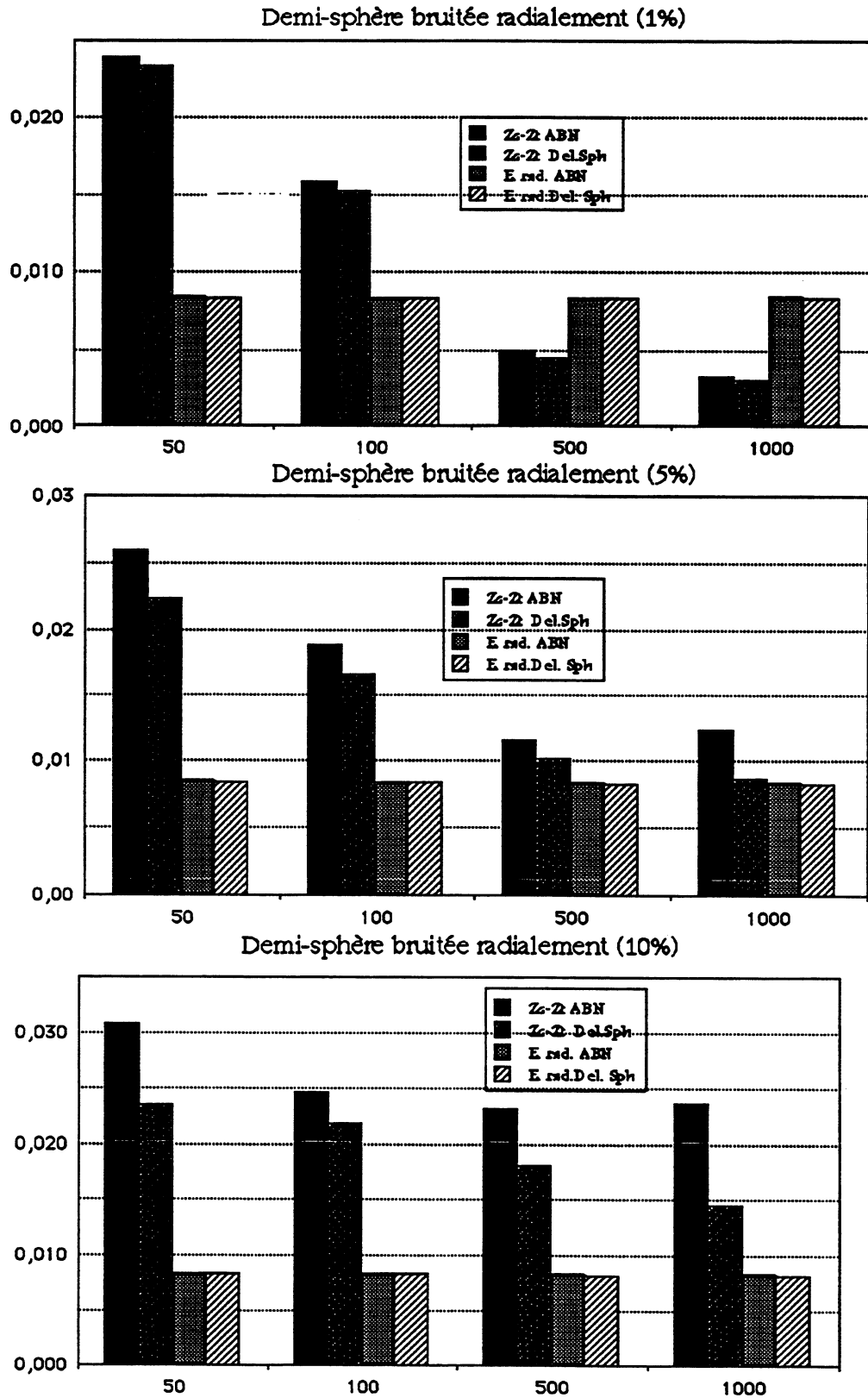


Figure 3.8 Erreurs "en z" et radiale pour une demi-sphère bruitée

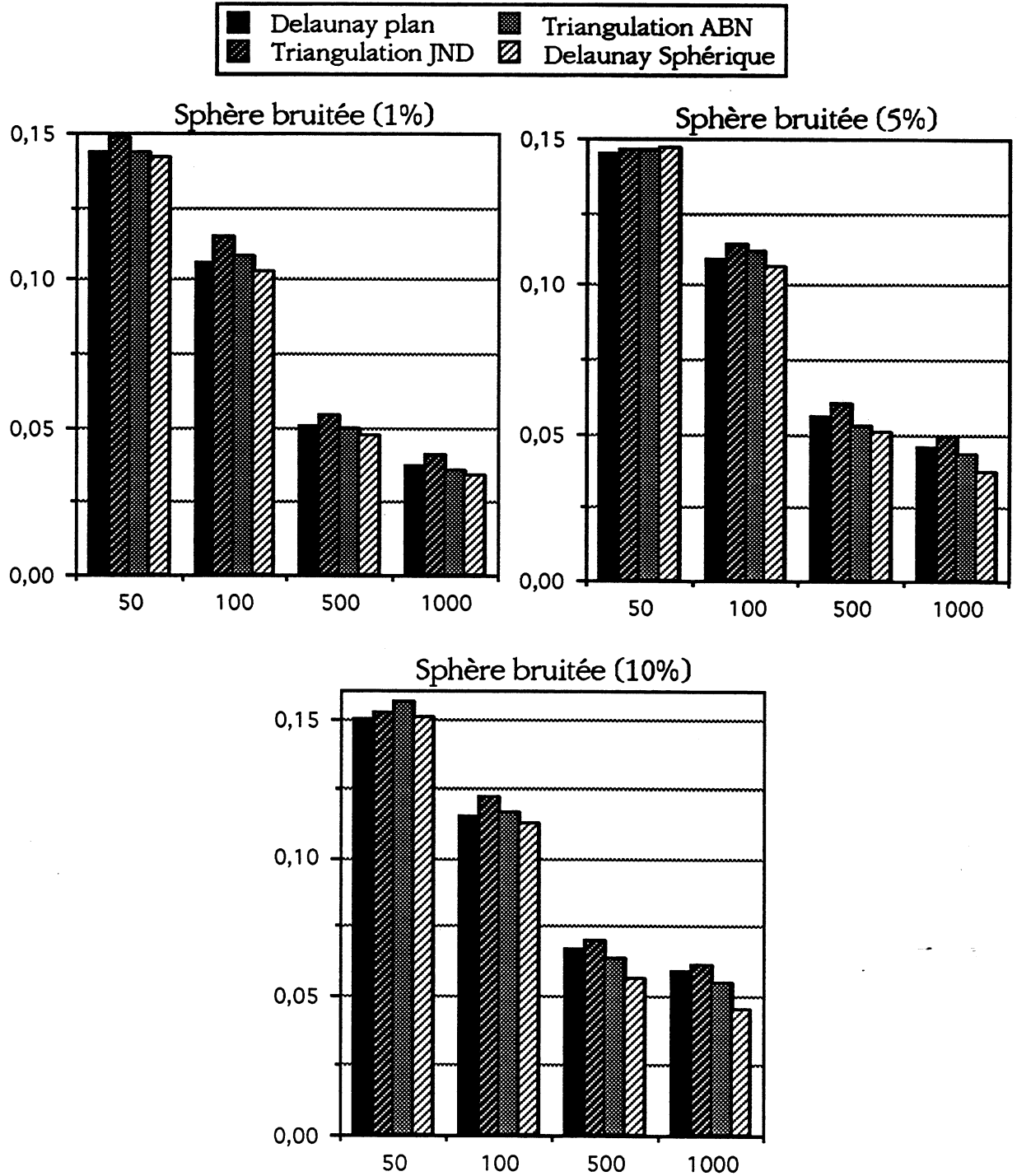


Figure 3.9 Longueur moyenne des arêtes pour une demi-sphère bruitée

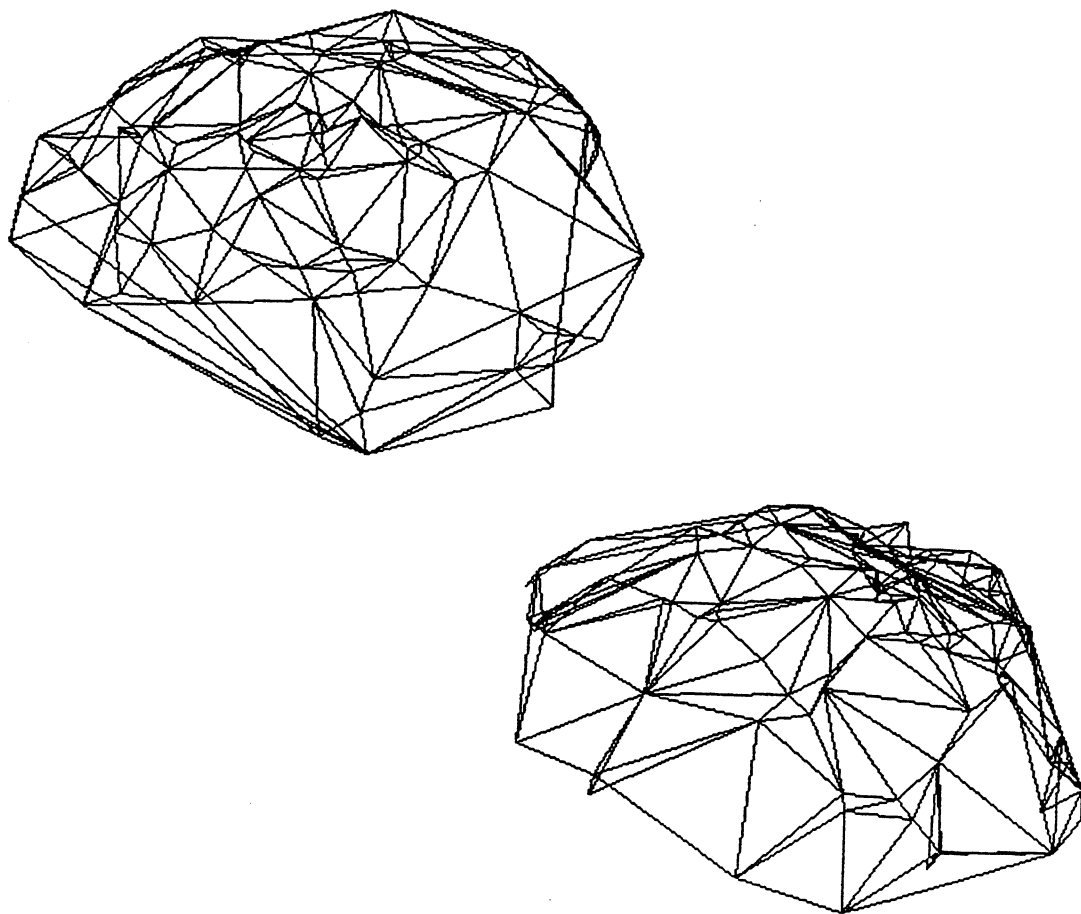


Figure 3.10 Delaunay sphérique et "ABN", 100 points

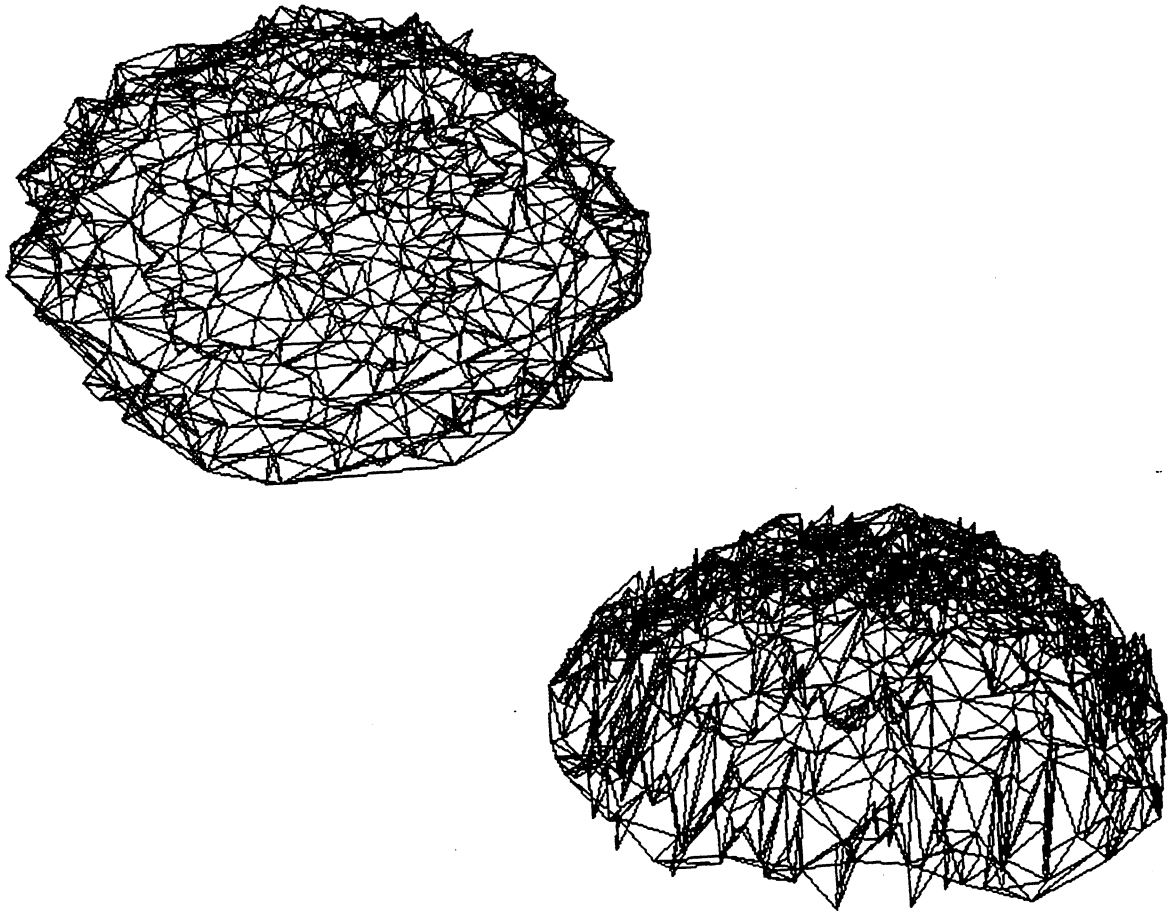


Figure 3.11 Delaunay sphérique et "ABN", 1000 points

Exemple de surface radiale

Nous avons considéré pour ce test la surface définie en coordonnées sphériques par :

$$r = 1/2 + 0,03 \cos[5(\theta + \varphi)], \text{ avec } 0 \leq \theta \leq 2\pi \text{ et } 0 \leq \varphi \leq 2\pi,$$

et représentée ci-dessous :

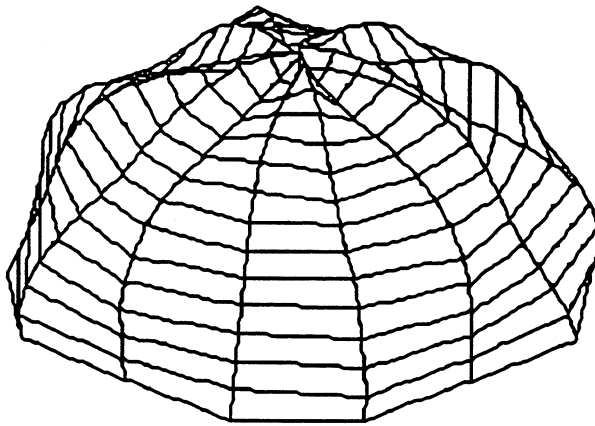


Figure 3.12

Le graphique de la figure 3.13 représente alors les erreurs radiales relatives aux surfaces approchantes générées par les quatre méthodes testées.

Il apparaît nettement que la triangulation à triangles plans fournit une meilleure approximation du point de vue radial, et qui va s'améliorant sensiblement lorsque le nombre de points s'accroît (ce à quoi on pouvait naturellement s'attendre), alors que ce n'est pas le cas des surfaces associées aux triangulations dépendantes de données de types ABN et JND.

D'autre part, la triangulation à triangles plans est synonyme une fois de plus d'une longueur moyenne des arêtes inférieure à celle des autres méthodes (cf. figure 3.14).

Les surfaces calculées par le triangulation à triangles plans et la triangulation selon le critère ABN sont ensuite représentées.

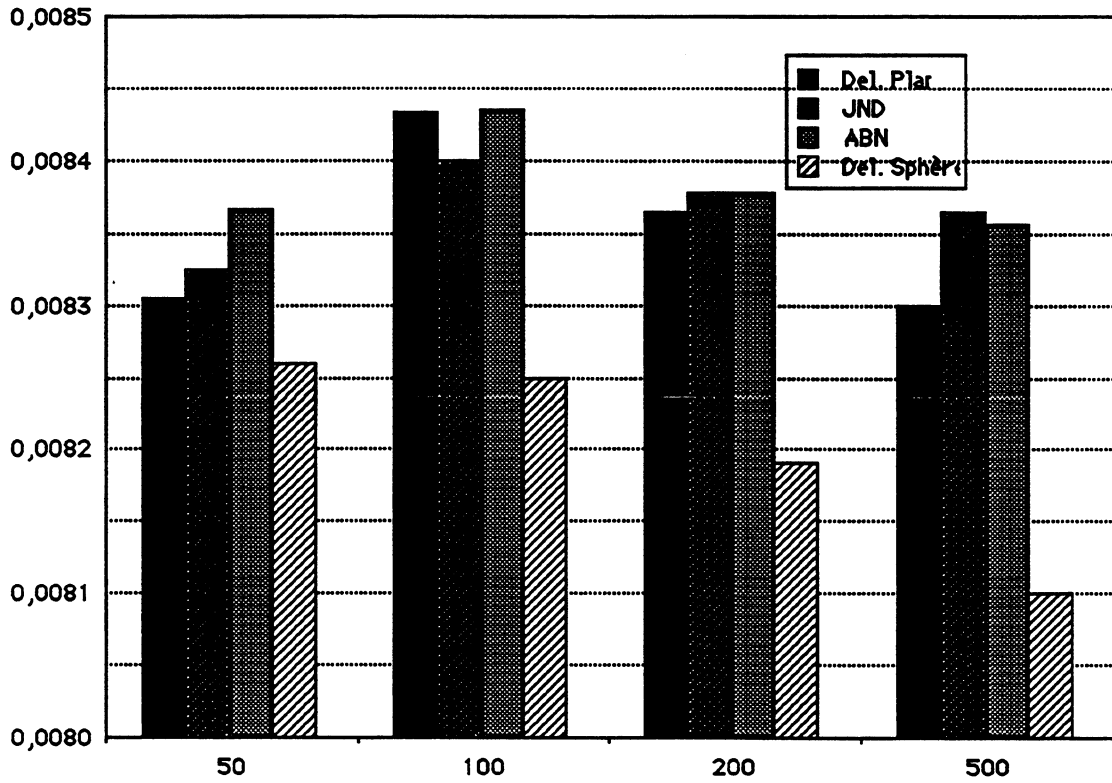


Figure 3.13 Erreurs radiales

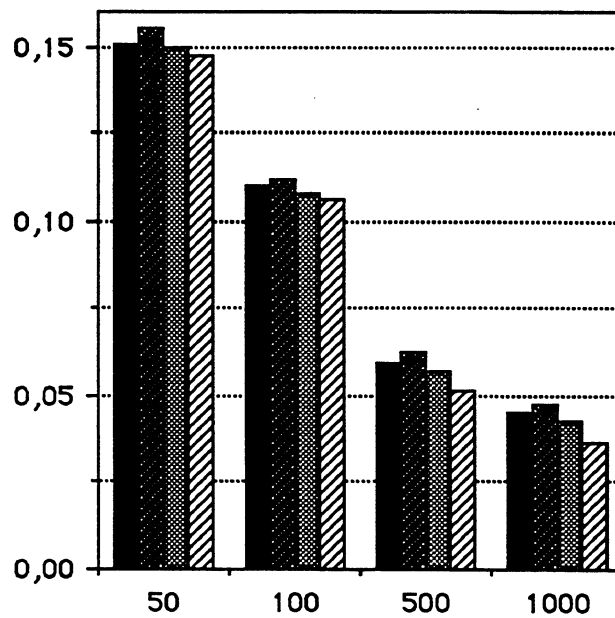


Figure 3.14 Longueur moyenne des arêtes

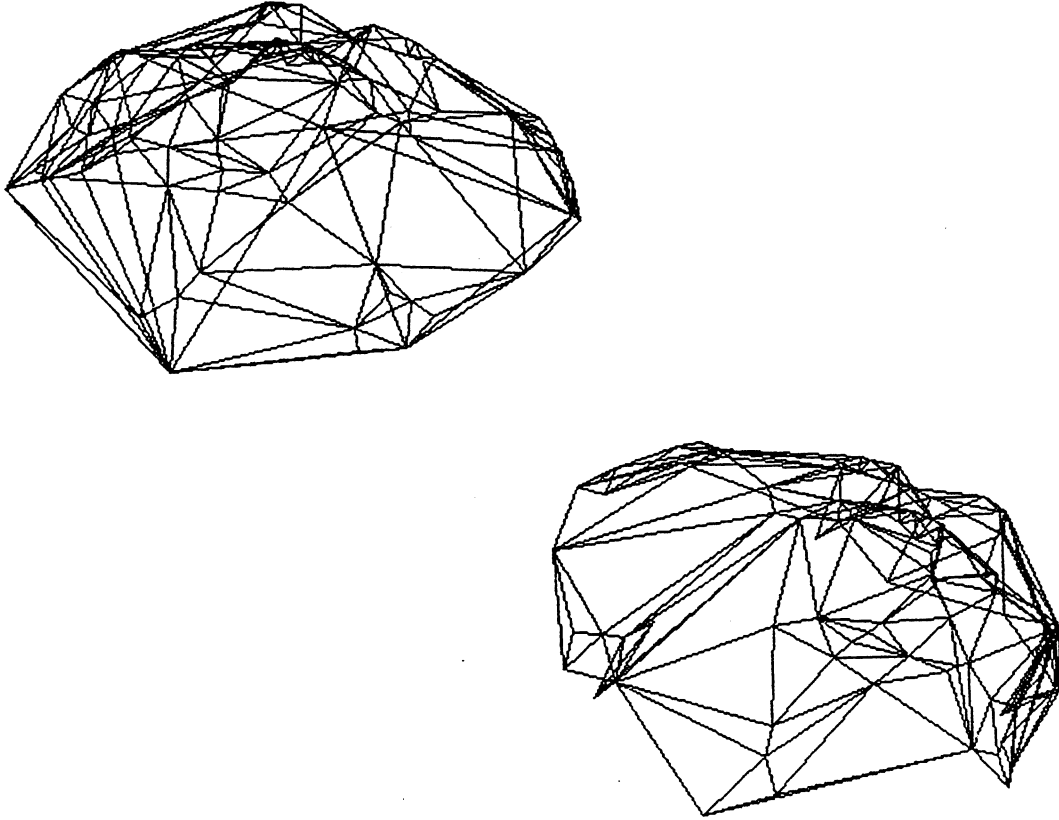


Figure 3.15 Delaunay sphérique et "ABN", 100 points

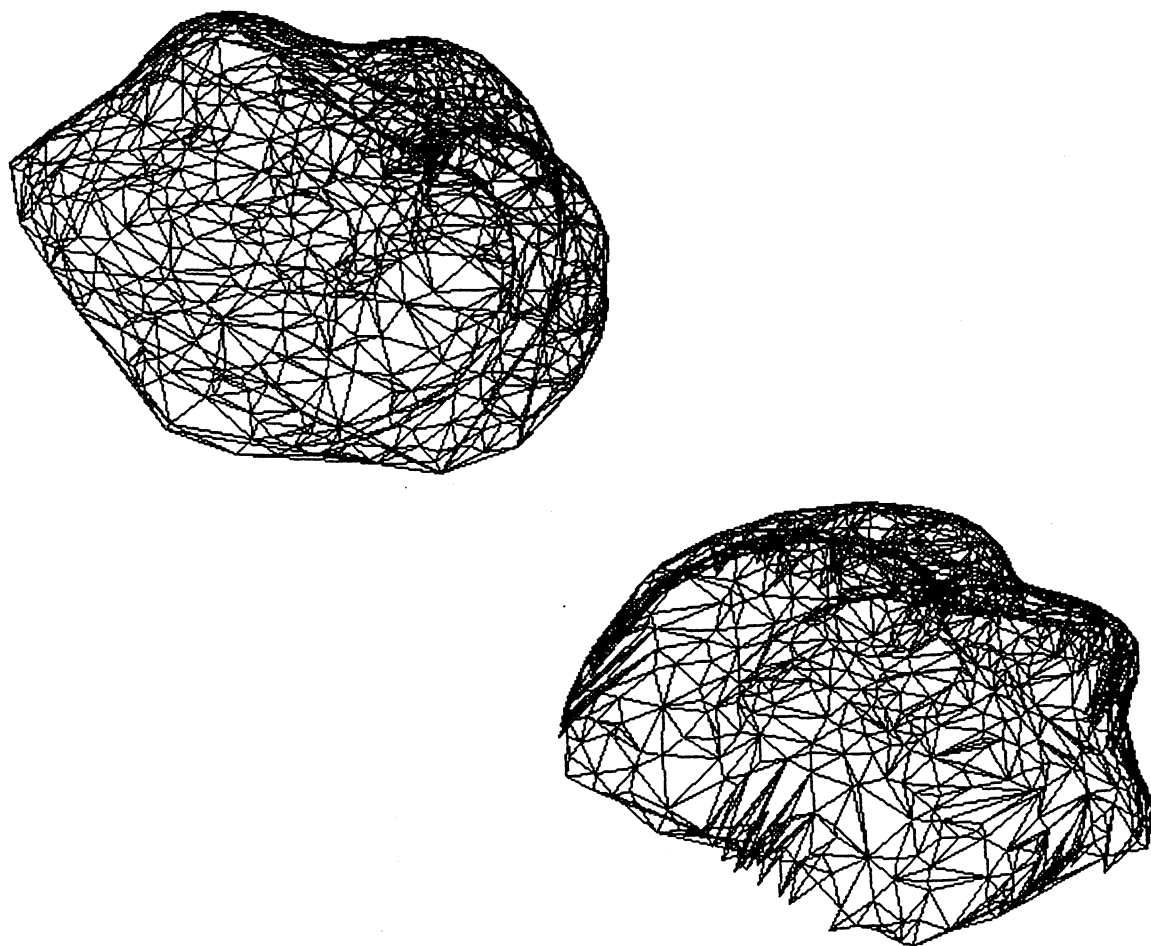


Figure 3.16 Delaunay sphérique et "ABN", 1000 points

Exemple de surface quelconque

Enfin, pour terminer cette série d'exemples, nous présentons un test de reconstruction basé sur une surface extraite des jeux d'essais réalisés par Dyn et *al.* [DLR 90,1].

Cette surface est définie, sur le carré unité, par l'équation :

$$z = 1/3 \exp \left[-81/16 \left((x - 1/2)^2 + (y - 1/2)^2 \right) \right].$$

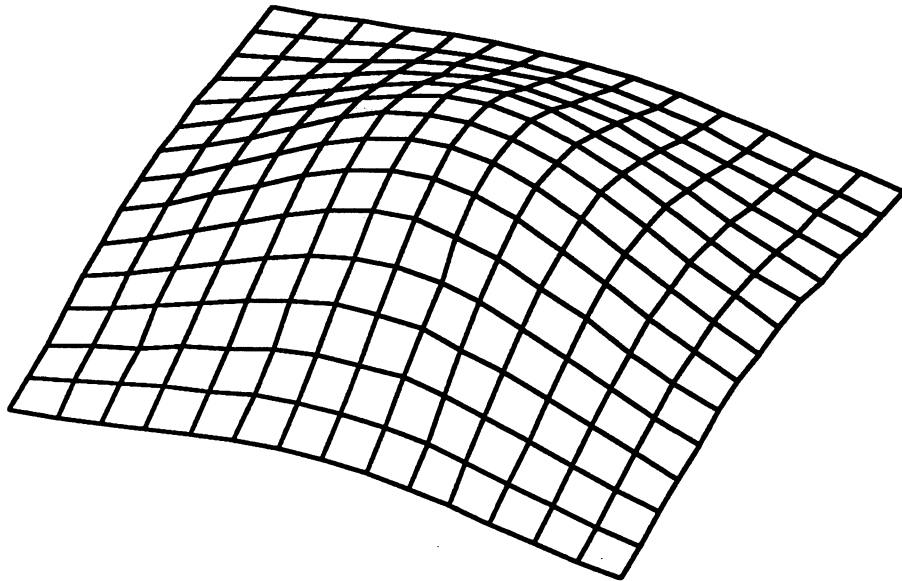


Figure 3.17

Nous présentons, comme pour les autres exemples, deux graphiques relatifs aux critères d'erreur et de longueur moyenne des arêtes.

Les résultats prouvent là encore que la triangulation à triangles plans donne des résultats satisfaisants sur des surfaces quelconques et relativement régulières.

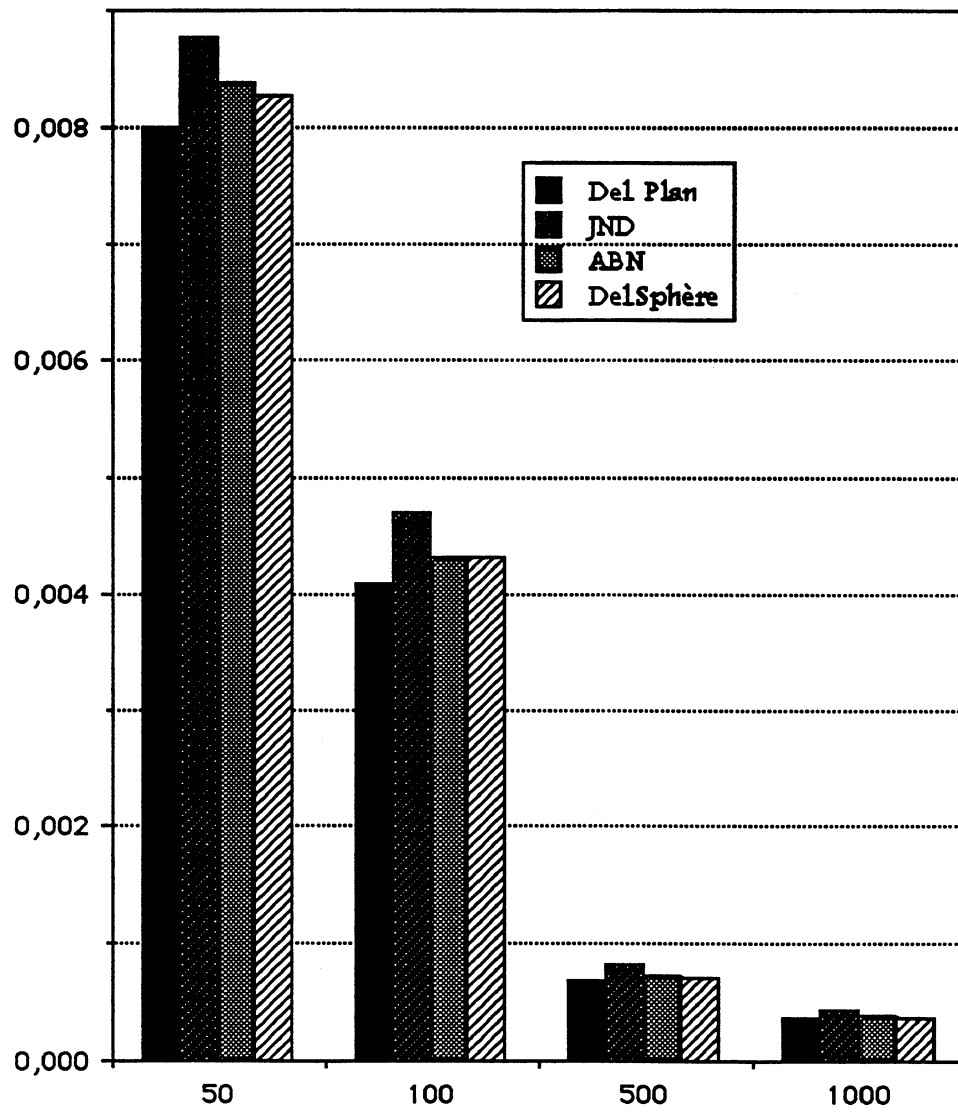


Figure 3.18 Erreur ε_z

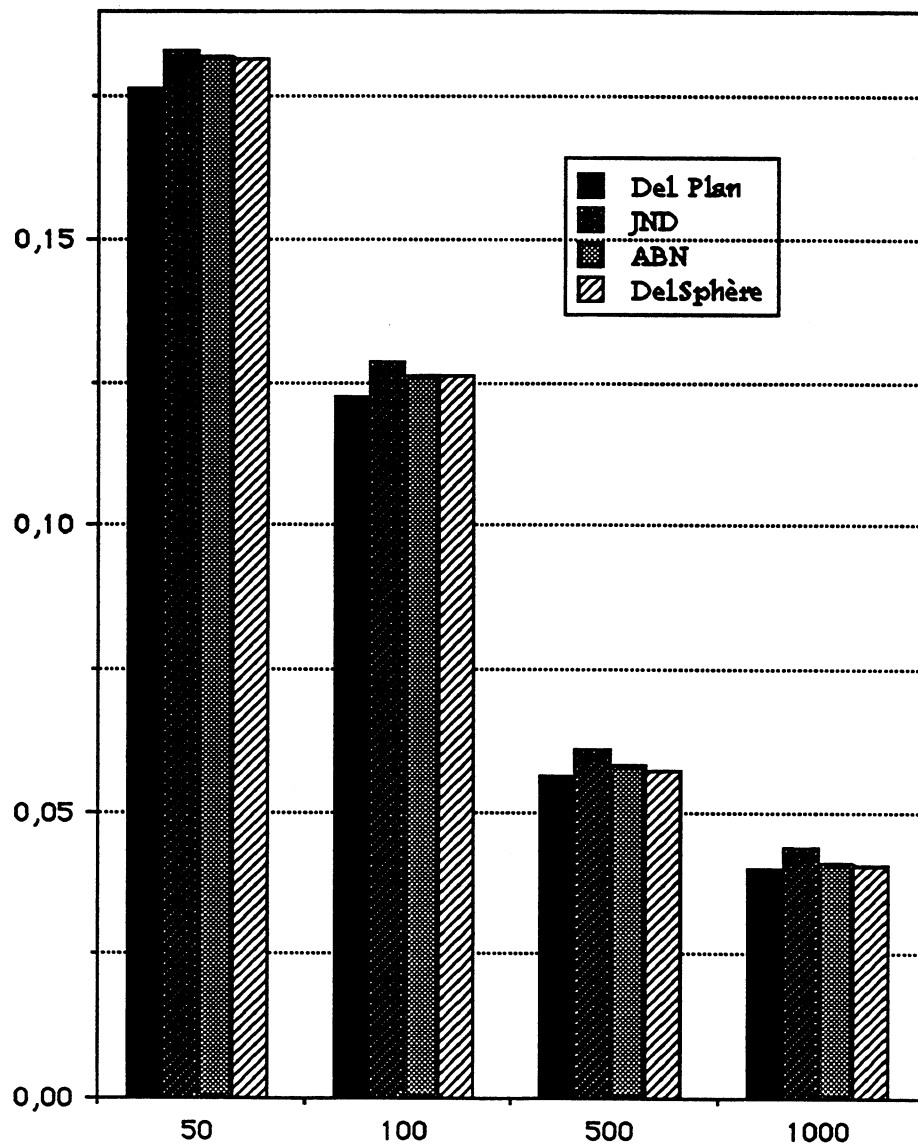


Figure 3.19 Longueur moyenne des arêtes

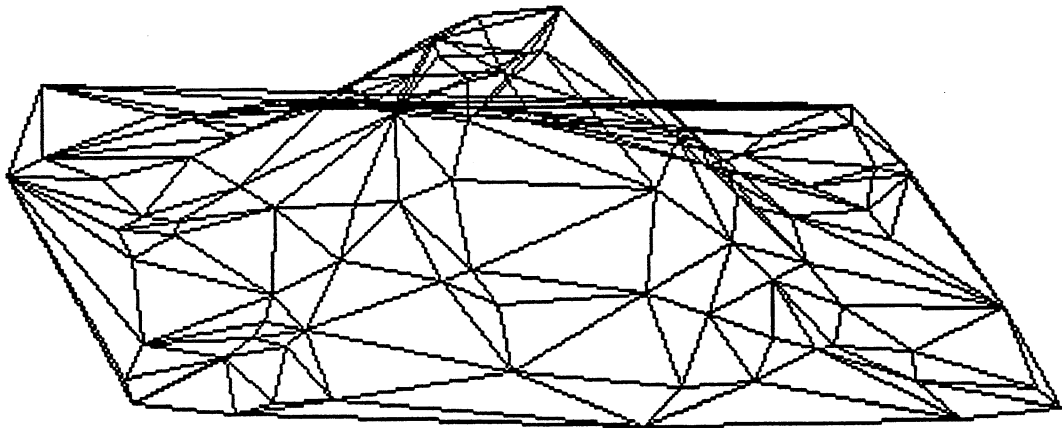


Figure 3.20 Delaunay sphérique , 100 points

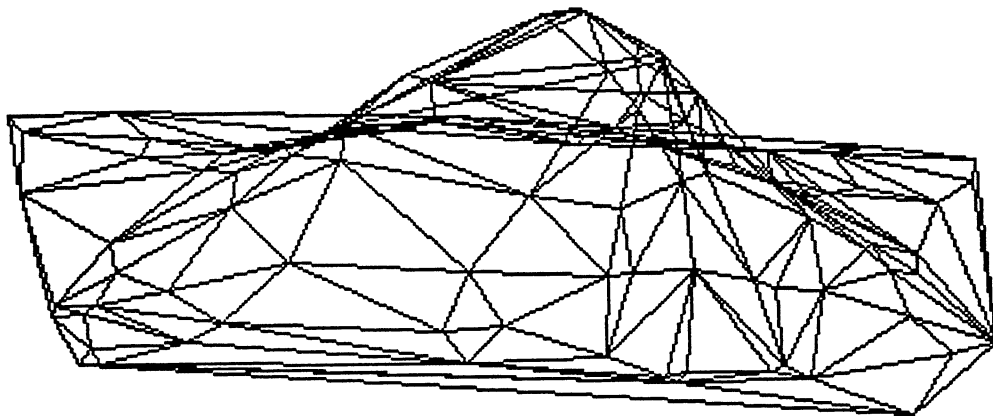


Figure 3.21 Triangulation ABN, 100 points

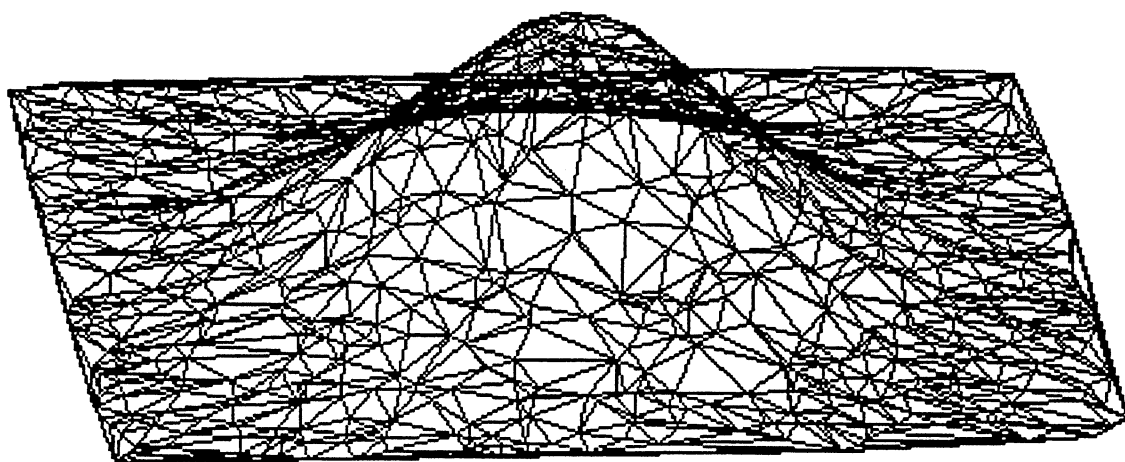


Figure 3.22 Delaunay sphérique, 1000 points

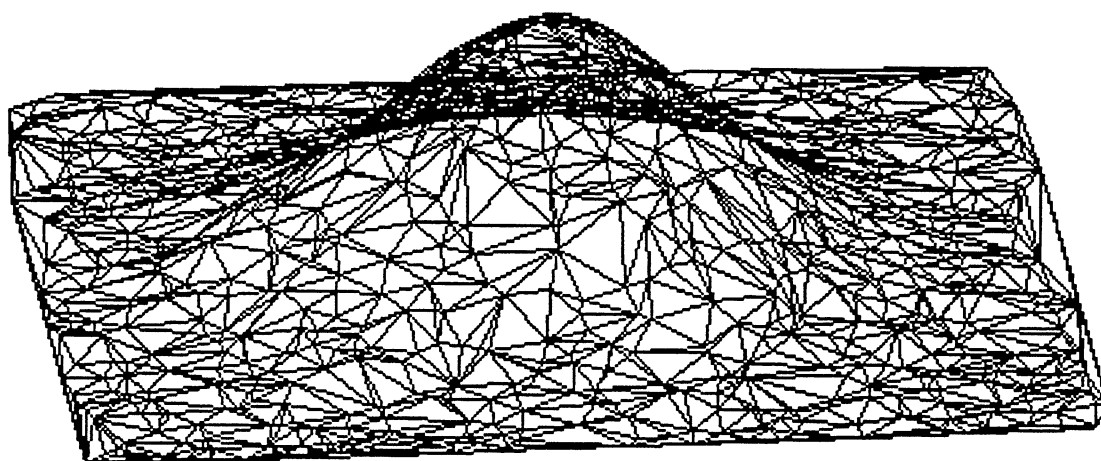


Figure 3.23 Triangulation ABN, 1000 points

Temps d'exécutions

Nous terminons ce paragraphe en représentant les temps d'exécutions en secondes des quatre méthodes en fonction du nombre de points du nuage échantillonné sur la surface.

Si la triangulation de Delaunay plane est évidemment la plus rapide (elle ne comporte en effet ni phase de post-optimisation par rapport aux triangulations dépendantes des données, ni calcul d'une sphère approchante et opérations de projection par rapport à notre méthode), la triangulation à triangles plans s'avère tout de même nettement plus performante que les triangulations selon les critères ABN et JND.

Ceci s'explique aisément par le fait que l'approximation par une sphère et la projection du nuage de points sur celle-ci sont des opérations linéaires en fonction du nombre de points, alors que les triangulations dépendantes des données impliquent le test d'un grand nombre d'arêtes pour des échanges éventuels lors de la phase de post-optimisation (n^2 dans le pire des cas, [Sch 87]).

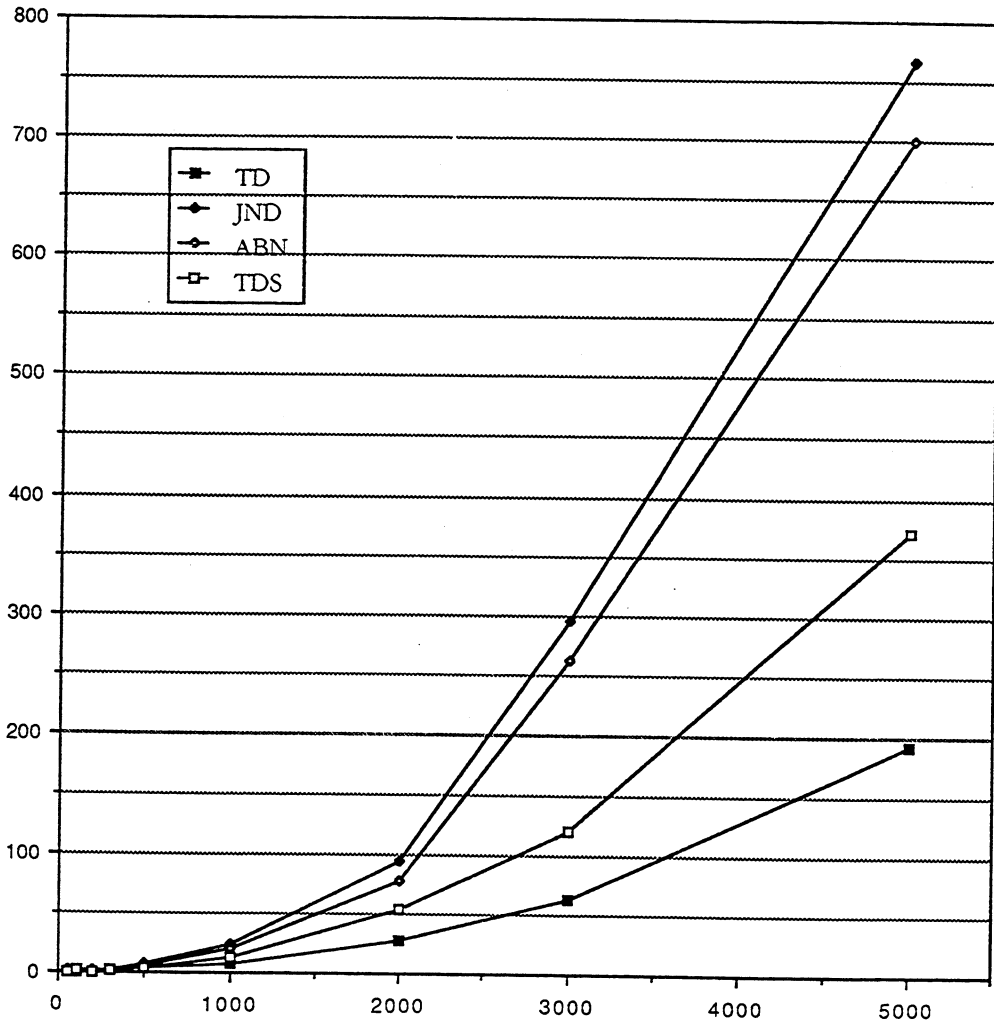


Figure 3.22 Temps d'exécutions en secondes en fonction du nombre de points

Conclusion

Nous avons traité dans cette étude des problèmes d'ordonnement d'un ensemble de points dans le plan et de reconstruction de surfaces, sous l'hypothèse que la seule connaissance que nous avons sur ces points est celle de leurs coordonnées.

Nous avons développé et présenté, dans la première partie, une méthode originale pour le problème de la reconstruction de courbe.

Cette approche, basée sur l'approximation locale de la courbe par des cercles, peut être qualifiée de composite puisqu'elle traite "globalement" des sous-ensembles de points, mais fait aussi intervenir des notions de proximité "locale" lorsqu'il s'agit de calculer l'union de ces sous-ensembles pour l'obtention du résultat final.

L'idée de traiter des sous-ensembles de points, et donc de découper l'ensemble initial donné, s'est justifiée par le fait qu'il n'est souvent pas possible de traiter globalement tous les points, mais aussi par l'objectif — atteint — d'obtenir une méthode performante en temps de calcul, et qui nous permette de traiter un nombre important de points. La volonté de traiter des jeux de données perturbées nous a conforté de même dans cette approche.

Nous avons donc développé une méthode fiable et robuste d'approximation d'un nuage de points par un cercle, au moyen d'un problème contraint de minimisation d'une fonctionnelle quadratique.

Cette méthode se généralise aisément à la sphère de \mathbb{R}^3 ou à une dimension quelconque. Si nous n'avons pu démontrer mathématiquement la convergence d'un tel cercle approximant vers le cercle osculateur en un point (dans le but d'évaluer ce dernier), l'étude expérimentale de cette convergence nous a donné des résultats allant en ce sens.

Des algorithmes et une structure de données, de complexité optimale en temps et en espace mémoire, nous ont ainsi permis de calculer en des temps très

raisonnables des ensembles contenant un grand nombre de points (de l'ordre de un million).

Nous avons pu constater d'autre part une bonne résistance de la méthode par rapport à des jeux de données perturbées.

Enfin, une stratégie originale d'énumération des différentes morphologies locales possibles d'une courbe nous a permis de venir à bout de certains cas d'auto-intersections, ou de cas de points de rebroussement.

Nous avons pu, pour terminer, appliquer l'ordre sur les points que nous avons calculé au problème de la paramétrisation de ces points. L'emploi des cercles approximatifs pour le calcul de ces paramètres dans le cas de points perturbés a donné des résultats probants sur des exemples d'approximation par des B-splines régularisées.

Nous avons ensuite, dans la deuxième partie, appliqué l'idée principale de la première partie — l'approximation d'arcs de courbe par des cercles — au problème de la reconstruction de surfaces.

Nous avons ainsi mis au point une méthode basée sur l'approximation par une sphère d'un nuage de points échantillonnés sur une surface.

Une fois les points projetés sur la sphère approchante, il nous est alors possible de calculer une triangulation de Delaunay de cet ensemble sur la sphère.

Nous obtenons ainsi une structure topologique cohérente liant ces points, qui nous fournit donc une surface polyédrique approchant la surface initiale.

Si la méthode est conçue pour fonctionner avec des surfaces plus générales (notamment des surfaces fermées radiales de \mathbb{R}^3), nous l'avons testée, dans un but de comparaison, avec les triangulations dépendantes des données, sur des surfaces du type $z = f(x, y)$. Dans ce genre d'application, où compte plus la "qualité" de l'approximation, nous avons obtenu des résultats probants, notamment avec des surfaces présentant une certaine régularité et avec des surfaces radiales.

Par rapport à ce travail, demeurent, en ce qui nous concerne, plusieurs questions relatives au problème de la reconstruction de surfaces.

Si la méthode que nous présentons ici ne fonctionne pas uniquement sur des surfaces du type $z = f(x, y)$, nous avons effectivement comparé les résultats obtenus par notre méthode à ceux issus des triangulations dépendantes des données.

Mais ne confond-t'on pas les problèmes qui sont "purement" de reconstruction, au sens où le but final est l'obtention d'un solide de \mathbb{R}^3 , et ceux d'approximation de surfaces, comme le sont plus les objectifs des triangulations dépendantes des données ?

Peu de réponses sont données à ce sujet.

On peut remarquer d'autre part que, pour les triangulations dépendantes des données par exemple, les méthodes usitées employent toujours un critère basé sur une idée intuitive, et sont validées sur des expérimentations, et les choses ont peu avancé sur ce problème, en ce qui concerne la méthodologie et au sens où peu de résultats mathématiques ont été obtenus.

On peut finalement se demander quels sont les "bons" critères par rapport à ce problème de reconstruction.

Faut-il des triangles "allongés" ou non ?

La triangulation obtenue doit-elle respecter la convexité (ce qui n'est pas le cas de toutes les méthodes développées) ?

La triangulation doit-elle être géodésique, au sens où elle relie le plus souvent les points par rapport à leurs plus proches voisins, ou en relation avec les plus courts chemins correspondants généralement correspondants sur la surfaces ?

L'approximation par une sphère de la surface entraîne une idée intuitive du calcul d'une triangulation géodésique (ou au moins approchante) des points situés sur la surface. Cet aspect est peut-être d'ailleurs une interprétation géométrique intéressante du problème de l'approximation et de la reconstruction d'une surface.

Bibliographie

- [ABY 90] P. Alevizos, J.D. Boissonnat & M. Yvinec
Non-convex contour reconstruction.
J. Symbolic Computation, Vol.10, pp. 225-252, 1990.
- [AB 85] P.F. Ash & E.D. Bolker
Recognizing Dirichlet tessellations .
Geometriae Dedicata, Vol.19, pg.175-206, 1985.
- [Bar 85] R. E. Barnhill
Surfaces in CAGD : a survey with new results.
Computer-Aided Geometric Design, Vol. 2, pg. 1-17, 1985.
- [Ber 77,1] M. Berger
Géométrie - volume 2 .
Cedic/F.Nathan, 1977.
- [Ber 77,2] M. Berger
Géométrie - volume 5 .
Cedic/F.Nathan, 1977 .
- [BFK 84] W. Böhm, G. Farin & J. Kahman
A survey of curve and surface methods inCAGD.
Computer-Aided Geometric Design, Vol. 1, pg. 1-60, 1984.
- [Boi 82] J.-D. Boissonnat
Representation of objects by triangulating points in 3-D space.
IEEE, pp.830-832, 1982.
- [Boi 84] J.-D. Boissonnat
Geometric Structures for Tree-Dimensional Shape Representation .
ACM Transactions on Graphics, Vol.3, N°4, pg.266-286, 1984 .
- [Boi 86] J.-D. Boissonnat
"Shape Reconstruction from Planar Cross-Section".
Rapport de recherche INRIA N° 546 , 1986.
- [Bro 91] J.L. Brown
Vertex based data dependent triangulations .
Computer Aided Geometric Design, Vol.8, pg.239-251, 1991 .
- [BZ 70] J. Bramble & M. Zlamal
Triangular elements in the finite element method.
Math. Comp., Vol.24, pg.809-820, 1970 .

- [Cia 82] P. G. Ciarlet
Introduction à l'analyse numérique matricielle et à l'optimisation.
Masson, 1982.
- [CH 90] C. Cherfilis & F. Hermeline
Diagonal swap procedures and characterizations of 2D-Delaunay Triangulations.
Modélisation mathématique et analyse numérique, Vol.24, N°5, pg.613-623, 1990.
- [Cla 87] K.L. Clarkson
New applications of random sampling in computational geometry.
Discrete and Computational Geometry, Vol.2, pg.195-222, 1987 .
- [CR 84] A.K. Cline & R.L. Renka
A storage-efficient method for construction of a thiesen triangulation .
Rocky Mountain Journal of Mathematics, Vol.14, N°1, pg.119-139, 1984 .
- [CS 89] K.L. Clarkson & P. W. Shor
New applications of random sampling in computational geometry, II.
Discrete and Computational Geometry, Vol.4, pg 387-421, 1989 .
- [DF 94] J.-P. Dedieu & C. Favardin
Algoritnms for ordering unorganized points along parametric curves.
To appear in : Numerical Algorithms.
- [DF 91] J.-P. Dedieu & C. Favardin
How to draw a curve using geometrical data.
Curves & Surfaces, P.J. Laurent, A. Le Méhauté and L.L. Scumaker (eds.), pp.
135-138, Springer Verlag, 1991.
- [DLR 90,1] N. Dyn, D. Levin & S. Rippa
Data dependent triangulations for piecewise linear interpolation
IMA J. of Num. Analysis, Vol.10, pg.137-154, 1990.
- [DLR 90,2] N. Dyn, D. Levin & S. Rippa
Algorithms for the construction of data dependent triangulations.
in Algorithms for approximation II, J. C. Mason & M. G. Cox eds., Chapman
and Hall, 1990.
- [Ede 87] Herbert Edelsbrunner
Algorithms in Combinatorial Geometry .
EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1987 .
- [Eps 76] M. P. Epstein
On the influence of parametrization in parametric interpolation.
SIAM J. Numer. Analysis, Vol. 13, N°2, pg. 261-268, 1976.
- [Far 88] G. Farin
Curves and surfaces for CAGD. A practical guide.
Academic Press, 1988.
- [Fav 93] C. Favardin
Détermination automatique de structures géométriques destinées à la reconstruction de courbes et surfaces à partir de données ponctuelles
Thèse de l'université Paul Sabatier de Toulouse, 1993

- [FB 74] R. A. Finkel & J. L. Bentley
Quadrees. A data structure for retrieval on composite keys.
Acta Informatica, Vol.4, pg. 1-9, 1982.
- [FN 82] R. Franke
Scattered data interpolation : tests of some methods.
Math. Comp., Vol.38, pg.181-200, 1989.
- [FN 89] R. Franke & G. M. Nielson
Scattered data interpolation and applications : a tutorial and survey.
in *Mathematical methods in CAGD*, T. Lyche & L. L. Schumaker (eds.), pg.131-160, Academic Press, 1989.
- [FN 89] T. A. Foley & G. M. Nielson
Knot selection for parametric spline interpolation.
Mathematical methods in CAGD, T. Lyche & L. L. Schumaker (eds.), pg.261-271, Academic Press, 1989.
- [Gre 75] J. A. Gregory
Error bounds for linear interpolation in triangles.
in : *The mathematics of finite elements and application II*, J. R. Whiteman (ed.), Academic Press, 1975.
- [Grü 67] B. Grünbaum
Convex polytopes.
Pure and applied mathematics, Vol. XVI, Interscience Publishers, 1967.
- [GKS 90] L. J. Guibas, D. E. Knuth & M. Sharir
Randomized incremental construction of Delaunay and voronoï diagrams.
Dept. of Computer Sci., Stanford University, report STAN-CS-90-1300, 1990.
- [GS 85] L.J. Guibas & J. Stolfi
Primitives for the manipulation of general subdivisions and the computation of Voronoï diagrams.
ACM Transaction on Graphics, Vol.10, N°1, pg.71-91, 1985 .
- [Her 82] F. Hermeline
"Triangulation automatique d'un polyèdre en dimension n".
R.A.I.R.O. Analyse numérique, Vol.16, N°3, pp.211-242, 1982.
- [HDD 92] H Hoppe, T. DeRose, T. Duchamp, J. McDonald & W. Stuetzle
Surface reconstruction from unorganized points.
Computer Graphics, Vol.26, N°2, pg.71-78, 1992.
- [JB 90] J. K. Johnstone & C. L. Bajaj
Sorting points along an algebraic curve.
SIAM J. Comput., Vol.19, N°5, pg.925-967, 1990.
- [Laf 84] Y. Lafranche
Application de l'interpolation polynomiale au dépouillement graphique de valeurs de \mathbb{R}^3 .
Thèse de l'université et de l'INSA de Rennes, 1984.

- [Lau 87] P. J. Laurent
Courbes ouvertes ou fermées par B-splines régularisées.
Rapport de recherche LMC, Université Joseph Fourier, RR 652-M-, mars 1987.
- [LT 86] P. Lascaux & R. Théodor
Analyse numérique matricielle appliquée à l'art de l'ingénieur, tome II.
Masson, 1986
- [Law 77] Charles L. Lawson
Software for C1 surface interpolation
Mathematical Software III, J.R. Rice Ed., pg.161-194, 1977 .
- [Law 84] C. L. Lawson
C1 surface interpolation for scattered data on a sphere .
Rocky Mountain Journal of Mathematics, Vol.14, N°1, pg.177- , 1984 .
- [Law 86] C. L. Lawson
Property of n dimensional triangulations.
1986 .
- [LFA 75] J. Lelong-Ferrand & J. M. Arnaudès
Cours de mathématiques, géométrie et cinématique.
Dunod, 1975.
- [LMS 91] P. J. Laurent, A. Le Méhauté & L. L. Schumaker (eds.)
Curves and surfaces.
Academic Press, 1991.
- [LS 80] D.T. Lee & B.J. Schachter
Two algorithms for constructing a Delaunay triangulation .
International Journal of Computer and Information Sciences, Vol.9, N°3, pg.219-242, 1980 .
- [Maz 91] M. L. Mazure
Geometric continuity and Frenet continuity of parametric curves.
Rapport de recherche LMC, Université Joseph Fourier, RR 846-M-, 1991
- [NR 87] G.M. Nielson & R. Ramaraj
Interpolation over a sphere based upon a minimum norm network .
Computer Aided Geometric Design, Vol.4 ,N°1, pg.41-57, 1987 .
- [Pra 87] V. Pratt
Direct least-square fitting of algebraic surfaces.
Computer Graphics, Vol.21, pg.145-152, 1987.
- [PH 77] F.P. Preparata & S. J. Hong
Convex hulls of finite sets of points in two and three dimensions.
Communications of the ACM, Vol.20,N°2, pg.87-93, 1977 .
- [PS 85] F.P. Preparata & M.I. Shamos
Computational Geometry - An Introduction .
Texts & Monographs in Computer Science, Springer Verlag, 1985 .

- [Ren 84] R.J. Renka
Interpolation of data on the surface of a sphere.
ACM Transactions on Mathematical Software, Vol.10, N°4, pg.417-436, 1984 .
- [Rip 90] S. Rippa
Minimal roughness property of the Delaunay triangulation.
Comput. Aided Geom. Design., Vol.7, pg. 489-497,1990.
- [Rip 93] S. Rippa
Scattered data interpolation using minimum enrgy Powell-Sabin elements and data dependant triangulations.
Numerical Algorithms, Vol.5, pg. 577-587,1993.
- [Sam 84] H. Samet
The Quadtree and Related Hierarchical Data Structures.
ACM Computing Surveys, Vol.16, N°2, pp.187-260, 1984.
- [Sch 87] L. L. Schumaker
Triangulation methods
in Topics in multivariate approximation, C. K. Shui, L. L. Schumaker & F. I. Utreras eds., Academic Press, 1987.
- [Sch 90] L.L. Schumaker
"Reconstructing 3D objects from cross-sections".
Computation of Curves and Surfaces, pp.275-309, W.Dahmen et al. (eds.), 1990.
- [Sch 93] L. L. Schumaker
Computing optimal triangulations using simulated annealing
Comput. Aided Geom. Design., Vol.10, pg. 329-345, 1993.
- [SL 89] Su & Liu
Computational geometry. Curve and surface modelling.
Academic Press, 1989.

RESUME

Nous abordons dans cette étude le problème de la reconstruction de courbes et de surfaces, à partir de points leur appartenant et sous l'hypothèse que la seule connaissance que nous avons sur ces points est celle de leurs coordonnées.

Dans le cas des courbes, nous proposons une méthode basée sur l'approximation locale de la courbe par des cercles et sur le traitement global de sous-ensembles de points.

Une méthode d'approximation robuste au moyen d'un problème de minimisation permet donc d'approcher localement la courbe par un cercle, et d'ordonner les sous-ensembles de points ainsi approchés.

Des méthodes algorithmiques de découpe et de raccord permettent alors de mener à bien la reconstruction d'une courbe.

L'existence de points multiples ou de points de rebroussement est prise en compte par une stratégie d'énumération des différentes morphologies locales de la courbe. La méthode s'avère aussi robuste lorsque les points initiaux sont perturbés.

Les complexités temporelle et en place mémoire optimales des algorithmes et de la structure de données, ainsi que l'ordonnement global permettent de traiter des ensembles initiaux comportant un grand nombre de points.

Des cas de surfaces radiales ou de surfaces correspondant au graphe d'une fonction ont été traités en approchant le nuage de points par une sphère. Les points sont projetés et triangulés selon la triangulation de Delaunay sur la sphère, et nous obtenons alors une surface polyédrique liant les points.

Des tests et des comparaisons avec des méthodes du type triangulations dépendantes des données sont établis sur ces catégories de surfaces.

Mots clefs : Courbes, Surfaces, Reconstruction, Approximation, Géométrie algorithmique, Cercle et sphère approchants, Triangulation de Delaunay sphérique, Paramétrisation.

ABSTRACT

We tackle the problem of curves and surfaces reconstruction, from points belonging to them and under the hypothesis such that the only knowledge we have is the coordinates of these points.

In the case of curves, we suggest a method based on local curve approximation with circles and on global computation of points subsets.

A robust approximation method by means of minimisation problem allows then to locally approximate the curve with circle, and to sort so approximated points subsets.

Computational methods for cutting up and link allow to complete curve reconstruction successfully.

Self-intersections and cusps are taken into account with an enumeration of different local curve morphologies. The method also proves to be robust, with noisy data.

Time and memory size optimal complexities of algorithms and data structure, as well as global sort of points, allows to compute initial sets with a great number of points.

Radial surfaces or surfaces resulting from functions cases have been computed by means of approximation of the set of points with a sphere. Points are projected and a spherical Delaunay triangulation is computed, to obtain a polyhedral surface linking that points.

Tests and comparisons are realized with data dependant triangulations on that kind of surfaces.

Keywords : Curves, Surfaces, Reconstruction, Approximation, Computational geometry, Approximating circles and sphere, Spherical Delaunay triangulation, Parametrization.