

Institut National de Génie Mécanique
(INGM) Boumerdes

THESE

PRESENTÉE POUR L'OBTENTION DU DIPLOME DE

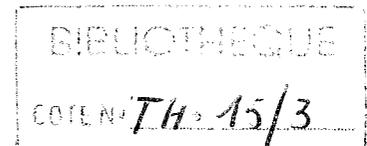
MAGISTER

Spécialité : GENIE MECANIQUE

PAR

Abdelmadjid HATNA

**CONTRIBUTION A L'ELABORATION D'UNE METHODOLOGIE D'EVIDEMENT DE POCHE
COMPLEXES
- APPLICATION A L'USINAGE DES SURFACES GAUCHES -**



Soutenue le 13 Décembre 1995 devant le jury composé de :

- MM. S. RECHAK, Professeur, ENP Alger, Président
A. BELAIDI, PhD, INGM Boumerdes, Directeur de thèse
H. BELHANECHÉ, Docteur, ENP Alger, Examineur
A. IGHILAZA, Docteur, USTBH - Beb azzouar, Examineur
K. RACHEDI, Chargé de cours, INGM Boumerdes, Examineur

REMERCIEMENTS

Le travail présenté dans ce mémoire de magister a été effectué au sein du laboratoire de CFAO de l'Institut National de Génie Mécanique (I.N.G.M) de Boumerdes.

Je tiens ici à remercier:

- Le professeur RECHAK Saïd de l'Ecole Nationale Polytechnique (ENP) - Alger pour bien avoir voulu accepter de juger ce travail et de présider le jury.
- M. H. BELHANECHÉ Hacène chargé de cours à l'ENP - Alger d'avoir accepté de juger ce travail et me faire l'honneur de participer à ce jury.
- M. IGHILAZA Abdelkrim chargé de cours à l'USTHB de Bebbouze pour l'intérêt qu'il a porté à mon travail, pour ces orientations et pour sa participation à ce jury.
- M. RACHEDI Kamel chargé de cours à l'INGM - Boumerdes pour ces conseils et pour avoir accepté de participer à ce jury.
- M. BELAÏDI Abderrahmane chargé de cours à l'INGM de Boumerdes et directeur de cette thèse pour son aide tout au long de ce travail.

Je remercie également le professeur Claude MARTY de l'INSA de Lyon (France) pour m'avoir permis de faire un séjour bénéfique dans son laboratoire et pour l'intérêt constant qu'il a porté pour ce travail de recherche. Sa lecture de ce mémoire et ses suggestions m'ont été très utiles. Que son équipe (B. KEDJA, C. BROURI, N. YUCEF et C. BARD) trouve ici tous mes remerciements.

Je voudrais aussi remercier tous ceux qui ont contribué à ce travail par leurs conseils et leur soutien:

- en premier lieu monsieur F. HOCINE directeur général de l'INGM pour ses encouragements et l'intérêt qu'il porte au développement des techniques nouvelles au sein de notre institut;
- M. T. KIRATI chef de département post-graduation à l'INGM;
- toute l'équipe du laboratoire de CFAO et en particulier M. I. BELAÏDI responsable du laboratoire et B. HASSAÏD pour ces encouragements;
- MM. M.O. SI-CHAÏB, M. BOUSSAÏD, A. DJERRADA et R. BENSAAÏDI ainsi que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

SOMMAIRE

REMERCIEMENTS	1
TABLES DES MATIERES	2
NOTATIONS UTILISEES	6
INTRODUCTION	7
<u>CHAPITRE I: MODELISATION MATHEMATIQUE DES COURBES ET DES SURFACES GAUCHES</u>	10
I-1. Introduction	10
I-2. Modélisation des courbes gauches intégrales	11
I-2-1. Courbes de Bézier	11
I-2-1-1. Propriétés des fonctions de base de Bernstein	12
I-2-1-2. Propriétés des courbes de Bézier	14
I-2-1-3. Algorithmes de calcul	15
I-2-2. Courbes B-splines	16
I-2-2-1. Notations	17
I-2-2-2. Expressions des fonctions de base	17
I-2-2-3. Vecteur nodal et différents types de fonctions B-splines	18
I-2-2-3-1. B-splines uniformes	18
I-2-2-3-2. B-splines non - uniformes	18
I-2-2-3-3. B-splines périodiques	19
I-2-2-4. Propriétés des fonctions de base B-splines	19
I-2-2-5. Propriétés des courbes B-splines	20
I-2-2-6. Algorithmes de calcul	21
I-3. Modélisation des courbes gauches rationnelles	24
I-3-1. Coordonnées homogènes	25
I-3-2. Courbes de Bézier rationnelles	25
I-3-2-1. Définition	25
I-3-2-2. Propriétés des courbes de Bézier rationnelles	26
I-3-2-3. Algorithme de calcul	27

I-3-3. Courbes NURBS	29
I-3-3-1. Définition	29
I-3-3-2. Propriétés des fonctions de base NURBS	30
I-3-3-3. Propriétés des courbes NURBS	31
I-3-3-4. Algorithmes de calcul	32
I-4. Modélisation des surfaces gauches	34
I-4-1. Surfaces paramétriques - formulation	35
I-4-2. Propriétés analytiques des fonctions de base	36
I-4-3. Propriétés des surfaces à pôles	36
I-4-4. Exemples	37
CHAPITRE II: EVIDEMENT DE POCHE PLANES	39
II-1. Introduction	39
II-2. Définition - poche et îlot -	40
II-3. Orientation des profils	43
II-3-1. Théorème de Green	43
II-3-2. Théorème des tangentes tournantes	43
II-3-3. Théorème de Jordan	44
II-3-4. Comparaison	44
II-3-5. Convention d'orientation	45
II-4. Calcul des courbes offset planes	47
II-4-1. Méthode de polygonisation	48
II-4-2. Méthode de discrétisation - conversion	48
II-4-3. Méthode d'approximation par des courbes de même type	49
II-4-3-1. Interpolation	49
II-4-3-2. Construction par lissage (méthode des moindres carrées)	52
II-4-3-3. Méthode du polygone caractéristique parallèle	53
II-4-3-4. Méthode de Coquillart	54
II-4-3-5. Méthode de Hoschek	55
II-4-3-6. Méthode de Klass	57
II-4-4. Problèmes de paramétrage	57
II-4-4-1. Paramétrage uniforme	58

II-4-4-2. Paramétrage proportionnel aux intervalles nodaux	58
II-4-4-3. Paramétrage proportionnel aux distances entre les points d'interpolation	58
II-4-4-4. Paramétrage proportionnel aux racines carrées des distances entre les points d'interpolation	59
II-4-5. Conclusion	60
II-5. Subdivision des courbes à pôles	62
II-5-1. Subdivision d'une courbe de Bézier	62
II-5-2. Elévation du degré d'une courbe de Bézier	63
II-5-3. Insertion de noeuds (B-splines non uniformes)	65
II-6. Calcul des intersections des courbes gauches	67
II-6-1. Techniques non linéaires	67
II-6-2. Techniques de subdivision	69
II-6-3. Techniques algébriques (technique d'implicitisation - inversion)	71
II-6-4. Autres techniques	72
II-6-5. Conclusion	73
II-7. Traitement des singularités et des interférences sur les trajectoires d'outil	75
II-7-1. Types de singularités	75
II-7-2. Solutions aux traitements des singularités	76
II-7-2-1. Points de rebroussement	76
II-7-2-1-1. Méthode de la tangente	76
II-7-2-1-2. Méthode de l'hodographe	78
II-7-2-1-4. Méthode proposée	79
II-7-2-2. Calcul des auto-intersections	80
II-7-2-2-1. Méthode de subdivision - élimination de Lasser	81
II-7-2-2-2. Méthode de Newton - Raphson	83
II-7-2-2-3. Comparaison	83
II-7-3. Zones indéfinies (les raccordements)	84
II-7-4. Zones de collision	85
II-7-5. Détection et traitement des boucles globales	86
II-7-5-1. Traitement des boucles intracontours	87
II-7-5-2. Traitement des boucles intercontours	95
II-7-5-3. Discussion	102

II-7-6. Les tests d'arrêt	104
II-7-7. Organisation des phases d'usinage	105
II-7-7-1. Evidement de poches sans îlots	105
II-7-7-2. Evidement de poches à un seul îlot	109
II-7-7-3. Généralisation à l'évidement de poches avec plusieurs îlots	111
II-8. Présentation du programme d'évidement de poches planes PMach	115
CHAPITRE III: APPLICATIONS	117
III-1. Usinage des surfaces gauches par la technique des plans de niveau	117
III-1-1. Intersection surface gauche / plan	118
III-1-2. Approximation des contours d'intersection	119
III-1-3. Evidement des poches	120
III-1-4. Finition	121
III-2. Usinage des poches sur les surfaces gauches	122
III-2-1. Calcul des intersections surface/droite	123
III-2-2. Calcul du pas de discrétisation	126
III-2-3. Calcul de l'erreur de projection	126
III-2-4. Calcul du pas de décalage (critère de la hauteur de crête)	127
III-3. Présentation du programme FFMach	130
III-4. Exemples de simulation d'usinage de poches sur des surfaces gauches	132
CONCLUSION	136
BIBLIOGRAPHIE	138

Notations utilisées

P_i : pôle d'une courbe à pôles

$P_{i,j}$: pôle d'une surface à pôles

h_i : coordonné homogène

H : application projective homogène

$P(u)$: point d'une courbe à pôles

$P(u,v)$: point d'une surface à pôles

d : distance de décalage

$B_{i,n}$: fonction de base de Bernstein

$N_{i,k}$: fonction de base B-spline

$R_{i,k}$: fonction de base rationnelle

$Q_{i,j}$: produit tensoriel de deux fonctions de base

U, V : suite nodale

u_i, v_i : élément de suites nodales

u, v : valeur paramétrique

n : vecteur normal unitaire, nombre de segments d'un polygone ou degré d'une courbe

m : nombre de segments d'un polygone ou degré d'une courbe

t : vecteur tangent

ε : erreur d'un calcul

ρ : rayon de courbure

N.B.: Certains algorithmes de ce document sont donnés dans un langage proche du PASCAL et du C. En effet les accolades { et } du C ont été prises pour désigner le début et la fin d'un bloc d'instructions tandis que le double slash // désigne un commentaire. Les conventions d'appel des procédures et des fonctions sont identiques à celles du C et du PASCAL.

« Dans quelques années on pourra dessiner directement des objets tridimensionnelles et les examiner comme s'ils étaient réels »

S.A. COONS (1967)

INTRODUCTION

INTRODUCTION

Depuis l'invention de l'ordinateur il y a un demi siècle, l'informatique ne cesse de s'introduire dans tous les secteurs de l'industrie moderne et de révolutionner les méthodes de travail. L'industrie mécanique n'échappe pas à cet événement; au contraire, c'est un domaine précurseur. Le technicien ou l'ingénieur de l'industrie mécanique jouit de l'assistance de l'ordinateur dans les étapes de conception, calcul, fabrication et contrôle du produit. La qualité et la productivité sont en croissance continue grâce à l'automatisation des tâches lentes, routinières ou difficiles et à la résolution de certains problèmes insolubles autrefois. Les techniques XAO prennent une importance considérable dans l'industrie mécanique, notamment la Conception et la Fabrication Assistées par Ordinateur (CFAO). Le taux de croissance annuel du marché mondial de la CFAO avoisine le chiffre de 40% pour un chiffre d'affaires d'une dizaine de milliards de dollars US (Revue « Industries et Techniques » N° 752 Septembre 1994).

La prolifération des systèmes de CFAO et de la commande numérique dans les P.M.E. et P.M.I. a été accompagnée d'un développement important des performances, des algorithmes et des fonctionnalités de ces systèmes. De simples formes géométriques élémentaires, la CFAO est passée aux surfaces de Coons [Coo 87], les formes de Bézier [Bezi 70, Bezi 86], les B-splines [Bar 88a], les bêta-splines [Bar 88b] et actuellement les NURBS [Pie 87, Pie 89a, Pie 89b, Pie 91] qui constituent le modèle géométrique le plus général et le plus performant connu jusqu'à présent.

De même, les algorithmes d'usinage deviennent davantage performants et les formes qu'ils peuvent traiter sont de plus en plus complexes. Dans plusieurs cas les procédures de calcul sont entièrement automatisées; l'utilisateur des Machines - Outils à Commande Numérique (MOCN) se voit ainsi libéré de tâches fastidieuses et parfois très difficiles de préparation des programmes d'usinage.

Le présent travail s'inscrit dans le cadre de l'automatisation de l'usinage d'un type de formes appelées poches rencontrées souvent en fabrication des outillages et des moules, en réalisation de maquettes et en gravure. Le calcul des trajectoires d'outil dans ce type d'usinage est très difficile, c'est pourquoi des recherches ont été entamées en vue de résoudre ce problème. Les poches que nous avons étudiées sont de formes presque quelconques. Elles peuvent être

décrites par des éléments géométriques simples ou gauches, avec ou sans îlots (le nombre d'îlots étant quelconque), situées sur un plan ou sur une surface gauche.

L'approche que nous avons retenue pour ce dernier cas était d'aplatir la poche, de calculer les trajectoires d'outil dans le plan par la méthode des décalages successifs et de les projeter ensuite sur la surface d'origine.

La génération des chemins d'outil pour le balayage des poches complexes planes avec ou sans îlots a fait l'objet d'une investigation détaillée et par un développement de méthodes inédites de traitement de ce type de formes. Certains systèmes de CFAO possèdent des fonctions similaires mais les techniques qu'ils utilisent sont inaccessibles au public. Par conséquent, les publications disponibles sur ce sujet sont rares [Per 78, Suh 90, Kimm 93, Hil 94] et celles disponibles ne traitent que des cas limités ou très peu détaillés.

La méthode des décalages successifs, qui est l'approche d'usinage que nous avons retenue, provoque souvent l'apparition de plusieurs types d'interférences; entre autres les boucles intracontours et les boucles intercontours. Une boucle intracontour est le résultat de recouvrement de plusieurs courbes d'un même contour tandis qu'une boucle intercontour provient du recouvrement de plusieurs courbes de deux contours différents. Le deuxième cas n'apparaît que dans le cas des poches avec îlots. Deux méthodes d'identification et de traitement de ce problème difficile ont été proposées et discutées en détail.

Un autre ensemble de techniques d'actualité en CFAO a fait l'objet d'une synthèse dans ce mémoire notamment les méthodes d'approximation des courbes offset (parallèles), la recherche des intersections et des auto-intersections et le traitement des collisions lors de l'usinage.

L'ensemble des solutions et techniques adoptées ou conçues dans la présente étude nous a permis de réaliser un utilitaire informatique appelé « PMach » pour l'évidement de poches complexes planes.

Deux applications utilisant cet utilitaire comme un sous programme ont été traitées. La première application concerne l'usinage des surfaces gauches par la technique des plans de niveau et la deuxième concerne l'usinage des poches sur des surfaces gauches. Nous avons proposé une optimisation du nombre de passes d'usinage de façon à respecter une hauteur de crête, une erreur de flèche et une erreur de projection admissibles. Pour ce type d'usinage l'étude réalisée concerne les fraiseuses à commande numérique de trois axes et en utilisant un outil à extrémité hémisphérique. Nous avons réalisé un programme informatique permettant la simulation de l'usinage des poches complexes sur les surfaces gauches.

Le présent mémoire se compose de trois chapitres:

Dans le premier chapitre nous avons défini les principaux modèles mathématiques utilisés en Conception Géométrique Assistée par Ordinateur (CGAO) notamment les courbes de Bézier rationnelles et non rationnelles, les courbes B-splines et NURBS et les surfaces biparamétriques rationnelles et non rationnelles. Les définitions mathématiques essentielles sont énoncées, les propriétés de ces formes ainsi que celles de leurs fonctions de base sont décrites et les algorithmes de calcul les plus utilisés sont donnés.

Au deuxième chapitre nous avons comparé en première étape les approches de balayage de poches planes, les méthodes d'approximation des trajectoires d'outil et les techniques de recherche des intersections entre les courbes gauches. Les méthodes de traitement des singularités et des interférences qui peuvent éventuellement se présenter sur une trajectoire d'outil (point de rebroussement, boucles, etc.) ont été abordées en deuxième étape. Ensuite des méthodologies de réalisation des opérations d'évidement de poches complexes ont été données.

Le troisième chapitre traite l'application des méthodes d'évidement de poches décrites dans le chapitre précédent dans l'usinage des surfaces gauches. Les problèmes d'intersection surface / plan et surface / droite ont été évoqués et les critères d'ajustement des erreurs de fabrication ont été discutés.

La conclusion générale du présent mémoire met en relief les difficultés majeures et l'orientation future de nos travaux de recherche.

« One picture is worth more than the thousand words »

Anonymous

CHAPITRE I

MODELISATION MATHEMATIQUE DES COURBES ET DES SURFACES GAUCHES

I. MODELISATION MATHEMATIQUE DES COURBES ET DES SURFACES GAUCHES

I-1. Introduction

En modélisation géométrique, les avantages des polynômes par rapport aux fonctions trigonométriques, exponentielles, etc. sont incontestables grâce notamment à la simplicité et à la rapidité des traitements qui leur sont appliqués au cours du processus de conception et à leur aptitude à représenter les formes les plus diverses.

Les polynômes peuvent être présentés sous trois formes:

- explicite du type $y = f(x)$ ou $z = g(x, y)$,
- implicite du type $f(x,y) = 0$ ou $g(x, y, z) = 0$,
- paramétrique du type $P(u)$ ou $P(u, v)$.

Le principal handicap des formes polynomiales explicites est la restriction du nombre de formes qu'elles peuvent modéliser. A titre d'exemple, les modèles explicites sont incapables de représenter les formes fermées qui sont pourtant très utilisées en CFAO toute comme elles sont incapables de représenter les courbes contenant des points multiples.

Le champ d'utilisation des formes implicites est plus étendu [Sed 84] mais elles se prêtent mal aux traitements de visualisation graphiques, aux transformations géométriques, au calcul automatique des chemins d'outil pour l'usinage [Fau 87] et au calcul des intersections [Sed 89].

Les formes polynomiales paramétriques apportent la solution à la plupart de ces problèmes: elles peuvent se calculer aisément, représenter les formes les plus diverses, subir des transformations géométriques sans changer leurs fonctions paramétriques correspondantes, etc.

Pour modéliser un ensemble de points qui ne possède pas préalablement de définitions mathématiques (les formes gauches), on fait appel généralement à l'une des trois méthodes [Che 86]: interpolation, lissage ou approximation.

Chacune des méthodes classiques d'interpolation (Lagrange, Newton, Hermite) [Ral 78, Dem 79] possède au moins un des inconvénients majeurs suivants:

- comportement oscillatoire,
- incapacité de générer des formes esthétiques,
- utilisation inconfortable en exigeant de l'utilisateur de fournir des tangentes ou des gradients sur certains points de la forme à modéliser.

L'utilisation d'une méthode de lissage comme les moindres carrées [Ral 78, Tre 83] à faibles degrés donne des approximations insuffisantes et imprécises des objets à modéliser. L'utilisation des degrés supérieurs pour palier à ce défaut provoque très souvent l'apparition des oscillations sur la forme modélisée [Vic 85].

L'emploi des formes polynomiales par morceaux de type splines cubiques [Fau 87, Qiu 87, Bar 88a, Ral 78] par exemple apporte des améliorations nettes aux résultats obtenus. Cependant le modèle splines cubiques est limité par les facteurs suivants:

- la modification, même d'un seul point sur l'objet, nécessite la réévaluation de tous les morceaux de cet objet;
- les splines sont parfois sujettes aux comportements oscillatoires;
- les splines sont généralement incapables de représenter les formes à tangentes verticales ou des formes à hauts degrés de continuité.

Les scientifiques et les chercheurs ont proposé des modèles mathématiques permettant de résoudre progressivement ces problèmes tels que les modèles de Coons [Coo 87], de Bézier [Bezi 70, Bezi 86] et les B-splines [Bar 88a].

I-2. Modélisation des courbes gauches intégrales

Les formes (courbes ou surfaces) gauches ou libres sont des formes imaginées ou calculées point par point par le concepteur sur la base d'un travail artistique ou expérimental, d'un calcul numérique ou d'un moyen d'acquisition numérique (digitalisation). Le concepteur connaît assez bien l'allure de cet objet mais pas l'équation mathématique. Le problème à résoudre se formule ainsi: étant donnés quelques points caractéristiques d'une forme géométrique, construire toute l'information nécessaire pour visualiser et/ou traiter la forme dans sa totalité. Le concepteur fournit ces points; un algorithme fournit la forme correspondante.

Plusieurs modèles mathématiques peuvent réaliser cette tâche. Quelques uns, parmi les plus employés en CFAO, feront l'objet de description dans les sections suivantes.

I-2-1. Courbes de Bézier

[Leo 91, Fau 87, Bar 88b, Bezi 70, Bezi 86, Rog 89a, Qiu 87, For 90]

Ce type de courbes était, il y a seulement une dizaine d'années, le modèle le plus répandu sur les systèmes de CFAO. Il était utilisé au début par le français P. Bézier [Bezi 70] sur le système

UNISURF de Renault pour la conception et la fabrication des carrosseries de voitures et ensuite vulgarisé par Forrest, Riesenfeld et autres pionniers de la CAO [Bezi 89, For 90].

Les courbes de Bézier sont représentées par des polynômes paramétriques. Un point de la courbe s'exprime par la formule:

$$P(u) = \sum_{i=0}^n B_{i,n}(u)P_i \quad (I-1)$$

où u est un paramètre réel variant de 0 à 1.

$P_i, i \in \{0,1,\dots,n\}$, est l'ensemble des points du plan ou de l'espace affine représentant les sommets - (pôles) d'un polygone appelé « *polygone caractéristique* » décrivant approximativement la forme de la courbe (Fig. I-1);

$n+1$ est le nombre de pôles de ce polygone caractéristique;

$B_{i,n}(u), i \in \{0,1,\dots,n\}$, est une suite de fonctions du paramètre réel u appelées fonctions de base.

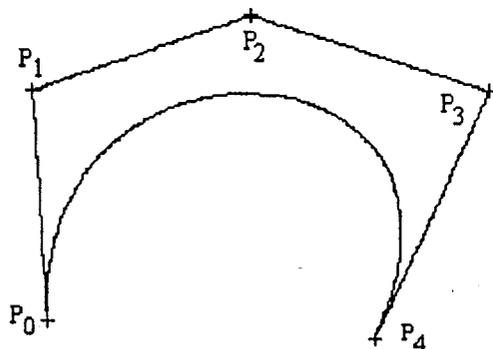


Fig. I-1 Une courbe de Bézier et son polygone caractéristique

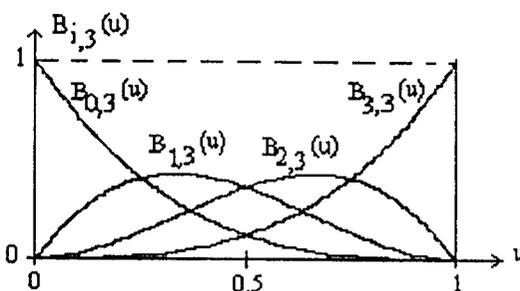


Fig. I-2 Fonctions de base de Bernstein cubiques

I-2-1-1. Propriétés des fonctions de base de Bernstein

(a) Les fonctions de base utilisées sur le modèle de Bézier sont les fonctions de Bernstein que l'on peut exprimer de la façon suivante:

$$B_{i,n}(u) = C_i^n \cdot u^i \cdot (1-u)^{n-i} \quad u \in [0,1], \quad (I-2)$$

$$C_i^n = \frac{n!}{i!(n-i)!} \quad (I-3)$$

C_i^n sont les coefficients du « triangle de Pascal ».

(b) Chaque fonction $B_{i,n}(u)$ est un polynôme de degré n , c'est à dire que le degré de la courbe est égal au nombre de cotés du polygone caractéristique (i.e. nombre de pôles - 1).

(c) Le degré de continuité est C^{n-1} et on démontre que pour le calcul des dérivées on a:

$$dB_{i,n}(u)/du = n[B_{i-1,n-1}(u) - B_{i,n-1}(u)] \quad (I-4)$$

donc un polynôme de degré $n-1$.

(d) Les fonctions de base de Bernstein vérifient la propriété de Cauchy. i.e.:

$$\sum_{i=0}^n B_{i,n}(u) = 1 \quad \forall u \in [0, 1] \quad (I-5)$$

cette propriété permet d'obtenir des courbes indépendantes du référentiel utilisé.

(e) Positivité: $B_{i,n}(u) \geq 0 \quad \forall u \in [0, 1]$; particulièrement aux extrémités on obtient:

$$* \text{ en } u = 0 \quad B_{0,n}(0) = 1, \quad B_{i,n}(0) = 0 \quad \forall i \in \{1, \dots, n\}$$

$$* \text{ en } u = 1 \quad B_{n,n}(1) = 1, \quad B_{i,n}(1) = 0 \quad \forall i \in \{0, \dots, n-1\}.$$

(f) Un calcul élémentaire [Leo 91] montre que $B_{i,n}(u) = B_{n-i,n}(1-u)$ ce qui signifie que les fonctions de base $B_{i,n}(u)$ sont deux à deux symétriques en u et $(u-1)$ par rapport à la droite $u=0.5$ (Fig. I.2).

(g) Relations de récurrence: une fonction de Bernstein peut s'exprimer à l'aide des fonctions du même type mais de degré inférieur [Leo 91, Pie 85, Pie 84] tel que:

$$B_{i,n}(u) = u B_{i-1,n-1}(u) + (1-u) B_{i,n-1}(u), \quad \forall u \in [0, 1], \quad (I-6a)$$

avec $B_{0,n}(u) = u B_{0,n-1}(u), \quad \forall i \in \{1, n-1\} \quad (I-6b)$

$$B_{n,n}(u) = u B_{n-1,n-1}(u) \quad (I-6c)$$

Ces relations sont à la base d'une partie importante des propriétés des courbes de Bézier.

(h) Sur l'intervalle $[0, 1]$, chaque fonction de base $B_{i,n}(u)$ est maximale pour une seule valeur qui est $u = i/n$ (Fig. I-2).

I-2-1-2. Propriétés des courbes de Bézier

(a) Les expressions (I-5 et I-6) montre que la courbe de Bézier est une forme polynomiale paramétrique de degré n tel que

$n = \text{nombre de pôles} - 1 = \text{nombre de cotés du polygone caractéristique}$.

Donc le degré du polynôme augmente impérativement avec la complexité de la courbe ce qui peut être considéré, du point de vue calcul et stabilité numérique, comme un inconvénient.

(b) Réversibilité: une courbe de Bézier décrite à partir du polygone caractéristique: P_0, P_1, \dots, P_n ne changera pas de forme lorsqu'elle sera décrite à partir du polygone caractéristique « inverse »: $P_n, P_{n-1}, \dots, P_1, P_0$.

(c) Définition intrinsèque: une courbe de Bézier est définie d'une façon intrinsèque grâce à ses fonctions de base vérifiant la propriété de Cauchy. Une courbe de Bézier est indépendante du référentiel utilisé ainsi que des transformations géométriques affines introduites [Bez 85, Leo 91, Rog 89].

(d) La courbe suit généralement la forme de son polygone caractéristique (Fig. I-3 et I-4).

(e) La courbe passe par le premier et le dernier point du polygone caractéristique tel que

$$P(u = 0) = P_0 \text{ et } P(u = 1) = P_n.$$

(f) Les vecteurs tangents aux extrémités de la courbe ont les mêmes directions que le premier et le dernier segment du polygone caractéristique.

(g) Propriété du polygone convexe: la courbe de Bézier est située à l'intérieur du polygone convexe associé à son polygone caractéristique.

(h) Comportement global: le changement de la position d'un pôle affecte l'ensemble de la courbe de Bézier (Fig. I-5).

(i) Propriété de la variation décroissante: le nombre de points d'inflexion d'une courbe de Bézier est inférieur ou égal au nombre d'ondulations sur son polygone caractéristique.

I-2-1-3. Algorithmes de calcul

On utilise généralement trois méthodes pour évaluer un point courant d'une courbe de Bézier: la méthode de Horner pour le calcul des formes polynomiales, l'algorithme de de Casteljau [Cas 85] et l'algorithme de Farouki [Leo 91]. L'algorithme de De Casteljau, bien

qu'il est lent par rapport aux deux autres algorithmes, offre l'avantage de permettre l'évaluation simultanée du point et des dérivées partielles en ce même point.

L'algorithme de De Casteljaou pour l'évaluation d'un point $P(u)$ comporte les étapes suivantes:

soit une courbe de Bézier de $n+1$ pôles P_0, P_1, \dots, P_n

{

// Initialisation:

FOR $i = 0$ TO n { $P_i^{[0]} = P_i$ }

// Récursion:

FOR $j = 1$ TO n

{ FOR $i = 0$ TO $n-j$

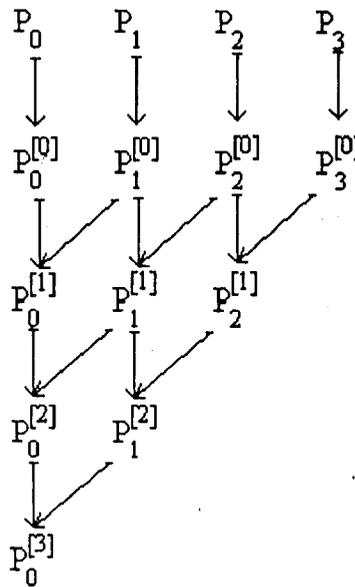
{ $P_i^{[j]} = (1-u)P_i^{[j-1]} + uP_{i+1}^{[j-1]}$ }

}

$P(u) = P_0^{[n]}$

}// END

Ce processus de calcul est représenté, pour une courbe cubique, par le schéma suivant:



La première dérivée est donnée par:

$$dP(u)/du = n(P_1^{[n-1]} - P_0^{[n-1]}) \quad \forall u \in [0, 1] \quad (I-7)$$

La deuxième dérivée a comme expression:

$$d^2P(u)/d^2u = n(n-1)(P_2^{[n-2]} - 2P_1^{[n-2]} + P_0^{[n-2]}) \quad \forall u \in [0, 1] \quad (I-8)$$

... etc.

Cette facilité de calcul des dérivées partielles sera utilisée pour des multiples traitements des courbes de Bézier dans les sections suivantes.

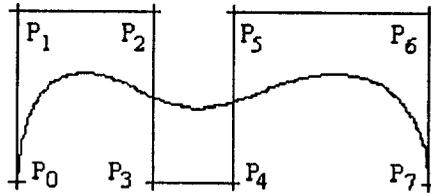


Fig. I-3 Une courbe de Bézier de degré 7

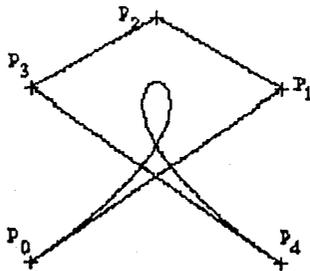


Fig. I-4 Une courbe de Bézier avec une boucle

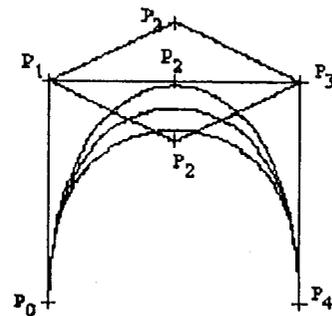


Fig. I-5 Comportement global d'une courbe de Bézier
(déplacement du pôle P_2)

I-2-2. Courbes B-splines

[Leo 91, Fau 87, Bar 88b, Rog 89a, Qiu 87, Pie 87]

Le modèle B-splines qui vient de « Basis Splines » i.e. « splines de base » est la généralisation logique de l'interpolation polynomiale cubique d'Hermite appelée splines naturelles [Bar 88a]. Physiquement le mot « spline » (ou latte en français) désigne un instrument utilisé depuis des siècles dans les chantiers navals pour la conception et le tracé des coques de bateaux ainsi que dans les industries aéronautiques; c'est une poutre mince en métal, plastique ou en bois qui se déforme facilement pour passer entre les points donnés en conception. L'utilisateur déforme la latte au moyen de poids pour obtenir la courbes désirée .

En CAO, la spline « physique » est remplacée par la spline mathématique et ce modèle existait pendant plusieurs années sur des modeleurs géométriques. Il a fait l'objet de plusieurs enrichissements en algorithmes de traitement [Bar 88a, Qiu 87, Fau87].

Le modèle B-splines qui utilise les splines comme fonctions de base fut utilisé pour la première fois en CAO par Riesenfeld. Il permet d'accroître considérablement la souplesse de manipulation des courbes et des surfaces.

I-2-2-1. Notations

On note:-

* $N_{i,m}(u)$ les fonctions de base B-splines de *degré* m , donc d'*ordre* $m+1$ pour le paramètre réel u variant de u_{\min} à u_{\max} ;

* P_i , $i \in \{0,1,\dots,n\}$, *pôles* ou sommets du *polygone caractéristique* de la courbe.

* $U (u_0, u_1, \dots, u_p)$ une suite croissante de valeurs réelles appelée *vecteur ou suite nodal* avec:

$$u_0 = u_{\min} \quad \text{et} \quad u_p = u_{\max}.$$

I-2-2-2. Expression des fonctions de base

Les fonctions de base $N_{i,m}(u)$ peuvent être définies sur l'intervalle $[u_{\min}, u_{\max}]$ par la formule de récurrence de de Boor [Qiu 87, Leo 91] suivante:

$$N_{i,m}(u) = \frac{u - u_i}{u_{i+m} - u_i} N_{i,m-1}(u) + \frac{u_{i+m+1} - u}{u_{i+m+1} - u_{i+1}} N_{i+1,m-1}(u) \quad (\text{I-9a})$$

avec

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1} \\ 0 & \text{si non} \end{cases} \quad (\text{I-9b})$$

Lorsque $u_{i+m} = u_i$ ou $u_{i+m+1} = u_{i+1}$ on adopte la convention $x / 0 = 0$

I-2-2-3. Vecteur nodal et différents types de courbes B-splines

L'expression (I-9a) montre clairement que la valeur des fonctions B-splines de base $N_{ij}(u)$, donc de la courbe à obtenir, est bien influencée par le choix des composantes du vecteur nodal U . La seule condition exigée lors de la construction d'un vecteur nodal est d'avoir des composantes à valeurs croissantes i.e. $u_{i+1} \geq u_i$. En effet, trois types fondamentaux de vecteurs nodaux (ou de B-splines) sont utilisés: les B-splines uniformes, les B-splines non - uniformes et les B-splines périodiques.

I-2-2-3-1. B-splines uniformes

Les fonctions de base et les courbes B-splines uniformes sont obtenues par un vecteur nodal à éléments croissant à pas constant: $u_{i+1} - u_i = \Delta u$. Souvent on choisit $u_0 = 0$ et $\Delta u = 1$ ou 0.1 .

exemples: $U = [0, 1, 2, 3, 4, 5]$, $U = [0, 0.2, 0.4, 0.6, 0.8, 1]$.

La figure I-6 montre la forme des fonctions de base B-splines uniformes quadratiques. La figure I-7 montre un exemple d'une courbe B-spline uniforme cubique de six pôles.

I-2-2-3-2. B-splines non - uniformes

Dans ce cas le vecteur nodal contient des éléments (ou noeuds) réels; l'intervalle entre deux noeuds successifs n'est pas constant bien que la suite reste toujours croissante.

Un vecteur non - uniforme peut contenir deux ou plusieurs éléments égaux appelés « noeuds multiples ».

exemples: $[0, 1, 2, 3, 3, 4, 5]$, $[0, 1, 1, 1, 2, 3]$, $[0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]$.

Le dernier exemple (uniforme hors ces deux extrémités et aux noeuds multiples aux extrémités) est le type le plus utilisé en CFAO. Sa forme courante est:

$0, 0, \dots, 0, 1, 2, \dots, k, k, \dots, k$

└──────────┘

$m+1$ fois

└──────────┘

$m+1$ fois

soit $u_i = 0$ $0 \leq i \leq m$

$$u_i = i-m \quad m+1 \leq i \leq n$$

$$u_i = n+1-m \quad n+1 \leq i \leq n+m+1$$

La figure I-8 montre la forme des fonctions de base B-splines non - uniformes quadratiques.

I-2-2-3-3. B-splines périodiques

Dans ce cas les fonctions de base sont périodiques de période $u_{n+1} - u_0$ avec:

$$u_{n+1+j} = u_j + (u_{n+1} - u_0) \text{ pour } 1 \leq j \leq m$$

$$\text{et } N_{i,k}(u) = N_{i,k}(u + u_{n+1}) \text{ pour } u \in [u_0, u_m] \text{ et } n-m+1 \leq i \leq n+1.$$

Le vecteur nodal peut être uniforme ou non - uniforme. Une telle disposition permet de construire des courbes fermées possédant une continuité élevée en tout point de la courbe [Leo 91, Bar 88a].

I-2-2-4. Propriétés des fonctions de base B-splines

Les fonctions de base $N_{i,k}(u)$ possèdent les propriétés importantes suivantes:

(a) Positivité: $N_{i,k}(u) \geq 0$ pour toute valeur de i, k et u .

(b) Propriété de Cauchy:
$$\sum_{i=0}^n N_{i,m}(u) = \sum_{i=j-m}^j N_{i,m}(u) = 1 \quad (\text{I-10})$$

où $m =$ degré des fonctions de base, pour tout $u \in [u_0, u_p]$ ou $u \in [u_j, u_{j+1}]$ où $p = n+m+1$ dans le cas des B-splines non - uniformes.

(c) Propriété du support local: $N_{i,m}(u) = 0$ si $u \notin [u_i, u_{i+m+1}]$. De plus, dans un intervalle nodal donné, il n'existe que $m+1$ fonctions de base $N_{i,m}(u)$ non nulles.

(d) Dérivation: $N_{i,m}(u)$ est un polynôme et toutes ses dérivées donnent des polynômes valides sur l'intervalle nodal de $N_{i,m}(u)$. Pour une valeur nodale de multiplicité k , la fonction $N_{i,m}(u)$ possède une continuité d'ordre $m-k$.

(e) Sauf pour $m = 0$, $N_{i,m}(u)$ possède un seul extremum.

(f) Le vecteur nodal ayant la forme $U[0, \dots, 0, 1, \dots, 1]$, où le 0 et le 1 sont répétés $m+1$ fois, donne une fonction de base du type Bernstein de degré $m = n$.

I-2-2-5. Propriétés des courbes B-splines

Une courbe B-spline de degré m est définie par l'expression:

$$P(u) = \begin{pmatrix} x(u) \\ y(u) \\ z(u) \end{pmatrix} = \sum_{i=0}^n N_{i,m}(u) P_i, \quad u \in [u_0, u_p], \quad (I-11)$$

où les $P_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$ sont les pôles du polygone caractéristique de la courbe et les $N_{i,m}(u)$ sont les

fonctions de base de degré m définies par l'équation (I-9) pour un vecteur nodal $U[u_0, \dots, u_p]$. Le nombre de cotés sur le polygone caractéristique (n), le degré (m) et le nombre de valeurs nodales ($p+1$) sont liés par la relation: $p = n+m+1$. Les figures I-11, I-12 et I-13 montrent quelques exemples de courbes B-splines.

Les courbes B-splines possèdent des propriétés géométriques intéressantes dues à la qualité des fonctions de bases employées (cf. §I-2-2-4) ainsi qu'à la forme de récurrence (I-9) de Cox-de Boor. Ces propriétés sont:

(a) Définition intrinsèque: l'utilisation d'un vecteur nodal dont la première et la dernière valeur nodale sont de multiplicité $m+1$ (donc vecteur nodal non uniforme) donne les conditions aux limites suivantes:

$$P(u_0) = P(u_m) = P_0$$

$$P(u_{n+1}) = P(u_{n+m+1}) = P_n$$

$$\frac{dP}{du}(u_0) = \frac{m(P_1 - P_0)}{u_{m+1} - u_m},$$

$$\frac{dP}{du}(u_{n+1}) = \frac{m(P_n - P_{n-1})}{u_{n+1} - u_n}$$

...

(b) Une courbe B-spline est définie intrinsèquement grâce à ses fonctions de base vérifiant la propriété de Cauchy. La forme d'une courbe B-spline est invariante si on lui applique une transformation affine; autrement dit une transformation affine appliquée à une courbe B-spline donne le même effet si elle est appliquée à son polygone caractéristique.

(c) Propriété du polygone enveloppe convexe: une courbe B-spline est toujours située à l'intérieur du polygone convexe associé à son polygone caractéristique.

(d) La courbe suit généralement l'allure de son polygone caractéristique; ce dernier représente l'approximation linéaire de la courbe. Cette approximation peut être affinée par l'insertion de noeuds [Boe 80, Bar 88b, Boe 85, Lee 92] et la subdivision.

(e) Modification locale: le support local des fonctions de base $N_{i,k}(u)$ limite la zone d'influence de chaque pôle. Le déplacement d'un pôle P_i n'affectera la courbe que pour $u \in [u_i, u_{i+m+1}]$ (Fig. I-12). Cette propriété est un avantage incontestable des B-splines par rapport aux courbes de Bézier.

(f) Dérivation et continuité: $P(u)$ est infiniment dérivable pour tout u ne présentant pas une valeur nodale. En effet, $P(u)$ possède une continuité C^{m-k} si u est une valeur nodale de multiplicité k .

(g) Propriété de la variation décroissante: le nombre d'intersections d'une courbe B-spline et une droite quelconque est toujours inférieur ou égal au nombre d'intersections de cette même droite avec le polygone caractéristique de la même courbe.

(h) Une courbe B-spline de degré égale au nombre de cotés de son polygone caractéristique ($m = n$) est une courbe de Bézier. Donc le modèle de Bézier n'est qu'un cas particulier du modèle B-spline (Fig. I-11).

I-2-2-6. Algorithmes de calcul

L'algorithme de calcul d'un point courant $P(u)$ d'une courbe B-spline le plus utilisé est l'algorithme de Cox-de Boor [Leo 91]. On peut l'énoncer de la façon suivante:

* données: n, m, P_i et U

* chercher l'indice j tel que: $u \in [u_j, u_{j+1}[$

* initialisation:

FOR $i = j - m$ TO j { $P_i^{[0]} = P_i$ }

* récursion:

FOR $k = 1$ TO m

{ FOR $i = j$ DOWN TO $j - m + k$

$$\{ P_i^{[k]} = P_i^{[k-1]} \left(\frac{u - u_i}{u_{i+m+1-k} - u_i} \right) + P_{i-1}^{[k-1]} \left(\frac{u_{i+m+1-k} - u}{u_{i+m+1-k} - u_i} \right) \}$$

}

$$P(u) = P_j^{[m]}$$

* FIN

Cet algorithme permet, comme dans le cas de l'algorithme de de Casteljau pour les courbes de Bézier, de déduire les valeurs des dérivées partielles de $P(u)$ tel que:

$$\frac{dP}{du}(u) = \frac{m}{u_{j+1} - u_j} (P_j^{[m-1]} - P_{j-1}^{[m-1]}) \quad (I-12)$$

$$\frac{d^2P}{du^2}(u) = \frac{m(m-1)}{u_{j+1} - u_j} \left(\frac{P_j^{[m-2]} - P_{j-1}^{[m-2]}}{u_{j+2} - u_j} - \frac{P_{j-1}^{[m-2]} - P_{j-2}^{[m-2]}}{u_{j+1} - u_{j-1}} \right) \quad (I-13)$$

...

Le schéma suivant illustre l'application de cet algorithme pour le calcul d'un point sur une courbe B-spline cubique ($m = 3$):

$$\begin{array}{cccccccccccc}
 P_0 & P_1 & \dots & P_{j-4} & P_{j-3} & P_{j-2} & P_{j-1} & P_j & P_{j+1} & \dots & P_n \\
 & & & & P_{j-3}^{[0]} & P_{j-2}^{[0]} & P_{j-1}^{[0]} & P_j^{[0]} & & & & \\
 & & & & & P_{j-2}^{[1]} & P_{j-1}^{[1]} & P_j^{[1]} & & & & \\
 & & & & & & P_{j-1}^{[2]} & P_j^{[2]} & & & & \\
 & & & & & & & P_j^{[3]} & & & &
 \end{array}$$

Les détails théoriques de cet algorithme sont donnés dans [Leo 91].

Remarque: L'algorithme d'insertion de noeuds de Boehm [Boe 80, Boe 85], l'algorithme d'Oslo [Bar 88b] et l'algorithme de Choi et al [Cho 90] peuvent être utilisés pour calculer les formes B-splines.

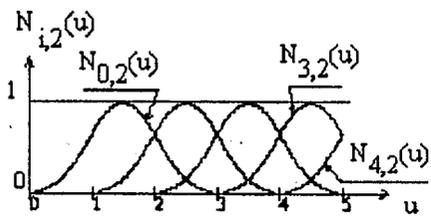


Fig. I-6 Fonctions de base B-splines uniformes
($n = 4, m = 2$)

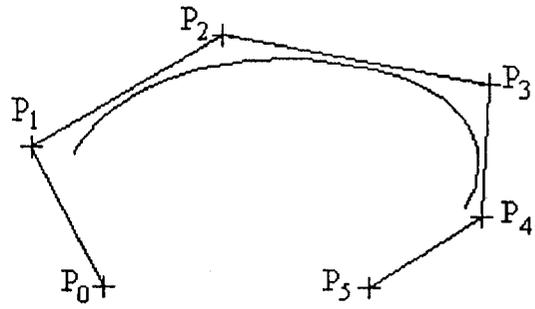


Fig. I-7 Une courbe B-spline cubique uniforme
à six pôles ($n = 5, m = 3$)

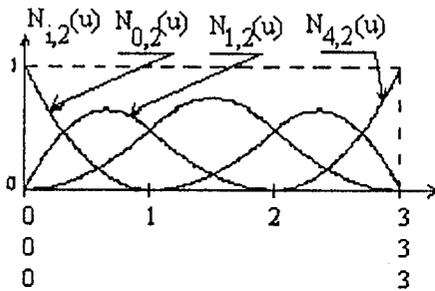


Fig. I-8 Fonctions de base B-splines non-uniformes
($n = 4, m = 2$)

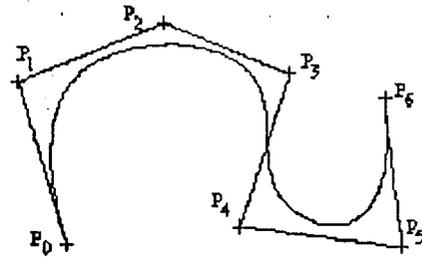


Fig. I-9 Une courbe B-spline non uniforme
($n = 6, m = 3$)

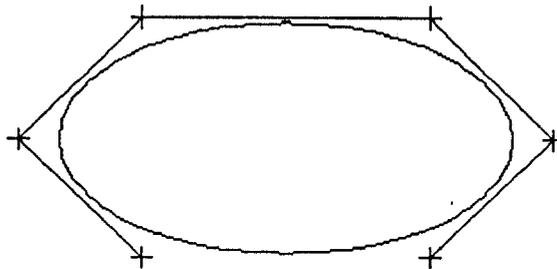


Fig. I-10 Une courbe B-spline périodique ($n = 5, m = 3$)

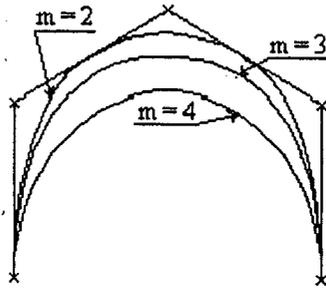


Fig. I-11 Influence du degré sur la forme d'une courbe B-spline

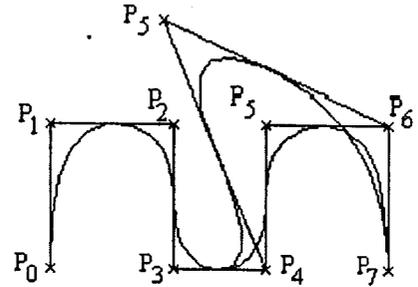


Fig. I-12 Influence locale du déplacement d'un pôle (P_5) sur une courbe B-spline

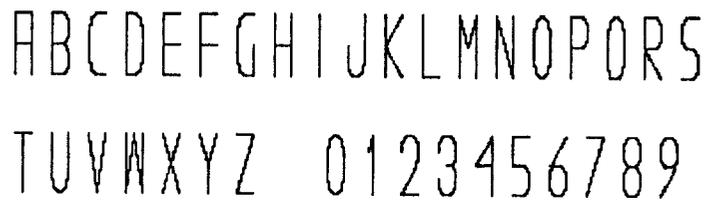


Fig. I-13 L'alphabet latin et les chiffres arabes modélisés par des courbes B-splines uniformes quadratiques

I-3. Modélisation des courbes gauches rationnelles

Les modèles géométriques rationnels font appel au concept des coordonnées homogènes [Rie 81]. Le besoin de possession d'un tel modèle a été ressenti depuis les premiers débuts des systèmes de CFAO. La première utilisation de ces modèles remonte, probablement, aux environs de 1970 avec l'intégration des courbes de Bézier rationnelles dans le logiciel CONSURF de la British Aircraft corporation.

Le premier avantage des formes rationnelles est leur aptitude à présenter une classe plus importante de formes (quadratiques et coniques) donc à faciliter l'homogénéisation des bases de données des logiciels [Leo 91, Gar 87].

Le premier modéleur géométrique qui a mis en profit cet avantage (utilisation d'un seul modèle mathématique pour représenter et gérer toutes les entités géométriques) est le système Geomod de la Structural Dynamics Research Corporation en 1978. Le système fut commercialisé aux environs de 1983 [Pie 91]; et depuis les formes rationnelles ne cessent pas de gagner le terrain.

I-3-1. Coordonnées homogènes

L'emploi des coordonnées homogènes en CAO offre plusieurs avantages [Leo 91, Che 86, Tos 90, Rie 81, Pie 91, Pie 87, Rog 89]; entre autres:

- * la possibilité de combinaison de toutes les transformations géométriques courantes (translation, rotation, symétrie, homothétie, ...) dans une seule matrice de transformation;
- * le codage des points situés à l'infini;
- * la représentation des points affines ordinaires:

Soit $P^* = (x \ y \ z \ h)^T$, $h \neq 0$, la représentation homogène d'un point P de coordonnées $(x/h \ y/h \ z/h)^T$ dans l'espace euclidien tridimensionnel E^3 . Deux situations particulières peuvent être constatées:

- si $h = 0$, le point $P^h(x \ y \ z \ 0)^T$ est projeté suivant un point rejeté à l'infini sur la droite de direction OP^h ; alors $(x \ y \ z \ 0)^T$ et $(-x \ -y \ -z \ 0)^T$ représentent le même point [Rie 81].
- $P^h = (0 \ 0 \ 0 \ 0)^T$ ne représente aucun point [Rie 81, Leo 91, Pie 91].

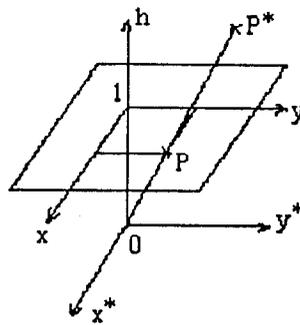


Fig. I-14 Coordonnées homogènes

I-3-2. Courbes de Bézier rationnelles

I-3-2-1. Définition

Soit un polygone caractéristique formé par les sommets (pôles) P_0, P_1, \dots, P_n ayant respectivement les poids h_0, h_1, \dots, h_n ; un point courant d'une courbe rationnelle de Bézier est défini par l'expression:

$$P(u) = \frac{\sum_{i=0}^n B_{i,n}(u)h_i P_i}{\sum_{i=0}^n B_{i,n}h_i} = \sum_{i=0}^n R_{i,n}(u)P_i, \quad u \in [0,1], \quad (I-14)$$

où les $B_{i,n}(u)$ sont les fonctions de base de Bernstein et les $R_{i,n}(u)$ sont les fonctions de base rationnelles de Bernstein. Les fonctions $R_{i,n}(u)$ sont définies par la relation

$$R_{i,n}(u) = \frac{B_{i,n}(u)h_i}{\sum_{j=0}^n B_{j,n}(u)h_j}, \quad u \in [0, 1] \quad (I-15)$$

Les figures I-15 et I-16 montrent l'effet de la variation du poids d'un pôle sur le comportement d'une courbe de Bézier rationnelle. Ces figures montrent un comportement « intuitif » et analogue à celui d'une courbe de Bézier non rationnelle.

Les figures I-17 et I-18 montrent que l'introduction d'une coordonnée homogène négative peut changer « radicalement » l'orientation de la tangente, de la courbure et le comportement de la courbe vis-à-vis du polygone caractéristique. Pour éviter ces désagréments et écarter la possibilité d'annulation du dénominateur en (I-15) en certaines valeurs de u , les coordonnées homogènes doivent être toutes de même signe. Le choix le plus naturel conduit à prendre tous les $h_i > 0$; ce type de coordonnées homogènes positives s'appelle *coordonnées homogènes admissibles*.

I-3-2-2. Propriétés des courbes de Bézier rationnelles

(a) Invariance de la forme de la courbe lors d'un changement de sens de parcours: la forme de la courbe reste inchangée lorsque l'ordre des sommets du polygone caractéristique ainsi que celui de la séquence des coordonnées homogènes h_i est inversé étant donné que la propriété de symétrie des polynômes de Bernstein est vérifiée.

(b) Comportement aux extrémités du polygone caractéristique:

$$P(u = 0) = P_0 \quad \text{et} \quad P(u = 1) = P_n$$

Le premier et le dernier coté du polygone caractéristique définissent la direction de la tangente de la courbe en $u = 0$ et $u = 1$ respectivement si les coordonnées homogènes sont admissibles.

- (c) Si les h_i sont toutes identiques, on obtient exactement une courbe de Bézier non rationnelle.
- (d) Pour une suite de coordonnées homogènes positives, la courbe se rapproche du sommet P_i lorsque la valeur de h_i augmente. Si la valeur de h_i tend à s'annuler, la forme de la courbe évolue de façon à s'éloigner du sommet P_i ; il y aura tendance à annuler l'influence de P_i . Si h_i est négative, le polygone caractéristique repousse la courbe (Fig. I-17 et I-18).
- (e) Invariance de la courbe lors d'une transformation affine: cette propriété est due au fait que les fonctions de base $R_{i,n}(u)$ vérifient la condition de Cauchy:

$$\sum_{i=0}^n R_{i,n}(u) = 1, \forall u \in [0,1], \quad (I-16)$$

- (f) Le support global: comme pour les courbes de Bézier non rationnelles, le changement de la position d'un pôle ou de la valeur d'un poids affecte l'ensemble de la courbe (Fig. I-15 et Fig. I-16)
- (g) Propriété du polygone enveloppe convexe: toute *séquence admissible* de coordonnées homogènes donne une courbe située entièrement à l'intérieur du polygone convexe enveloppe associé au polygone caractéristique.
- (h) Propriété de la variation décroissante: pour une séquence de coordonnées homogènes admissibles, le nombre de points d'intersection d'une courbe avec un plan quelconque est inférieur ou égal au nombre de points d'intersection de son polygone caractéristique avec ce même plan.

I-3-2-3. Algorithme de calcul

L'algorithme de calcul d'un point courant sur une courbe de Bézier rationnelle que nous donnons ici est l'extension de l'algorithme de de Casteljau:

* données:

n : nombre de cotés du polygone caractéristique,

$P_i, i \in \{0, 1, \dots, n\}$, est l'ensemble des pôles du polygone caractéristique,

$h_i, i \in \{0, 1, \dots, n\}$, sont les coordonnées homogènes associées aux pôles P_i ;

* calcul:

- initialisation:

FOR $i = 0$ TO n { $P_i^{[0]} = P_i, h_i^{[0]} = h_i$ }

- récursion:

FOR j = 1 TO n

{ FOR i = 0 TO n-j

{

$$h_i^{[j]} = (1-u)h_i^{[j-1]} + uh_{i+1}^{[j-1]}$$

$$P_i^{[j]} = P_i^{[j-1]} \frac{h_i^{[j-1]}}{h_i^{[j]}} (1-u) + P_{i+1}^{[j-1]} \frac{h_{i+1}^{[j-1]}}{h_i^{[j]}} u$$

}

}

$$P(u) = P_0^{[n]}$$

* FIN

Le lecteur intéressé pourra consulter la référence [Leo 91] pour les détails de l'interprétation géométrique de cet algorithme.

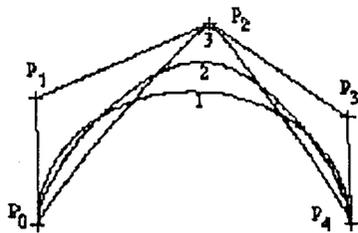


Fig. I-15 Influence du poids sur la forme d'une courbe rationnelle; $h_2 = \{1, 30, 50\}$

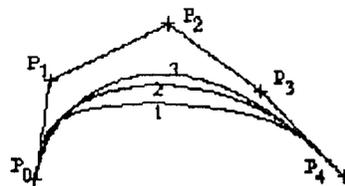


Fig. I-16 Influence de la valeur de h ; $h_2 = \{0, 0.5, 1\}$

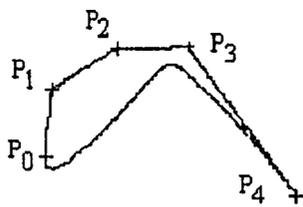


Fig. I-17 Inversion de la tangente ($h_2 = -0.9$)

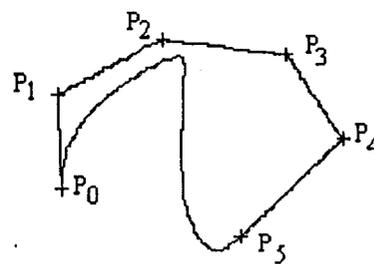


Fig. I-18 Inversion de la tangente ($h_4 = -0.9$)

I-3-3. Courbes NURBS

[Leo 91, Bar 88b, Rog 89a, Qiu 87, Pie 87, Pie 91, Pie 89a, Bal 86, Cho 90]

Les courbes NURBS (Non Uniform Rational B-Spline) sont, comme leur nom l'indique, des courbes B-splines non uniformes rationnelles.

Les NURBS sont le modèle géométrique le plus général utilisé jusqu'aujourd'hui en CAO. Il inclue les propriétés des modèles de Bézier, Bézier rationnelles et les B-splines. Les excellentes propriétés géométriques et analytiques des NURBS ont permis leur prolifération et leur incorporation rapide dans les systèmes de CAO, CFAO et dans les standards nationaux et internationaux d'échange de données de CAO tels que IGES, PHIGS+, Product Data Exchange Specification et International Standard Office.

I-3-3-1. Définition

Un point $P(u) = (x \ y \ z)^T$ d'une courbe NURBS dans l'espace cartésien 3D est représenté par quatre coordonnées homogènes $(hx \ hy \ hz \ h)^T$, $h > 0$, notées P^h . Le point $P(u)$ est alors décrit par:

$$P(u) = H\{ P^h(u) \} = H\left\{ \sum_{i=0}^n N_{i,m}(u) P_i^h \right\} \quad (I-17)$$

$$P(u) = \frac{\sum_{i=0}^n N_{i,m}(u) h_i P_i}{\sum_{i=0}^n N_{i,m}(u) h_i} = \sum_{i=0}^n R_{i,m}(u) P_i \quad (I-18)$$

où $P_i = H\{ P_i^h \} = (x_i \ y_i \ z_i)^T$ est un pôle du polygone caractéristique associé à la courbe;

n : le nombre de cotés du polygone caractéristique;

m : le degré de la courbe;

u : un paramètre variant sur l'intervalle de la séquence nodale associée;

H : l'application projective homogène d'un point P^h de R^4 vers R^3 (de R^3 vers R^2 pour les courbes planes) [Rie 81].

Les quantités définies par l'expression:

$$R_{i,m}(u) = \frac{N_{i,m}(u)h_i}{\sum_{j=0}^n N_{j,m}(u)h_j} \quad (I-19)$$

sont les fonctions de base NURBS.

Les figures I-19 et I-20 montrent des exemples de fonctions de base NURBS.

I-3-3-2. Propriétés des fonctions de base NURBS

(a) Positivité: $R_{i,m}(u) \geq 0$ pour toute valeur de i , m et u .

(b) Propriété de Cauchy: $\sum_{i=0}^n R_{i,m}(u) = 1, \forall u \in [u_{\min}, u_{\max}]$.

(c) Support local: $R_{i,m}(u) = 0$ si $u \notin [u_i, u_{i+m+1}]$; sur un intervalle nodal donné, il existe au maximum $m+1$ fonctions de base $R_{i,m}(u)$ non nulles.

(d) Dérivabilité: à l'intérieur des intervalles nodaux, toutes les dérivées de $R_{i,m}(u)$ existent. Pour un noeud de multiplicité k , $R_{i,m}(u)$ est $(m-k)$ fois continûment dérivable.

(e) Extremum: sauf pour le cas où $m = 0$, $R_{i,m}(u)$ possède uniquement un seul extremum qui est en fait un maximum (Fig. I-19 et Fig. I-20).

(f) $R_{i,m}(u)$ est une généralisation des fonctions de base B-splines. En effet, $R_{i,m}(u) = N_{i,m}(u)$ si tous les éléments de la suite des coordonnées homogènes h_i sont égaux:

$$\{h_0, h_1, \dots, h_n\} = \{h, h, \dots, h\}.$$

$R_{i,m}(u)$ est une généralisation des fonctions de base de Bernstein rationnelles si $m = n$.

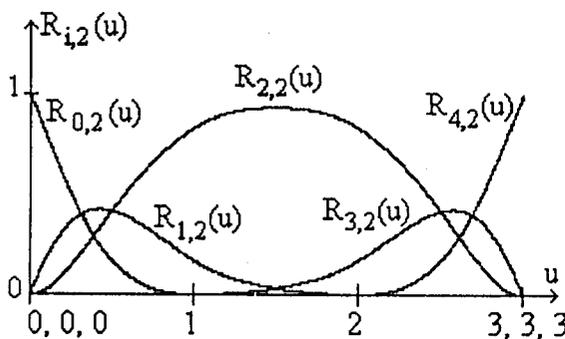


Fig. I-19 Exemple de fonctions de base NURBS
($n = 4, m = 2$); $h_i = \{1, 1, 5, 1, 1\}$

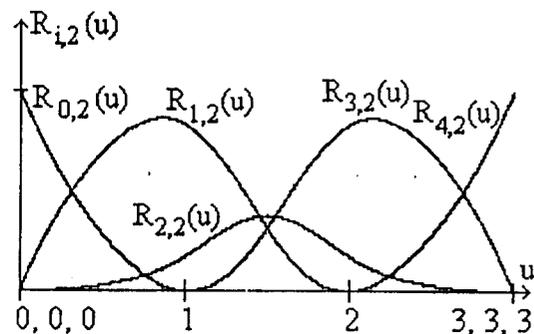


Fig. I-20 Exemples de fonctions de base NURBS
($n = 4, m = 2$); $h_i = \{1, 1, 0.2, 1, 1\}$

I-3-3.3. Propriétés des courbes NURBS

Les propriétés suivantes sont données pour *les séquences de coordonnées homogènes admissibles*.

(a) Comportement aux extrémités:

$$P(u = u_{\min}) = P_0, \quad P(u = u_{\max}) = P_n \quad (\text{I-20})$$

$$dP/du(u = u_{\min}) = mh_1(P_1 - P_0)/h_0(u_{m+1} - u_m) \quad (\text{I-21})$$

$$dP/du(u = u_{\max}) = mh_{n-1}(P_n - P_{n-1})/h_n(u_{n+1} - u_n) \quad (\text{I-22})$$

...

(b) Invariance affine et projective: une application affine ou projective appliquée sur le polygone caractéristique de la courbe donne le même résultat si elle est appliquée sur la courbe elle-même.

(c) Polygone enveloppe convexe: pour tout $u \in [u_j, u_{j+1}[$, $m \leq j \leq n$, la courbe $P(u)$ est entièrement située à l'intérieur du polygone enveloppe convexe issu du polygone caractéristique associé à la courbe.

(d) Si $m = 1$ alors la courbe est exactement confondue avec son polygone caractéristique.

(e) Support local: si on change la position d'un pôle ou la valeur d'un poids, la courbe ne sera affectée que sur $m+1$ intervalles nodaux.

(f) Dérivation et continuité: pour tout u qui n'est pas une valeur nodale, $P(u)$ est infiniment dérivable. Si u est égal à une valeur nodale de multiplicité k , $P(u)$ est $(m-k)$ fois dérivable (mais pas nécessairement dérivable continue [Pie 91]).

(g) Propriété de la variation décroissante: aucun plan n'a plus de points d'intersections avec la courbe qu'avec le polygone caractéristique associé.

(h) L'adoption d'une séquence nodale sans noeuds intérieurs du type $U = \{0, \dots, 0, 1, \dots, 1\}$ donne une courbe de Bézier rationnelle, tandis que la séquence de coordonnées homogènes identiques ($h_i = \text{constante}$, $0 \leq i \leq n$) donne une courbe B-spline non uniforme. L'adoption des deux conditions à la fois conduit à l'obtention d'une courbe de Bézier. Les NURBS sont donc la généralisation des modèles de Bézier, Bézier rationnelle et B-spline.

(i) L'effet de la variation du poids d'un pôle est analogue à celui vu sur les Bézier rationnelles (cf. §I-3-2-2) à l'exception que la variation, dans le cas des NURBS, est locale.

Piegel [Pie 89a, Pie 89b] a donné une méthode ingénieuse pour mettre l'effet des poids au profit des designers.

(j) Les courbes NURBS peuvent représenter **exactement** toutes les formes coniques usuelles de conception (cercle, ellipse, ...) [Leo 91, Rog 89, Pie 87, Pie 90], les segments de droite et les courbes gauches. Ainsi, les NURBS peuvent constituer le modèle mathématique unique de modélisation et de traitement dans un système de CFAO très complet afin de faciliter la gestion de sa base de données [Leo 91, Gar 87].

I-3-3-4. Algorithmes de calcul

La généralisation de l'algorithme de Cox - de Boor (cf. §I-2-2-6) donne un procédé simple et efficace pour le calcul des points et des dérivées partielles d'une courbe NURBS:

* données: $n, m, P_i, h_i, 0 \leq i \leq n$, et U

* chercher l'indice j tel que: $u \in [u_j, u_{j+1}[$

* initialisation:

FOR $i = j-m$ TO j { $P_i^{[0]} = P_i ; h_i^{[0]} = h_i$ }

* récursion:

FOR $k = 1$ TO m

{ FOR $i = j$ DOWN TO $j-m+k$

{

$$h_i^{[k]} = h_i^{[k-1]} \left(\frac{u - u_i}{u_{i+m+1-k} - u_i} \right) + h_{i-1}^{[k-1]} \left(\frac{u_{i+m+1-k} - u}{u_{i+m+1-k} - u_i} \right) \}$$

$$P_i^{[k]} = P_i^{[k-1]} \frac{h_i^{[k-1]}}{h_i^{[k]}} \left(\frac{u - u_i}{u_{i+m+1-k} - u_i} \right) + P_{i-1}^{[k-1]} \frac{h_{i-1}^{[k-1]}}{h_i^{[k]}} \left(\frac{u_{i+m+1-k} - u}{u_{i+m+1-k} - u_i} \right)$$

}

}

$$P(u) = P_j^{[m]}$$

* Fin

Les expressions des dérivées partielles peuvent être calculées d'une façon analogue à celle de la section §I-2-2-6.

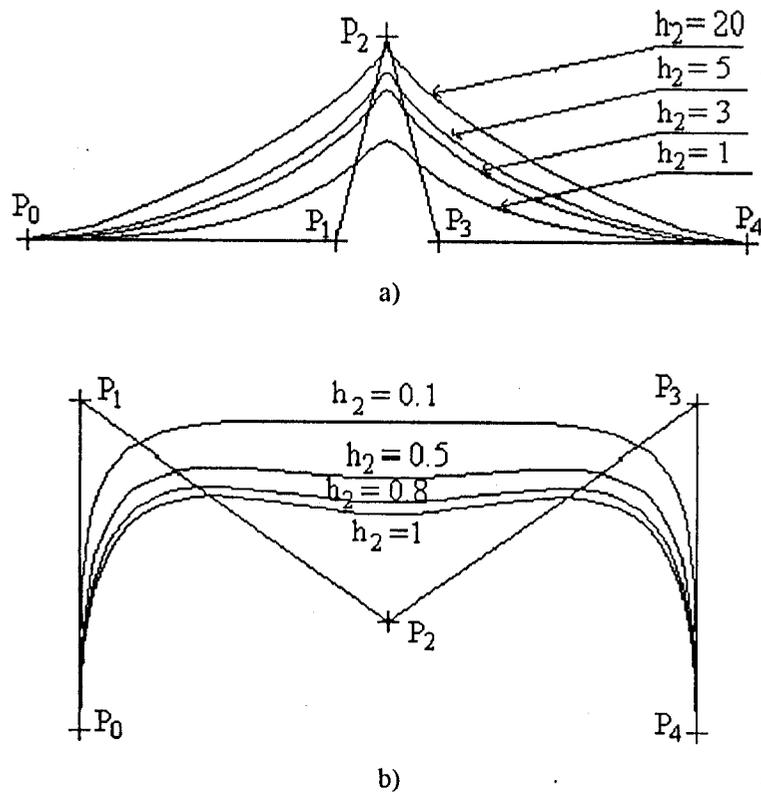


Fig. 1-21 Influence du poids sur la forme d'une courbe NURBS cubique de 5 pôles. Excepté h_2 , le reste des poids est égal à l'unité

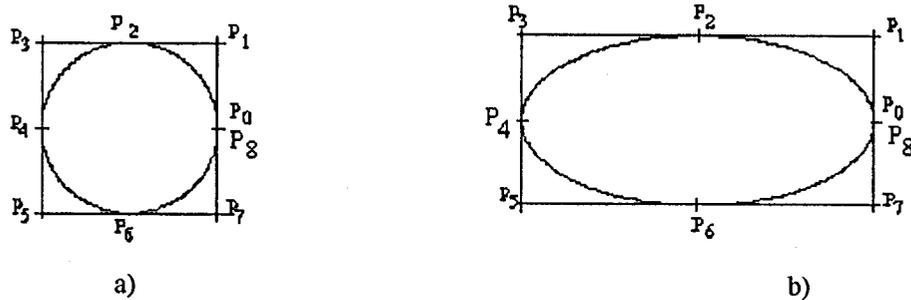


Fig. 1-22 Représentation exacte d'un cercle et d'une ellipse par des courbes NURBS quadratiques

Notons cependant que l'algorithme de calcul des NURBS le plus général connaît jusqu'à présent est l'algorithme dit d' « Oslo » basé sur la théorie des B-splines discrètes et proposé par Riesenfeld et al [Bar 88a, Bar 88b]. L'algorithme d'insertion de noeuds de Boehm [Boe 84, Boe 85] et l'approche symbolique de Choi et al [Cho 91] peuvent servir pour l'évaluation des points courants d'une forme NURBS.

I-4. Modélisation des surfaces gauches

[Leo 91, Fau 87, Bar 88a, Bar 88b, Bezi 86, Rog 89a, Qiu 87, Pie 84, Pie 85, Pie 87, Pie 89b, Pie 91]

Probablement, l'article de J. Fergusson paru en 1964 dans le « Journal of the Association for Computing Machinery » est la première publication en Conception Géométrique Assistée par Ordinateur (CGAO ou CAGD en Anglais) décrivant l'utilisation des surfaces paramétriques pour la conception des ailes et des fuselages des avions. La théorie des espaces paramétriques a été développée en 1953 par I. Schonberg. Le modèle de Fergusson était « trop simple » pour supporter la description des formes complexes conçues et réalisées jusque là uniquement par des segments de droites et des arcs de cercles. En 1963, le constructeur d'automobiles Ford chargea S. Coons du développement d'une représentation mathématique des formes complexes des carrosseries de voitures. C'est ainsi qu'ont vu le jour les carreaux et le modèle de Coons [Coo 87, Bezi 89]. Ce modèle utilise, pour définir un carreau entier, les quatre courbes le délimitant; ces courbes sont interpolées par deux fonctions appelées fonctions de Coons. On obtient ainsi une équation biparamétrique.

Le résultat des travaux de Coons a été publié en 1967 par le MIT et en fait adopté par un grand nombre de constructeurs.

D'autres modèles ont fait leur apparition, mais ils sont beaucoup moins utilisés, comme celui de Mehlum (1969) développé pour utilisation sur les chantiers navals et les surfaces Inaba (1970) utilisant l'espace cartésien et exprimant la cote z en fonction de x et y [Bezi 89].

Actuellement, c'est les surfaces paramétriques de Bézier, B-splines et NURBS qui sont les plus employées en CFAO; elles feront l'objet des paragraphes suivants.

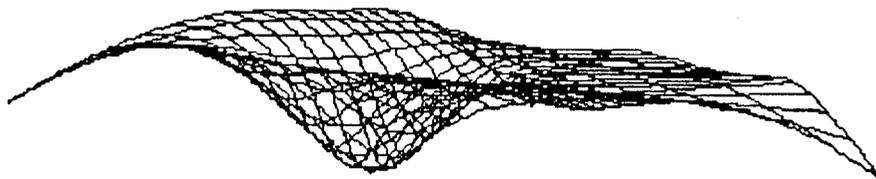


Fig. I-23 Une surface gauche

I-4-1. Surfaces paramétriques - formulation

Comme une courbe paramétrique utilise dans sa définition un paramètre u , une surface utilise deux paramètres : u et v définis sur un domaine rectangulaire de \mathbb{R}^2 de type $[u_{\min}, u_{\max}] * [v_{\min}, v_{\max}]$.

Soit un réseau de points P_{ij} , $i \in \{0, 1, \dots, m\}$, $j \in \{0, 1, \dots, n\}$. Un point $P(u, v)$ sur une surface à pôles, de degré p suivant la direction u et q suivant la direction v , est défini par:

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) P_{ij} \quad (23a)$$

ou encore

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n Q_{i,j}(u, v) P_{ij} \quad (23b)$$

avec

$$Q_{i,j}(u, v) = N_{i,p}(u) N_{j,q}(v) \quad (24)$$

$N_{i,p}(u) N_{j,q}(v)$ est le produit tensoriel des fonctions de base $N_{i,p}(u)$ et $N_{j,q}(v)$ [Bar 88a, Pie 87, Bezi 89, Pie 85]. Ce produit est appelé *fonction de base biparamétrique*.

Les P_{ij} représentent un réseau rectangulaire (au sens topologique) de points de l'espace affine, appelé « *réseau caractéristique* » de la surface (Fig. I-24). Ce réseau comporte $(m+1)(n+1)$ points.

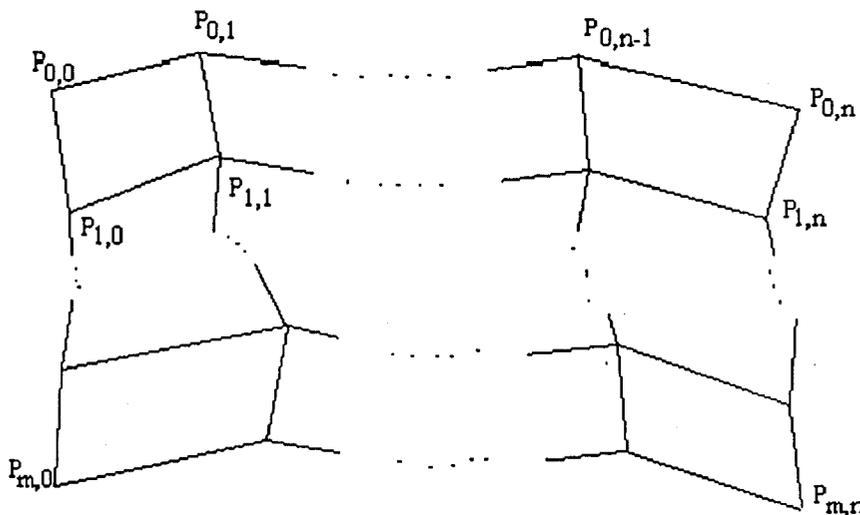


Fig. I-24 Réseau caractéristique de $(m+1)*(n+1)$ pôles

I-4-2. Propriétés analytiques des fonctions de base

Les fonctions de base $Q_{ij}(u,v)$ sont le produit des fonctions de base $N_{i,p}(u)$ et $N_{j,q}(v)$ ($B_{i,p}(u)$ et $B_{j,q}(v)$ pour le modèle de Bézier). Elles sont donc du même type que les fonctions $N_{i,k}(u)$ (cf. §I-2); leurs propriétés permettent de simplifier considérablement la manipulation des surfaces paramétriques à pôles.

Les propriétés analytiques des fonctions de base $Q_{ij}(u,v)$ sont:

(a) Positivité: $Q_{ij}(u,v) \geq 0$ pour toutes les valeurs i, j, p, q, u et v .

(b) Propriété de Cauchy:

$$\sum_{i=0}^m \sum_{j=0}^n Q_{ij}(u,v) = 1, \quad \forall (u,v) \in [u_{\min}, u_{\max}] * [v_{\min}, v_{\max}], \quad (I-25)$$

(c) Support local: $Q_{ij}(u,v) = 0$ si $(u,v) \notin [u_i, u_{i+p+1}] * [v_j, v_{j+q+1}]$

de plus, pour un carreau défini sur $[u_i, u_{i+p+1}] * [v_j, v_{j+q+1}]$, au plus $(p+1)(q+1)$ fonctions de base sont non nulles.

(d) Dérivabilité: pour tout (u,v) non nodal, les dérivées partielles de $Q_{ij}(u,v)$ existent; ces dérivées sont des polynômes bivariables. Si u (ou v) est une valeur nodale de multiplicité k , alors $Q_{ij}(u,v)$ est $p-k$ ($q-k$) continûment dérivable sur u (sur v).

(e) Extremum: si $p > 0$ et $q > 0$, alors $Q_{ij}(u,v)$ possède exactement un seul extremum (c'est un maximum).

I-4-3. Propriétés des surfaces à pôles

Excepté la propriété de la variation décroissante, les propriétés des courbes à pôles (cf. §I-1-2, §I-2-5, §I-3-2-2 et §I-3-3-3) sont extensibles aux surfaces à pôles [Pie 87, Rog 89].

(a) Pour les modèles de Bézier et B-splines non uniformes, la surface passe par les quatre coins du réseau caractéristique. Les dérivées partielles (sauf les dérivées mixtes) en ces points, suivant les directions u et v , coïncident avec les dérivées partielles des courbes frontières en ces points [Pie 87].

(b) Les surfaces de Bézier et B-splines sont définies intrinsèquement grâce à la propriété de Cauchy vérifiée par leurs fonctions de base.

- c) Propriété du réseau convexe enveloppe: si $u \in [u_r, u_{r+1}[$ et $v \in [v_s, v_{s+1}[$, alors la surface $P(u,v)$ est située à l'intérieur du réseau enveloppe convexe formé par les pôles P_{ij} , $i \in \{r-p, r-p+1, \dots, r\}$ et $j \in \{s-q, s-q+1, \dots, s\}$.
- (d) si $p = 1$ et $q = 1$, alors la surface doit coïncider, exactement, avec son réseau caractéristique.
- (e) Pour une surface de Bézier, le déplacement d'un pôle P_{ij} affecte l'ensemble de la surface; tandis que le déplacement d'un pôle d'une surface B-spline ne la modifie que localement; l'effet se limite sur l'intervalle $[u_i, u_{i+p+1}[$ et $[v_j, v_{j+q+1}[$.
- (f) Les fonctions $Q_{ij}(u,v)$ travaillent comme des commutateurs On/Off sur chaque intervalle.
- (g) $P(u,v)$ est $(p-k)$ fois dérivable suivant u et/ou $(q-k)$ dérivable suivant v pour un noeud de multiplicité k .
- (h) Les surfaces de Bézier représentent un cas particulier des surfaces B-splines. Par conséquent, pour qu'une surface B-spline devienne une surface de Bézier, il suffit d'avoir $p=m$ et $q = n$.

I-4-4. Exemples

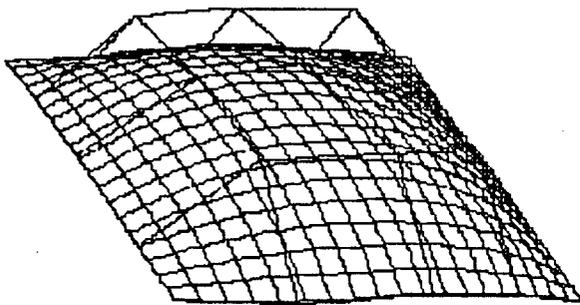


Fig. I-25 Une surface de Bézier biquadratique et son réseau caractéristique de 5*5 pôles

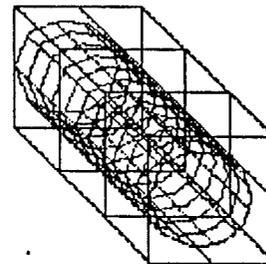


Fig. I-26 Modélisation d'un cylindre par une surface de Bézier de degré 8*3

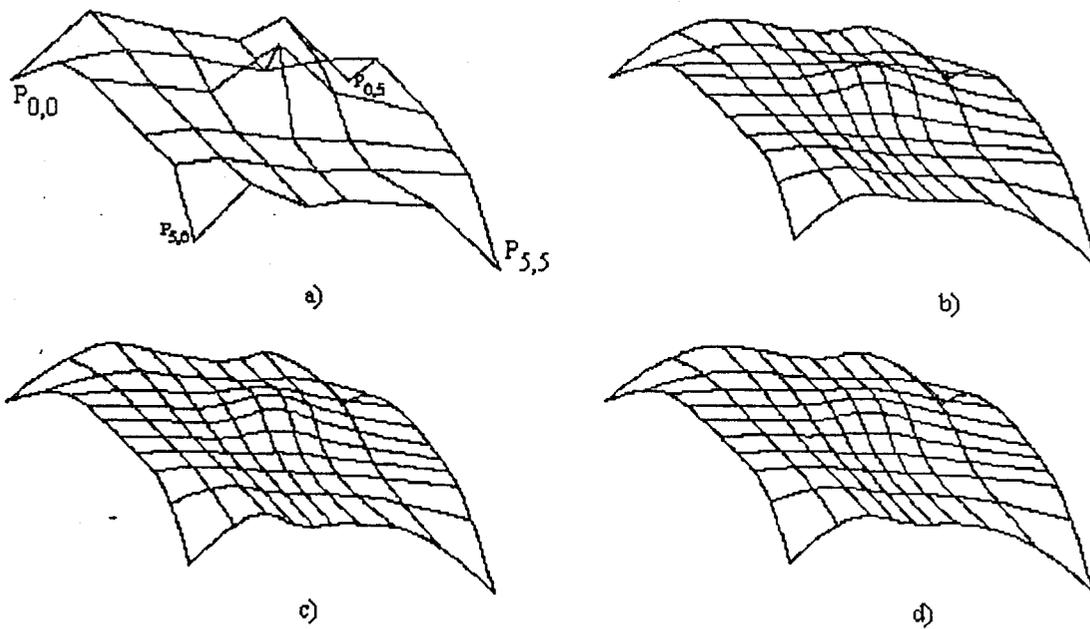


Fig. I-27 Des surfaces B-spline non uniformes de degrés différents créées à partir du même réseau caractéristique: a) le réseau caractéristique b) degré 2×3 c) degré 3×2 d) degré 3×3

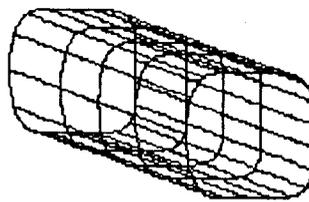


Fig. I-28 Le tuyau de la Fig. I-26 tracé par une surface B-spline non uniforme biquadratique

« Il serai vain de croire que l'on sait faire un problème si l'on s'est contenté, sans rédiger soi-même les calculs et la solution, de lire les résultats encadrés ou de regarder les images »

J. TAILLE

CHAPITRE II

EVIDEMENT DE POCHE PLANES

II. EVIDEMENT DE POCHE PLANES

II-1. Introduction

En gravure et marquage, en fabrication des outillages et des moules et en usinage des surfaces gauches par la technique des plans de niveaux [Lee 92], la tâche principale consiste à enlever la matière contenue à l'intérieur d'un contour fermé appelé poche (Fig. II-1). Si à l'intérieur de ce contour existent des zones de matière à ne pas enlever, le profil sera considéré comme une poche avec îlot(s) (Fig. II-1, II-3b et II-3c). Cette opération d'enlèvement de matière s'appelle évidement de poche (ou vidage ou encore balayage de poche).

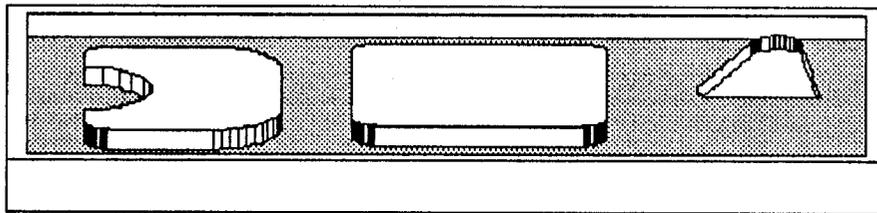


Fig. II-1 Poche avec trois îlots dont celui de droite est aussi une poche à évider

Persson [Per 78] a publié un algorithme de balayage de poches *sans îlots* et dont le contour est constitué uniquement d'éléments géométriques simples, à savoir les arcs de cercles et les segments de droite. La méthode comporte deux étapes:

- Subdivision de la surface de la poche en plusieurs sous surfaces indépendantes. Les frontières séparant ces sous surfaces sont les lieux des bissectrices des éléments du contour de base.
- Génération des chemins d'outil pour chaque sous surface indépendamment des autres.

Outre l'incapacité de traiter les formes libres, la méthode de Persson nécessite plusieurs calculs successifs des trajectoires décalées ainsi que des procédures de raccordements délicates.

Le principe de l'approche de [Per 78] a été repris et développé en [Hel 94] pour traiter les poches avec un nombre quelconques d'îlots mais sans toute fois le généraliser aux éléments géométriques gauches. Actuellement, plusieurs MOCN possèdent des fonctions incorporées de vidage de poches de contours constitués de segments de droite et/ou d'arcs de cercles (fonction G45 de NUM par exemple [Num 88]).

Kimmel et Bruckstein [Kimm 93] ont proposé une méthode de balayage de poches *sans îlots*; elle est basée sur la théorie de propagation du front d'une flamme en combustion ou d'une

onde de choc en Mécanique des Fluides. La méthode aboutit à l'obtention d'un système d'équations différentielles que les auteurs suggèrent de résoudre par l'algorithme d'Osher et Sethian. Cette méthode comporte l'inconvénient de passer par le calcul des formes implicites des contours de la poche; ce qui est considéré comme difficile et parfois impossible pour des formes de type Béziérs ou B-splines [Mic 92, Sed 84].

Suh et Lee [Suh 90] ont traité le cas de vidage de poches sans îlots ou uniquement avec *un seul îlot*. Les trajectoires planes (2D) obtenues sont utilisées pour former une surface extrudée. L'intersection de cette surface avec la surface gauche lieu de la poche à usiner donne l'ensemble des positions du centre de l'outil d'une fraiseuse à commande numérique de trois axes. L'approche de Suh et Lee emploie, pour générer les trajectoires planes, une procédure de décalages successifs des contours pour obtenir une trajectoire (de l'outil) en spirale. L'ensemble des traitements, relatifs au calcul des courbes parallèles pour les NURBS, proposés par Tiller et Hanson [Til 84] ont été repris.

Ces méthodes sont incomplètes puisqu'elles n'autorisent pas de traiter:

- (1) les poches à formes libres ou
- (2) les poches à plusieurs îlots.

Dans les paragraphes suivants nous exposerons une méthode générale comblant ces lacunes et qui permettra de balayer des poches à contours quelconques (segments de droites, arcs de cercles ou courbes paramétriques) et avec ou sans îlots.

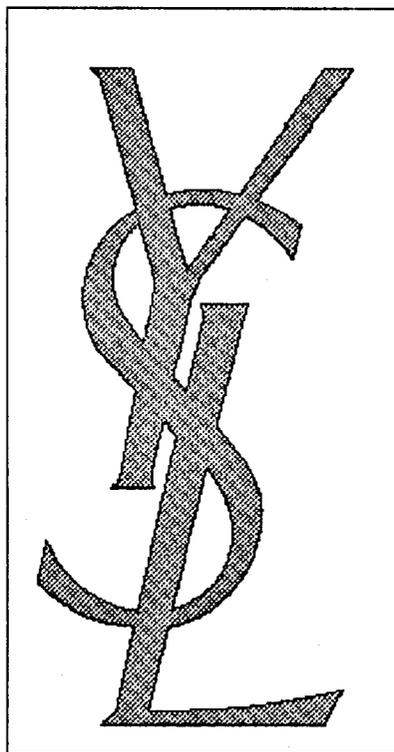
Dans ce chapitre, nous exposerons la méthode de définition d'une poche (profil extérieur, profil(s) intérieur(s) et leurs orientations), le calcul des courbes offset pour les opérations de décalage, la détection et la localisation des singularités dans les chemins d'outil, le traitement de ces singularités ainsi que les tests d'arrêt du balayage. Ces problèmes font appel, pour leur résolution, à des techniques d'approximation des courbes parallèles, d'intersection et d'auto-intersection des courbes gauches, ainsi qu'au traitement de la dégénérescence des contours offset.

II-2. Définition - poche et îlot -

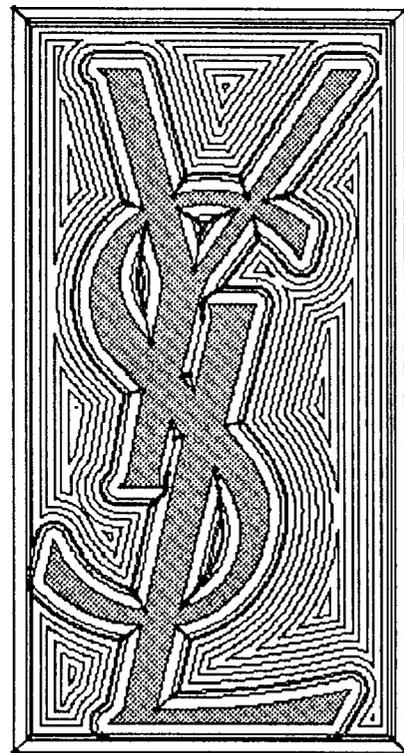
Une poche plane (2D) est définie par l'extrusion d'un contour extérieur et de n ($n \geq 0$) contours intérieurs appelés îlots (Fig. II-1 et II-3). Un contour extérieur ou intérieur est constitué généralement de plusieurs éléments géométriques (segments, arcs, courbes...) orientés d'une manière précise.

L'évidement des poches par fraisage est avant tout une opération d'ébauche [Piq 89]; par conséquent, il est nécessaire, dans plusieurs cas, de faire appel à un autre procédé d'enlèvement de matière afin d'obtenir la pièce finale. Ce que nous allons traiter dans toutes les sections suivantes concernera au premier lieu les phases d'ébauche et de semi - finition réalisées sur des fraiseuses 2 ½ axes au minimum (pour les poches planes) en utilisant une fraise cylindrique à deux tailles et à bout plat.

Plusieurs modes de balayage peuvent être employés. Le mode « zigzag » (Fig. II-4) assure un balayage suivant des passes parallèles à une direction donnée conduisant à une alternance travail en opposition et travail en concordance. Dans le mode « one way » (Fig. II-5), l'outil est remonté à l'arrivée sur le profil, puis ramené à la verticale du point de plongée; étant donné que la plongée s'effectue en pleine matière sur le profil opposé, on sera obligé d'employer une fraise à coupe centrale (problème de durée de vie et de condition de coupe [Piq 89, Bezi 70]). Le temps de déplacement à vide est trop important, ce qui constitue un handicap du mode de balayage « one way ».



a)



b)

Fig. II-2 Le logo d'Yves Saint Laurent (a) et les chemins d'outil (b) de sa gravure réalisés par le logiciel

TYPE3

La méthode qui est couramment employée maintenant est le mode dit « évidement par décalages successifs » dans le quel la poche est vidée par des passes parallèles au profil de définition (Fig. II-6). C'est le mode de balayage que nous avons retenu dans notre étude.

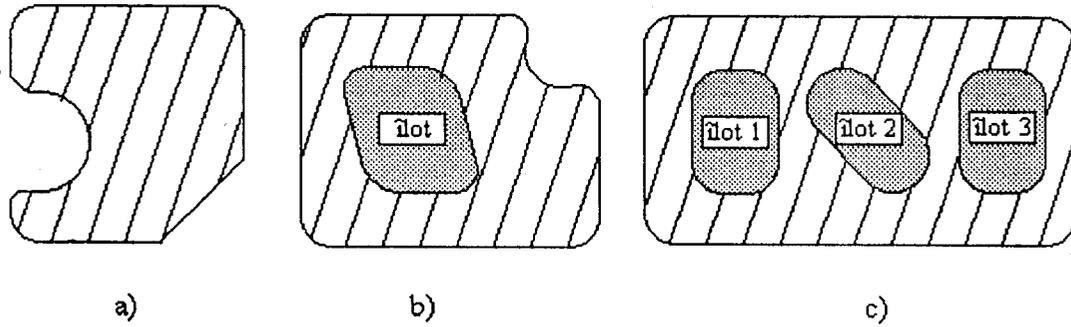


Fig. II-3 a) Poche sans îlots b) Poche avec un îlot c) Poche avec plusieurs îlots
les parties hachurées sont les parties à usiner

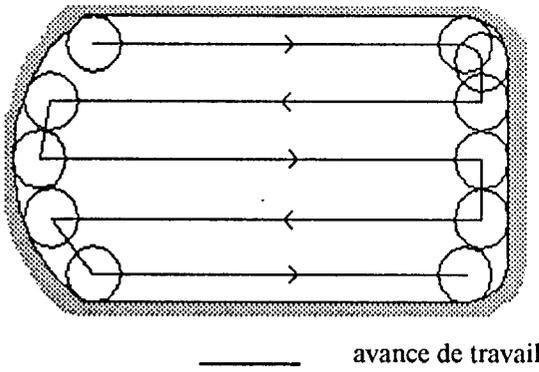


Fig. II-4 Vidage d'une poche en « zigzag »

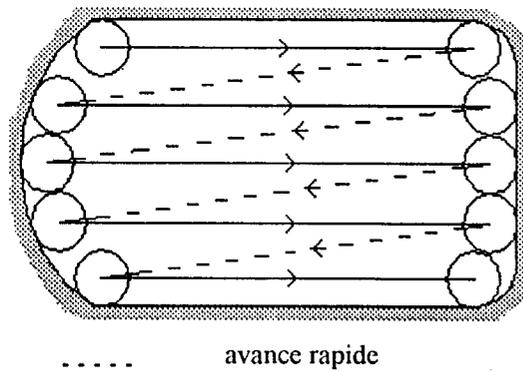


Fig. II-5 Vidage d'une poche en « one way »

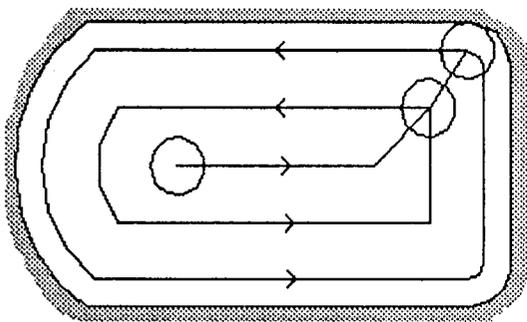


Fig. II-6 Méthode des décalages successifs

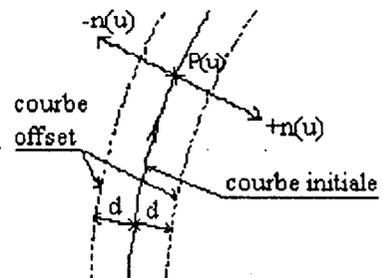


Fig. II-7 Les courbes offset

II-3. Orientation des profils

Soit une courbe paramétrique $(C) = P(u)$ contenue dans le plan (O, x, y, z) . L'expression d'un point courant de la courbe $(C') = P'(u)$ parallèle à (C) et située à une distance d est

$$P'(u) = P(u) + d \cdot n \quad (\text{II-1})$$

où n est le vecteur normal unitaire de la courbe (C) au point $P(u)$.

Le choix du côté dans lequel se situera la nouvelle courbe (la courbe parallèle) (Fig. II-7) est réalisé soit par le choix du signe de l'offset d , soit par l'orientation de la courbe (ou de son polygone caractéristique pour une courbe à pôles). Ils existent trois méthodes permettant de déterminer le sens d'orientation d'une forme géométrique plane fermée et sans boucles [Fil 90]. Ces méthodes permettent également de savoir si un point donné est à l'intérieur ou à l'extérieur de cette forme. Cette information (sens d'orientation) est utile en usinage car elle nous permet de connaître le sens de décalage des courbes parallèles et donc de préciser le côté à usiner dans la pièce. Ces méthodes sont:

II-3-1. Théorème de Green

Si le signe de l'aire délimitée à l'intérieur d'un contour est positif, alors celui-ci est orienté dans le sens de rotation des aiguilles d'une montre (sens horaire); en revanche, si le signe est négatif alors le contour est orienté dans le sens inverse de la rotation des aiguilles d'une montre (sens trigonométrique).

II-3-2. Théorème des tangentes tournantes

Si la somme totale des angles de variation des tangentes dans chacune des courbes du contour et la variation des tangentes aux points de jonction entre ces courbes est égale à 2π alors le contour est orienté dans le sens trigonométrique sinon (donc la somme est égale à -2π) il est orienté dans le sens horaire.

II-3-3. Théorème de Jordan

Une demi droite R démarrant d'un point P et dirigé vers l'infini doit avoir un nombre impair d'intersections avec les courbes composant le contour si ce point P est situé à l'intérieur du contour (Fig. II-10); et inversement, si le nombre de points d'intersection est pair, alors le point P est à l'extérieur du contour. Cette méthode est très souvent utilisée en infographie [Per 90, Mart 89]. Dans notre cas et du fait qu'on ne cherche pas à savoir l'appartenance ou non d'un point à l'intérieur d'un contour, ce théorème devra s'appliquer de la façon suivante: Prenons un point P sur une courbe du profil fermé (où P n'est pas un point de jonction) et « traçons » une demi droite R d'origine le point P et orientée de telle sorte qu'elle forme un angle positif avec le vecteur tangent de la courbe au point P (Fig. II-11). Si le nombre d'intersections de R avec le contour est impair alors ce dernier est orienté dans le sens trigonométrique (Fig. II-11a); sinon il est dans le sens horaire (Fig. II-11b). Les particularités et les difficultés de l'application de cette méthode sont traitées dans la référence [Fil 90]; les méthodes rapides de calcul des intersections d'un segment de droite et d'un polygone sont discutées dans la référence [Mil 93].

II-3-4. Comparaison

L'implémentation du théorème de Jordan tel qu'il est, est relativement délicate du fait des nombreux cas particuliers d'intersections à prendre en considération. Il risque, par conséquent, d'aboutir à des codes informatiques peu fiables. Néanmoins on peut, pour des contours très particuliers, réduire le problème à des simples opérations de comparaison de valeurs numériques [Mil 93, Yan 84].

L'utilisation du théorème des tangentes tournantes est peu pratique car il fait appel à l'évaluation des fonctions trigonométriques inverses; cette opération est très coûteuse en temps machine.

Le théorème de Green offre plus de robustesse et moins de calcul si on fait remplacer les courbes gauches du profil par leurs polygones caractéristiques. Cette opération est admise du moment que le but d'utilisation du théorème n'est plus de calculer la valeur exacte d'une surface mais uniquement de connaître son signe.

Dans notre travail c'est la méthode de Green que nous avons retenue pour l'orientation des contours fermés et pour les tests d'arrêt des décalages (cf. §II-7-6). La méthode de Jordan a été choisie pour utilisation dans le traitement des interférences (cf. §II-7-5).

II-3-5. Convention d'orientation

Par conséquent, et pour un offset $d > 0$, l'orientation du contour extérieur de la poche sera dans le sens de rotation des aiguilles d'une montre tandis que l'orientation des contours des îlots doit se faire dans le sens inverse (sens trigonométrique). De ce fait le décalage des contours extérieurs doit être vers l'intérieur, tandis que le décalage des contours des îlots doit être vers l'extérieur (cf. les figures II-8 et II-9).

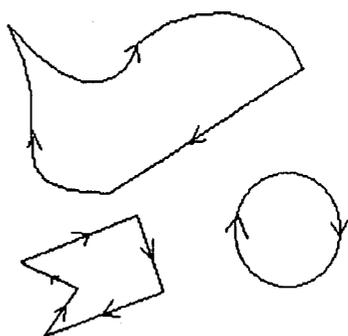


Fig. II-8 Contours orientés dans le sens horaire

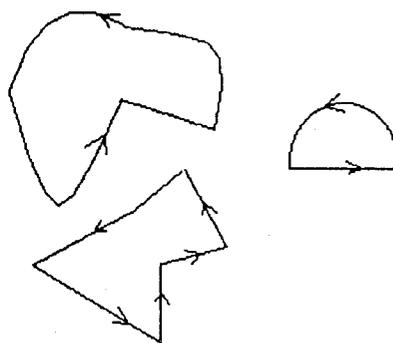


Fig. II-9 Contours orientés dans le sens trigonométrique

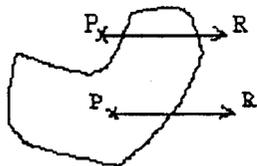


Fig. II-10 Théorème de Jordan
(test d'appartenance)

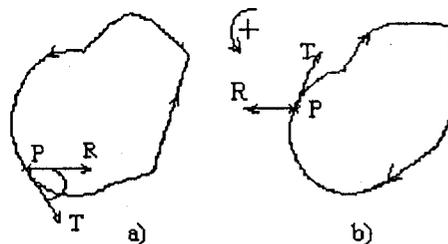
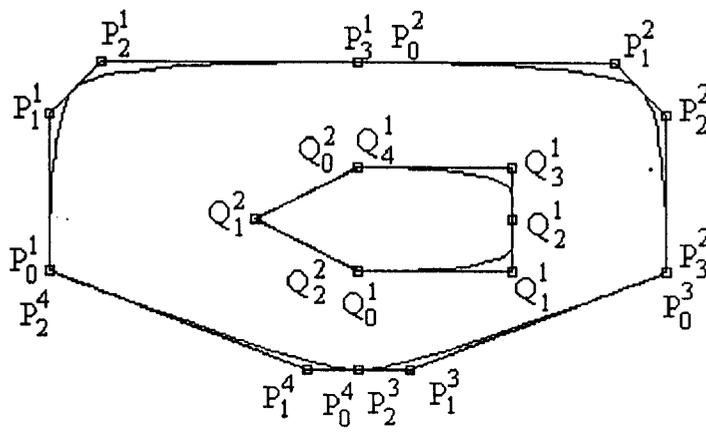
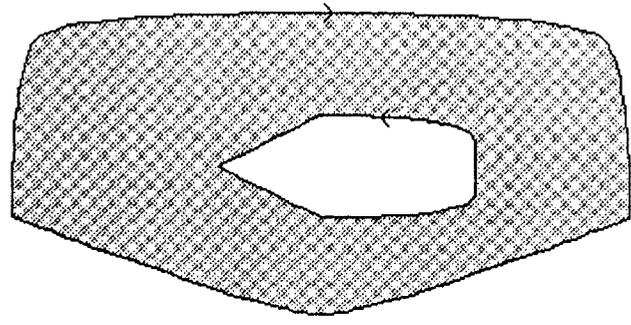


Fig. II-11 Application du théorème de Jordan à l'orientation des contours



a)



b)

Fig. II-12 a) Orientation des courbes à pôles sur la poche et sur l'îlot
 b) Cette orientation permet d'usiner uniquement la surface colorée

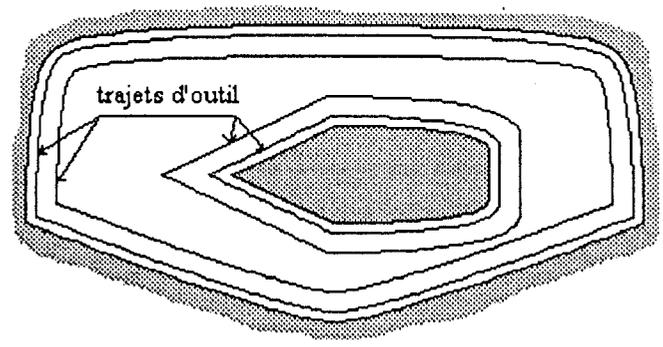


Fig. II-13 Sens de décalage des contours de la poche et de l'îlot

II-4. Calcul des courbes offset planes

Le calcul des courbes et des surfaces parallèles (ou courbes et surfaces offset) trouve des applications dans plusieurs domaines comme la génération des trajectoires d'outil pour les Machines - Outil à Commande Numérique (MOCN), la programmation des trajectoires des poignets de robots, la détermination de l'épaisseur d'un objet en conception, la propagation des ondes de choc, la couture, l'industrie automobile dans la conception des outillages d'estampage, etc. Malgré cette importance l'étude analytique des propriétés des formes offset n'a été entreprise que récemment [Far 90a, Far 90b].

Soit une courbe plane $(C') = P'(u)$ parallèle à la courbe plane $(C) = P(u)$ à une distance d constante (Fig. II-7). L'expression de $P'(u)$ est donnée par:

$$P'(u) = P(u) + d \frac{z \wedge \frac{d}{du} P(u)}{\left\| z \wedge \frac{d}{du} P(u) \right\|} \quad (\text{II-2})$$

où

$$\frac{z \wedge \frac{d}{du} P(u)}{\left\| z \wedge \frac{d}{du} P(u) \right\|} = n \quad (\text{II-3})$$

représente le vecteur normal unitaire de (C) au point $P(u)$.

Comme on peut le constater dans l'expression (II-2), la courbe offset exacte d'une courbe polynomiale ne peut pas être, généralement, définie par une autre courbe exacte de même nature [Til 84, Far 85, Far 86, Far 90a, Far 90b]. Ainsi, une courbe offset exacte d'une courbe B-spline n'est plus une courbe B-spline [Pha 92]. Or l'exigence d'usage d'une procédure unique de décalage des profils ainsi que le souci de sauvegarder l'homogénéité de la base de données nous oblige à trouver une solution à ce problème.

La solution adoptée souvent consiste à approximer les formes offset; les approches utilisées sont classées en trois catégories: les méthodes de polygonisation, les méthodes de discrétisation - conversion et les méthodes d'approximation par une courbe de même type.

II-4-1. Méthodes de polygonisation

On calcule par l'expression (II-2) un ensemble de points sur la courbe parallèle. Ces points sont répartis sur la courbe suivant un critère de conception et/ou de fabrication (tolérance, erreur de flèche...) défini par l'utilisateur. Les points ainsi calculés sont ensuite reliés pour former un polygone. Le procédé est itératif et ne permet pas de connaître le nombre de points (donc de segments) nécessaires qu'à la fin du processus.

Dans le cas de courbes à pôles l'évaluation de l'expression (II-2) est relativement rapide du fait que les valeurs des dérivées partielles nécessaires pour le calcul de la normale n peuvent être obtenues comme un « sous - produit » pendant le calcul du point $P(u)$ (cf. §I-2-1-3 et §I-2-2-6).

En pratique, cette méthode ne s'utilise en CFAO que dans les problèmes de contournage (intérieur ou extérieur) d'un profil et ne convient guère aux opérations répétées de décalage qui sont fondamentales en balayage des poches. Cette inadaptabilité est due au fait que la manipulation doit se faire, dans ce cas, sur une quantité trop grande de points (au lieu de courbes) ce qui conduit, d'une part, à la nécessité d'un espace de stockage important et de l'autre part à un manque de souplesse dans le traitement.

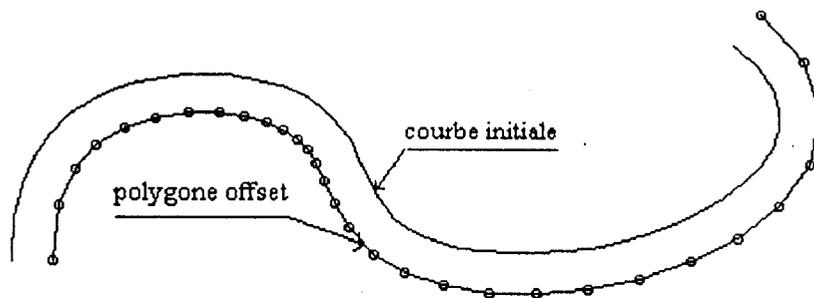


Fig. II-14 Méthode de polygonisation

II-4-2. Méthodes de discrétisation - conversion

Dans ces méthodes on discrétise la courbe mère en un ensemble de segments de droite en respectant un critère d'erreur de flèche [Qiu 87, Fau 87] et/ou un ordre de continuité géométrique [Mee 92]; ensuite on calcule leurs segments parallèles correspondants. En dernière étape on procède à l'approximation des points constituant les extrémités de ces

segments par un ensemble plus réduit de segments de droite et éventuellement d'arcs de cercle [Cra 85, Mee 92, Rob 85, Pie 86, Mor 91].

Même si cette méthode conduit à une diminution importante de l'espace de stockage [Rob 85] par rapport à la première méthode, néanmoins l'hétérogénéité de la base de données reste un problème persistant.

II-4-3. Méthodes d'approximation par des courbes de même type

Pour éviter les problèmes cités ci-dessus, l'utilisation des courbes offset de même type que les courbes mères semble inévitable. L'offset d'un polygone sera un polygone, l'offset d'un arc de cercle sera un autre arc de cercle et l'offset d'une courbe doit être une courbe de même type (même modèle mathématique). Ainsi l'homogénéité de la base de données du logiciel sera pleinement conservée. Cette approche est celle qu'on trouve dans la plupart des publications traitant l'approximation des offset [Pha 92, Hos 85, Pha 88, Suh 90, Til 84, Hos 88a, K1a 83], etc. Cette approximation se fait, généralement, de trois manières: interpolation, lissage par la méthode des moindres carrées et approximation par le polygone caractéristique parallèle. Les trois méthodes permettent de trouver, d'une manière ou d'une autre, le polygone caractéristique de la courbe offset et son vecteur nodal (dans le cas des courbes B-splines ou NURBS) les plus adaptés.

Les deux premières méthodes sont très utilisées en conception où l'utilisateur introduit un certain nombre de points et le système de CAO se charge du calcul du polygone caractéristique permettant à la courbe de passer par tous ou quelques uns de ces points [Rog 89b, Fog 84, For 90] en respectant certains critères géométriques et esthétiques [Spa 90].

II-4-3-1. Interpolation

Soit un ensemble de $\{p+1\}$ points P^*_i , $i \in \{0, 1, \dots, p\}$, calculés par l'expression (II-2); chaque point représente une position exacte sur la courbe offset. La courbe offset $P'(u)$ qu'on cherche à construire doit passer par ces points (Fig. II-15).

On choisit un degré m , une séquence nodale U' et un paramétrage (cf. §II-4-4) qui affecte à chaque point P^*_i un point $P'(\mu_i)$ tel que: $P'(\mu_i) = P^*_i$, $i \in \{0, 1, \dots, p\}$

on obtient le système linéaire suivant:

$$\sum_{j=k-m}^k B_{j,m}(\mu_i) P'_j = P^*_i, \quad i \in \{0, 1, \dots, p\} \quad (\text{II-4})$$

pour $u_k \leq \mu_i \leq u_{k+1}$

où : P'_j sont les pôles du polygone caractéristique de la courbe offset à construire,

$B_{j,m}(\mu_i)$ sont des fonctions de base d'interpolation de degré m ; elles sont soit des fonctions de Bernstein pour les courbes de Bézier, soit des fonctions B-splines pour les courbes B-splines.

L'expression (II-4) peut s'exprimer sous la forme matricielle

$$[B] \begin{bmatrix} P'_0 \\ P'_1 \\ \vdots \\ P'_p \end{bmatrix} = \begin{bmatrix} P^*_0 \\ P^*_1 \\ \vdots \\ P^*_p \end{bmatrix} \quad (\text{II-5})$$

Le problème devient, donc, un problème de résolution d'un système d'équations linéaires. La matrice $[B]$ du système est du type bande avec une largeur de bande, généralement, égale à m . Ainsi, on pourra employer des algorithmes exploitant au mieux cette particularité. La méthode de Gauss - Seidel présente un bon outil de résolution de ce système [Pha 88]. Elle est souvent préférable sur la méthode de Jacobi par sa rapidité de convergence [Ral 78] et sur la méthode de Gauss - Jordan car cette dernière fait apparaître des éléments non nuls, dans les étapes intermédiaires de résolution, à des places où il y avait des zéros dans la matrice initiale (donc perte de l'avantage des matrices bandes) [Nou 93].

L'examen de l'expression (II-5), dans le cas de construction par interpolation des courbes B-splines *uniformes*, montre qu'elle donne $(p+1)$ équations alors que $(p+m)$ sommets P'_j apparaissent dans le terme de gauche. L'obtention d'une solution unique au problème nécessite d'ajouter $(m-1)$ équations indépendantes [Leo 91, Bar 88a].

Le théorème de Schoenberg et Withney [Leo 91], qui exprime les conditions nécessaires pour que la matrice du système linéaire de (II-5) soit inversible quelque soit l'ensemble des points P^*_i et le degré m , exige d'adopter un paramétrage qui affecte à chaque point P^*_i un point de la courbe $P'(u)$ tel que $u = i$. Le non - respect de ce paramétrage peut produire des matrices singulières [Leo 91], toutefois son adoption risque de donner des courbes de formes indésirables comme il a été constaté par [Lee 89]. Il faut noter, cependant, que ces deux

problèmes sont spécifiques au modèle B-spline uniforme qui est, actuellement, rarement utilisé dans les modeleurs géométriques de CFAO. La question du paramétrage sera l'objet de discussion dans §II-4-4.

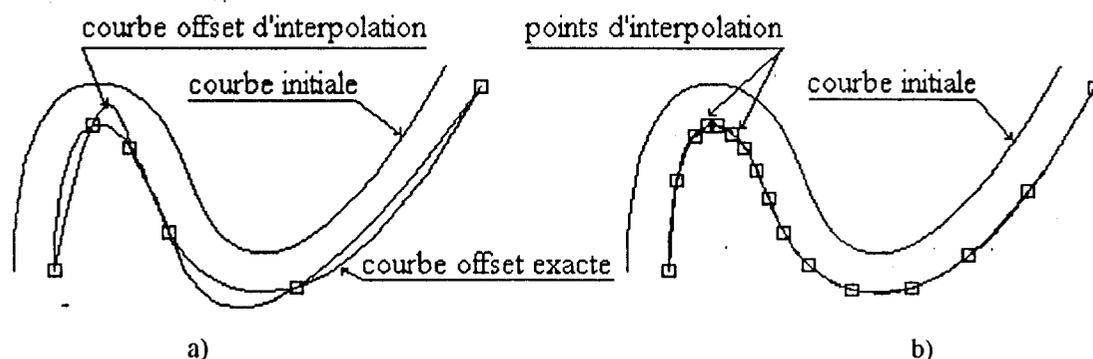


Fig. II-15 Amélioration de l'approximation de la courbe offset d'une courbe cubique
 a) $n' = n = 5$; b) $n' = 3n = 15$

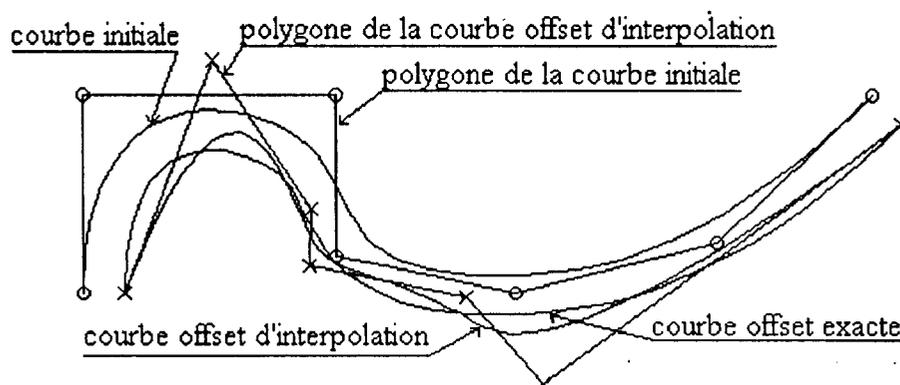


Fig. II-16 Oscillations sur la courbe offset d'interpolation

La figure II-15a montre l'approximation de la courbe offset obtenue en utilisant une autre courbe de même nombre de pôles, de même degré et de même suite nodale que la courbe initiale. Une augmentation considérable du nombre de points d'interpolation a permis d'améliorer considérablement la trajectoire comme on peut le constater sur la figure II-15b.

La figure II-16 montre un autre exemple d'une courbe offset ainsi que son polygone caractéristique obtenus par interpolation. Le polygone et la courbe interpolés comportent des oscillations. Ils seront peu pratiques dans des manipulations telles que la recherche des intersections ou des interférences.

II-4-3-2. Construction par lissage (méthode des moindres carrées)

Soit un ensemble de points P_i^* , $i \in \{0, 1, \dots, p\}$, qu'on désire lisser par une courbe B-spline non uniforme $P'(u)$, de degré m , de séquence nodale U' donnée, et de polygone caractéristique P'_i , $i \in \{0, 1, \dots, q\}$, en utilisant $(q+1)$ points parmi les $(p+1)$ points P_i^* donnés (si $q = p$ on revient alors au cas de l'interpolation). On affecte à chaque point P_i^* un paramètre μ_i tel que $\mu_{i+1} \geq \mu_i$ et $u_{\min} \leq \mu_i \leq u_{\max}$.

Les écarts entre la courbe à construire $P'(u)$ définie par:

$$P'(u) = \sum_{j=0}^{q+m+1} B_{j,m}(u) P'_j \quad (\text{II-6})$$

et les points P_i^* sont mesurés par $\varepsilon_i = P'(\mu_i) - P_i^*$, $i \in \{0, 1, \dots, p\}$;

ou encore, en notant 1 et 2 les composantes x et y des vecteurs $P'(\mu_i)$ et P_i^* :

$$\varepsilon_{ij} = P'_j(\mu_i) - P_{ij}^*, \quad i \in \{0, 1, \dots, p\}; \quad j \in \{1, 2\}$$

La minimisation de la somme des carrés de ces écarts i.e.

$$\min_{P'_{ij}} G_j = \min_{P'_{ij}} \sum_{i=0}^p \varepsilon_{ij}^2, \quad j \in \{1, 2\}, \quad (\text{II-7})$$

engendre le système linéaire [Leo 91] suivant:

$$\frac{\partial G_j}{\partial P'_{ij}} = 0, \quad i \in \{0, 1, \dots, q+m-1\}, \quad j \in \{1, 2\}, \quad (\text{II-8})$$

La résolution du système d'équations obtenu par l'expression (II-8) peut être exprimée sous la forme matricielle [Leo 91, Rog 89a]:

$$\begin{bmatrix} P'_0 \\ P'_1 \\ \vdots \\ P'_{q+m-1} \end{bmatrix} = ([B]^T [B])^{-1} [B]^T \begin{bmatrix} P^*_0 \\ P^*_1 \\ \vdots \\ P^*_p \end{bmatrix} \quad (\text{II-9a})$$

$$\text{avec } [B] = \begin{bmatrix} B_{0,m}(\mu_0) & B_{1,m}(\mu_0) & \dots & 0 \\ B_{0,m}(\mu_1) & B_{1,m}(\mu_1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B_{q+m-1,m}(\mu_p) \end{bmatrix} \quad (\text{II-9b})$$

L'évaluation des éléments B_{ij} se fait par le calcul des fonctions de base du modèle choisi.

Le vecteur $(P'_0 P'_1 \dots P'_{q+m-1})^T$ représente les sommets du polygone caractéristique de la courbe $P'(u)$ approximant les points P^*_i donnés.

II-4-3-3. Méthode du polygone caractéristique parallèle (méthode de Tiller et Hanson)

Cette méthode a été proposée par Tiller et Hanson [Til 84]. C'est une méthode itérative d'approximation des courbes offset planes de type Bézier, Bézier rationnel, B-spline non uniforme et NURBS. La démarche d'approximation s'effectue selon les étapes suivantes:

- 1- La première itération consiste à construire un polygone $P'^{[k=1]}_i$ parallèle au polygone initial P_i , $i \in \{0, 1, \dots, n\}$, à la distance de décalage d pour obtenir une courbe offset $P'^{[k=1]}(u)$;
- 2- On compare alors la courbe $P'^{[k]}(u)$ à la courbe parallèle théorique en un certain nombre de points (généralement au milieu des intervalles nodaux [Pha 88]). Notons que k représente le nombre d'itérations effectuées.
- 3- Si l'écart mesuré dépasse la tolérance imposée au point $P(u^*)$ on affine la courbe initiale $P(u)$ en ce point soit par insertion de noeuds dans le vecteur nodal de la courbe $P(u)$ (cas des B-splines et des NURBS) [Boe 80, Boe 85] soit par insertion de sommets supplémentaires dans le polygone P_i de la courbe initiale (cas des Bézier) [Leo 91, Fau 87, Qiu 87, Pen 84, Bezi 86, Bar 88a].
- 4- On construit alors le nouveau polygone $P'^{[k+1]}_i$ décalé de d et on revient à l'étape 2.
- 5- On arrête le processus dès qu'on obtient une courbe parallèle située à l'intérieur de l'intervalle de tolérance imposé.

La figure II-17 montre un exemple d'une courbe B-spline de degré 3 et de 6 pôles et sa courbe offset approximée avec une précision relative de 1% par une courbe du même type et de même degré mais de 30 pôles en 24 itérations. La figure II-18 montre une courbe B-spline de degré 5 et de 9 pôles; sa courbe offset de même degré a été obtenue en 4 itérations pour une précision relative de 10%.

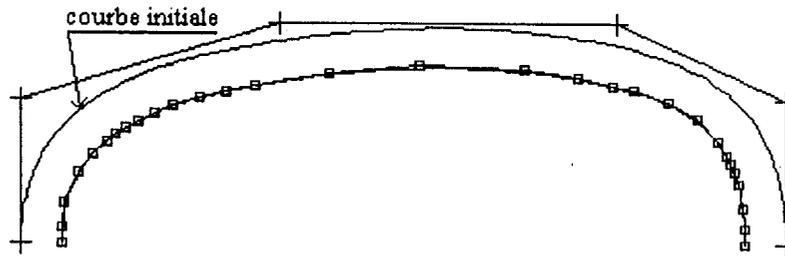


Fig. II-17 Méthode de Tiller et Hanson ($n = 5, m = 3, \text{précision} = 1\%$)

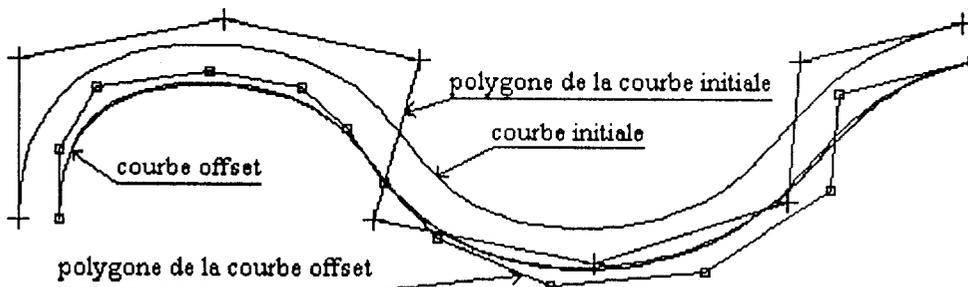


Fig. II-18 Méthode de Tiller et Hanson ($n = 8, m = 5, \text{précision} = 10\%$)

II-4-3-4. Méthode de Coquillart

Cette méthode proposée par S. Coquillart [Coq 87] est une méthode directe (sans itérations). Le déplacement imposé aux pôles du polygone caractéristique de la courbe mère est régi par l'équation:

$$P_i^* = P_i + d (1 + \|P_i - P(\mu_i)\| / \rho(\mu_i)) n(\mu_i), \quad i \in \{0, 1, \dots, n\}, \quad (\text{II-10})$$

les paramètres μ_i sont calculés par:

$$\mu_i = (u_{i+1} + \dots + u_{i+m}), i \in \{0, 1, \dots, n\}, \quad (\text{II-11})$$

où:

- * $P_i, i \in \{0, 1, \dots, n\}$, sont les pôles du polygone caractéristique de la courbe mère $P(u)$;
- * u_i sont les éléments de la suite nodale de la courbe mère $P(u)$ au point $P(\mu_i)$;
- * $\rho(\mu_i)$ est le rayon de courbure de la courbe mère $P(u)$;
- * $n(\mu_i)$ est le vecteur normal unitaire de la courbe $P(u)$ au point $P(\mu_i)$.

Cette technique convient pour la construction des courbes offset des courbes spatiales et donne, une fois appliquée aux NURBS, une solution exacte pour les arcs de cercle. Par opposition aux méthodes itératives, la construction réalisée selon cette méthode pour une courbe quelconque ne permet pas d'approcher le résultat théorique avec une précision ε arbitrairement fixée.

II-4-3-5. Méthode de Hoschek

Soit $P(u)$ une courbe de Bézier de degré n et de polygone caractéristique associé $P_i, i \in \{0, 1, \dots, n\}$. Et soit $P'(u)$ la courbe parallèle à $P(u)$ à une distance d suivant la normale $n(u)$ telle que

$$P'(u) = P(u) + d n(u) \quad (\text{II-12})$$

Pour approximer, avec une précision ε donnée, la courbes parallèle $P'(u)$ par une ou plusieurs courbes $P^*(u)$ du même type, J. Hoschek [Hos 88a, Hos 88b, Hos 88c] a proposé d'utiliser une série de *courbes de Bézier* de degré m reflétant l'ordre de la continuité géométrique imposé aux points de jonction entre ces courbes. On prend $m = 3$ pour une continuité géométrique G^1 et $m = 5$ pour une continuité G^2 .

Pour $m = 3$, l'étape suivante consiste à déterminer quatre points P^*_0, P^*_1, P^*_2 et P^*_3 constituant le polygone caractéristique de $P^*(u)$. On peut écrire immédiatement

$$P^*_0 = P_0 + d n(0) \quad \text{et} \quad P^*_3 = P_n + d n(1) \quad (\text{II-13})$$

Les tangentes aux extrémités de la courbe offset et de la courbe initiale sont liées par les relations suivantes:

$$P^*_1 = P^*_0 + \lambda_1(P_1 - P_0) \quad \text{et} \quad P^*_2 = P^*_3 + \lambda_2(P_n - P_{n-1}) \quad (\text{II-14})$$

où λ_1 et λ_2 sont des inconnues (de valeurs réelles).

Choisissons $(q+1)$ points Q_i situés sur la courbe parallèle exacte $P'(u)$. L'erreur de position δ_i entre un point Q_i et son équivalent $P'(u_i)$ sera:

$$\delta_i = Q_i - \sum_{j=0}^3 B_{j,3}(u_i)P_j^* \quad 0 \leq u_i \leq 1 \quad (\text{II-15})$$

En remplaçant dans (II-15) P^*_1 et P^*_2 par leurs expressions données en (II-14) on obtient

$$\delta_i = Q_i - P^*_0(B_{0,3}(u_i) + B_{1,3}(u_i)) - P^*_3(B_{2,3}(u_i) + B_{3,3}(u_i)) - \lambda_1(P_1 - P_0)B_{1,3}(u_i) - \lambda_2(P_{n-1} - P_n)B_{2,3}(u_i) \quad (\text{II-16})$$

L'erreur globale sera:
$$\delta = \sum_{i=0}^q (\delta_i)^2 \quad (\text{II-17})$$

Le critère suffisant pour minimiser la quantité δ est

$$\frac{\partial \delta}{\partial \lambda_1} = 0 \quad \text{et} \quad \frac{\partial \delta}{\partial \lambda_2} = 0 \quad (\text{II-18})$$

On obtient ainsi un système de deux équations linéaires qui donne, une fois résolu, les inconnues λ_1 et λ_2 .

L'obtention de λ_1 et λ_2 permet d'évaluer la déviation δ qui doit être comparée avec la précision ε imposée au départ. Si $\delta > \varepsilon$ on subdivise la courbe initiale $P(u)$ en deux morceaux et on refait le processus d'approximation précédent pour chaque morceau jusqu'à l'obtention de la précision demandée.

La même logique peut être appliquée pour l'obtention des continuités géométriques supérieures (G^2 , G^3 , etc.), mais cette fois-ci le nombre d'inconnues λ_i sera plus important (il sera de 4 pour G^2 par exemple) et le système d'équations à résoudre deviendra un système non-linéaire. Toutefois et sauf pour le premier décalage, il est inutile d'exiger des ordres de continuité élevés pour les trajectoires d'outil.

II-4-3-6. Méthode de Klass

Dans cette méthode, R. Klass [Kla 83] propose d'approximer la courbe offset d'une courbe *spline cubique* plane par une autre courbe de même type. Comme une courbe spline cubique se définit par deux points extrêmes et les tangentes en ces points, Klass propose de calculer les points extrêmes de la courbe parallèle directement à partir de la courbe mère en utilisant l'expression (II-1). Les directions des tangentes en ces points (pour la courbe parallèle) peuvent être prises identiques à celles des extrémités de la courbe initiale tandis que leurs modules sont estimés de telle sorte que la différence des rayons de courbure entre la courbe initiale et la courbe parallèle sera égale à l'offset d .

Pour contrôler la précision de l'approximation, on procède à la comparaison dans un nombre déterminé de points de la déviation entre la courbe parallèle exacte et celle approximée. Si cette déviation est plus importante que la tolérance fixée par l'utilisateur, on subdivise la courbe initiale en deux morceaux et on répète le processus pour chacun de ces morceaux jusqu'à l'obtention de la précision désirée.

II-4-4. Problèmes de paramétrage

En interpolation des formes à pôles le choix du type de paramétrage dans la construction des courbes est un facteur important pour éviter certains désagréments (cf. §II-4-3-1). L'effet du paramétrage est souvent comparable à celui du polygone caractéristique et parfois d'une influence plus significative que celle de la distribution des valeurs nodales intérieures [Har 80]. Dans le cas de calcul des offset ce problème ne concerne que les méthodes nécessitant l'attribution de valeurs paramétriques aux points d'interpolation; de ce fait la méthode de Tiller et Hanson n'est pas concernée.

Pour les modèles de Bézier, Bézier rationnelle, B-spline non uniforme et NURBS on utilise souvent l'une des méthodes suivantes:

- paramétrage uniforme;
- paramétrage proportionnel aux intervalles nodaux;
- paramétrage proportionnel aux distances entre les points d'interpolation;
- paramétrage proportionnel aux racines carrées des distances entre les points d'interpolation.

II-4-4-1. Paramétrage uniforme

Soit un ensemble de $(p+1)$ points P_i^* , $i \in \{0, 1, \dots, p\}$, à interpolier (ou à approximer).

Le paramétrage uniforme produira la séquence de valeurs paramétriques suivante:

$$\mu_i = u_{\min} + (u_{\max} - u_{\min})(i/p), \quad i \in \{0, 1, \dots, p\}, \quad (\text{II-19})$$

où u_{\min} et u_{\max} sont les valeurs, respectivement, minimale et maximale de la suite nodale adoptée; on obtient, par conséquent, $\mu_i \in [u_{\min}, u_{\max}] \quad \forall i \in \{0, 1, \dots, p\}$.

Pour les courbe de Bézier, on prend le cas particulier $u_{\min} = 0$ et $u_{\max} = 1$ donc

$$\mu_i = i/p, \quad \mu_i \in [0, 1], \quad i \in \{0, 1, \dots, p\}, \quad (\text{II-20})$$

II-4-4-2. Paramétrage proportionnel aux intervalles nodaux

Comme il a été évoqué dans (§II-4-3-1), cette méthode pose certains problèmes (la matrice du système d'interpolation pourrait être singulière). Un paramétrage tenant compte de la forme du vecteur nodal de la courbe à construire et permettant de respecter la condition de Schoenberg - Whitney a été donné [Har 80]. Dans ce cas, les valeurs du paramètre sont données par:

$$\mu_i = (u_{i+1} + \dots + u_{i+m})/m, \quad i \in \{0, 1, \dots, p\} \quad (\text{II-21})$$

où m est le degré de la courbe.

II-4-4-3. Paramétrage proportionnel aux distances entre les points d'interpolation

Le paramétrage proportionnel aux distances entre les points d'interpolation P_i^* s'exprime par la relation:

$$\mu_i = \mu_{i-1} + \frac{\|P_i^* P_{i-1}^*\|}{\sum_{j=0}^{p-1} \|P_{j+1}^* P_j^*\|} (u_{\max} - u_{\min}), \quad i \in \{1, \dots, p\} \quad (\text{II-22})$$

avec $\mu_0 = u_{\min}$

On obtient ainsi des valeurs sur $[u_{\min}, u_{\max}]$.

Le paramétrage uniforme donne souvent des formes oscillantes avec des allures inattendues et contenant parfois des points singuliers (Fig. II-19a et II-20b) [Leo 91, Lee 89, Eps 76].

Les mêmes constatations sont valables pour le paramétrage proportionnel aux intervalles nodaux (Fig. II-19b et Fig. II-20c).

Le paramétrage proportionnel aux distances entre les points est souvent présenté comme la meilleure méthode de paramétrage en construction des formes paramétriques [Leo 91, Rog 89, Eps 76]. En réalité il peut être aussi à l'origine de l'obtention de formes avec oscillations (Fig. II-19c et II-20d) comme il a été montré par [Lee 89].

II-4-4-4. Paramétrage proportionnel aux racines carrées des distances entre les points d'interpolation

Pour résoudre ces problèmes, Lee [Lee 89] a proposé d'employer les cumuls des racines carrées des distances entre les points de l'interpolation. On calcule, donc, les valeurs du paramétrage par:

$$\mu_i = \mu_{i-1} + \frac{\sqrt{\|P_i^* P_{i-1}^*\|}}{\sum_{j=0}^{p-1} \sqrt{\|P_{j+1}^* P_j^*\|}} (u_{\max} - u_{\min}), i \in \{1, \dots, p\} \quad (\text{II-23})$$

pour un ensemble de points P_i^* , $i \in \{0, \dots, p\}$.

Cette méthode est appelée « méthode de la centripète » car elle repose sur le principe de minimisation de la quantité $(ds/du)^2/\rho$ où s est l'abscisse curviligne, ρ le rayon de courbure et u le paramètre de la courbe d'interpolation. En mécanique, la minimisation de cette grandeur conduit à minimiser la force centrifuge agissant sur un point en mouvement sur la courbe d'interpolation.

La méthode de Lee est plus performante en la comparant aux méthodes précédentes comme on peut le constater sur les exemples des figures II-19 et II-20. Signalons toutefois que le paramétrage de Lee engendre un supplément non négligeable de calcul numérique.

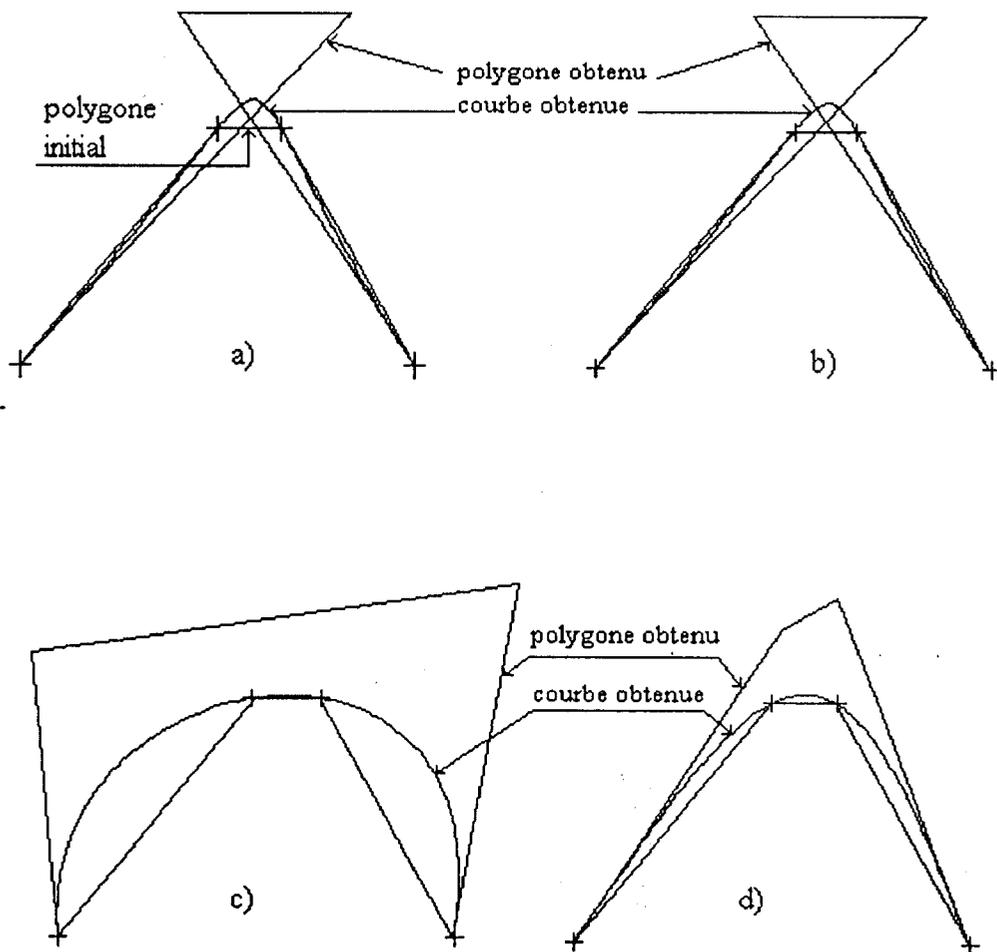


Fig. II-19 Influence du paramétrage sur la forme du polygone et de la courbe d'interpolation
 a) paramétrage uniforme b) paramétrage proportionnel aux intervalles nodaux
 c) paramétrage proportionnel aux distances entre points d) paramétrage de Lee

II-4-5. Conclusion

Les méthodes de polygonisation et de discrétisation - conversion ont l'inconvénient d'exiger des espaces de stockage importants et ne permettent pas de conserver l'homogénéité de la base de données du système de CFAO utilisateur.

Les méthodes d'interpolation et de lissage par les moindres carrées donnent parfois des formes indésirables et sont sujettes à des problèmes de stabilité numérique. En effet, ces deux méthodes ne sont recommandées que pour approximer des formes sculptées (dessins à main levée) où la précision n'est pas prioritaire [Fau 90].

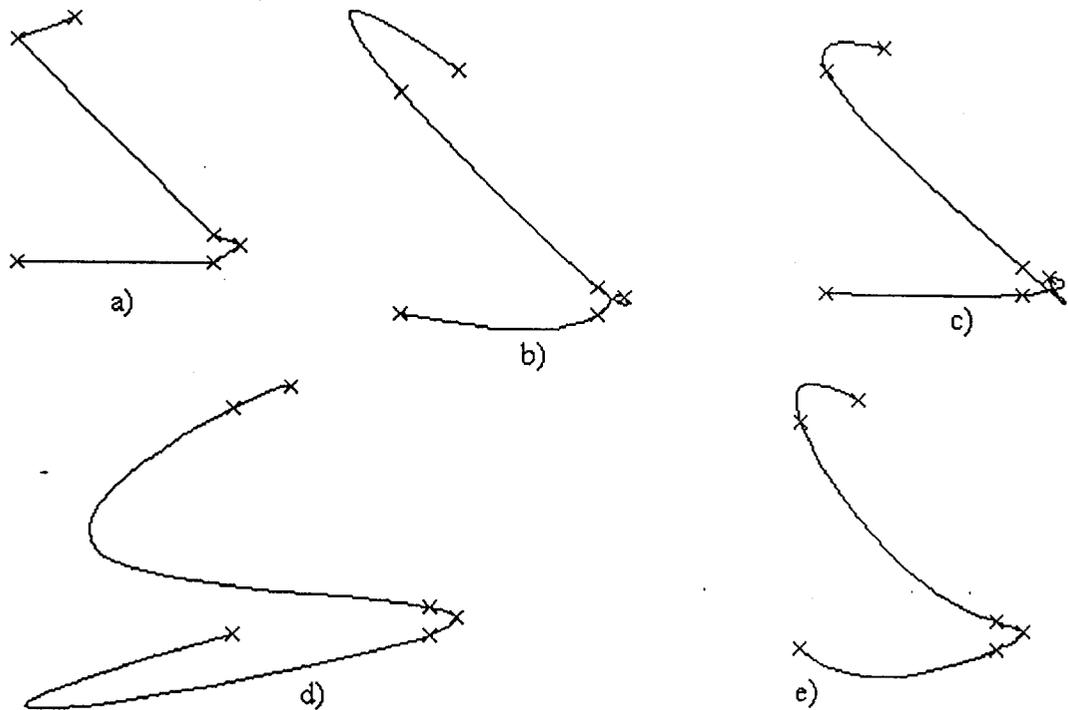


Fig. II-20 Influence du paramétrage sur l'oscillation des courbes d'interpolation

a) polygone initial des points à interpoler

b) paramétrage uniforme

c) paramétrage proportionnel aux intervalles nodaux

d) paramétrage proportionnel aux distances entre points

e) paramétrage de Lee

La méthode de Coquillart ne permet pas d'approcher la courbe parallèle exacte avec une précision arbitrairement fixée.

La méthode de Hoschek, bien qu'elle soit facile à mettre en oeuvre pour la continuité géométrique G^1 , est inutilisable pour les courbes autre que celle de Bézier qu'après transformation (en une forme de Bézier). Cette approche provoque, aussi, une augmentation considérable de la taille de la base de données du fait des subdivisions successives que subit la courbe initiale.

La méthode de Klass souffre des inconvénients de la méthode de Hoschek ajoutant à cela les difficultés de manipulation des formes splines cubiques. Ces dernières nécessitent à chaque fois l'introduction des tangentes aux points de jonction [Fau 87].

La méthode de Tiller et Hanson (méthode du polygone caractéristique parallèle) est valable pour les formes de Bézier et B-splines non uniformes (intégrales ou rationnelles). Elle permet, en outre, d'approcher la courbe parallèle exacte avec n'importe quelle précision. Cette

méthode est considérée actuellement comme la meilleure approche de construction des courbes gauches parallèles planes [Pha 92, Suh 90].

Les performances de la méthode de Tiller et Hanson nous ont conduit à l'adopter dans notre travail après les quelques adaptations suivantes.

La précision d'approximation utilisée n'est pas la même pour tous les décalages. Nous avons choisi de réitérer le processus « calcul du polygone parallèle - comparaison - insertion » jusqu'à l'obtention de la précision fixée mais uniquement pour le premier décalage (que se soit pour la poche ou pour les îlots). Les trajectoires suivantes sont calculées en une seule itération sans prise en compte de la précision exigée au départ. L'expérience pratique nous a montré dans ce cas que la déviation entre les courbes offset exactes et celles approximées est très proche de la précision demandée. Un tel résultat s'explique par l'affinement important que les courbes initiales avaient subi au premier décalage.

II-5. Subdivision des courbes à pôles

La subdivision des courbes, l'augmentation du nombre de sommets d'un polygone caractéristique et l'insertion de noeuds (cas des B-splines et NURBS) constituent autant d'opérations nécessaires en conception comme en fabrication. Dans ce dernier cas, ces opérations sont utilisées, particulièrement, pour traiter les interférences (cf. §II-7) ou pour affiner la forme d'une courbe afin de permettre l'obtention d'une courbe offset approximant au mieux sa courbe exacte associée (cf. §II-4).

II-5-1. Subdivision d'une courbe de Bézier

La subdivision d'une courbe intervient, en particulier, lorsqu'on désire la découper en un point I défini par une valeur paramétrique t tel que $I = P(t)$. L'objectif est alors de déterminer les polygones caractéristiques de deux courbes $P_1(u)$ et $P_2(u)$, $u \in [0, 1]$, de même degré que $P(u)$ à condition de conserver la même forme initiale *exacte* de $P(u)$.

Cette construction pourrait se réaliser en utilisant l'algorithme de de Casteljau (cf. §I-2-1-3). Le polygone caractéristique de $P_1(u)$ correspondra à l'intervalle $[0, t]$, tandis que celui de $P_2(u)$ sera sur l'intervalle $[t, 1]$.

L'algorithme est le suivant:

* soit n pôles P_i formant le polygone caractéristique de la courbe $P(u)$ à subdiviser;

```

* initialisation:
  FOR j = 0 TO n
    { Pj[0] = Pj }
* FOR k = 1 TO n
  { FOR j = 0 TO n-k
    { Pj[k] = (1-t) Pj[k-1] + t Pj+1[k-1] }
  }
* résultat:
  FOR j = 0 TO n
    { P1j = P0[j],
      P2j[n-j]
    }

```

La figure II-21 montre un exemple de subdivision d'une courbe de Bézier quadrique.

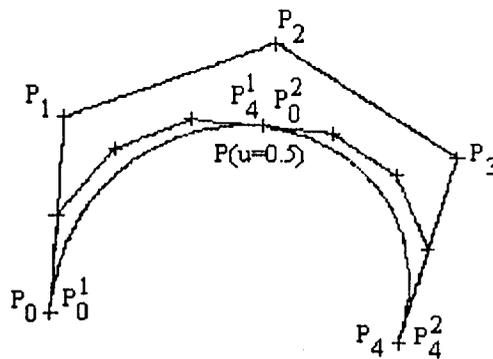


Fig. II-21 Subdivision au milieu d'une courbe de Bézier quadrique

II-5-2. Elévation du degré d'une courbe de Bézier

L'insertion d'un sommet dans le polygone caractéristique d'une courbe de Bézier est recommandée lorsqu'on veut affiner la forme de cette courbe afin de localiser sa modification ou de contrôler la précision d'approximation de sa courbe parallèle. Ce procédé s'utilise aussi pour construire des surfaces à partir d'un ensemble de courbes de nombres de sommets différents.

Soit $P_i, i \in \{0, 1, \dots, n\}$, un polygone caractéristique d'une courbe de Bézier de degré n définit par l'équation:

$$P(u) = \sum_{i=0}^n B_{i,n}(u) P_i, u \in [0, 1]$$

L'insertion d'un pôle consiste à construire une courbe de Bézier $P^*(u)$ correspondant *exactement* à $P(u)$ et dont le polygone caractéristique est $P^*_i, i \in \{0, 1, \dots, n+1\}$. Les pôles P^*_i se déduisent des pôles P_i par l'algorithme suivant [Leo 91, Fau 87, Qiu 87]:

```

{
P*0 = P0 ;
P*n+1 = Pn ;
FOR i = 1 TO n
  { P*i = Pi-1 + (n+1-i)(Pi - Pi-1)/(n+1)
  }
} // END

```

La figure II-22 montre deux exemples de cette opération.

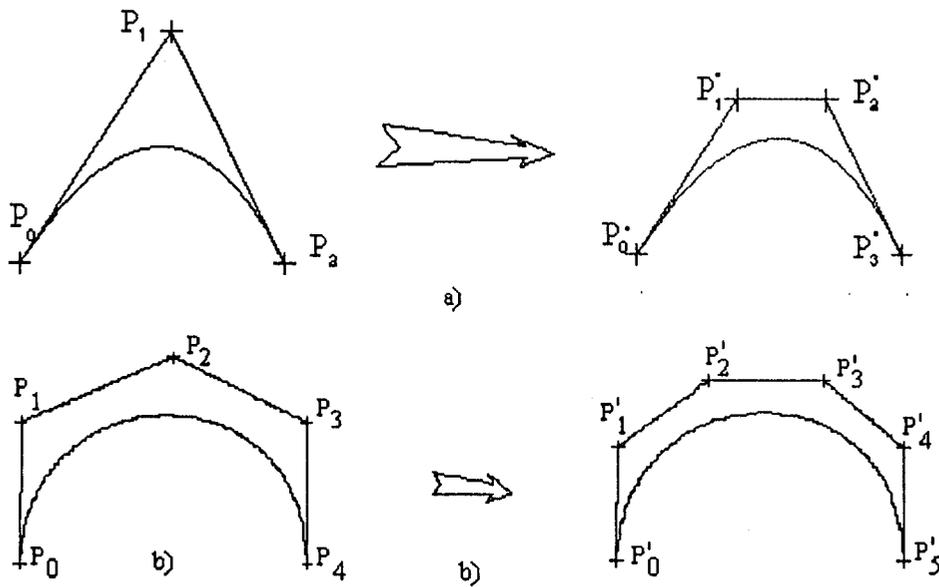


Fig. II-22 Elévation du degré d'une courbe de Bézier: a) $n = 2$ b) $n = 4$

II-5-3. Insertion de noeuds (B-splines non uniformes)

Pour les mêmes buts mentionnés ci-dessus, l'affinement ou le découpage d'une courbe B-spline ou NURBS est réalisé par insertion de noeuds dans la séquence nodale associée. L'opération d'insertion de noeuds, dans le cas des B-splines *uniformes*, est incompatible avec le principe de structuration de la séquence nodale de ce modèle.

La répétition de m fois (m étant le degré de la courbe) du processus d'insertion d'une même valeur t pour une courbe donnée conduit à la subdivision de celle-ci au point $u = t$.

Le problème de détermination du polygone caractéristique d'une courbe B-spline non uniforme $P(u)$ de degré m , après avoir modifié son vecteur nodal en y ajoutant un ou plusieurs noeuds tout en laissant le paramétrage et la forme *inchangés*, est résolu en utilisant, généralement, soit l'algorithme d'Oslo [Bar 88b], soit l'algorithme d'insertion de noeuds de Boehm [Boe 80, Boe 85]. Nous exposerons ici ce dernier algorithme.

* données

** une courbe B-spline non uniforme $P(u)$:

- de polygone caractéristique $P_i, i \in \{0, 1, \dots, n\}$;

- de degré m ;

- et de vecteur nodal $u_i, i \in \{0, 1, \dots, n+m+1\}$;

** et une valeur t à insérer;

* recherche de l'intervalle nodal d'insertion, i.e.:

chercher l'indice p tel que $u_p \leq t < u_{p+1}$ avec $m \leq p \leq n$;

* insertion

$i = p+m$;

$k = p+1+m$;

$P_{k-m} = P_{i-m}$

IF $t \leq u_i$ AND $i > p$

THEN { $u_k = u_i$

$i = i - 1$

$k = k - 1$

}

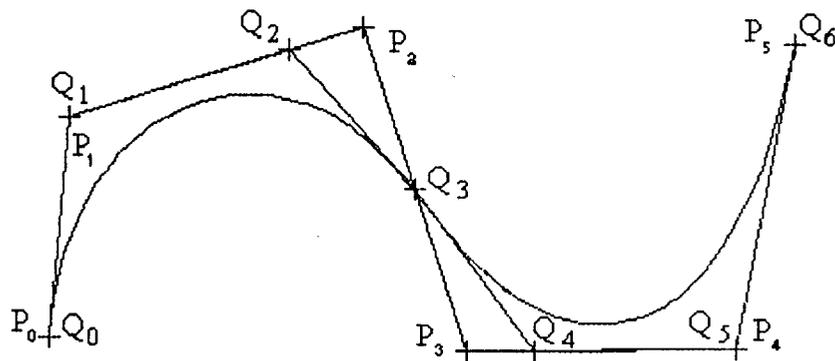
ELSE { FOR $l = 1$ TO n

```

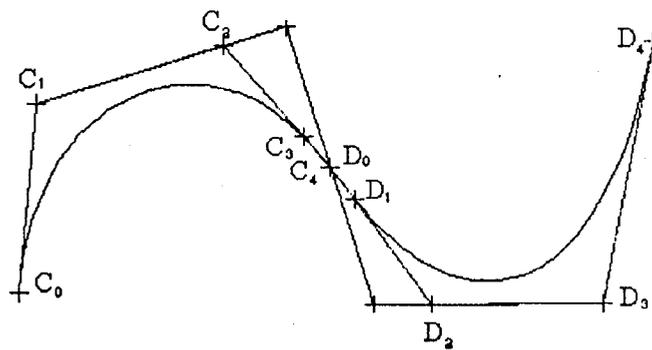
{ q = uk+1 - t
  IF q ≤ 0 THEN { Pk-n+1-1 = Pk-m+1 }
  ELSE { q = q / (uk+1 - ui-m+1)
        Pk-m+1-1 = q Pk-m+1-1 + (1 - q) Pk-m+1
      }
} // FOR
uk = t
} // ELSE
n = n + 1 // le nombre de pôles augmente de 1
* END

```

Cet algorithme est écrit de telle sorte que les anciens pôles du polygone caractéristique et l'ancien vecteur nodal soient écrasés et remplacés par les nouvelles valeurs.



a)



b)

Fig. II-23 Courbe B-spline non uniforme cubique de 6 pôles (initialement)

a) insertion d'un noeud $t = 1.5$

b) subdivision par insertion triple du noeud $t = 1.5$

La figure II-23a montre un exemple d'insertion d'un noeud $t = 1.5$ dans la suite nodale $U = [0, 0, 0, 0, 1, 2, 3, 3, 3, 3]$ d'une courbe B-spline non uniforme de degré $m = 3$ et de polygone caractéristique $P_i, i \in \{0, 1, \dots, 5\}$. On obtient ainsi une courbe de polygone caractéristique $Q_i, i \in \{0, 1, \dots, 6\}$, de degré 3 et de vecteur nodal $[0, 0, 0, 0, 1, 1.5, 2, 3, 3, 3, 3]$.

La figure II-23b montre la subdivision de cette même courbe au point $P(u = 1.5)$ par l'insertion triple de la valeur nodale $t = 1.5$ dans sa séquence nodale. On obtient ainsi deux courbes de suites nodales respectivement $[0, 0, 0, 0, 1, 1.5, 1.5, 1.5, 1.5]$ et $[1.5, 1.5, 1.5, 1.5, 2, 3, 3, 3, 3]$.

Notons enfin que la forme de la courbe originale est sauvegardée intacte dans les deux cas.

II-6. Calcul des intersections des courbes gauches

L'intersection des courbes gauches est un moyen fondamental de conception (construction sous contraintes, ...) et de fabrication (traitement des interférences dans les trajectoires d'outil) des formes géométriques.

Un algorithme de recherche des points d'intersection doit être fiable, précis et rapide afin de permettre le traitement de tous les cas possibles et l'obtention des résultats avec le minimum d'erreurs numériques et de favoriser le fonctionnement hautement interactif nécessaire dans les modeleurs géométriques.

Il existe plusieurs méthodes de détermination des points d'intersection de deux courbes paramétriques planes; les plus utilisées sont:

- les techniques non linéaires utilisant, en général, une méthode numérique de type Newton - Raphson [Leo 91, Faux 87];
- les techniques de subdivision, utilisant les propriétés liant les courbes à leurs polygones caractéristiques associés [Leo 91, Sed 86, Sed 90, Pen 84];
- les techniques algébriques, permettant de calculer l'expression implicite de l'une des courbes et de résoudre par la suite une équation non linéaire [Leo 91, Sed 84, Sed 85, Sed 86, Sed 89, Gol 87].

II-6-1. Techniques non linéaires

Le calcul du point d'intersection de deux courbes paramétriques planes $P_1(u_1) = (x_1(u_1), y_1(u_1))^T$ et $P_2(u_2) = (x_2(u_2), y_2(u_2))^T$ revient à résoudre le système d'équations non linéaires

$$\begin{cases} x_1(u_1) - x_2(u_2) = 0 \\ y_1(u_1) - y_2(u_2) = 0 \end{cases} \quad (\text{II-24}) \quad (u_1, u_2) \in [u_{1 \min}, u_{1 \max}] * [u_{2 \min}, u_{2 \max}]$$

qu'on peut écrire

$$\begin{cases} f_x(u_1, u_2) = x_1(u_1) - x_2(u_2) = 0 \\ f_y(u_1, u_2) = y_1(u_1) - y_2(u_2) = 0 \end{cases} \quad (\text{II-25})$$

Ce système de deux équations non linéaires à deux inconnues (u_1 et u_2) peut être résolu par plusieurs méthodes numériques [Che 86, Ral 78, Dem 79, Nou 93]. Parmi ces méthodes utilisées (dans le contexte de la CFAO) on trouve que la méthode de Newton - Raphson est d'un emploi très courant [Vic 85, Til 84, Mar 89, Hos 85, Che 86, Fau 87, Mic 92, Aom 90, Bar 88b, etc.].

La solution du système (II-25) peut donc s'écrire

$$u_1^{i+1} = u_1^i - \frac{1}{J(u_1^i, u_2^i)} \begin{vmatrix} f_x(u_1^i, u_2^i) & \frac{\partial f_x}{\partial u_2}(u_1^i, u_2^i) \\ f_y(u_1^i, u_2^i) & \frac{\partial f_y}{\partial u_2}(u_1^i, u_2^i) \end{vmatrix} \quad (\text{II-26})$$

$$u_2^{i+1} = u_2^i - \frac{1}{J(u_1^i, u_2^i)} \begin{vmatrix} \frac{\partial f_x}{\partial u_1}(u_1^i, u_2^i) & f_x(u_1^i, u_2^i) \\ \frac{\partial f_y}{\partial u_1}(u_1^i, u_2^i) & f_y(u_1^i, u_2^i) \end{vmatrix}$$

avec

$$J(u_1^i, u_2^i) = \begin{vmatrix} \frac{\partial f_x}{\partial u_1}(u_1^i, u_2^i) & \frac{\partial f_x}{\partial u_2}(u_1^i, u_2^i) \\ \frac{\partial f_y}{\partial u_1}(u_1^i, u_2^i) & \frac{\partial f_y}{\partial u_2}(u_1^i, u_2^i) \end{vmatrix} \quad (\text{II-27})$$

$J(u_1^i, u_2^i)$ représente le Jacobien du système (II-25); il doit être non nul pour pouvoir obtenir une solution unique [Dem 79]. Les u_1^i et u_2^i sont les solutions à l'itération (i).

La méthode de Newton - Raphson, bien qu'elle offre l'avantage d'être rapide au voisinage de la solution, ne permet pas de calculer, à la fois, qu'une seule racine (on peut parler donc d'une convergence locale plutôt que d'une convergence globale [Iva 85]). Le choix des solutions initiales, pour démarrer le processus itératif, conditionne en grande partie l'aptitude de la méthode pour converger vers la bonne solution.

La présence d'un Jacobien nul (on parle dans le cas général d'une matrice Jacobienne singulière) pendant une itération conduit à l'impossibilité de définir les données de l'itération suivante. Ce cas se présente, dans le cas de l'intersection de deux courbes planes, lorsque les tangentes aux points $P_1(u_1^i)$ et $P_2(u_2^i)$ sont parallèles (Fig. II-24).

L'application de la méthode de Newton - Raphson pour déterminer tous les points d'intersection de deux courbes planes s'effectue suivant les étapes suivantes:

- (1) On cherche les points d'intersection entre les polygones caractéristiques associés aux courbes;
- (2) si ces points existent, on calcule les valeurs initiales des racines en prenant les valeurs des paramètres proportionnelles à leurs positions sur les polygones caractéristiques;
- (3) pour chaque point d'intersection localisé dans l'étape (1) on calcule les solutions par les expressions (II-26) et (II-27);
- (4) on arrête le processus si:
 - * $\|P_1(u_1^{i+1}) - P_2(u_2^{i+1})\| < \epsilon$, où ϵ est une tolérance fixée par l'utilisateur,
 - * le nombre maximal d'itérations est dépassé; dans ce cas on considère qu'il n'existe pas d'intersection en ce lieu.

Si l'algorithme s'arrête à cause de la présence d'un Jacobien nul, il sera impératif soit d'utiliser une autre méthode, soit de reformuler le problème [Mar 89].

La méthode de Newton possède un défaut majeur au niveau de l'étape 1; car on risque de rencontrer des cas où il y a des intersections entre les courbes sans qu'il y ait autant sur leurs polygones caractéristiques associés (Fig. II-25b). La résolution du problème inverse (Fig. II-25a) ne pose pas généralement de grandes difficultés.

L'une des solutions qu'on peut employer, pour résoudre ce problème, est celle utilisée par Tiller et Hanson [Til 84]. Ils ont proposé d'affiner les deux courbes en leur insérant un nombre « suffisant » de noeuds supplémentaires. Ce procédé peut permettre d'obtenir des polygones

de contrôles très proches de leurs courbes associées et de diminuer, par conséquent, la probabilité de présence de cas similaires à ceux de la figure II-25.

II-6-2. Techniques de subdivision

Ces méthodes utilisent la propriété du polygone convexe des courbes (Fig. II-26) et les techniques de subdivision [Pen 84]. La propriété du polygone convexe signifie que la courbe est située à l'intérieur du polygone convexe associé au polygone caractéristique [Bez 85], comme le montre la figure II-26. Dans le cas des courbes rationnelles cette propriété n'est pas vérifiée que si les coordonnées homogènes associés aux pôles sont admissibles (cf. §I-3).

L'algorithme de recherche d'un point d'intersection entre deux courbes par subdivision procède par comparaison des polygones convexes des deux courbes. S'il n'existe pas de recouvrement entre ces polygones, les courbes sont considérées comme non sécantes. Si les polygones se recouvrent, on subdivise les deux courbes, et on refait de nouveau le test de recouvrement pour les polygones convexes résultants. Dans chaque itération, on élimine les parties (de chaque courbe) qui ne contient pas d'intersection, i.e. celles qui ne se recouvrent pas.

Le processus de subdivision - élimination est poursuivi jusqu'au point où on arrive à approximer les deux courbes par des segments de droite (en réalité, la courbe doit se située à l'intérieur d'un rectangle de hauteur fixée par l'utilisateur [Sed 90, Sed 86]). Le point d'intersection de ces deux segments représente le point d'intersection des deux courbes. Sederberg et Parry [Sed 86] ont donné une méthode d'estimation du nombre de subdivisions nécessaires pour trouver le point d'intersection entre deux courbes de Bézier.

Des simplifications peuvent être portées à cette méthode en remplaçant, par exemple, les polygones convexes par des rectangles minimax [Las 89]. La figure II-27 montre les quatre premières itérations de cet algorithme sur un exemple d'intersection de deux courbes de Bézier en adoptant cette proposition.

Pour profiter de la rapidité de l'algorithme de de Casteljau pour la subdivision des courbes de Bézier, Peng [Pen 84] propose de convertir les courbes B-splines en courbes de Bézier [Ben 90, Bar 88a] avant de démarrer le processus de subdivision. Cette solution assure une convergence plus rapide du polygone vers la courbe et réduit globalement le temps d'exécution de l'algorithme.

La technique de subdivision est très fiable; elle permet de localiser souvent tous les points d'intersection et d'évaluer « approximativement » toutes les valeurs des paramètres u_1 et u_2 sur chaque courbe à condition d'imposer une tolérance suffisamment faible ou un nombre de subdivisions suffisamment grand [Leo 91, Sed 86]. En contre partie elle est lente.

Pour résoudre le problème d'intersection des courbes tangentes (Fig. II-24b), Sederberg et Nishita [Sed 90] ont introduit la notion du foyer d'une courbe. Cette notion permet, en l'utilisant avec un procédé de découpage « clipping » particulier des courbes, de traiter les intersections en des points tangents et de séparer les points d'intersection très proches l'un de l'autre.

II-6-3. Techniques algébriques (technique d'implicitisation - inversion)

Ces techniques utilisent le fait que chaque courbe paramétrique rationnelle peut être exprimée par une équation implicite [Sed 86, Sed 84, Sed 89, Sed 85]. En d'autres termes, une courbe de Bézier rationnelle plane $P(u)$ de degré n et de forme:

$$P(u) = \frac{\sum_{i=0}^n B_{i,n}(u)h_i P_i}{\sum_{i=0}^n B_{i,n}(u)h_i} \quad (\text{II-28})$$

peut être exprimée par la forme implicite suivante

$$f(x,y) = \begin{vmatrix} L_{0,0}(x,y,h) & \dots & L_{0,n-1}(x,y,h) \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ L_{n-1,0}(x,y,h) & \dots & L_{n-1,n-1}(x,y,h) \end{vmatrix} = 0 \quad (\text{II-29})$$

où

$$L_{i,j}(x,y,h) = a_{i,j}x + b_{i,j}y + c_{i,j}h = \sum_{\substack{k \leq \min(i,j) \\ k+m=i+j+1}} C_k^n C_m^n h_k h_m (P_k - P) \times (P_m - P) \quad (\text{II-30})$$

avec $P_1 = (x_1 y_1)^T$ représente les pôles du polygone caractéristique associé à la courbe $P(u)$, h_i représente les poids associés à ces pôles et P représente le vecteur position $(x y)^T$ des points de la courbe. La forme implicite (II-29) décrit exactement la même courbe paramétrique $P(u)$.

Soit Q une deuxième courbe paramétrique de degré m définit, en coordonnées homogènes, par le vecteur position $Q(t) = (x(t).h(t) y(t).h(t) h(t))^T$. Si on substitue les coordonnées de cette courbe dans l'équation (II-29) on obtient:

$$f(t) = \begin{vmatrix} a_{0,0}x(t) + b_{0,0}y(t) + c_{0,0}h(t) & \dots & a_{0,n-1}x(t) + b_{0,n-1}y(t) + c_{0,n-1}h(t) \\ \vdots & \ddots & \vdots \\ a_{n-1,0}x(t) + b_{n-1,0}y(t) + c_{n-1,0}h(t) & \dots & a_{n-1,n-1}x(t) + b_{n-1,n-1}y(t) + c_{n-1,n-1}h(t) \end{vmatrix} = 0 \quad (\text{II-31})$$

$f(t)$ est un polynôme de degré mn et ses racines sont les valeurs du paramètre t des points d'intersection. La détermination de ces racines peut se faire par une méthode de type Jenkins - Traub [Ral 78] qui permet l'obtention de toutes les racines du polynôme, mais dans notre cas on ne doit retenir que les racines réelles comprises entre 0 et 1.

En ce qui concerne le calcul de la valeur du paramètre u de la première courbe, on le réalise par la résolution d'une équation du système suivant [Sed 86]:

$$\begin{bmatrix} L_{0,0}(x, y, h) & \dots & L_{0,n-1}(x, y, h) \\ \vdots & \ddots & \vdots \\ L_{n-1,0}(x, y, h) & \dots & L_{n-1,n-1}(x, y, h) \end{bmatrix} \begin{bmatrix} (1-u)^{n-1} \\ (1-u)^{n-2}u \\ \vdots \\ (1-u)u^{n-2} \\ u^{n-1} \end{bmatrix} = 0 \quad (\text{II-32})$$

Une telle opération s'appelle inversion [Sed 86, Sed 85].

La technique algébrique du calcul des points d'intersection des courbes paramétriques planes est très rapide dans le cas des courbes de degré inférieur à 5 [Sed 86]; mais au-delà de cette valeur, elle devient de moins en moins performante. Ce comportement est dû au fait que cette approche repose essentiellement sur la recherche des racines d'une forme polynomiale de degré égal au produit des degrés des deux courbes d'intersection. A titre d'exemple, l'intersection de

deux courbes de Bézier de 11 pôles pour chacune nécessite la recherche des racines d'un polynôme de degré 100?

II-6-4. Autres techniques

D'autres méthodes de recherche des intersections existent. Elles utilisent:

- les procédures d'affichage des écrans graphiques [Fab 85],
- la résolution des systèmes d'équations différentielles [Sac 88],
- des approches analytiques [Gol 87],
- la méthode du résultant [Sed 89, Mic 92],
- ...

Ces méthodes sont moins connues et parfois n'ont aucun intérêt réel [Fab 85].

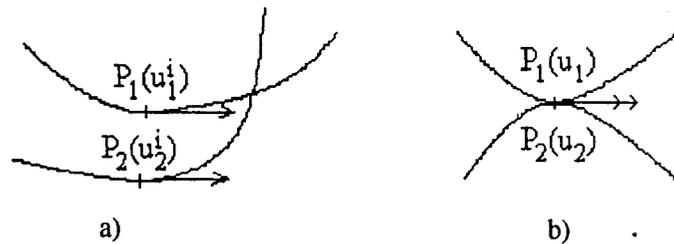


Fig. II-24 Deux cas de tangentes parallèles (itération i)

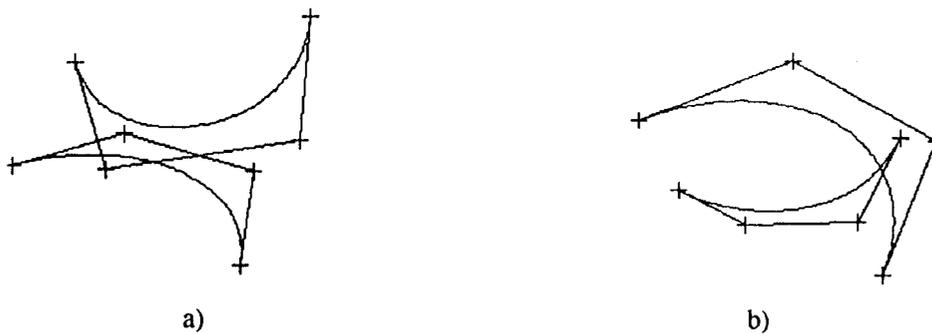


Fig. II-25

a) Intersection entre les polygones uniquement

b) Intersection entre les courbes uniquement

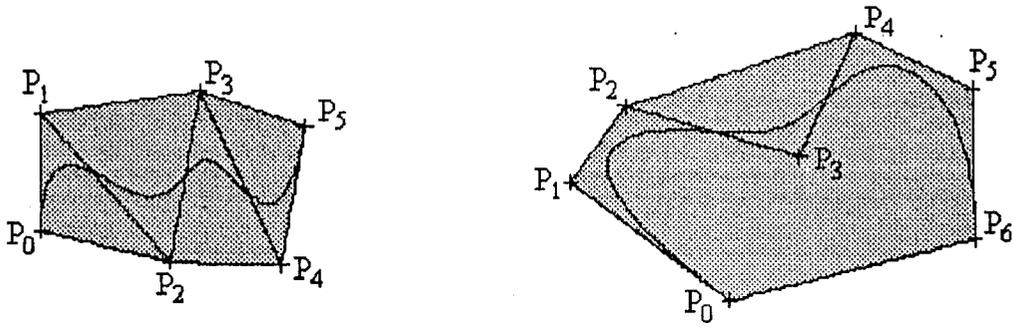


Fig. II-26 Polygones convexes

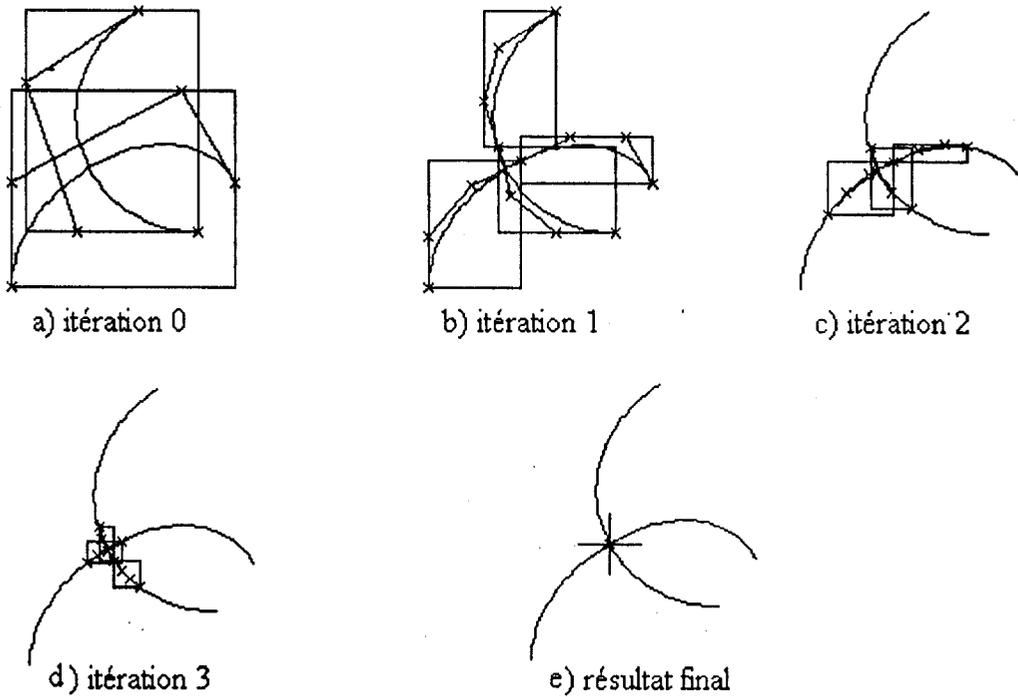


Fig. II-27 Méthode de la subdivision - élimination

II-6-5. Conclusion

Bien que les techniques d'intersection soient nombreuses, il n'existe pas actuellement de méthodes capables de traiter rapidement et efficacement les problèmes généraux d'intersection [Leo 91].

Une solution à ce problème consiste à utiliser plus d'une méthode à la fois (méthodes hybrides). On emploie, souvent, la méthode de subdivision pour les premières itérations afin de mieux séparer les racines; des solutions plus affines seront, ensuite, obtenues par la méthode de Newton - Raphson.

II-7. Traitement des singularités et des interférences sur les trajectoires d'outil

La détermination de la meilleure approximation des contours offset (donc des trajectoires de l'outil) n'est qu'une partie du problème de balayage des poches par les décalages successifs des profils. Un ensemble de traitements doit être effectué sur ces trajectoires afin d'éviter toute collision outil - profil et interférence.

II-7-1. Types de singularités

Les problèmes d'interférence sont soit d'ordre local, i.e. apparaissent sur une courbe, soit d'ordre global, i.e. mettent en question deux ou plusieurs courbes.

Dans le premier cas, une courbe représentant une trajectoire d'outil, peut contenir les singularités suivantes:

- les points de rebroussement (Fig. II-28a);
- les auto-intersections (self-intersections) qui apparaissent sur *une* courbe et provoquent des interférences appelées boucles locales (Fig. II-28b).

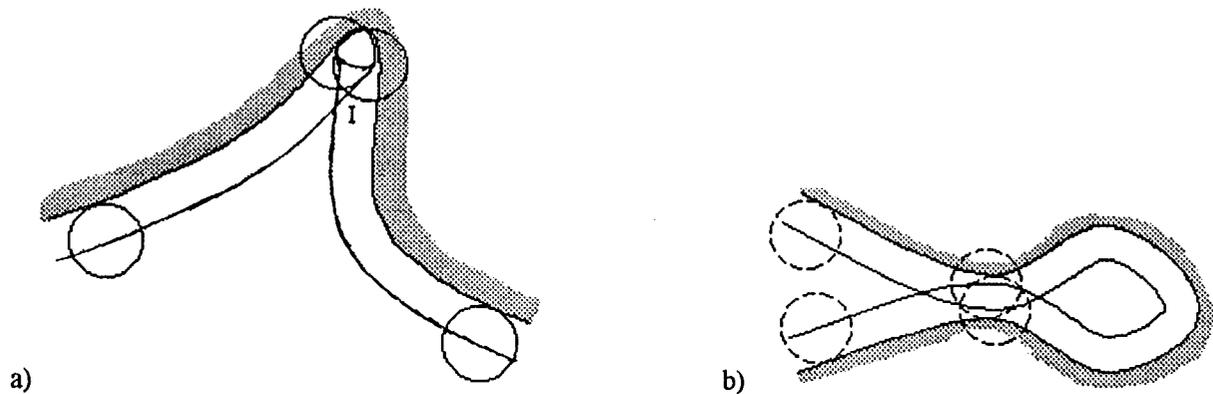


Fig. II-28 Singularités sur les courbes offset: a) points de rebroussement b) auto-intersections

Dans le deuxième cas (singularités d'ordre global), les contours, supposés constitués de plusieurs courbes, peuvent présenter les singularités suivantes:

- des zones indéfinies (Fig. II-29a);
- des zones de collision (Fig. II-29b)
- des sous poches dégénérées et qui sont de type:

- boucles intracontours causées par la présence de(s) recouvrement(s) entre les courbes offset de la poche (le profil extérieur) (Fig. II-30a) ou de l'îlot (Fig. II-30c);
- boucles intercontours provoquées par l'existence d'intersections entre les contours offset poche - îlot (Fig. II-30e) ou îlot - îlot (Fig. II-30g).

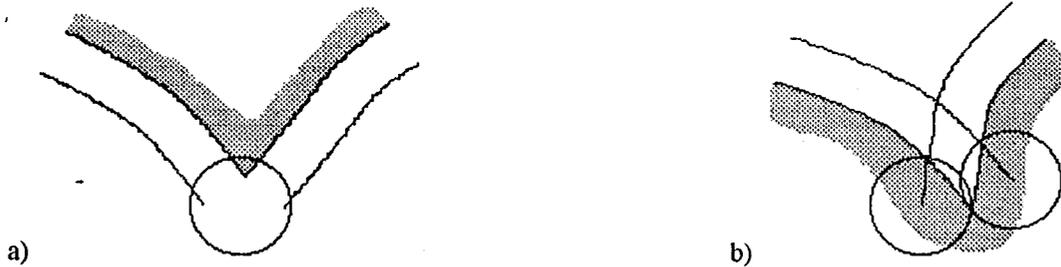


Fig. II-29 Zones d'interférence: a) zone indéfinie b) zone de collision

II-7-2. Solutions aux traitements des singularités

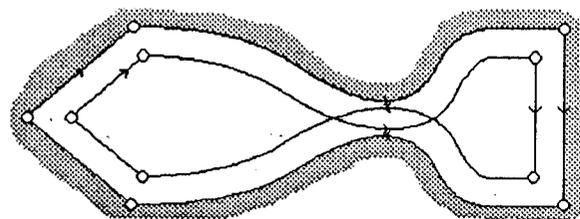
II-7-2-1. Points de rebroussement

Un point de rebroussement aura lieu sur une courbe offset $P'(u)$ lorsque le rayon de courbure de la courbe initiale $P(u)$ est plus petit que la distance de décalage (qui est souvent le diamètre de la fraise diminuée d'une distance de recouvrement (cf. §III-1-3)). Nous exposerons dans cette section quelques méthodes utilisées pour éliminer les interférences causées par la présence de tels points.

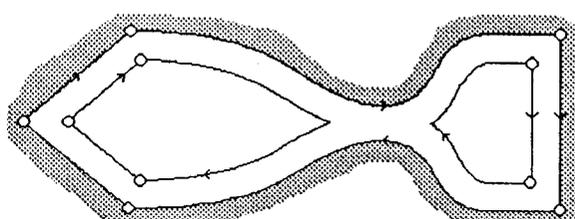
II-7-2-1-1. Méthode de la tangente

Soit

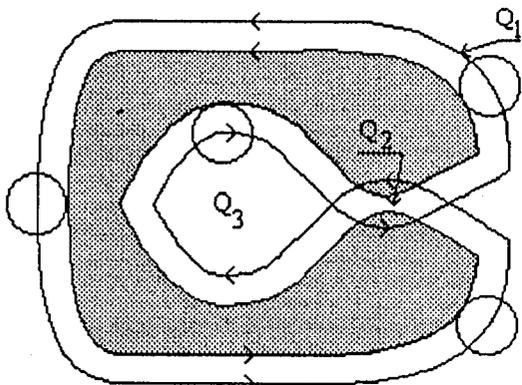
- d : une distance de décalage;
- $\rho(u^*)$: le rayon de courbure au point $P(u^*)$;
- $t(u^*)$, $n(u^*)$: respectivement le vecteur tangent unitaire et le vecteur normal unitaire de la courbe $P(u)$ au point $P(u^*)$;
- $t^*(u^*)$: le vecteur tangent unitaire de la courbe offset $P^*(u)$ au point $P^*(u^*)$.



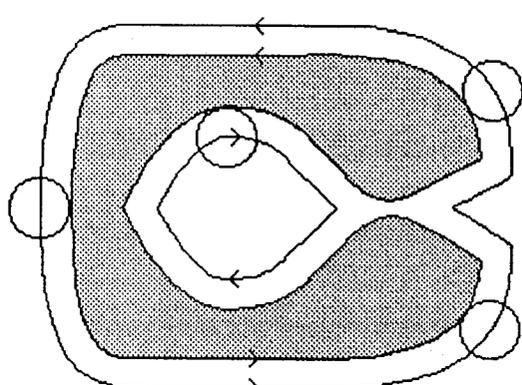
a)



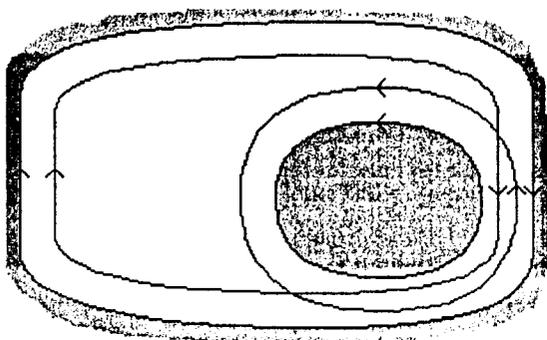
b)



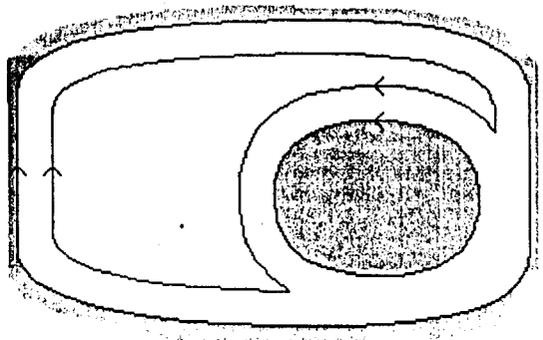
c)



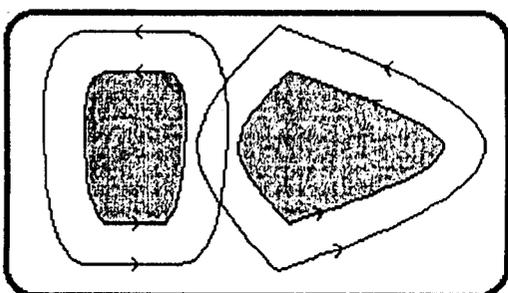
d)



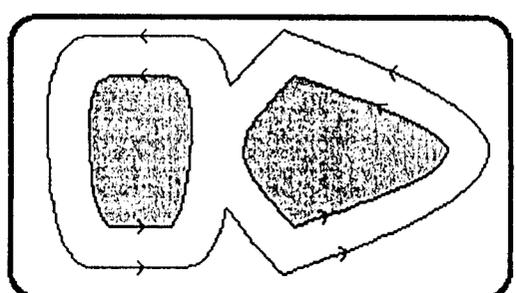
e)



f)



g)



h)

Fig. II-30 Boucles: a) et c) boucles intracontours; e) et g) boucles intercontours

On démontre que:

- $t^*(u^*) = t^*(u^*)$ si $d.n(u^*)$ n'est pas dirigé vers le centre de courbure de $P(u)$ en u^*

ou si $\rho(u^*) > |d|$.

- $t^*(u^*)$ n'existe pas si $\rho(u^*) = |d|$ et le vecteur $d.n(u^*)$ est dirigé vers le centre de courbure de la courbe $P(u)$.

- $t^*(u^*) = -t^*(u^*)$ si $\rho(u^*) < |d|$ et $d.n(u^*)$ est dirigé vers le centre de courbure de la courbe $P(u)$.

Autrement dit, si $d.n(u^*)$ est dirigé vers le centre de courbure d'une courbe alors, la tangente de sa courbe parallèle doit changer de direction lorsque $\rho(u^*)$ devient égale à $|d|$ [Til 84].

Sur ce constat, il est bien possible théoriquement de déterminer tous les points de rebroussement sur une courbe de décalage en procédant, en deuxième étape, à la subdivision de cette courbe au milieu de chaque deux paires de points de rebroussement; le problème devient ainsi un problème de recherche des points d'intersection entre deux courbes (cf. §II-6).

Dans chaque opération on obtient deux points d'intersection, l'un présente le point de jonction et l'autre le début et la fin de la boucle; à cette étape nous sommes en mesure d'enlever la partie indésirable de la trajectoire.

La figure II-31a montre une trajectoire qui contient quatre points de rebroussement

$B_1 = P(u_1)$, $B_2 = P(u_2)$, $B_3 = P(u_3)$ et $B_4 = P(u_4)$. La courbe est subdivisée aux points

$u^*_1 = (u_1+u_2)/2$ et $u^*_2 = (u_3+u_4)/2$ pour donner lieu à trois courbes. Après l'élimination des zones de collision, on obtient la trajectoire montrée sur la figure II-31b.

II-7-2-1-2. Méthode de l'hodographe

Soit $P(u)$ une courbe de Bézier cubique plane de polygone caractéristique P_i . Cette courbe est définie par l'expression:

$$P(u) = \sum_{i=0}^3 B_{i,3}(u)P_i \quad (\text{II-33})$$

Soit $H(u)$ la courbe hodographe de $P(u)$ [Bezi 86] définie par:

$$H(u) = \frac{dP}{du}(u) = \sum_{i=0}^3 \frac{dB_{i,3}}{du}(u)P_i \quad (\text{II-34})$$

où la dérivée de la fonction de Bernstein $B_{i,3}(u)$ est donnée par:

$$\frac{dB_{i,3}}{du}(u) = 3 (B_{i-1,2}(u) - B_{i,2}(u)) \quad (\text{II-35})$$

l'expression de $H(u)$ peut s'écrire:

$$H(u) = 3 \sum_{i=0}^2 P_i (B_{i-1,2}(u) - B_{i,2}(u)) = 3 \sum_{i=0}^2 (P_{i+1} - P_i) B_{i,2}(u) \quad (\text{II-36})$$

La dérivée d'une courbe de Bézier de degré n est une courbe de Bézier de degré $n-1$; la courbe hodographe $H(u)$ est une courbe quadratique (c'est une parabole); les pôles notés H_i de son polygone caractéristique associé sont exprimés par la relation:

$$H_i = 3(P_{i+1} - P_i), \quad i \in \{0, 1, 2\} \quad (\text{II-37})$$

Kim [Kim 93] a utilisé une propriété connue des courbes de Bézier cubiques [Fau 87] pour détecter les points de rebroussement: le point $P(u^*)$ est un point de rebroussement si et seulement si $H(u)$ passe par l'origine, et si ce point existe, alors il est unique. Le problème donc devient un problème de recherche d'un point d'intersection entre une courbe de Bézier quadratique (l'hodographe en l'occurrence) et une droite (verticale ou horizontale) passant par l'origine. Des méthodes de résolution analytiques peuvent être ainsi envisagées; la valeur u^* trouvée comme solution pour ce problème d'intersection représente elle même le point de rebroussement sur la courbe cubique initiale. Notons, pour première simplification, que si l'origine n'est pas située à l'intérieur du triangle (H_0, H_1, H_2) la courbe ne possède aucun point de rebroussement (propriété de la variation décroissante).

II-7-2-1-4. Méthode proposée

Dans notre travail, nous avons retenu la démarche suivante:

(a) On procède pendant le calcul de chaque position du centre d'outil sur la courbe mère au calcul de la quantité $(\rho(u) - d)$ et on repère les points dans lesquels le signe de cette quantité change. On obtient une série de valeurs paramétriques $u_i, i \in \{1, \dots, p\}$.

(b) Pour chaque couple de valeurs (u_i, u_{i+1}) on calcule l'aire du triangle $(P(u_i), P((u_{i+1}+u_i)/2), P(u_{i+1}))$. Si cette aire possède le même signe que l'aire du contour contenant la courbe $P(u)$ alors on considère qu'il existe un point de rebroussement sur l'intervalle $]u_i, u_{i+1}[$ sinon on rejette les valeurs u_i et u_{i+1} . On obtient une nouvelle série de valeurs $u_i, i \in \{1, \dots, q \leq p\}$.

(c) On élimine par subdivision les portions de courbe situées entre chaque couple de valeurs (u_j, u_{j+3}) pour $j = 1, 5, 9, \dots, q-3$. On obtient une série de courbes qu'on insère dans le contour à la place de la courbe initiale.

Une fois ces courbes sont décalées, les points de rebroussement n'apparaîtront plus; mais on obtient des zones de collision. Le traitement de ce problème sera discuté dans §II-7-4.

Les résultats obtenus par ce procédé sont très satisfaisants. Le problème du choix de la solution initiale et de la convergence ne se pose pas avec cette méthode comme nous l'avons constaté dans l'approche de Tiller et Hanson [Til 84] qui ont considéré le problème d'élimination des points de rebroussement comme un problème d'auto-intersection. Ce problème sera étudié dans la section suivante.

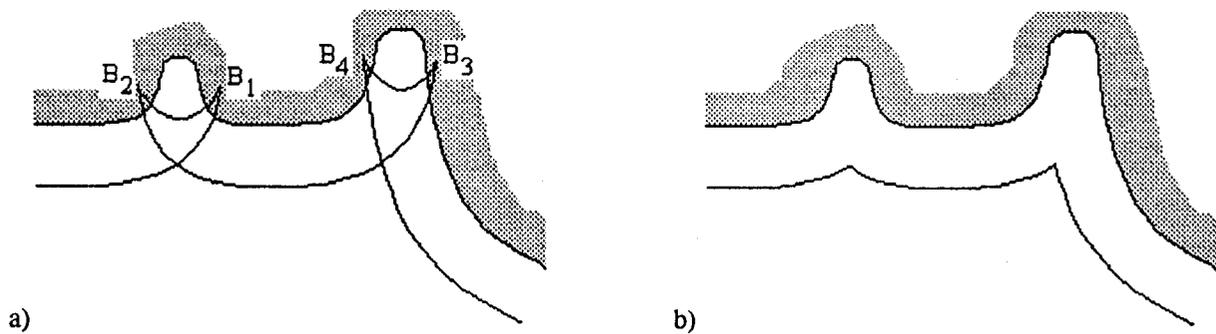


Fig. II-31 Points de rebroussement: a) trajectoire avec 4 points de rebroussement b) trajectoire « filtrée »

II-7-2-2. Calcul des auto-intersections

On dit qu'une courbe $P(u)$ possède une auto-intersection s'il existe un ou plusieurs couples (u_1, u_2) de valeurs du paramètre u pour lesquels la relation suivante est vérifiée:

$$P(u_1) = P(u_2) \quad \text{avec} \quad u_1 \neq u_2 \quad (\text{II-38})$$

L'étude de ce problème, dans le cas d'une courbe gauche paramétrique, est généralement difficile du fait que:

- une courbe gauche peut avoir un ou plusieurs points d'auto-intersection bien que d'une part son polygone caractéristique n'en comporte aucun (Fig. II-32a) d'autre part:
 - l'existence d'une auto-intersection dans un polygone de contrôle n'est pas une condition suffisante pour que sa courbe associée possède une boucle (Fig. II-32b);
 - l'application de l'algorithme de subdivision tel qu'il est utilisé pour déterminer les intersections de deux courbes (cf. §II-6) en introduisant, comme données, deux courbes identiques n'est pas envisageable car il n'y aura aucune partie à éliminer de la courbe du fait que le test de recouvrement est toujours positif;
 - l'emploi de la méthode de Newton - Raphson risque de converger vers des solutions triviales car toute valeur du paramètre u peut vérifier le système d'équations (II-38).
- Nous examinerons ici deux méthodes de résolution de ce problème; la méthode de subdivision - élimination de Lasser [Las 89] et la méthode de Newton - Raphson.

II-7-2-2-1. Méthode de subdivision - élimination de Lasser

La méthode de Lasser [Las 89] est, dans son principe, analogue à la méthode de subdivision utilisée pour le calcul des points d'intersection entre deux courbes paramétriques gauches (cf. §II-6-2). Néanmoins elle repose sur les deux constatations importantes suivantes:

(1)- Soit $C = P(u)$ une courbe de Bézier de degré n et de polygone caractéristique associé P_i avec $i \in \{0, 1, \dots, n\}$.

Posons $\alpha = \sum_{i=1}^n |\alpha_i|$ où α_i est l'angle entre deux segments $P_{i-1}P_i$ et P_iP_{i+1} du polygone caractéristique de la courbe $P(u)$ (Fig. II-33).

Théorème du critère de l'angle de rotation: la courbe de Bézier $P(u)$ ne possède aucun point d'auto-intersection si l'angle de rotation α de son polygone caractéristique associé est inférieur ou égal à 180° . Si $\alpha > 180^\circ$, alors la courbe peut avoir une auto-intersection. Les figures II-33a et II-33b illustrent la première situation tandis que les figures II-33c et II-33d illustrent la deuxième.

(2)- La figure II-34 montre une courbe de Bézier (C) composée des segments de courbe $\{C_0, C_1, C_2, \dots, C_7\}$. Elle comporte les configurations suivantes:

- configuration 1: un segment de courbe (C_5) qui contient un point d'auto-intersection P_1 ;

- configuration 2: deux segments adjacents (C_2 et C_3) donnant à leur point d'intersection P_2 un point d'auto-intersection;
- configuration 3: deux segments non adjacents (C_1 et C_6) donnant à leur point d'intersection P_3 un point d'auto-intersection sur la courbe C ;
- configuration 4: toute autre situation ne créant pas un point d'auto-intersection.

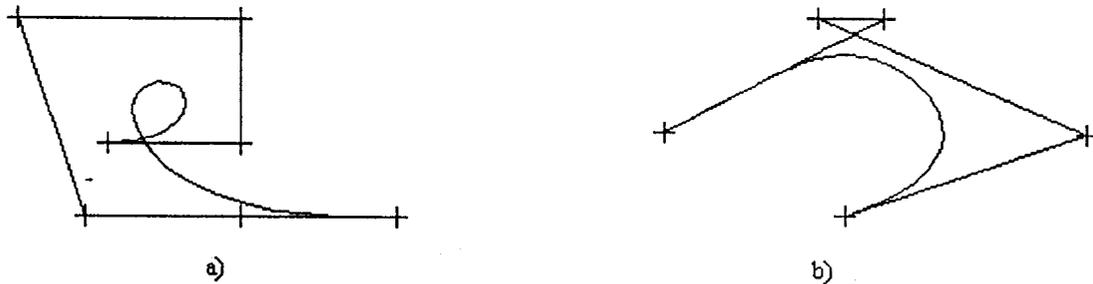


Fig. II-32 a) Polygone sans auto-intersections et courbe avec une auto-intersection
 b) polygone avec une auto-intersection et courbe sans auto-intersections

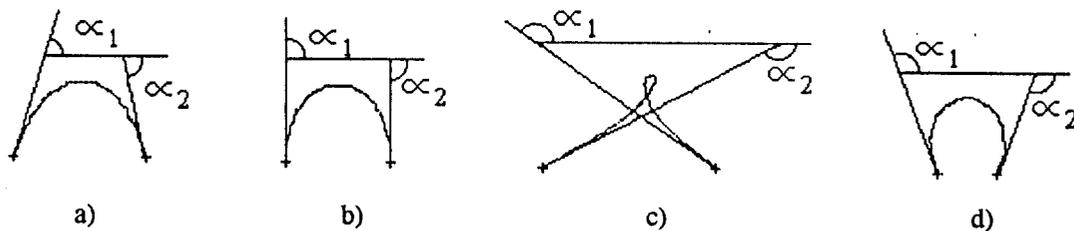


Fig. II-33 Mesure des angles pour le critère de l'angle de rotation

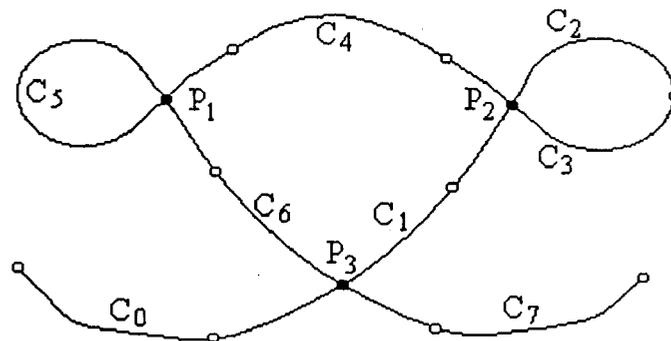


Fig. II-34 Les configurations possibles en subdivision d'une courbe avec boucles

La subdivision d'une courbe de configuration 1 donne des segments de configuration 1 et/ou de configuration 2. La subdivision d'une courbe de configuration 2 donne des segments de configuration 2 et/ou de configuration 3 tandis que la subdivision d'une courbe de

configuration 3 donne des segments de configurations 3 et/ou de configuration 4. La subdivision conduit toujours à des configurations d'ordre égal ou supérieur au numéro de la configuration de la courbe subdivisée..

Les subdivisions successives et l'utilisation conjuguée du critère de l'angle de rotation et du test de recouvrement entre les polygones convexes aboutissent à ne retenir que les segments de configuration 3 après avoir éliminer les segments des autres configurations. Le test de recouvrement peut être effectué soit sur les polygones convexes soit sur les rectangles minimax associées aux courbes.

En ne retenant que les segments de la troisième configuration le problème devient ainsi un problème de détermination, à chaque fois, d'un point d'intersection entre deux courbes. Un tel problème peut être évidemment résolu par l'une des méthodes exposées dans §II-6.

La méthode de Lasser est applicable uniquement pour les courbes de Bézier. Son utilisation pour d'autres modèles exige leur transformation en courbes de Bézier à l'aide d'algorithmes appropriés [Ben 90].

II-7-2-2. Méthode de Newton - Raphson

Dans cette méthode on cherche l'existence des points d'intersection sur le polygone caractéristique de la courbe; ces points servent comme une solution initiale au système d'équations suivant:

$$\begin{cases} x(u_1) = x(u_2) \\ y(u_1) = y(u_2) \end{cases} \quad (\text{II-39})$$

avec $u_{\min} \leq (u_1, u_2) \leq u_{\max}$ et $u_1 \neq u_2$.

Comme il a été signalé auparavant, une courbe peut présenter un point d'auto-intersection sans qu'il existe dans son polygone caractéristique. Un tel cas peut être évité, plus ou moins, en utilisant la méthode de Tiller et Hanson dans l'approximation des courbes offset de décalage. En effet cette méthode donne des polygones caractéristiques très proches de la courbe approximée au point où à toute courbe possédant une boucle est associé un polygone caractéristique contenant aussi une boucle.

II-7-2-2-3. Comparaison

La méthode de Lasser est caractérisée par une stabilité numérique importante et une indépendance vis-à-vis de l'estimation des solutions initiales; cependant elle est lente à cause du temps élevé nécessaire pour les opérations de subdivision d'une part et valable uniquement pour les courbes de Bézier de l'autre part. La méthode de Newton - Raphson par contre est très rapide mais elle présente l'inconvénient de converger toujours vers la solution la plus proche. La probabilité de tomber dans ce cas sur une solution triviale est relativement importante. Néanmoins, l'utilisation de la méthode de Tiller et Hanson en calcul des courbes offset permet de minimiser cette tendance.

Des choix retenus auparavant (calcul des offset) ainsi que la facilité remarquable d'implémentation de la méthode de Newton - Raphson ont été les principales raisons pour choisir au début cette méthode dans notre programme. Par la suite nous avons jugé inutile de chercher les solutions précises et nous avons opté pour la subdivision systématique des courbes auxquelles sont associées des polygones caractéristiques avec des auto-intersections. Constatons toutefois que cette action n'élimine pas les interférences détectées; cette tâche est laissée au module de traitement des boucles intracontours (cf. §II7-5-1).

II-7-3. Zones indéfinies (les raccordements)

Quand les vecteurs normaux aux points de jonction de deux courbes adjacentes d'un profil ne sont pas identiques on obtient sur le profil offset une zone indéfinie (Fig. II-29a) ou une zone d'interférence (Fig. II-29b) suivant le sens de décalage effectué (vers l'extérieur ou vers l'intérieur). Deux solutions sont généralement utilisées pour traiter le premier cas:

- (1)- Rajouter deux segments de droite entre les courbes (Fig. II-35a). Ces segments sont définis par la direction et le point d'intersection des tangentes des courbes aux extrémités.
- (2)- En utilisant les segments de droite, l'outil s'éloignerait trop de la trajectoire désirée au-delà d'un angle déterminé θ ce qui occasionnerait des pertes de temps et surtout des risques de collision. On préfère souvent de rajouter un arc de cercle de raccordement délimité par les deux normales élevées aux extrémités par rapport à chacun des segments (Fig. II-35b) si l'angle θ dépasse certaine valeur. Cette dernière est, par exemple, de 60° sur les directeurs de commande Num [Num 88].

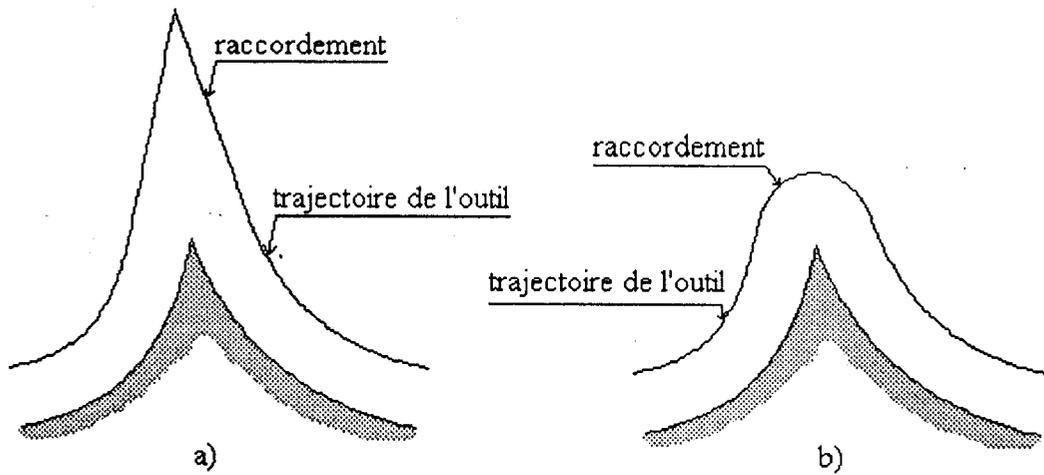


Fig. II-35 Les raccordements: a) raccordement avec deux segments de droite
b) raccordement avec un arc de cercle

Pour garder un seul modèle mathématique utilisable sur tout le contour on utilise des courbes de degré 1 pour présenter les segments de droite (dans la première solution) et des quadratiques pour les arcs de cercle (dans la deuxième solution). Un arc de cercle peut être approximé par une courbe de Bézier ou de B-spline tandis qu'il est exactement représentable par une courbe NURBS [Rog 89, Pie 87, Pie 91].

L'utilisation de la technique de l'accroissement des intervalles utiles des courbes [Leo 91] pour résoudre le problème de raccordement est très déconseillée. En effet, l'expérience nous a montré que cette technique donne des formes oscillantes et même des boucles imprévisibles et ne permet d'utiliser aucun moyen de contrôle de la précision du raccordement.

II-7-4. Zones de collision

Le traitement de ce problème se fait en deux étapes:

- 1- On détermine le point d'intersection des deux courbes décalées.
- 2- On subdivise ces deux courbes dans ce point d'intersection; on garde la première partie de la première courbe et la deuxième partie de la deuxième courbe et on rejette la deuxième partie de la première courbe et la première partie de la deuxième courbe (Fig. II-36). Cependant cette méthode engendre des zones non usinées et des coins ronds au lieu d'être aigus contrairement aux désirs du concepteur.

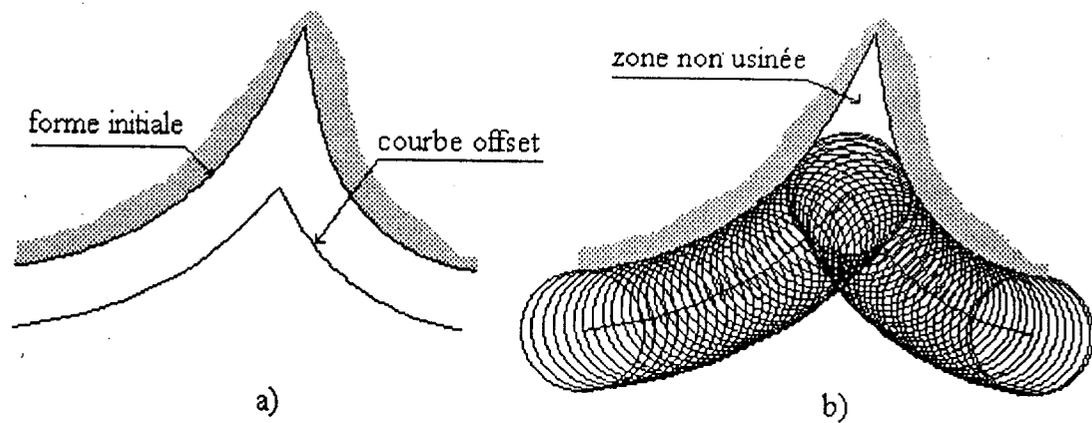


Fig. II-36 Trajectoire d'outil sans interférences

II-7-5. Détection et traitement des boucles globales

Malgré l'importance du problème de l'identification des boucles globales (le terme boucle globale est utilisé pour désigner les boucles résultant de l'intersection de plusieurs courbes à l'opposé d'une boucle locale qui est le résultat d'une intersection d'une courbe avec elle-même), la documentation traitant ce cas est rare; nous n'avons pu trouver qu'une seule référence [Mul 88].

[Mul 88] a donné une méthode de décomposition des contours issus de recouvrement de courbes elliptiques approximées par des courbes de Bézier rationnelles quadratiques. Cette méthode utilise la théorie des graphes d'Euler. C'est le principe général de cette approche que nous avons repris et développé pour écrire le module de traitement des boucles globales de notre système.

L'algorithme que nous proposons est composé de quatre étapes:

- (1)- Définition des courbes composant le contour fermé initial.
- (2)- Détermination pour chaque couple de deux courbes de tous les points d'intersection (à l'exception des points de jonction). De tels points d'intersection sont appelés *carrefours* (Fig. II-41). La localisation d'un carrefour devra être suivie par la subdivision de ses deux courbes correspondantes et la formation du vecteur des branchements. Un vecteur des branchements exprime les liens existants entre les courbes; il forme en quelque sorte un vecteur de liaisons. Ces trois opérations seront répétées jusqu'à l'épuisement de tous les points carrefours.
- (3)- A partir du vecteur des branchements on construit les sous - contours résultants. S'il n'y a pas d'intersection il n'y aura pas de boucles donc de sous - contours.

(4)- Sélection des contours désirés et élimination des autres contours présentant des zones de collision par l'emploi d'un test approprié (Fig. II-38b).

Nous avons employé dans le développement de nos implémentations pratiques des courbes paramétriques de type Bézier et B-splines non uniformes; néanmoins la discussion ultérieure reste tout à fait indépendante, dans son principe, du modèle choisi. C'est la rapidité d'exécution qui sera affectée du fait que l'algorithme fait des appels intensifs aux procédures de recherche des points d'intersection et de subdivision.

Ce problème se manifeste de plus en plus avec l'emploi d'un algorithme de subdivision pour calculer les intersections. Malgré tout, l'utilisation de la méthode de subdivision est vivement recommandée en détection des boucles afin d'obtenir un algorithme fiable et robuste. La moindre défaillance dans le calcul des intersections aura des effets imprévisibles sur le reste du traitement et de l'usinage; elle causera souvent des collisions outil - pièce.

II-7-5-1. Traitement des boucles intracontours

Les boucles intracontours sont dues à l'existence de zones de recouvrement entre les éléments géométriques d'un contour fermé (Fig. II-30a et II-37a). Nous utiliserons l'exemple de la figure II-37 pour expliquer le principe de la méthode d'identification des boucles intracontours que nous avons pu développer. Les détails algorithmiques de cette méthode seront discutés dans les sections suivantes.

Le contour de la figure II-37 est formé initialement de cinq courbes C_1 , C_2 , C_3 , C_4 et C_5 orientées dans le même sens et liées par le vecteur des branchements $F = \{2, 3, 4, 5, 1\}$. Ce dernier signifie que la deuxième extrémité de la courbe C_1 est liée à la première extrémité de la courbe C_2 , et celle de C_2 est liée à celle de C_3 , etc., et que la deuxième extrémité de la courbe C_5 est identique à la première extrémité de la courbe C_1 .

Cherchons un point d'intersection entre deux courbes. La recherche doit commencer tout naturellement dans l'ordre C_1 et C_2 , C_1 et C_3 , ..., C_4 et C_5 . On interrompt l'opération à la rencontre d'un carrefour. Dans notre exemple on doit s'arrêter dans le test liant C_1 et C_4 . Ces deux courbes possèdent un point d'intersection P_1 . On subdivise les deux courbes en ce point pour obtenir la séquence des courbes $\{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$ qui remplace l'ancienne séquence et un nouveau vecteur des branchements $F = \{6, 3, 4, 5, 2, 7, 1\}$. En continuant la recherche des carrefours (points d'intersection hors les points de jonction), on détecte le point P_2 (point d'intersection entre C_3 et C_4 dans la Fig. II-37b). On subdivise les courbe C_3 et C_4 au

point P_2 pour obtenir une séquence de neuf courbes (Fig. II-37c) et un vecteur des branchements $F = \{8, 3, 6, 5, 4, 7, 2, 9, 1\}$ comme le montre le tableau II-1. Le processus de recherche des intersections doit être arrêté car tous les carrefours sont épuisés (notons encore que les points de jonction ne doivent pas être pris en compte).

Utilisons maintenant le dernier vecteur des branchements obtenu pour construire les sous-contours (ou simplement les contours) dégénérés qu'on notera Q_i . Le premier contour Q_1 comporte la courbe C_1 . Celle-ci est branchée sur C_8 comme il est indiqué par le vecteur des branchements F , donc la deuxième composante de Q_1 sera C_8 . C_8 est branchée sur C_9 qui est branchée à son tour sur C_1 qui est la courbe de départ. Le contour est ainsi fermé. On peut écrire: $Q_1 = \{C_1, C_8, C_9\}$.

La courbe C_2 n'est pas encore utilisée, on peut donc la prendre comme un élément de départ dans la construction de Q_2 . C_2 est branchée sur C_3 , C_3 est branchée sur C_6 qui est elle-même branchée sur C_7 . Le vecteur des branchements oriente C_7 vers C_2 pris comme courbe de départ ce qui donne une boucle fermée. On peut donc écrire: $Q_2 = \{C_2, C_3, C_6, C_7\}$.

La construction de Q_3 commence par C_4 car les courbes C_1 , C_2 , et C_3 sont déjà utilisées par les contours précédents. C_4 est branchée sur C_5 qui est elle-même branchée sur C_4 pour former une boucle fermée. On peut donc écrire $Q_4 = \{C_4, C_5\}$.

En examinant la totalité des courbes on constate qu'elles sont toutes utilisées et que tous les sous contours sont construits conformément à la figure II-37d (sans l'utiliser évidemment).

Cet exemple donne une idée suffisante pour dégager un algorithme général d'identification des boucles intracontours.

Algorithme d'identification et de calcul des boucles intracontours

L'algorithme que nous proposons comporte les tâches suivantes:

(1) Construction de la séquence des courbes

Supposons qu'un carrefour ait été localisé entre les courbes C_i et C_j aux valeurs paramétriques respectivement u_1 et u_2 . On subdivise C_i au point $C_i(u_1)$ pour obtenir deux segments de courbe $C_{i,1}$ et $C_{i,2}$. Ensuite on subdivise C_j au point $C_j(u_2)$ pour obtenir deux autres segments de courbe $C_{j,1}$ et $C_{j,2}$. On remplace C_i par $C_{i,1}$ et on insère $C_{i,2}$ immédiatement après $C_{i,1}$ dans la séquence des courbes composant le contour (cette séquence est généralement une liste chaînée [Mart 89]). Le nombre de courbes (nc) se trouve ainsi incrémenté de 1. On

refait le même processus pour la courbe C_j . On se trouve en fin du compte avec un nombre de courbes augmenté de 2.

La routine suivante réalise l'insertion d'une courbe A dans la liste C entre les éléments C_i et C_{i+1} ; cette procédure a été appelée Insertion.

Procedure Insertion(C, A, i)

```
{
  FOR k = ne DOWN TO i+2
    {  $C_{k+1} = C_k$  }
   $C_{i+1} = A$ 
  nc = nc+1
} // END
```

La routine *Subdiv_Courbes1(C)* qui réalise la phase de subdivision aura besoin, en plus d'Insertion, de deux autres procédures:

- *Rec_Carrefour(B, D)* : une fonction permettant de localiser et de calculer le point carrefour, s'il existe, entre les courbes B et D. La fonction retourne la valeur logique TRUE (vrai) si un carrefour est localisé.
- *Reorientation1*: une procédure de mise à jour du vecteur des branchements noté F (voir plus loin).

Procedure Subdiv_Courbes1(C)

```
{ // nc : le nombre de courbes
  // initialisation du vecteur des branchements
  FOR i = 1 TO nc-1 {  $F_i = i + 1$  }
   $F_{nc} = 1$  // fin de l'initialisation
  i = 1
  REPEAT
    j = i + 1
    REPEAT
      B =  $C_i$ ;    D =  $C_j$ 
      // appel de la routine de recherche des carrefours Rec_Carrefour
      IF Rec_Carrefour(B, D) = TRUE THEN
```

```

{ Subdiviser B pour obtenir deux courbes B1 et B2
  Ci = B1 // remplacement de Ci par B1 dans la séquence des courbes
  Insertion(C, B2, i)
// ces opérations incrémentent nc de 1, par conséquent, Cj devient Cj+1
Subdiviser D pour obtenir deux courbes D1 et D2
  Cj+1 = D1 // remplacement de Cj+1 par D1 dans la séquence des courbes
  Insertion(C, D2, j + 1)

  // en total nc a été incrémenté de 2
Reorientation1(F, i, j) // mise à jour du vecteur des branchements F
  }
  j = j + 1
  UNTIL j > nc
  i = i + 1
  UNTIL i >= nc
} // END

```

(2) Construction du vecteur des branchements

Dans le cas général, et pour deux courbes C_i et C_j formant un carrefour, la subdivision donne quatre courbes $C_{i,1}$, $C_{i,2}$, $C_{j,1}$ et $C_{j,2}$. Les nouvelles réorientations qu'on devra introduire consistent à brancher $C_{i,1}$ sur $C_{j,2}$ et $C_{j,1}$ sur $C_{i,2}$ (Fig. II-41a) et d'inscrire ces changements au niveau du vecteur des branchements.

On réalise en première phase la transition entre l'ancien vecteur des branchements (avant subdivision et insertion) et le nouveau en utilisant un vecteur intermédiaire noté OF qui mémorise l'état précédent de F mais sans tenir compte des changements des branchements au niveau des deux dernières courbes subdivisées. A titre d'exemple, OF à l'étape (1) de l'exemple de la figure II-37 sera $OF = \{2, 2, 3, 4, 5, 1, 1\}$. Pour introduire les nouveaux branchements et obtenir ainsi le nouveau vecteur F il suffit de réorienter deux segments du contour, i.e. C_1 et C_5 respectivement vers C_6 et C_2 et d'effectuer les changements d'indices causés par les opérations de subdivision - insertion précédentes. Le vecteur des branchements deviendra donc $F = \{6, 3, 4, 5, 2, 7, 1\}$. On répète la procédure pour chaque étape, on obtient les résultats figurant sur le tableau II-1.

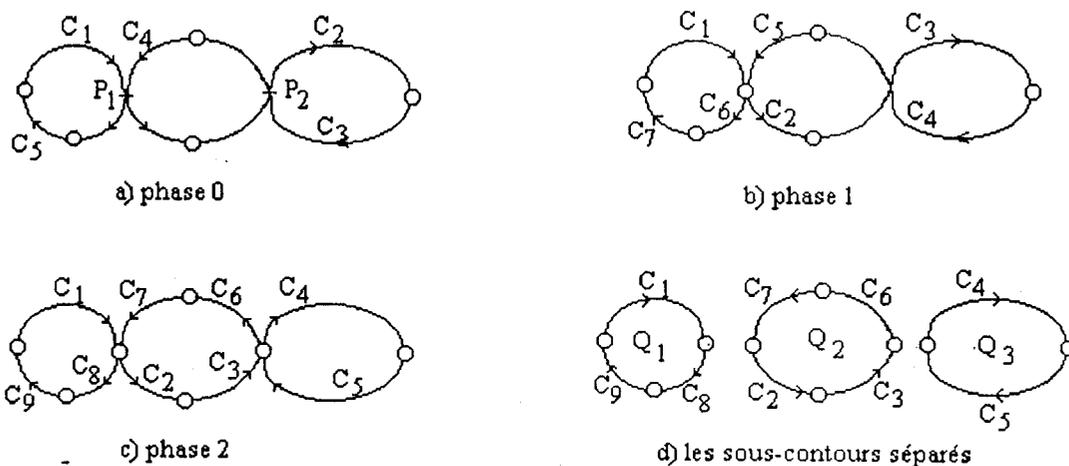


Fig. II-37 Les étapes de décomposition d'un contour fermé avec boucles

phase	0	1	2
nombre de courbes	5	7	9
séquence des courbes	C_1, C_2, C_3, C_4, C_5	$C_1, C_2, C_3, C_4, C_5, C_6, C_7$	$C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9$
vecteur des branchements	2, 3, 4, 5, 1	6, 3, 4, 5, 2, 7, 1	8, 3, 6, 5, 4, 7, 2, 9, 1

Tableau II-1

L'algorithme que nous proposons pour effectuer cette étape aura donc la forme suivante:

Procedure Reorientation1(F, i, j)

{

$k = 1$

$r = 1$

REPEAT

IF ($k > i$ AND $k > j + 1$) THEN

{ $OF_k = F_r$

$k = k + 1$

$r = r + 1$

}

ELSE { $OF_k = F_r$

```

        k = k+1
    }
UNTIL k > nc

FOR k = 1 TO nc { Fk = OFk }

FOR k = 1 TO nc
{ IF k = i THEN Fk = j+2
  ELSE IF k = j+1 THEN Fk = i+1
  ELSE IF Fk >= i+1 AND Fk <= j THEN Fk = Fk +1
  ELSE IF Fk >= j+1 AND Fk <= nc THEN Fk = Fk +2
}
} // END

```

(3) Construction des sous contours

Maintenant la phase de construction de tous les sous-contours dégénérés peut être entamée. C'est la dernière configuration du vecteur des branchements qui sera employée, i.e. le vecteur $F = \{8, 3, 6, 5, 4, 7, 2, 9, 1\}$ dans notre exemple. Les sous-contours (ou tout simplement contours) obtenus seront notés Q_i .

Notons en fin que pour reconnaître les courbes utilisées il est judicieux d'employer un tableau f de valeurs entières (ou logiques) jouant le rôle de bascules. Ce tableau sera initialisé à zéro au début du processus puis chaque élément prendra la valeur 1 une fois sa courbe correspondante est utilisée.

On propose de réaliser cette étape par la routine suivante:

Procedure Sub_Contours1(C, Q)

```

{
  // nc est le nombre de courbes dans le contour initial
  FOR i = 1 TO nc { fi = 0 } // initialisation du tableau des bascules
  i = 0; j = 0
  FOR k = 1 TO nc-1
  { IF fk = 0 THEN
    { i = i+1

```

```

j = 1
Qij = Ck
fk = 1
r = Fk
WHILE fr = 0
    { j = j+1
      Qij = Cr
      fr = 1
      r = Fr
    }
ni = j // ni représente le nombre de courbes dans le sous contour Qi
} // IF
} // FOR
ns = i // ns représente le nombre de sous-contours construits
} // END

```

(4) Test de sélection des sous contours

Examinons maintenant le sens d'orientation des contours construit précédemment. On remarque que Q₁ et Q₃ sont orientés dans le sens de rotation des aiguilles d'une montre tandis que Q₂ est orienté dans le sens inverse (trigonométrique).

Faisons le même examen pour les sous-contours de la figure II-38a représentant les limites d'une poche et une première trajectoire d'outil qui comporte cinq boucles dont deux sont des zones de collision (Q₂ et Q₄). Le sens du contour initial est horaire ainsi que ceux de Q₁, Q₃ et Q₅, par contre le sens d'orientation de Q₂ et Q₄ est trigonométrique c'est à dire l'inverse du sens du contour initial. La règle suivante peut être dégagée:

Critère de l'orientation initiale: tout sous-contour orienté dans le sens inverse du contour initial est une zone de collision; il doit être par conséquent éliminé de la trajectoire de l'outil.

L'application de ce critère sur la trajectoire de la figure II-38a a abouti à l'élimination de toutes les portions gênantes sur le chemin d'outil comme on peut le constater sur la fig. II-38b.

Le critère énoncé ci-dessus ne s'applique en réalité qu'aux cas de boucles issues d'un décalage vers l'intérieur effectué sur le contour extérieur de la poche. En effet et comme le montre la figure II-30c le contour initial de l'îlot est orienté dans le sens trigonométrique; après

décalage on obtient trois contours Q_1 , Q_2 et Q_3 . Q_1 et Q_2 sont orientés dans le sens trigonométrique (comme le contour initial) et Q_3 dans le sens horaire. Si on applique le critère de l'orientation initiale on se trouve dans l'obligation d'éliminer Q_3 bien qu'en réalité c'est Q_2 qu'il faut éliminer. La solution dans ce cas (et également pour *les cas où le nombre de boucles est plus grand*) consiste à retenir tous les contours orientés dans le sens horaire ainsi que le contour qui possède le plus grand périmètre parmi les contours orientés dans le sens trigonométrique (Fig. II-30d).

Algorithme général de traitement des boucles intracontours

Le programme général de calcul des boucles intracontours peut s'écrire:

```
{
  Subdiv_Courbes1(C)
  Sub_Contours1(C,Q)
  Select(Q) // Test de sélection des contours Q à retenir
} // END
```

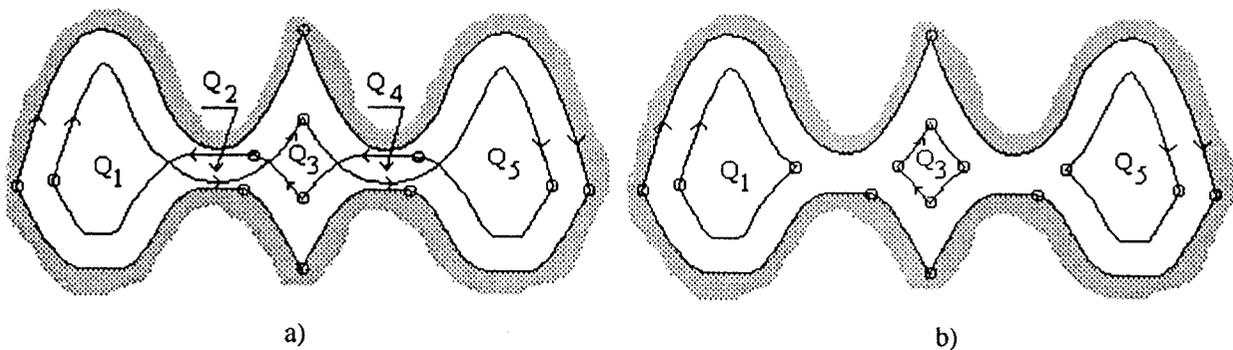


Fig. II-38 Application du critère de l'orientation initiale

a) trajectoire avec interférences b) Interférences éliminées

Remarques

Remarque 1: Certaines boucles qui apparaissent au-delà de la première opération de décalage peuvent ne pas être des zones de collision au sens propre du terme mais leur parcours (usinage) constitue une perte de temps. C'est pour quoi cet algorithme constitue t - il un excellent moyen d'optimisation des trajectoires d'usinage.

Remarque 2: Les tests d'orientation exposés dans la section II-3 peuvent être utilisés. Le théorème de Green est un excellent outil pour faire face à ce problème où on peut remplacer

les courbes par leurs polygones caractéristiques. Il est également judicieux de procéder à cette simplification pour évaluer approximativement les périmètres des contours. Les procédures de calcul deviennent ainsi plus simples et plus rapides.

Remarque 3: Si, pour deux courbes données, plusieurs points d'intersection ont été détectés, alors il faut prendre en considération uniquement le point le plus proche de la première extrémité de la première courbe (la courbe C_i); les autres points vont être « redétectés » et traités de la même façon dans la prochaine itération.

Remarque 4: Dans certains cas, le balayage d'une nouvelle sous-poche nécessite la préparation d'un point de perçage à son point de départ afin de permettre à l'outil, censé d'effectuer un déplacement rapide vers ce point, de plonger sans problème dans la matière et de continuer le balayage. La simulation graphique des mouvements de l'outil permettra de supprimer certains de ces points soit manuellement, soit automatiquement en introduisant une distance au-delà de laquelle le programme prendra en considération les mouvements de dégagement - engagement.

II-7-5-2. Traitement des boucles intercontours

Les boucles intercontours résultent de l'existence d'un ou de plusieurs recouvrements entre deux contours. Ces contours peuvent être issus soit des décalages de la limite extérieure de la poche et d'un îlot (Fig. II-30e), soit des décalages de deux îlots (Fig. II-30g). Nous utiliserons, pour simplification, le terme poche pour désigner le premier contour et îlot pour désigner le deuxième. L'identification des boucles intercontours est réalisée suivant un principe analogue à celui employé pour calculer les boucles intracontours.

La différence entre les deux démarches réside au niveau des points suivants:

- Cette deuxième méthode utilise deux listes: la première, désignée par la lettre C, contient les courbes composant la poche et la deuxième, désignée par la lettre I, contient les courbes de l'îlot. Le nombre d'éléments dans la liste C est « n_c » tandis que le symbole « n_i » se réfère au nombre d'éléments dans la liste I.
- Le vecteur des branchements de la poche a été désigné par le symbole FE alors que celui de l'îlot utilise le symbole FI. Du fait qu'une courbe de la poche ou de l'îlot peut se brancher sur une autre courbe de l'un ou de l'autre des deux contours, nous avons introduit deux autres tableaux unidimensionnels HE et HI de valeurs 0 ou 1 et de dimensions identiques au nombre

de courbes respectivement dans la poche et dans l'îlot. HE et HI sont appelés vecteurs saut. Ce terme leur a été attribué car ils devront garder les traces des sauts effectués d'un contour à l'autre par n'importe quelle courbe de la poche ou de l'îlot.

Le rôle d'un élément HE_i sera de prendre la valeur 0 tant que l'élément C_i est branché sur une courbe de la poche et de prendre la valeur 1 dès que C_i sera branché sur une courbe appartenant à l'îlot. Un rôle analogue est attribué au tableau HI: un élément HI_i demeure égale à 0 tant que la courbe I_i est branchée sur une courbe de l'îlot; il prendra la valeur 1 une fois I_i sera branchée sur une courbe de la poche.

- La recherche des carrefours doit se faire à chaque fois entre une courbe de la poche et une autre de l'îlot.

L'algorithme est constitué des tâches suivantes:

(1) *Recherche des carrefours et subdivision*: On cherche un point carrefour entre les courbes de la poche et celles de l'îlot. Soit C_i et I_j deux courbes formant un carrefour dont la première appartient à la poche et la deuxième à l'îlot. On coupe les deux courbes en leur point d'intersection pour obtenir quatre courbes $C_{i,1}$, $C_{i,2}$, $I_{j,1}$ et $I_{j,2}$. On remplace C_i par $C_{i,1}$ et on insère $C_{i,2}$ entre C_i et C_{i+1} dans la séquence des courbes qui compose le contour extérieur. Même traitement doit se faire pour l'îlot: on remplace I_j par $I_{j,1}$ et on insère $I_{j,2}$ entre I_j et I_{j+1} . Le nombre de courbes pour chaque contour se trouve finalement incrémenté de 1.

Chaque opération de subdivision doit être suivie d'une mise à jour des vecteurs des branchements FE et FI et des vecteurs saut HE et HI. La procédure d'insertion est exactement identique à celle de l'algorithme d'évidement de poches sans îlots.

La routine de subdivision et d'établissement des branchements prendra la forme suivante:

Procedure Subdiv_Courbe2

```
{  
  // initialisation des vecteurs des branchements et des vecteurs saut  
  FOR i = 1 TO nc-1  
    { FEi = i+1; HEi = 0 }  
  FEnc = 1; HEnc = 0  
  
  FOR i = 1 TO ni-1  
    { FIi = i+1; HIi = 0 }  
  FIni = 1; HIni = 0
```

```

// fin d'initialisation
i = 1
REPEAT
  j = 1
  REPEAT
    B = Ci;    D = Ij
    // appel de la routine de recherche des carrefours
    IF Rec_Carrefour(B, D) = TRUE THEN
      { Subdiviser B pour obtenir deux courbes B1 et B2
        Ci = B1 // remplacement de Ci par B1 dans la liste C
        Insertion(C, B2, i)
        // l'appel de « Insertion » incrémente « ne » de 1
        Subdiviser D pour obtenir deux courbes D1 et D2
        Ij = D1 // remplacement de Ij par D1 dans la liste I
        Insertion(I, D2, j)
        // l'appel de « Insertion » incrémente « ni » de 1
        Reorientation2(FE, HE, nc, i, j) // mise à jour des vecteurs FE et HE
        Reorientation2(FI, HI, ni, i, j) // mise à jour des vecteurs FI et HI
      } // IF
    j = j + 1
  UNTIL j > ni
  i = i + 1
UNTIL i >= ne
} // END Procedure

```

(2) Construction des vecteurs des branchements

La détermination d'un carrefour et le découpage des courbes correspondantes engendrent l'établissement de nouveaux branchements pour les segments issus de cette opération. On branche C_{i,1} sur I_{j,2} et I_{j,1} sur C_{i,2} (Fig. II-41b), les branchements de C_{i,2} et I_{j,2} restent inchangés mais nécessitent une « mise à jour » du fait du changement d'indices établi.

Les branchements s'effectuent sur deux contours différents ce qui nécessite de distinguer les opérations faites dans le même contour de celles faites dans l'autre; une telle tâche est réalisée par les vecteurs saut HE et HI.

L'algorithme de construction des vecteurs des branchements peut s'énoncer ainsi:

Procedure Reorientation2(F, H, n, i, j)

```

{
  k = 1; r = 1
  REPEAT
    IF k < i THEN
      { OFk = Fr; OHk = Hr
        k = k + 1; r = r + 1
      }
    ELSE
      { OFk = Fr; OHk = Hr
        k = k + 1
      }
  UNTIL k > n

  FOR k = 1 TO n
    { Fk = OFk; Hk = OHk }

  FOR k = 1 TO n
    { IF k = i THEN
      { Fk = j + 1; Hk = 0 }
      ELSE IF (Fk > i AND Hk = 0) OR (Fk > j AND Hk = 1) THEN Fk = Fk + 1
    }
  } // End Procedure

```

(3) *Construction des sous-contours*: Cette étape ne peut être entamée qu'après épuisement des carrefours, subdivision des courbes et construction des branchements. C'est les séquences de courbes et les vecteurs des branchements de la phase finale uniquement qui peuvent être utilisés pour construire les sous-contours. Dans l'exemple de la Fig. II-39, la séquence finale de courbes de la poche se compose des éléments suivants $C = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$ avec un vecteur des branchements $FE = [6, 4, 4, 5, 2, 7, 1]$ et un vecteur saut $HE = [1, 1, 0, 0, 1, 1, 0]$; la séquence finale de courbes de l'îlot comporte les éléments suivants $\{I_1, I_2, I_3, I_4, I_5,$

I_6, I_7 }, le vecteur des branchements de l'îlot $FI = [6, 3, 3, 5, 2, 7, 1]$ et $HI = [1, 0, 1, 0, 1, 1, 0]$ (cf. tableau II-2).

La construction du premier sous-contour Q_1 débute par la courbe C_1 qui est le premier élément de la première séquence des courbes C (en réalité, on peut commencer dans ce cas par n'importe quel élément), le premier élément de FE est égal à 6 dont le saut correspondant est 1 (cf. tableau II-2) donc C_1 doit se brancher sur I_6 . I_6 doit se brancher sur C_7 car les valeurs correspondantes dans FI et HI sont respectivement 7 et 1. Le septième élément de FE est 1 et le septième élément de HE est 0 donc C_7 doit être branché sur C_1 qui est la courbe de départ. La boucle est ainsi fermée et le premier sous-contour Q_1 construit sera $Q_1 = \{C_1, I_6, C_7\}$.

Tableau II-2								
phase	i	j	poche			îlot		
			No. courbe	branchements		No. courbe	branchements	
				HE	FE		HI	FI
0			1	0	2	1	0	2
			2	0	3	2	0	3
			3	0	1	3	0	1
1	1	2	1	1	3	1	0	2
			2	0	3	2	1	2
			3	0	4	3	0	4
			4	0	1	4	0	1
2	1	4	1	1	5	1	0	2
			2	1	3	2	1	3
			3	0	4	3	0	4
			4	0	5	4	1	2
			5	0	1	5	0	1
3	5	1	1	1	6	1	1	6
			2	1	4	2	0	3
			3	0	4	3	1	3
			4	0	5	4	0	5
			5	1	2	5	1	2
			6	0	1	6	0	1
4	6	6	1	1	6	1	1	6
			2	1	4	2	0	3
			3	0	4	3	1	3
			4	0	5	4	0	5
			5	1	2	5	1	2
			6	1	7	6	1	7
			7	0	1	7	0	1

La construction du reste des sous-contours qui sont au nombre de 4 se fait exactement de la même manière à condition d'éliminer toute courbe utilisée. Ce rôle est confié à deux vecteurs bascules fe et fi (le premier pour la poche et le deuxième pour l'îlot) dont les éléments peuvent prendre des valeurs 1 ou 0. $fe_i = 0$ signifie que la courbe C_i n'est pas encore utilisée dans la

construction d'un sous-contour, tandis que $fe_i = 1$ signifie que la courbe C_i est déjà prise. Le vecteur fi joue exactement le même rôle dans le cas des courbes de l'îlot. On obtient donc $Q_2 = \{C_2, I_4, I_5\}$, $Q_3 = \{C_3, C_4, C_5, I_2, I_3\}$ et $Q_4 = \{C_6, I_7, I_1\}$.

On peut constater en comparant les résultats obtenus et la Fig. II-39e que seuls les contours les « plus extérieurs » sont construits tandis qu'un autre contour que nous désignons par exemple par Q^* et qui doit être normalement composé de $\{C_2, I^*_3, I^*_2, C_6, I^*_6\}$ ou de $\{C^*_2, I_6, C^*_6, I_2, I_3\}$ (où le signe * signifie courbe inversée) a été écarté. Q^* ne décrit pas actuellement une zone d'interférence mais son maintien est inutile. Il a été éliminé par le fait que l'algorithme n'autorise d'utiliser une courbe que dans un seul contour.

La routine de construction des sous contours en boucles intercontours sera donc la suivante:

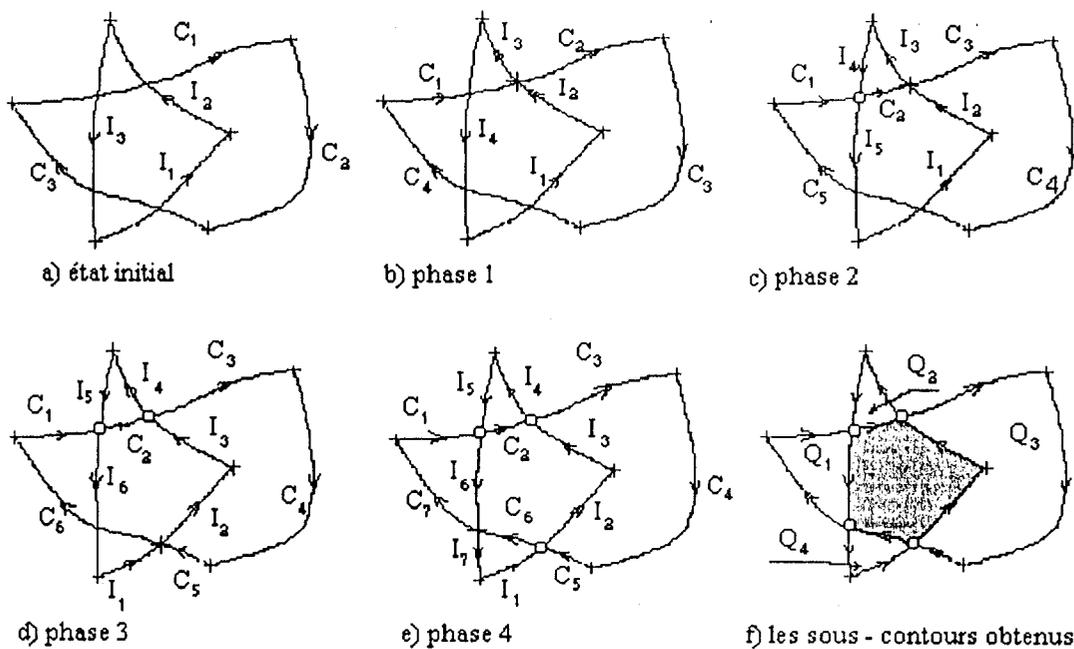


Fig. II-39 Exemple des étapes de construction des boucles intercontours

Procedure Sub_Contours2(C, I, Q)

{

 // Initialisation des tableaux fe et fi

 FOR $i = 1$ TO ne { $fe_i = 0$ }

```

FOR i = 1 TO ni { fi = 0 }
i = 0; j = 0
FOR k = 1 TO ne-1
{ IF fek = 0 THEN
{ i = i+1; j = 1
Qij = Ck; fek = 1; m = k; r = FEk
IF HEm = 1 THEN a = 1 ELSE a = 2;
IF HEk = 1 THEN g = fer ELSE g = fir;
WHILE g = 0
{ j = j+1
IF a = 1 THEN
{ Qij = Cr; FEr = 1; m = r; r = FEr
IF HEm = 1 THEN {a = 1; g = fer} ELSE {a = 2; g = fir}
}
ELSE { Qij = Ir; fir = 1; m = r; r = FIr
IF HIm = 1 THEN {a = 2; g = fir} ELSE {a = 1; g = fer}
} // ELSE
} // WHILE
ni = j // ni est le nombre de courbes dans le sous-contour Qi
} // IF fek = 0
} // FOR k
ns = i // ns est le nombre de sous-contours
} // END Procedure

```

(4) Test de sélection des sous-contours

La règle du critère de l'orientation initiale est valable dans le cas d'interférence poche - îlot. Le sens d'orientation de référence qu'on doit prendre dans ce cas est le sens horaire (sens d'orientation de la poche). En appliquant cette règle on retient ainsi, pour la Fig. II-39, les contours Q₁ et Q₃ et on élimine Q₂ et Q₄.

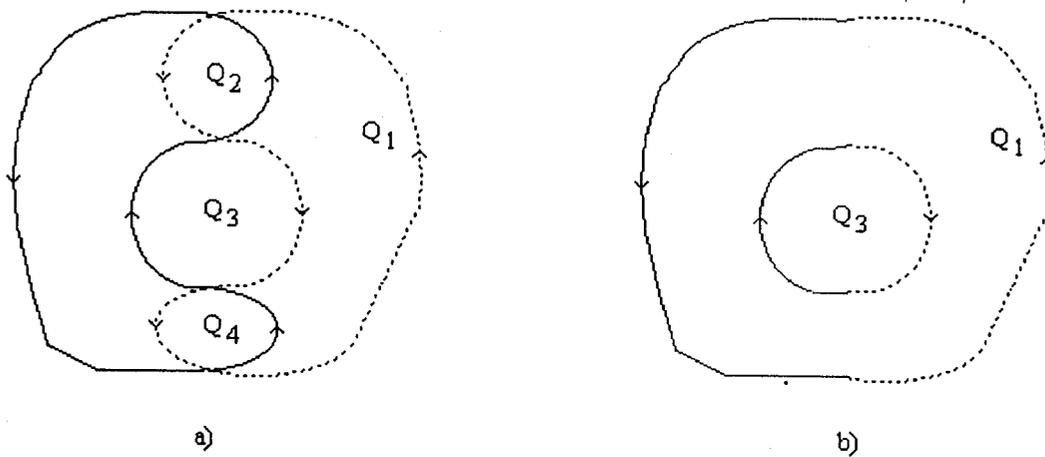


Fig. II-40 Interférence îlot - îlot avec plus de deux carrefours

Dans le cas d'interférence îlot - îlot avec uniquement deux carrefours (Fig. II-30g) on obtient deux sous-poches orientées dans le sens d'orientation des îlots (sens trigonométrique). Dans ce cas il faut retenir uniquement la plus grande sous-poche, i.e. celle qui possède le plus grand périmètre (Fig. II-30h).

Dans le cas d'une interférence îlot - îlot avec plus de deux carrefours, on doit retenir les sous-poches orientées dans le sens horaire et celle qui possède le plus grand périmètre parmi les sous-poches orientées dans le sens trigonométrique (Fig. II-40).

Algorithme général de traitement des boucles intercontours

Le programme général de calcul des boucles intercontours peut s'écrire:

```
{
  Subdiv_Courbes2(C, I)
  Sub_Contours2(C, I, Q)
  Select2(Q) // Test de sélection des contours Q à retenir
} // END
```

Remarque

L'ensemble des remarques concernant le traitement des boucles intracontours mentionnées dans (§II-7-5-1) est également valable pour les problèmes d'identification des boucles intercontours.

II-7-5-3. Discussion

Mis à part les problèmes soulevés dans les étapes de recherche des carrefours, les autres parties des deux algorithmes que nous avons développés ici ne doivent poser aucun problème de fiabilité du moins dans le contexte d'utilisation discutée dans ce document. La fiabilité de ces algorithmes, et par conséquent la qualité du traitement qui s'en suit, est conditionnée uniquement par un seul facteur: la méthode de recherche des intersections. En effet, toute défaillance au niveau de la détection des points d'intersection peut mettre en cause tout le travail déjà ou devant être effectué car elle provoquera l'exécution de zones de collision.

L'emploi de la méthode de subdivision - élimination pour faire face à ce problème est fort conseillé malgré sa lenteur et sa mise en oeuvre relativement délicate. Dans les cas pratiques que nous avons examinés, nous avons constaté que même avec deux courbes sans intersection et « suffisamment » éloignées cette méthode ne décèle ce résultat qu'après quelques itérations; ce qui signifie plusieurs opérations de subdivision inutiles (deux pour chaque itération). Si le processus de subdivision est quasi instantané pour les formes mathématiques usuelles et rapide pour les formes de Bézier, il est lent pour les B-splines et les NURBS. Les appels intensifs aux procédures de subdivision rendent les algorithmes de recherche des boucles très coûteux en temps machine.

La Fig. II-42 montre quelques cas particuliers d'intersections qui peuvent apparaître sur un contour. La considération ou non de ces points comme des carrefours est sans importance puisque le centre de l'outil, une fois arrivé sur l'un de ces points, ne provoque plus de collision en passant à la courbe suivante.



Fig. II-41 Subdivision et réorientation aux carrefours
 a) cas des boucles intracontours b) cas des boucles intercontours

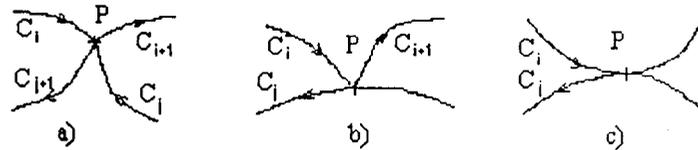


Fig. II-42 Configurations particulières d'intersection

Signalons aussi qu'il est utile de vérifier que le nombre total de carrefours détectés doit être toujours pair si on considère que les courbes *ne contiennent pas de points de rebroussement*. Si cette condition est insatisfaite, il sera clair que l'algorithme de recherche et de calcul de ces points est mis à défaut.

Notons finalement qu'il est judicieux de vérifier avant d'entamer la recherche des boucles intercontours que tous les points caractéristiques des éléments géométriques de l'îlot (les points caractéristiques pour un segment de droite sont ses extrémités et pour une courbe ses pôles) ne soient pas tous situés uniquement à l'intérieur du profil extérieur. Si tous ces éléments sont à l'intérieur du profil extérieur alors il n'y a aucun recouvrement et par conséquent aucune possibilité de formation de boucles (en réalité cette remarque n'est pas valable que pour les profils qui possèdent des polygones caractéristiques très proches de leurs courbes associées comme ceux calculés par la méthode de Tiller et Hanson (cf. §II-4-3-3)). Le théorème de Jordan (cf. §II-3-3) est un excellent moyen pour savoir si un point est à l'intérieur ou à l'extérieur d'un contour fermé.

II-7-6. Les tests d'arrêt

La stratégie de décalage que nous avons adoptée pour le balayage des poches, y compris celles qui possèdent un ou plusieurs îlots, doit aboutir après un certain nombre d'itérations au cas particulier de poches sans îlots. Or le décalage répété vers l'intérieur du contour d'une poche non « dégénérative » provoque son rétrécissement constant pour tendre, au fur et à mesure de l'augmentation du nombre d'itérations, vers une forme ponctuelle (donc de surface nulle). Ainsi l'opération de décalage devra être arrêtée une fois cet état est atteint.

Toutefois et pour des distances de décalage plus ou moins importantes par rapport aux dimensions totales de la poche, cette convergence vers l'intérieur n'aura lieu que momentanément pour diverger ensuite vers l'extérieur et aboutir à des formes inattendues (cf. Fig. II-43 et II-44). A ce point le décalage ne s'arrêtera qu'après avoir bloqué le système. Déterminer d'autres critères d'arrêt plus fiables s'avère nécessaire.

Deux méthodes peuvent être utilisées simultanément. La première consiste à arrêter le décalage une fois le contour change son sens d'orientation comme le montre la figure II-43. La deuxième méthode consiste à arrêter le décalage si toutes les extrémités des éléments géométriques constitutifs du contour offset sont situées à l'extérieur du contour générateur (Fig. II-44).

Les procédés de calcul de l'orientation des contours fermés et la situation d'un point par rapport à un contour fermé ont été l'objet de discussions antérieures (cf. §II-3).

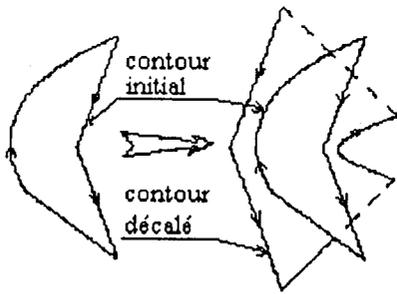


Fig. II-43 Changement du sens d'orientation du contour après décalage

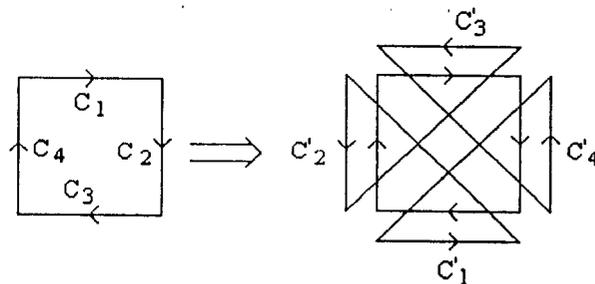


Fig. II-44 Le contour offset devient à l'extérieur du contour initial

II-7-7. Organisation des phases d'usinage

Le décalage répété du contour de la poche (et celui des îlots s'ils existent) conduit souvent à la dégénérescence de sous poches qui dégènèrent à leur tour en d'autres sous-poches et ainsi de suite; on obtient ainsi une arborescence qui est a priori de structure aléatoire. Cette structuration peut être utilisée telle qu'elle est pour établir un ordre de balayage des sous-poches élémentaires [Suh 90]. Pendant l'usinage, le parcours de cette arborescence devra se faire dans le sens inverse au sens de son développement. Cette structure arborescente pourra être améliorée, en vu d'une plus grande simplicité informatique, en la transformant en un arbre binaire [Mart 89, Tre 83].

Pour notre part, nous avons choisi de garder une seule liste de poches et d'écarter l'emploi des arbres. Les poches formées en fur et à mesure de l'avancement du décalage vers l'intérieur sont insérées dans cette même liste. Si une poche de la liste vérifie la condition d'arrêt, elle doit être retirée. Le programme de balayage s'arrête une fois tous les éléments de la liste sont retirés (liste vide).

II-7-7-1. Evidement de poches sans îlots

Pour traiter ce cas particulier nous avons conçu un algorithme que nous avons appelé Evid_PSI (pour Evidement de Poches Sans Ilots). Il sera la base des algorithmes de balayage de poches à un ou plusieurs îlots que nous développerons par la suite.

La procédure Evid_PSI utilisera entre autres la routine de recherche des boucles intracontours et une autre routine appelée ici OFFSET. Cette dernière devra permettre de calculer le contour offset avec la précision demandée pour tout contour initial introduit. Evid_PSI utilise comme donnée le contour fermé C qui initialise une liste L de contours dans laquelle vont s'insérer les éventuels sous-contours détectés par la routine de recherche des boucles intracontours. L'algorithme est ainsi un processus récursif; il doit s'arrêter lorsque la liste des contours L sera vide.

Voici maintenant les étapes de cet algorithme.

- (1) $L_1 = C$; // initialisation du premier élément de la liste L
 // C est le contour de la poche à balayer
- (2) $i = 1$ // initialisation du nombre de contours dans la liste L

- (3) $L_1 = \text{OFFSET}(L_1)$; // décalage du contour
- (4) Elimination de toutes les interférences sur L_1 sauf les boucles intracontours;
- (5) Faire le test d'arrêt pour L_1 ;
 si la condition d'arrêt est vérifiée et $i = 1$ alors aller vers l'étape (9);
 si la condition d'arrêt est vérifiée mais $i > 1$ alors supprimer L_1 de la liste L et aller vers l'étape (3);
- Notons que cette opération de suppression engendre le décalage de tous les éléments de L vers le bas et la décrémentation de i d'une valeur de 1 ($i = i - 1$);
- (6) Identifier les boucles intracontours;
 s'il existe des sous-contours Q_k , $k = 1, \dots, p$ alors affecter Q_1 dans L_1 tel que $L_1 = Q_1$ et insérer les autres contours Q_k , $k = 2, \dots, p$ dans la liste L ;
- (7) Discrétiser le contour L_1 pour obtenir les positions du point piloté outil et sauvegarder les résultats dans un fichier;
- (8) Retourner à l'étape (3);
- (9) Faire parcourir le fichier créé précédemment dans le sens inverse pour obtenir l'enchaînement final des positions successives de l'outil;
- (10) Fin

La figure II-45 montre cet algorithme sous forme d'un organigramme.

Les figures II-46 et II-47 montrent quelques exemples d'évidement de poches sans îlots réalisés par la procédure Evid_PSI. Ces poches sont constituées par des segments de droite, des arcs de cercle et des courbes B-splines de différents degrés. Les segments et les arcs sont convertis en formes B-spline avant d'être traitées.

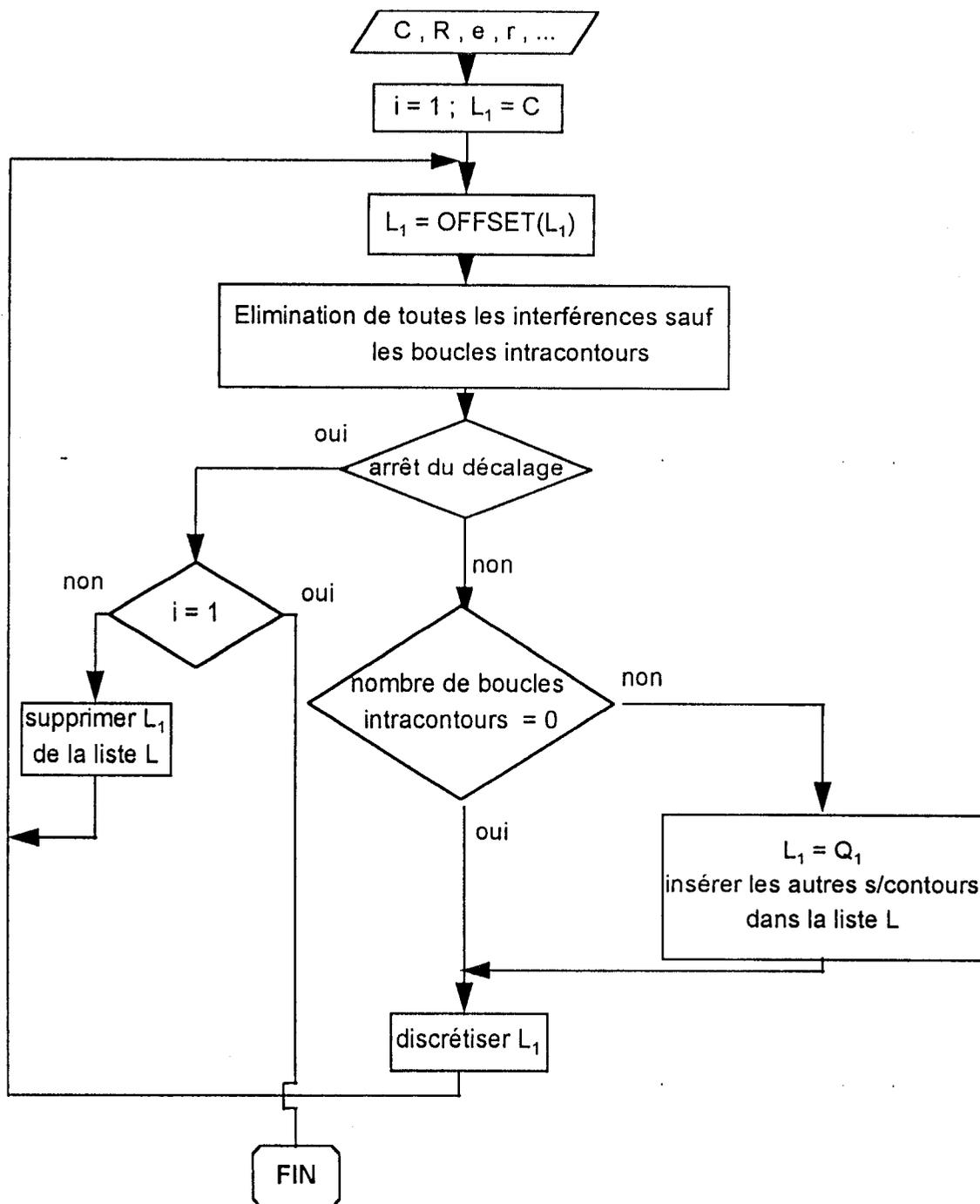


Fig. II-45 Organigramme de la procédure d'évidement de poches sans îlots

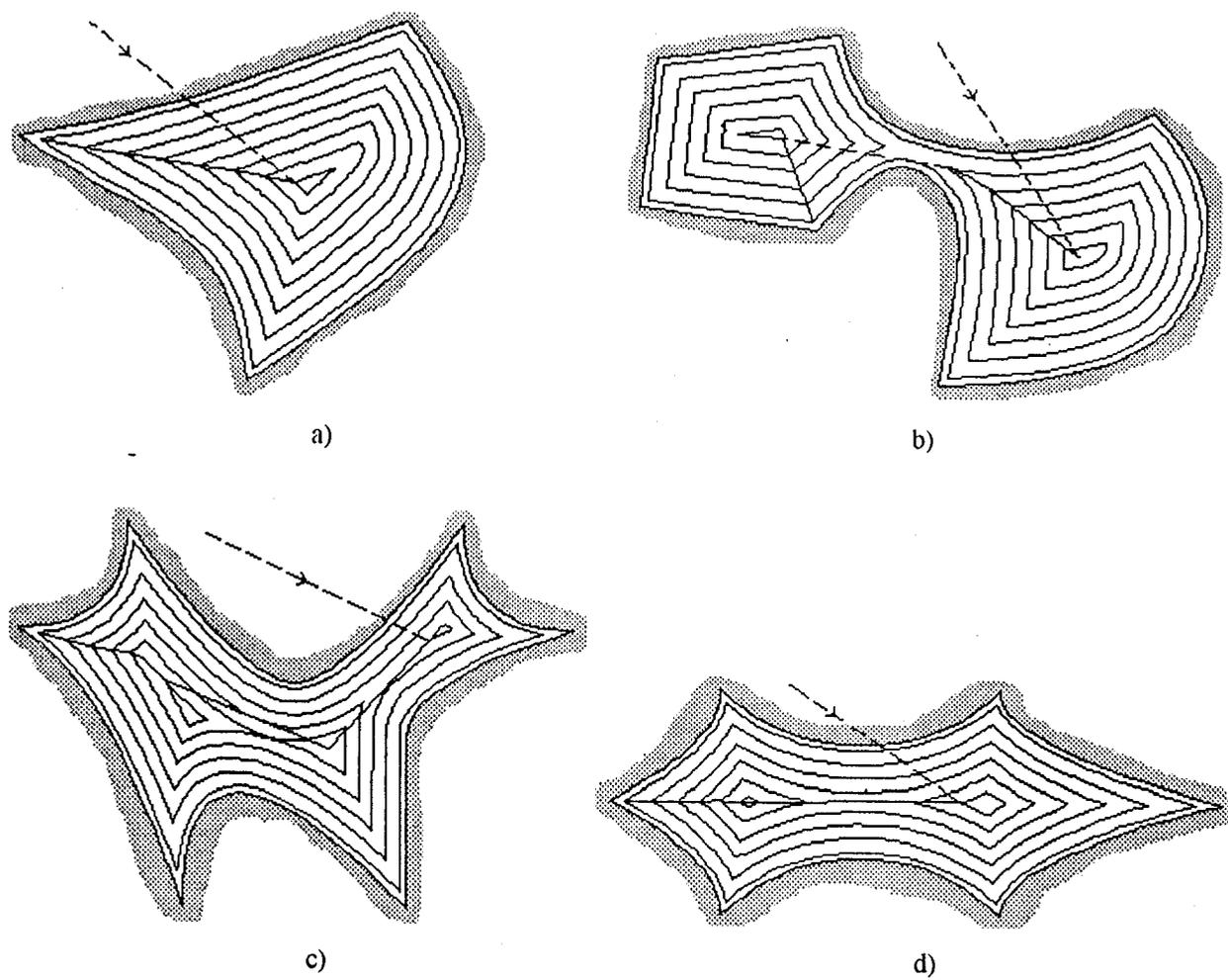


Fig. II-46 Exemples de balayage des poches planes sans îlots (contours à courbes gauches et segments de droite)

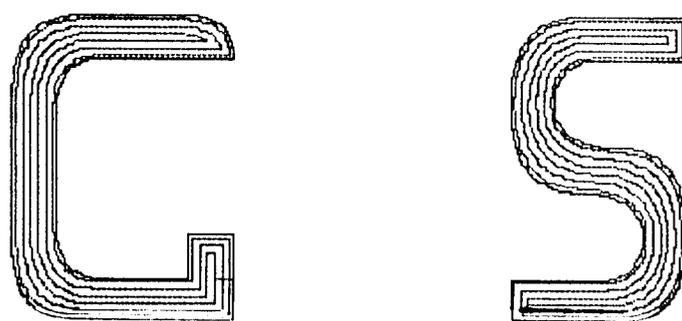


Fig. II-47 Gravure des lettres G et S modélisées par des courbes B-splines quadratiques

II-7-7-2. Evidement de poches à un seul îlot

Le deuxième algorithme que nous proposons réalise le balayage des poches avec un seul îlot. Il utilise l'algorithme Evid_PSI comme un sous-programme pour exécuter les phases qui ne concernent que les poches sans îlots: type qui apparaît dès la formation des premières boucles intracontours ou intercontours. Il faut noter que l'apparition de boucles intercontours signifie la disparition de l'îlot (Fig. II-30e et II-30g) et le travail deviendra ainsi une « simple » opération de vidage de poches sans îlots identique à ce que traite l'algorithme Evid_PSI. Nous profiterons de cette propriété pour concevoir un deuxième test d'arrêt (ce test ne provoque pas en réalité l'arrêt du balayage mais il passe la tâche au sous-programme Evid_PSI).

L'algorithme de balayage des poches à un seul îlot utilise un enregistrement C pour décrire le profil de la poche et un enregistrement I pour décrire l'îlot.

Le profil de l'îlot ne sera décalé qu'une seule fois durant tout le processus; c'est celui de la poche qui doit subir les décalages successifs. Une telle technique nous permettra de gagner un temps machine considérable.

Comme pour le premier algorithme, les profils décalés seront à chaque fois surchargés « overwriting » sur les profils générateurs afin d'économiser de l'espace mémoire.

Voici les étapes de cet algorithme.

- (1) $I = \text{OFFSET}(I)$; // décalage de l'îlot
- (2) Elimination des interférences (sauf boucles intracontours);
- (3) Recherche des boucles intracontours;
si ce type de boucles existe alors:
 - (b) Affecter la grande boucle dans l'enregistrement I;
 - (a) Utiliser le sous-programme Evid_PSI pour traiter les autres contours orientés dans le sens horaire ;
- (4) $C = \text{OFFSET}(C)$; // décalage du contour C;
- (5) Recherche des boucles intracontours dans C;
si ce type de boucles existe alors:

- (a) traiter les contours qui ne contiennent pas l'îlot I comme des poches sans îlot en utilisant le sous-programme Evid_PSI;
- (b) affecter le contour qui contient l'îlot dans C;
- (6) Recherche des boucles intercontours entre C et I;
- si ce type de boucles existe alors:
 - (a) traiter chaque sous-contour formé comme une poche sans îlot en utilisant le sous programme Evid_PSI;
 - (b) Aller à l'étape (9);
- (7) Discrétisation du contour C pour l'obtention des positions du point piloté outil et sauvegarde des résultats dans le fichier approprié;
- (8) Retour à l'étape (4);
- (9) Parcours du fichier crée précédemment dans le sens inverse pour l'obtention de l'enchaînement final des positions successives du point piloté de l'outil;
- (10) Fin.

La figure II-48 montre quelques exemples d'application de cet algorithme pour l'évidement de poches à un seul îlot.

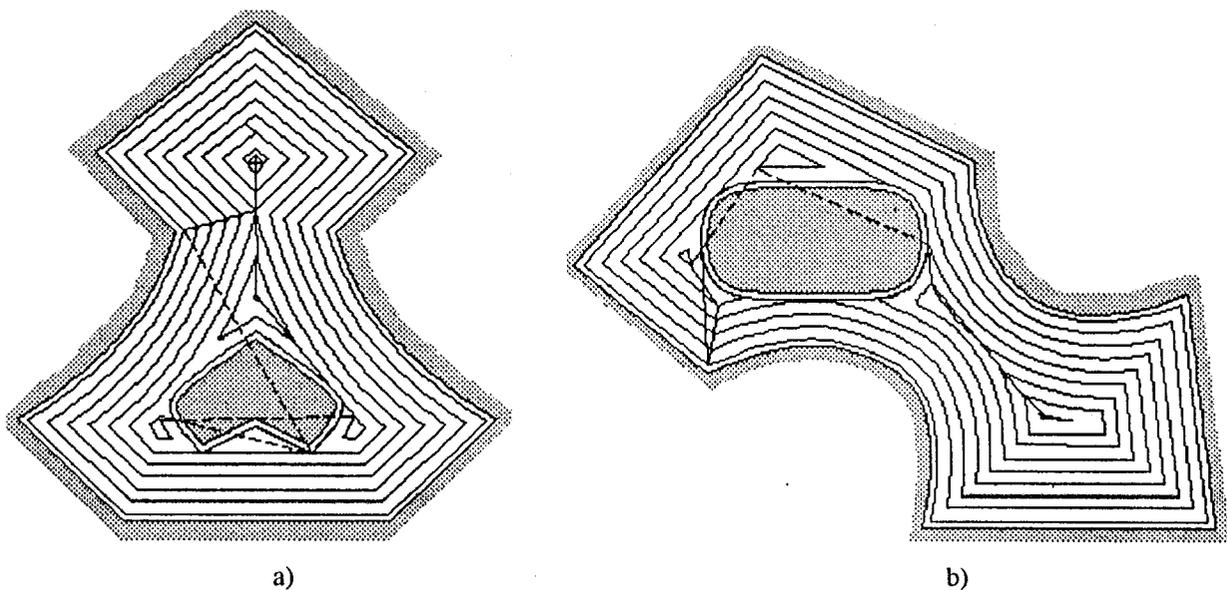


Fig. II-48 Balayage de poches avec un seul îlot

Remarques

- Pour réaliser l'étape 5a, i.e. savoir le contour qui contient l'îlot, on peut utiliser la solution proposée dans la section II-7-5-2 qui est basée sur le théorème de Jordan.

- Une autre approche de balayage des poches à un seul îlot peut être envisagée. Elle consiste à réitérer le décalage de l'îlot et de ne décaler, en revanche, le contour extérieur de la poche qu'une seule fois. Si on peut supposer que les îlots qu'on utilise sont convexes on peut alors envisager d'éliminer carrément le code de recherche des boucles intracontours sur l'îlot; l'algorithme deviendra plus simple et plus rapide.

Du point de vue conditions de coupe en usinage, la solution qui consiste à décaler la poche est meilleure car elle permet de s'approcher du centre de la pièce et de minimiser par conséquent le risque de son éventuel déséquilibre.

II-7-7-3. Généralisation à l'évidement de poches avec plusieurs îlots

Nous aurons besoin pour définir une poche avec plusieurs îlots d'un enregistrement C qui contient la description du contour extérieur et d'une liste L d'enregistrements de même type que C pour décrire les îlots. Pour faciliter l'exposé et sans restreindre la généralité de l'algorithme nous supposons dans cette section que le décalage de la poche et ses îlots ne donne pas lieu à l'apparition de boucles intracontours.

La méthode que nous développerons dans la suite repose sur deux remarques importantes:

* le recouvrement de deux îlots conduit à leur fusion et à leur remplacement par un nouvel îlot qui les englobe tous les deux (cf. Fig. II-30h);

** le recouvrement d'une poche et d'un îlot provoque la disparition de l'îlot (cf. Fig. II-30f);

Si on choisit un îlot parmi les « n » îlots de la poche et on le décale plusieurs fois vers l'extérieur on s'aperçoit de l'effet de « gonflement » qu'il subit. Un tel phénomène provoquera tôt ou tard soit un recouvrement îlot / îlot soit un recouvrement îlot / poche (ou éventuellement les deux en même temps). Pour l'un ou l'autre, l'îlot en question disparaîtra. En réitérant la procédure on éliminera tous les îlots pour se trouver en fin

de compte uniquement avec des sous poches sans îlots qu'on peut balayer sans peine en utilisant le sous-programme Evid_PSI discuté auparavant.

La méthode complète de vidage des poches avec plusieurs îlots peut être résumée dans les étapes suivantes:

- (1) Calcul des contours offset du contour extérieur et des îlots;
- (2) recherche des recouvrements entre l'îlot L_1 et les autres îlots;
s'il existe un recouvrement entre L_1 et un autre îlot L_j alors:
 - (a) fusionner les deux îlots L_1 et L_j ;
 - (b) éliminer l'îlot L_j de la liste des îlots L : il y aura donc un décalage vers le bas de tous les éléments de L ;
 - (c) remplacer L_1 par le nouvel îlot formé par la fusion effectuée en (a);
sinon continuer la vérification entre L_1 et le reste des îlots et reprendre les étapes a, b et c ci-dessus en cas de détection de boucles intercontours;
- (3) Recherche de l'existence d'un recouvrement entre L_1 et le contour extérieur de la poche;
s'il existe un recouvrement alors:
 - (a) calculer les sous-poches;
 - (b) éliminer l'îlot L_1 de la liste L ;
 - (c) si le nombre d'éléments de L est nul alors:
 - (a1) traiter les sous-poches obtenues comme des poches sans îlots en utilisant le sous-programme Evid_PSI;
 - (b1) aller à l'étape (5);
 - sinon (i.e. la liste n'est pas encore vide) alors:
 - (a2) Chercher la sous-poche Q_i qui contient les îlots et balayer les autres sous-poches par le sous programme Evid_PSI;
 - (b2) Remplacer le contour extérieur C par Q_i tel que: $C = Q_i$;
 - (c2) Discrétiser l'îlot L_1 pour obtenir les positions du centre de la fraise et sauvegarder les résultats dans le fichier approprié;
 - (d2) Réitérer le décalage de l'îlot L_1 tel que: $L_1 = \text{OFFSET}(L_1)$;
- s'il n'existe pas de recouvrement entre L_1 et C alors

- (a) Discrétiser l'îlot L_1 pour obtenir les positions du centre de la fraise et sauvegarder les résultats dans le fichier approprié;
- (b) Répéter le décalage de l'îlot L_1 tel que: $L_1 = \text{OFFSET}(L_1)$;
- (4) Retourner à l'étape (2);
- (5) FIN.

Considéré tel qu'il est, le contour extérieur C ne fera l'objet de décalage (vers l'intérieur) qu'une seule fois; c'est l'îlot qui se trouve au début de la liste L qui subira les décalages successifs. Ce choix a été fait pour éliminer la recherche des boucles intracontours sur le contour extérieur.

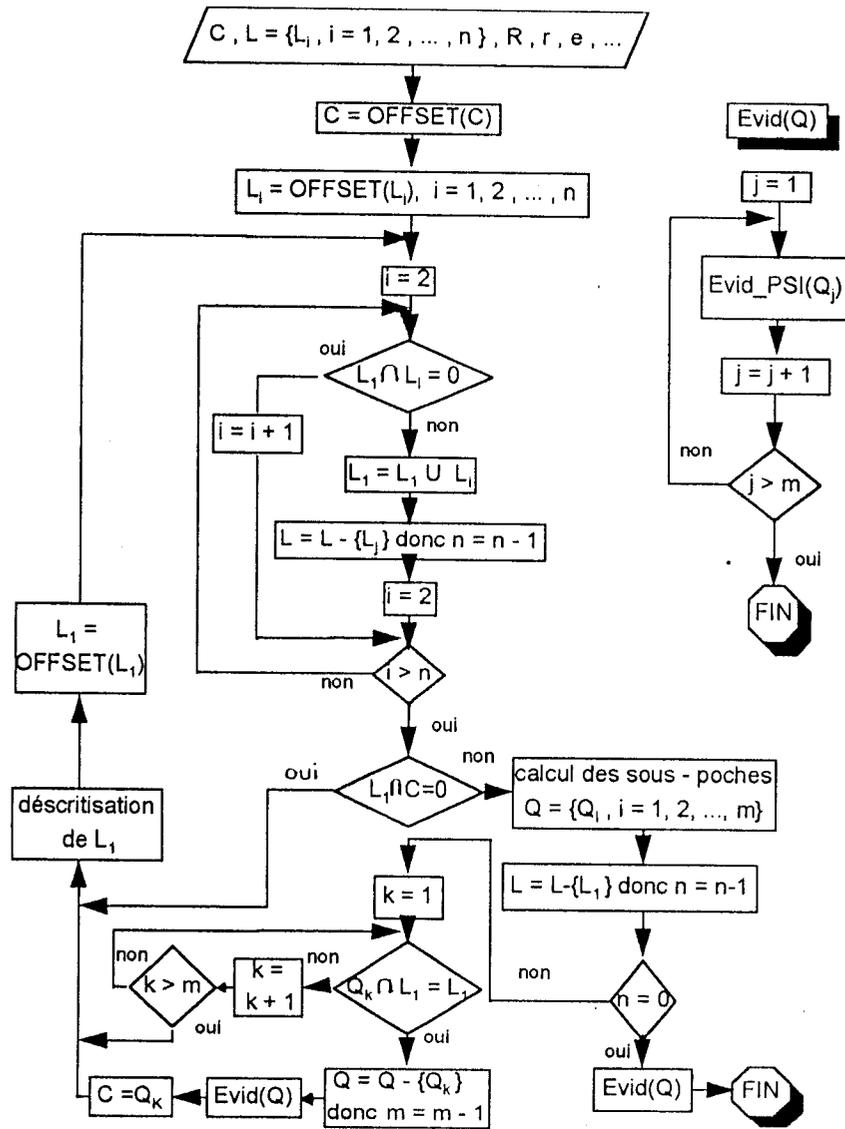


Fig. II-49 Organigramme du module d'évidement de poches avec plusieurs îlots

La figure II-49 montre cet algorithme interprété sous forme d'un organigramme.

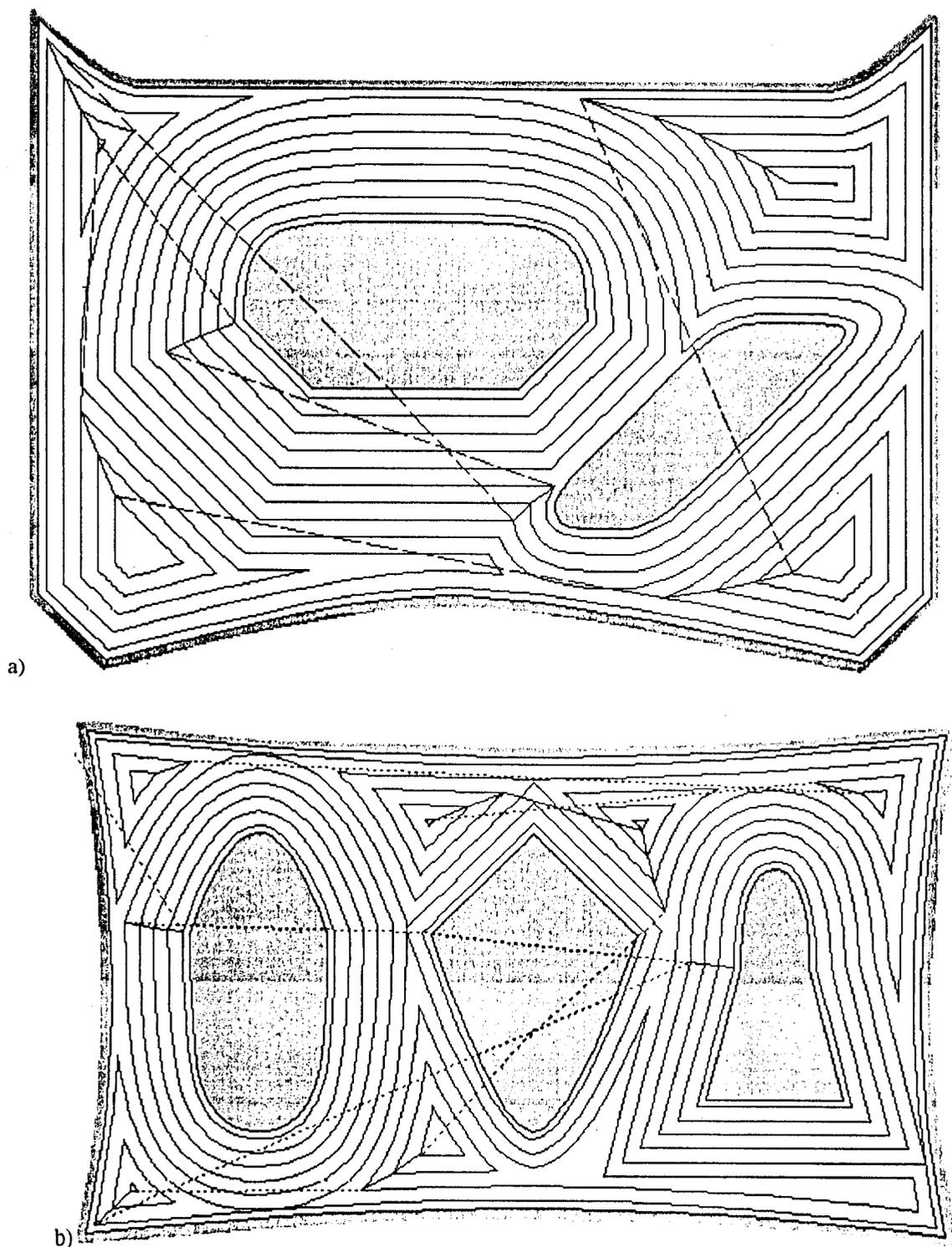


Fig. II-50 Balayage de poches avec deux et trois îlots

Des exemples de démonstration d'évidement de poches avec plusieurs îlots sont donnés dans la figure II-50. Les chemins d'outil sont calculés par notre module informatique d'évidement de poches planes de géométrie et de nombre d'îlots quelconque « PMach ».

Remarque: Dans les exemples d'évidement de poches présentés ci-dessus, les lignes continues représentent les trajectoires d'outil en vitesse de travail et les lignes discontinues représentent les dégagements - déplacements - engagements d'outil en vitesse rapide.

II-8. Présentation du programme d'évidement de poches planes PMach

PMach (pour **P**ocket **M**achining) est un utilitaire informatique d'évidement de poches planes fonctionnant sur un micro-ordinateur de type PC ou compatible. Il utilise les solutions discutées dans ce présent mémoire. Il est composé des fonctions principales suivantes:

- Fonctions d'entrée - sortie permettant la lecture et la sauvegarde des formes géométriques de traitement. Les fichiers de saisie de contours de PMach sont de type texte.
- Fonction de conception de formes utilisant comme modèle de présentation les B-splines non uniformes.
- Fonction de contournage plan intérieur ou extérieur.
- Fonction de gravure en creux.
- Fonction de balayage de poches avec ou sans îlots.
- Fonction de simulation graphique pas à pas ou en continu des trajectoires d'outil lis à partir d'un fichier générique.
- Fonction de simulation de l'usinage à partir des données contenues dans un fichier de G-code en langage de programmation des MOCN EMCO F1 CNC.
- Un post-processeur pour les machines EMCO F1 CNC.

- Des possibilités de modification des paramètres d'usinage (diamètre d'outil, erreur de flèche, etc.) et les paramètres système (couleurs, etc.).

Les modules d'usinage et de génération de G-code de PMach ont été testés sur plusieurs exemples de formes complexes contenant à la fois des segments de droites, des arcs de cercle et des courbes à pôles de type B-splines non uniformes et Bézier. Tous ces éléments géométriques sont considérés comme des B-splines non uniformes afin d'unifier les routines de traitement. Les usinages effectués ont abouti à des résultats très satisfaisants bien que le programme a été écrit dans un but de démonstration de la faisabilité des solutions proposées.

La figure II-51 montre quelques menus graphiques proposés par PMach. Les figures II-46 .. 50 montrent des exemples de calcul de trajets d'outil effectués par PMach.

EMCO F1 CNC language						
N	G	X	Y	Z	F	
#!*						
N:	G:	X :	Y :	Z		
000	91					
001	00	00000	0000	010		
002	00	01334	3393			
003	01	00000	0000	-010		
004	01	00046	0033			
005	01	00048	0033			
006	01	00050	0033			
007	01	00052	0033			
008	01	00054	0032			
009	01	00055	0032			
010	01	00057	0032			
011	01	00059	0032			
012	01	00061	0032			
Ret = continue		Esc				

<p>Tool radius = 6.000 Cut speed [m/min] = 45 Tool speed [tr/min] = 154 Feed speed [mm/min] = 80 Overlap = 2.000 Overthink = 1.000 Mini length for G00 = 100.00 Chord deviation = 0.100</p>
--

<p>TOOL PATH EVALUATION</p> <hr/> <p>G00 distance = 36.97 G01 distance = 561.67 Distance ratio = 93.82 % G00 time = 0.09 G01 time = 7.02 Time ratio = 98.70 %</p>
--

<p>Design Save Load Contour Modify Contourning Poc. WO isl Poc. ++ isl Engrave Simulation Path check Cut paramet Post-Pro G-code simu Syst param Cls/Redraw CL scr/mem Quit</p>

Fig. II-51 Menu principal, paramètres de coupe, évaluation quantitative et traçage (en arrière plan) des trajectoires d'outil et édition d'un fichier ISO réalisés par le programme PMach

« Un développement de logiciel réussi est une activité à long terme »

B. STROUSTRUP

CHAPITRE III

APPLICATIONS

III. APPLICATIONS

Dans ce chapitre nous présenterons deux applications de l'évidement des poches planes par fraisage.

La première application concerne l'usinage des surfaces gauches par la technique des plans de niveau. Le balayage de poches dans ce cas ne doit concerner que les opérations d'ébauche. De telles opérations ne diffèrent pas du mode d'évidement des poches planes discuté dans le chapitre II. Les machines réalisant ce travail peuvent être des MOCN de 2 ½ axes uniquement. Par contre la finition doit se faire avec une MOCN d'au moins 3 axes. Les opérations d'ébauche peuvent être réalisées par une fraise cylindrique à bout plat. Les passes de finition sont généralement réalisées par une fraise à bout hémisphérique (fraise boule).

La deuxième application concerne l'usinage des poches complexes sur des surfaces gauches. Les machines nécessaires pour ces usinages doivent être dotées au minimum de trois axes. On utilise généralement les fraises toriques pour l'ébauche et des fraises boules (à bout hémisphérique) pour la semi - finition et la finition. Notre étude a été limitée au cas des poches de fond parallèle à la surface de projection et à la génération des trajectoires d'outil pour des MOCN 3 axes.

III-1. Usinage des surfaces gauches par la technique des plans de niveau

L'usinage par fraisage des surfaces gauches dont la forme brute est obtenue par moulage ou forgeage se fait généralement par la génération d'une surface enveloppe dite surface offset [Far 86] décrivant les positions du point piloté de l'outil [Cho 88, Cho 89]. La surface offset est souvent générée suivant l'un des isoparamètres u ou v de la surface $P(u,v)$ (Fig. III-1) [Cho 88, Hwa 92, Kim 88].

Dans plusieurs cas on se trouve dans l'obligation de partir d'un bloc de métal (généralement un parallélépipède) pour creuser la forme à obtenir dans la masse. La première solution possible consiste à utiliser la technique de balayage isoparamétrique en trichant sur le correcteur de la longueur de l'outil (déclarer un correcteur supérieur à la dimension réelle) ou sur les surépaisseurs [Piq 89]. Cette solution conduit à une perte de temps car, au cours des premières passes, l'outil est longtemps hors de la matière. Les distances offset importantes risquent de créer des auto-intersections sur les trajectoires [Aom 90] même pour des surfaces à faibles replis.

La méthode la plus employée actuellement pour l'ébauche automatique de ce type d'usinage consiste à faire des ébauches dans des plans perpendiculaires à l'axe de l'outil avec une altitude z dégressive [Lee 92]. On cherche l'intersection de la surface finie avec chacun des plans de travail; on en déduit des poches planes avec ou sans îlots en connaissant le coté de la matière dans chaque plan de travail. L'outil utilisé est généralement une fraise cylindrique à extrémité plate « deux tailles » ou torique [Piq 89, Lee 92]. Les formes que donne cette technique sont de type escalier (Fig. III-2a). Par conséquent, elles doivent être lissées par une ou plusieurs passes en utilisant la méthode d'usinage par balayage isoparamétrique (Fig. III-2b) moyennant une fraise boule (fraise à extrémité hémisphérique).

III-1-1. Intersection surface gauche / plan

Les techniques de calcul des intersections surface / surface [Pen 84, Sac 88] sont utilisées pour déterminer les poches des différents plans d'usinage.

Le principe de la méthode de subdivision - élimination, traitée dans la section II-6-2 pour le cas des courbes, est un excellent moyen pour résoudre ce problème [Pen 84] où dans ce cas les polyèdres convexes jouent le rôle des polygones convexes et les rectangles deviennent des boîtes « minimax » [Mar 88, Rat 93, Yen 91].

La méthode de Newton - Raphson (cf. §II-6-1) peut également être utilisée; si on prend le plan de travail perpendiculaire à l'axe z on peut définir cette intersection par l'expression suivante:

$$z_k - z(u, v) = z_k - \sum_{i=0}^m \sum_{j=0}^n Q_{ij}(u, v) z_{ij} = 0 \quad (\text{III-1})$$

où Q_{ij} définit l'expression des fonctions de base de la surface $P(u, v)$;

z_{ij} , $i \in \{0, 1, \dots, m\}$, $j \in \{0, 1, \dots, n\}$, sont les cotes z des pôles du réseau caractéristique de la surface;

z_k est la cote k sur l'axe Z qui peut être évaluée, pour une hauteur de coupe h constante, par l'expression

$$z_k = z_0 - k \cdot h \quad (\text{III-2})$$

où z_0 est la hauteur initiale.

L'équation (III-1) représente l'équation implicite de la courbe d'intersection entre la surface $P(u,v)$ et le plan $Z = z_k$; elle est de type non linéaire résolue par rapport à l'un parmi les deux paramètres u et v [Leo 91, Qiu 87]. La question de la recherche des solutions initiales pour cette équation est discutée en détail dans [Qiu 87].

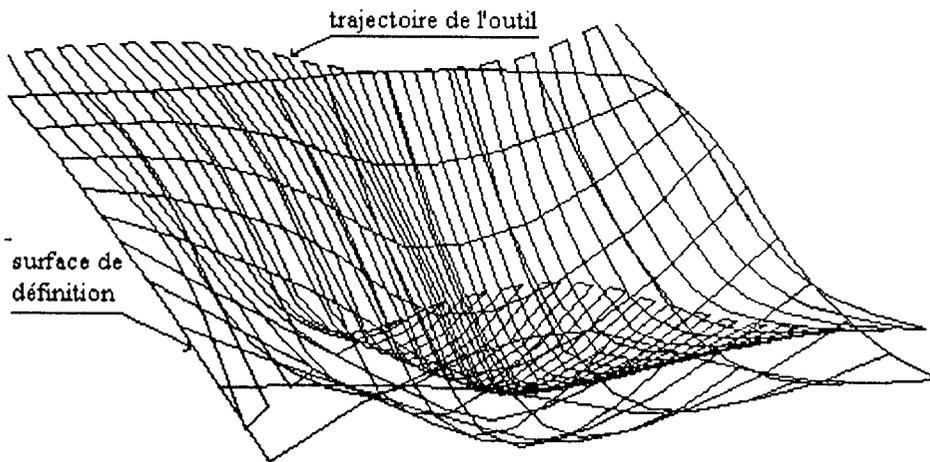


Fig. III-1 Usinage par balayage isoparamétrique

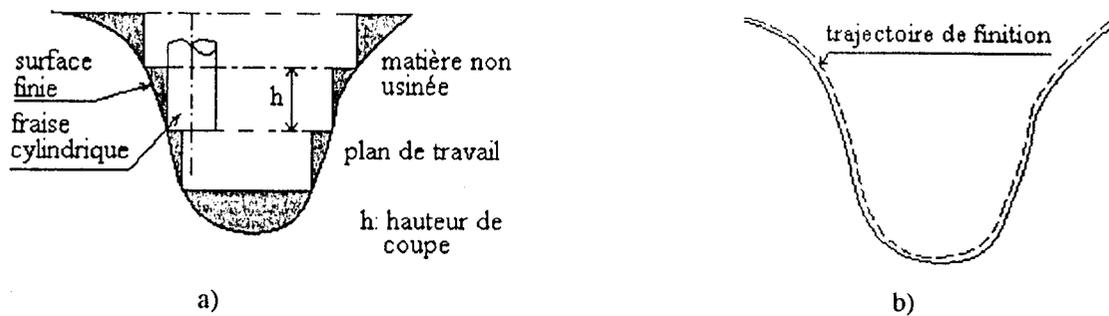


Fig. III-2 Usinage par plans de niveau: a) ébauche b) finition

III-1-2. Approximation des contours d'intersection

L'étape ci-dessus ne permet d'obtenir qu'un ensemble discret de points dans le plan qu'on devrait lier pour former des contours fermés. On peut envisager d'utiliser des méthodes de polygonisation [Lee 92], d'approximation par des segments de droite et arcs de cercle [Cra 85, Rob 85, Mor 91] ou d'approximation par des courbes paramétriques [Pie 86, Rog 89b, Mic 92, Fau 87, Mee 92, Leo 91].

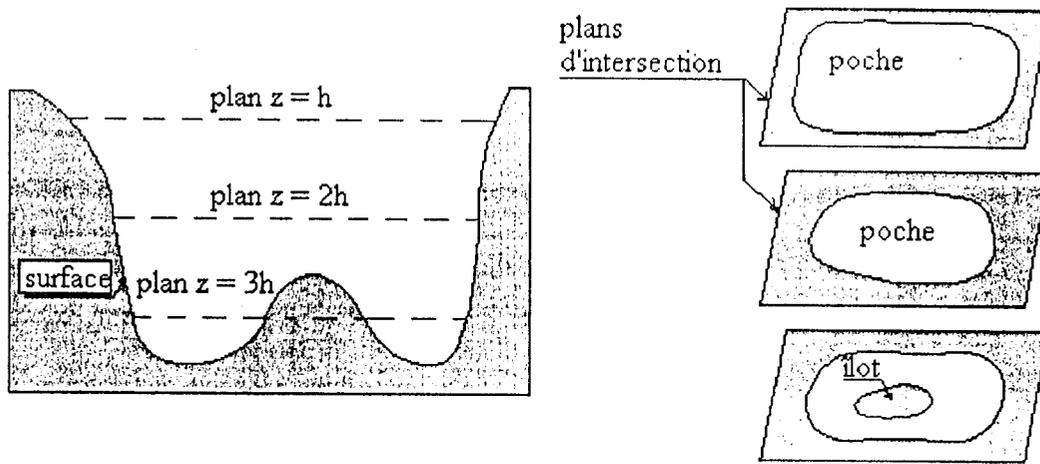


Fig. III-3 Poches des intersections surface / plans

III-1-3. Evidement des poches

En réalité, cette phase est composée de trois sous - phases:

- (a) Définition du coté de la matière à usiner en attribuant aux contours les orientations adéquates (cf. §II-3);
- (b) calcul de la distance de balayage (le décalage d);
- (c) balayage de la poche.

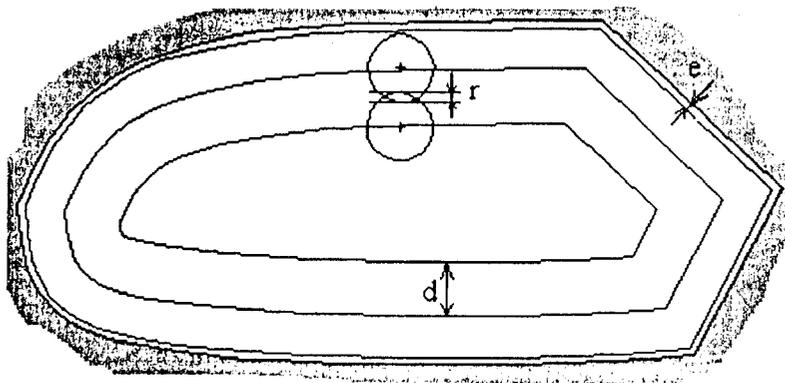


Fig. III-4 Calcul de l'incrément de prise de passe dans le plan

Les sous phases (a) et (c) sont déjà décrites en détail dans le chapitre II; quant à la sous phase (b), le calcul de la distance d de décalage doit tenir compte de deux paramètres: la surépaisseur e qu'on doit laisser pour la passe de finition et le recouvrement r (Fig. III-4). La distance de décalage d pour la première trajectoire est:

$$d = R + e \quad (\text{III-3})$$

et pour les trajectoires suivantes

$$d = 2R - r \quad (\text{III-4})$$

où R représente le rayon de la fraise.

III-1-4. Finition

Dans cette phase l'outil va parcourir une succession de segments linéaires dans l'espace. Comme le montre la figure III-5, le point $P = P(u,v)$ qui est le point de contact de l'outil avec la surface de définition s'appelle CC point (Cutter Contact point). La densité des CC points est proportionnelle à l'erreur de flèche et à la hauteur de crête admissibles exigées par l'utilisateur (cf. §III-2-2 et §III-2-4). Si le vecteur n représente la normale unitaire de la surface au point P , alors le couple (P, n) s'appelle le CC data (Cutter Contact data). Le centre P' de l'hémisphère de l'extrémité de l'outil s'appelle le point offset, il est donné par l'expression:

$$P' = P + R n \quad (\text{III-5})$$

où

$$n = \frac{\frac{\partial P}{\partial u}(u,v) \wedge \frac{\partial P}{\partial v}(u,v)}{\left\| \frac{\partial P}{\partial u}(u,v) \wedge \frac{\partial P}{\partial v}(u,v) \right\|} \quad (\text{III-6})$$

Supposons que l'axe de rotation de l'outil est représenté par le vecteur unitaire u ($u=[0,0,1]^T$ pour une MOCN 3 axes), alors le point B, appelé point piloté outil ou Cutter Référence point, sera exprimé par:

$$B = P' - R u = P + R (n - u) \quad (\text{III-7})$$

et le couple (B, u) s'appelle CL data (Cutter Location data).

Certains directeurs de commande ne demandent que les CC data pour faire les corrections d'outil nécessaires (comme le NUM 760F en mode G29) alors que d'autres nécessitent l'introduction des CL data.

Notons enfin que les directeurs de commande des MOCN ne savent pas gérer les interférences dans le cas des surfaces gauches; c'est aux systèmes de CFAO de traiter de tels

problèmes, plusieurs techniques sont proposées [Cho 88, Cho 89, Tan 90, Aom 90, Hwa 92, Kur 92, Yua 91]. Si la surface présente des zones non accessibles, la solution consiste à changer l'outil pour un autre de diamètre plus petit ou éventuellement à utiliser une machine à quatre ou cinq axes [Krz 87, Cho 93].

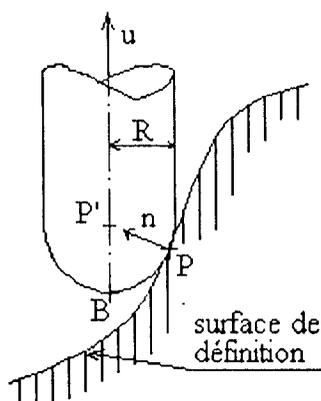


Fig. III-5 CC point et CL point

III-2. Usinage des poches sur les surfaces gauches

L'usinage des cavités et des gravures (notamment les enseignes de fabricants) est un problème courant dans la réalisation des moules et des outillages de fabrication. Or ces deux problèmes peuvent être traités parfaitement comme des balayages de poches. Si la surface de l'objet contenant ces formes est non plane alors le problème devient très complexe et difficile à résoudre sauf pour des cas très simples.

Bien qu'il existe des tentatives limitées de calcul des trajectoires d'outil directement sur les surfaces [Pat 89], l'approche qui commence à s'imposer aujourd'hui dans les systèmes de CFAO (le cas du logiciel SurfCAM de Surfware par exemple) comporte trois étapes:

- (a) aplatissement de la poche moyennant une projection orthogonale ou tout autre algorithme;
- (b) génération des chemins d'outil dans le plan;
- (c) projection des trajets sur la surface de l'usinage.

Cette procédure d'évidement de poches non planes convient parfaitement aux opérations d'usinage de surfaces à faibles courbures en phase de semi finition et de finition (Fig. III-10).

Notre programme FFMach offre une procédure de mise en oeuvre de cette technique pour l'usinage des surfaces à pôles sur des machines 3 axes.

Sauf pour les surfaces à très faible courbure la finition par une fraise boule exige de mener conjointement les étapes (b) et (c). En effet, l'emploi de cet outil révèle qu'entre celui ci et la surface à usiner, il n'existe généralement qu'un contact ponctuel (Fig. III-5 et III-8a) de sorte que l'outil creuse des sillons séparés par des arêtes appelées crêtes. La hauteur de ces sillons doit être réduite à une valeur au-dessous d'une limite appelée « hauteur de crête admissible » (Fig. III-8a). Cette limite doit faire l'objet d'une vérification avant chaque opération de décalage des profils de la poche.

En ébauche où le but consiste à enlever le maximum de matière il est souvent conseillé d'employer un outil torique. Le calcul de la hauteur de crête dans ce cas est peu utile. Nous pensons qu'un décalage égal au maximum au rayon moyen de l'outil est suffisant pour réaliser un compromis entre un temps de calcul et d'usinage réduit et un volume de copeaux optimal.

Deux autres sources d'erreur doivent être vérifiées. La première est l'erreur de discrétisation des courbes constitutives des trajets de l'outil (Fig. III-6). La seconde est l'erreur de projection des trajectoires sur la surface d'usinage (Fig. III-7).

Nous indiquerons dans les paragraphes suivants la manière de conduire les calculs de vérification de ces trois sources d'imprécision.

III-2-1. Calcul des intersections surface / droite

L'équation implicite générale d'une droite dans l'espace est définie par l'intersection de deux plans.

$$\begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases} \quad (\text{III-8})$$

Le point d'intersection de cette droite avec une surface biparamétrique $P(u,v) = (x(u,v) \ y(u,v) \ z(u,v))^T$ est obtenu en substituant l'expression:

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n Q_{ij}(u, v) P_i \quad (\text{III-9})$$

Notre programme FFMach offre une procédure de mise en oeuvre de cette technique pour l'usinage des surfaces à pôles sur des machines 3 axes.

Sauf pour les surfaces à très faible courbure la finition par une fraise boule exige de mener conjointement les étapes (b) et (c). En effet, l'emploi de cet outil révèle qu'entre celui-ci et la surface à usiner, il n'existe généralement qu'un contact ponctuel (Fig. III-5 et III-8a) de sorte que l'outil creuse des sillons séparés par des arêtes appelées crêtes. La hauteur de ces sillons doit être réduite à une valeur au-dessous d'une limite appelée « hauteur de crête admissible » (Fig. III-8a). Cette limite doit faire l'objet d'une vérification avant chaque opération de décalage des profils de la poche.

En ébauche où le but consiste à enlever le maximum de matière il est souvent conseillé d'employer un outil torique. Le calcul de la hauteur de crête dans ce cas est peu utile. Nous pensons qu'un décalage égal au maximum au rayon moyen de l'outil est suffisant pour réaliser un compromis entre un temps de calcul et d'usinage réduit et un volume de copeaux optimal.

Deux autres sources d'erreur doivent être vérifiées. La première est l'erreur de discrétisation des courbes constitutives des trajets de l'outil (Fig. III-6). La seconde est l'erreur de projection des trajectoires sur la surface d'usinage (Fig. III-7).

Nous indiquerons dans les paragraphes suivants la manière de conduire les calculs de vérification de ces trois sources d'imprécision.

III-2-1. Calcul des intersections surface / droite

L'équation implicite générale d'une droite dans l'espace est définie par l'intersection de deux plans.

$$\begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases} \quad (\text{III-8})$$

Le point d'intersection de cette droite avec une surface biparamétrique $P(u,v) = (x(u,v) \ y(u,v) \ z(u,v))^T$ est obtenu en substituant l'expression:

$$P(u,v) = \sum_{i=0}^m \sum_{j=0}^n Q_{ij}(u,v)P_i \quad (\text{III-9})$$

dans (III-8); on obtient ainsi deux équations non linéaires qu'on doit résoudre par rapport aux inconnues u et v . Une fois les solutions obtenues, la substitution des valeurs de u et v dans (III-9) donne les coordonnées du point d'intersection.

Si on se limite au cas d'une machine 3 axes (c'est d'ailleurs la machine la plus utilisée à cause de sa rigidité importante) le problème sera de trouver la coordonnée Z de la surface pour des coordonnées X et Y données (coordonnées calculées dans l'étape (b) ci-dessus). Le système d'équations deviendra:

$$\begin{cases} x(u, v) - X = 0 = f_1(u, v) \\ y(u, v) - Y = 0 = f_2(u, v) \end{cases} \quad (\text{III-10})$$

Les solutions de ce système, obtenues par la méthode de Newton - Raphson, sont de la forme [Dem 79]:

$$u^{k+1} = u^k - \left[f_1(u^k, v^k) \frac{\partial f_2}{\partial v}(u^k, v^k) - f_2(u^k, v^k) \frac{\partial f_1}{\partial v}(u^k, v^k) \right] / J(u^k, v^k) \quad (\text{III-11})$$

$$v^{k+1} = v^k - \left[f_2(u^k, v^k) \frac{\partial f_1}{\partial u}(u^k, v^k) - f_1(u^k, v^k) \frac{\partial f_2}{\partial u}(u^k, v^k) \right] / J(u^k, v^k)$$

où $J(u^k, v^k)$ est le Jacobien de la fonction $f(u^k, v^k) = (f_1 \ f_2)^T$ tel que

$$J(u^k, v^k) = \frac{\partial f_1}{\partial u}(u^k, v^k) \frac{\partial f_2}{\partial v}(u^k, v^k) - \frac{\partial f_2}{\partial u}(u^k, v^k) \frac{\partial f_1}{\partial v}(u^k, v^k) \quad (\text{III-12})$$

Le processus est itératif; il doit être arrêté lorsque la condition

$$s = \sqrt{f_1(u^{k+1}, v^{k+1})^2 + f_2(u^{k+1}, v^{k+1})^2} < \varepsilon \quad (\text{III-13})$$

sera vérifiée. La quantité ε est une distance présentant l'erreur admissible sur le calcul des intersections.

L'estimation de la solution initiale se fait en cherchant le point d'intersection entre la droite et le réseau caractéristique associé à la surface.

Supposons que ce point d'intersection P soit situé sur le carreau $(P_i, P_{i+1}, P_j, P_{j+1})$, alors la solution initiale devra être:

$$u^0 = u_{\min} + \left[\frac{\sum_{k=1}^i \|P_{k,j} P_{k-1,j}\|}{\sum_{k=1}^m \|P_{k,j} P_{k-1,j}\|} + \|PP_{i,j}\| \right] (u_{\max} - u_{\min}) \quad (\text{III-14a})$$

$$v^0 = v_{\min} + \left[\frac{\sum_{k=1}^j \|P_{i,k} P_{i,k-1}\|}{\sum_{k=1}^n \|P_{i,k} P_{i,k-1}\|} + \|PP_{i,j}\| \right] (v_{\max} - v_{\min}) \quad (\text{III-14b})$$

L'estimation de la solution initiale se fait au point de départ du trajet; un point suivant utilise la solution calculée au point d'intersection précédent comme solution initiale. Cette recherche de la solution initiale ne se fera donc qu'une seule fois pour chaque trajet. La méthode est très rapide; une réalisation pratique nous a montré que le nombre d'itérations effectuées avec cette « astuce » ne dépasse pas généralement 4 pour une précision relative $s\% = 100*s/(2R)$ de moins de 1% tandis que le programme effectue, pour la même précision, en moyenne 6 itérations s'il démarre à chaque fois avec une solution initiale calculée par l'expression (III-14).

La méthode de Newton - Raphson est mise en défaut lorsque le Jacobien J s'annule; il est donc nécessaire de prévoir une autre alternative pour continuer la recherche de la solution. On peut utiliser pour parvenir à cette fin la méthode de la dichotomie ou la méthode de subdivision - élimination. Dans cette dernière on peut remplacer les polyèdres convexes associés aux subsurfaces de subdivision par des boîtes minimax pour n'effectuer, le long du processus, qu'un nombre réduit d'opérations de comparaison [Mart 88, Yen 91, Mic 92].

Nous avons tenu dans notre travail à supposer que les surfaces de projection sont de faibles replis; les problèmes d'accessibilité de l'outil [Tse 91] sont de ce fait écartés et la convergence de l'algorithme de Newton - Raphson est généralement assurée.

La figure III-9 montre un exemple d'utilisation d'une procédure d'intersection surface / droite pour réaliser une simulation de l'usinage en creux des motifs plans simples. Nous avons gravé dans cet exemple les mots « CFAO » et « CAD CAM » sur une surface B-spline biquadratique.

III- 2-2. Calcul du pas de discrétisation

La majorité des MOCN ne « savent » utiliser que les interpolations linéaires et circulaires. Or les trajets générés en vidage des poches sont des courbes paramétriques; leur approximation par des segments linéaires ou circulaires s'avère incontournable.

Dans le premier cas (utilisation des interpolations linéaires), la solution la plus simple consiste à fixer un pas Δu pour une courbe $P(u)$ et à calculer l'ensemble des CC points correspondants $\{P(u_{\min}), P(u_{\min}+\Delta u), P(u_{\min}+2\Delta u), \dots, P(u_{\max})\}$ mais cette solution ne permet pas d'optimiser le nombre de points générés.

La solution préconisée en pratique utilise un pas Δu variable déterminé en comparant la grandeur δ appelée « flèche » (Fig. III-6) avec une valeur δ_a introduite par l'utilisateur et appelée « erreur de flèche admissible ». La flèche δ peut être estimée avec une précision acceptable par l'expression [Qiu 87]:

$$\delta = \left\| P\left(u + \frac{1}{2} \Delta u\right) - \frac{1}{2} [P(u + \Delta u) + P(u)] \right\| \quad (\text{III-15})$$

Si $\delta \leq \delta_a$ alors le point $P(u+\Delta u)$ doit être retenu comme un CC point sinon on divise le pas Δu sur 2, on calcule la nouvelle flèche δ et on refait la comparaison.

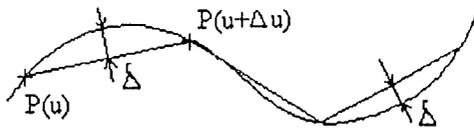


Fig. III-6 Discrétisation d'une trajectoire

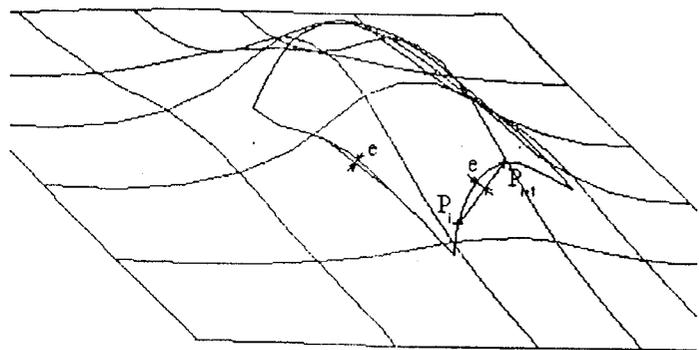


Fig. III-7 Schématisation de l'erreur de projection

III-2-3. Calcul de l'erreur de projection

Le module de balayage des poches génère des fichiers pour les interpolations linéaires; si la poche est décrite entre autres par des segments de droite, ces derniers, une fois projetés, ne

vont plus suivre le relief de la surface (Fig. III-7). Dans ce cas le risque d'avoir des conditions de coupe mauvaises est très probable à cause d'une profondeur de coupe parfois trop importante; c'est le cas d'un segment dont les extrémités sont situées d'un côté et de l'autre d'une bosse.

La solution consiste à calculer l'erreur de projection « e » au milieu de chaque segment de deux points consécutifs projetés et de la comparer à l'erreur de projection admissible e_a . Si $e > e_a$ alors on subdivise le segment au milieu, on projette le nouveau point calculé et on refait la comparaison jusqu'à la satisfaction de la condition $e \leq e_a$. L'erreur de projection peut être approximée par l'expression

$$e = \left\| P^* - \frac{1}{2}(P_{i+1} + P_i) \right\| \quad (\text{III-16})$$

où $P_i = P(u_i, v_i)$; $P_{i+1} = (u_{i+1}, v_{i+1})$; $P^* = P((u_{i+1} + u_i)/2, (v_{i+1} + v_i)/2)$;

P_i et P_{i+1} présentent les positions des deux extrémités du segment projeté par rapport à la surface.

III-2-4. Calcul du pas de décalage (critère de la hauteur de crête)

Le pas de décalage d entre deux trajets successifs appelé « incrément de prise de passe » (Fig. III-4) doit être évalué avant chaque décalage des contours afin de minimiser le volume des sillons creusés par la fraise boule (Fig. III-8a). L'état de surface de la pièce usinée est proportionnel à la hauteur de ces sillons appelée « hauteur de crête » [Ral 92].

Parmi les méthodes d'évaluation du pas de balayage [Lon 87, Cho 88, Ber 89, Suh 90, Cho 93] nous avons retenu celle donnée par [Cho 88].

Le rayon de courbure ρ de la surface dans la direction perpendiculaire à la tangente de la courbe trajet, l'incrément de prise de passe sur la pièce l , le rayon d'outil R et la hauteur de crête h sont liés par la relation:

$$l = \frac{\rho \sqrt{4(R + \rho)^2 (h + \rho)^2 - (\rho^2 + 2R\rho + (h + \rho)^2)^2}}{(R + \rho)(h + \rho)} \quad (\text{III-17})$$

La valeur de ρ peut être calculée par l'expression [Fau 87, Suh 90]:

$$\rho = \frac{U'^T [D] U'}{U'^T [G] U'} \quad (\text{III-18})$$

$$[D] = \begin{bmatrix} N \cdot P_{uu} & N \cdot P_{uv} \\ N \cdot P_{uv} & N \cdot P_{vv} \end{bmatrix} \quad (\text{III-19}); \quad [G] = \begin{bmatrix} P_u \cdot P_u & P_u \cdot P_v \\ P_u \cdot P_v & P_v \cdot P_v \end{bmatrix} \quad (\text{III-20})$$

$$U = [u' \quad v']^T;$$

$$u' \text{ et } v' \text{ sont liés par la relation: } \frac{u'}{v'} = -\frac{P_v \cdot t}{P_u \cdot t} \quad (\text{III-21})$$

où t représente la tangente de la surface de projection au point $P(u,v)$.

$N = P_u \wedge P_v$ est le vecteur normal de la surface au CC point avec:

$$P_u = \frac{\partial}{\partial u} P(u, v); \quad P_v = \frac{\partial}{\partial v} P(u, v);$$

$$P_{uu} = \frac{\partial^2}{\partial u^2} P(u, v); \quad P_{uv} = \frac{\partial^2}{\partial u \partial v} P(u, v); \quad P_{vv} = \frac{\partial^2}{\partial v^2} P(u, v).$$

où $P(u,v)$ représente la surface de projection.

Si on continue le développement des calculs dans (III-18) et si on divise le numérateur et le dénominateur par $(v')^2$ on obtient:

$$\rho = \frac{\frac{u'^2}{v'^2} D_{11} + 2 \frac{u'}{v'} D_{12} + D_{22}}{\frac{u'^2}{v'^2} G_{11} + 2 \frac{u'}{v'} G_{12} + G_{22}} \quad (\text{III-22})$$

Quand ρ est assez important par rapport à R , la simplification suivante sera possible [Lon 87, Cho 88, Suh 90]:

$$l = 2\sqrt{h(2R - h)} \quad (\text{III-23})$$

En réalité c'est la distance d , i.e. l'incrément de prise de passe dans le plan, qui est recherchée; sa valeur peut être déduite par la projection de l sur le plan xy . On obtient ainsi:

$$d = l |q - (q \cdot b)b| \quad (\text{III-24})$$

où $b = (0 \ 0 \ 1)^T$;

$$q = (n \wedge t) / \|n \wedge t\| \quad (\text{III-25})$$

$$t = (n \wedge (t_p \wedge b)) / \|n \wedge (t_p \wedge b)\| \quad (\text{III-26})$$

n est la normale unitaire de la surface et t_p est le vecteur unitaire directeur de la trajectoire dans le plan xy (Fig. III-8b).

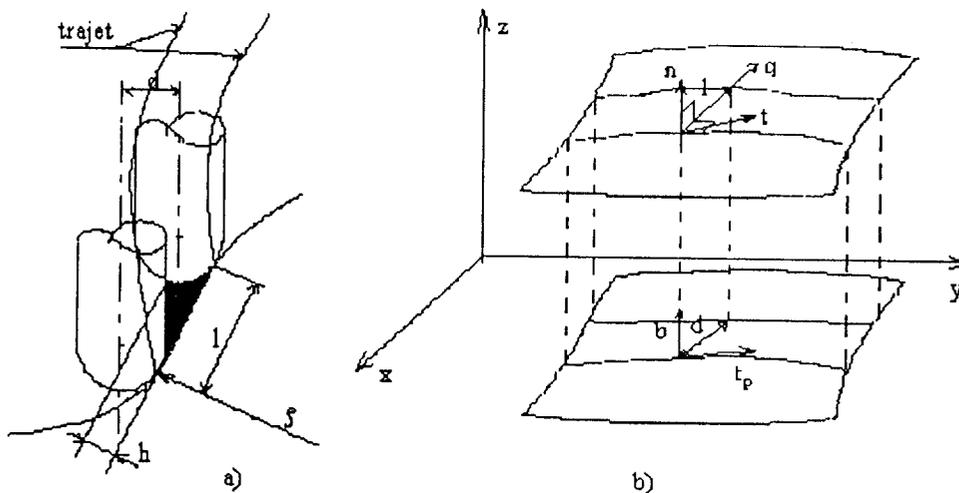


Fig. III-8 *Incrément de prise de passe*

La détermination de la distance de décalage suppose l'évaluation de l'expression (III-17) et ensuite (III-24) sur tous les points du trajet et de prendre ainsi la valeur minimale de d ; une telle démarche est très coûteuse; on se contente généralement de calculer l sur certains points distribués sur le trajet et de prendre la valeur minimale. Dans le cas du balayage isoparamétrique, le choix de la distribution de ces points pourra se faire en fixant un pas de calcul sur la valeur du paramètre de balayage. Dans notre cas (usinage de poches sur surfaces gauches), le problème n'est pas assez facile si on sait que les contours décalés dégèrent

souvent des formes très différentes des contours initiaux (formation de sous-poches). Parmi les solutions qu'on peut avancer, il y a celle qui consiste à réaliser le calcul de l'incrément de prise de passe avant le traitement des interférences.

Notons enfin que des critères de contrôle de l'état de surface basés non pas sur la hauteur de crête mais sur le maintien d'une section ou d'un volume de copeau constant sont utilisés sur certains systèmes de CFAO [Ber 89, Ral 92].

Les figures III-12, ..., III-15 montrent quelques exemples d'usinage de poches de formes complexes sur des surfaces de type Bézier et B-spline par le processus décrit ci-dessus.

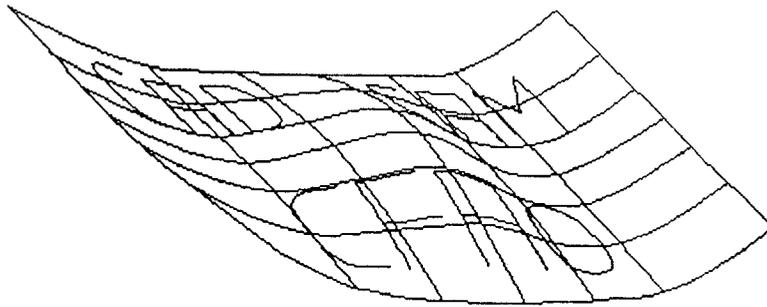
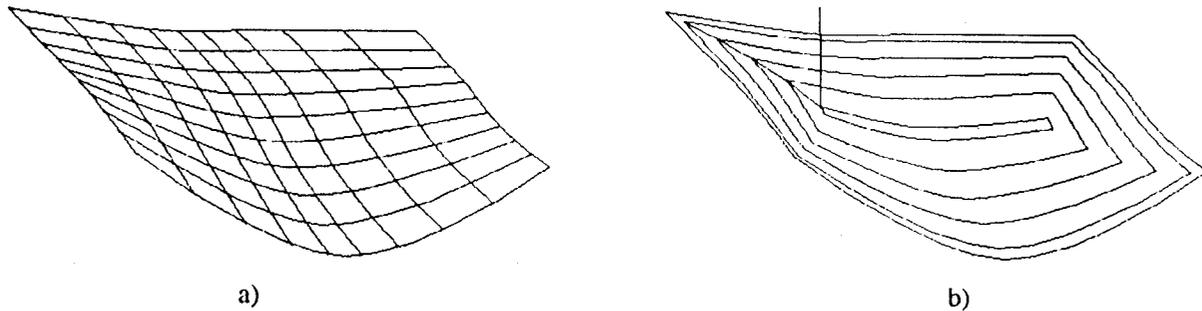


Fig. III-9 Projection de motifs plans sur une surface gauche: usinage en creux des initiales « CAD/CAM » et « CFAO »



*Fig. III-10 Usinage d'une surface gauche par la technique d'évidement de poches
a) la surface à usiner b) les trajectoires d'outil*

c- Usinage par décalages successifs en combinant l'évidement de poches planes et la projection.

6- La gravure des caractères alphanumériques sur les surfaces gauches.

7- La simulation graphique en continue ou en pas à pas de l'usinage.

9- FFMach possède une fonction de contrôle du chemin d'outil qui permet de signaler les points où la vitesse de coupe est nulle, de calculer les distances et les temps totaux d'usinage effectif et de déplacement en mode G00 ainsi que leurs ratios.

10- La génération et l'édition des programmes ISO pour les MOCN à 3 axes.

FFMach a été pour nous une première expérience pour la réalisation d'un système de FAO 3D. Nous espérons dans l'avenir remodeler entièrement sa structure et sa conception afin de l'adapter aux réalités de l'utilisation industrielle.

III-4. Exemples de simulation d'usinage de poches sur des surfaces gauches

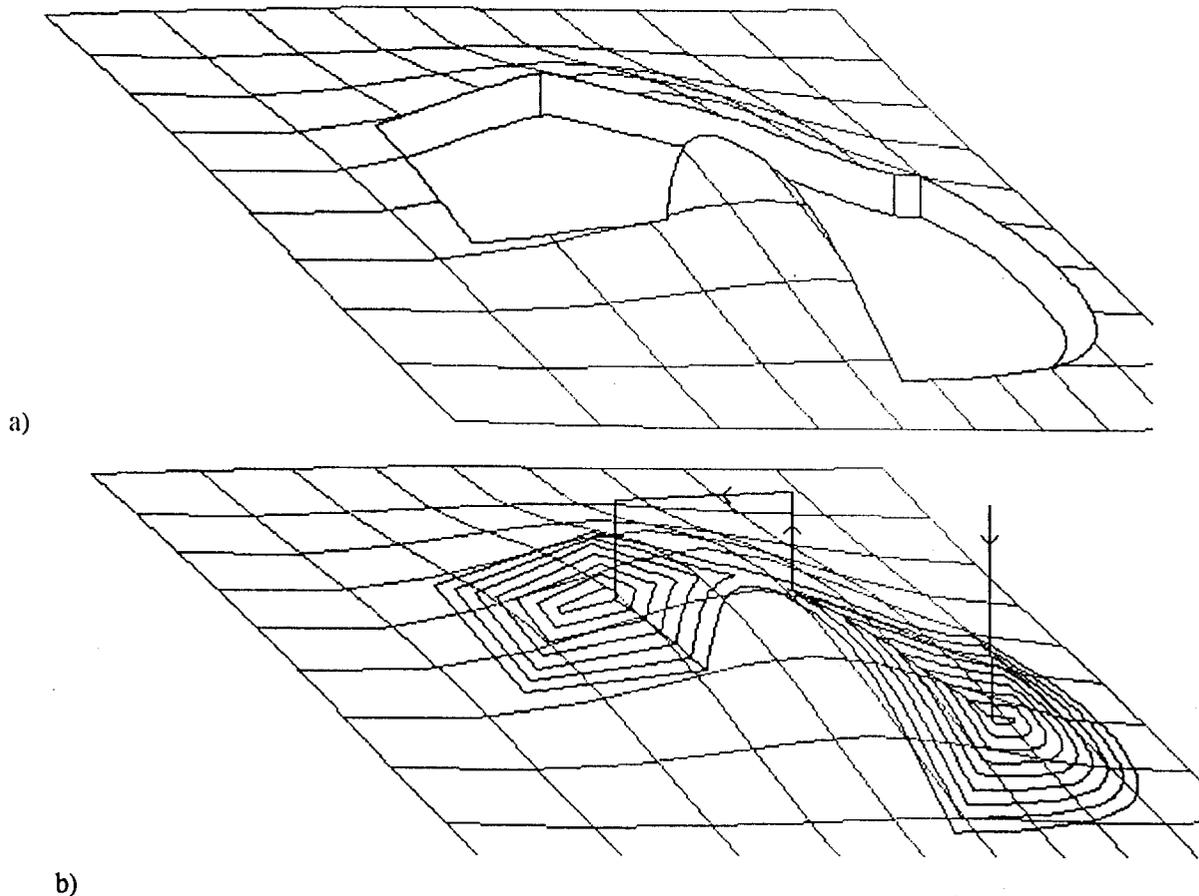
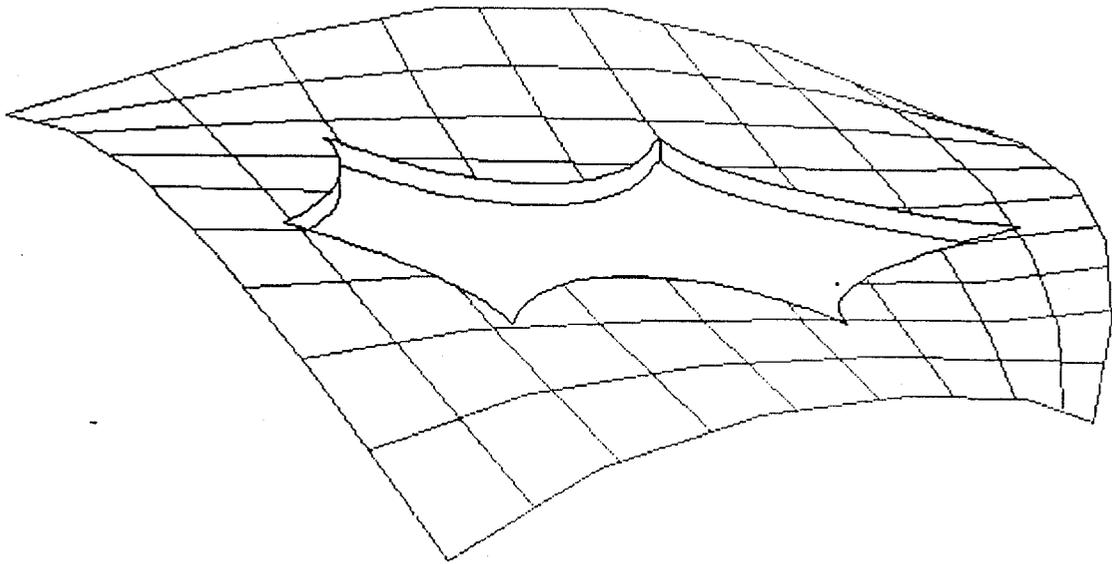
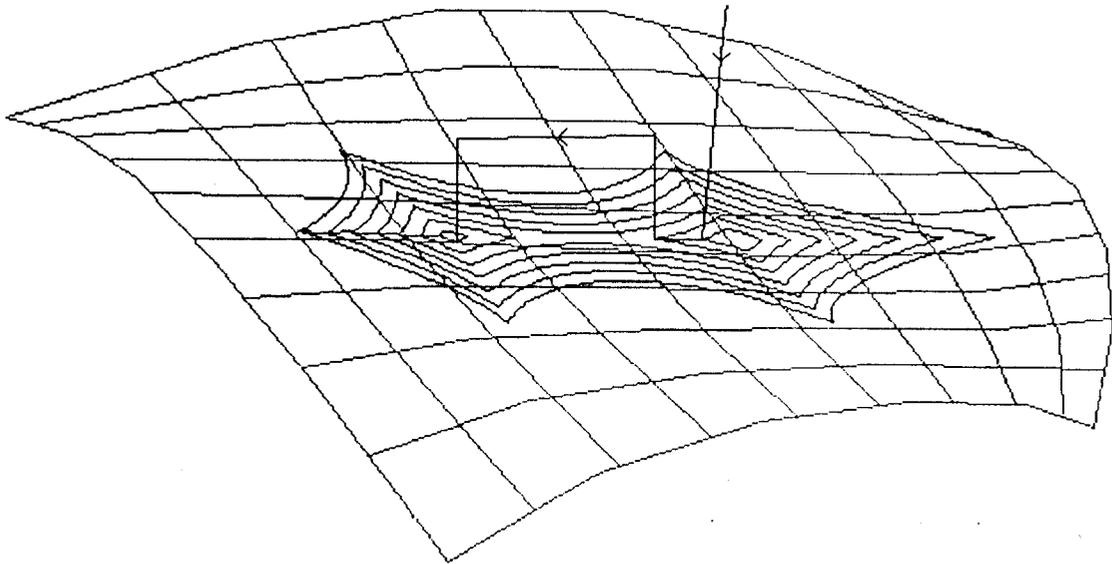


Fig. III-12 La poche de la Fig. III-46b usinée sur une surface B-spline
(a) la forme à creuser (b) trajets de l'outil



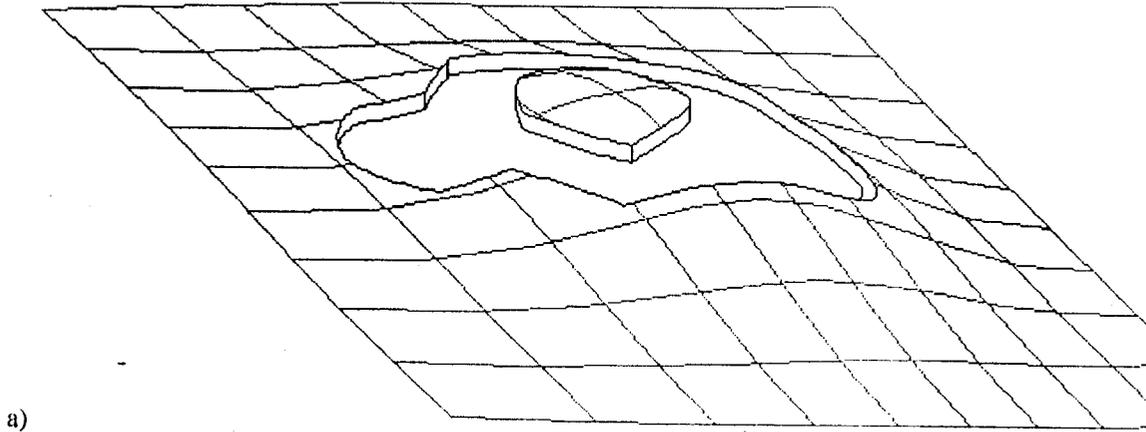
a)



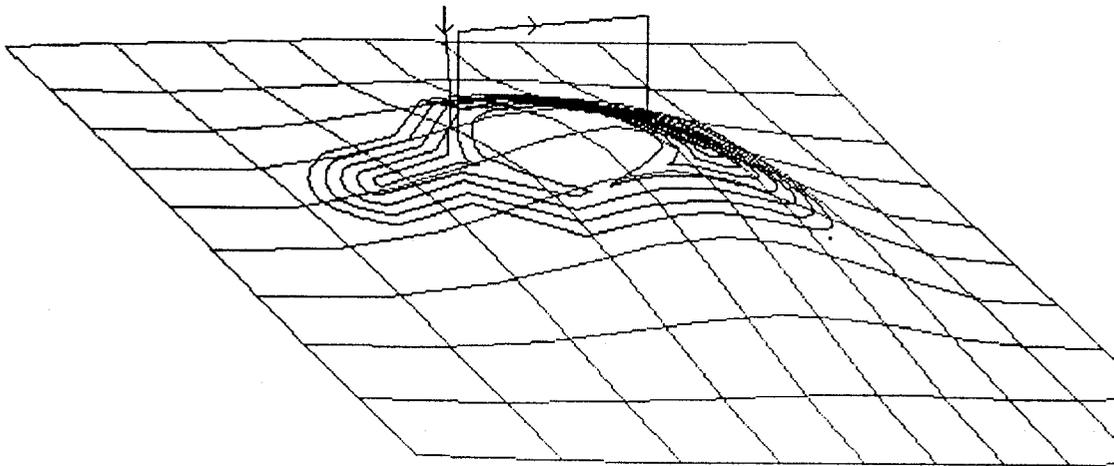
b)

Fig. III-13 La poche de la Fig. III-46d usinée sur une surface B-spline

(a) la forme à creuser (b) trajets de l'outil



a)

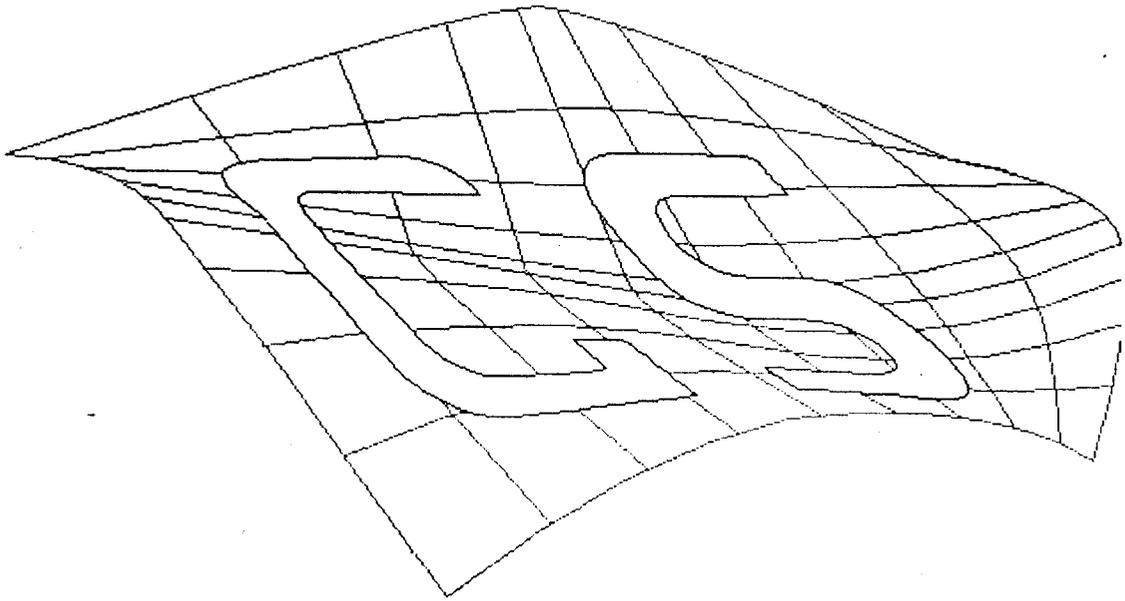


b)

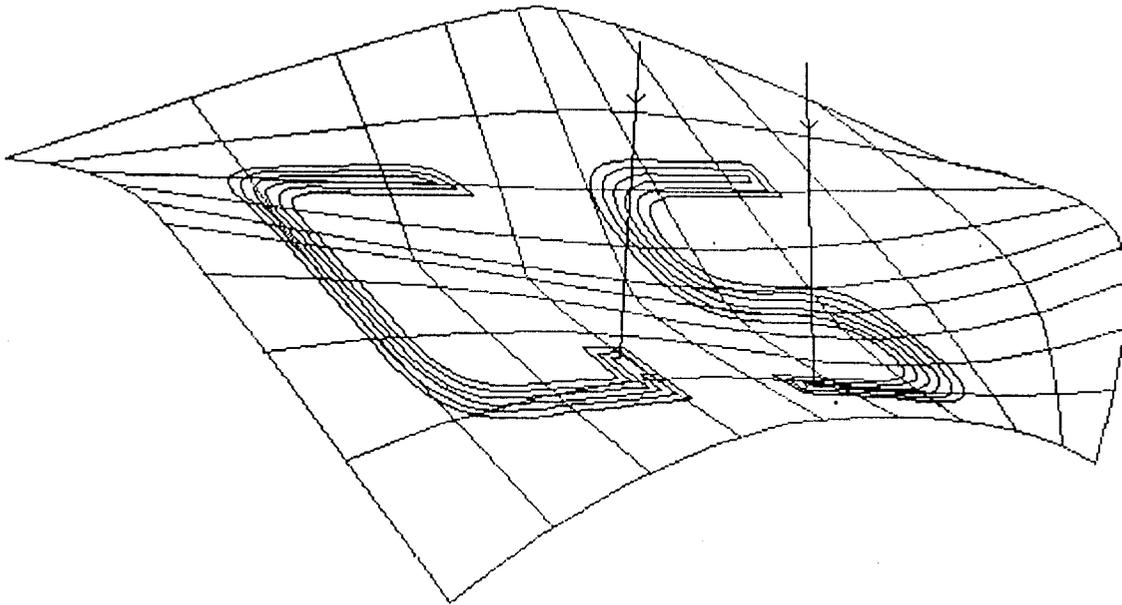
Fig. III-14 Usinage d'une poche avec îlot sur une surface B-spline bicubique

(a) la forme à creuser

(b) Trajets de l'outil



a)



b)

Fig. III-15 Gravure des lettres G et S sur une surface gauche

CONCLUSION

CONCLUSION

Les objectifs qui nous ont été fixés dans cette étude de recherche étaient:

(1) Concevoir une méthode générale pour la réalisation des évidements de poches planes de formes gauches avec ou sans îlots. La conception de cette méthode est une tâche difficile et très complexe même pour des poches simples contrairement à ce que l'on pourrait penser à première vue. Cette difficulté est due principalement aux deux facteurs suivants:

- La nature des problèmes traités qui font appel souvent aux techniques de la géométrie algorithmique. Or les méthodes de traitement automatique des problèmes géométriques sont généralement très éloignées de la méthode de résolution qu'utiliserait un être humain d'une part et d'autre part beaucoup de problèmes, qu'on pourrait résoudre instantanément sur un simple examen d'un dessin (par exemple les boucles, les intersections ou l'inclusion d'un point dans un contour), demandent des programmes de traitement assez complexes.

- L'emploi de la méthode des décalages pour l'évidement des poches nécessite le traitement de certains problèmes de CFAO qui ne sont pas encore totalement et efficacement résolus, à savoir le calcul des courbes offset, les intersections des courbes gauches et la caractérisation des courbes à pôles ou de leurs courbes offset (recherche des points singuliers et points multiples).

La plupart de ces difficultés ont été surmontées. Un effort particulier a été consacré pour concevoir des solutions efficaces pour le traitement des interférences. Ces efforts sont soldés par la mise en oeuvre de méthodes simples et efficaces pour l'identification et le traitement des collisions et notamment celles dues aux boucles intracontours et intercontours. Une approche inédite d'évidement de poches avec plusieurs îlots a été également conçue.

L'ensemble des techniques et des choix mentionnés dans ce document nous a permis de réaliser un programme informatique d'évidement de poches planes gauches dénommé « PMach » (pour Pocket Machining). Celui-ci nous a permis de valider ce travail de recherche au travers de nos expérimentations. Des comparaisons avec les logiciels « SurfCAM » de SURFWARE Inc et DMT10 de MecaSoft Industries ont donné lieu à des résultats analogues du point de vue de la forme des trajectoires générées comme du côté concernant la planification de ces trajectoires.

(2) L'application des résultats obtenus dans l'usinage des surfaces gauches:

- La première application était l'usinage des surfaces gauches par la technique des plans de niveau. Hormis la recherche des intersections surface - plan et le suivi des contours d'intersection, cette application est relativement simple du fait qu'elle utilise d'une façon indépendante les routines de vidage de poches planes. Néanmoins, toutes les étapes de calcul, en ébauche comme en finition, ont été suffisamment discutées.

La deuxième application concerne le creusage des poches sur des surfaces gauches. Dans ce cas la génération des trajectoires d'outil dans le plan et leur projection sur la surface d'usinage doivent se faire conjointement. Cette approche demande des méthodes de calcul sophistiquées, un bon sens et des expérimentations pratiques des trajectoires générées sans oublier autant l'influence des paramètres technologiques lors d'un tel usinage tridimensionnel. Nous espérons qu'un travail de validation des trajectoires d'outil générées par les programmes que nous avons réalisés fera l'objet d'une future étude.

Il est opportun de mentionner aussi que notre utilitaire informatique FFMach peut réaliser également l'usinage de surfaces gauches par une procédure d'évidement de poches par décalages successifs et projection. Des comparaisons avec la méthode de balayage isoparamétrique, du point de vue état de surface et temps d'usinage, seront nécessaires pour son éventuelle implantation sur un système de FAO.

Avec les programmes informatiques que nous avons développés nous disposons maintenant d'un ensemble d'outils très efficaces pour la réalisation de tâches considérées jusqu'à présent onéreuses. Ces programmes ne demandent comme moyens matériels qu'un micro-ordinateur standard (PC ou compatibles). Les temps de réponse n'excèdent pas généralement quelques secondes.

Les développements futurs que nous espérons réaliser pour ce travail vont porter sur (i) l'optimisation du pas de balayage et de la profondeur de passe en usinage par plans de niveau, (ii) l'extension du travail aux machines 5 axes et à d'autres types d'outils et (iii) le contrôle des collisions pour les surfaces à courbure quelconque.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [Aom 90] Aomura S and Uehara T: *Self - intersection of an offset surface*;
CAD vol 22 No 7 (1990) pp 417-422
- [Aud 82] Audy J: *Fabrication assistée par ordinateur*; Techniques de l'ingénieur 9 - 1982
- [Bal 86] Ball A.A and Storry D.J.T: *A matrix approach to the analysis of recursively generated B-splines surfaces*; CAD vol 18 No 8 (1986) pp 437-442
- [Bar 88a] Bartels R.H, Beaty J.C et Barsky B.A: *B-splines*;
collection mathématiques et CAO 6, Hermès Paris 1988
- [Bar 88b] Bartels R.H, Beaty J.C et Barsky B.A: *Beta-splines*
collection mathématiques et CAO 7, Hermès Paris 1988
- [Ben 90] Ben salah S: *Equivalences et algorithmique des transformations dans les modèles mathématiques pour la CFAO des courbes et des surfaces*;
Thèse de Doctorat, ENSAM - Paris 1990
- [Ber 89] Bernard A: *Usinage tridimensionnel d'outillages de topologie complexe: analyse des contraintes de production et contribution à l'optimisation du processus d'usinage*;
Thèse de doctorat, Ecole Centrale Paris 1989
- [Bez 85] Bez H.E: *Technical note: cubic curves*; CAD vol 17 No 1 (1985) pp 367-368
- [Bezi 70] Bezier P: *Emploi des machines à commande numérique*;
Masson et Cie - Eyrolles Paris 1970
- [Bezi 86] Bezier P: *Courbes et surfaces - Mathématiques et CAO 4*; Hermès Paris 1986
- [Bezi 89] Bezier P: *First steps of CAD*; CAD vol 21 No 5 (1989) pp 259-261
- [Bob 85] Bobrow J.E: *NC Machine tool path generation from CSG part representation*;
CAD vol 17 No 2 (1985) pp 69-76
- [Boe 80] Boehm W: *Inserting new knots into B-spline curves*;
CAD vol 12 No 4 (1980) pp 199-202
- [Boe 85] Boehm W and Prautzsch H: *The inserting algorithm*;
CAD vol 17 No 2 (1985) pp 58-59
- [Cas 85] De Casteljaou P: *Formes à pôles - Mathématiques et CAO 2*, Hermès Paris 1985
- [Che 86] Chenin P et al: *Mathématiques et CAO: méthodes de base*; Hermès Paris 1986
- [Cho 88] Choi B.K, Lee C.S, Hwang J.S and Jun C.S: *Compound surface modeling and machining*; CAD vol 20 No 3 (1988) pp 127-136

- [Cho 89] Choi B.K and Jun C.S: *Ball - end cutter interference avoidance in NC machining of sculptured surfaces*; CAD vol 21 No 6 (1989) pp 371-378
- [Cho 90] Choi B.K, Yoo W.S and Lee C.S: *Matrix representation for NURB curves and surfaces*; CAD vol 22 No 4 (1990) pp 235-240
- [Cho 93] Choi B.K, Park J.W and Jun C.S: *Cutter location data optimization in 5 - axis surface machining*; CAD vol 25 No 6 (1993) pp 377-386
- [Coo 87] Coons S.A: *Méthode matricielle - Mathématiques et CAO 5*; Hermès Paris 1987
- [Coq 87] Coquillart S: *Computing offsets of B-spline curves*;
CAD vol 19 No 6 (1987) pp 305-309
- [Cra 85] Crampin M, Guifo R and Read G.A: *Linear approximation of curves with bounded curvature and data reduction algorithm*; CAD vol 17 No 6 (1985) pp 257-261
- [Dem 79] Demidovitch B et Maron I: *Eléments de calcul numérique*; Mir Moscou 1979
- [Eps 76] Epstein M.P: *On the influence of parametrization in parametric interpolation*;
SIAM Journal. Numer. Anal. Vol 13 No 2 (1976) pp 261-268
- [Fab 85] Fabrikant V.L and Sankar T.S: *An efficient graphical method for CAD*;
CAD vol 17 No 8 (1985) pp 369-373
- [Far 85] Farouki R.T: *Exact offset procedure for simple solids*;
Comput. Aided Geom. Design No 2 (1985) pp 257-279
- [Far 86] Farouki R.T: *The approximation of non - degenerate offset surfaces*;
Comput. Aided Geom. Design No 3 (1986) pp 15-43
- [Far 90a] Farouki R.T and Neff C.A: *Analytic properties of planes offset curves*;
Comput. Aided Geom. Design No 7 (1990) pp 83-99
- [Far 90b] Farouki R.T and Neff C.A: *Algebraic properties of planes offset curves*;
Comput. Aided Geom. Design No 7 (1990) pp 101-127
- [Fau 87] Faux I.D and Pratt M.J: *Computational geometry for design and manufacture*;
Ellis Horwood publishers Chichester 1987
- [Fil 90] Filip D.J: *Determining the orientation of closed planar curves*;
CAD vol 22 No 7 (1990) pp 401-406
- [Fog 84] Fog N.G: *Creative definition and fairing of ship hulls using B-spline surface*;
CAD vol 16 No 4 (1984) pp 225-229
- [For 90] Forrest A.R: *Interactive interpolation and approximation by Bezier polynomials*;
CAD vol 22 No 9 (1990) pp 527-537
- [Gar 87] Gardan Y: *La CFAO introduction, techniques, et mise en oeuvre - 2eme édition*

Hermès Paris 1987

- [Gol 87] Goldman R.N and Sederberg T.W: *Analytic approach to intersection of all piecewise parametric rational cubic curves*; CAD vol 19 No 6 (1987) pp 282-292
- [Har 80] Hartley P.J and Judd C.J: *Parametrization and shape of B-spline curves For CAD*; CAD vol 12 No 5 (1980) pp 235-238
- [Hel 94] Held M, Lukacs G and Andor L: *Pocket machining based on contour-parallel tool paths generated by means of proximity maps*; CAD vol 26 No 3 (1994) pp189-203
- [Hol 83] Holzle G.E: *Knot placement for piecewise polynomial approximation of curves* CAD vol 15 No 5 (1983) pp 295-296
- [Hos 85] Hoschek J: *Offset curves in the plane*; CAD vol 17 No 2 (1985) pp 77-82
- [Hos 88a] Hoschek J and Wissel N: *Optimal approximate conversion of spline curves and spline approximation of offset curves*; CAD vol 20 No 8 (1988) pp 475-483
- [Hos 88b] Hoschek J: *Spline approximation of offset curves*;
Comput. Aided Geom. Design No 5 (1988) pp 33-40
- [Hos 88c] Hoschek J: *Intrinsic parametrization for approximation*;
Comput. Aided Geom. Design No 5 (1988) pp 27-31
- [Hwa 92] Hwang J.S: *Interference free tool path generation in the NC machining of parametric compound surfaces*; CAD vol 24 No 12 (1992) pp 667-676
- [Kim 88] Kim K and Biegel J.E: *An integrated approach to sculptured surface design and manufacture*; Comput. Ind. Eng. vol 14 No 3 (1988) pp 271-280
- [Kim 93] Kim D.S: *Hodograph approach to geometric characterization of parametric cubic curves*; CAD vol 25 No 10 (1993) pp 644-654
- [Kimm 93] Kimmel R and Bruckstein A.m: *Shape of offsets via level sets*;
CAD vol 25 No 11 (1993) pp 154-162
- [Kla 83] Klass R: *An offset spline approximation for plane cubic spline*;
CAD vol 15 No 4 (1983) pp 297-299
- [Kur 92] Kuragano T: *FRES DAM System for design of aesthetically pleasing free - form objects and generation of collision - free tool path*;
CAD vol 24 No 11 (1992) pp 573-581
- [Las 89] Lasser D: *Calculating the self intersection of Bezier curves*;
Computers in Industry 12 (1989) pp 259-268
- [Lee 89] Lee E.T.Y: *Choosing nodes in parametric curve interpolation*;
CAD vol 21 No 6 (1989) pp 363-370

- [Lee 92] Lee Y.S, Choi B.K and Chang T.C: *Cut distribution and cutter selection for sculptured surface cavity machining*,
Int. Journal. Prod. Res. vol 30 No 6 (1992) pp 1447-1470
- [Leo 91] Leon J.C: *Modélisation et construction de surfaces pour la CFAO*
Hermès Paris 1991
- [Lon 87] Loney G.C and Ozsoy T.M: *NC machining of free form surfaces*
CAD vol 19 No 2 (1987) pp 85-90
- [Mar 89] Martin P: *Informatique - méthodes et applications*, Ellipse Paris 1989
- [Marc 87] Marciniak K: *Influence of surface shape on admissible tool position in 5 - axis face milling*, CAD vol 19 No 5 (1987) pp 233-236
- [Mark 89] Markot R.P and Magedson R.L: *Solutions of tangential surface and curve intersections*, CAD vol 21 No 7 (1989) pp 421-429
- [Mart 88] Martin R.R and Stephenson P.C: *Putting objects into boxes*,
CAD vol 20 No 9 (1988) pp 506-514
- [Mee 92] Meek D.S and Walton D.J: *Approximation of discrete data by GI arc splines*,
CAD vol 24 No 6 (1992) pp 301-306
- [Mic 92] Michel D: *Contribution à la conception, mise en oeuvre et l'amélioration des algorithmes des intersections des carreaux NURBS*,
Thèse de doctorat, université de Metz 1992
- [Mil 93] Milenkovic V: *Robust polygon modeling*, CAD vol 25 No 9 (1993) pp 546-566
- [Mor 91] Moreton D.N, Parkinson D.B and Wu W.K: *The application of biarc technique in CNC machining*, Comput. Aided Eng. Journal April (1991) pp 54-60
- [Mul 88] Mullineux 88: *Finding the composite outer boundary and holes of a set of closed curves*, Comput. Aided Geom. Design No 5 (1988) pp 51-58
- [Nou 89] Nougier J.P: *Méthodes de calcul numérique - 3eme édition*, Masson Paris 1993
- [Num 88] NUM: *Manuel de programmation NUM 720F*, publication No 938702 (1988)
- [Pat 89] Patrikalakis N.M: *Offsets on curves on rational B-spline surfaces*,
Engineering with Computers 5 (1989) pp 39-46
- [Pen 84] Peng Q.S: *An algorithm for finding the intersection lines between two B-spline surfaces*, CAD vol 16 No 4 (1984) pp 191-196
- [Per 78] Persson H: *NC machining of arbitrarily shaped pockets*,
CAD vol 10 No 3 (1978) pp 169-174
- [Per 90] Peroche P et al: *La synthèse d'images*, Hermès Paris 1990

- [Pha 88] Pham B: *Offset approximation of uniform B-splines*;
CAD vol 20 No 8 (1988) pp 471-474
- [Pha 92] Pham B: *Offset curves and surfaces - a brief survey*
CAD vol. 24 No 4 (1992) pp 223-229
- [Pie 84] Piegl L: *A generalization of the Bernstein - Bezier method*;
CAD vol 16 No 4 (1984) pp 209-215
- [Pie 85] Piegl L: *Recursive algorithm for the representation of parametric curves and
-surfaces*; CAD vol 17 No 5 (1985) pp 225-229
- [Pie 86] Piegl L: *Curve fitting algorithm for rough cutting*; CAD vol 18 No 2 (1986) pp 79-82
- [Pie 87] Piegl L and Tiller W: *Curve and surface constructions using rational B-splines*;
CAD vol 19 No 9 (1987) pp 485-498
- [Pie 89a] Piegl L: *Modifying the shape of rational B-splines Part 1: curves*;
CAD vol 21 No 8 (1989) pp 509-518
- [Pie 89b] Piegl L: *Modifying the shape of rational B-splines Part 2: surfaces*;
CAD vol 21 No 9 (1989) pp 538-546
- [Pie 91] Piegl L: *On NURBS: A survey*; IEEE CG & Applications January 1991 pp 55-71
- [Piq 89] Piquet F, Poitou J.P et Tasse J.C: *La CFAO concevoir et produire autrement*;
Nathan Communication Paris 1989
- [Qiu 87] Qiulin D and Davies B.J: *Surface engineering geometry for computer aided design
and manufacture*; Ellis Horwood Chichester 1987
- [Ral 78] Ralston A and Rabinowitz P: *A first course in numerical analysis*;
McGraw Hill Tokyo 1978
- [Ral 92] Ralph Ip W.L and Loftus M: *Cusps geometry analysis in free - form surface
machining*; Int. Journal. Prod. Res. vol 30 No 11 (1992) pp 2697-2711
- [Rat 89] Ratschek H and Rokne J: *Test for intersection between plane and box*;
CAD vol 25 No 4 (1989) pp 249-250
- [Rie 81] Riesenfeld R.F: *Homogeneous coordinates and projective planes in Computer
Graphics*; IEEE CG & Applications (1981) pp 50-55
- [Rob 85] Robergé J: *A data reduction algorithm for planar curves* ;
Comput. Vision, Graph. and Image Proc 29 (1985) pp 168-195
- [Rog 89a] Rogers D.F and Adams J.A: *Mathematical elements for computer graphics*;
Mc Graw Hill New York 1989
- [Rog 89b] Rogers D.F and Fog N.G: *Constrained B-spline curve and surface fitting*;

- CAD vol 21 No 10 (1989) pp 641-648
- [Ron 91] Ronghang W and Wenhai J: *An algorithm of the offset shape*;
Comput. & Graphics vol 15 No 3 (1991) pp 435-439
- [Sac 88] Sachs G.A, Schwartz A.J and Sleator F.B: *Determining the intersection of parametric surfaces by solving ordinary differential equations*;
publication 870874 Society of Automotive Engineers (1988)
- [Sap 90] Sapidis N and Farin G: *Automatic fairing algorithm for B-spline curves*;
CAD vol 22 No 2 (1990) pp 121-129
- [Sed 84] Sederberg T.W, Anderson D.C and Goldman R.N: *Implicit representation of parametric curves and surfaces*;
Comput. Vision, Graph., and Image Proc. No 28 (1984) pp 72-84
- [Sed 85] Sederberg T.W, Anderson D.C and Goldman R.N: *Implicitization, inversion and intersection of planar rational cubic curves*;
Comput. Vision, Graph., and Image Proc. No 31 (1985) pp 89-102
- [Sed 86] Sederberg T.W and Parry S.R: *Comparison of three curve intersection algorithms*;
CAD vol 18 No 1 (1986) pp 58-63
- [Sed 89] Sederberg T.W: *Algorithm for algebraic curve intersection*;
CAD vol 21 No 9 (1989) pp 547-554
- [Sed 90] Sederberg T.W and Nishita T: *Curve intersection using Bezier clipping*;
CAD vol 22 No 9 (1990) pp 538-549
- [Suh 90] Suh Y.S and Lee K: *NC milling tool path generation for arbitrary pockets defined by sculptured surfaces*; CAD vol 22 No 5 (1990) pp 273-284
- [Tan 90] Tan S.T, Sze W.S and Wong W.Y: *A survey and classification of cutter path generation methods for three axis NC milling*;
Adv. Manuf. Eng. vol 2 (1990) pp 103-113
- [Til 84] Tiller W and Hanson E.G: *Offsets of two-dimensional profiles*;
IEEE CG & Applications (1984) pp 36-46
- [Tos 90] Tosan E: *Modélisation géométrique cours de techniques graphiques avancées*;
Université Claude Bernard Lyon 1990
- [Tre 83] Tremblay J.P and Bunt R.B: *An introduction to computer science - An algorithmic approach*, 3rd edition; McGraw - Hill Tokyo 1983
- [Tse 91] Tseng Y.T and Joshi S: *Determining feasible tool - approach direction for machining Bezier curves and surfaces*; CAD vol 23 No 5 (1991) pp 367-379

RESUME

Le balayage de poches en fabrication mécanique consiste à enlever toute la matière délimitée par un contour extérieur fermé tout en laissant celle qui est à l'intérieur des îlots. Ce procédé d'usinage est courant en fabrication des moules et outillages, en gravure et en usinage des surfaces gauches. L'approche que nous proposons est basée sur la méthode des décalages successifs; elle concerne les poches décrites à partir de combinaison d'éléments géométriques simples et courbes à pôles (Bézier et B-splines). Notre approche traite les poches concaves ou convexes sans îlots comme celles possédant plusieurs îlots; chose non disponible dans la littérature mise à notre disposition.

Le premier chapitre du mémoire comporte une synthèse largement documentée sur les caractéristiques des formes à pôles (courbes et surfaces). Le deuxième chapitre est consacré à l'élaboration d'une méthodologie d'évidement de poches planes. Plusieurs sujets d'actualité en CFAO ont été abordés (approximation des offset, calcul des intersections, caractérisation des courbes gauches, etc.). L'approche élaborée dans ce chapitre peut servir en outre pour la modélisation de certains phénomènes de propagation d'ondes. Des algorithmes prêts à utiliser sont aussi inclus. Le troisième chapitre évoque l'application du balayage de poches à l'usinage des surfaces gauches. La première application concerne l'usinage par la technique des plans de niveau rencontrée souvent en ébauche des surfaces réalisées à partir des blocs de matière (fabrication des prototypes et pièces en petite série). La deuxième application concerne l'usinage de poches sur des surfaces ou l'usinage des surfaces gauches par l'évidement de poches.

Cette thèse qui est la première en ce genre dans les instituts de Boumerdes et les régions avoisines constitue un premier pas vers la maîtrise des techniques d'usinage de pointe.