

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



Ecole Nationale Polytechnique

DEPARTEMENT DE GENIE MECANIQUE

Projet de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état
en Génie mécanique

المدرسة الوطنية للتكنولوجيا
BIBLIOTHEQUE ← المكتبة
Ecole Nationale Polytechnique

Thème

**ETUDE COMPARATIVE DES DIFFERENTES
METHODES DE REDUCTION DE DEGRE DES
COURBES GAUCHES**

- Application à l'échange de données entre différents systèmes
de C.F.A.O.

Proposé et dirigé par :

Mr : S. RECHAK

Etudié par :

HAMRI OKBA

Membre du jury :

Mr : Y.BELKACEMI

Mr : H. BELHANECHÉ

Mr : S. RECHAK

Promotion : septembre 2000

E.N.P.10, Avenue Hassen-Badi, 16200 EL-HARRACH, ALGER.



Dédicace

*Je dédie ce modeste travail,
Aux êtres les plus tendres à mes
yeux et les plus
chers à mon cœur, à qui je dois
énormément
et que je ne remercierai jamais
assez :*

Mes parents.

A mes frères et sœurs.

A Toute la famille HAMRI.

A tous mes amis de promotion.

OKBA.

Remerciements

Je remercie dieu de m'avoir donné la volonté et le courage afin d'arriver à la finalité de ce modeste travail.

Je tiens à remercier en premier lieu mon promoteur M^{er} : S.Rechak, pour ses encouragements, son suivi, et pour les conseils qu'il a apporté pour l'achèvement de ce projet, ainsi M^{er} : A.Bouras , M^{er} : P.Gautier chercheurs au laboratoire de C.F.A.O. de Lyon (France), M^{er} : F.Aasma enseignant à l'université de Tizi-Ouzou .

Je tiens également à remercier messieurs les membres de jury pour l'honneur qu'ils m'ont fait en acceptant de siéger à ma soutenance, ainsi que l'ensemble des enseignants qui ont contribué à ma formation.

Mes remerciements vont également à mes collègues et amis de la promotion, à Djamel, Mourad, Fateh, Lahcene.

ملخص

الغاية من هذه الدراسة عرض مختلف الطرق لتخفيض رجة منحنيات لBEZIER لغرض تبادل المعطيات بين مختلف أنظمة CFAO مع دراسة مقارنة بين مختلف الطرق كذلك سوف نقتراح طريقة فعالة التي تسمح لنا بحساب منحني BEZIER بدرجة أتدني تقارب منحني أصلي مع فرض دقة معينة.

الكلمات المفتاح : درجة تخفيض , تقريب , منحنيات لBEZIER , استكمال , تحويل المعطيات , CFAO , خصائص ذاتية , مربعات صغيرة.

Résumé

L'objectif de ce travail est de présenter les différentes méthodes de réduction de degré des courbes de Bézier non rationnelles, en vue de l'échange de données entre différents systèmes de C.F.A.O et en faisant une étude comparative de ces différentes méthodes. Ainsi, nous proposons une méthode efficace permettant de calculer la courbe de Bézier de degré minimale approchant la courbe originale avec une précision donnée et temps de calcul optimal.

Mots clés : réduction de degré, approximation, courbe de Bézier, moindres carrés, paramétrage optimal, paramétrisation intrinsèque, optimisation, polynôme de Tchybecheff, interpolation, transfert de données, C.F.A.O.

Abstract

The aim of this work is to present the different methods of degree reduction of non rational Bézier curves, in sight of the exchange between different systems of CAD/CAM and a comparative study of these different methods. So, we suggest an efficient method who let's to compute the minimal degree of Bézier curve that approach the original curve with a given accuracy .

Keywords : degree reduction, approximation, Bézier curves, least squares, optimal parametrization, intrinsic parametrization, optimization ,Tshybechev polynomials, interpolation , data exchange, CAD/CAM.

Etude comparative des méthodes de degré des courbes gauches.

Application à l'échange de données entre systèmes.

Tables des matières

Dédicace

Remerciements

Tables des matières

Tables des illustrations

Liste des abréviations

Notations

Introduction

1. Courbes et surfaces : modèles de Bézier et B-spline

1.2. Courbes et surfaces de Bézier

1.2.1. Algorithme de De Casteljau. 5

1.2.2. Polynômes de Bernstein. 6

1.2.3. Courbes de Bézier non rationnelles. 7

1.2.3.1. Dérivation d'une courbe de Bézier. 8

1.2.3.2. Propriétés des courbes de Bézier non rationnelles. 9

1.2.3. Courbes de Bézier rationnelles. 10

1.2.4. Surfaces de Bézier non rationnelles. 13

1.2.4.1. Interpolation bilinéaire. 14

1.2.4.2. Algorithme de De Casteljau direct. 14

1.2.4.3. Approche produit tensoriel. 15

1.2.5. Surfaces de Bézier rationnelles. 15

1.3. courbes et surfaces B-spline

1.3.1. Fonction de base B-spline. 16

1.3.2. Courbes B-spline non rationnelles. 18

1.3.3. Courbes B-spline rationnelles. 21

1.3.4. Surfaces B-spline non rationnelles. 24

1.3.5. Surfaces B-spline rationnelles. 26

2. Algorithmique des transformations

2.1. Transformations dans le modèle de Bézier

2.1.1. Subdivision. 28

2.1.1.1. Courbes non rationnelles. 28

2.1.1.2. Courbes rationnelles. 29

2.1.2. Elévation de degré. 30

2.1.2.1. Courbes non rationnelles. 30

2.1.2.2. Courbes rationnelles. 31

2.1.3. Réduction de degré. 31

2.1.3.1. Courbes non rationnelles. 31

2.1.3.2. Courbes rationnelles. 32

2.1.4. Raccordement des courbes. 33

2.1.4.1. Courbes non rationnelles. 33

2.1.4.2. Courbes rationnelles. 34

2.2. Transformation dans le modèle B-spline	
2.2.1. Insertion des nœuds.	34
2.2.1.1. Courbes non rationnelles.	34
2.2.1.2. Courbes rationnelles.	36
2.2.2. Suppression de nœuds.	37
2.2.2.1. Courbes non rationnelles.	37
2.2.2.2. Courbes rationnelles.	39
2.2.3. Subdivision et l'algorithme d'Oslo.	40
2.2.3.1. Courbes non rationnelles.	40
2.2.3.2. Courbes rationnelles.	43
2.3. Changement de bases	
2.3.1. B-spline – Bézier.	45
2.3.1.1. Le polygone local de Bézier d'une courbe B-spline.	45
2.3.1.2. Les polygones de Bézier d'une courbe B-spline.	45
2.3.2. Bézier - B-spline.	46
2.3.2.1. Polygone B-spline d'une courbe de Bézier.	46
2.3.2.2. Polygone B-spline des courbes locales de Bézier.	46
2.3.2.3. Algorithme WM.	47
3. Les principales méthodes de réduction de degré.	
Problématique.	50
3.1. Principe général de conversion par approximation.	50
3.2. Méthodes et Algorithmique de réduction.	52
3.2.1. Méthode de Forrest.	52
3.2.1.1. Algorithme de réduction.	52
3.2.1.2. Algorithme de conversion par approximation.	59
3.2.2. Méthode de Bensalah.	61
3.2.2.1. Algorithme de réduction.	61
3.2.2.2. Algorithme de conversion par approximation.	63
3.2.3. Méthode de Hoschek.	67
3.2.3.1. Paramétrisation intrinsèque.	67
3.2.3.1.1. Approximation optimale des courbes.	67
3.2.3.1.2. Approximation optimale des surfaces.	69
3.2.3.1.3. Perspective.	71
3.2.3.2. Réduction de degré.	71
3.2.3.2.1. L'erreur d'approximation due à la réduction de degré.	72
3.2.3.2.2. Algorithme de réduction.	74
3.2.3.2.3. Algorithme de conversion par approximation.	77
3.2.4. Méthode de Watkins et Worsey.	81
3.2.4.1. Polynômes de Tchebycheff.	81
3.2.4.2. Algorithme de réduction.	82
3.2.4.3. Algorithme de conversion par approximation.	85
3.2.4.4. Conversion de base.	89
3.2.5. Méthode de Eck.	92
3.2.5.1. Réduction de degré et continuité géométrique.	92
3.2.5.2. Algorithme de conversion par approximation.	97
3.3. Analyse comparative des méthodes de conversion.	
3.3.1. Etude critique.	101
3.3.2. Comparaison analytique.	102
3.3.3. Comparaison expérimentale.	103
3.3.4. Conclusion.	104

4. Méthode proposée.	
4.1. Introduction.	106
4.2. Critères influants sur l'interpolation de courbes de Bézier. pour la réduction de degré.	107
4.2.1. Analyse des caractéristiques de la courbe et choix des points à interpoler.	107
4.2.2. Le paramétrage.	109
4.2.2.1. Le paramétrage d'origine.	109
4.2.2.2. Comparaisons de différents paramétrages.	112
4.2.3. Conclusion.	112
4.3. Recherche d'un paramétrage optimal.	112
4.3.1. Principe.	112
4.3.2. Définition et recherche de l'écart entre deux courbes 'proches'.	113
4.3.3. Minimisation de l'écart par modification de réseau de pôles.	113
4.3.4. Algorithme.	114
4.4. Réduction de courbes et surfaces.	115
4.4.1. Recherche du degré minimal par élévation de degré.	115
4.4.2. Respect des tangentes aux extrémités.	116
4.5. Résultats.	118
4.5.1. Courbe.	118
4.5.2. Surface.	119
4.6. Comparaison analytique.	121
4.7. Conclusion.	121
5. Application à l'échange de données entre systèmes	
5.1. Problématique.	122
5.2. Transmission de données.	122
5.3. Les principaux standards d'échanges.	123
5.3.1. IGES.	123
5.3.1.1. Architecture.	123
5.3.1.2. Performances et limitations.	123
5.3.2. VDA.	124
5.3.2.1. Architecture.	124
5.3.3. SET.	125
5.3.3.1. Architecture.	125
5.3.3.2. Performances et limitations.	126
5.3.4. STEP.	126
5.4. DXF.	127
5.5. Application.	128
5.6. Programme mis en œuvre.	131
5.6.1. OPENGL - Présentation.	131
5.6.1.2. OpenGL : Principe et bases.	132
5.6.1.3. OpenGL : Récapitulation.	133
5.6.2. Les différents module du système élaboré.	133
5.6.2.1. Module de construction.	134
5.6.2.2. Module de réduction.	135
5.6.2.3. Module de conversion.	135

Conclusion

Annexes

A Notions de bases.

B Théorème d'approximation de Weierstrass, Géométrie différentielle, continuité géométrique d'après Hoschek.

C Changement de Base Bézier - Tchebycheff.

Références bibliographiques

Table des illustrations

Chapitre 1

1.1.	Parabole : construction par interpolation linéaire répétée.	5
1.2.	Polynôme de Bernstein de degré 5.	7
1.3.	Courbe de Bézier non rationnelle de degré 3, construite par l'algorithme de De Casteljaou.	8
1.4.	(a) Polynômes rationnels de Bernstein de degré 5 avec $(w_0, \dots, w_5) = (1, 3, 1, 1, 1, 1)$, (b) Polynômes rationnels de Bernstein de degré 5 avec $(w_0, \dots, w_5) = (1, 1, 1, 0, 3, 1, 1)$	
1.5.	Courbes de Bézier rationnelles de degré 5 obtenues en appliquant les différentes valeurs à (a) w_1 (b) w_3 .	13
1.6.	(a) surface de Bézier non rationnelles de degré 3×4 , (b) surface de Bézier rationnelle de degré 3×4 avec $w_{12} = w_{22} = 20$.	13
1.7.	Interpolation bilinéaire : un paraboloïde hyperbolique est défini par les quatre points b_{ij} .	14
Figure ci contre : Algorithme de De Casteljaou direct pour les surfaces		14
1.8.	a) fonction de base B-spline quadratique définie par le vecteur nœuds $\{0, 0, 1 / 3, 1, 1, 1\}$, b) fonction de base B-spline cubique définie par le vecteur nœuds $\{0, 0, 0, 0, 1 / 4, 1 / 2, 3 / 4, 1, 1, 1, 1\}$, c) fonction de base Bernstein cubique définie par le vecteur nœuds $\{0, 0, 0, 0, 1, 1, 1, 1\}$.	17
1.9.	Algorithme de De Boor : exemple avec $n=3, L=2$	19
1.10.	Courbes B-spline rationnelle cubiques	24
1.11.	Surface B-spline non rationnelle de degré (2×3) , les vecteurs nœuds sont : $U = \{0, 0, 0, 1 / 3, 2 / 3, 1, 1, 1\}$, $V = \{0, 0, 0, 0, 1, 1, 1\}$.	26
1.12.	Surfaces B-spline rationnelle	27

Chapitre 2

2.1.	Courbe de Bézier non rationnelle $P(t)$ de degré 5 subdivisé en deux courbes $R(t)$ et $Q(t)$ de même degré.	29
2.2.	Courbe de Bézier rationnelle de degré 5 (de poids w_i) subdivisé en deux courbes rationnelles de même degré (de poids α_j et β_j)	30
2.3.	Courbe de Bézier non rationnelle de degré 3 élevée au degré 5.	31
2.4.	a) Courbe de Bézier non rationnelle de degré 3 réduite au degré 2 par les deux méthodes polygone 1 et 2, b) Courbe précédente de degré 3 réduite par la méthode du polygone moyen.	32

2.5.	Courbe de Bézier rationnelle de degré 6 réduite au degré 5 par la méthode du polygone moyen $W = \{1,1,1,0,8,0,5,0,8,1,2,1\}; W_r = \{1,1,304,0,516,0,516,1,304,1\}$.	
2.6.	Insertion de nœud 0.5 pour une courbe B-spline non rationnelle avec $T = \{0,0,0,0,0,2,0,8,1,1,1,1\}$.	35
2.7.	Insertion de nœud 0.8 pour une courbe B-spline rationnelle de degré 3 avec $T = \{0,0,0,0,0,2,0,6,1,1,1,1\}$ et $W = \{1,1,3,0,7,0,7,1,3,1,1\}; W_i = \{1,1,3,0,7,0,7,1,1,5,1,1,5,1\}$	37
2.8.	Suppression de nœud 0.4 pour une courbe B-spline non rationnelle de degré 4 , Avec $T = \{0,0,0,0,0,0,2,0,4,0,6,0,8,1,1,1,1,1\}$.	39
2.9.	Suppression de nœud 0.5 pour une courbe B-spline rationnelle de degré 4 , Avec $T = \{0,0,0,0,0,0,2,0,5,0,6,0,8,1,1,1,1,1\}$ et $W = \{1,1,2,0,8,1,3,0,7,1,3,0,8,1,2,1,1\};$ $W_s = \{1,1,2,0,72,1,648,1,6,0,8,1,2,1\}$	40
2.10.	Subdivision d'une courbe B-spline de degré 3 au point $t=0.5$ avec $T = \{0,0,0,0,0,3,0,6,1,1,1,1,1\}$ et $T_1 = \{0,0,0,0,0,6,1,1,1,1,1,1\}$ et $T_2 = \{0,0,0,0,0,2,1,1,1,1,1,1\}$	43
2.11.	Courbe B-spline rationnelle de degré 3 subdivisé en deux courbes à $t=0.5$ Avec $T = \{0,0,0,0,0,2,0,6,0,8,1,1,1,1,1\}$ et $T_1 = \{0,0,0,0,0,4,1,1,1,1,1,1\}; T_2 = \{0,0,0,0,0,2,0,6,1,1,1,1,1\}$ $W = \{1,1,2,0,8,1,0,8,1,2,1,1\}; W_1 = \{1,1,2,0,87,0,92,0,92\}; W_2 = \{0,92,0,93,0,93,0,8,1,2\}$	44
2.12.	Courbe B-spline non rationnelle de degré 3 transformée en 3 courbes de Bézier non rationnelles de degré 3 avec $T = \{0,0,0,0,0,3,0,7,1,1,1,1,1\}$	45

Chapitre 3

3.1.	Organigramme de conversion par approximation	51
3.1.a.	Courbe de Bézier de degré 6 réduite au degré 5	52
3.1.b.	Erreur de position sur la courbe précédente	53
3.1.c.	Variation de courbure , courbe précédente et sa courbe réduite	53
3.2.a.	Courbe de Bézier de degré 9 réduite au degré 8	56
3.2.b.	Erreur de position de la courbe précédente	56
3.2.c.	Variation de courbure courbe précédente et sa courbe précédente	57
3.3.a.	Courbe de degré 16 réduite en 5 courbes de degré 6 ($\epsilon_0 = 0.01$ et $k = 10$)	59
3.3.b.	Erreur de position le long de la courbe de degré 16	59
3.3.c.	Variation de l'erreur sur la courbure	60
3.4.	Courbe de degré 5 réduite au degré 4 par BEN1 et BEN2	62
3.5.	Courbe de degré 5 précédente réduite au degré 4 par BENm	62
3.6.a.	Courbe de degré 10 réduite en 8 courbes de degré 6 par l'algorithme(CONV-BEN), $\epsilon_0 = 0.01$.	64
3.6.b.	Erreur de position sur la courbe précédente	65
3.6.c.	Variation de l'erreur sur la tangente	65
3.6.d.	Variation de l'erreur sur la courbure après ajustement	66

3.7.	Paramétrisation des points donnés P_i Et l'erreur vectorielle D_i .	68
3.8.	Projection de l'erreur vectorielle D_i	68
3.9.	Projection de l'erreur vectorielle sur le plan tangent à la surface $Y(u,v)$ au point $Y(u_i, v_i)$.	71
3.9.a.	Courbe de Bézier de degré 7 réduite au degré 5	76
3.9.b.	Variation de l'erreur d'approximation	76
3.9.c.	Courbure des deux courbes	77
3.11.	Polynômes de Tchebycheff de degrés 5	81
3.12.a.	Courbe de degré 6 réduite au degré 5 par la méthode de Watkins et Worsey.	84
3.13.c.	Variation de l'erreur sur la tangente	85
3.13.a.	Courbe de degré 8 réduite en deux courbes de degré 6 par l'algorithme CONV-WW.	88
3.13.b.	Variation de l'erreur de position	88
3.13.c.	Variation de l'erreur sur la tangente	89
3.14.a.	Courbe de Bézier de degré 8 réduite au degré 7 par l'algorithme de ECK en respectant la continuité de courbure ($\alpha = 3$)	95
3.14.b.	Variation de l'erreur de position	96
3.14.c.	Variation des courbures	97
3.15.a.	Courbe de Bézier de degré 8 réduite en deux courbes de degré 7 par l'algorithme CONV-ECK	99
3.15.b.	Variation de l'erreur de position	99
3.15.c.	Variation de l'erreur sur la courbure	100

Chapitre 4

4.1.	Sélection : courbures max	107
4.2.	Variation de courbure	108
4.3.	Sélection : répartition régulière	108
4.4.	Degré 8	109
4.5.	Paramétrage	109
4.6.	Degré 3	110
4.7.	Paramétrage	110
4.8.	Courbe d'origine	110
4.9.	Paramétrage d'origine	110
4.10.	Méthode de Eck	111
4.11.	Paramétrage de Eck	111
4.12.	Modification manuelle	111
4.13.	Paramétrage manuelle	111
4.14.	Degré 10 initial	116
4.15.	Méthode de Eck	116
4.16.	Degré 3 Précision 0.0009	116
4.17.	Degré 4 Précision 0.0055	116

4.22.	Degré 7 initial	117
4.23.	Degré.4.continuité G^1	117
4.24.	Degré.4.continuité G^0	118
4.25.	Vue de face	118
4.26.	Vue de coté	118
4.27.	Paramétrage	119
4.28.	Vue de face	119
4.29.	Vue de coté	119
4.30.	Paramétrage	119
4.31.	Ecart	119
4.32.	Surface B-spline	120
4.33.	Pôles	120
4.34.	Bézier5*5	120
4.35.	Ecarts	120
4.36.	Bézier6*6	120
4.37.	Ecarts	120

Chapitre 5

5.1.	Modes de transfert de données	122
5.2.	Structure de changement de format pour ConverCurves	129
5.3.	Pssibilités de transfert de données à l'aide du système élaboré CONVER-CURVES	129
Tableau 5. Transformation possibles pour un transfert de données		130



Notations

$B_i^n(t)$	$i^{ème}$ fonction de Bernstein non rationnelle de degré n .
$B_{i,j}^{n,m}(u,v)$	fonction de Bernstein non rationnelle bi-variable.
C^α	continuité géométrique d'ordre α .
$G_i^n(t)$	$i^{ème}$ fonction de Bernstein rationnelle de degré n .
$G_{i,j}^{n,m}(u,v)$	fonction de Bernstein rationnelle bi-variable.
$N_i^k(t)$	fonction de base B-spline non rationnelle de d'ordre k .
$N_{i,j}^{k,k'}(u,v)$	fonction de base B-spline non rationnelle bi-variable.
$R_i^k(t)$	$i^{ème}$ fonction de base B-spline rationnelle d'ordre k .
$R_{i,j}^{k,k'}(u,v)$	fonction de base B-spline rationnelle bi-variable.
$S(u,v)$	surface paramétrique modèle de Bézier ou B-spline.
T	vecteur nœuds des courbes B-spline.
$T_i(x)$	polynômes de Tchebycheff de degré i .
t	paramètre curviligne d'une courbe, généralement $t \in [0,1]$
u,v	paramètres curvilignes d'une surface de Bézier ou B-spline,
généralement $(u,v) \in [0,1]^2$.	
w_i	poids d'une courbe rationnelle.
P_i, Q_i, R_i, S_i, b_i	pôles d'une courbe paramétrique.
$P_{i,j}, Q_{i,j}, R_{i,j}, S_{i,j}, b_{i,j}$	réseau de pôles d'une surface paramétrique.
M_i	point d'une courbe paramétrique.
E^3	espace affine.

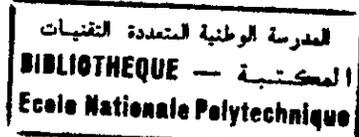
Listes des abréviations

C.A.O	Conception Assisté par Ordinateur
F.A.O	Fabrication Assisté par Ordinateur
C.F.A.O	Conception et Fabrication Assisté par Ordinateur
NURBS	Non Uniform Rational B-Spline
IGES	International Graphic Exchange Specification
SET	Système d'Echange et de Transfert de données géométriques.
STEP	Standard for Exchange of Product model data.
VDA/FS	Verband der Deutschen Ingenieure/ FlächenSchnittstellen exchange in
Automobil-Industrie.	
DXF	Data exchange Format.
DES	format des fichiers de CONVER.

المدرسة الوطنية المتعددة الفنون
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Introduction

Introduction



L'un des problèmes essentiels rencontrés en C.F.A.O. (conception et fabrication assistée par ordinateur) est la communication et l'échange de données entre différents systèmes de C.F.A.O. Les systèmes de C.F.A.O. imposent souvent la modification du modèle pour l'adapter aux contraintes du système de réception notamment au niveau des formes libres où chaque modeleur géométrique définit ses propres limitations pour chaque type de description. Ainsi, la réduction de degré des courbes et surfaces paramétriques est une étape nécessaire dès que le degré maximal imposé par le modeleur cible est inférieur à celui du modeleur émetteur.

Les systèmes de C.A.O. (conception assistée par ordinateur) des courbes et des surfaces reposent sur plusieurs modèles mathématiques. Ces différents modèles (Bézier, B-spline, Coons, NURBS,...) sont fondés sur des fonctions de base polynomiales différentes et par conséquent tout transfert de données entre deux systèmes de modèles différents nécessite un changement de base d'une part, D'autre part, la conception des courbes et des surfaces à l'aide des systèmes de C.A.O. exige souvent des modèles polynomiaux de degré élevé. Cela est justifié aisément par la flexibilité recherchée par l'utilisateur dans la conception des formes complexes et le respect des contraintes de conception géométrique, le calcul des courbes définies par l'intersection de deux surfaces et des courbes d'offset... etc. l'utilisateur peut, en effet, agir sur la forme de la courbe sans modifier ses caractéristiques géométriques.

Compte tenu de la multiplicité des systèmes et des fournisseurs, l'échange de données en C.F.A.O. recouvre en fait l'échange entre systèmes hétérogènes et l'échange entre différents logiciels applicatifs puisque chaque firme est souvent équipée en fonction de ses besoins sans se préoccuper de la redondance des données ou des tâches de ressaisie et de l'archivage de données, car les versions successives d'un même logiciel ou la nécessité de remplacer un système pose le problème de compatibilité. Il est donc plus rationnel de songer à des formats standards d'échange de données. Aussi un standard d'échange pleinement satisfaisant doit être indépendant du matériel, du logiciel et du mode opératoire de l'exécutant. En outre, Il doit être évolutif pour assurer la pérennité des échanges. A ce jour, divers standards d'échange de type IGES, SET, VDA, STEP, DXF sont normalisés et sont très souvent utilisés en pratique.

Dans le contexte de notre étude, l'objectif fixé se rapporte à la mise en œuvre d'une application à l'échange de données entre systèmes de C.F.A.O. (programme mis en œuvre, et Autocad 14), basé sur une étude comparative des méthodes de réduction de degré des courbes de Bézier non rationnelles qui nous permet de choisir une méthode optimale de réduction de degré présentant des caractéristiques de précision et de rapidité requise. Finalement à la base de cette étude comparative, nous proposons une méthode simple et efficace adaptée au transfert de données entre systèmes de bases et degrés différents. Ainsi, nous allons structurer ce document en cinq chapitres organisés de la manière suivante :

Dans le chapitre 1, nous allons, de prime abord, présenter un aperçu général des fondements mathématiques et les caractéristiques géométriques des courbes et des surfaces de Bézier et B-spline.

Dans le chapitre 2, nous allons présenter toutes les transformations nécessaires à la conversion par approximation des courbes de Bézier et B-spline, sous forme d'algorithmes destinés à l'application, et à l'extension de la réduction de degré des modèles rationnels et à la base B-spline.

Dans la première partie du chapitre 3, nous allons présenter le principe général de conversion par approximation, sous forme d'un organigramme. La deuxième partie est une synthèse bibliographique approfondie des principales méthodes de réduction de degré des courbes de Bézier non rationnelles, où sont présentés le principe et les algorithmes de réduction et de conversion par approximation. Ces différentes méthodes de réduction nous conduisent à faire une étude comparative présentée dans la dernière partie, basé sur des résultats pratiques, ainsi sur les structures algorithmiques, afin de choisir une méthode de réduction optimale.

Dans le chapitre 4, nous allons introduire la notion de paramétrage optimal qui est souvent négligé par les autres méthodes de réduction, en assurant une meilleure approximation par l'application d'un algorithme itérative de minimisation des écarts normaux entre deux courbes, les résultats de cette méthode sont présentés dans la cinquième partie de ce chapitre.

L'application à l'échange de données entre systèmes de C.F.A.O. fait l'objet du chapitre 5. Ce dernier explique en première étape, le processus de transmission de données entre systèmes via des fichiers neutre, ensuite nous présentons les principaux standards d'échanges normalisés, ainsi leur architecture, performances et limitations. Par la suite, nous adoptons le format DXF-R12 à notre application, et nous présentons l'architecture générale du programme mis en œuvre avec ces différents modules.

Une conclusion générale clôture notre étude en présentant l'intérêt du paramétrage sur les courbes et les surfaces en qualité d'approximation et temps de calcul, qui est négligé par les autres méthodes de réduction, ainsi nous présentant quelques perspectives d'extension de la méthode proposée qui peuvent être envisagées dans des études ultérieures.

CHAPITRE 1

Courbes et surfaces: modèles de Bézier et B-Spline

1. Courbes et surfaces : Modèles de Bézier et B-spline

1.1. Introduction

Le modèle de données géométriques représente un ensemble de données mathématiques caractérisant une forme dans l'espace. De nos jours, les systèmes de CFAO sont encore construits autour d'un modèleur essentiellement géométrique bien que la tendance actuelle soit de proposer des modèles de produit à forte sémantique intégrant ces caractéristiques géométriques, mais aussi des caractéristiques technologiques (features définissant des surfaces fonctionnelles), de cotation, de fabrication et administratives. L'aspect modèleur géométrique n'est donc pas novateur, mais il reste le cœur d'un modèle produit dont il est nécessaire de connaître ses spécificités.

De nombreux modèles mathématiques peuvent tenter de traduire la forme qu'on cherche à concevoir. Ils caractérisent donc par :

leur richesse sémantique ;

le fait qu'ils modélisent la forme de façon exacte ou approchée ;

le fait qu'ils soient facilement créés et modifiés (on parle d'opérateurs spécifiques à un modèle)

le fait qu'ils soient plus ou moins adaptés à une application donnée (visualisation, usinage, cinématique, dynamique, calcul de structures...)

Cinq grandes familles de modèleurs existent ; ils sont cités dans un ordre croissant de richesse sémantique et dans l'ordre historique d'apparition :

modèleurs 2D ;

modèleurs filaires 3D ;

modèleurs surfaciques ;

modèleurs volumiques ;

modèleurs sous contraintes ;

Actuellement, tous les modèleurs géométriques utilisent en plus d'une représentation volumique, un noyau surfacique destiné principalement à la définition géométrique des formes libres dont on ne connaît pas de formulation analytique. Le choix d'un modèle de représentation se pose dès qu'il s'agit d'utiliser, sur un ordinateur, un ensemble de données établies expérimentalement (scannérisation d'une surface à partir d'une maquette physique) ou par calcul.

Les contraintes posées par la conception de formes libres nécessitent l'utilisation de courbes et des surfaces paramétriques dont la particularité est l'invariance sous des transformations élémentaires. Qu'il s'agisse de la modélisation de données existantes sous forme de points ou de l'expression d'un sens esthétique pour une forme, on peut citer les contraintes suivantes :

le modèle doit appartenir à une classe de fonctions continues, suffisamment dérivables pour autoriser une manipulation mathématique ;

le modèle doit être facile à manipuler et à stocker dans un ordinateur ;

l'évaluation d'un point courant par ce modèle doit se faire avec un nombre restreint d'opérations ;

le modèle doit permettre de mettre au point des algorithmes d'intersection à coût faible ;

la manipulation des paramètres du modèle par le biais d'une interface doit pouvoir se faire de façon intuitive ;

le modèle doit permettre de couvrir une grande variété de formes .

Parmi les formes qui satisfaisant les contraintes cités précédemment , les formes polynomiales paramétriques , dont les fonctions de base fréquemment utilisées dans les déférent systèmes de CFAO sont Bernstein-Bézier , Schoenberg- B-spline , et Hermite – Coons . Ces fonctions de base ont une grande importance sur le comportement géométrique des courbes et des surfaces .

pour ce présent travail on se limite aux deux modèles paramétriques les plus utiliser , Bézier et B-spline .

les paragraphes suivants donnent plus de détails sur les qualités géométriques de ces modelés de courbes suivant leurs formes rationnelle et non rationnelle.

1.2. Courbes et surfaces de Bézier

Une des techniques qui à été , et qui reste une des plus employées dans le monde et la technique des courbes et surfaces à pôles .

L'idée fondamentales de cette théorie est de décrire une courbe uniquement avec les coordonnées de quelques points particuliers , appelés pôles de la courbe . L'équation d'un point courant d'une courbe à pôles , obtenue à partir des coordonnées des pôles , peut s'écrire sous une forme polynomiales , de degré (n-1) si il y a n pôles .

Les caractéristiques de cette technique sont les suivantes :

- la définition de la courbe à pôles est unique et indépendante du repère dans lequel on se déplace et le polynôme est invariant par transformation linéaire (ce qui permis l'utilisation de cette technique pour la définition de pièces) ;
- le calcul du polynômes est rapide ;
- l'information en mémoire dans l'ordinateur est compacte ;
- l'extrapolation de la courbe est possible ;
- les défauts majeurs sont liés à l'aptitude à l'oscillations des courbes à pôles , que l'on minimise en n'utilisons que des courbes de degré faibles .

1.2.1. Algorithme de De Casteljau

l'algorithme décrit dans ce chapitre est probablement le plus fondamental dans le domaine de la conception des courbes et des surfaces , et est cependant étonnamment simple . son principal intérêt et de mêler également la géométrie et l'algèbre : une construction très intuitive mène à une puissante théorie .

1.2.1.1. Paraboles

Nous donnons une construction simple pour générer une parabole ; la généralisation mènera ensuite naturellement aux courbes de Bézier . Soit b_0, b_1, b_2 trois points quelconques de

l'espace E^3 , et soit $t \in \mathfrak{R}$.

Construisons : $b_0^1(t) = (1-t)b_0 + tb_1$

$$b_1^1(t) = (1-t)b_1 + tb_2$$

$$b_0^2(t) = (1-t)b_0^1(t) + tb_1^1(t) .$$

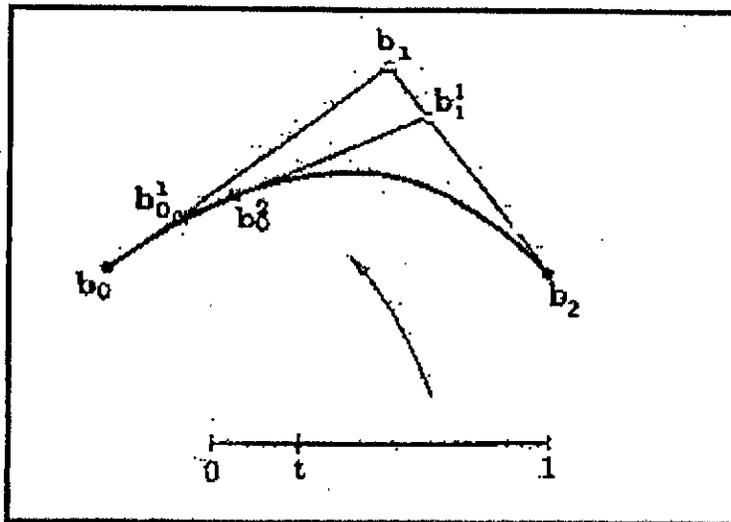


Figure 1.1. parabole : construction pour interpolation linéaire répétée.

Nous intégrons les deux premières équations dans la troisième, nous obtenons :

$$b_0^2(t) = (1-t)^2 b_0 + 2t(1-t)b_1 + t^2 b_2$$

Ceci est expression quadratique en t (l'indice supérieur caractérise le degré), et donc $b_0^2(t)$ décrit une parabole lorsque t varie de $]-\infty, +\infty[$, nous notons cette parabole b^2 .

Cette construction consiste en une interpolation linéaire répétée ; (figur 1.1).

Pour $t \in [0,1]$, $b^2(t)$ est à l'intérieur du triangle formé par b_0, b_1, b_2 , en particulier

$$b^2(0) = b_0, b^2(1) = b_2.$$

Les rapports de proportionnalité sont de la

$$\text{forme : } \text{ratio}(b_0, b_0^1, b_1) = \text{ratio}(b_1, b_1^1, b_2) = \text{ratio}(b_0^1, b_0^2, b_1^1) = \frac{t}{1-t}$$

Donc notre construction d'une parabole est affinement invariante parce que l'interpolation linéaire par morceaux est affinement invariante (voir annexe A).

Nous remarquons également qu'une parabole est une courbe plane, ce qui est du fait que $b^2(t)$ est toujours une combinaison barycentrique de trois points.

Nous pouvons finalement établir à partir de la géométrie analytique un théorème étroitement lié à la construction de notre parabole. soit a, b, c trois points distincts sur une parabole. soit la tangente en b , intersectant les tangentes en a et c , respectivement en e et f . soit d l'intersection des tangentes en a et c . alors $\text{ratio}(a, e, d) = \text{ratio}(e, b, f) = \text{ratio}(d, f, c)$.

Ce théorème des trois tangentes décrit une propriété des paraboles ; l'algorithme de De Casteljau peut être vu comme l'équivalent constructif de ce théorème.

1.2.1.2. Algorithme de De Casteljau

La constructions d'une parabole vue ci-dessus peut être généralisée pour engendrer une courbe polynomiale d'un degré n arbitraire :

L'algorithme de De Casteljau :

Entrées : $b_0, b_1, \dots, b_n \in E^3$ et $t \in \mathfrak{R}$, posons

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t) \quad \begin{cases} r = 1, \dots, n \\ i = 0, \dots, n-r \end{cases}$$

et $b_i^0(t) = b_i$.

Alors $b_0^n(t)$ est le point de paramètre t sur la courbe de Bézier b^n .

Le polygone P formé par les pôles b_0, b_1, \dots, b_n est appelé polygone de contrôle de la courbe de Bézier.

Les coefficients intermédiaires $b_i^r(t)$ peuvent être commodément écrits sous forme d'un Tableau triangulaire de points, appelé schéma de De Casteljaou :

$$\begin{array}{l} b_0 \\ b_1 \quad b_0^1 \rightarrow b_0^1(t) = (1-t)b_0 + tb_1 \\ b_2 \quad b_1^1 \quad b_0^2 \rightarrow b_1^1(t) = (1-t)b_1 + tb_2 \rightarrow b_0^2(t) = (1-t)b_0^1(t) + tb_1^1(t) = (1-t)^2 b_0 + 2t(1-t)b_1 + t^2 b_2 \\ b_3 \quad b_2^1 \quad b_1^2 \quad b_0^3 \end{array}$$

1.2.2. Polynômes de Bernstein

Les courbes de Bézier peuvent être définies par un algorithme récursif, et c'est de cette manière que De Casteljaou les a d'abord développés. Il est également nécessaire, cependant d'en avoir une représentation explicite, c'est à dire d'exprimer les courbes de Bézier sous forme d'une formule non récursive plutôt qu'en terme d'algorithme. Ceci facilitera considérablement le développement par la suite.

Nous exprimons les courbes de Bézier en terme de polynômes de Bernstein définis explicitement par :

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad (1)$$

ou les coefficients du binôme sont donnés par

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{si } 0 \leq i \leq n. \\ 0 & \text{sinon.} \end{cases}$$

Une de leurs propriétés importantes est qu'ils satisfont la relation de récurrence suivante :

$$B_i^n(t) = (1-t) \cdot B_i^{n-1}(t) + t \cdot B_{i-1}^{n-1}(t)$$

avec

$$\begin{aligned} B_0^0(t) &\equiv 1 \\ B_i^n(t) &\equiv 0 \quad \text{pour } i \notin \{0, \dots, n\} \end{aligned}$$

les polynômes $B_i^n(t)$ possèdent les propriétés importantes suivantes dans l'intervalle $t \in [0, 1]$:

- Non négativité : les polynômes $B_i^n(t) \geq 0, \forall t \in [0, 1], \forall n, \forall i \in \{0, \dots, n\}$.

- Partition unité : $\sum_{i=0}^n B_i^n(t) = 1 \quad \forall t \in [0,1]$ (2)
- Les Polynômes de Bernstein dans la base canonique :
 Dans l'espace P_n , qui est de dimension $n+1$, des polynômes de degré $\leq n$, les polynômes $\{1, t, t^2, \dots, t^n\}$ constituent une base appelé base canonique.
 Les polynômes de Bernstein peuvent s'écrire dans cette base :

$$B_i^n(t) = \sum_{k=0}^n (-1)^{i-k} \binom{n}{i} \binom{i}{k} t^i$$

par conséquent les polynômes $B_i^n(t)$ ou n finie, forme une base, appelé base de Bernstein, des polynômes de degré n . ($B_i^n(t)$ partie génératrice de l'espace P_n).

- Symétrie : $B_i^n(t) = B_{n-i}^n(1-t) \quad \forall t \in [0,1]$.
- Extremum : $B_i^n(t)$ atteint exactement un seul maximum à $t = i/n$:

$$\text{Max}_{t \in [0,1]} B_i^n(t) = B_i^n(i/n)$$

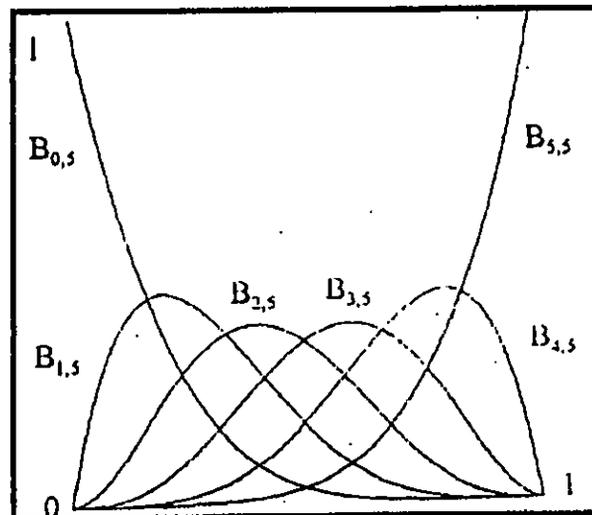


Figure 1.2. polynômes de Bernstein de degré 5

1.2.3. courbes de Bézier non rationnelles

Une courbe de Bézier non rationnelle de degré n est définie par :

$$P(t) = \sum_{i=0}^n P_i \cdot B_i^n(t) \quad \text{avec } t \in [0,1] \quad (3)$$

ou $B_i^n(t)$, $i \in \{0, \dots, n\}$ la base de Bernstein, et P_i Pôles formant le polygone de contrôle de la courbe.

Nombre de pôles = degré + 1 ;

Nous sommes maintenant prêts à voir pourquoi les polynômes de Bernstein sont importants pour le développement des courbes de Bézier.

Les points de De Casteljaou intermédiaires b_i^r peuvent être exprimé en terme de polynômes de Bernstein de degré r :

$$b_i^r(t) = \sum_{j=0}^r b_{i+j} B_j^r(t) \quad \begin{cases} j \in \{0, \dots, r\} \\ i \in \{0 \dots n-r\} \end{cases}$$

pour le cas $r = n$ le point de De Casteljaou correspondant le point sur la courbe de Bézier

non rationnelle : $b^n(t) = b_0^n(t) = \sum_{j=0}^n b_j \cdot B_j^n(t)$ (figure 1.3).

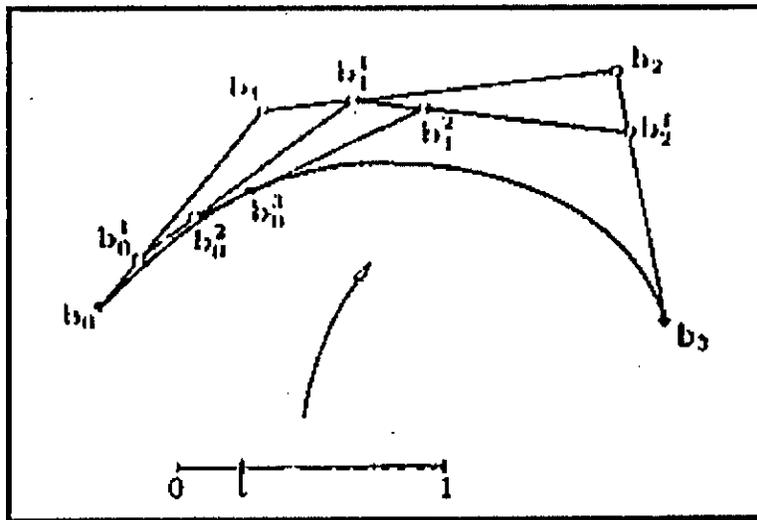


figure 1.3. courbe de Bézier non rationnelle de degré 3, construite par l'algorithme de De Casteljaou .

1.2.3.1 Dérivation d'une courbe de Bézier

la dérivée d'une courbe de Bézier non rationnelle est défini par : [Farin'92]

$$\frac{d}{dt} P(t) = n \cdot \sum_{i=0}^{n-1} \Delta P_i \cdot B_i^{n-1}(t); \quad \Delta P_i \in \mathbb{R}^3 \quad (3.1)$$

$\Delta P_i = P_{i+1} - P_i$ représente l'opérateur de différence.

la dérivée d'une courbe de Bézier non rationnelle est donc une autre courbe de Bézier non rationnelle ,appelé Hodographe d'ordre 1 de la courbe $P(t)$,obtenue par l'application d'opérateur différence sur le polygone de contrôle original .

1.2.3.2. Dérivées d'ordre supérieur

la dérivées d'ordre r d'une courbe de Bézier non rationnelle est défini pat :

$$\frac{d^r}{dt^r} P(t) = \frac{n!}{(n-r)!} \cdot \sum_{i=0}^{n-r} \Delta^r P_i \cdot B_i^{n-r}(t) \quad (3.2)$$

avec :

$$\Delta^0 P_i = P_i$$

$$\Delta^1 P_i = P_{i+1} - P_i$$

$$\Delta^2 P_i = P_{i+2} - 2.P_{i+1} - P_i$$

$$\text{d'ou } \Delta^r P_i = \Delta^{r-1} P_{i+1} - \Delta^{r-1} P_i = \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} . P_{i+j}$$

la démonstration de (3.2) se fait par l'application successive de (3.1)

1.2.3.3 Propriétés des courbes de Bézier non rationnelles

Les courbes de Bézier non rationnelles possèdent les propriétés géométriques suivantes :

- invariance affine : les combinaisons barycentriques sont invariantes par cartes affines, (2) donne la vérification de cette propriété, donc une carte affine est appliqué à la courbe si elle est appliqué aux pôles.
- invariance par transformations affine du paramètre : Algébriquement cette propriété s'écrit

$$\sum_{i=0}^n P_i B_i^n(t) = \sum_{i=0}^n P_i B_i^n\left(\frac{u-a}{b-a}\right) \quad t, u \in [a, b]$$

- Enveloppe convexe : elle est due à ce que pour $t \in [0, 1]$, les polynômes de Bernstein sont positifs. Leur somme égale à 1 selon (2), donc la courbe de Bézier est toujours à l'intérieur du polygone convexe formé par les pôles P_i .
- Interpolation des extrémités : ce si est une conséquence des identités et (2).

$$B_i^n(0) = 1 \quad \text{si et seulement si } i = 0$$

$$B_i^n(1) = 1 \quad \text{si et seulement si } i = n$$

donc la courbe de Bézier passe par le premier et le dernier pôles en étant tangente

au polygone avec : $\begin{cases} P'(0) = n.(P_1 - P_0) \\ P'(1) = n.(P_n - P_{n-1}) \end{cases}$

- Invariance par combinaisons barycentriques : la construction d'une courbe de Bézier à partir d'un polygone de contrôle laisse invariantes les combinaisons barycentriques : pour $\alpha + \beta = 1$ nous

$$\text{avons : } \sum_{j=0}^n (\alpha . b_j + \beta . c_j) . B_j^n(t) = \alpha . \sum_{j=0}^n b_j . B_j^n(t) + \beta . \sum_{j=0}^n c_j . B_j^n(t)$$

Autrement dit, nous pouvons construire la moyenne pondéré de deux courbes, en prenons la moyenne pondéré des pôles correspondantes, puis calculons la courbe.

- Contrôle pseudo-local : le polynôme de Bernstein $B_i^n(t)$ n'a qu'un maximum, qu'il atteint en $t = i/n$, donc si nous déplaçons un des pôles du polygone de contrôle, alors toute la courbe est affecté par ce changement et principalement dans la région autour des points où le paramètre prend la valeur $t = i/n$.
- Propriété de symétrie : lorsque nous changeons t par $(1-t)$ la courbe est parcourue dans le sens inverse :

$$\sum_{i=0}^n P_i B_i^n(t) = \sum_{i=0}^n P_{n-i} B_{n-i}^n(1-t)$$

- Propriété de la diminution de la variation : la courbe de Bézier n'a pas plus d'intersections avec un plan quelconque que le polygone de contrôle. (voir Annexe A).
- Propriété de l'hodographe : si l'hodographe d'ordre r d'une courbe $P(t)$ de degré n ($r < n$), alors il existe une courbe $c(t)$ de degré r équivalente à $P(t)$.

1.2.3. Courbes de Bézier rationnelles

L'idée principale du modèle rationnelle, est la projection d'une courbe de Bézier non rationnelle définie dans l'espace des coordonnées homogènes (4D) dans l'espace (3D). Plus précisément nous associons un vecteur poids w_i pour chaque pôles P_i , et construire des nouveaux pôles dans (4D) définies par [Tran'97][Farin'92]:

$$\bar{P}_i = \begin{cases} (w_i, w_i \cdot x_i, w_i \cdot y_i, w_i \cdot z_i)^T = (w_i, w_i \cdot P_i)^T & \text{pour } w_i \neq 0, \\ (0, x_i, y_i, z_i)^T = (0, P_i)^T & \text{pour } w_i = 0 \end{cases}$$

Noté que pour $w_i = 0$ alors \bar{P}_i est un pôle (point de contrôle) à l'infini.

D'où la formule paramétrique correspondante à la courbe de Bézier dans l'espace (4D) est :

$$\bar{P}(t) = \sum_{i=0}^n \bar{P}_i \cdot B_i^n(t)$$

Après projection dans (3D) nous obtenons la formule paramétrique d'une courbe de Bézier rationnelle :

$$P(t) = \frac{\sum_{i=0}^n w_i \cdot P_i \cdot B_i^n(t)}{\sum_{i=0}^n w_i \cdot P_i} \quad \text{pour } w_i \neq 0 \quad (4)$$

Si quelques w_i sont négatifs, des singularités peuvent survenir ; nous utilisons par conséquent que des w_i non négatifs.

nous pouvons écrire $P(t)$ sous la forme :

$$\begin{cases} P(t) = \sum_{i=0}^n P_i G_i^n(t) \\ G_i^n(t) = \frac{w_i \cdot B_i^n(t)}{\sum_{i=0}^n w_i \cdot B_i^n(t)} \text{ avec } w_i > 0 \end{cases} \quad (5)$$

ou $G_i^n(t)$ fonctions de Bernstein rationnelles, les poids w_i sont utilisés comme des paramètres de forme.

Si $w_i = 1$ pour tous i , nous obtenons la courbe de Bézier non rationnelle standard.

Si $w_i \rightarrow \infty$ alors $P(t) \rightarrow P_i$ (polygone de contrôle).

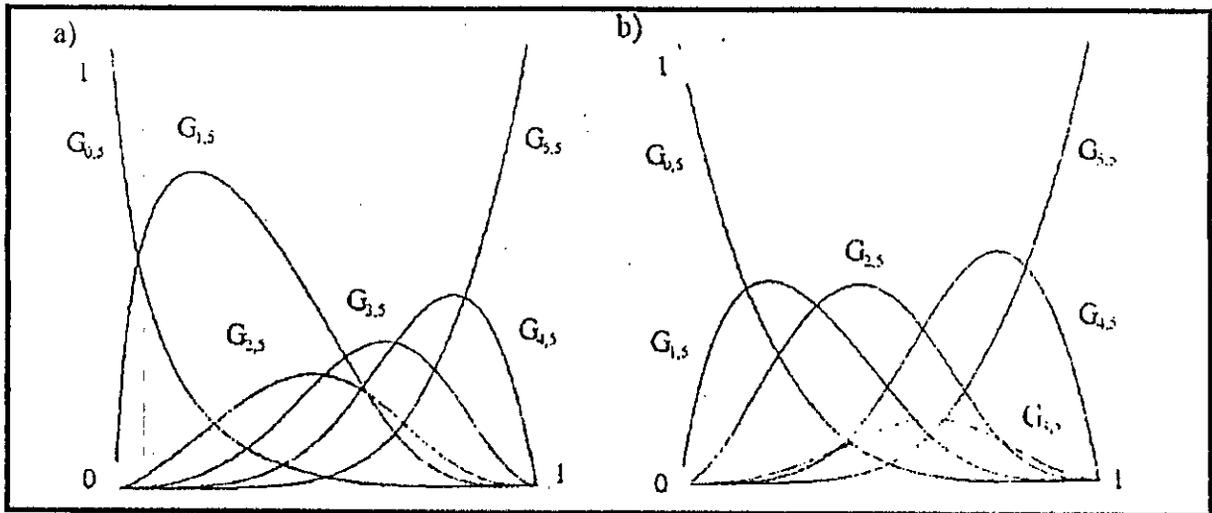


Figure 1.4. (a) polynômes rationnels de Bernstein de degré 5 avec $(w_0, \dots, w_5) = (1, 3, 1, 1, 1, 1)$
(b) polynômes rationnels de Bernstein de degré 5 avec $(w_0, \dots, w_5) = (1, 1, 1, 0.3, 1, 1)$

Les propriétés analytiques des fonctions $G_i^n(t)$ sont :

- Non négativité : les polynômes $G_i^n(t) \geq 0, \forall t \in [0,1], \forall n, \forall i \in \{0, \dots, n\}$.
- Partition unité : $\sum_{i=0}^n G_i^n(t) = 1, \forall t \in [0,1]$
- Symétrie : $G_i^n(t) = G_{n-i}^n(t) \quad \forall t \in [0,1]$.
- Extremum : $G_i^n(t)$ atteint exactement un seul maximum à $t = i/n$:

$$\max_{t \in [0,1]} G_i^n(t) = G_i^n(i/n)$$
- Si tous les $w_i = 1$ alors $G_i^n(t) = B_i^n(t) \quad \forall t \in [0,1]$

Ces propriétés proviennent de la définition des fonctions $G_i^n(t)$ et des propriétés analogue des propriétés $B_i^n(t)$.

Comme pour les fonctions de base les propriétés des courbes de Bézier rationnelles sont semblables à celles des courbes non rationnelles :

- invariance affine : les combinaisons barycentriques sont invariantes par cartes affines, donc une carte affine est appliqué à la courbe si elle est appliqué aux pôles.
- invariance par transformations affine du paramètre : Algébriquement cette propriété s'écrit

$$\sum_{i=0}^n P_i G_i^n(t) = \sum_{i=0}^n P_i G_i^n\left(\frac{u-a}{b-a}\right) \quad t, u \in [a, b]$$

- Enveloppe convexe : elle est due à ce que pour $t \in [0,1]$, les fonctions de base $G_i^n(t)$ sont positifs. Leur somme égale à 1, donc la courbe de Bézier est toujours à l'intérieur du polygone convexe formé par les pôles P_i .

•

- Interpolation des extrémités : la courbe de Bézier passe par le premier et le dernier pôles en étant tangente au polygone

avec :
$$\begin{cases} P'(0) = n.(P_1 - P_0) \\ P'(1) = n.(P_n - P_{n-1}) \end{cases}$$

- Invariance par combinaisons barycentriques : la construction d'une courbe de Bézier à partir d'un polygone de contrôle laisse invariante les combinaisons barycentriques : pour $\alpha + \beta = 1$ nous

avons :
$$\sum_{j=0}^n (\alpha.b_j + \beta.c_j).G_j^n(t) = \alpha.\sum_{j=0}^n b_j.G_j^n(t) + \beta.\sum_{j=0}^n c_j.G_j^n(t)$$

Autrement dit, nous pouvons construire la moyenne pondérée de deux courbes, en prenons la moyenne pondérée des pôles correspondantes, puis calculons la courbe.

- Contrôle pseudo-local : le polynôme de Bernstein $G_i^n(t)$ n'a qu'un maximum, qu'il atteint en $t = i/n$, donc si nous déplaçons un des pôles du polygone de contrôle, alors toute la courbe est affectée par ce changement et principalement dans la région autour des points où le paramètre prend la valeur $t = i/n$.

Le polygone formé par les P_i représente une approximation globale de la courbe.

- Propriété de symétrie : lorsque nous changeons t par $(1-t)$ la courbe est parcourue dans le sens inverse :

$$\sum_{i=0}^n P_i G_i^n(t) = \sum_{i=0}^n P_{n-i} G_{n-i}^n(1-t)$$

- Propriété de la diminution de la variation : la courbe de Bézier n'a pas plus d'intersections avec un plan quelconque que le polygone de contrôle.
- Propriété de l'hodographe : si l'hodographe d'ordre r d'une courbe $P(t)$ de degré n ($r < n$) est un point, alors il existe une courbe $c(t)$ de degré r équivalente à $P(t)$.
- Si nous augmentons un w_i , la courbe est tirée vers le P_i correspondant, si nous laissons tous les poids tendre vers l'infini au même rythme, nous nous n'approcherons pas le polygone de contrôle puisqu'un facteur commun (même grand) dans les poids importe peu, et si w_i décroît la courbe s'éloigne du pôle P_i , alors on peut utiliser les w_i comme des paramètres de forme.

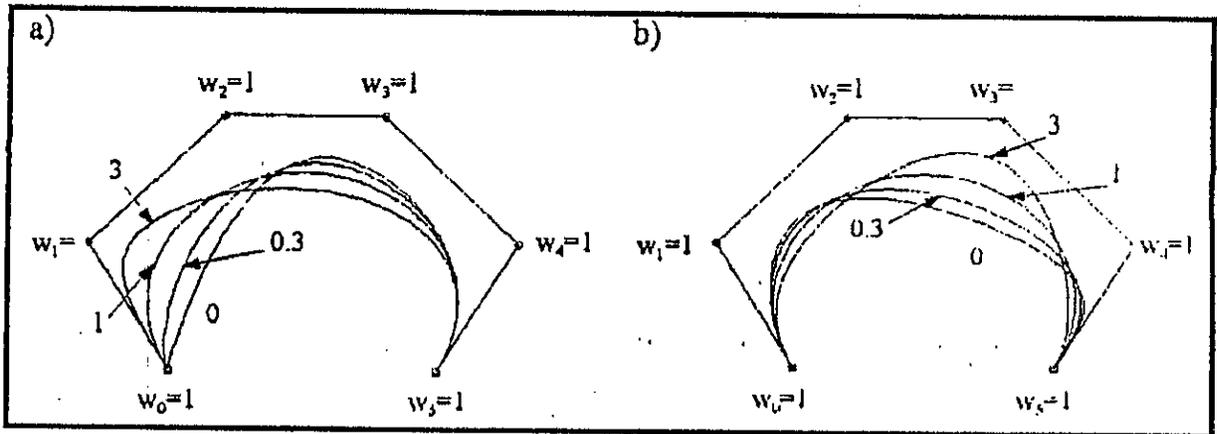


Figure 1.5. courbes de Bézier rationnelles de degré 5 obtenues en appliquant différentes valeurs (a) w_1 (b) w_3 .

1.2.4. Surfaces de Bézier non rationnelle

Une surface de Bézier non rationnelle de degré (m,n) est définie par :

$$S(u,v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \cdot B_i^n(u) \cdot B_j^m(v) \quad (5)$$

avec $u, v \in [0,1]$ paramètres curvilignes suivant les deux directions, et P_{ij} réseau de contrôle de la surfaces, arrangées dans une matrice rectangulaire.

$B_i^n(u), B_j^m(v)$ les polynômes de Bernstein données par l'équation(1)

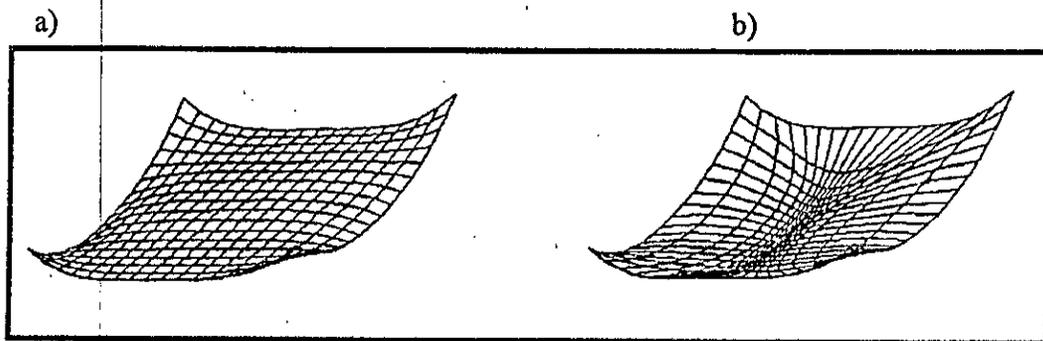


figure 1.6. (a) surface de Bézier non rationnelles de degré 3×4 (b) surface de Bézier rationnelles de degré 3×4 avec $w_{12} = w_{22} = 20$.

1.2.4.1. Interpolation bilinéaire

soient $b_{0,0}, b_{0,1}, b_{1,0}, b_{1,1}$ quatre points distincts dans E^3 . L'ensemble de tous les points $S \in E^3$ de la forme

$$S(u, v) = \sum_{i=0}^1 \sum_{j=0}^1 b_{ij} \cdot B_i^1(u) \cdot B_j^1(v) \quad (6)$$

est appelé un paraboloides hyperbolique passant par les quatre points $b_{i,j}$. Dans la forme matricielle :

$$S(u, v) = \begin{bmatrix} 1-u & u \end{bmatrix} \cdot \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} \cdot \begin{bmatrix} 1-v \\ v \end{bmatrix} \quad (7)$$

Puisque (6) est linéaire à la fois en u et en v et puisqu'elle interpole les points d'entrée, La surface S est appelée l'interpolant bilinéaire.

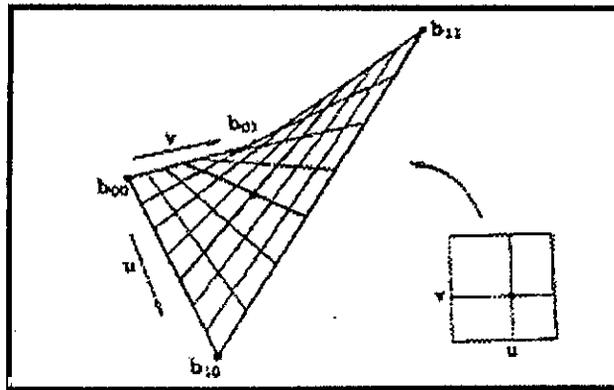


Figure 1.7. interpolation bilinéaire : un paraboloides hyperbolique est définie par les quatre points b_{ij}

1.2.4.2. Algorithme de De Casteljaou Direct

les courbes de Bézier peuvent être obtenues par application répétée de l'interpolation linéaire. Nous obtenons maintenant les surfaces par application répétée d'interpolation bilinéaire.

Données : $\{b_{i,j}\}_{i,j=0}^{n,n}$ et $0 \leq u, v \leq 1$

$$b_{i,j}^{r,r} = \begin{bmatrix} 1-u & u \end{bmatrix} \cdot \begin{bmatrix} b_{i,j}^{r-1,r-1} & b_{i,j+1}^{r-1,r-1} \\ b_{i+1,j}^{r-1,r-1} & b_{i+1,j+1}^{r-1,r-1} \end{bmatrix} \cdot \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

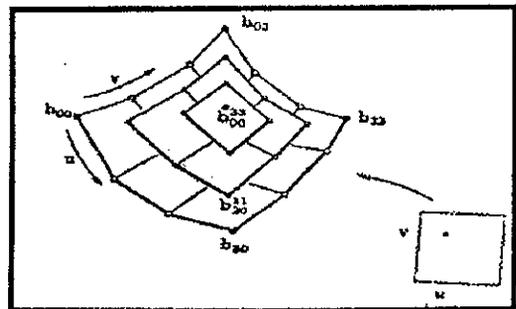
$$r = 1, \dots, n$$

$$i, j = 0, \dots, n-r$$

et $b_{i,j}^{0,0} = b_{i,j}$ réseau (matrice) de contrôle.

Alors $b_{0,0}^{n,n}$ est le point de paramètre (u, v) sur la surface de Bézier $b^{n,n}$

Figure ci contre : algorithme de De Casteljaou direct pour les surfaces : le point de la surface est obtenue par interpolation linéaire répétée



1.2.4.3. Approche produit tensoriel [Farin'92][Hoschek'90]

définition : une surface est le lieu d'une courbe déplacée à travers l'espace en changeant simultanément de forme.

Soit la courbe initiale une courbe de Bézier non rationnelle de degré m :

$$b^m(u) = \sum_{i=0}^m b_i \cdot B_i^m(u) \quad \text{avec} \quad u \in [0,1]$$

ou chaque b_i décrive une courbe de Bézier non rationnelle de degré n :

$$b_i = b_i(v) = \sum_{j=0}^n b_{i,j} \cdot B_j^n(v) \quad \text{avec} \quad v \in [0,1]$$

nous pouvons maintenant combiner les deux équations et obtenir le point $S(u, v)$ sur la surface $b^{m,n}$.

$$S(u, v) = b^{m,n}(u, v) = \sum_{i=0}^m \sum_{j=0}^n b_{i,j} \cdot B_i^m(u) \cdot B_j^n(v) \quad (8)$$

l'équation (8) est appelé surface-produit-tensoriel, elle est également une surface de Bézier non rationnelle définie par (5).

de cette définition découle les propriétés des surfaces de Bézier non rationnelles :

- Invariance affine : l'algorithme de De Casteljau direct consiste en interpolation bilinéaire répétées, et éventuellement par la suite linéaire répétées. toutes ces opérations sont affinement invariantes, donc la composition l'est aussi. nous pouvons également dire que pour (8) soit une combinaison barycentrique (et par conséquent affinement invariante), nous devons avoir :

$$\sum_{j=0}^n \sum_{i=0}^m B_i^m(u) \cdot B_j^n(v) \equiv 1. \quad (9)$$

Ceci est aisément vérifié algébriquement.

- Propriété d'enveloppe convexe : pour $0 \leq u, v \leq 1$ les termes $B_i^m(u) \cdot B_j^n(v)$ sont non négatifs. Alors, en prenant en compte (8), (9) est une combinaison convexe.
- Courbes frontières : les courbes frontières $S(0, v), S(1, v), S(u, 0), S(u, 1)$ sont des courbes de Bézier non rationnelle délimitant la surface, leurs polygones de contrôle sont données par les polygones frontières du réseau de contrôle.

1.2.5. Surfaces de Bézier rationnelle

Une surface de Bézier rationnelle de degré (m, n) est la projection d'une surface non rationnelle de degré (m, n) définie dans l'espace des coordonnées homogènes (4D) dans l'espace (3D), après la projection dans l'espace (3D) nous obtenons :

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{i,j} \cdot R_{i,j}^{m,n}(u, v) \quad \text{avec} \quad 0 \leq u, v \leq 1 \quad (10)$$

$$\text{avec } R_{i,j}^{m,n}(u,v) = \frac{B_i^m(u) \cdot B_j^n(v) \cdot w_{ij}}{\sum_{r=0}^m \sum_{s=0}^n B_r^m(u) \cdot B_s^n(v) \cdot w_{rs}}$$

Les $R_{i,j}^{m,n}(u,v)$ sont les fonctions de base rationnelle bi-variable.

Les poids $w_{ij} \geq 0$ pour toutes les valeurs de i, j .

1.3. Courbes et surfaces de B-splines

L'idée fondamentale de la modélisation avec le modèle B-spline, est de corriger les défauts majeurs des courbes à pôles simples (tendance à l'oscillation des courbes à grand nombre de pôles, impossibilité de décrire un cercle), tout en conservant les avantages.

1.3.1. fonction de base B-spline

On adopte ici la définition récursive connue sous le nom de l'algorithme de Cox-De Boor :

Soit le vecteur nœuds $T = \{t_0, \dots, t_i, t_{i+1}, \dots, t_m\}$ donné par une séquence croissante de nombres réels appelées nœuds. La $i^{\text{ème}}$ fonction B-spline d'ordre k notée $N_i^k(t)$ est définie par [Colm'99][De Boor '20]:

Posons :

$$N_i^1(t) = \begin{cases} 1 & \text{si } t_i \leq t < t_{i+1}, \quad i = 0, \dots, m-1 \\ 0 & \text{sinon} \end{cases}$$

$$N_i^k(t) = \omega_{i,k} \cdot N_i^{k-1}(t) + (1 - \omega_{i+1,k}) \cdot N_{i+1}^{k-1}(t) \quad \text{avec } 1 < k \leq m \quad \text{et} \quad i = 0, \dots, m-k \quad (11)$$

$$\text{avec } \omega_{i,k} = \begin{cases} \frac{t - t_i}{t_{i+k-1} - t_i} & \text{si } t_i < t_{i+k-1}, \quad i = 0, \dots, m-k \\ 0 & \text{sinon} \end{cases}$$

Pour un vecteur nœuds T donné est, et $k \leq m$ fixe, nous avons $m - k + 1$ fonctions B-splines de degré $(k - 1)$ d'ordre k définie dans l'intervalle $[t_{k-1}, t_{m-k+1}]$.

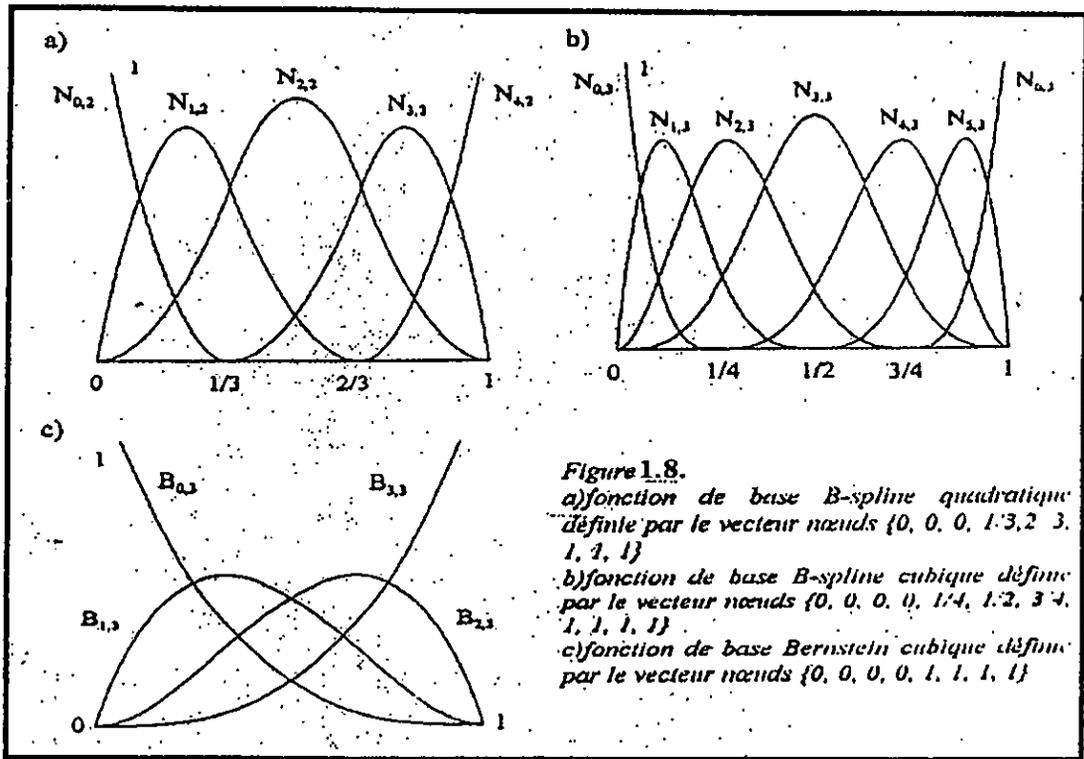
Pour les formes non uniformes et non-périodiques, le vecteur nœuds prends la forme

$$T = \left\{ \underbrace{\alpha, \alpha, \alpha, \alpha, \dots, \alpha}_{k+1}, t_{k-1}, \dots, t_{m-k+1}, \underbrace{\beta, \beta, \beta, \beta, \dots, \beta}_{k+1} \right\}, \text{ en pratique, et dans toutes les applications } \alpha = 0, \beta = 1.$$

Les fonctions $N_i^k(t)$ ont les propriétés importantes suivantes : [DeBoor'20]

- Non négativité : $N_i^k(t) > 0, \quad t_i < t < t_{i+k}$
- Partition unité : $\sum_{i=0}^n N_i^k(t) = 1 \quad \forall t \in [t_0, t_m]$, n est le nombre total des fonctions B-spline d'ordre k pour un vecteur nœuds donné (n dépend de k et m , $n = m - k + 1$).
- Support local : $N_i^k(t) = 0$ si t est à l'extérieur de l'intervalle $[t_{k-1}, t_{m-k+1}]$

- **Dérivabilité** : Toutes les dérivées des fonctions B-spline existent (par leur forme polynomiale). Pour un nœuds de multiplicité r elles sont $(k - r - 1)$ continûment dérivable.
- **Extremum** : Excepté la cas $k=1$, les fonctions B-spline ont un seul maximum.



Le choix du vecteur nœuds T influe directement sur la forme de la fonction définie par l'équation (11). On considère plusieurs types de vecteur nœuds. On suppose que k est fixé à l'avance.

Alors $T = \{t_0, \dots, t_m\}$ vecteur nœuds non périodique si le premier et le dernier nœuds ont une multiplicité $k+1$, c'est à dire : $t_0 = t_1 = \dots = t_k$ et $t_{m-k} = t_{m-k+1} = \dots = t_m$

Pour $t_0 = 0, t_m = 1$, les vecteurs nœuds non périodique non uniforme ont la forme suivante :

$$T = \left\{ \underbrace{0, 0, \dots, 0}_{k+1}, t_{k+1}, \dots, t_{m-k+1}, \underbrace{1, 1, \dots, 1}_{k+1} \right\}$$

- Si $\exists d > 0$, telque $t_i = t_{i+1} = d$ pour $k \leq i \leq m - k + 1$ alors T est un vecteur nœuds uniforme, sinon le vecteur est non uniforme.

L'utilisation des vecteurs nœuds non uniforme répond à un contrôle, et modélisation meilleure des formes que les vecteurs uniforme.

- Si $T = \{0, 0, \dots, 0, 1, 1, \dots, 1\}$ alors les fonctions de B-spline deviennent des fonctions de Base de Bernstein : $N_i^k(t) = B_i^k(t)$ voir figure(1.8.c)

1.3.2. Courbes B-spline non rationnelles

Soit donné le vecteur nœuds $T = \{t_0, \dots, t_m\}$, correspondant à $m - k + 1$ B-splines $N_i^k(t)$ d'ordre k , pour une valeur de k fixe ($k \leq m$), nous avons également $m - k + 1$ point de contrôles P_0, \dots, P_{m-k} , définissons une courbe B-spline non rationnelle de degré $k-1$ définie dans l'intervalle $[t_{k-1}, t_{m-k+1}]$ par [Colm'99][Haden'95][Eck'95]:

$$S(t) = \sum_{i=0}^{m-k} P_i \cdot N_i^k(t) \quad t \in [0, 1] \quad (12)$$

P_i : points de contrôle de la courbe, où points de De Boor.

$N_i^k(t)$: sont les fonctions de base B-spline de degré k définie par l'algorithme récursif de De Boor donnée par l'équation (11).

Noté que Le degré de la courbe est contrôlé par le degré des fonctions B-spline.

Algorithme de De Boor non rationnelle : [Farin'92]

Soit $t \in [t_i, t_{i+1}] \subset [t_{k-1}, t_{L+k-1}]$. avec L est le nombre potentiel de segments de courbe .

Si tous les nœuds du domaine de définition $[t_{k-1}, t_{L+k-1}]$ sont simples, L caractérise le nombre d'intervalles du domaine de définition, pour chaque multiplicité de nœuds dans le domaine le nombre d'intervalles chute de un. la somme de toutes les multiplicités de nœuds du domaine est reliée à L par :

$$\sum_{i=k-1}^{L+k-1} r_i = L + 1 \quad (13)$$

où r_i la multiplicité du nœud du domaine t_i .

Définissons :

$$P_i^p(t) = \frac{t_{i+k-p} - t}{t_{i+k-p} - t_{i-1}} \cdot P_{i-1}^{p-1}(t) + \frac{t - t_{i-1}}{t_{i+k-p} - t_{i-1}} \cdot P_i^{p-1}(t) \quad , \text{avec} \begin{cases} p = 1, \dots, k - r \\ i = I - k + p + 1, \dots, I + 1 \end{cases}$$

Et $P_i^0(t) = P_i$.

Alors ;

$S(t) = P_{I+1}^{k-r}(t)$ est la valeur de la courbe B-spline pour la valeur de paramètre t .

Dans notre description de l'algorithme de De Boor [Farin'92], nous n'avons pas renuméroté la suite de nœuds et les pôles à chaque niveau, puisque nous nous sommes intéressés que par le résultat final $S(t) = P_{I+1}^{k-r}(t)$. Bien sur, à chaque niveau p nous générons un nouveau polygone de contrôle qui décrit la même B-spline que le précédent polygone de contrôle. En particulier, pour $p=1$, nous obtenons l'algorithme d'insertion de nœuds de Boehm.

La figure (1.9) montre un exemple. Nous pouvons également voir cette exemple comme un cas d'insertion multiple de nœuds. dans ce contexte, nous avons construit plusieurs polygones qui décrivent la même B-spline :

$p=1$: les ordonnées de De Boor P_0, P_1, P_2, P_3, P_4 correspondant à la séquence de nœuds $t_0, t_1, t_2, t_3, t_4, t_5, t_6$.

$p=2$: les ordonnées de De Boor P_0, P_1, P_2, P_3, P_4 correspondant à la séquence de nœuds $t_0, t_1, t_2, t_3, t_4, t_5, t_6$.

$p=3$: les ordonnées de De Boor P_0, P_1, P_2, P_3, P_4 correspondant à la séquence de nœuds $t_0, t_1, t_2, t_3, t_4, t_5, t_6$.

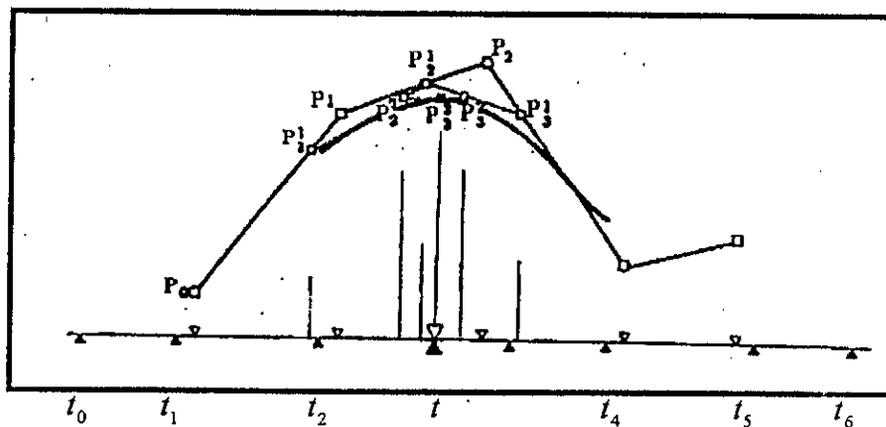


Figure 1.9. Algorithme de De Boor : exemple avec $n=3, L=2$

Cas spécial : $0 = t_0 = t_1 = \dots = t_{k-1} < t_k = t_{k+1} = \dots = t_{2k-1} = 1$

Ici à la fois t_0, t_n ont une multiplicité k , pour $t \in [0,1]$, L'algorithme de De Boor pose $I = k - 1$ et

$$P_i^p(t) = \frac{t_{i+k-p} - t}{t_{i+k-p} - t_{i-1}} \cdot P_{i-1}^{p-1}(t) + \frac{t - t_{i-1}}{t_{i+k-p} - t_{i-1}} \cdot P_i^{p-1}(t)$$

puisque $k - 1 \geq i - k \geq 0$, nous avons $t_{i+k-p} = 1, t_{i-1} = 0$ pour tous les i, k ;

Donc

$$P_i^p(t) = (1-t) \cdot P_{i-1}^{p-1} + t \cdot P_i^{p-1}, p = 1, \dots, k.$$

C'est justement l'algorithme De Casteljau, par conséquent nous pouvons conclure plusieurs points important de ce cas spécial.

Donc, si deux nœuds adjacents d'une suite de nœuds quelconque ont tous deux une multiplicité k , la courbe B-spline correspondantes est courbe de Bézier entre ces deux nœuds. Le polygone de contrôle B-spline est le polygone de Bézier.

Pour une courbe B-spline sur un vecteur nœuds arbitraire, nous pouvons toujours insérer les nœuds en question dans la suite de nœuds jusqu'à ce que chaque nœud soit de multiplicité k .

Le polygone correspondant à cette suite de nœuds est le polygone de Bézier par morceaux de la courbe, nous avons donc montré que les courbes B-splines sont polynomiales par morceaux sur $[t_{k-1}, t_{L+k-1}]$.

les propriétés géométriques des courbes B-spline définie par l'équation(12) sont [De Boor'20][Farin'93][GJ'98]:

- Une courbe B-spline est C^{k-r} continûment différentiable ,aux nœuds de multiplicité r , en particulier la courbe est $k-1$ fois continûment différentiable si tous les nœuds sont simples($r=1$)
- Précision Linéaire : si $l(t)$ est la ligne droite de la forme $l = a.t + b$, et si nous lisons les valeurs des abscisses de Greville ,la B-spline résultante reproduit la ligne droite :

$$\sum_i l(\xi_k) \cdot N_i^k(t) = l(t)$$

les abscisses de Greville sont données par : $\xi_k = \frac{t_{i+1} + \dots + t_{i+k-1}}{k-1}$, pour tout i .

- Propriété de l'enveloppe convexe : chaque point de la courbe est dans l'enveloppe convexe définie seulement par les $k+1$ pôles les plus proches, c'est à dire si $t \in [t_i, t_{i+1}]$ et $k-1 \leq i < m-k+1$ alors $P(t)$ est dans l'enveloppe convexe des pôles P_{i-k+1}, P_i .
- Propriété de diminution de variation : la courbe n'est pas intersectée par une droite en plus de points que le polygone de contrôle .ce résultat à une démonstration très simple, présenté par [Lane&Reisenfeld'83] :nous pouvons insérer chaque nœud jusqu'à avoir une multiplicité complète. Ceci est un processus à diminution de variation de variation, puisque c'est une interpolation linéaire par morceaux. Une fois que tous les nœuds sont de multiplicité complète ,le polygone de contrôle B-spline est le polygone de Bézier par morceaux , pour lequel nous avons montré la propriété de diminution de variation section (Annexe A).
- Comportement local : si on déplace un pôle ,la courbe sera localement déformée, et affecte la courbe dans $k+1$ intervalles .
- Phénomène commutateur : En se déplacent le long de la courbe de $t=0$, à $t=1$ les fonctions B-spline agissent comme un commutateur , d'un intervalle à l'autre . en passant par un nœud une fonction « s'éteint » une autre « s'allume».
- A l'intérieur d'un intervalle non nul $[t_i, t_{i+1}]$ toutes les dérivées existent , dans un nœud de multiplicité r elle sont $k-r$ fois continûment dérivable :

$$\frac{d}{dt} S(t) = k \cdot \sum_i \frac{P_i - P_{i-1}}{t_{i+k-1} - t_{i-1}} \cdot N_i^{k-1}(t)$$

- Le polygone formé par les pôles P_i représente une approximation linéaire par morceaux de la courbe .
- Invariance affine :une transformation affine est appliquée à la courbe si elle est appliquée aux pôles .
- La multiplicité $(k+1)$ des nœuds extrême engendre ces conditions aux extrémités de la courbe :
 $S(0) = P_0, S(1) = P_n, S'(0) = (P_1 - P_0) / t_{k+1}, S'(1) = k \cdot (P_n - P_{n-1}) / (1 - t_{m-k+1})$.
- Type de Bézier : une courbe de B-spline sans nœuds internes est une courbe de Bézier.

Exemple : si $U = \{0,0,0,0,1,1,1,1\}$ Vecteur nœuds non uniformes non périodiques et $k = 4$,nous obtenons une courbe de Bézier cubique (de degré3).

1.3.3. Courbes de B-spline rationnelles (NURBS)

les B-splines rationnelles non uniforme [NURBS- NonUniform Rational B-splines] font partie de IGES [Initial Graphics Exchange Specification] de puis 1983, ils sont incorporées à un grand nombre de modéleur géométriques et dans certain cas dans les micros processeur des stations graphiques évoluées.

Une courbe B-spline rationnelle 3D est la projection à travers l'origine d'une courbe B-spline non rationnelle 4D dans l'hyperplan $w=1$, elle est donc donnée par: [Farin'92] [GJ'98]

Pour $n+1$ points de contrôle, la B-spline rationnelle non uniforme de vecteur nœuds non uniforme $T = \{t_0, \dots, t_m\}$ est définie,

$$S(t) = \sum_{i=0}^n P_i^h \cdot N_i^k(t)$$

où P_i^h point de contrôle en coordonnées homogènes (4D)

$N_i^k(t)$ fonctions de base B-spline non rationnelle définie par l'algorithme récursive de Cox-DeBoor (11).

la projection en 3D en divisant par la coordonnée homogène,

$$S(t) = \frac{\sum_{i=0}^n w_i \cdot P_i \cdot N_i^k(t)}{\sum_{i=0}^n w_i \cdot N_i^k(t)} \quad (14)$$

où P_i point de contrôle en 3D.

nous pouvons écrire $S(t)$ sous la forme :

$$S(t) = \sum_{i=0}^n P_i \cdot R_i^k(t) \quad (15)$$

$$R_i^k(t) = \frac{w_i \cdot N_i^k(t)}{\sum_{i=0}^n w_i \cdot N_i^k(t)} \text{ avec } w_i > 0$$

où $R_i^k(t)$ fonctions de base des courbes B-splines rationnelle.

avec n le nombre de pôles données par $n = m - k$

les propriétés analytiques des fonctions $R_i^k(t)$ sont :

- Non négativité : $R_i^k(t) \geq 0, \forall t, i, k$
- Partition unité : $\sum_{i=0}^n R_i^k(t) = 1, \forall t \in [t_0, t_m]$
- Support local : $R_i^k(t) = 0$ si t est à l'extérieur de l'intervalle $[t_{k-1}, t_{m-k+1}]$.
- Dérivabilité : Toutes les dérivées des fonctions $R_i^k(t)$ existent pour un nœuds de multiplicité r elle sont $(k-r-1)$ continûment dérivable.

- Les fonctions $R_i^k(t)$ sont une généralisation des fonctions non rationnelle B-spline ,et des fonctions rationnelles et non rationnelle de Bernstein :si tous les $w_i = 1$,et

$$T = \left\{ \underbrace{0, \dots, 0}_{k+1}, \underbrace{1, \dots, 1}_{k+1} \right\} \text{ alors } R_i^k(t) = B_i^k(t).$$

Un schéma peut servir à illustrer les relations d'inclusion entre ces divers modèles :

$F = \{\text{Fractions rationnelles}\}$	$BSR = \{\text{B-splines rationnelles}\}$	$BR = \{\text{Bézier rationnelles}\}$	$BSR = \{\text{B-splines rationnelles}\}$
$P \subset F$ ↓	$BSP \subset BSR$ ↓	$BP \subset BR$ ↓	$BR \subset BSR$ ↓
$P = \{\text{Polynômes}\}$	$BSP = \{\text{B-splines Polynomiales}\}$	$BP = \{\text{Bézier polynomiales}\}$	$BR = \{\text{Modèle Bézier rationnelles}\}$

Les preuves de ces propriétés découlent de la définition des fonctions $R_i^k(t)$ et des propriétés analogues des fonctions $N_i^k(t)$.

Une courbe B-spline rationnelle est donnée par sa suite de nœuds T , sont polygone de contrôle 3D, et sa séquence de poids $w_i > 0$, les points de contrôles P_i sont les projections des points de contrôles 4D, $P_i^w = [w_i \cdot P_i \quad w_i]^T$.

Pour évaluer une B-spline rationnelle en une valeur de paramètre t , nous pouvons appliquer l'algorithme de De Boor à la fois au numérateur, et au dénominateur de (14), puis finalement effectuer la division. Ceci correspond à l'évaluation de la courbe non rationnelle 4D de points de contrôles $[w_i \cdot P_i \quad w_i]^T$ et à la projection du résultat dans E^3 , exactement comme le cas des courbes de Bézier. Ceci peut amener à des instabilités, et nous donnons donc une version rationnelle de l'algorithme de De Boor plus stable, mais demandant également plus de temps de calculs [Farin'93] :

Algorithme de De Boor, rationnelle :

Soit $t \in [t_j, t_{j+1}] \subset [t_{k-1}, t_{L+k-1}]$

L est donnée par l'équation (13)

Définissons

$$P_i^p(t) = [(1 - \alpha_i^p) \cdot w_{i-1}^{p-1} \cdot P_{i-1}^{p-1}(t) + \alpha_i^p \cdot w_i^{p-1} \cdot P_i^{p-1}(t)] / w_i^p$$

pour $p = 1, \dots, k-r$, et $i = I-n+k+1, \dots, I+1$, où

$$\begin{cases} \alpha_i^p = \frac{t - t_{i-1}}{t_{i+k-p} - t_{i-1}} \\ w_i^p(t) = (1 - \alpha_i^p) \cdot w_{i-1}^{p-1} + \alpha_i^p \cdot w_i^{p-1} \end{cases}$$

Et posons $P_i^0 = P_i$ et $w_i^0 = w_i$

Alors $S(t) = P_{j+1}^{k-r}(t)$ est le point sur la courbe B-spline pour la valeur de paramètre t . Ici r désigne la multiplicité de nœud, dans le cas où il faisait partie des nœuds. Sinon, posons $r=0$.

L'insertion de nœuds est, comme dans le cas non rationnelle, effectuée en n'exécutant qu'une étape de l'algorithme de De Boor, c'est à dire en fixant $p=1$ dans l'algorithme ci-dessus. Les sommets du polygone original $P_{l-k+2}, \dots, P_{l+1}$ sont remplacés par les

$P_{l+k-2}^{(1)}, \dots, P_{l+1}^{(1)}$; les poids respectifs sont les nombres $w_{l+k-2}^{(1)}, \dots, w_{l+1}^{(1)}$.

La courbe B-spline rationnelle, étant polynomiale rationnelle par morceaux, à une représentation de Bézier rationnelle par morceaux. Nous pouvons trouver le polygone de Bézier et les poids associés pour chaque segment en insérant chaque nœud jusqu'à ce qu'il soit de multiplicité k , c'est à dire en appliquant l'algorithme de De Boor à chaque nœud.

Les propriétés géométriques des courbes B-splines rationnelles définie par l'équation (14) sont [Asma'99]:

- Les conditions aux extrémités de la courbe sont :

$$S(0) = P_0 = P_0^w / w_0, S(1) = P_n = P_n^w / w_n$$

$$S'(0) = (P_1^w \cdot (P_1 - P_0)) / (w_0 \cdot t_{k+1})$$

$$S'(1) = (P_{n-1}^w \cdot (P_n - P_{n-1})) / (w_n \cdot (1 - t_{m-k+1}))$$

- Invariance affine : une transformation affine est appliqué à la courbe si elle est appliqué aux points de contrôles.
- Enveloppe convexe locale : la courbe est contenue dans l'enveloppe convexe des points de contrôles, c'est à dire si $t \in [t_i, t_{i+1}]$ et $k-1 \leq i < m-k+1$ alors $P(t)$ est dans l'enveloppe convexe des pôles P_{i-k+1}, P_i .
- le polygone formé par les points de contrôles P_i représente une approximation linéaire par morceaux de la courbe.
- Comportement local : si on déplace un des pôles, la courbe sera localement déformée et affecte la courbe dans les $k+1$ intervalles.
- Phénomène commutateur : En se déplaçant le long d'une courbe de $t=0$, à $t=1$ les fonctions B-spline agissent comme un commutateur, d'un intervalles à un autre, en passant par un nœud une fonction « s'éteint » une autre « s'allume ».
- A l'intérieur de l'intervalle non nul $[t_{k-1}, t_{m-k+1}]$ toutes les dérivées existent, dans un nœud de multiplicité r elle sont $k-r-1$ fois dérivable.
- Propriété de diminution de variation : Aucun plan n'a plus d'intersections avec la courbe qu'avec le polygone de contrôle.
- Type de Bézier : une courbe B-spline rationnelle sans nœuds internes est une courbe de Bézier rationnelle. les courbes B-spline rationnelle sont une généralisation des courbes de Bézier rationnelles et non rationnelles.
- Les poids w_i affectent la courbe localement.
- Pour une valeur de t fixée, $t \in [0,1]$, si le poids w_i croit, la courbe tend vers le point de contrôle P_i , Si w_i décroît la courbe s'éloigne du pôle P_i . (figure 1.10.a,b)
- Précision Linéaire : L'effet de rapprochement -éloignement sur la courbe peut être quantifié avec précision. Pour $k-1 < i < n$, on note par $S[w_i](t)$ une courbe parmi une famille paramétrique des courbes B-spline générées par les w_i variant entre $0 < w_i < \infty$, soit $t \in [t_i, t_{i+1}]$. Alors pour toutes les valeurs des w_i , le point $S[w_i](t)$ et lié par un segment passant par P_i . $S[w_i](t)$ quand $w_i \rightarrow \infty$. alors quand les w_i variant, le point $S[w_i](t)$ balaye le segment passant par P_i .

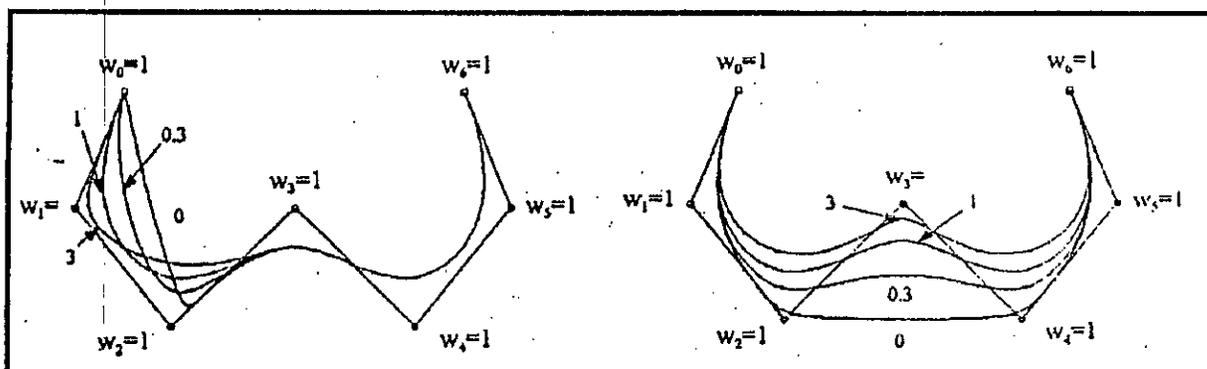


Figure 1.10. Courbes B-spline rationnelle cubiques

1.3.4. surfaces B-spline non rationnelles

Une surface B-spline de degré $(k_u - 1, k_v - 1)$, d'ordre (k_u, k_v) , donné par deux vecteurs nœuds $U = \{u_0, \dots, u_{m_u}\}$, $V = \{v_0, \dots, v_{m_v}\}$ de différentes longueurs possible, et de $(m_u - k_u + 1) \times (m_v - k_v + 1)$ pôles est définie par :

$$S(u, v) = \sum_{i=0}^{m_u - k_u} \sum_{j=0}^{m_v - k_v} N_i^{k_u}(u) \cdot N_j^{k_v}(v) \cdot P_{ij} \quad (16)$$

pour $(u, v) \in [u_{k_u - 1}, u_{m_u - k_u + 1}] \times [v_{k_v - 1}, v_{m_v - k_v + 1}]$

$N_i^{k_u}(u)$, $N_j^{k_v}(v)$ sont les fonctions de Base B-spline données par l'algorithme récursif (11)

Approche produit tensoriel [Hoschek'93]:

Comme le cas des surfaces de Bézier produit-tensoriel, une surface B-spline produit-tensoriel, peut être construite comme suit :

Soit la courbe initiale une courbe de B-spline non rationnelle de degré $(k_u - 1)$:

$$P_i(u) = \sum_{i=0}^{m_u - k_u} P_i \cdot N_i^{k_u}(u) \quad \text{avec} \quad u \in [u_{k_u - 1}, u_{m_u - k_u + 1}]$$

ou chaque P_i décrit une courbe B-spline non rationnelle de degré $(k_v - 1)$:

$$P_i = P_i(v) = \sum_{j=0}^{m_v - k_v} P_{i,j} \cdot N_j^{k_v}(v) \quad \text{avec} \quad v \in [v_{k_v - 1}, v_{m_v - k_v + 1}]$$

nous pouvons maintenant combiner les deux équations et obtenir le point $S(u, v)$ sur la surface $P^{m_u - k_u, m_v - k_v}$.

$$S(u, v) = P^{m_u - k_u, m_v - k_v}(u, v) = \sum_{i=0}^{m_u - k_u} \sum_{j=0}^{m_v - k_v} P_{ij} \cdot N_i^{k_u}(u) \cdot N_j^{k_v}(v) \quad (17)$$

l'équation (17) est appelée surface B-spline produit-tensoriel, elle est également une surface de B-spline non rationnelle définie par (16).

En particulier pour $(u, v) \in [0,1] \times [0,1]$ alors $S(0, v), S(1, v), S(u, 0), S(u, 1)$ sont les quatre courbes délimitant la surfaces $S(u, v)$.

les $(m_u - k_u + 1) \times (m_v - k_v + 1)$ produits de fonctions $N_i^{k_u}(u) \cdot N_j^{k_v}(v)$, de l'équation (16) sont appelés fonctions de Base B-spline bi-variable de degré $(k_u - 1, k_v - 1)$.

Ces fonctions ont les propriétés analytiques suivantes : [Colm'99][Asma'99] :

- Non négativité : $N_i^{k_u}(u) \cdot N_j^{k_v}(v) \geq 0$, pour toutes les valeurs de i, j, k_u, k_v, u, v .
- Partition unité : $\sum_{i=0}^{m_u - k_u} \sum_{j=0}^{m_v - k_v} N_i^{k_u}(u) \cdot N_j^{k_v}(v) = 1$ tout couple pour $(u, v) \in [0,1] \times [0,1]$.
- Support local : $N_i^{k_u}(u) \cdot N_j^{k_v}(v) = 0$ si $(u, v) \notin [u_{k_u-1}, u_{m_u-k_u+1}] \times [v_{k_v-1}, v_{m_v-k_v+1}]$.
- Dérivabilité : à l'intérieur des rectangles formés par U, V lignes nodales, toutes les dérivées partielles de $N_i^{k_u}(u) \cdot N_j^{k_v}(v)$ existent. au nœud u (nœud v) elle est $k_u - r - 1$ ($k_v - r - 1$) fois continûment dérivable suivant la direction u (direction v), où r est la multiplicité du nœud.

Les propriétés des surfaces B-splines non rationnelles sont :

- Invariance affine.
- Enveloppe convexe : si $(u, v) \in [u_{k_u-1}, u_{m_u-k_u+1}] \times [v_{k_v-1}, v_{m_v-k_v+1}]$, alors $S(u, v)$ est dans l'enveloppe convexe formée par le réseau des pôles $(m_u - k_u + 1) \times (m_v - k_v + 1)$.
- Approximation local : le déplacement d'un pôle P_{ij} n'affecte la surface que dans $(u, v) \in [u_{k_u-1}, u_{m_u-k_u+1}] \times [v_{k_v-1}, v_{m_v-k_v+1}]$ avec $i \in \{0, \dots, m_u - k_u + 1\}, j \in \{0, \dots, m_v - k_v + 1\}$
- $S(u, v)$ est $k_u - r - 1$ ($k_v - r - 1$) fois continûment dérivable suivant la direction u (direction v) pour nœud suivant u (suivant v) de multiplicité r .
- Type de Bézier : une surface B-spline non rationnelle sans nœuds internes est une surface de Bézier non rationnelles.

Exemple : pour deux vecteurs nœuds non uniformes non périodiques $U = \{0, 0, 0, 0, 1, 1, 1, 1\}$ et $V = \{0, 0, 0, 0, 1, 1, 1, 1\}$, et les deux d'ordre 4, nous obtenons une surfaces de Bézier (bi-cubique) non rationnelle de degré (3,3).

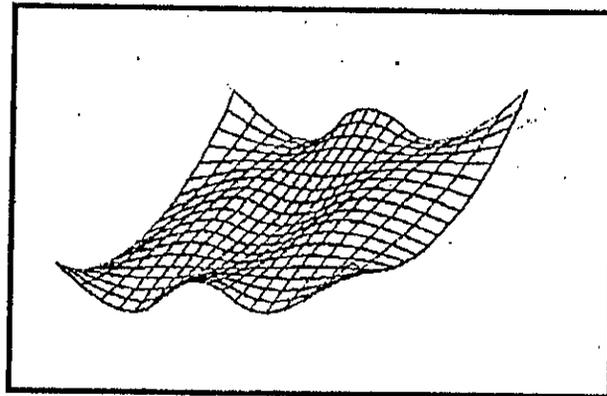


Figure 1.11. surface B-spline non rationnelle de degré (2×3) , les vecteurs nœuds sont : $U = \{0,0,0,1/3,2/3,1,1,1\}$, $V = \{0,0,0,0,1,1,1\}$

1.3.5. surfaces B-spline rationnelles

Une surface B-spline rationnelle de degré $(k_u - 1, k_v - 1)$, d'ordre (k_u, k_v) , est définie dans l'espace (3D) par :

$$S(u, v) = \sum_{i=0}^{m_u-k_u} \sum_{j=0}^{m_v-k_v} R_{i,j}^{k_u, k_v}(u, v) \cdot P_{ij} \text{ avec } (u, v) \in [u_{k_u-1}, u_{m_u-k_u+1}] \times [v_{k_v-1}, v_{m_v-k_v+1}] \quad (18)$$

où P_{ij} sont les points de contrôles de la surfaces .

et $R_{i,j}^{k_u, k_v}(u, v)$ sont les fonctions de Base rationnelles B-spline bi-variables données par :

$$R_{i,j}^{k_u, k_v}(u, v) = \frac{N_i^{k_u}(u) \cdot N_j^{k_v}(v) \cdot w_{ij}}{\sum_{r=0}^{m_u-k_u} \sum_{s=0}^{m_v-k_v} N_r^{k_u}(u) \cdot N_s^{k_v}(v) \cdot w_{rs}}$$

les fonctions $R_{i,j}^{k_u, k_v}(u, v)$ ont les propriétés analytiques suivantes :

- Non négativité : $R_{i,j}^{k_u, k_v}(u, v) \geq 0$ pour toutes les valeurs de i, j ,
 $(u, v) \in [u_{k_u-1}, u_{m_u-k_u+1}] \times [v_{k_v-1}, v_{m_v-k_v+1}]$;
- Partition unité : $\sum_{i=0}^{m_u-k_u} \sum_{j=0}^{m_v-k_v} R_{i,j}^{k_u, k_v}(u, v) = 1$ pour tout couple $(u, v) \in [0,1] \times [0,1]$
- Support local : $R_{i,j}^{k_u, k_v}(u, v) = 0$ si $(u, v) \notin [u_{k_u-1}, u_{m_u-k_u+1}] \times [v_{k_v-1}, v_{m_v-k_v+1}]$.
- Dérivabilité : à l'intérieur des rectangles formés par U, V lignes nodales , toutes les dérivées partielles de $R_{i,j}^{k_u, k_v}(u, v)$ existent .au nœud u (nœud v) elle est

$k_u - r - 1$ ($k_v - r - 1$) fois continûment dérivable suivant la direction u (direction v), où r est la multiplicité du nœud.

- Les fonctions $R_{i,j}^{k_u,k_v}(u,v)$ sont une généralisation des fonctions de base B-spline non rationnelles, c'est à dire $R_{i,j}^{k_u,k_v}(u,v) = N_i^{k_u}(u) \cdot N_j^{k_v}(v)$ si tout les $w_{ij} = 1$.

Les propriétés géométriques des surfaces B-splines rationnelles sont à celles des surfaces B-spline non rationnelles :

- Les quatre pôles extrêmes du réseau de contrôles coïncident avec les quatre points extrêmes de la surfaces. de plus, les dérivées partielles par rapport à u, v coïncident avec les dérivées des points de la courbe.
- Invariance affine.
- Enveloppe convexe : identique au cas non rationnelle.
- Approximation local : identique au cas non rationnelle.
- $S(u,v)$ est $k_u - r - 1$ ($k_v - r - 1$) fois continûment dérivable suivant la direction u (direction v) pour nœud suivant u (suivant v) de multiplicité r .
- une surface de B-spline rationnelle sans nœuds interne est une surface de Bézier rationnelle. les surfaces de B-spline rationnelle sont une généralisation des surfaces B-spline non rationnelle et des surfaces de Bézier rationnelles et non rationnelles.
- les poids w_{ij} sont des paramètre de forme. il affecte la surface localement et ont un effet de rapprochement -éloignement qui peut être quantifié.

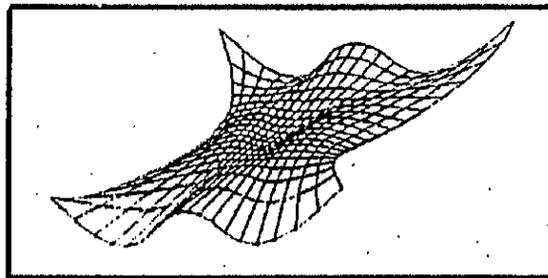


Figure 1.12. surfaces B-spline rationnelle

CHAPITRE 2
Algorithmique
des
transformations

2. Algorithmique des transformations

2.1. Transformations dans le modèle de Bézier :

Nous présentons dans cette partie les méthodes de transformation du polygone caractéristique des courbes de Bézier qui seront utilisées dans la conversion par approximation des courbes .

2.1.1. Subdivision

Cette procédure permet de partager une courbe de bézier de degré n en deux courbes de Bézier de même degré en un point défini t_0 .

2.1.1.1. Subdivision des courbes non rationnelles :

Une courbe de Bézier $C(t)$ de degré n définie par ses (n+1) pôles $[P_0, \dots, P_n]$. Peut être subdivisée à t_0 en deux courbes de bézier $R(t)$ et $Q(t)$ toutes les deux de degré n et ayant le point commun $R(t_0)=Q(t_0)=C(t_0)$. En utilisant le polygone $[P_0, \dots, P_n]$ et l'algorithme de récurrence : [Bensalah '90], [Farouki & Neef '90]

En posant $C_i^0(t) = P_0$ (19)
 $C_i^k(t) = (1-t).C_i^{k-1}(t) + t.C_{i+1}^{k-1}(t)$

Les deux polygones $[R_0, \dots, R_n]$ et $[Q_0, \dots, Q_n]$ seront :

$$R_i = C_i^1(t_0) \quad i=0, \dots, n$$

$$C_0^0 = R_0 \quad \rightarrow C_0^1 = R_1 \quad \rightarrow C_0^2 = R_2 \quad \rightarrow C_0^3 = R_3 \quad \rightarrow C_0^4 = R_4 = Q_0$$

$$C_1^0 \quad \rightarrow C_1^1 \quad \rightarrow C_1^2 \quad \rightarrow C_1^3 = Q_3$$

$$C_2^0 \quad \rightarrow C_2^1 \quad \rightarrow C_2^2 = Q_2$$

$$C_3^0 \quad \rightarrow C_3^1 = Q_3$$

$$C_4^0 = Q_4$$

$$Q_i = C_i^{n-i}(t_0) \quad i=0, \dots, n$$

Les $C_p^q(t_0)$ seront calculés en utilisant l'équation de récurrence (19)

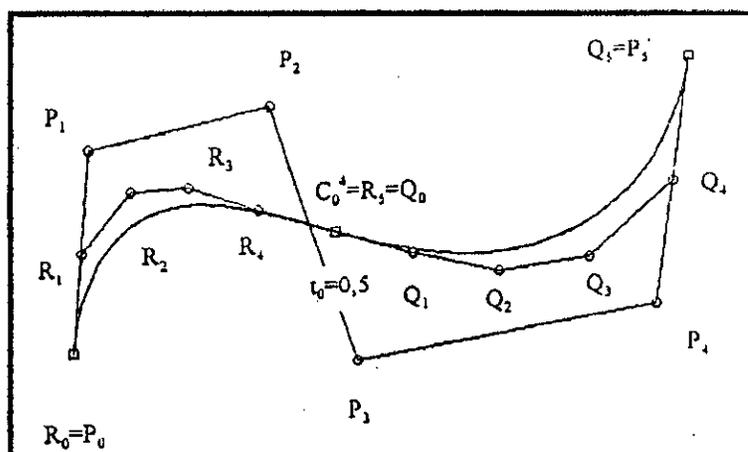


Figure 2.1. courbe de Bézier non rationnelle $P(t)$ de degré 5 subdivisé en deux courbes $R(t)$ et $Q(t)$ de même degré

2.1.1.2. Subdivision des courbes rationnelles

Une courbe de Bézier rationnelle $C(t)$ de degré n définie par ses $(n+1)$ pôles $[P_0, \dots, P_n]$ chargés respectivement par les poids (w_0, \dots, w_n) peut être subdivisée à t_0 en deux courbes de Bézier rationnelles $R(t)=Q(0)=C(t_0)$. En utilisant le polygone $[P_0, \dots, P_n]$ les charges (w_0, \dots, w_n) et l'algorithme de récurrence (21) : [Bensalah '90].

L'équation des charges s'écrit $w_i^r(t) = \sum_{j=0}^r w_{i+j} \cdot B_j^r(t)$,

$$\text{Avec } w_i^r(t) = (1-t) \cdot w_i^{r-1}(t) + t \cdot w_{i+1}^{r-1}(t) \quad (20)$$

L'algorithme de récurrence pour les courbes de Bézier rationnelles s'écrit alors

$$C_i^r(t) = (1-t) \cdot \frac{w_i^{r-1}(t)}{w_i^r(t)} \cdot C_i^{r-1}(t) + \frac{w_{i+1}^{r-1}(t)}{w_i^r(t)} \cdot C_{i+1}^{r-1}(t) \quad (21)$$

avec $C_i^0(t) = P_i$; $w_i^0(t) = w_i$

Le polygone $[R_0, \dots, R_n]$ et ses charges $(\alpha_0, \dots, \alpha_n)$ de la courbe $R(t)$, et le polygone $[Q_0, \dots, Q_n]$ et ses charges $(\beta_0, \dots, \beta_n)$ seront déterminés récursivement par

$$R_i = C_0^i(t_0), \text{ et } \alpha_i = w_0^i(t_0) \quad i=0, \dots, n$$

$$Q_i = C_i^{n-i}(t_0), \text{ et } \beta_i = w_i^{n-i}(t_0) \quad i=0, \dots, n$$

Les $C_p^q(t_0)$ seront calculés par l'équation (21) et les $w_p^q(t_0)$ par l'équation (20).

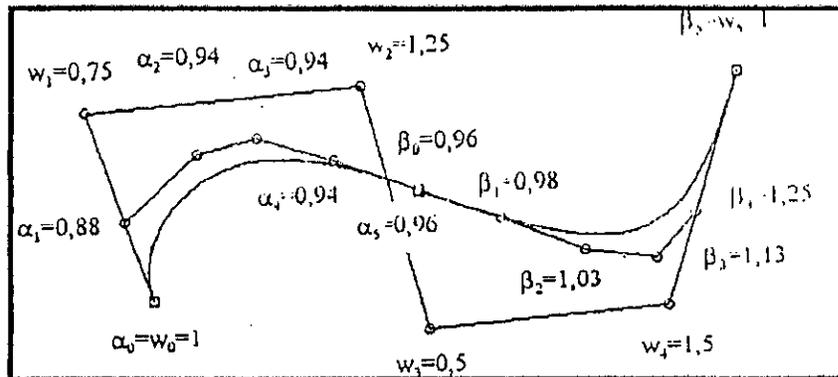


Figure 2.2. courbe de Bézier rationnelle de degré 5 (de poids w_i) subdivisée en deux courbes rationnelles de même degré (de poids α_j et β_j)

2.1.2. Elévation de degré

La procédure d'élévation de degré permet d'augmenter le nombre de pôles d'une courbe. Les courbes initiales et finale sont identiques mais de degrés différents.

2.1.2.1 Elévation de degré des courbes non rationnelles

Soit une courbe $C(t)$ de degré n définie par ses $(n+1)$ pôles P_i ($i=0, \dots, n$). La courbe $Q(t)$ de degré $(n+1)$ est définie par ses $(n+2)$ pôles Q_j ($j=0, \dots, n+1$) déterminés par la formule récurrente suivante : [Bensalah '90], [Piegl & Tiller '94][Asma'99].

$$Q_i = (1 - \alpha_i) \cdot P_i + \alpha_i \cdot P_{i-1} \quad (22)$$

avec $\alpha_i = i/(n+1)$ ($i=0, \dots, n+1$)

il s'en suit que $Q_0 = P_0$ et $Q_{n+1} = P_n$. En utilisant les nouveaux pôles Q_j , on peut élever encore le degré de $n+2$ en appliquant une nouvelle fois l'algorithme.

L'algorithme d'élévation de degré de $n+p$ s'écrit alors de la manière suivante

($i=0, \dots, n$)

$$Q_i = P_i$$

($j=1, \dots, p$)

$$Q_{n+j} = P_{n+j+1};$$

($k=n+j-1, \dots, 1$ pas -1)

$$Q_k = \alpha_{j,k} \cdot Q_{k-1} + (1 - \alpha_{j,k}) \cdot Q_k;$$

Fin

Une des qualités les plus importantes de cet algorithme est que les α_i dépendent seulement du degré. Ils peuvent par conséquent être calculés une seule fois et enregistrés dans le fichier.

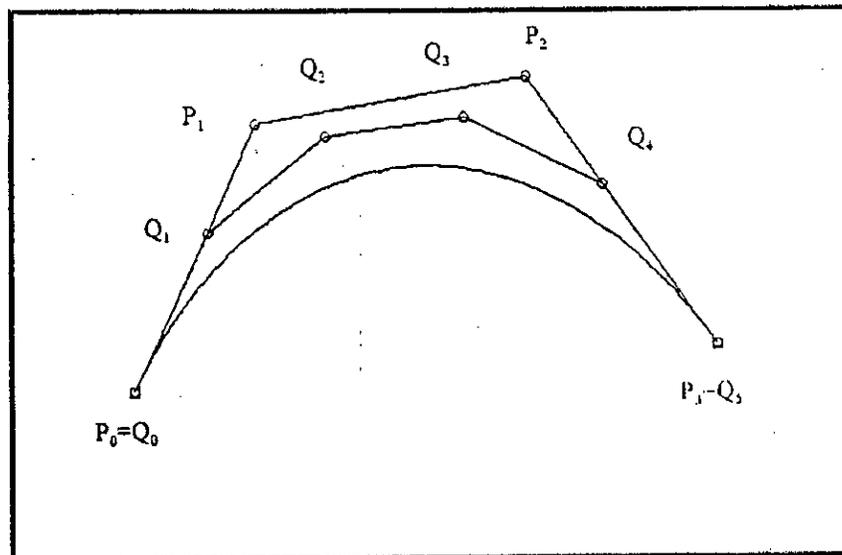


Figure 2.3. Courbe de Bézier non rationnelle de degré 3 élevée au degré 5

2.1.2.2 Elévation de degré des courbes rationnelles

Soit une courbe $C(t)$ de degré n définie par ses $(n+1)$ pôles P_i ($i=0, \dots, n$), chargée par les poids w_i ($i=0, \dots, n$). La courbe $Q(t)$ de degré $(n+1)$ est définie par ses $(n+2)$ pôles Q_j ($j=0, \dots, n+1$) chargés par les poids β_j ($j=0, \dots, n+1$) déterminés par : [Bensalah '90], [Piegl & Tiller '94].

$$Q_i = C_{i-1}^1 \left(\frac{n+1-i}{n+1} \right); i=0, \dots, n+1 \quad (23)$$

$$\beta_i = w_{i-1}^1 \left(\frac{n+1-i}{n+1} \right); i=0, \dots, n+1 \quad (24)$$

Les $C_p^q(t)$ seront calculés pour l'équation (21) et les $w_p^q(t_0)$ par l'équation (20).

L'élévation de degré à l'ordre supérieur se fait par application répétée de cet algorithme.

2.1.3 Réduction de degré

la réduction de degré d'une courbe de Bézier se fait en diminuant le nombre de pôle de la courbe initiale. La courbe obtenue est une approximation de la courbe traitée.

2.1.3.1 Réduction de degré des courbes non rationnelles

Si une courbe de Bézier $C(t)$ de degré $(n+1)$ définie par ses $(n+2)$ pôles P_i ($i=0, \dots, n$), est mathématiquement identique à une courbe $Q(t)$ de degré n définie par ses $(n+1)$ pôles Q_j

($j=0, \dots, n$) alors la propriété de l'hodographe est vérifiée. Les pôles Q_j peuvent être calculés par la résolution inverse de l'équation de l'élévation de degré (22). Cette résolution peut se faire de deux façons (25) et (26) selon que la résolution commence par Q_0 ou par Q_n : [Farin '83],[Forrest '72] :

$$R_i = (1 + \beta_i) \cdot P_i - \beta_i \cdot R_{i-1} \quad (25)$$

avec $\beta_i = i/(n+1-i)$ ($i=0, \dots, n$)

$$S_{i-1} = (1 + \gamma_i) \cdot P_i - \gamma_i \cdot S_i \quad (26)$$

avec $\gamma_i = 1/\beta_i = (n+1-i)/i$ ($i=n, \dots, 0$)

Si la propriété de l'hodographe n'est pas vérifiée alors (25) et (26) donnent deux polygones R_j et S_j différents.

Une première approximation à été proposé par Farin [Farin '83] et consiste à prendre pour la courbe réduite $Q(t)$ la moyenne pondérée des deux polygones R et S :

$$Q_i = \frac{n-i}{n} \cdot R_i + \frac{i}{n} \cdot S_i; i = 0, \dots, n \quad (27)$$

L'application successive de cette méthode produit une réduction du degré n au degré $m < n$, par une réduction d'un degré à chaque étape. Les erreurs d'approximation s'ajoutent à chaque étape.

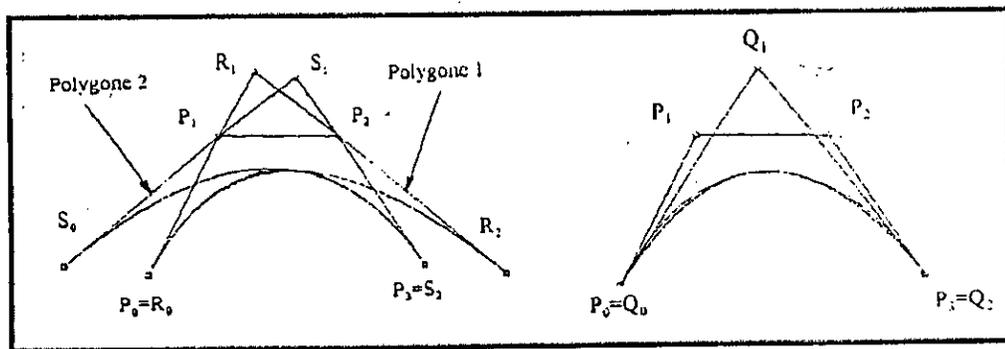


Figure 2.4.a. Courbe de Bézier non rationnelle de degré 3 réduite au degré 2 par les deux méthodes polygone 1 et 2.

Figure 2.4.b. Courbe précédente de degré 3 réduite par la méthode du polygone moyen.

2.1.3.2 Réduction de degré des courbes rationnelles

En reprenant le même raisonnement que pour les courbes non rationnelles, les équations (23) et (24) peuvent être résolues de deux manière : [Farin '92]

$$R_i = \frac{n+1}{n+1-i} \cdot \frac{w_i}{\alpha_i} \cdot P_i - \frac{\alpha_{i-1}}{\alpha_i} \cdot \frac{i}{n+1-i} \cdot R_{i-1} \quad (28.a)$$

$$\alpha_i = \frac{n+1}{n+1-i} \cdot w_i - \frac{i}{n+1-i} \cdot \alpha_{i-1} \quad (29.b)$$

$i=n, \dots, 0$

$$S_{i-1} = \frac{n+1}{i} \cdot \frac{w_i}{\beta_{i-1}} \cdot P_i - \frac{\beta_i}{\beta_{i-1}} \cdot \frac{n+1-i}{i} \cdot S_i \quad (28.b)$$

$$\beta_{i-1} = \frac{n+1}{i} \cdot w_i - \frac{n+1-i}{i} \cdot \beta_i \quad (29.a)$$

Les R_i et les S_i sont les pôles recherchés chargés respectivement par les poids α_i et β_i .
L'approximation de Farin pour le cas rationnel devient

$$Q_i = \frac{n-i}{n} \cdot \frac{\alpha_i}{\mu_i} \cdot R_i - \frac{\beta_i}{\mu_i} \cdot \frac{i}{n} \cdot S_i \quad (30.a)$$

$$\mu_i = \frac{n-i}{n} \cdot \alpha_i - \frac{i}{n} \cdot \beta_i \quad (30.b)$$

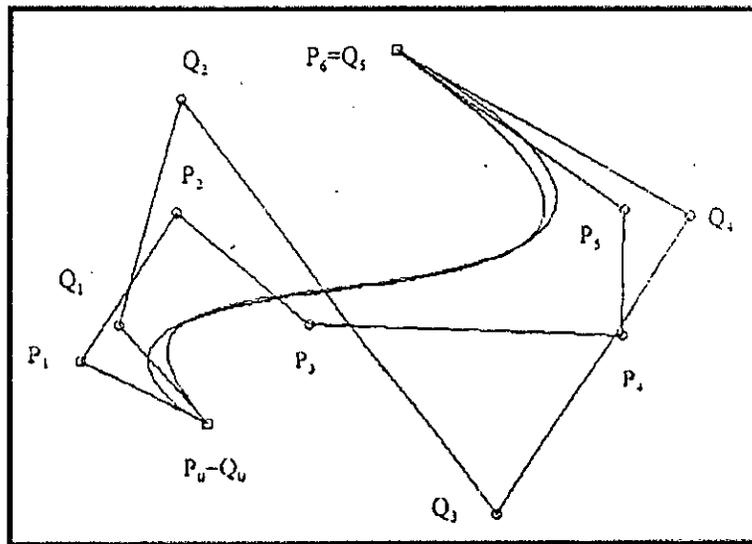


Figure 2.5. Courbe de Bézier rationnelle de degré 6 réduite au degré 5 par la méthode du polygone moyen $W = \{1,1,1,0,8,0,5,0,8,1,2,1\}$; $Wr = \{1,1,304,0,516,0,516,1,304,1\}$

2.1.4 Raccordement des courbes

Le raccordement de deux courbes de Bézier en un point donné consiste à rassembler les deux courbes en ce point en respectant une continuité géométrique désirée.

2.1.4.1 Raccordement des courbes non rationnelles

Le raccordement de deux courbes non rationnelles de Bézier $R(t)$ et $S(t)$ de même ordre, au point $R(1)=S(0)$, pose autant de contraintes que l'ordre de continuité géométrique C^k

(d'ordre k) entre deux courbes $R(t)$ définie par le polygone (R_0, \dots, R_n) et $S(t)$ définie par le polygone (S_0, \dots, S_n) , il suffit de satisfaire les $(k+1)$ conditions :

$$\left[\frac{d}{dt} \right]^i R(t) = \left[\frac{d}{dt} \right]^i S(t); i = 0, \dots, k \leq n \quad (31)$$

Sachant que les k premières dérivées de $R(t)$ au point $t=1$ dépendent de ses $(k+1)$ derniers pôles, et que les k premières dérivées de $S(t)$ dépendent de ses $(k+1)$ derniers pôles, il suffit seulement de calculer les $(k+1)$ premiers pôles de $S(t)$ en continuité des $(k+1)$ derniers pôles de $R(t)$, les autres pôles n'ont pas d'influence sur la continuité C^k désirée. Les pôles S_i seront alors calculés par

$$S_i = R_{n-i}^i(2); i=0, \dots, k. \quad (32)$$

Il s'agit d'une extrapolation de $R(t)$ pour le point $S(1)$ donc $t=2$.

Si les deux courbes $R(t)$ et $S(t)$ ne sont pas de même degré, par l'élévation du degré de la plus faible.

Si $k=n$ les deux courbes $R(t)$ et $S(t)$, sont la subdivision d'une autre courbe $C(t)$ d'après le principe même de la subdivision.

II.1.4.2 Raccordement des courbes rationnelles

Pour raccorder une courbe rationnelles de Bézier $R(t)$ définie par ses pôles R_i , et poids α_i à une courbe rationnelle de Bézier $S(t)$ de même ordre, définie par ses pôles S_i chargés par les poids β_i , au pont $R(1)=S(0)$, en respectant une continuité géométrique d'ordre k , il suffit de calculer les $(k+1)$ premiers pôles de $S(t)$ et leurs charges par

Pour $i=0, \dots, k$

$$S_i = R_{n-i}^i(2) \quad (33.a)$$

$$\beta_i = \alpha_{n-i}^i(2) \quad (33.b)$$

2.2 Transformations dans le modèle B-spline

Pour permettre le passage à la réduction de degré des courbes B-spline, nous avons besoin des transformations du polygone et du vecteur nodal. Nous présentons ici les méthodes indispensables à cette application.

2.2.1 Insertion de nœuds

Cette procédure permet d'ajouter un nouveau nœud dans le vecteur nodal d'une courbe B-spline, et donc d'élever le degré de la courbe.

2.2.1.1 Insertion de nœuds pour les courbes non rationnelles

Soit une courbe B-spline non rationnelle $C(t)$ de degré p définie par :

$$C(t) = \sum_{i=0}^n P_i \cdot N_i^p(t) \quad 0 \leq t \leq 1$$

où P_i sont des pôles de la courbe et $N_i^p(t)$ sont des fonctions de base de B-spline de degré p définies par l'équation (6) et le vecteur nœuds (non périodique) $\{t_0, \dots, t_m\}$. Le nombre de nœuds, le degré, et le nombre de pôles sont liés par la relation : $m=n+p+1$.

Soit $x_r \leq \dots \leq x_s$ une séquence de nœuds à insérer dans l'intervalle $[t_a, t_b]$ c'est à dire $t_a \leq x_r$, et $x_s \leq t_b$. Les nœuds de multiplicité k seront insérer k fois (la multiplicité ne doit pas dépasser $(n-1)$).

Pour insérer ces nœuds les pôles P_{a-p}, \dots, P_{b-1} doivent être connus. Le nouveau vecteur nœuds sera : $U = \{u_0, \dots, u_{m+s-r}\}$, et les nouveaux pôles seront Q_i déterminés par l'algorithme suivant

Pour simplifier l'algorithme, les nœuds seront toujours notés t_i et les pôles P_j . [Böehm & Prautzsch]

Algorithme 1

$i = b + p - 1$; $l = b + s + p - r$

pour $j = b, b-1, \dots, a$

(*) $P_{l-p} = P_{i-p}$

si $x_j \leq t_i$ et $i > a$

alors $t_l = t_i$

$i = i - 1$; $l = l - 1$

aller à (*)

sinon pour $k = 1, 2, \dots, p$

$\beta = t_{l-k} - x_j$

si $\beta \leq 0$

alors $P_{l-p+k-1} = P_{l-p+k}$

sinon $\beta = \beta / (t_{l-k} - t_{l-p+k})$

$P_{l-p+k-1} = \beta \cdot P_{l-p+k-1} + (1-\beta) \cdot P_{l-p+k}$

Fin pour

$t_l = x_j$; $l = l - 1$

Fin pour

Fin

L'algorithme s'arrêtera lorsque tous les nœuds x_i seront insérés, pendant que les pôles de la partie gauche restent inchangés.

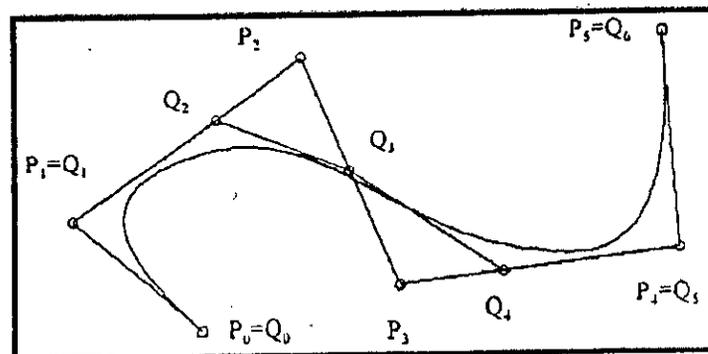


Figure 2.6. Insertion de nœud 0.5 pour une courbe B-spline non rationnelle avec $T = \{0,0,0,0,0,2,0,8,1,1,1,1\}$

2.2.1.2 Insertion de nœuds pour les courbes rationnelles

Soit une courbe B-spline rationnelle $C(t)$ de degré p définie par :

$$C(t) = \frac{\sum_{i=0}^n N_i^p(t) \cdot w_i \cdot P_i}{\sum_{j=0}^n N_j^p(t) \cdot w_j}$$

où P_i sont les pôles, w_i les poids et $N_{i,p}(t)$ sont les fonctions de base B-spline de degré p définies par le vecteur nœuds (non périodique) $t = \{t_0, \dots, t_m\}$.

Soit x_1, \dots, x_r une séquence de nœuds à insérer dans l'intervalle $[t_a, t_b]$.

Dans le cas rationnel l'algorithme 1 précédent devient [Böehm & Prautzsch '85] l'algorithme 2

l'algorithme 2

$i = b + p - 1$; $l = b + s + p - r$

pour $j = b, b-1, \dots, a$

(*) $P_{l-p} = P_{i-p}$

$w_{l-p} = w_{i-p}$

si $x_j \leq t_i$ et $i > a$

alors $t_l = t_i$

$i = i - 1$; $l = l - 1$

aller à (*)

sinon pour $k = 1, 2, \dots, p$

$\beta = t_{l+k} - x_j$

si $\beta \leq 0$

alors $w_{l-p+k-1} = w_{l-p+k}$

$P_{l-p+k-1} = P_{l-p+k}$

sinon $\beta = \beta / (t_{l-k} - t_{l-p+k})$

$w_{l-p+k-1} = \beta \cdot w_{l-p+k-1} + (1-\beta) \cdot w_{l-p+k}$

$P_{l-p+k-1} = (\beta \cdot P_{l-p+k-1} + (1-\beta) \cdot P_{l-p+k}) / w_{l-p+k}$

Fin pour

$t_l = x_j$; $l = l - 1$

Fin pour

Fin

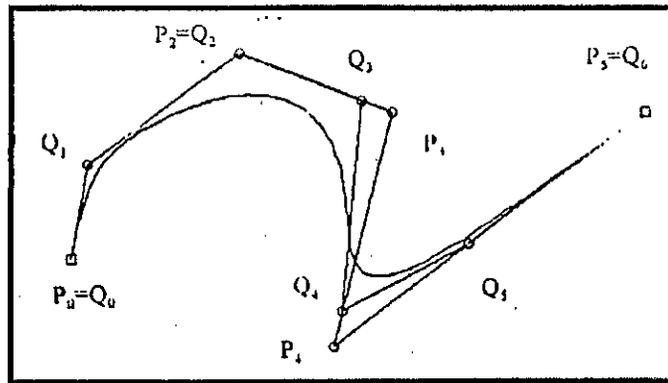


Figure 2.7. Insertion de nœud 0.8 pour une courbe B-spline rationnelle de degré 3 avec $T = \{0,0,0,0,0,2,0,6,1,1,1,1\}$ et $W = \{1,1,3,0,7,0,7,1,3,1\}$; $W_i = \{1,1,3,0,7,0,7,1,1,5,1,1,5,1\}$

2.2.2 Suppression de nœuds

Cette procédure permet de supprimer un nœud du vecteur nodal d'une courbe B-spline, la courbe devient alors de degré immédiatement inférieur. La suppression de nœuds est une transformation par approximation.

2.2.2.1 Suppression de nœuds pour les courbes non rationnelles

La suppression de nœuds est considérée comme étant le processus inverse de l'insertion de nœuds. L'insertion étant une transformation exacte. C'est à dire que la courbe obtenue est exactement similaire à la courbe initiale, mais la suppression de nœuds produit en général une approximation de la courbe initiale. Un nœud inséré peut être supprimé en retrouvant exactement la courbe initiale (nœuds surabondant).

Pour faciliter l'explication nous prenons un exemple spécifique : considérons une courbe P (les pôles de courbe initiale sont P_0^0, \dots, P_6^0)

Pour supprimer le nœud $t=t_6$ il faut et il suffit que la courbe soit de continuité géométrique C^1 à t , c'est à dire :

$$P_3^0 = \alpha_3 \cdot P_3^1 + (1 - \alpha_3) \cdot P_2^1 \quad \text{avec } \alpha_3 = (t - t_3) / (t_7 - t_3) \quad (34)$$

si encore la seconde dérivée est aussi continue en ce point le nœud peut alors être supprimé une seconde fois, les conditions suivantes sont alors nécessaires et suffisantes :

$$P_2^1 = \alpha_2 \cdot P_2^2 + (1 - \alpha_2) \cdot P_1^2, \quad P_3^1 = \alpha_3 \cdot P_3^2 + (1 - \alpha_3) \cdot P_2^2 \quad (35)$$

$$\alpha_i = \frac{t - t_i}{t_{i+p+2} - t_i}, \quad i=1,2 \quad (36)$$

P_2^2 peut être tiré à partir des équations (37)

$$P_2^2 = \frac{P_2^1 - (1 - \alpha_2) \cdot P_1^2}{\alpha_2}; \quad P_2^2 = \frac{P_3^1 - (1 - \alpha_3) \cdot P_2^2}{1 - \alpha_3} \quad (37)$$

Finalement pour que le nœud soit supprimable une tierce fois, il faut respecter les conditions suivantes :

$$P_1^2 = \alpha_1 \cdot P_1^3 + (1 - \alpha_1) \cdot P_0^3; \quad P_2^2 = \alpha_2 \cdot P_2^3 + (1 - \alpha_2) \cdot P_1^3; \text{ et } P_3^2 = \alpha_3 \cdot P_3^3 + (1 - \alpha_3) \cdot P_2^3;$$

$$\alpha_i = \frac{t - t_i}{t_{i+p+3} - t_i}, \quad i = 1, 2, 3. \quad (38)$$

de la première et troisième équation (38) nous avons

$$P_1^3 = \frac{P_2^2 - (1 - \alpha_1) \cdot P_0^3}{\alpha_1}; \quad P_2^3 = \frac{P_3^2 - (1 - \alpha_3) \cdot P_3^3}{1 - \alpha_3} \quad (39)$$

en remplaçant dans la deuxième équation de (39) les formules générale sont :

$$P_i^1 = \frac{P_i^0 + (1 - \alpha_i) \cdot P_{i-1}^1}{\alpha_i} \quad r - p \leq i \leq (2r - p - s - 1) / 2 \quad (40)$$

$$P_j^1 = \frac{P_j^0 - (1 - \alpha_j) \cdot P_{j+1}^1}{1 - \alpha_j} \quad (2r - p - s + 2) \leq j \leq r - s \quad (41)$$

$$\alpha_k = \frac{t - t_k}{t_{k+p+1} - t_k}$$

où $t = t_r$ est le nœud de multiplicité s à supprimer et $s \leq p$, avec $t \neq t_{r+1}$. Pour supprimer le nœud $t = t_r$ plusieurs fois alors à chaque étape supprimer le nœud, s et r seront décréments et les numéros exposant des pôles sont incréments ;
l'algorithme 3 suivant permet de supprimer k fois un nœud t . [Piegl & Tiller '94]

Algorithme 3

a=r-p+1

b=r-s-1

Pour l=1,2,...,k

 a=a-1

 b=b+1

 i=a

 j=b

 Tant que (j-i>1-1) faire

$$\alpha_i = (t - t_i) / (t_{i+p+1} - t_i)$$

$$\alpha_j = (t - t_{j-i+1}) / (t_{j+p+1} - t_{j-i+1})$$

$$P_i^l = (P_i^{l-1} - (1 - \alpha_i) \cdot P_{i-1}^l) / \alpha_i$$

$$P_j^l = (P_j^{l-1} - \alpha_j \cdot P_{i+1}^l) / (1 - \alpha_j)$$

Fin faire

Fin pour

Fin

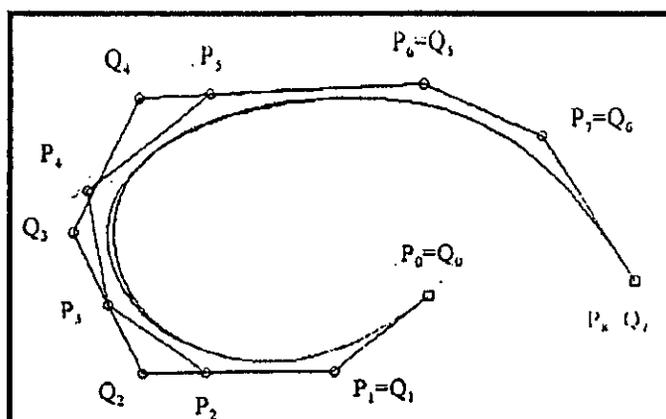


Figure 2.8. Suppression de nœud 0.4 pour une courbe B-spline non rationnelle de degré 4, Avec $T = \{0,0,0,0,0,0,2,0,4,0,6,0,8,1,1,1,1,1\}$

2.2.2.2. Suppression de nœuds pour les courbes rationnelles

Dans le cas des courbes rationnelles les formules générales deviennent
 $r - p \leq i \leq (2.r - p - s - 1) / 2 :$

$$w_i^1 = \frac{w_i^0 - (1 - \alpha_i) \cdot w_{i-1}^1}{\alpha_i} \text{ et } P_i^1 = \frac{P_i^0 - (1 - \alpha_i) \cdot P_{i-1}^1}{\alpha_i \cdot w_i^1}$$

$(2.r - p - s + 2) / 2 \leq j \leq r - s :$

$$w_j^1 = \frac{w_j^0 - \alpha_j \cdot w_{j+1}^1}{1 - \alpha_j} \text{ et } P_j^1 = \frac{P_j^0 - \alpha_j \cdot P_{j+1}^1}{(1 - \alpha_j) \cdot w_j^1} \quad (42)$$

avec : $\alpha_k = \frac{t - t_k}{t_{k+p+1} - t_k}$; ce qui conduit à l'algorithme 4 suivant

Algorithme 4

```

a=r-p+1
b=r-s-1
Pour l=1,2,...,k
  a=a-1
  b=b+1
  i=a
  j=b
  Tant que (j-i>l-1) faire
     $\alpha_i = (t - t_i) / (t_{i+p+1} - t_i)$ 
     $\alpha_j = (t - t_{j-l+1}) / (t_{j+p+1} - t_{j-l+1})$ 
     $w_i^l = (w_i^{l-1} - (1 - \alpha_i) \cdot w_{i-1}^l) / \alpha_i$ 
     $P_i^l = (P_i^{l-1} - (1 - \alpha_i) \cdot P_{i-1}^l) / (\alpha_i \cdot w_i^l)$ 
     $w_j^l = (w_j^{l-1} - \alpha_j \cdot w_{j+1}^l) / (1 - \alpha_j)$ 

```

```


$$P_j^i = (P_j^{i-1} - \alpha_j \cdot P_{j+1}^i) / ((1 - \alpha_j) \cdot w_j^i)$$

    i=i+1
    j=j-1
    Fin faire
  Fin pour
Fin

```

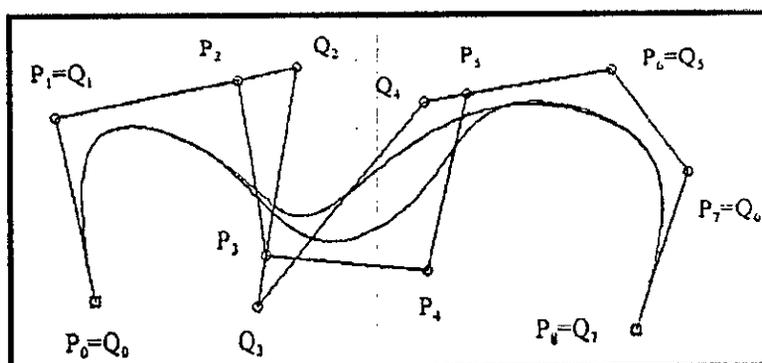


Figure 2.9. Suppression de nœud 0.5 pour une courbe B-spline rationnelle de degré 4. Avec $T = \{0,0,0,0,0,0,2,0,5,0,6,0,8,1,1,1,1\}$ et $W = \{1,1,2,0,8,1,3,0,7,1,3,0,8,1,2,1\}$; $W_s = \{1,1,2,0,72,1,648,1,6,0,8,1,2,1\}$

2.2.3. Subdivision et l'algorithme d'Oslo

L'algorithme d'Oslo [Bensalah '90] représente une généralisation de l'insertion et de la subdivision, il peut servir à :

- insérer un nœud et retrouver le nouveau polygone caractéristique,
- subdiviser une courbe B-spline en deux courbes B-spline au point $u=u_0$,
- déterminer la valeur paramétrique d'intersection de deux courbes B-spline différentes,
- convertir une courbe B-spline en courbes de Bézier,
- produire un algorithme de contournage B-spline.

2.2.3.1. Subdivision des courbes non rationnelles

Soit la courbe de base des fonctions B-spline $N_i^{k-1}(t)$ de degré $(k-1)$ définie par :

La fonction $C(t) = \sum_{i=0}^n p_i \cdot N_i^{k-1}(t)$ associée du vecteur nœuds $X = [X_0, \dots, X_{n+k-1}]$

On désire additionner des nœuds supplémentaires $Y = [Y_1, \dots, Y_l]$.

En posant $m=n+1$ la nouvelle répartition sera $Z = [Z_0, \dots, Z_{m+k-1}] = X \cup Y$, répartie en ordre non décroissant.

Si $N_{i,k}(t)$ sont les fonctions de base de B-spline sur cette nouvelle répartition, la courbe $C(t)$ peut être écrite sous la forme (43) avec les inconnues Q_i :

$$C(t) = \sum_{i=0}^n Q_i \cdot N_i^k(t) \text{ associée du vecteur nœuds } Z=[Z_0, \dots, Z_{m+k-1}] \quad (43)$$

Le problème consiste à calculer les pôles Q_i en fonction des P_i , de k , n , m X et Z .

La méthode utilisée pour résoudre ce problème est basée sur le calcul récurent. Les pôles Q_i peuvent être écrits sous forme d'une combinaison linéaire des P_i (44).

$$Q_j = \sum_{i=1}^n \alpha_{i,k}(j) \cdot P_i \quad (44)$$

On peut montrer que :

$$\alpha_{i,1}(j) = N_{i,1}(Z_j)$$

$$\alpha_{i,2}(j) = N_{i,2}(Z_{j+1})$$

Les relations pour $k > 2$ ne sont pas si simple, mais la relation entre les $N_i^k(t)$ et les $N_i^{k-1}(t)$ est donnée par :

Pour tout t :

$$N_{i,k}^k(t) = \sum_{j=0}^m \alpha_{i,k}(j) \cdot N_i^k(t) \quad (45)$$

$$\text{avec : } \alpha_{i,k}(j) = (X_{i+k} - X_i) \cdot [X_i, \dots, X_{i+k}] \cdot \Phi_{j,k}(t) \quad (46)$$

$$\text{et } \Phi_{j,k}(t) = (t - a_j)_+^0 \cdot \prod_{r=1}^{k-1} (t - Z_{j+r}) \quad (47)$$

$$\text{et } (t - a_j)_+^0 = \begin{cases} 1 & \text{si } t > a_j \\ 0 & \text{0 ailleurs} \end{cases}$$

a_j est un nœud de $[Z_j, \dots, Z_{j+k}]$, et $[X_i, \dots, X_{i+k}]$. $\Phi_{j,k}(t)$ est la différence divisée équivalente.

On peut aussi avoir une relation de récurrence entre les $\alpha_{i,k}(t)$:

Supposons que $X_{i+k} > X_i$ et que vient de l'équation (46)

Or $\alpha_{i,1}(j) = 1$ si $X_i \leq Z_j < X_{i+1}$ et $\alpha_{i,1}(j) = 0$ ailleurs

De plus pour $k \geq 2$ et pour tout i, j

$$\alpha_{i,k}(j) = (Z_{j+k-1} - X_i) \cdot \beta_{i,k-1}(j) + (X_{i+k} - Z_{j+k-1}) \cdot \beta_{i+1,k-1}(j) \quad (48)$$

$$\text{avec } \beta_{i,k}(j) = \begin{cases} \alpha_{i,k}(j) / (X_{i+k} - X_i) & \text{si } X_{i+k} > X_i \\ 0 & \text{ailleurs} \end{cases} \quad (49)$$

Les Q_i peuvent être déterminés pour les valeurs de t définies par les deux algorithmes suivants : [Bensalh '90], [Sablonnière '87-1] [Asma'99]

Algorithme 5

Pour $k > 1$ et j, s sont tel que :

$X_{s+2-k}, \dots, X_{s+k-1}$ et $Z_{j+1}, \dots, Z_{j+k-1}$
Avec $X_{s+2-k} \leq \dots \leq X_s < X_{s+1} \leq \dots \leq X_{s+k-1}$
 $X_s \leq Z_j \leq X_{s+1}$

$\alpha(s, 1) = 1$

$s_2 = s$

Pour $r = 1, 2, \dots, k-1$

$\beta_1 = 0$

$Z_j = Z_{j+r}$

Pour $i = s_2, s_2+1, \dots, s$

$d_1 = Z_j - X_j$

$d_2 = X_{j+r} - Z_j$

$\beta = \alpha(i, r) / (d_1 + d_2)$

$\alpha(i-1, r+1) = d_2 \cdot \beta + \beta_1$

$\beta_1 = d_1 \cdot \beta$

Fin pour

$\alpha(s, r+1) = \beta_1$

$s_2 = s_2 - 1$

Fin pour

Fin pour

Fin

Les

Q_i peuvent être calculés par l'algorithme 6 suivant sachant que :

$Q_j = \sum_{i=s-k+1}^s \alpha_{i,k}(t) \cdot P_i$ et k, j, s, X_i ainsi que Z_i définis suivant l'algorithme 5 [Bensalh

'90], [Sablonnière '87-1] [Asma'99].

Algorithme 6

$s_2 = s - k + 1$

Pour $i = s_2, s_2+1, \dots, s$

$P_{i,1} = P_i$

Fin pour

Pour $r = 1, 2, \dots, k-1$

$s_2 = s_2 - 1$

$k_r = k - r$

$Z_j = Z_{j+k_r}$

Pour $i = s, s-1, \dots, s_2$

$d_1 = Z_j - X_j$

$d_2 = X_{j+k_r} - Z_j$

$P_{i,r+1} = (d_1 \cdot P_{i,r} + d_2 \cdot P_{i+1,r}) / (d_1 + d_2)$

Fin pour

Fin pour

$Q_j = P_{s,k}$

Fin

$(d_1+d_2)>0$ tant que $X_s < X_{s+1}$ [Bensalh '90][Asma'99].

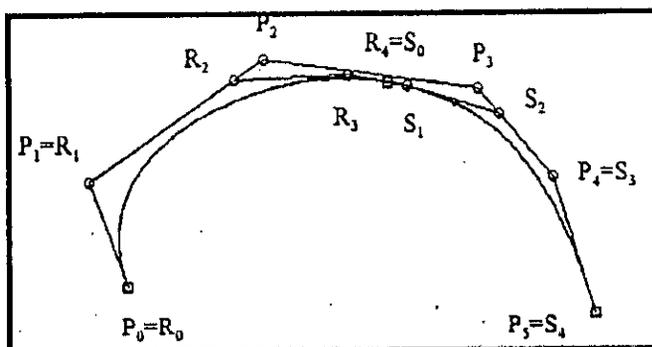


Figure 2.10. Subdivision d'une courbe B-spline de degré 3 au point $t=0.5$ avec $T = \{0,0,0,0,0,0,0,3,0,6,1,1,1,1\}$ et $T1 = \{0,0,0,0,0,0,6,1,1,1,1,1\}$ et $T2 = \{0,0,0,0,0,0,2,1,1,1,1,1\}$

2.2.3.2 subdivision des courbes rationnelles

Soit la courbe rationnelle B-spline $C(t)$ à base $N_{i,k}(t)$:

$$C(t) = \frac{\sum_{i=0}^n w_i \cdot P_i \cdot N'_{i,k}(t)}{\sum_{i=0}^n w_i \cdot N'_{i,k}(t)} \text{ dans } X=[X_0, \dots, X_{n+k-1}] \quad (50)$$

et $Y=[Y_1, \dots, Y_1]$, le vecteur nœuds supplémentaire à rajouter. En posant $m=n+1$, la nouvelle répartition sera $Z=[Z_0, \dots, Z_{m+k-1}]=X \cup Y$, répartie en ordre non décroissant. La courbe $C(t)$ peut s'écrire sous la forme suivante :

$$C(t) = \frac{\sum_{i=0}^n \sigma_i \cdot P_i \cdot N_i^k(t)}{\sum_{i=0}^n \sigma_i \cdot P_i \cdot N_i^k(t)} \text{ dans } Z=[Z_0, \dots, Z_{m+k-1}]. \quad (51)$$

Avec $N_{i,k}(t)$ les fonctions de base B-spline sur cette nouvelle répartition et les Q_i , σ_i les pôles et les poids inconnus à déterminer.

L'algorithme 6 peut être généralisé par l'algorithme 7 suivant :

Après utilisation de l'algorithme 5 les Q_i , et σ_i peuvent être calculés par l'algorithme 7 suivant sachant que $Q_j = \sum_{i=s-k+1}^s (\alpha_{i,k}(t) \cdot P_i) / \sigma_i$, $\sigma_j = \sum_{i=s-k+1}^s \alpha_{i,k}(t) \cdot w_i$ et k, j, s, X_i ainsi que Z_i définis suivant l'algorithme 5 [Asma '99].

Algorithme 7

```

s2=s-k+1
Pour i=s2,s2+1,...,s

Pi,1=Pi
Wi,1=wi
Fin pour
Pour r=1,2,...,k-1
s2= s2-1
kr=k-r
Zj=Zj+kr
Pour i=s,s-1,..., s2
    d1=Zj-Xj
    d2= Xj+kr-Zj
    Pi,r+1=(d1.Pi,r+d2.Pi+1,r)/(d1+d2)
    wi,r+1=(d1.wi,r+d2.wi+1,r)/(d1+d2)
Fin.pour
Fin pour
Qj=Ps,k
σj = ws,k
Fin
(d1+d2)>0 tant que Xs<Xs+1 [Asma '99].

```

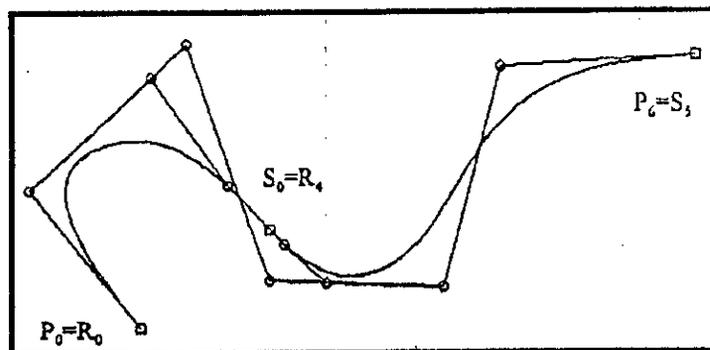


Figure 2.11. courbe B-spline rationnelle de degré 3 subdivisé en deux courbes à $t=0.5$

Avec

$$T = \{0,0,0,0,0,2,0,6,0,8,1,1,1,1\} \text{ et } T1 = \{0,0,0,0,0,4,1,1,1,1\}; T2 = \{0,0,0,0,0,2,0,6,1,1,1,1\}$$

$$W = \{1,1,2,0,8,1,0,8,1,2,1\}; W1 = \{1,1,2,0,87,0,92,0,92\}; W2 = \{0,92,0,93,0,93,0,8,1,2\}$$

2.3. Changement de base

Nous développons dans cette partie les algorithmes permettant le passage du modèle de Bézier au modèle B-spline et inversement.

2.3.1. B-spline –Bézier

Détermination de la forme Bézier d'une courbe B-spline .

2.3.1.1 . Le polygone local de Bézier d'une courbe B-spline :

Soit la courbe B-spline de degré k ; pour $t \in [t_i, t_{i+1}]$ le i -ème morceau de la courbe peut être représenté par :

$$C(t) = \sum_{i=l-k}^n P_i \cdot N_i^{k+1}(t) \quad (52)$$

Ce l -ième morceau est une courbe de Bézier de degré k dont les pôles peuvent être calculés en utilisant les algorithmes 42.a et 42.b suivants :

On pose

$$P_{i,0}^0 = P_i \quad \text{pour } l-k \leq i \leq l \text{ et}$$

$$\text{pour } 0 \leq r \leq k, 1 \leq j \leq k-r, \text{ et } l+j+r-k \leq i \leq l$$

$$\alpha = \frac{t_1 - t_i}{t_{i+k+l-j-r} - t_i} \quad (53a)$$

$$P_{i,j}^r = \alpha \cdot P_{i,j-1}^r + (1-\alpha) \cdot P_{i-1,j-1}^r$$

$$\text{pour } 0 \leq j \leq k, 1 \leq r \leq k-1, \text{ et } l+j+r-k \leq i \leq l$$

$$\beta = \frac{t_{i+1} - t_i}{t_{i+k+l-j-r} - t_i} \quad (53b)$$

$$P_{i,j}^r = \beta \cdot P_{i,j}^{r-1} + (1-\beta) \cdot P_{i-1,j}^{r-1}$$

Les pôles de la l -ième courbe de Bézier seront :

$$Q_{i,j} = P_{i,k-j}^j \quad \text{pour } j=0,1,\dots,k \quad (54)$$

Alors les $k+1$ pôles $Q_{i,j}$ ($j=0,1,\dots,k$) de la courbe de Bézier locale (55) sont identique à la l -ième portion de la courbe B-spline définie par son support local (52) :

$$Q_l(t) = \sum_{i=0}^k Q_{l,i} \cdot B_i^k(t) \quad (55)$$

Les deux algorithmes (53a) et (53b) doivent fonctionner simultanément. Tous les polygones locaux de Bézier de toute la courbe B-spline peuvent donc être calculés.

2.3.1.1 Les polygones de Bézier d'une courbe B-spline :

La courbe B-spline d définie par ses pôles P_i ($i=0,\dots,n$), son ordre k , et son vecteur nœuds $T=\{t_0,\dots,t_m\}$ est un ensemble de courbes de Bézier de degré k . On peut démontrer qu'en assignant la multiplicité $(k-1)$ à tous les nœuds internes, et en calculant tous les nouveaux pôles par l'algorithme suivant, on retrouve les pôles des courbes locales de Bézier équivalentes : [Asma '99],[Eck &Lasser '92].

Pour $t_1 < t_{i+1} = t_{k-1}, \dots, t_1 < t_{i+1} = t_m$
 Si t_{i+1} est de multiplicité $s < k-1$
 Calculer $A_{i,p}(t_{i+1})$ par l'Algorithme 1 avec $p = k-1-s$
 Augmenter la valeur de m par p
 Pour $j > 1$ augmenter les indices des P_i par p
 Pour $i = 1, \dots, p$ remplacer
 Les t_{i+j} par t_{i+1}
 Les P_{i-p+i} par $A_{i-p+i,i}$

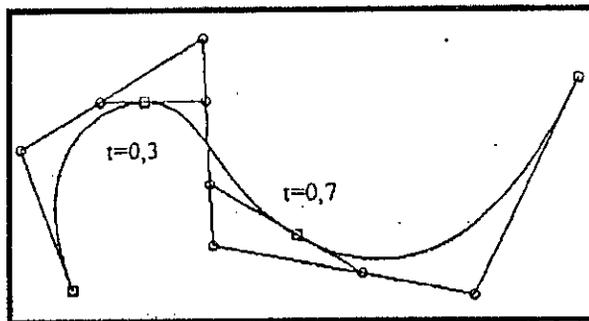


Figure 2.12. Courbe B-spline non rationnelle de degré 3 transformée en 3 courbes de Bézier non rationnelles de degré 3 avec $T = \{0,0,0,0,0,3,0,7,1,1,1,1\}$

2.3.2. Bézier - B-spline

Détermination de la forme B-spline d'une ou de plusieurs courbe de Bézier.

2.3.2.1 Polygone B-spline d'une courbe de Bézier

Soit la courbe de Bézier $C(t)$ de degré n , définie par ses $(n+1)$ pôles $p_i (i=0, \dots, n)$, d'après la propriété (§ I.3.1) des fonctions de base B-spline, la courbe B-spline définie par le

vecteur nœuds $T = \left\{ \underbrace{0, \dots, 0}_{n+1}, \underbrace{1, \dots, 1}_{n+1} \right\}$ et les mêmes pôles est la courbe de Bézier $C(t)$.

2.3.2.2. Polygone B-spline des courbes locales de Bézier :

Considérons un ensemble Q de courbes de Bézier de degré m raccordées avec la continuité C^{n-1} ; les pôles P_i de la courbe B-spline englobant toutes ces courbes peuvent être calculés à partir des polygones de l'ensemble Q .

Soit $Q = \{Q_{i,0}, \dots, Q_{i,m}, \dots, Q_{j,0}, \dots, Q_{j,m}\}$, le vecteur nœuds est calculé en assignant une multiplicité $(m+1)$ au premier et au dernier nœuds, et en mettant des intervalles proportionnelles aux valeurs limites du paramètre t de chaque courbe de

Bézier par rapport à la somme des intervalles. Pour chaque polygone local, P_i locaux sont calculés par (45.a) et (45.b) : [Asma'99]

En posant $P_{i,0}^0 = Q_{i,m-1+i}$ pour $1-m \leq i \leq 1$ et

$$P_{i,0}^0 = P_i \quad \text{pour } 1-m \leq i \leq 1 \text{ et}$$

pour $0 \leq k \leq m$, $1 \leq j \leq m-k$, et $1+j+k-m \leq i \leq 1$

$$\alpha = \frac{t_{1+j} - t_1}{t_{1+j} - t_1} \quad (56a)$$

$$P_{i,j}^k = \alpha \cdot P_{i,j-1}^k + (1-\alpha) \cdot P_{i-1,j-1}^k$$

pour $0 \leq j \leq m$, $1 \leq k \leq m-j$, et $1+j+k-m \leq i \leq 1$

$$\beta = \frac{t_{1+k} - t_1}{t_{1+k} - t_1} \quad (56b)$$

$$P_{i,j}^k = \beta \cdot P_{i,j}^{k-1} + (1-\beta) \cdot P_{i-1,j}^{k-1}$$

Les pôles de la courbe B-spline seront :

$$P_{1-j} = P_{1,j}^{m-j} \quad \text{pour } j=0,1,\dots,m \quad (57)$$

L'algorithme donne ainsi les pôles de la courbe B-spline englobant les courbes de Bézier définies par l'ensemble Q.

2.3.3 Algorithme WM

A partir de la subdivision définie, les pôles de la courbe B-spline non uniforme se déterminent comme suit : [Meyer '87]

Considérons $t \in [t_i, t_{i+1}]$ et $i \in [p, np-1]$, de l'algorithme de Boor nous avons la formule de la B-spline locale :

$$C(t) = \sum_{j=i-p}^i P_j \cdot N_j^p(t)$$

et comme l'arc de Bézier dans l'intervalle $[t_i, t_{i+1}]$ peut s'écrire :

$$C(t) = C' \left(\frac{t-t_i}{t_{i+1}-t_i} \right) = \sum_{j=0}^p Q_{j,i} \cdot B_j^p \left(\frac{t-t_i}{t_{i+1}-t_i} \right)$$

où

$Q_{j,i}$ sont les pôles de l'arc de Bézier

$C'(t) = C_h(t)$ ou un point extrémité de $C_h(t)$

de ces deux expressions nous obtenons :

$$\sum_{j=i-p}^i P_j \cdot N_{j,p}^p(t) = \sum_{k=0}^p Q_{k,i} \cdot B_k^p \left(\frac{t-t_i}{t_{i+1}-t_i} \right)$$

Si on connaît la décomposition d'une B-spline non uniforme dans la base de Bernstein de degré p sur l'intervalle $[t_i, t_{i+1}]$:

$$N_{i,p}(t) = \sum_{k=0}^p \varphi_{k,p,j} \cdot B_k^p \left(\frac{t-t_i}{t_{i+1}-t_i} \right)$$

La courbe B-spline donc s'exprime :

$$C(t) = \sum_{j=0}^p Q_{j,i} \cdot B_j^p \left(\frac{t-t_i}{t_{i+1}-t_i} \right)$$

$$C(t) = \sum_{j=i}^i \sum_{k=0}^p \varphi_{k,p,j} \cdot P_j \cdot B_k^p \left(\frac{t-t_i}{t_{i+1}-t_i} \right)$$

Comme les fonctions de Bernstein $B_k^p(t)$ ($k \in [0, p]$) forment une base de l'espace vectoriel de dimension $p+1$ des polynômes de degré p , les pôles B-spline P_j peuvent alors être calculés par la résolution du système linéaire suivant :

$$\sum_{j=i-p}^i \varphi_{k,p,i} \cdot P_j - Q_{k,i} = 0 \quad \text{pour } k = 0, \dots, p$$

Ce système peut aussi s'écrire sous la forme matricielle :

$$\{\varphi_{k,p,j}\} \{P_j\} = \{Q_{k,i}\}$$

Pour $j \in [i-p, i]$, $k \in [0, p]$ et $i \in [p, np-1]$

où $\{\varphi_{k,p,j}\}$ est la matrice de transformation de représentation B-spline en représentation Bézier,

$\{P_j\}$ sont les pôles de la courbe B-spline pour $t \in [t_i, t_{i+1}]$

$\{Q_{k,i}\}$ les pôles de l'arc de Bézier dans l'intervalle $[t_i, t_{i+1}]$

Ce qui peut s'écrire :

$$\begin{bmatrix} \varphi_{0,p,i-p} & \varphi_{0,p,i-p+1} & \dots & \varphi_{0,p,i} \\ \varphi_{1,p,i-p} & \varphi_{1,p,i-p+1} & \dots & \varphi_{1,p,i} \\ \dots & \dots & \dots & \dots \\ \varphi_{k,p,i-p} & \varphi_{k,p,i-p+1} & \dots & \varphi_{k,p,i} \\ \dots & \dots & \dots & \dots \\ \varphi_{p,p,i-p} & \varphi_{p,p,i-p+1} & \dots & \varphi_{p,p,i} \end{bmatrix} \begin{bmatrix} P_{i-p} \\ P_{i-p+1} \\ \dots \\ P_j \\ \dots \\ P_i \end{bmatrix} = \begin{bmatrix} Q_{0,i} \\ Q_{1,i} \\ \dots \\ Q_{k,i} \\ \dots \\ Q_{p,i} \end{bmatrix}$$

p étant le degré

$j \in [i-p, i]$, $k \in [0, p]$ et $i \in [p, np-1]$.

La solution est :

$$\{P_j\} = \{\varphi_{k,p,j}\}^{-1} \{Q_{k,i}\}$$

Les formules de récurrence sur les fonctions Bernstein et sur les fonctions B-spline permettent d'établir une formule de récurrence pour calcul des coefficients $\{\varphi_{k,p,j}\}$ de la suivante : [Meyer'87].

$$\begin{aligned} \varphi_{k,p,j} = & \frac{k}{p} \left\{ \varphi_{k-1,p-1,j} \left(\frac{t_i-t_j}{t_{j+p}-t_j} \right) + \varphi_{k-1,p-1,j+1} \left(\frac{t_{i+p+1}-t_{i+1}}{t_{j+p+1}-t_{j+1}} \right) \right\} \\ & + \frac{p-k}{p} \left\{ \varphi_{k-1,p-1,j} \left(\frac{t_i-t_j}{t_{j+p}-t_j} \right) + \varphi_{k-1,p-1,j+1} \left(\frac{t_{i+p+1}-t_{i+1}}{t_{j+p+1}-t_{j+1}} \right) \right\} \end{aligned} \quad (58)$$

Dans tous les intervalles $[t_i, t_{i+1}]$, $i \in [p, n-1]$: $C(t) = C'(t)$.

En effet pour $t \in [t_n, t_{np-1}]$, sur tous les intervalles $[t_i, t_{i+1}]$, en utilisant l'algorithme WM, nous pouvons calculer $p+1$ pôles B-spline P_{i-p}, \dots, P_i . alors les pôles B-spline, qui sont définis dans les intervalles voisins $[t_i, t_{i+1}]$, et $[t_{i+1}, t_{i+2}]$, $i \in [p, n-1]$, ont p pôles communs.

Successivement, nous pouvons complètement définir tous les pôles de cette représentation B-spline non uniforme, et obtenir le polygone P_i .

L'algorithme WN nous propose un système linéaire entre le polygone de Bézier et le polygone B-spline local. En résolvant ce système, nous pouvons résoudre le problème consistant à représenter une ligne composée par un ensemble d'arcs de Bézier. Cet algorithme est également valable pour le problème de surface. Il s'agit là d'une méthode locale dans la mesure où elle traite le problème tronçon par tronçon. La base Bernstein et la base B-spline. Ces coefficients sont déterminés suivant une procédure récurrente. L'algorithme WM est une méthode globale, facile à implémenter et à utiliser.

CHAPITRE 3

Les principales méthodes de réduction de degré des courbes de Bézier non rationnelles

3. Les principales méthodes de réduction de degré des courbes de Bézier non rationnelles .

Problématique

La plupart des systèmes CAD (computer aided design) pour la modélisation des courbes et des surfaces utilisent une représentation paramétriques de différent fonctions de base ,et maximum de degré possible , en vu de leur donner d'avantage plus de variance de forme ,par conséquent on à besoin de communication ,et d'échange de données entre différent système ,pour cela on utilise des méthodes effective de conversion par approximation des courbes et des surfaces ,on combine la réduction de degré avec la subdivision .

Dans le processus de conversion par approximation l'erreur d'approximation dû à la réduction de degré est inévitable, donc on doit minimiser cette erreur tout en respectant certaines propriétés géométriques des courbes et des surfaces ,comme répondre à un ordre de continuité géométriques en respectant plusieurs tolérances spécifiques à l'application (position ,tangente ,courbure , torsion,...).

Si on subdivise la courbe de degré n en deux courbes de degré n , qu'on réduit au degré m ($m < n$) voulu , l'erreur d'approximation est d'autant plus petite que le nombre de subdivision est grand , l'utilisation de l'algorithme de subdivision permet de respecter la tolérance de position en le combinons à la réduction de degré . Ce qui donne un ensemble de portions de courbes réduites. Cette idée est la base de toutes les méthodes de conversion actuelles, appelées aussi sub-réduction.

Nous présenterons dans ce chapitre les différentes méthodes des conversions par approximation,

Faisons des approximation raisonnables de façon à surmonter les contraintes rencontrés dans le processus de transmission de données entre systèmes de C.F.A .O (CAD.CAM).

3.1.Principe général de conversion par approximation :

le principe général de la conversion par approximation est donné par l'organigramme (Fig .3.1)

Une courbe $y(t)$ de degré n de pôles (points de contrôles) P_i sera convertit en respectant un ordre de continuité k , par réduction de degré en courbe de degré n de pôles Q_i ,puis on subdivise la courbe par l'algorithme de De Casteljaou en deux courbes de même degré m lorsque la distance maximale entre la courbe convertie et la courbe mère ne respecte pas la tolérance requise . Chacune des deux courbes subira la même chose à nouveau jusqu'à la conversion de toutes la courbes mère et fin de subdivision.

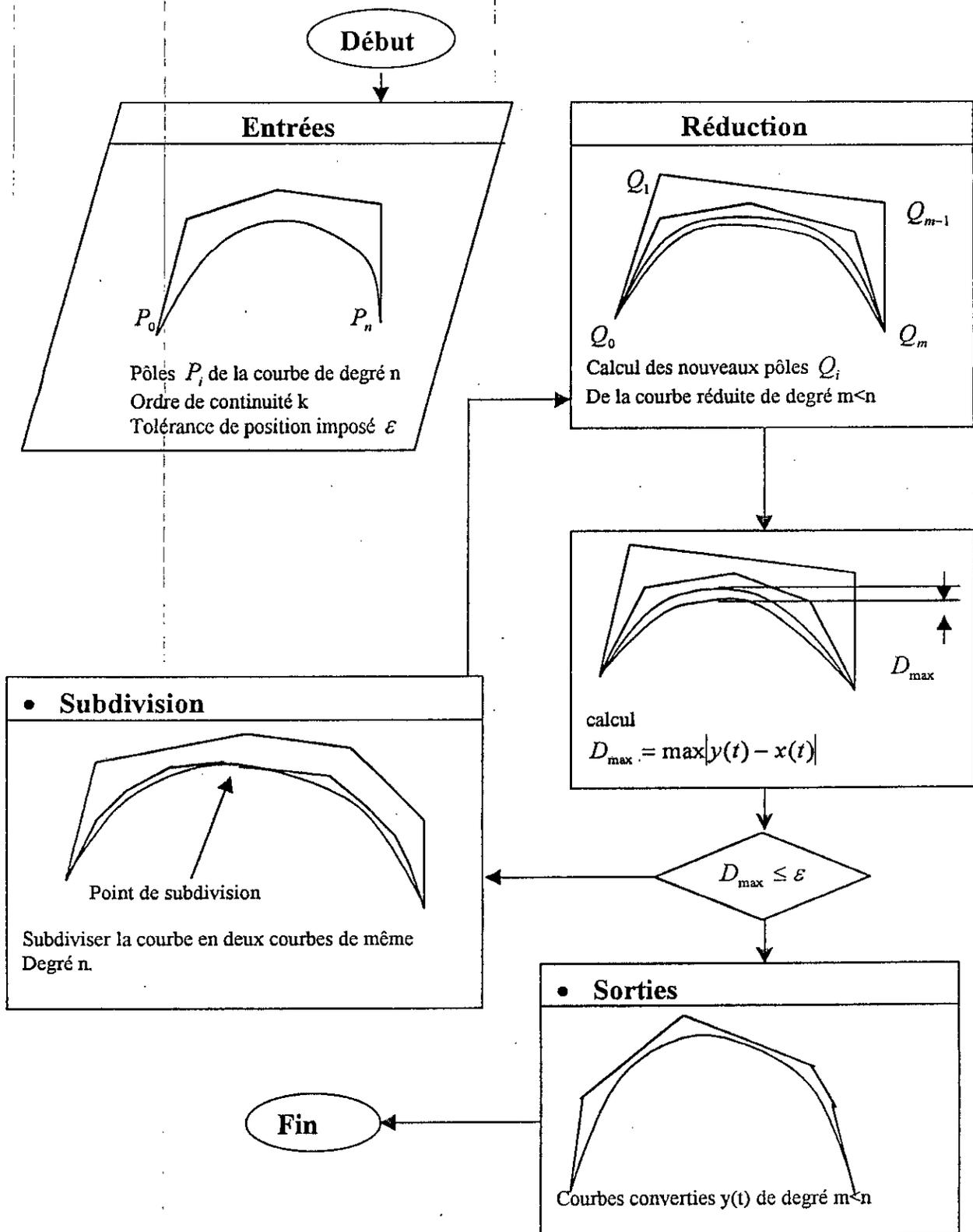


Figure.3.1. :organigramme de conversion par approximation

3.2. Méthodes et Algorithmique de réduction.

3.2.1. Méthode de Forrest

En inversant l'algorithme de l'élévation de degré, Forrest a proposé un algorithme de réduction [Forrest '72][Forrest '90], la méthode présentée ici est celle de Piegl et Tiller basé sur les fondements de Forrest.[Asma'99].

3.2.1.1. Algorithme de réduction

soit une courbe $P(t)$ de degré n définie par ces $n+1$ points de contrôles P_i à réduire en une courbe $Q(t)$ de degré n définie par ses n points de contrôles Q_i .

la réduction de degré d'une courbe de Bézier non rationnelle proposée dans le paragraphe 2.1.3.1 donne deux solutions généralement différentes $R(t)$ (17.a) produisent une bonne approximation au voisinage de $t=0$ et $S(t)$ (17.b) produisent une bonne approximation au voisinage de $t=1$.

L'approximation proposée par Forrest consiste à prendre pour la courbe réduite $Q(t)$ les $(n/2)$ premiers points de contrôles de la courbe $R(t)$ et les $(n/2)$ derniers points de contrôles de la courbe $S(t)$.

Selon que n est pair ou impair, L'algorithme de Forrest s'écrit

Pour n pair :

Pour $i=0, \dots, (n-2)/2$
 $Q_i = R_i$

Fin pour

Pour $i=(n+2)/2, \dots, n-1$
 $Q_i = S_i$

Fin pour

Fin

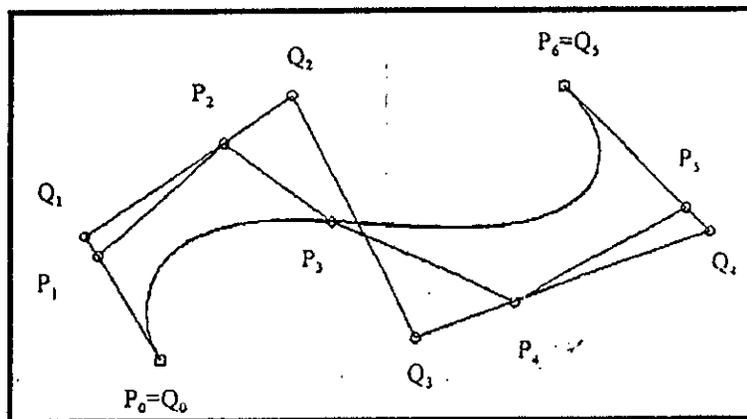


Figure 3.1.a. Courbe de Bézier de degré 6 réduite au degré 5

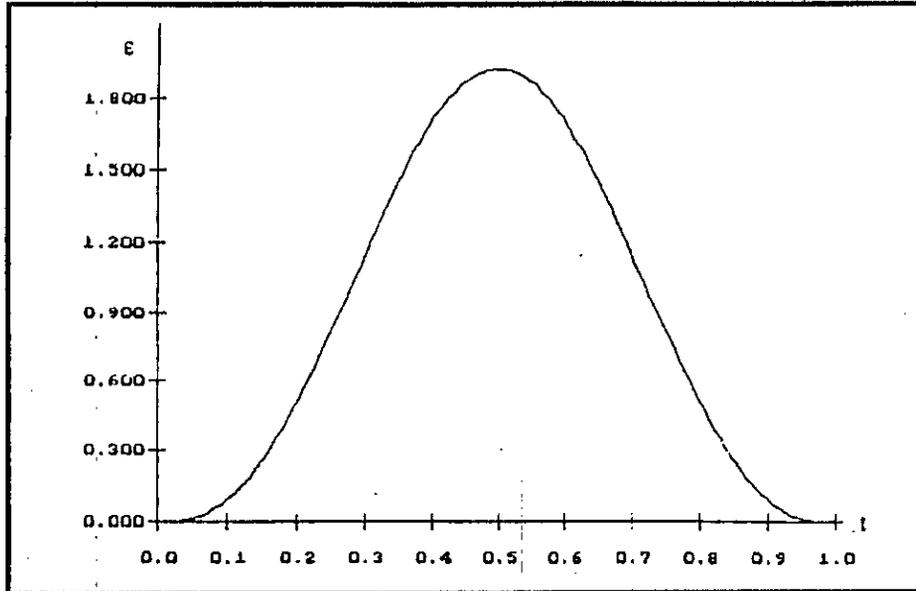


Figure 3.1.b. Erreur de position sur la courbe précédente

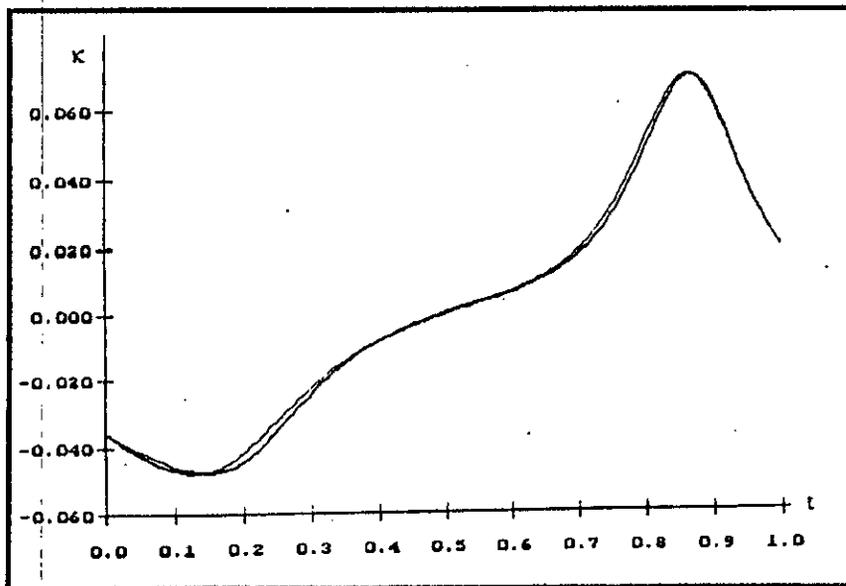


Figure 3.1.c. Variation de courbure, courbe précédente et sa courbe réduite

L'erreur de cette approximation est calculée par [Piegl & Tiller '95]

$$|P(t) - Q(t)| = \left| B_{(n+1)/2}^n(t) \cdot \left[P_{(n+1)/2} - \frac{1}{2} \cdot (Q_{(n-1)/2} - Q_{(n+1)/2}) \right] \right|$$

il est intéressant de noter que l'erreur maximale est obtenue à $t=1/2$ (figure 3.1.b) puisque la fonction de base du milieu $B_{(n+1)/2}^n(t)$ atteint sa valeur maximale à $t=1/2$.

Pour n impair :

Pour $i=0, \dots, (n-3)/2$

$$Q_i = R_i$$

Fin pour

Pour $i=(n+1)/2, \dots, n-1$

$$Q_i = S_i$$

Fin pour

$$Q_{(n-1)/2} = (R_{(n-1)/2} + S_{(n-1)/2}) / 2$$

Fin

L'erreur de cette approximation est calculée par [Piegl & Tiller '95]

$$|P(t) - Q(t)| = \frac{1}{2} \cdot \left(1 - \frac{n-1}{2 \cdot n}\right) \left| \left(B_{(n-1)/2}^n(t) - B_{(n+1)/2}^n(t) \right) \cdot (R_{(n-1)/2} - S_{(n+1)/2}) \right|$$

noter que cette fonction s'annule à $t=1/2$, puisque $B_{(n-1)/2}^n(1/2) = B_{(n+1)/2}^n(1/2)$ il est évident à cause de la symétrie des polynômes de Bernstein, et que l'erreur possède deux pics (figure 3.2.b) calculés par l'équation quadratique suivante :

$$t^2 - t + \frac{n}{4 \cdot (n+1)} = 0$$

En considérant la formule précédente de l'erreur, nous constatons que :

- Cette erreur a un contrôle paramétrique de l'erreur. le maximum géométrique n'est pas nécessairement atteint à la même valeur paramétrique t .
- L'erreur est connue précisément le long de la courbe, ce qui évite de calculer l'erreur en chaque point de la courbe,
- L'erreur maximale est poussée vers le milieu de la courbe,
- La courbe initiale et la courbe réduite ont une continuité d'ordre $(n-1)/2$ aux extrémités, ce qui est une propriété très importante à considérer lors de lors d'une réduction de degré.

L'application successive de cette méthode produit une réduction du degré n au degré $m < n$, par une réduction de degré à chaque étape. les erreurs d'approximation s'ajoutent à chaque étape. Ce qui nous conduit à l'algorithme FOR suivant :

On considère la courbe $P(t)$ de degré n à réduire en $Q(t)$ de degré $m < n$

Algorithme FOR

Données : P_i ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 k ordre de réduction

```

Pour i=0,...,n
   $W_i = P_i$ 
Fin pour
Pour r=1,2,...,k
   $R_0 = W_0$ 
   $S_{n-1} = W_n$ 
  Pour i=1,2,...,n-1
     $\beta_i = i / (n-i)$ 
     $R_i = (1 + \beta_i) \cdot W_i - \beta_i \cdot R_{i-1}$ 
     $W_i = R_i$ 
  Fin pour
  Pour i=n-1,n-2,...,1
     $\gamma_i = (n-i) / i$ 
     $S_{i-1} = (1 + \gamma_i) \cdot W_i - \gamma_i \cdot S_i$ 
     $W_i = S_i$ 
  Fin pour
  n=n-1
Fin pour
Si (n-k) est pair alors
  Pour i=0,...,((n-k)/2)-1
     $Q_i = R_i$ 
  Fin pour
  Pour i=((n-k)/2+1),..., (n-k)
     $Q_i = S_i$ 
  Fin pour
Sinon (n-k) impair
  Pour i=0,1,...,(n-k-1)/2
     $Q_i = R_i$ 
  Fin pour
  Pour i=(n-k-3)/2,...,n-k, pas 1
     $Q_i = S_i$ 
  Fin pour
   $Q_{(n-k+1)/2} = (R_{(n-k+1)/2} + S_{(n-k+1)/2}) / 2$ 
Fin Si
Résultats :  $Q_i$  (i=0,1,...,n-k) polygone de contrôles de la courbe de degré (n-k)
Fin

```

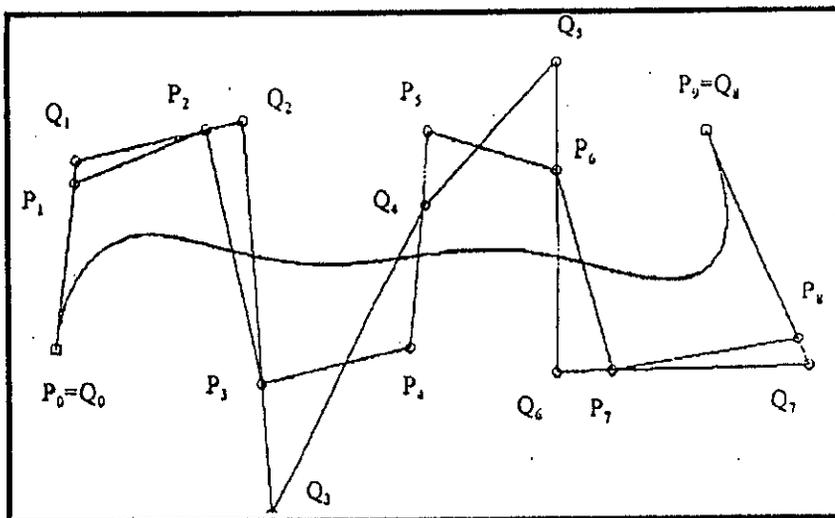


Figure 3.2.a. Courbe de Bézier de degré 9 réduite au degré 8

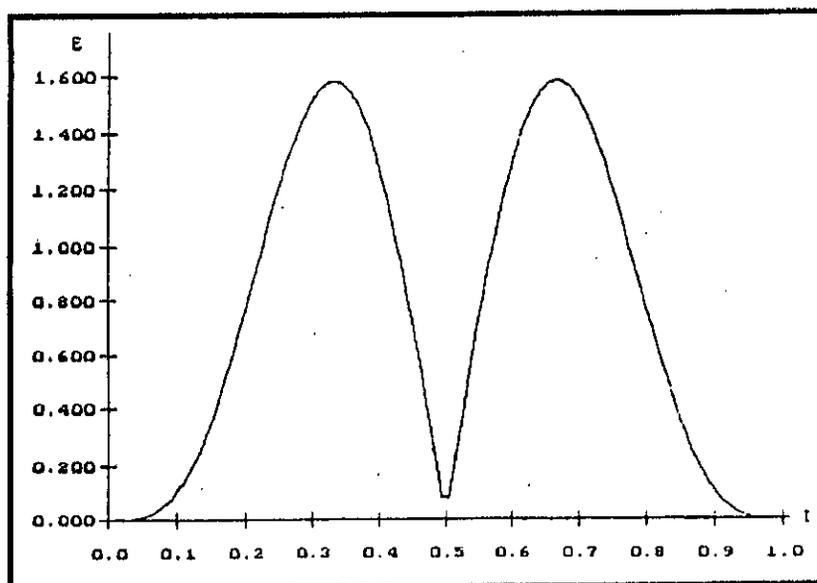


Figure 3.2.b. Erreur de position de la courbe précédente

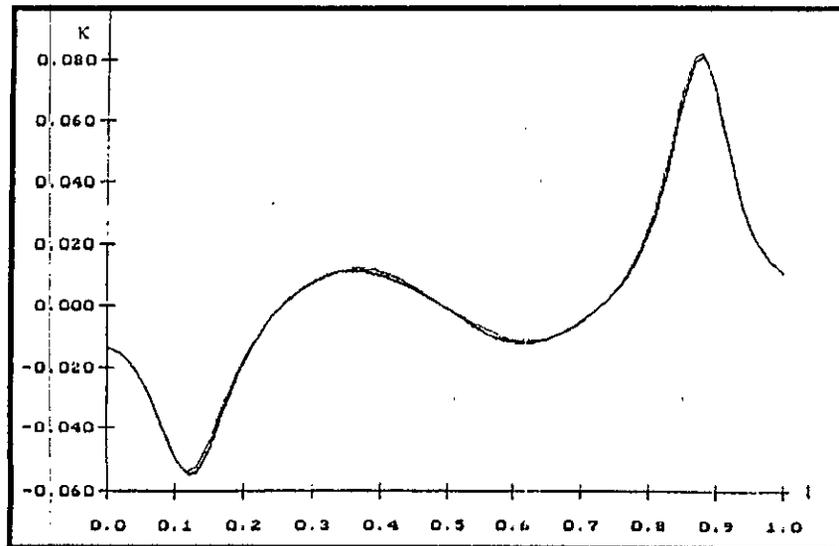


Figure 3.2.c. Variation de courbure *réduite* courbe précédente et sa courbe précédente

3.2.1.2. Algorithme de conversion par approximation :

l'algorithme FOR présenté dans le paragraphe précédent, produit une bonne approximation en imposant une continuité maximale $C^{(n-m)/2}$ aux extrémités de la courbe. En associant à cette méthode de réduction l'algorithme de subdivision (11) nous obtenons l'algorithme de conversion par approximation CONVER-FOR permettant de respecter une tolérance. L'algorithme de subdivision qui sera utilisé dans toutes les méthodes de conversion par approximation, est l'algorithme SUB suivant :

Algorithme SUB

Données : P_i ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 t_0 abscisse curviligne du point de subdivision

```

pour i=0,1,..n
     $P_i^0(t_0) = P_i$ 
Fin pour
Pour j=1,2,..n
    Pour i=0,1,..,n-j
         $P_i^j(t_0) = (1-t_0) \cdot P_i^{j-1}(t_0) + t_0 \cdot P_{i+1}^{j-1}(t_0)$ 
    Fin pour
Fin pour

```

Pour $i=0, 1, \dots, n$
 $Q_i = P_0^i(t_0); \quad C_i = P_i^{n-i}(t_0)$
 Fin pour
 Résultats : Q_i, C_i ($i=0, 1, \dots, n$) deux courbes de degré n
 Fin

Algorithme CONV-FOR

Données : P_i^0 ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 k : ordre de réduction
 ε_0 : tolérance de position
 N : nombre de pas
 $n1=0 ; n2=0$
 pour $j=0, 1, \dots, n1$
 FOR ($P^j, k \rightarrow Q$)
 $T=0 ; E_{\max} = 0$
 Tant que ($t < 1$) faire
 $t=t+(1/N) ; d=|Q(t) - P^j(t)|$
 Si $d > E_{\max}$ alors $E_{\max} = d$
 Fin faire
 Si $E_{\max} > \varepsilon_0$ alors
 SUB ($P^j, t_0 = 1/2 \rightarrow P0, P1$)
 $N1=n1+2 ; N=N \text{ div } 2$
 Pour $i=0, 1, \dots, n$
 $P_i^{n1-1} = P0_i ;$
 $P_i^{n1} = P1_i ;$
 Fin pour
 Sinon faire
 Pour $j=0, 1, \dots, n-k$
 $Q_i^{n2} = Q_i$
 Fin pour
 $n2=n2+1$
 Fin Si
 Fin pour
 Résultats : Q_i^j ($i=0, 1, \dots, n-k$) ($j=0, 1, \dots, n2-1$)
 $n2$: nombre des courbes de degré $(n-k)$
 Fin

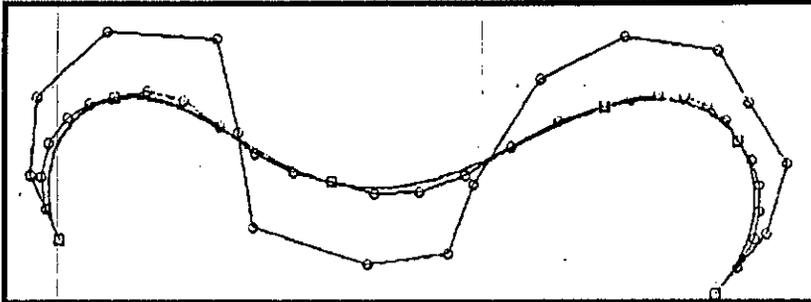


Figure 3.3.a. courbe de degré 16 réduite en 5 courbes de degré 6
($\varepsilon_0 = 0.01$ et $k = 10$)

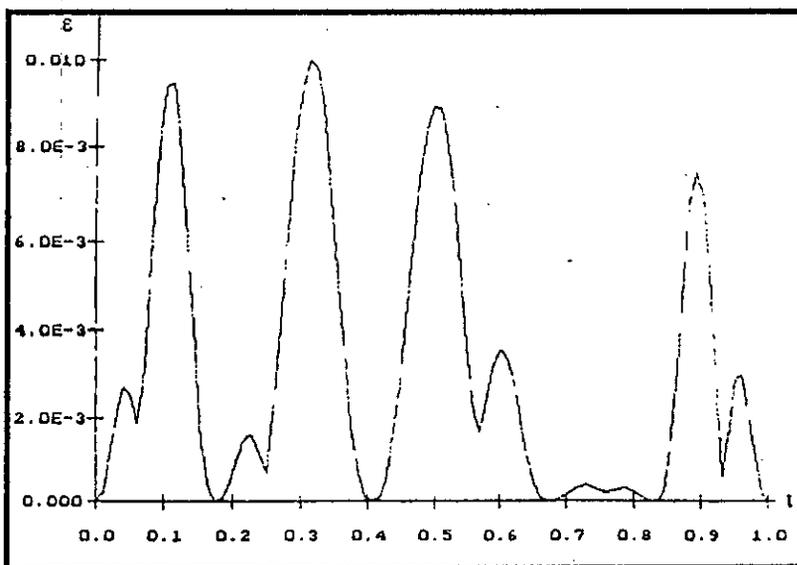


Figure 3.3.b. erreur de position le long de la courbe de degré 16

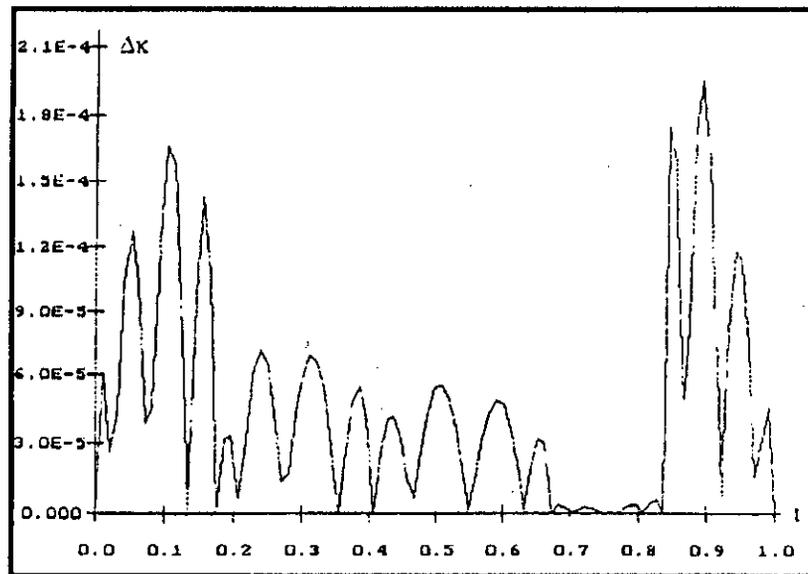


Figure 3.3.c. Variation de l'erreur sur la courbure

Interprétations des figures : la figure 3.1.a. représente une courbe de Bézier non rationnelle de degré 6 (n pair) réduite au degré 5, la figure 3.1.b. montre que l'erreur de position le long de la courbe réduite est continue, nulle aux extrémités, et possède un maximum à $t=0.5$ (au milieu). la continuité des courbures des deux courbes est assurée (figure 3.1.c), dans le cas où n est impair figure 3.2.a. la figure 3.2.b. montre que l'erreur de position est continue le long de la courbe réduite ainsi elle est symétrique au milieu de la courbe avec un minimum à $t=0$ la figure 3.2.c représente la variation de courbure des deux courbes nous constatons que la variation de courbure est nulle aux extrémités. La figure 3.3.a. représente une courbe de Bézier non rationnelle de degré 16 réduite en 5 courbe de degré 6 par l'algorithme CONV-FOR avec $\varepsilon_0 = 0.01$ et $k=10$. La Figure 3.3.b représente la variation de l'erreur de position le long de la courbe, montre que la tolérance fixée est respectée, et que les courbes obtenues sont continues au point commun, tandis que la figure 3.3.c. exprime la variation de courbure.

3.2.2 .Méthode de Bensalah

3.2.2.1. Algorithme de réduction

la réduction utilisée dans cette méthode n'est autre que la méthode présentée au paragraphe précédent qui donne deux solutions possibles $R(t)$ (polygone 1 (17.a)) et $S(t)$ (polygone 2 (17.b)) . Dans le cas général les deux courbes sont différentes , mais donnent chacune une bonne approximation à son début (figure 3.4) . La meilleure approximation est celle donnée par $Q(t)$ (polygone moyen (18)) qui est une moyenne pondérée des deux (figure 3.5) . Ces trois solutions reprenons sous la forme suivante [Farin '93].

Algorithme BEN1 :

Données : P_i ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 k ordre de réduction

Pour $r=1, 2, \dots, k$

$$R_0 = P_0$$

Pour $i=1, 2, \dots, n-1$

$$\beta_i = i / (n - i)$$

$$R_i = (1 + \beta_i) \cdot P_i - \beta_i \cdot R_{i-1}$$

$$P_i = R_i$$

Fin pour

$n=n-1$

Fin pour

Résultats : R_i ($i=0, 1, \dots, n-k$) polygone de contrôles de la courbe de degré $(n-k)$

Fin

Algorithme BEN2

Données : P_i ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 k ordre de réduction

Pour $r=1, 2, \dots, k$

$$S_{n-1} = P_n$$

Pour $i=n-1, n-2, \dots, 1$

$$\gamma_i = (n - i) / i$$

$$S_{i-1} = (1 + \gamma_i) \cdot P_i - \gamma_i \cdot S_i$$

$$P_i = S_i$$

Fin pour

$n=n-1$

Fin pour

Résultats : S_i ($i=0, 1, \dots, n-k$) polygone de contrôles de la courbe de degré $(n-k)$

Fin

Algorithme BENm

Données : P_i ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 k ordre de réduction

BEN1($P, k \rightarrow R$)

BEN2($P, k \rightarrow S$)

Pour $i=0, 1, \dots, n-k$

$$Q_i = \frac{n-k-i}{n-k} \cdot R_i + \frac{i}{n-k} \cdot S_i$$

Fin pour

Résultats : Q_i ($i=0, 1, \dots, n-k$) polygone de la courbe de degré $(n-k)$

Fin

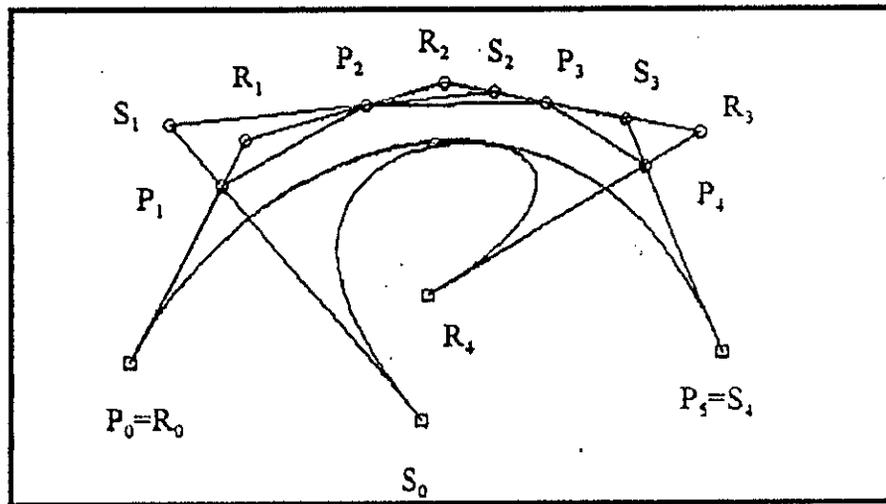


Figure 3.4. courbe de degré 5 réduite au degré 4 par BEN1 et BEN2

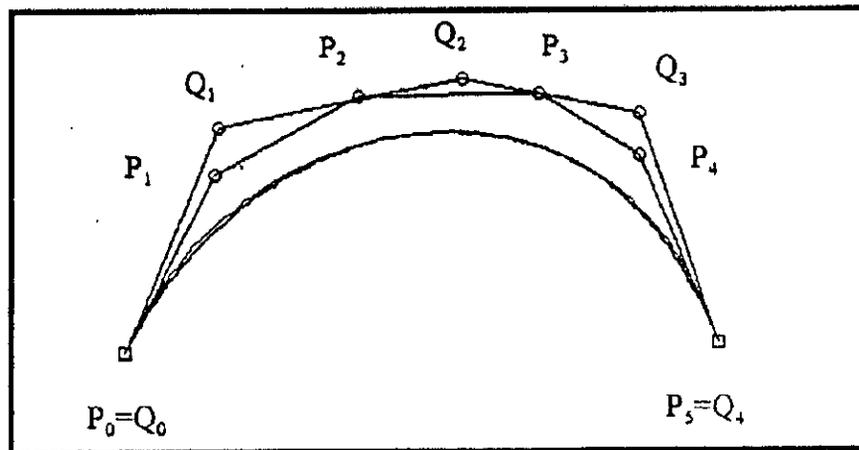


Figure 3.5. courbe de degré 5 précédente réduite au degré 4 par BENm

3.2.2.2. Algorithme de conversion par approximation

Les deux solutions de l'algorithme de réduction de degré des courbes de Bézier ne sont pas très efficaces pour toute la courbe, mais le sont quand même pour une partie, donc en partant d'une courbe $P(t)$ de degré n on peut faire une réduction du degré puis au point où on atteint la tolérance on subdivise en deux courbes la première de degré $(n-k)$ et l'autre de degré n puis on fait la même chose à la deuxième courbe et ainsi de suite, jusqu'à la fin de la courbe. Ce raisonnement est la base du fonctionnement de l'algorithme CONV-BEN. Nous remarquons que pour chaque itération on trouve deux subdivisions, une au début de la courbe et par la méthode du polygone 1, l'autre à la fin de la courbe et par le polygone 2, la méthode du polygone moyen n'est pas utilisée sachant qu'elle ne conserve pas les tangentes aux limites.

Alors que les deux autres la conservent mais seulement d'un côté d'où la partie d'ajustement des tangentes en fin de l'algorithme permettant d'ajuster les deux derniers pôles des polygones 1 et les deux premiers du polygone 2. [Asma'99]

Algorithme CONV-BEN

Données : P_i ($i=0, \dots, n$) polygone de contrôle de la courbe de degré n
 k : ordre de réduction
 ε_0 : tolérance de position
 N : nombre de pas.

```
(*)  BEN1 ( $P, k \rightarrow Q$ )
       $s=0$  ;  $d = |P_n - Q_{n-k}|$ 
      Si ( $d > \varepsilon_0$ ) alors
         $t = 1/N$ 
        tant que ( $|P(t) - Q(t)| < \varepsilon_0$ ) faire
           $t = t + (1/N)$ 
        Fin faire
         $t_0 = t - (1/N)$  ;  $s=s+1$ 
        SUB ( $Q, t_0 \rightarrow Q_0, Q_1$ ) ; SUB ( $P, t_0 \rightarrow P_0, P_1$ )
          Pour  $i=0, 1, \dots, n-k$ 
             $Q_i^s = Q_0$ 
          Fin pour
          Pour  $i=0, 1, \dots, n$ 
             $P_i = P_1$ 
          Fin pour
         $N=N-(1/t)$ 
        BEN2 ( $P, k \rightarrow Q$ )
         $s=0$  ;  $d = |P_0 - Q_0|$ 
        Si ( $d > \varepsilon_0$ ) alors
```

```

t= 1/N
Tant que ( $|P(1-t) - Q(1-t)| < \varepsilon_0$ ) faire
t=t+(1/N)
Fin faire
 $t_0 = 1-t + (1/N); s = s+1$ 
SUB ( $Q, t_0 \rightarrow Q0, Q1$ )
SUB ( $P, t_0 \rightarrow P0, P1$ )
  Pour i=0,1,...,n-k
     $Q_i^s = Q1_i$ 
  Fin pour
  Pour i=0,1,...,n
     $P_i = P0_i$ 
  Fin pour
N=N-(1/t)
  Sinon aller à (**)
Sinon aller à (**)
Aller à (*)
(**) s=s+1
  Pour i=0,1,...,n-k
     $Q_i^s = Q_i$ 
  Fin pour
Ajustements des tangentes des  $Q_i^s$ 
Résultats :  $Q_i^j$  ( $i=0,1,\dots,n-k$ ) ( $j=1,2,\dots,s$ )
           s : nombre de courbes de degré (n-k).
Fin

```

Pour la partie ajustement des tangentes l'auteur préconise l'algorithme BENm qui répartit l'erreur de tangence le long de la courbe.

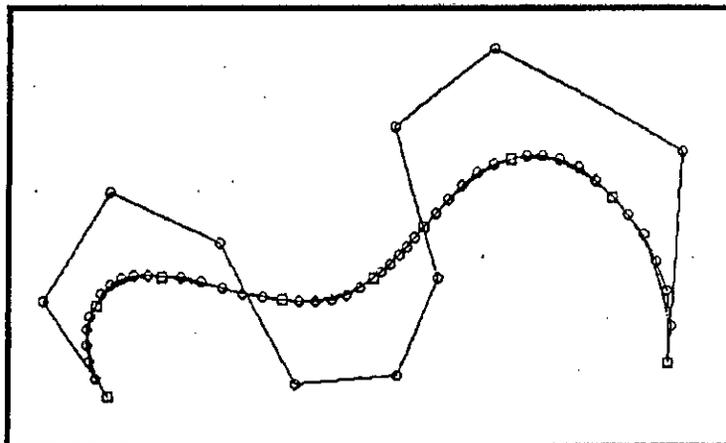


Figure 3.6.a. Courbe de degré 10 réduite en 8 courbes de degré 6 par l'algorithme (CONV-BEN), $\varepsilon_0 = 0.01$

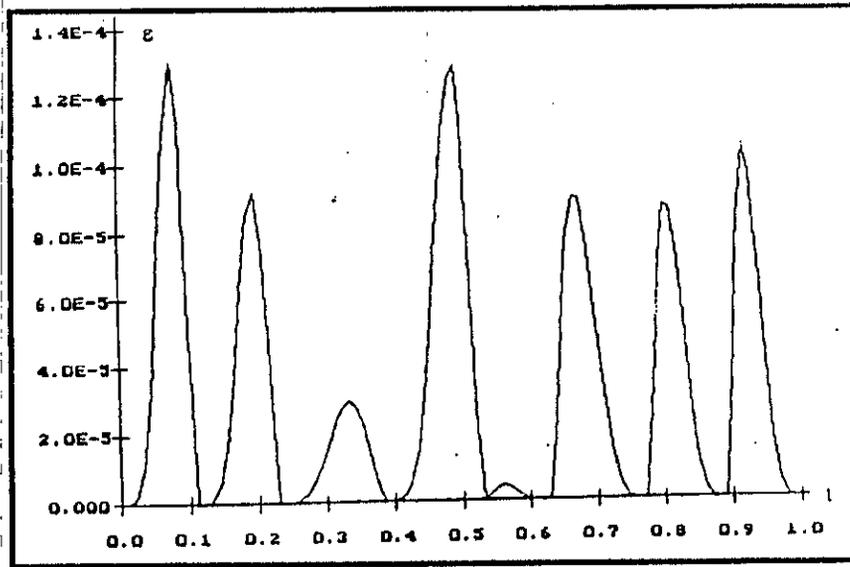


Figure 3.6.b. Erreur de position sur la courbe précédente

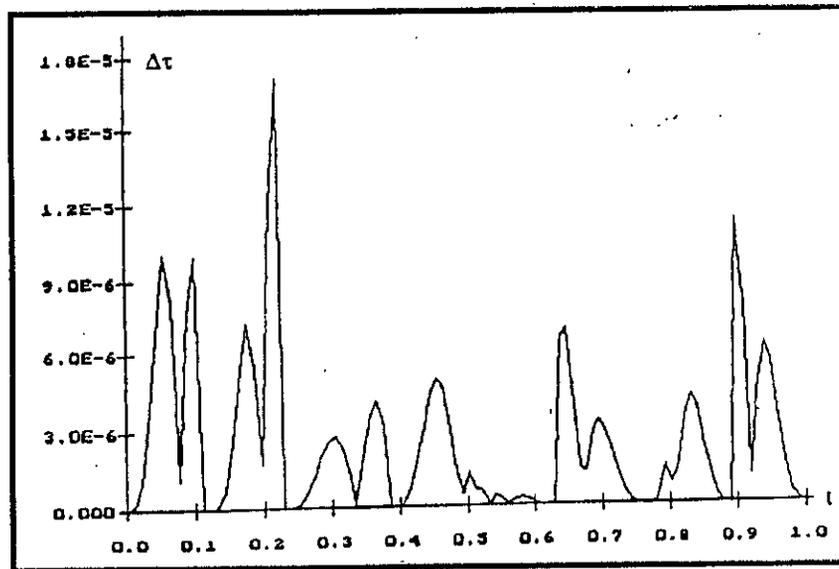


Figure 3.6.c. Variation de l'erreur sur la tangente

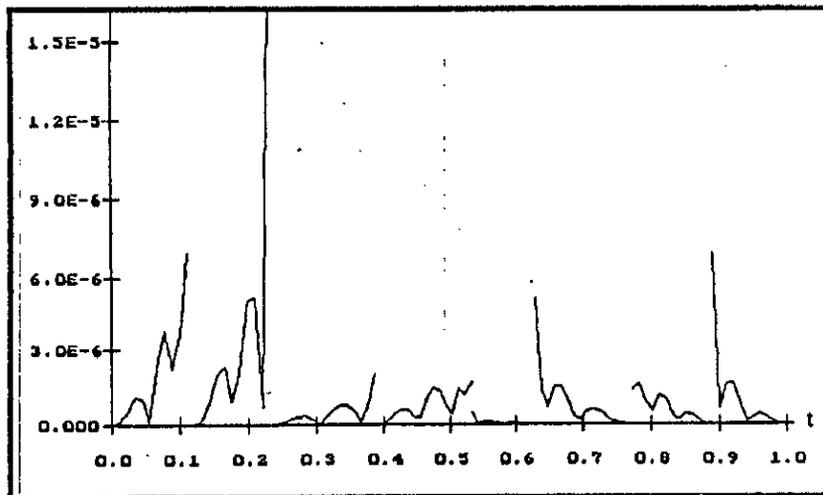


Figure 3.6.d. Variation de l'erreur sur la courbure après ajustement

Interprétations des figures : nous constatons que la continuité de position (figure 3.6.b), ainsi la continuité de tangentes est respectés garces à l'ajustement des tangentes (figure 3.6.c), mais la méthode de Bensalah ne respecte pas la continuité de la courbure la figure 3.6.d . montre la discontinuité de courbure.

3.2.3.Méthode de Hoschek

Introduction

la méthode introduite dans cette partie, est une méthode effective pour la conversion par approximation des courbes et des surfaces, cette méthode utilise principalement [hoscheck90] :

- Transformation du paramètre (paramétrisation).
- Techniques d'optimisation non linéaire.

De cet algorithme découle deux applications [Hoscheck&wissel'88] :

- Subdiviser une courbe de degré n donnée en une ou plusieurs courbes de degré $m < n$ (réduction de degré) en respectant une tolérance de position fixée ϵ_0 .
- Transformer autant que possible une courbe de degré N en une courbe de degré $M > N$ (élévation de degré) en respectant une tolérance de position fixée ϵ_0 .

L'idée essentielle de cet algorithme est d'utiliser la paramétrisation comme nouveau paramètre : la forme d'une courbe passant par un ensemble des points changera, si on change les valeurs du paramètre t durant le processus d'approximations.

3.2.3.1.Paramétrisation intrinsèque

le but de la paramétrisation est de trouver une bonne répartition des paramètres t_i , où (u_i, v_k) respectivement sur les points P_i, P_{ik} , permettant une approximation optimale d'une courbe ou d'une surface.

3.2.3.1.1. Approximation optimale des courbes.

Soit donné un ensemble $N+1$ de points P_i ($i=0$ à N), la Paramétrisation permet d'affecter un paramètre (coefficient) t_i à chaque point. L'approximation de ces points par une courbe de Bézier non rationnelle donne [Hoschek'88] :

$$Y(t) = \sum_{i=0}^n b_i B_i^n(t) \quad (t \in I, I \in [a, b]) \text{ et } n < N$$

Avec $B_i^n(t)$ fonction de base de Bernstein, et b_i les points de contrôlé de la courbe $Y(t)$.

L'approximation est optimale si la distance minimale entre les points P_i et la courbe approximée $Y(t)$ est minimisée, ce qui revient à la résolution du problème suivant :

$$\sum_{i=0}^N \min |Y(t) - P_i| \rightarrow \min \text{ avec } t \in I$$

Ce problème de minimisation non linéaire peut être réduit à un processus linéaire en introduisant l'erreur vectorielle :

$$D_i = P_i - Y(t_i) \tag{3.1}$$

ce qui revient à minimiser la somme :

$$D = \sum_{i=0}^N (P_i - Y(t_i))^2 \rightarrow \min \quad (3.2)$$

la méthode des moindres carrées permet d'écrire :

$$\frac{\partial D}{\partial b_i} = 0 \quad (3.3)$$

généralement cette erreur vectorielle n'est pas orthogonal à l'approximation $Y(t)$. la résolution est linéaire, mais la distance entre les points P_i et leur approximation n'est pas minimiser .figure.3.7.

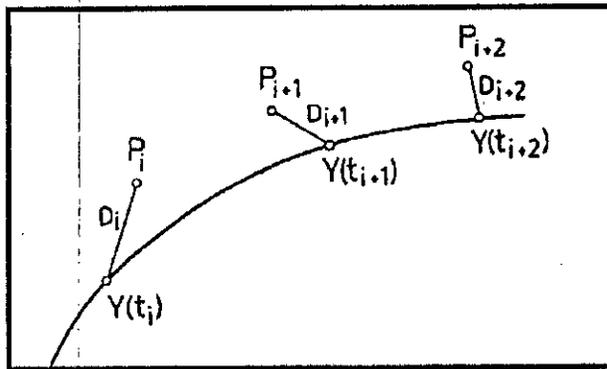


Figure.3.7. Paramétrisation des points donnés P_i
Et l'erreur vectorielle D_i .

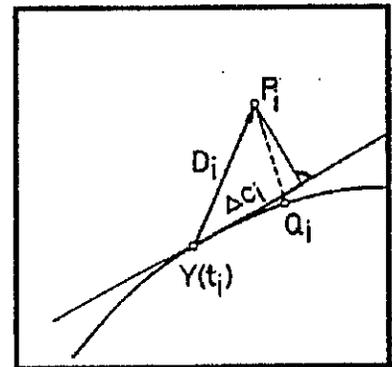


figure.3.8. Projection de
l'erreur vectorielle D_i

Pour une optimisation optimal il devient nécessaire de corriger la paramétrisation, en faisant une approximation en plus sur cette distance .

Pour cela il faut construire un nouveau paramètre \tilde{t}_i pour avoir une nouvelle erreur vectorielle \tilde{D}_i qui converge à la normale de la courbe approximé $Y(t)$, pour cela il faut remplacer la courbe en chaque point $Y(t_i)$ par la tangente (approximation linéaire de $Y(t)$ au voisinage de $Y(t_i)$), puis projeter l'erreur vectorielle D_i sur la tangente et obtenir Les valeurs Δc_i comme mesure du changement du paramètre t , qui converge vers la perpendiculaire de P_i sur $Y(t_i)$. (figure 3.8)

Δc_i doit être mesuré en respectant la longueur de la courbe et celle de l'intervalle $I \in [a, b]$ dans notre cas en prend $a=0, b=1$, et la longueur de la courbe μ est approximé par la longueur du polygone obtenu par les points P_i . Ce qui donne la correction du paramètre t , suivante :

$$\tilde{t}_i = t_i + \Delta c_i \times \frac{b-a}{\mu} \quad \text{avec} \quad \Delta c_i = D_i \times \frac{Y'(t_i)}{|Y'(t_i)|} \quad (3.4)$$

la correction (4) est admissible si $|Y(\tilde{t}_i) - P_i| \leq |Y(t_i) - P_i|$, sinon la correction sera dépassé la position du point Q_i (figure 2) de la perpendiculaire à $Y(t)$ passant par P_i , dans ce cas t_i sera seulement corrigé par $\frac{\Delta c_i}{2}$.

La méthode des moindres carrés (3.3) est de nouveau appliqué en utilisant les valeurs obtenues en (3.4). Cette étape est répétée jusqu'à ce que l'erreur vectorielle \tilde{D}_i soit orthogonale à la courbe $Y(t)$ en respectant une tolérance de position fixe.

Cette méthode converge localement avec des conditions similaires comme l'algorithme de Newton pour l'évaluation des zéros. La vitesse de convergence augmente si $Y(t)$ est approximé par des cercles oscillatoires au lieu des tangentes.

L'algorithme de la paramétrisation intrinsèque s'écrit [Hoschek'88] :

Algorithme HOS-PAR

Données $P_i, i=0, N$

Déterminer $Y(t)$ de degré $n < N$ par la méthode des moindres carrés en appliquant l'équation (3).

Intervalle du paramètre $t_i \in I, I \in [0, 1]$.

Étape 0 : choisir les valeurs du paramètre t_i , en prends un paramètre équidistant $t_i = i/N, i=0, N$.

Limite $L=4$, erreur imposé ε_0 .

$J=0$

Étape 1 : calculer μ ; (longueur du polygone P_i).

Étape 2 : déterminer $Y(t)$ la courbe approximé par la méthode des moindres carrés (en respectant (3)). (3.3)

Étape 3 : pour $i=0, 1, \dots, N$

Étape 4 : Calculer Δc_i ;

Étape 5 : Calculer $\tilde{t}_i = t_i + \Delta c_i \times \frac{b-a}{\mu}$

Calculer $|D_i| = |Y(t_i) - P_i|$;

Calculer $|\tilde{D}_i| = |Y(\tilde{t}_i) - P_i|$;

Si $D_i \geq \tilde{D}_i$ aller a l'étape 6.

Sinon $\Delta c_i = \frac{\Delta c_i}{2}$ aller à l'étape 5.

Étape 6 : Calculer $\alpha_i = \arccos \frac{D_i \times Y'(t_i)}{|D_i| \times |Y'(t_i)|}$;

Étape 7 : Calculer $\alpha = \max_{i=0, \dots, N}(\alpha_i)$;

$J=J+1$;

Si $\left| \frac{\pi}{2} - \alpha \right| < \varepsilon_0$ stop ;

Sinon

Si $(J < L)$ aller à l'étape 2 ;

Stop ;

3.2.3.1.2. Approximation optimal des surfaces

pour l'approximation optimal des surfaces définit par $(N+1) \times (M+1)$ points P_{ik} (réseau de contrôle), et (u_i, v_k) paramètres correspondant . la surface d'approximation prend sa représentation paramétrique :

$$Y(u, v) = \sum_{j=0}^n \sum_{k=0}^m B_j^n(u) B_k^m(v) b_{jk} \text{ avec } u \in [0,1], v \in [0,1], N > n, M > m$$

ou $B_j^n(u), B_k^m(v)$ représente les fonctions de base de Bernstein donné par l'équation (1) par la méthodes des moindres carrés ,avec l'introduction de l'erreur vectorielle

$$D_{ik} = P_{ik} - Y(u_i, v_k)$$

par la suite en minimise la somme :

$$D = \sum_{i=0}^N \sum_{k=0}^M (D_{ik})^2 = \sum_{i=0}^N \sum_{k=0}^M (P_{ik} - Y(u_i, v_k))^2 \tag{3.5}$$

en générale cette erreur vectorielle n'est pas orthogonal à la surface d'approximation ,donc (5) ne permet pas de minimiser la distance minimale entre les points P_{ik} est la surface approximé .

d'ou la nécessité de corriger la paramétrisation pour que l'erreur vectorielle converge vers la normale de la surface $Y(u, v)$,pour cela en projetant l'erreur vectorielle D_{ik} sur le plan tangent au point $Y(u_i, v_k)$,et obtenir $\Delta c_i, \Delta d_k$ comme des correction des paramètre (u_i, v_k) .(figure.3.9)

$\Delta c_i, \Delta d_k$ est mesuré en respectant la longueur complète de ligne de paramètre $u_i = const, v_k = const$. respectivement , et la longueur de l'intervalle correspondant.

Si μ est la longueur de la ligne de paramètre $u_i = const$.

Et ν est la longueur de la ligne de paramètre $v_k = const$.

Alors les correction sont donné par :

$$\bar{u}_i = u_i + \Delta c_i \times \frac{b-a}{\mu} \tag{3.6}$$

$$\bar{v}_k = v_k + \Delta d_k \times \frac{b-a}{\nu}$$

avec

$$\Delta c_i = D_{ik} \times \frac{Y_u(u_i, v_k)}{|Y_u(u_i, v_k)|} \tag{3.7.a}$$

$$\Delta d_k = D_{ik} \times \frac{Y_v(u_i, v_k)}{|Y_v(u_i, v_k)|} \tag{3.7.b}$$

ou $Y_u = \frac{\partial Y(u_i, v_k)}{\partial u_i}$

$$Y_v = \frac{\partial Y(u_i, v_k)}{\partial v_k}$$

avec les nouveaux paramètres obtenue par (3.6) ,la méthode des moindres carrés permet d'écrire :

$$\frac{\partial D}{\partial b_{jk}} = 0$$

ces étapes sont répété jusqu'à ce que l'erreur vectorielle soient approximativement orthogonale à la surface approximé.(en respectant une tolérance de position ϵ).

Cette méthode de paramétrisation des surfaces est l'extension de la paramétrisation Optimal des courbes présenté dans la section précédente .

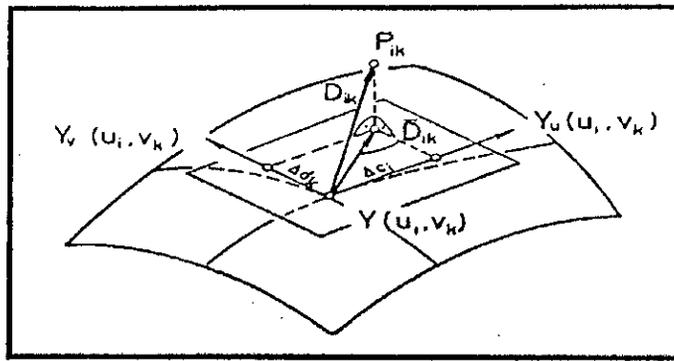


Figure 3.9. Projection de l'erreur vectorielle sur le plan tangent à la surface \$Y(u,v)\$ au point \$Y(u_i, v_k)\$.

3.2.3.1.3. Perspective de la méthode

La méthode d'approximation dite paramétrisation intrinsèque introduite dans ce chapitre, appliquées aux courbes et surfaces peuvent être appliquée pour l'approximation optimal des volumes paramétriques, définis par des fonctions de bases trivariable.

3.2.3.2. Réduction de degré

Si on utilise la paramétrisation comme but principal, on oubliera les conditions entre les courbes qui sont invariantes pour la paramétrisation. Ces conditions sont très connues sous le nom de continuité géométrique d'ordre \$k\$ de deux courbes.

Deux courbes \$P\$ et \$Q\$ ont une continuité géométrique d'ordre \$k\$ si les conditions suivantes sont vérifiées pour le point commun entre \$P\$ et \$Q\$. [Hoschek & Wissel '88], [Hoschek '88]

$$k = 1 \rightarrow P' = \lambda_1 \cdot Q'$$

$$k = 2 \rightarrow P'' = \lambda_1^2 \cdot Q'' + \lambda_2 \cdot Q' \text{ (avec la condition } k = 1)$$

$$k = 3 \rightarrow P''' = \lambda_1^3 \cdot Q''' + 3 \cdot \lambda_1 \cdot \lambda_2 \cdot Q'' + \lambda_3 \cdot Q' \text{ (avec les conditions } k = 1, k = 2) \quad (3.7)$$

$$k = 4 \rightarrow P'''' = \lambda_1^4 \cdot Q'''' + 6 \cdot \lambda_1^2 \cdot \lambda_2 \cdot Q''' + (3 \cdot \lambda_2^2 + 4 \cdot \lambda_1 \cdot \lambda_3) \cdot Q'' + \lambda_4 \cdot Q'$$

(avec les conditions \$k = 1, 2, 3\$)

pour un choix arbitraire des paramètres \$\lambda\$, ces conditions peuvent être développées en utilisant la géométrie différentielle introduite par W. Boehm (voir Annexe B).

Le modèle utilisé pour le développement de l'algorithme est le modèle de Bézier.

Alors la courbe \$P\$ de degré \$n\$, et la courbe \$Q\$ de degré \$m < n\$ sont données par (représentation paramétrique) :

$$P = P(t) = \sum_{i=0}^n P_i \cdot B_i^n(t) \quad t \in [0, 1] \quad (3.8)$$

$$Q = Q(\tau) = \sum_{i=0}^m Q_i \cdot B_i^m(\tau) \quad \tau \in [0, 1] \quad (3.9)$$

Les conditions de continuité dans le modèle de Bézier des courbes P et Q, sont obtenue en injectons les équations (3.8) et(3.9) dans (3.7), avec les conditions aux limites : $P(0)=Q(0)$ et $P(1)=Q(1)$ nous obtenons :

$k = 1:$

$$\begin{aligned} Q_0 &= P_0 \\ Q_1 &= P_0 + \lambda_1 \cdot (P_1 - P_0) \\ Q_m &= P_m \\ Q_{m-1} &= P_n + \mu_1 \cdot (P_n - P_{n-1}) \end{aligned} \tag{3.10.a}$$

$k = 2:$

$$\begin{aligned} Q_2 &= P_1 + \lambda_1^2 \cdot \omega_1 \cdot (P_2 - P_1) + \lambda_2 \cdot (P_1 - P_0) \\ Q_{m-2} &= P_{n-1} + \mu_1^2 \cdot \omega_1 \cdot (P_{n-2} - P_{n-1}) + \mu_2 \cdot (P_n - P_{n-1}) \\ \omega_1 &= \frac{m \cdot (n-1)}{n \cdot (m-1)} \end{aligned} \tag{3.10.b}$$

avec (59.a)

pour $k=3$ et $k=4$ (voir Annexe B)

3.2.3.2.1. L'erreur d'approximation due à la réduction de degré

l'approximation global ,d'une courbe de Bézier P de degré n, par une courbe de Bézier Q de degré $m < n$ est optimale si l'erreur vectorielle est minimiser.

L'erreur de position est mesuré à $(s+1)$ points P_j d'une courbe P de degré n ($s > n$, par

exemple $s=2 \cdot n$), en prend un paramètres curviligne équidistant $t_j = \frac{j}{n}, (j = 0, \dots, s)$

$$P_j = P(t_j) = P\left(\frac{j}{n}\right) \tag{3.11}$$

(ne pas confondre entre les points P_j de la courbe P de paramètre t_j , et le polygone de contrôle P_i de $n+1$ points de contrôles)

Si on insère les valeurs du paramètre t_j dans Q, alors l'erreur vectorielle devienne :

$$D_i = P_j - Q(t_j) \tag{3.12}$$

ce qui revient à minimiser la somme :

$$D = \sum_{j=0}^s D_j^2 \tag{3.13}$$

pour les différents conditions de continuité, l'erreur vectorielle est déterminé par (avec les conditions (10.a) et (10.b)) :

continuité d'ordre 1 : $k=1$ et $m=3$

$$D_j = R_j - (P_1 - P_0) \cdot \lambda_1 \cdot B_1^3(t_j) - (P_{n-1} - P_n) \cdot \mu_1 \cdot B_2^3(t_j)$$

avec

$$R_j = P_j - P_0 \cdot (B_0^3 + B_1^3)(t_j) - P_n \cdot (B_2^3 + B_3^3)(t_j)$$

par conséquent on obtient une somme d'erreurs linéaires, dans ce cas on utilise la méthode des moindres carrés afin de minimiser la somme totale D par les conditions :

$$\frac{\partial D}{\partial \lambda_1} = 0, \quad \frac{\partial D}{\partial \mu_1} = 0 \tag{3.14}$$

Ces conditions conduisent à un système linéaire en λ_1 et en μ_1 mais le résultat dépend de la paramétrisation des points P_j . Il faut donc résoudre le système linéaire obtenu à partir de (3.14) et insérer la solution avec (3.10.a) dans (3.9), on obtient alors la première approximation de la courbe Q(t).

Généralement l'erreur vectorielle donnée par l'équation (3.12) n'est pas orthogonale à l'approximation Q(t), mais seule la somme D donnée par (3.13) conduit à une approximation optimale. Donc, on doit changer la paramétrisation des points P_j pour avoir une meilleure approximation en utilisant la paramétrisation intrinsèque précédente. Sachant que la projection Δc_j est normé par μ (approximation de la longueur de la courbe P(t) par la longueur de son polygone formé par les points P_j).

On peut évaluer l'erreur de position maximale à :

$$\delta = \text{Max} |D_j| = \text{Max} |P(t_j) - Q(t_j)|$$

da la courbe P donnée et la courbe Q(t) obtenue pour quelques valeurs du paramètre t_i .

Les t_i sont choisis suivant :

1. Les valeurs du paramètre corrigé t_j des points $P_j = P(t_j)$.
2. En incrémentant les valeurs t_j entre t_j et t_{j+1} en respectant 1. le nombre de paramètre t_i dépend de la distance $P_j P_{j+1}$.

Cette approche respecte l'erreur D_j qui est l'approximation de la distance entre les points correspondants aux courbes P(t) et Q(t), quand la procédure de la paramétrisation donne l'approximation de l'erreur normale à la courbe Q(t) au point Q(t_j).

Si $\delta < \epsilon_0$ (ϵ_0 tolérance fixée), la procédure s'arrête.

Sinon on subdivise la courbe P(t) à $t = t_0$ (t_0 est le paramètre curviligne au point où δ est maximale). Et on répète la procédure pour les deux courbes obtenues P1(t) et P2(t).

Si $\delta < \epsilon_0$, la procédure s'arrête pour la courbe considérée.

Sinon la courbe sera subdivisée elle aussi de la même manière.

Continuité d'ordre 2 : k=2, m=5

$$D_j = R_j - (P_1 - P_0) \cdot (\lambda_1 \cdot B_1^5(t_j) + \lambda_2 \cdot B_2^5(t_j)) \\ - (P_2 - P_1) \cdot \lambda_1^2 \cdot \omega_1 \cdot B_2^5(t_j) \\ - (P_{n-1} - P_n) \cdot (\mu_1 \cdot B_4^5(t_j) + \mu_2 \cdot B_3^5(t_j)) \\ - (P_{n-2} - P_{n-1}) \cdot \mu_1^2 \cdot \omega_1 \cdot B_3^5(t_j)$$

avec

$$R_j = P_j - P_0 \cdot (B_0^5 + B_1^5 + B_2^5)(t_j) \\ - P_n \cdot (B_3^5 + B_4^5 + B_5^5)(t_j)$$

par conséquent la somme totale $D = \sum_{j=0}^s D_j^2$, devient une fonction non linéaire, pour cela on utilise des méthodes d'optimisation des systèmes non linéaire (qu'on va voir par la suite)

de la même façon comme le cas (k=1) :

On évaluons l'erreur de position maximale à :

$$\delta = \text{Max} |D_j| = \text{Max} |P(t_j) - Q(t_j)|$$

da la courbe P donnée et la courbe Q(t) obtenue pour quelques valeurs du paramètre t_j .

Les t_j sont choisis suivant :

1. Les valeurs du paramètre corrigé t_j des points $P_j = P(t_j)$.
2. En incrémentant les valeurs t_j entre t_j et t_{j+1} en respectant 1. le nombre de paramètre t_j dépend de la distance $P_j P_{j+1}$.

Cette approche respecte l'erreur D_j qui est l'approximation de la distance entre les points correspondants aux courbes P(t) et Q(t), quand la procédure de la paramétrisation donne l'approximation de l'erreur normale à la courbe Q(t) au point $Q(t_j)$.

Si $\delta < \varepsilon_0$ (ε_0 tolérance fixée), la procédure s'arrête.

Si non on subdivise la courbe P(t) à $t = t_0$ (t_0 est le paramètre curviligne au point où δ est maximale). Et on répète la procédure pour les deux courbes obtenues P1(t) et P2(t).

Si $\delta < \varepsilon_0$, la procédure s'arrête pour la courbe considérée.

Si non la courbe sera subdivisée elle aussi de la même manière.

Dans le cas k=3 et k=4 (voir Annexe B)

En générale k>1, la somme D devient une fonction non linéaire en λ_i et μ_i par conséquent on utilise des méthodes non linéaire.

La généralisation de cet algorithme pour une continuité d'ordre k=3, m=7 et k=4, m=9 peut se faire en respectant les 3 ou 4 conditions dans (3.7).

3.2.3.2.2. Algorithme de réduction

Algorithme HOS

Données : s+1 nombre de points P_j ($j=0, \dots, s$) de la courbe, avec $s=2.n$

P_i ($i=0, \dots, n$) polygone de contrôle de la courbe P(t)

L=4 ; tolérance imposé : ε_0 .

m : degré de la courbe approximé
 conditions de continuité : $m=2.k+1$

Pour $j=0,1,\dots,s$

$$t_j = \frac{j}{s};$$

$$P_j = P(t_j);$$

Fin pour

$r=1$

calculer μ ; (longueur du polygone P_j).

- (1) Calculer λ_i, μ_i ($i = 1,2,\dots,k$) par un algorithme de résolution non linéaire .
 Calculer les points de contrôles Q_i ($i = 0,1,\dots,m$)

Pour $j=0,1,\dots,N$

Calculer Δc_j ;

- (2) Calculer $\tilde{t}_j = t_j + \Delta c_j \times \frac{b-a}{\mu}$

Calculer $|D_j| = |Y(t_j) - P_j|$;

Calculer $|\tilde{D}_j| = |Y(\tilde{t}_j) - P_j|$;

Si $D_j \geq \tilde{D}_j$ alors aller a (3)

Sinon $\Delta c_j = \frac{\Delta c_j}{2}$ aller à (2).

$$\varepsilon_1 = \text{Max}(D_j)$$

- (1) Si $D_j \geq \varepsilon_1$ alors

$$t_j = \tilde{t}_j ;$$

Fin pour

$r=r+1$;

Si $\varepsilon_1 \geq \varepsilon_0$ alors

Si ($r < L$) alors aller à (1) ;

FinSi

FinSi

Résultats : les points de contrôles Q_i ($i = 0,1,\dots,m$)

ε_1 erreur d'approximation

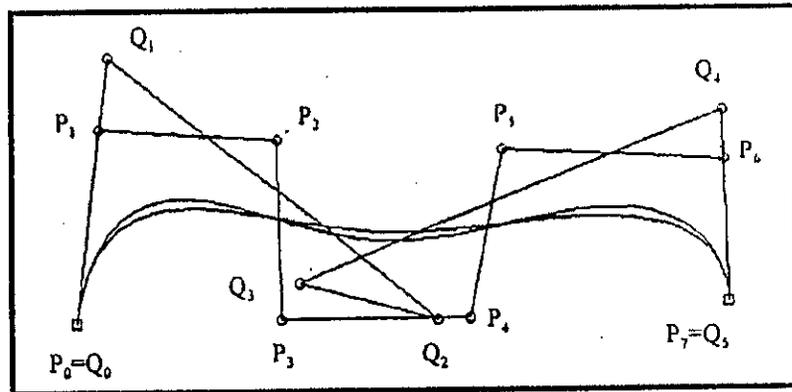


Figure 3.9.a. courbe de Bézier de degré 7 réduite au degré 5

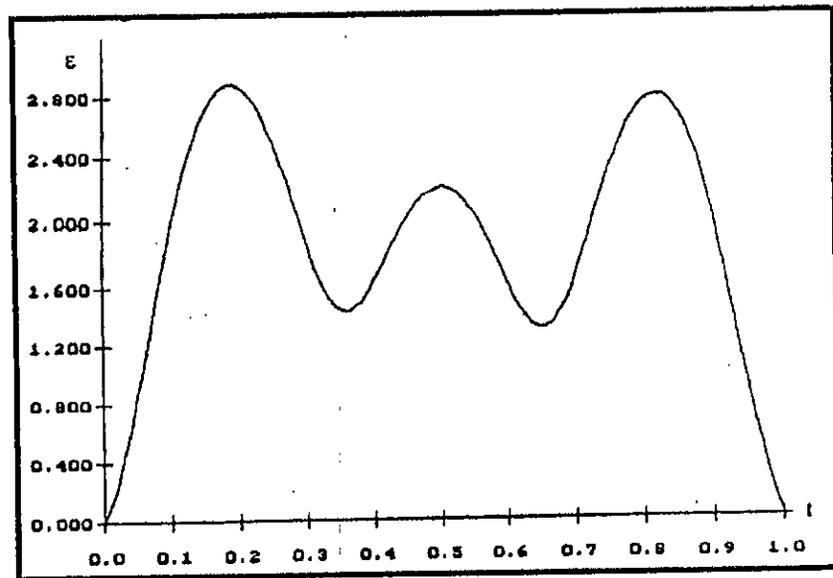


Figure 3.9.b. Variation de l'erreur d'approximation

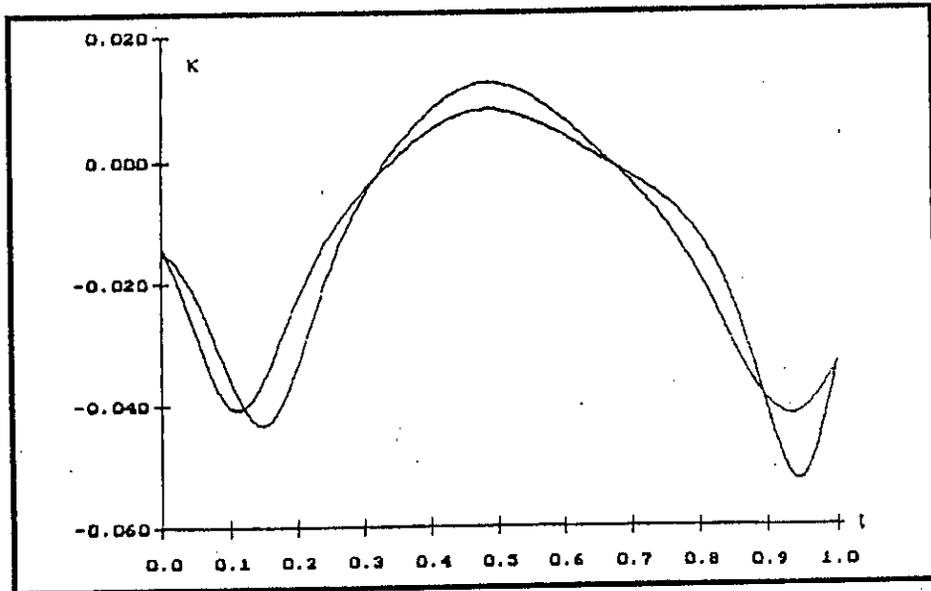


Figure 3.9.c. Courbure des deux courbes

3.2.3.2.3. Algorithme de conversion par approximation

l'algorithme de conversion (où subdivision/réduction) est conditionné par la tolérance de position ϵ_0 puisque les autres conditions de continuité sont respectées. la méthode consiste à réduire la courbe initiale puis vérifier l'erreur maximale entre celle ci et la courbe obtenue. Si

l'erreur est inférieur à la tolérance imposé ϵ_0 , la procédure s'arrête sinon la courbe sera subdiviser en deux courbes sur lesquelles le processus s'appliquera.

Algorithme CONV-HOS

Données : $s+1$ nombre de points P_j ($j=0, \dots, s$) de la courbe P, avec $s=2.n$

P_i^0 ($i=0, \dots, n$) polygone de contrôle de la courbe P(t)

k : ordre de continuité.

ϵ_0 : tolérance de position

$n1=0 ; n2=0 ;$

Pour $j=0, 1, \dots, n1$

HOS($s, P^j, k, \epsilon_0 \rightarrow Q, \epsilon_1$)

```

Si  $\varepsilon_i > \varepsilon_0$  alors
  SUB ( $P^j \rightarrow P_0, P_1$ )
  n1=n1+2
  Pour i=0,1,...,n
     $P_i^{n1-1} = P_{0_i}; P_i^{n1} = P_{1_i}$ 
  Fin pour
Sinon faire
  Pour j=0,1,...,n-k
     $Q_i^{n2} = Q_i$ 
  Fin pour
n2=n2+1
FinSi
Fin pour
Résultats :  $Q_i^j$  ( $i=0,1,\dots,n-k$ )( $j=0,2,\dots,n2-1$ )
n2 : nombre de courbes de degré ( $m=2.k+1$ )
    
```

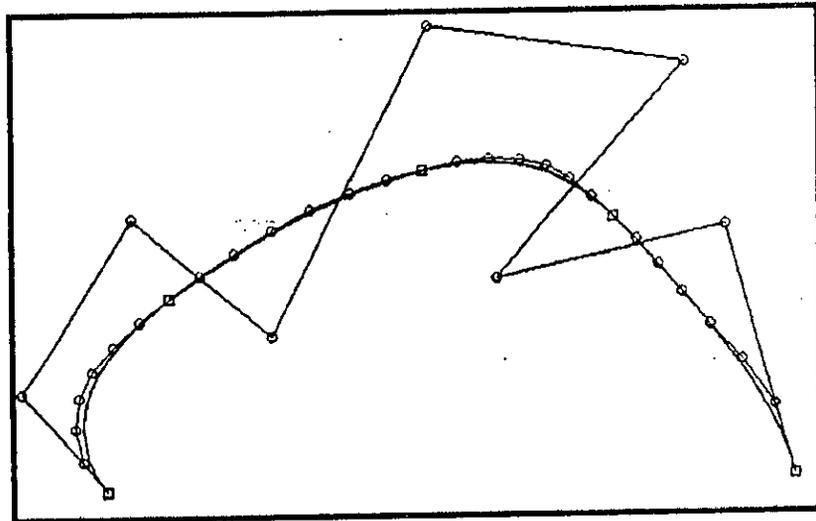


Figure 3.10.a. Courbe de Bézier de degré 8 réduite en 4 courbe de degré 7($k=3$) par l'algorithme CONV-HOS avec $\varepsilon_0 = 0.01$.

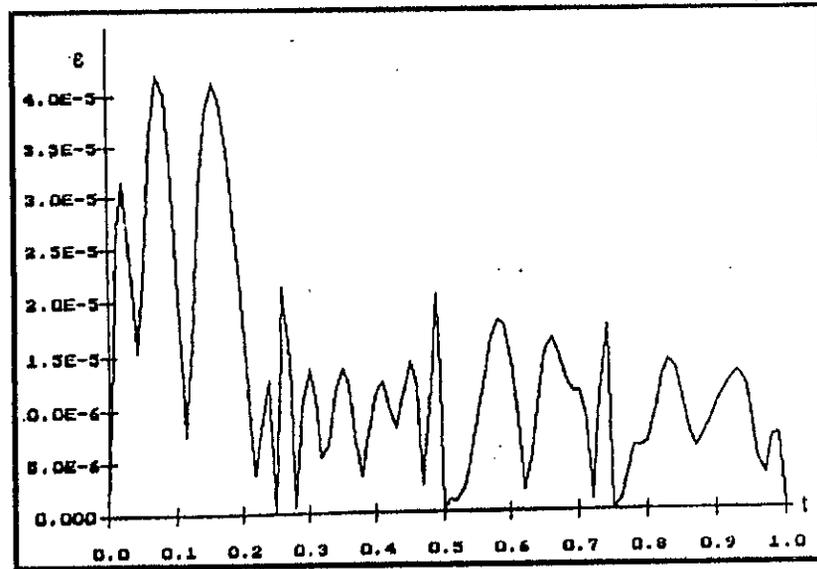


Figure 3.10.b. Variation de l'erreur de l'approximation pour la courbe convertie.

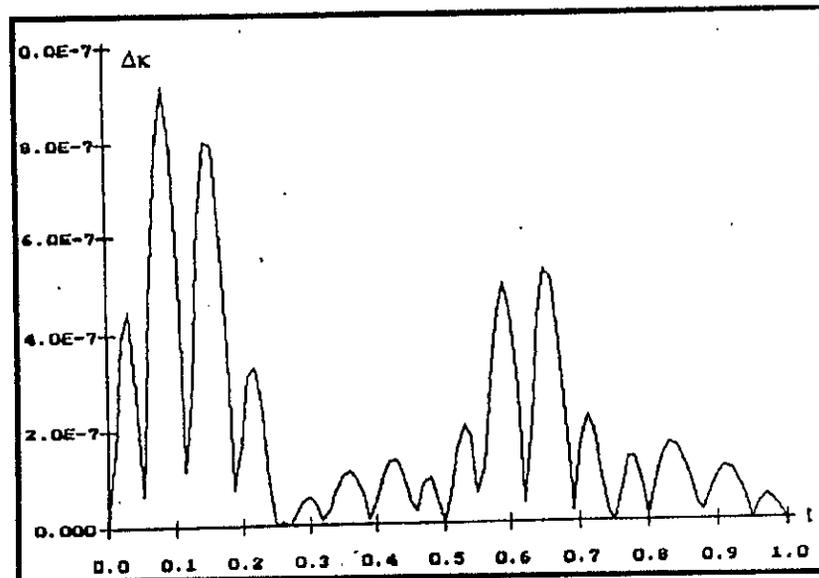


Figure 3.10.c. Variation de l'erreur sur courbure pour la courbe précédente.

Interprétations des figures : la méthode de Hoschek assure la continuité imposé, la figure 3.9.a représente une courbe de Bézier non rationnelle de degré 7 réduite au degré 5 par l'algorithme de HOS, la variation de l'erreur d'approximation montré par la figure 3.9.b. continue le long de la courbe , et nulle aux extrémités ainsi elle possède deux maximum . La variation des deux courbures et montré par la figure 3.9.c . qui montre la continuité De courbure de la courbe réduite , et la variation de courbures aux extrémités est nulle. La figure 3.10.a montre une courbe de Bézier non rationnelle de degré 8 réduite en 4 courbes de degré 7 par l'application de l'algorithme de CONV-HOS , les figures 3.10.b,c montrent le respect de continuité de tangente ainsi la courbure .

3.2.4.Méthode de Watkins et Worsley

la méthode présentée ici est basée sur les techniques d'approximation minimax, en utilisant comme résultat central dans le domaine d'approximation, le théorème de l'alternation, pour permettre le développement détaillé de l'analyse d'erreur. [Watkins & Worsley '88] en combinant l'algorithme de subdivision avec l'algorithme de réduction, l'erreur d'approximation commise lors de la réduction est réduite.

on montre que pour une courbe $P(t)$ de degré n , il est possible de déterminer a priori le nombre de subdivisions avant de faire l'approximation par un polynôme de degré moindre en respectant une tolérance donnée.

l'algorithme proposé par Watkins et Worsley pour la réduction de degré des courbes de Bézier non rationnelles se résume par les trois étapes suivantes :

Exprimer la courbe dans la base de Tchebycheff.

Tronquer le terme de plus haut degré.

Reconvertir la courbe dans la base de Bézier -Bernstein.

3.2.4.1 Polynômes de Tchebycheff

Définition : les polynômes de Tchebycheff de degré n , sont des polynômes orthogonaux définie par :

$$T_0(x) = 1 \quad T_1(x) = x$$

avec la relation de récurrence :

$$T_n(x) = 2x.T_{n-1}(x) - T_{n-2}(x)$$

ces polynômes satisfont à la relation remarquable suivante :

$$T_n(\cos \theta) = \cos(n.\theta) \quad (3.14)$$

avec un paramètre qui varie de $[-1,1]$.

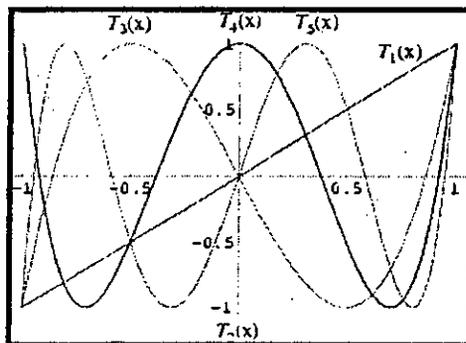


figure 3.11. polynômes de Tchebycheff de degré 5

alors les polynômes de Tchebycheff se sont des polynômes orthogonaux, donc pour tout polynômes de degré $\leq n$, peuvent s'écrire sous forme d'une combinaison linéaire :

$$P_n(x) = \sum_{i=0}^n \alpha_i . T_i(x) \quad (3.15)$$

La plus importante propriété dans notre contexte est la propriété de minimax, à partir de l'équation (3.14) tout polynôme $P(t)$ de degré $\leq n$ peut être écrit comme une combinaison linéaire de $T_i(x)$.

Si nous tronquons le terme $T_n(x)$ de cette somme, nous obtenons un polynôme de degré $\leq n - 1$:

$$P_{n-1}(t) = \sum_{i=0}^{n-1} \alpha_i \cdot T_i(t)$$

$P_{n-1}(t)$ à la propriété que tous les polynômes de degré $\leq n - 1$ le maximum de la différence absolue :

$$E_n = \text{MAX}(|P_n(t) - P_{n-1}(t)|)$$

est minimum sur l'intervalle $[-1, 1]$. de plus $E_n = |\alpha_n|$ est le maximum de la différence absolue du coefficient du polynôme de Tchebysheff

c'est à dire que si nous voulons remplacer un polynôme de degré n avec un autre polynôme de degré $n-1$, alors le moyen de minimiser l'erreur maximale est de convertir le polynôme dans la base de Tchebysheff et de tronquer le terme de plus haut degré des polynômes de Tchebysheff.

Théorème de l'alternation :

Si V un sous-espace vectoriel de dimension n de $C[a, b]$ vérifiant la condition de Haar,

Une condition nécessaire et suffisante pour que le polynôme $P_{n-1}(x) = \sum_{i=0}^{n-1} \alpha_i \cdot x^i$ soit la

meilleure approximation polynomiale, suivant la norme uniforme de $f \in C[a, b]$, si la fonction d'erreur $r(x) = f(x) - P_{n-1}(x)$ satisfait à la condition suivante [Watkins & Worsey'88][Pierre'72]

$$r(x_i) = -r(x_{i+1}) = \pm \|r\|, \quad i = 0, \dots, n-1$$

où

$$a \leq x_0 < \dots < x_n \leq b$$

et

$$\|r\| = \max_{x \in [a, b]} |r(x)|$$

Condition de Haar : on dira que le sous espace vectoriel V vérifie la condition de Haar si tout élément $g \in V$ admet au plus $n-1$ zéros dans S (S compact compose d'au moins n points).

3.2.4.2. Algorithme de réduction

Nous utilisons le théorème de l'alternation pour la constricton d'une meilleure approximation du polynôme de degré n par un polynôme de degré $(n-1)$ dans l'intervalle $[-1, 1]$.

Considérons le polynôme $P_n(x)$ de degré n , exprimer dans la base de Tchebysheff :

$$P_n(x) = \sum_{i=0}^n \alpha_i \cdot T_i(x)$$

(3.16.a)

où $T_i(x)$ est le polynôme de Tchebysheff de degré i donnée par :

$$T_i(y_k) = (-1)^k \quad (k=0, \dots, i) \quad \text{où } y_k = \cos(k \cdot \pi / i)$$

T_i étant de degré i . d'après l'équation (8) et Pour $x \in [-1, 1]$, nous pouvons écrire :

$$T_i(x) = \cos(i \cdot \arccos(x)) \Rightarrow \|T_i\| = 1, \quad \forall i$$

on considérons le polynôme de degré $n-1$ donnée par : $Q_{n-1}(t) = \sum_{i=0}^{n-1} \alpha_i \cdot T_i$

si on considère que le polynôme $Q_{n-1}(x)$ est la meilleure approximation de degré $n-1$ de $P_n(x)$ alors l'erreur $r(x)$ est donnée par :

$$r(x) = P_n(x) - Q_{n-1}(x) = \alpha_n \cdot T_n(x)$$

donc $\|r(x)\| = |\alpha_n|$

Alors $Q_{n-1}(x)$ est la meilleure approximation polynomiale de degré $n-1$ de $P_n(x)$ dans l'intervalle $[-1, 1]$ d'après le théorème de l'alternation.

Si l'intervalle considéré est un intervalle $[a, b]$ inclus dans $[-1, 1]$, alors les polynômes transformés de Tchebysheff sont :

$$T_i^*(x) = T_i\left(\frac{2x - a - b}{b - a}\right); \quad x \in [a, b]$$

si l'intervalle est $[0, 1]$, comme le cas des courbes de Bézier, les polynômes transformés de Tchebysheff s'écriront sous la forme suivante :

$$T_i^*(x) = T_i(2x - 1); \quad x \in [0, 1]$$

dans ce qui va suivre nous utilisons les polynômes transformés de Tchebysheff.

Il est important de noter qu'en général, $Q_{n-1}(x)$ ne respecte pas la continuité aux extrémités par rapport à $P_n(x)$. Puisque la courbe sera subdivisée, on peut forcer donc les points extrêmes pour maintenir une continuité d'ordre 0 aux extrémités. Ce qui est particulièrement simple dans le cas des courbes de Bézier, il suffit simplement de changer le premier et le dernier pôles (points de contrôle). Après cette modification la courbe obtenue n'est pas la meilleure approximation dans la norme uniforme (appelé aussi norme de Tchebysheff).

Algorithme WW

Données : $P_i (i=0, 1, \dots, n)$ points de contrôle de la courbe de degré n

k : ordre de la réduction

Pour $i=0, 1, \dots, n$

$$V_p(i+1) = P_i$$

Fin pour

Pour $i=1, 2, \dots, n$

$$V_a(i) = 0$$

pour $j=1, 2, \dots, n$

$$V_a(i) = V_a(i) + (T_{BM} \cdot T_{MT})_{ij} \cdot V_p(j)$$

Fin pour

Fin pour

$m=n-k$

Pour $i=1, 2, \dots, m+1$

$$V_p(i) = 0$$

Pour $j=1,2,\dots,m+1$

$$V_p(i) = V_p(j) + (T_{MT}^{-1} \cdot T_{BM}^{-1})_{ij} \cdot V_p(j)$$

Fin pour

Fin pour

Pour $i=1,2,\dots,m-1$

$$Q_i = V_p(i+1)$$

Fin pour

$$Q_0 = P_0$$

$$Q_m = P_n$$

résultats : Q_j ($j=0,1,\dots,n$) courbe réduite de degré $m=n-k$

Fin

Pour réduire une courbe $P(x)$ de degré n en une courbe $Q(x)$ de degré $m=n-k$, il suffit d'appliquer k fois l'algorithme WW.

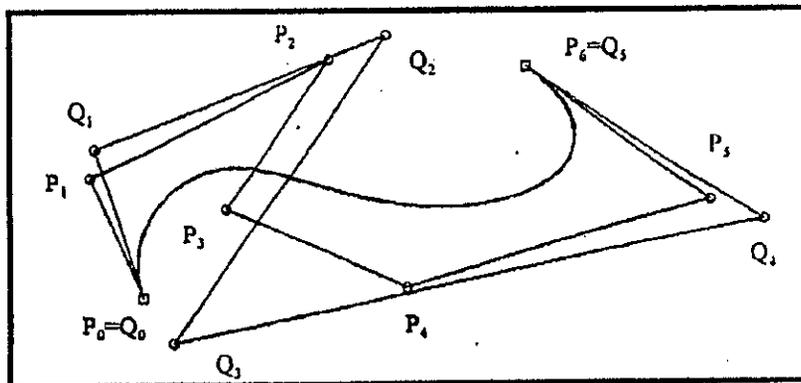


Figure 3.12.a. Courbe de degré 6 réduite au degré 5 par la méthode de Watkins et Worsey

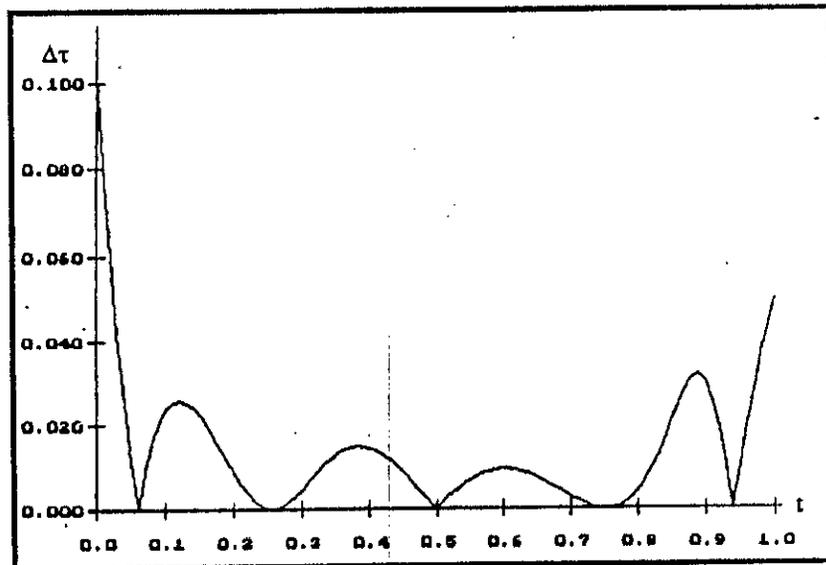


Figure 3.13.c. Variation de l'erreur sur la tangente

3.2.4.3. Algorithme de conversion par approximation

dans cette section, on se propose de déterminer l'erreur commise dans le processus de réduction ainsi que pour l'interpolation forcée. On décrira alors un nouveau algorithme de subdivision/réduction (conversion), résultant de cette méthode. Des résultats ont été présentés montrant comment et quand doit-on subdiviser ou essayer de réduire. En particulier, on montre que pour une courbe $P(x)$ de degré n , il est possible de déterminer a priori combien de fois doit-on subdiviser $P(x)$ avant de faire l'approximation par un polynôme de degré moindre en respectant une tolérance donnée. Donc, il n'est pas nécessaire d'essayer de réduire ou de contrôler la linéarité après chaque subdivision.

On suppose que l'erreur commise pour la réduction d'une courbe $P(x)$ est ε . Alors, puisque l'erreur est donnée par $\alpha_n \cdot T_n(x)$ et les valeurs extrémales de $T_n(x)$ sont ± 1 à $t=0$ et $t=1$, l'erreur maximale (en valeur absolue) est certainement atteinte aux extrémités. Si la courbe réduite est $Q(x)$ et la courbe réduite avec interpolation forcée $\hat{Q}(x)$, alors on peut calculer $\|P - \hat{Q}\|$ par l'inégalité suivante :

$$\|P - \hat{Q}\| \leq \|P - Q\| + \|Q - \hat{Q}\| \quad (3.16)$$

la fonction $Q - \hat{Q}$ peut être écrite sous forme d'une courbe de Bézier dont tous les coefficients sont nuls excepté le premier et le dernier, qui prennent la valeur $\pm \varepsilon$. Donc d'après les propriétés de l'enveloppe convexe des courbes de Bézier :

$$\|Q - \hat{Q}\| \leq \varepsilon \quad (3.17)$$

les inégalités (3.16) et (3.17) impliquent que si l'erreur due à la réduction seule est ε . Alors l'erreur due à la réduction avec interpolation forcée est inférieure ou égale à $2 \cdot \varepsilon$.

Ce qui montre exactement comment l'erreur de réduction diminue avec les subdivisions au milieu de l'intervalle en question. Ce qui nous permet de réduire seulement quand c'est nécessaire, plutôt qu'après chaque subdivision.

Soit $P(x)$ donnée par les polynômes transformés de Tchebycheff par :

$$P_n(x) = \sum_{i=0}^n \alpha_i \cdot T_i^*(x)$$

pour $x \in [a, b]$, pour $x \in [a, (a+b)/2]$ $P(x)$ est donnée par :

$$P_n(x) = \sum_{i=0}^n \beta_i \cdot \hat{T}_i(x)$$

ou $\hat{T}_i(x) = T_i^*(2x + b)$ les polynômes transformés de Tchebycheff définies dans l'intervalle $[a, (a+b)/2]$, donc

$$P_n(x) = \sum_{i=0}^n \alpha_i \cdot T_i^*(x) = \sum_{i=0}^n \beta_i \cdot T_i^*(2x + b)$$

noter bien que $|\alpha_n|$ est l'erreur de la réduction de degré dans l'intervalle $[a, b]$, et $|\beta_n|$ l'erreur de réduction dans $[a, (a+b)/2]$.

Nous désirons d'avoir une relation entre les coefficients α_n et β_n , alors les polynômes T_i^* de degré i , exprimés dans la base monomiale :

$$P_n(x) = \sum_{i=0}^n \alpha_i \cdot \sum_{k=0}^i \gamma_{ik} \cdot x^k = \sum_{i=0}^n \beta_i \cdot \sum_{k=0}^i \gamma_{ik} \cdot (2x - a)^k$$

égalisons les coefficients de terme x^n .

$$\alpha_n \cdot \gamma_{nn} \cdot x^n = \beta_n \cdot \gamma_{nn} \cdot 2^n \cdot x^n$$

d'où

$$\beta_n = \frac{1}{2^n} \cdot \alpha_n$$

le résultat est similaire dans l'intervalle $[(a+b)/2, b]$.

D'où l'erreur due à la réduction diminue par multiplication au facteur $\frac{1}{2^n}$ à chaque

subdivision, ce qui donne après s subdivision une erreur égale à $\left(\frac{1}{2^{n \cdot s}}\right) \cdot \alpha_n$. Donc pour

obtenir une approximation de degré $(n-1)$ à une courbe de degré n en respectant une tolérance ε Donnée, le nombre de subdivision s nécessaire est :

$$s > \frac{1}{n} \cdot \log_2 \left(\frac{\alpha_n}{\varepsilon} \right)$$

ces résultats conduisent au schéma de conversion par approximation

(subdivision/réduction) suivant, pour l'obtention des courbes de degré d , approximation d'une courbe de degré n en respectant une tolérance ε : [Watkins & Worsey'88][Asma '99]

Algorithme CONV-WWDonnées : $P_i (i=0,1,\dots,n)$ points de contrôles de la courbe de degré n k : ordre de la réduction ε_0 tolérance de position $m=n$ $a=1, b=1, c=1$ Pour $i=0,1,\dots,n$ $R_i^1 = P_i$

Fin Pour

Pour $r=1,2,\dots,k$ Pour $f=a, a+1, \dots, b$ Pour $i=0,1,\dots,m$ $S_i = R_i^f$

Fin pour

 Calculer α_m

$$ns = \frac{1}{m} \log_2 \left(\frac{k \cdot \alpha_m}{2 \cdot \varepsilon_0} \right)$$

 pour $d=1,2,\dots,ns$ $t_0 = 1/(ns-d+2)$ SUB($S, t_0 \rightarrow Q0, Q1$) Pour $j=0,1,\dots,m$ $S_j = Q1_j$

Fin pour

 $C=c+1$; WW($Q0, 1 \rightarrow Q$) Pour $j=0,1,\dots,m-1$ $R_j^c = Q_j$

Fin pour

Fin pour

 $a=a+1$; $b=c$; $m=m-1$;

Fin pour

 $f=0$ pour $i=a, a+1, \dots, b$ $f=f+1$; pour $j=0,1,\dots,m$ $Q_j^f = R_j^i$

Fin pour

Fin pour

 $nb=b-a$ résultats : $Q_j^i (j=0,1,\dots,m)$ et $(i=1,2,\dots,nb)$ nb : nombre de courbe de degré $m=n-k$

Fin

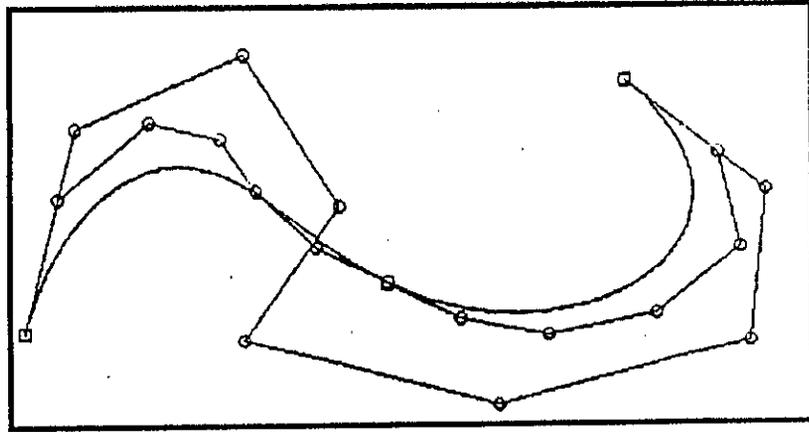


Figure 3.13.a. Courbe de degré 8 réduite en deux courbes de degré 6 par l'algorithme CONV-WW

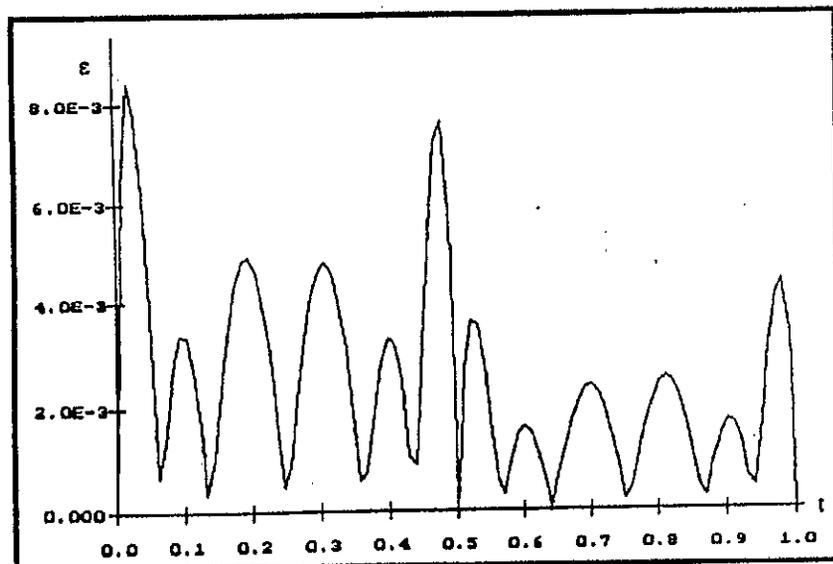


Figure 3.13.b. Variation de l'erreur de position

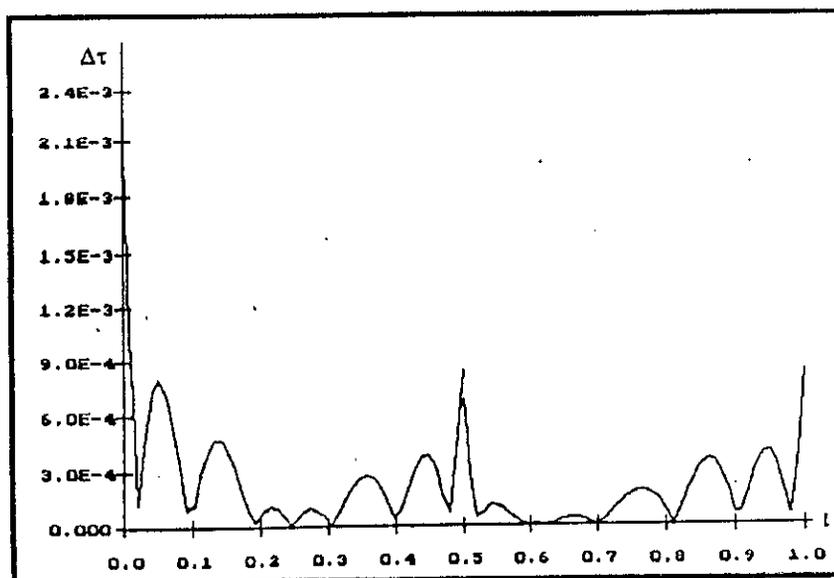


Figure 3.13.c. Variation de l'erreur sur la tangente

Interprétations des figures : la figure la figure 3.12.a.représente une courbe de Bézier non rationnelle de degré 6 réduite en courbe Q(t) de degré 5 et mais en évidence la continuité de position aux extrémités , par contre la figure 3.12.b. montre la discontinuité de la tangente aux extrémités.

la figure 3.13. a. représente une courbe de Bézier non rationnelle P(t) de degré 8 réduite par l'algorithme CONV-WW en deux courbes de degré 6 avec $\epsilon_0 = 0.01$,l'erreur de position est nulle aux extrémités (figure3.13.b),par contre la figure 3.13.c montre que la continuité aux extrémités n'est pas respecté, l'approximation obtenue à un comportement symétrique figure 3.13.c.

3.2.4.4. Conversion de base

Puisque la conversion de bases polynomiales est une transformation linéaire , la conversion de base entre les formes de Bernstein et Tchebycheff peut se présenter sous la forme d'un produit matriciel .

Pour convertir un vecteur V_p à coefficients de Bézier en un vecteur V_a à coefficients de la base de Bézier à la base de Tchebycheff , on utilise l'équation suivante : $V_a = V_p \cdot T_{BM} \cdot T_{MT}$.

Pour la conversion inverse on écrit : $V_p = V_a \cdot T_{MT}^{-1} \cdot T_{BM}^{-1}$

Où T_{BM} est la matrice qui converti les coefficients de la base de Bézier à la base monomiale , T_{MT} est la matrice qui convertit les coefficient de la base monomiale à la base de Tchebycheff , T_{BM}^{-1} , T_{MT}^{-1} sont leurs matrices inverses respectives .

Les matrices T_{BM} et T_{BM}^{-1} sont connues explicitement, la matrice T_{MT}^{-1} peut être calculée par une relation réursive pour les polynômes transformés de Tchebycheff.

T_{MT} est aussi déterminé explicitement. Heureusement, les matrices impliquant les polynômes de Tchebycheff ne dépendent pas du degré de la courbe, et par conséquent on n'a besoin de les calculer qu'une seule fois. [Watkins & Worsey'88]

les matrices impliquant les bases dépendent du degré de la courbe.

on doit alors les calculer séparément pour chaque degré. En pratique, il est préférable de calculer ces matrices pour quelques degrés prédéterminés et calculer les produits

$T_{BM} \cdot T_{MT}$ et

$T_{MT}^{-1} \cdot T_{BM}^{-1}$ une seule fois pour chaque degré et enregistrer les résultats dans un fichier. Donc cet algorithme de réduction nécessite seulement deux produits matriciels et ne nécessite pas d'évaluation de courbes. On donne les matrices dans le cas général d'une courbe de degré n ;

toutes les matrices sont données pour une multiplication à gauche des vecteurs.

T_{BM} :

$$(T_{BM})_{ij} = (-1)^{i+j} \cdot C_{i-1}^n \cdot C_{n+1-j}^{n+1-i}; \text{ pour } \begin{cases} i = 1, 2, \dots, n+1 \\ j = i, i+1, \dots, n+1 \end{cases}$$

avec $C_i^n = \frac{n!}{i!(n-i)!}$ est le coefficient binomial.

toutes les autres composantes sont nulles.

T_{BM}^{-1} :

$$(T_{BM}^{-1})_{ij} = C_{j-i}^{j-1} / C_{i-1}^n; \text{ pour } \begin{cases} i = 1, 2, \dots, n+1 \\ j = i, i+1, \dots, n+1 \end{cases}$$

toutes les autres composantes sont nulles.

T_{MT} :

$$(T_{MT})_{11} = 1, \quad (T_{MT})_{21} = \frac{1}{2}, \quad (T_{MT})_{22} = \frac{1}{2}$$

$$(T_{MT})_{ij} = C_{i-3}^{2i-3} / 2^{2i-3} \text{ pour } \begin{cases} i \geq 3 \\ j = 1 \end{cases}$$

$$(T_{MT})_{ij} = C_{i-j}^{2i-3} / 2^{2i-3} \text{ pour } \begin{cases} i \geq 3 \\ j \geq 2 \end{cases}$$

Toutes les autres composantes sont nulles.

T_{MT}^{-1} :

$$(T_{MT}^{-1})_{11} = 1, \quad (T_{MT}^{-1})_{21} = -1, \quad (T_{MT}^{-1})_{22} = 2$$

$$(T_{MT}^{-1})_{ij} = 4 \cdot (T_{MT}^{-1})_{i-1, j-1} - 2 \cdot (T_{MT}^{-1})_{i-1, j} - (T_{MT}^{-1})_{i-2, j} \text{ pour } j \leq i$$

$$(T_{MT}^{-1})_{ij} = 0 \text{ pour } j > i$$

Toutes les autres composantes sont nulles.

Quelques produits matriciels $T_{BM} \cdot T_{MT}$ et $T_{MT}^{-1} \cdot T_{BM}^{-1}$ pour quelques valeurs de n sont donnés en (Annexe D) C

Remarques : il est intéressant de considérer les effets de cette conversion de base sur la précision du processus de réduction de degré, particulièrement les résultats de Farouki et Rajan d'une part ont démontré que les bases de Bernstein conviennent la mieux numériquement que les base de puissance. C'est à dire qu'elles augmentent la stabilité de l'algorithme, puisque le conditionnement de ces bases est meilleure. D'autre part Farouki A aussi montré que les bases de Tchebycheff sont numériquement mieux conditionnées que les bases de Bernstein. Donc si on améliore la conversion de bases entre les bases de Bernstein et Tchebycheff, du point vue de précision et stabilité, alors aucune perte de précision n'affectera l'algorithme de réduction.

Les éléments des matrices de conversion peuvent être déterminés avec exactitude analytiquement, et seront convertis à virgule flottante par une simple division. Par conséquent, le nombre d'opérations à virgule flottantes est modestement réduit.

3.2.5. Méthode de Eck

Forrest, à proposé de combiner deux formules d'extrapolation (28.a)(28.b) dérivons de l'équation d'élévation de degré (22), en prenons la moitié gauche des pôles à partir de R (polygone1) qui produit une bonne approximation au voisinage de P_0 , et la moitié droite à partir de S (polygone2), qui produit une bonne approximation au voisinage de P_n .

Pour construire une meilleure approximation, on étend la construction de Forrest par l'introduction en plus, des degrés de liberté appelés facteurs de pondération $\lambda_{i,n}$ qui rassemblent les coefficients R_i et S_i entre eux. Pour plus de détail on utilise la formule d'interpolation linéaire suivante :

$$Q_i = (1 - \lambda_{i,n}) \cdot R_i + \lambda_{i,n} \cdot S_i \quad (i = 0, \dots, n-1) \quad (3.18)$$

pour définir les nouveaux pôles Q_i ($i = 0, \dots, n-1$)

les facteurs $\{\lambda_{i,n}\}_{i=0}^{n-1}$ sont déterminés par :

$$\lambda_{i,n} = 2^{1-2n} \cdot \sum_{j=0}^i \binom{2n}{2j} \quad (i = 0, \dots, n-1) \quad (3.19)$$

de plus l'erreur d'approximation est donnée par :

$$d(P, Q) = 2^{1-2n} \cdot |\Delta^n P_0| \quad (3.20)$$

remarques :

- Si $\Delta^n P_0 = 0$ alors les facteurs $\lambda_{i,n}$ peuvent être choisis arbitrairement puisque $R_i = S_i$ pour tout i .
- On a $0 < \lambda_{0,n} < \lambda_{1,n} < \dots < \lambda_{n-1,n} < 1$.

Un autre avantage de la formule (3.19) est que les facteurs $\lambda_{i,n}$ ne dépendent pas des pôles P_i .

Donc ce schéma peut être interprété dans le cas de la réduction de degré des courbes de Bézier

Puisque seules les interpolations linéaires et itérations d'extrapolation sont utilisées.

On peut noter que P(t) et Q(t) ne coïncident pas aux extrémités puisque $\lambda_{0,n} \neq 0$ et

$$\lambda_{n-1,n} \neq 0$$

Pour des applications pratiques en CFAO ce comportement est indésirable. On va voir dans le paragraphe suivant comment modifier le schéma.

3.2.5.1. Réduction de degré et continuité géométrique

Dans l'algorithme de Watkins & Worsey la continuité C^0 aux extrémités est obtenue en changeant le premier et le dernier pôle de Q(t) pour que l'interpolation soit valable. Dans notre cas cette correction est simplement obtenue en définissant $\lambda_{0,n} = 0$ et $\lambda_{n-1,n} = 0$

Et les autres facteurs sont toujours déterminés par l'équation (3.19).

Il est évident que l'inconvénient est que la propriété de meilleure approximation est perdue.

Ce problème peut être maîtrisé grâce aux polynômes de Tchebycheff contraints $T_{m,\alpha}(t)$.

Ce qui permet de déterminer la meilleure approximation forcée pour une continuité C^{u-1} aux deux extrémités.

Ces polynômes uniques $T_{m,\alpha}(t)$ de degré m ayant α solution à $t=-1$ respectivement à $t=1$, et sont d'égale oscillation ayant $m-2, \alpha-1$ valeurs extrémales ± 1 dans l'intervalle $[-1,1]$ qui est une caractéristique de la propriété de la meilleure approximation forcée. Néanmoins, les formules explicites de $T_{m,\alpha}(t)$ pour $\alpha \geq 0$ ne sont pas connues à l'exception de :

$$T_{m,0}(t) = T_m(t) \text{ et } T_{m,1}(t) = T_m(t \cdot \cos \frac{\pi}{2 \cdot m}) \quad (3.21)$$

Donc, ces polynômes doivent être déterminés numériquement par l'intermédiaire de l'algorithme « type Remes » [Eck'93], qui est une méthode standard du calcul de la meilleure approximation. Sans entrer dans les détails, on suppose pour la suite que l'expression en terme de Bernstein est :

$$T_{m,\alpha}(t) = \sum c_{i,m,\alpha} \cdot B_i^m \left(\frac{t+1}{2} \right), \quad t \in [-1,1] \quad (3.22)$$

En calculons numériquement les coefficients $c_{i,m,\alpha}$. Ces coefficients non nuls sont alternés, c'est à dire :

$$c_{i,m,\alpha} = c_{m-i,m,\alpha} = 0, \quad (i = 0,1,\dots,\alpha-1) \quad (3.23)$$

Donc, puisque $T_{m,\alpha}$ est d'égale oscillation, les signes des coefficients non nuls sont alternés, c'est à dire :

$$c_{i,m,\alpha} = (-1)^{m+i} \cdot |c_{i,m,\alpha}|, \quad (i = \alpha, \alpha+1, \dots, m-\alpha) \quad (3.24)$$

Le calcul des nouveaux coefficients précédants ne change pas le principe. Il suffit simplement de modifier la formule(3.18) par :

$$Q_i = (1 - \lambda_{i,n,\alpha}) \cdot R_i + \lambda_{i,n,\alpha} \cdot S_i, \quad (i = 0,1,\dots,n-1) \quad (3.25)$$

les facteurs $\{\lambda_{i,n,\alpha}\}_{i=0}^{n-1}$ sont déterminés par :

$$\lambda_{i,n,\alpha} = \begin{cases} 0 & (i = 0,1,\dots,\alpha-1) \\ \frac{1}{\gamma_{n,\alpha}} \cdot \sum_{j=\alpha}^i \binom{n}{j} \cdot |c_{j,n,\alpha}| & (i = \alpha, \alpha+1, \dots, n-\alpha-1) \\ 1 & (i = n-\alpha, \dots, n-1) \end{cases} \quad (3.26)$$

où

$$\gamma_{n,\alpha} = \sum_{j=\alpha}^{n-\alpha} \binom{n}{j} \cdot |c_{j,n,\alpha}|$$

De plus, en utilisant le facteur de turbulence $s_{n,\alpha} = 2^{2 \cdot n-1} / \gamma_{n,\alpha}$ l'erreur d'approximation est donnée par : $d(P,Q) = s_{n,\alpha} \cdot 2^{1-2 \cdot n} \cdot |\Delta^n P_0|$

(3.27)

Remarques :

- Le facteur de turbulence $s_{n,\alpha} \geq 1$ est introduit pour représenter la qualité de la continuité $C^{\alpha-1}$ en relation avec l'approximation non forcée où $s_{n,0} = 1$.
 - On a $0 = \lambda_{0,n,\alpha} = \dots = \lambda_{\alpha-1,n,\alpha} < \lambda_{\alpha,n,\alpha} < \dots < \lambda_{n-\alpha-1,n,\alpha} < \lambda_{n-\alpha,n,\alpha} = \dots = \lambda_{n-1,n,\alpha} = 1$.
 - Le cas $n = 2, \alpha$ (exclu dans le théorème [Eck'93]) n'est pas intéressant puisqu'il représente simplement le schéma d'interpolation d'Hermite (construction de Forrest)
- Dans la pratique on ne calcule $\lambda_{i,n,\alpha}$ et $\gamma_{n,\alpha}$ qu'une seule fois pour chaque combinaisons de n et $\alpha \geq 1$ (la solution pour $\alpha = 0$ est donnée dans (3.19)). Alors les résultats obtenu pour $\{\lambda_{i,n,\alpha}\}_{i=0}^{n-1}$ et $s_{n,\alpha}$ doivent être enregistrées dans fichier. Pour une application on les rappelle et on calcule les nouveaux coefficients par (3.25), [Eck'93]

i	α	1	2	3
	0			
0	0.00003	0	0	0
1	0.00369	0.00287	0.03609	0
2	0.05923	0.05526	0.03609	0
3	0.30362	0.30011	0.28136	0.23083
4	0.69638	0.69989	0.71864	0.76917
5	0.94077	0.94474	0.96391	1
6	0.99631	0.99713	1	1
7	0.99997	1	1	1

Tableau 1 : les facteurs $\lambda_{i,n,\alpha}$ pour $\alpha = 0,1,2,3$

n	α	1	2	3
	0			
1	1			
2	1			
3	1	1.5397		
4	1	1.3726		
5	1	1.2852	4.5795	
6	1	1.2313	3.3886	
7	1	1.1946	2.7858	15.2332
8	1	1.1680	2.4235	9.8126

Tableau 2 : quelques facteurs $s_{n,\alpha}$

L'application k fois successive de ce processus nous donne la réduction du degré n au degré $n-k$. Ce qui nous conduit à l'algorithme suivant :

Algorithme ECK

Données : P_i ($i=0,1,\dots,n$) pôles de la courbe de degré n
 k : ordre de la réduction

α ordre de continuité

$m=n$

Pour $i=0,1,\dots,n$

$Q_i = P_i$

Fin pour

Pour $j=1,2,\dots,k$

BEN1 ($Q,1 \rightarrow R$)

BEN2 ($Q,1 \rightarrow S$)

Lecture ($\lambda_{i,m,\alpha}$)

Pour $i=0,1,\dots,m-1$

$$Q_i = (1 - \lambda_{i,m,\alpha}) \cdot R_i + \lambda_{i,m,\alpha} \cdot S_i$$

Fin pour

$m=m-1$

Fin pour

Résultats : Q_i ($i=0,1,\dots,m$) courbe de degré $m=n-k$

Fin

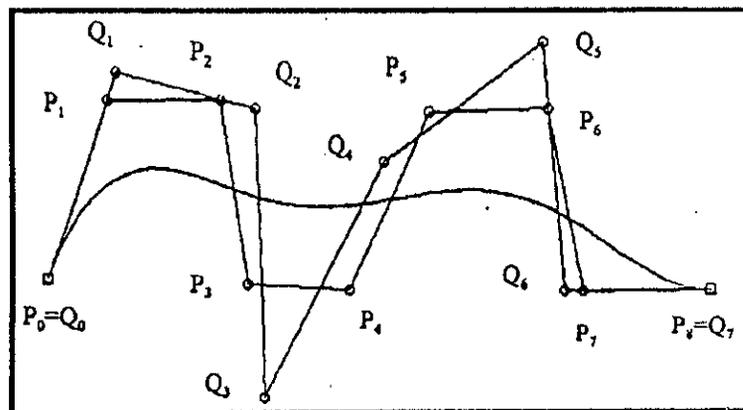


Figure 3.14.a. Courbe de Bézier de degré 8 réduite au degré 7 par l'algorithme de ECK en respectant la continuité de courbure ($\alpha = 3$)

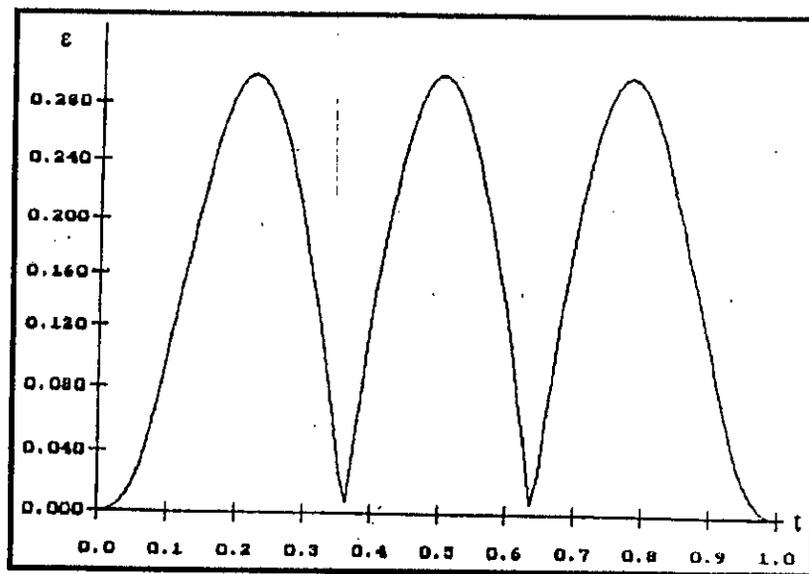


Figure 3.14.b. Variation de l'erreur de position

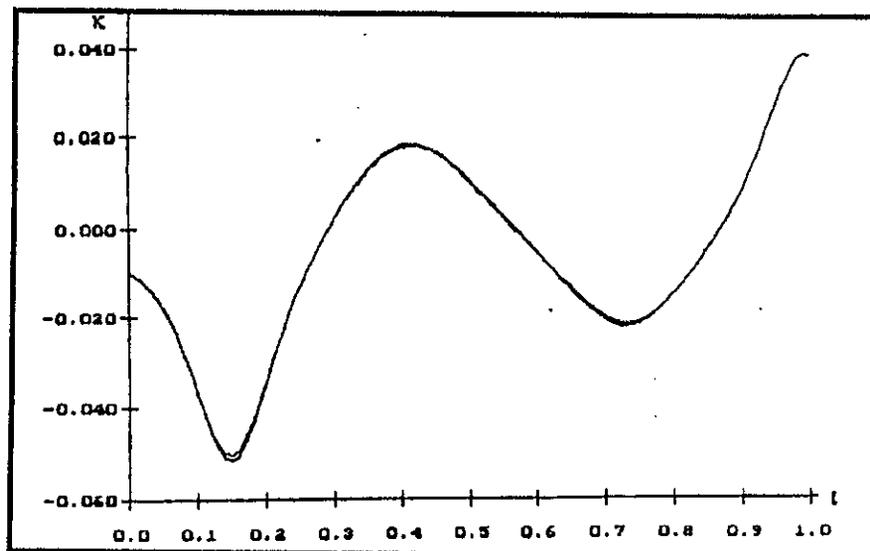


Figure 3.14.c. Variation des courbures

3.2.5.2. algorithme de conversion par approximation

Maintenant, on peut soulever la question sur le schéma de réduction de degré en respectant une tolérance de position ε fixée au préalable. a priori l'erreur estimée (3.20) est utile combinée à l'observation suivante : si on subdivise la fonction f de degré n ns -fois aux valeurs équidistantes de l'abscisse curviligne $t_i = i / (ns + 1)$ ($i=1,2,\dots,ns$), alors les n différences de pôles sur chacun des $ns+1$ segments de courbe est $\frac{1}{(ns+1)^n}$ fois la

différence originale $\Delta^n P_0$ de la fonction $P(t)$.

Donc, pour satisfaire la tolérance fixée ε , on doit d'abord subdiviser $Q(t)$ ns -fois à t_i et réduire en suite les segments de courbe obtenus. de préférence pour $\alpha \geq 1$ pour réaliser un ordre de continuité voulu. le nombre de subdivisions ns obtenu à partir de (3.27) est

$$ns = \left\lceil \frac{1}{4} \left(\frac{2 \cdot s_{n,\alpha} \cdot |\Delta^n P_0|}{\varepsilon} \right)^{1/n} \right\rceil \quad (3.28)$$

Evidemment, $ns=0$ veut dire qu'il n'est pas nécessaire de subdiviser pour respecter la tolérance donnée.

Brièvement, on reprend le problème de la réduction du degré n au $m < n$. Une solution peut être obtenue en appliquant successivement la méthode décrite ici, pour une réduction d'un degré à chaque étape.

Ce qui fait que la fonction obtenue $Q(t)$ n'est pas réellement une meilleure approximation à notre sens de minimisation de $d(P,Q)$. Mais dans le cas particulier où $\alpha = 0$, une autre distance est toujours minimisée. Il s'agit de la norme au sens des moindres carrés pondérée :

$$\left(\int_0^1 \frac{(P(t) - Q(t))^2}{\sqrt{t(1-t)}} dt \right)^{1/2} \quad (3.29)$$

puisque les polynômes de Tchebycheff sont orthogonaux d'après cette norme pondérée. de plus on déduit à partir de cette propriété que $P(t)-Q(t)$ possède $m+1$ solutions (zéros), ce qui veut dire que $P(t)$ et $Q(t)$ sont proches l'une de l'autre.

on ne veut valider ces segments dans le cas où $\alpha > 0$.

Néanmoins, des expériences pratiques ont montré que cette approximation est satisfaisante dans la majorité des cas. On peut déterminer l'erreur $d(P,Q)$ en rassemblant toutes les erreurs commises à chaque étape.

Finalement, si on veut respecter une tolérance ε fixée, on doit satisfaire une tolérance de $\frac{\varepsilon}{(n-m)}$ à chaque réduction comme précédemment en subdivisant k -fois si nécessaire.

Ce qui nous conduit à l'algorithme de conversion suivant : [Eck'93]

Algorithme CONV-ECK

Données : P_i ($i=0,1,\dots,n$) pôles de la courbe de degré n

K : ordre de la réduction

α : ordre de continuité

ε_0 : tolérance de position

```

m=n, a=1, b=1, c=1
Pour i=0,1,...,n
   $R_i^1 = P_i$ 
Fin pour
Pour r=1,2,...,k
  Pour f=a,a+1,...,b
    Pour i=0,1,...,m
       $S_i = R_i^f$ 
    Fin pour
    Lecture ( $\lambda_{i,m,\alpha}, s_{m,\alpha}$ )
    Calculer  $\Delta^m P_0$ 
    
$$ns = \left[ \frac{1}{4} \cdot \left( \frac{2 \cdot k \cdot s_{m,\alpha} \cdot |\Delta^m P_0|}{\varepsilon_0} \right)^{1/m} \right]$$

    Pour d= 1,2,...,ns
       $t_0 = 1/(ns - d + 2)$ 
      SUB ( $S, t_0 \rightarrow Q0, Q1$ )
      Pour j=0,1,...,m
         $S_j = Q1_j$ 
      Fin pour
      c=c+1
      ECK ( $Q0,1, \alpha \rightarrow Q$ )
      Pour j=0,1,...,m-1
         $R_j^c = Q_j$ 
      Fin pour
    Fin pour
    c=c+1
    ECK ( $S,1, \alpha \rightarrow Q$ )
    Pour j=0,1,...,m-1
       $R_j^c = Q_j$ 
    Fin pour
  Fin pour
  A=a+1, b=c, m=m-1
Fin pour
f=0
  Pour i=a,a+1,...,b
    f=f+1
    Pour j=0,1,...,m
       $Q_j^f = R_j^i$ 
    Fin pour
  Fin pour
nb =b-a

```

Résultats : Q_j^i ($j=0,1,\dots,m$) et ($i=1,2,\dots,nb$) nb courbe de degré $m=n-k$

Fin

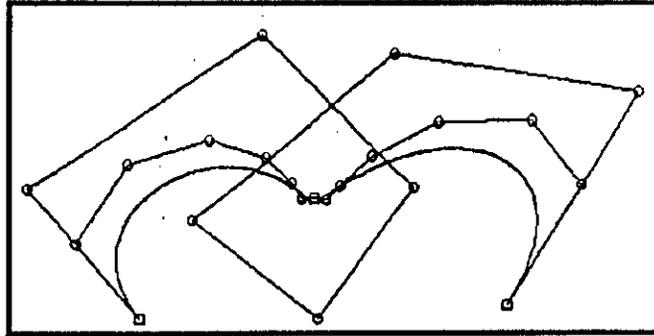


Figure 3.15.a. Courbe de Bézier de degré 8 réduite en deux courbes de degré 7 par l'algorithme CONV-ECK

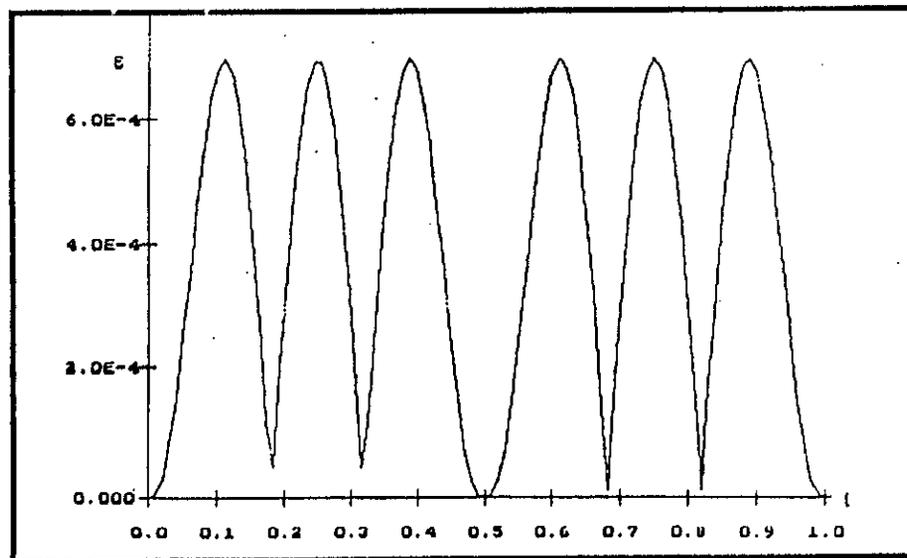


Figure 3.15.b. Variation de l'erreur de position

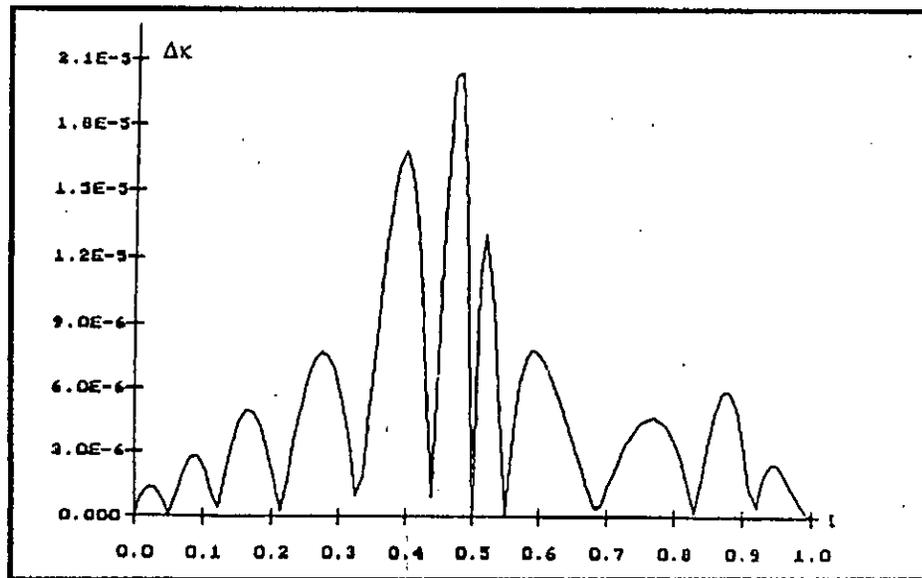


Figure 3.15.c. Variation de l'erreur sur la courbure

Interprétations des figures : la figure 3.14.a. représente une courbe de Bézier de degré 8 réduite au degré 7 par l'algorithme de ECK, tout en respectant la continuité de position ainsi la continuité de tangente (figures 3.14.b,c). la figure 3.15.a montre une courbe de Bézier non rationnelle de degré 8 réduite en deux courbes de degré 7 par l'algorithme CONV-ECK avec $\alpha = 3$ et $\varepsilon_0 = 0.01$, l'approximation obtenue est uniforme symétrique, représenté par la variation de l'erreur de position sur la figure 3.15.b., la continuité de courbure est vérifiée par la figure 3.15.c.

3.3. Analyse comparative des méthodes de conversion

3.3.1. Etude critique

La réduction de degré étant une approximation, par conséquent nous devons prendre en considération dans notre étude critique les critères s'y affairant qui sont la nature, la norme et l'erreur d'approximation, sont négliger les caractéristiques algorithmiques principales dont le temps de calcul, l'espace mémoire nécessaire pour le fonctionnement de l'algorithme ainsi que la facilité d'implémentation. Pour notre application la continuité géométrique et le nombre de courbes obtenues sont d'une importance primordiale. Compte tenu de ces critères, une première étude critique des méthodes présentées donne les appréciations suivantes :

Algorithme de Bensalah : la réduction a été considérée comme le processus inverse de l'élévation de degré, deux solutions sont alors possibles selon que l'on commence par définir le premier ou le dernier pôle ; l'auteur combine ces deux solutions par une moyenne pondérée qui oblige la solution à passer par les pôles extrêmes, ce qui donne une continuité d'ordre 0. nous remarquons alors qu'aucune notion de meilleure approximation n'est utilisée, et qu'on ne peut exiger un ordre de continuité géométrique, ce qui est un inconvénient de taille. L'algorithme générale à l'avantage d'être facile à implémenter rapide d'exécution mais pour des réductions de degré $n > 2$, le nombre de courbes réduites est relativement important. Pour la continuité d'ordre C^1 un raccordement des tangentes est indispensable.

Algorithme de Forrest : l'algorithme de Forrest est facile à implémenter et possède un temps de calcul très intéressant dû à l'absence de notions d'approximation.

Une continuité maximale est respectée aux extrémités de la courbe puisqu'on prend les $(n-1)/2$ premiers pôles du polygone 1 et les $(n-1)/2$ derniers pôles du polygone 2. Cette Grande exigence en continuité géométrique rétrécit le champ de l'approximation ce qui donne une erreur très élevée induisant un nombre de courbes réduites très élevé.

Algorithme de Hoschek : l'approximation utilisée dans cet algorithme est la méthode des moindres carrés enrichie par une paramétrisation intrinsèque donnant une erreur d'approximation très intéressante. Malheureusement cet algorithme fait appel à des méthodes de résolution de système non linéaires, qui engendrent un temps de calcul très lourd. Néanmoins, les résultats pratiques sont satisfaisants, comparés à ceux de la méthode de Bensalah du point de vue continuité et nombre de segments obtenus principalement dans le cas où $(n-m)/2$, n représentant le degré de la courbe initiale. C'est une méthode qui donne un lien entre l'ordre de continuité k à respecter et le degré des courbes réduites à obtenir $m = 2.k + 1$. L'inconvénient majeur reste donc le temps de calcul.

Algorithme de Eck : la méthode de Eck se Base sur la notion de la meilleure approximation qui sera perdue par la suite à cause de l'ordre de continuité qu'on se propose de respecter, mais se problème est maîtriser grâce à une transformation des polynômes de Tchebycheff appelés alors contraints. cette méthode se base sur la réduction par les polygones moyens dans la quelle l'auteur introduit les facteurs de pondérations $\lambda_{i,n,\alpha}$ (pour une continuité d'ordre $\alpha - 1 : C^{\alpha-1}$) qui est dépendant de l'ordre de continuité à respecter nécessitant un

algorithme de calcul externe très complexe. Grâce à cette combinaison et à la subdivision, on arrive à respecter une tolérance fixée pour un ordre de continuité voulu.

Algorithme de Watkins et Worsey : en basons sur la technique d'approximation dite minimax nous retrouvons dans l'algorithme de Watkins et Worsey une notion très importante concernant la meilleure approximation, utilisant la norme d'approximation donnée par le théorème de l'alternation, Malheureusement un problème de continuité aux extrémités se pose. pour remédier à cela les auteurs suggèrent de changer directement les pôles extrêmes obtenus donnant une approximation dite forcée qui fait perdre la notion de la meilleure approximation, donnant aussi une continuité de position. l'algorithme global est d'une facilité d'implémentation, mais nécessite des matrices de passage de la base de Bernstein à la base de Tchebycheff et inversement qui doivent être calculées par un algorithmes externe. les résultats pratiques obtenues par l'auteur donnent une erreur d'approximation et un nombre de segment corrects.

3.3.2. Comparaisons analytique

Nous remarquons après analyse mathématique parmi ces méthodes que la notion de la meilleure approximation n'est pas encore respectée, puisqu'elle se trouve à chaque fois perdue, ce qui fait que le problème de meilleure réduction de degré reste toujours posé. En CFAO, La continuité géométrique est une caractéristique très importante qui est donc indispensable dans les approximations de réduction de degré en vue de l'échange de données entre système. A priori la méthode de Bensalah, Forrest ainsi que celle de Watkins et Worsey ne répondant pas à ce critère sont à éviter.

Une raison qui élimine ces trois méthodes est qu'elle sont incluses dans la méthode de Eck, si on réduit dans l'algorithme de Eck les $\lambda_{i,n,\alpha} = i/n$, c'est à dire $\alpha = 0$ on retombe dans l'algorithme de Bensalah, de même si on choisit la valeur maximale pour $\alpha (m/2)$ suivant le nombre de pôles à obtenir après réduction, on retombe sur l'algorithme de Forrest; et finalement si on prend $\alpha = 1$ ce qui réduit les $\lambda_{i,n,\alpha}$ aux $\lambda_{i,n}$ on retrouve l'algorithme de Watkins et Worsey. Finalement, il ne reste à comparer objectivement que les algorithmes de Hoschek et de Eck.

L'avantage principal de l'algorithme de Hoschek est l'utilisation de la paramétrisation intrinsèque des courbes afin de minimiser l'erreur vectoriel au sens des moindres carrés. les inconvénients de l'algorithme de Hoschek sont l'utilisation d'une méthode numérique non linéaire qui risque d'alourdir l'erreur ainsi que le temps de calcul, un autre inconvénient est que l'algorithme nécessite de calculer l'interpolation de la courbe à chaque modification du paramétrage.

Les inconvénient de l'algorithme de Eck se retrouvent pratiquement dans tous les autres algorithmes. C'est que les avantages de l'algorithme ne sont justifiés que par des résultats pratiques vu l'absence de notion de meilleure approximation, et principalement aucune notion mathématique ne peut dire que l'erreur est minimale, ce qui maintient encore posé le problème de la meilleure réduction de degré par approximation.

Analytiquement nous pouvons conclure que la méthode de Eck trouvera une utilisation plus large que celle de Hoschek. les résultats pratiques sont plus significatifs que les justifications analytiques et algorithmiques du fait de l'absence de notion de meilleure approximation, il est donc indispensable de compléter cette étude par une comparaison expérimentale.

3.3.3.comparaison expérimental

Pour la méthode de Eck nous ne pouvons faire de comparaison sur la qualité de l'approximation car nous disposons que des coefficients $\lambda_{i,8,\alpha}$ et donc on ne peut appliquer la méthode que pour réduire du degré 8 au degré 7 avec $\alpha = 0,1,2,3$.

Les temps de calcul pour la méthode de Hoschek sont minimiser au maximum puisque nous considérons que la paramétrisation converge à la première itération et pour la résolution du système d'équation non linéaire la méthode de Newton converge à la troisième itération .

Des tests ont été faits dans [Asma'99] , en implémentant les algorithmes dans l'environnement d'un logiciel de C.F.A.O (CONVER de L'IGM de l'université de Boumerdés) ce qui peut donner en temps de calcul des résultats légèrement supérieurs aux algorithmes programmés seuls .

Compte tenu des épreuves les plus fréquentes à subir dans la réduction de degré nous avons choisi les tests relatifs aux manipulations les plus rencontrées .

Les tests sont appliquées sur une machine P.C.486/DX2 66 MHz avec un coprocesseur mathématique . En fonction de la méthode le temps CPU est calculer après exécution d'un nombre défini de fois l'algorithme .

Compte tenu des résultats pratiques de ces méthodes nous pouvons tirer les conclusions suivantes

- Les méthodes de Forrest et Bensalah ayant le même temps de calcul sont les plus rapides .
- La méthode de Hoschek produit pour une même continuité la meilleure approximation parmi toutes les méthodes présentées ;
- L'approximation de Forrest donne l'erreur la plus élevée ;
- L'algorithme de Hoschek donne le temps le plus élevé bien que l'on ait supposé que la paramétrisation converge à la première itération et que le temps de résolution du système non linéaire soit minimal ;
- La méthode de Watkins et Worsey donne un temps de calcul intéressant et une bonne approximation ;
- Le temps de calcul de la méthode de Eck est encore plus intéressant que celui de l'algorithme de watkins et Worsey ;

Sachant que l'erreur d'approximation décroît avec la contrainte de continuité , l'approximation de Watkins et Worsey ne peut être jugée intéressante puisqu'elle ne respecte que la continuité minimale . Contrairement la méthode de Forrest respecte la continuité maximale et donc évidemment l'erreur d'approximation la plus élevée (et quelquefois intolérable) . L'erreur est encore répandant au critère de la continuité géométrique seront rejetés .

Par conséquent la méthode de Hoschek demeure la méthode la plus adéquate pour l'échange de données entre systèmes .

Ce qui est résumé par le tableau comparatif suivant :

Méthodes/ Critères	Bensalah	Forrest	Watkins et Worsey	Eck	Hoschek
Nombre de courbe obtenue	Très Elevé	Elevé	Réduit	Réduit	Réduit
Temps de calcul	Très Elevé	Très court	Court	Court	Très Elevé
Critères imposables	Erreur de position ε_0	Erreur de position ε_0	Erreur de position ε_0	Erreur de position ε_0 , ordre de continuité C	Erreur de position ε_0 , ordre de continuité $k(m=2.k+1)$
Continuité	Continuité de position et de tangente	Continuité maximale	Continuité de position	Assure la continuité imposée C	Assure la continuité imposée k.
Passage à d'autres modèles	L'algorithme ne permet pas le passage à d'autres modèles	L'algorithme ne permet pas le passage à d'autres modèles	Nécessite un algorithme de passage du modèle voulu à la base de Tchebycheff	Nécessite un algorithme de passage du modèle voulu à la base de Tchebycheff	Possible en modifiant la base de Bernstein par la base du modèle voulu
Amélioration possibles	On peut prévoir une amélioration de l'algorithme de façon à réduire le nombre de courbe	On peut prévoir une amélioration de l'algorithme de façon à réduire le nombre de courbe	L'amélioratio n de cet algorithme nous conduit simplement à l'algorithme de Eck	Détermination intrinsèque des coefficients de la moyenne pondérée	Possibilité de déterminer le nombre de courbes à obtenir avant de commencer la méthode

3.3.4. Conclusion

La méthode de Bensalah a l'avantage d'avoir un temps de calcul très réduit, et une facilité remarquable pour le passage aux surfaces de Bézier non rationnelles, mais produit une erreur très élevée, et donc un nombre de courbes réduites très grand, sans pour cela respecter de continuité géométrique au-delà de l'ordre 1.

La continuité géométrique maximale étant respectée automatiquement dans l'algorithme de Forrest, induit une erreur d'approximation très élevée produisant un nombre de courbe réduite élevé. Le passage aux surfaces est facile à implémenter.

L'inconvénient majeur de l'algorithme de Watkins et Worsey est la continuité géométrique qui n'est respecté qu'à l'ordre 0. Cette méthode est basée sur un développment mathématiques

Intéressant, qui fait que la méthode est d'une rapidité appréciable, grâce à la détermination du nombre de courbes nécessaire pour respecter une tolérance fixée.

Le temps de calcul dans la méthode de Eck est intéressant pour une méthode respectant un ordre de continuité désiré. Cette approximation fait appel à un algorithme spécifique complexe pour la détermination des constantes de la moyenne pondérée. Pour manque de valeurs de ces coefficients, les tests sont limités pour les réductions du degré 8 au degré 7.

Bien que la méthode de Hoschek utilise une méthode de résolution d'équations non linéaire dont les équations nécessitent un grand effort de calcul, elle est jugée satisfaisante puisqu'elle répond à toutes les exigences d'un transfert entre systèmes. Elle utilise une paramétrisation intrinsèque et assure la continuité désirée. Son temps de calcul est très élevé, mais on se propose d'approcher une étude pour le réduire.

CHAPITRE 4

Méthode proposée

4. Méthode proposée

4.1. Introduction

La réduction de degré est souvent nécessaire lors de la conversion des modèles de courbe et surface en vue de l'échange de données entre différents systèmes ..

Nous proposons une méthode permettant de calculer la courbe de Bézier de degré minimal approchant la courbe originale avec une précision donnée. Cette méthode basée sur la minimisation progressive des écarts entre les courbes a l'avantage de modifier le paramétrage de façon transparente et de respecter les contraintes de continuité très facilement. Nous présentons pour finir une introduction à l'extension du procédé aux surfaces[GJA'99].

Une première solution consiste à inverser le procédé d'augmentation du degré [Forrest'88]. Les résultats obtenus sont donc directement liés au réseau de pôles de la courbe d'origine et peuvent être différents selon le sens de parcours du réseau. De plus certaines conditions essentielles telles que les continuités ne sont pas toujours respectées.

La solution pour palier à ces problèmes est alors d'utiliser une combinaison des différents résultats.

De nombreuses études portent sur ce type de méthodes et ont permis d'améliorer considérablement leurs performances [Eck93] [Eck95] [PBY96] [WW88] [GBB96] [Lac88][Hoschek'88]. La seconde possibilité est d'interpoler un certain nombre de points de la forme d'origine par une forme paramétrique du degré désiré. Pour cela, il faut dans un premier temps extraire les points interpoler et ensuite leur attribuer une valeur de paramètre. Cette opération de paramétrage est habituellement le point critique du procédé d'interpolation notamment à cause de la difficulté à définir un critère de qualité mais dans le cas de la réduction de degré il est possible de donner une définition simple du paramétrage optimal : « *Pour une sélection de points donnée, le paramétrage optimal est le paramétrage qui permet d'interpoler la forme la plus proche de la forme d'origine* »

Pratiquement, la méthode communément employée est l'approximation au sens des moindres carrés d'un grand nombre de points en utilisant le paramétrage d'origine. Dans les deux cas, les résultats ne sont pas toujours acceptables et l'optimum est rarement atteint. Cependant le second type de méthode semble plus intéressant à étudier, d'abord parce que les calculs utilisent naturellement les caractéristiques intrinsèques de la courbe, ensuite car il pose clairement la question du paramétrage trop souvent sous estimée. Cette étude se limitera donc à la

réduction de degré des courbes et surfaces de Bézier par l'interpolation. Dans une première partie nous développerons une réflexion sur les critères les plus importants en ce basant sur le cas des courbes planes. Nous proposerons ensuite une méthode permettant de calculer ces critères dans le cas général des courbes. Enfin, nous présenterons quelques résultats obtenus en utilisant une généralisation de la méthode aux courbes et surfaces.

4.2. Critères influants sur l'interpolation de courbes de Bézier pour la réduction de degré.

4.2.1 Analyse des caractéristiques de la courbe et choix des points à interpoler

La première étape est d'extraire de la courbe d'origine les points à interpoler. Dans le cas d'une résolution par approximation du type moindres carrés le nombre de points n'a que peu d'importance théorique, il est donc aisé de choisir beaucoup de points. Le choix devient par contre plus difficile dans le cas de l'interpolation car le nombre de points est directement lié au degré que l'on souhaite obtenir.

De plus il n'est pas rare d'obtenir, à paramétrage équivalent, des résultats différents selon les points interpolés. La façon la plus simple d'opérer est une répartition régulière des points le long de la courbe d'origine (figure 4.3). Une autre possibilité est d'étudier les caractéristiques intrinsèques de la courbe et d'utiliser des points particuliers de la courbe. Cette seconde proposition permet aussi de calculer une estimation du degré minimal nécessaire pour interpoler la courbe.

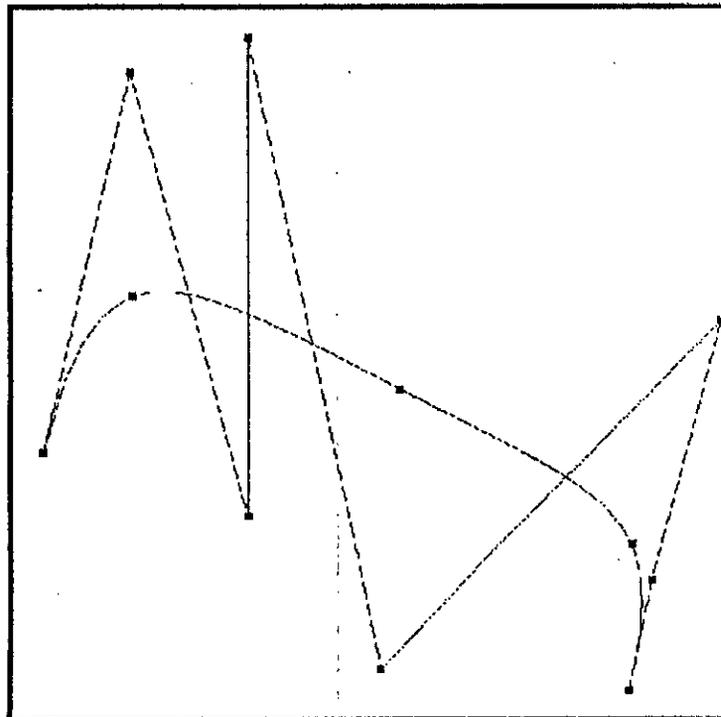


Figure 4.1. Sélection : courbures max

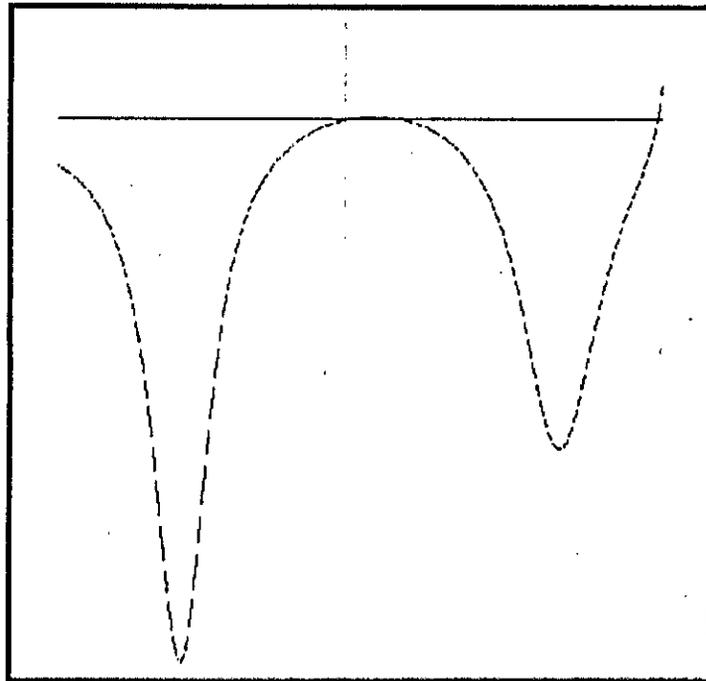


Figure 4.2. Variation de courbure

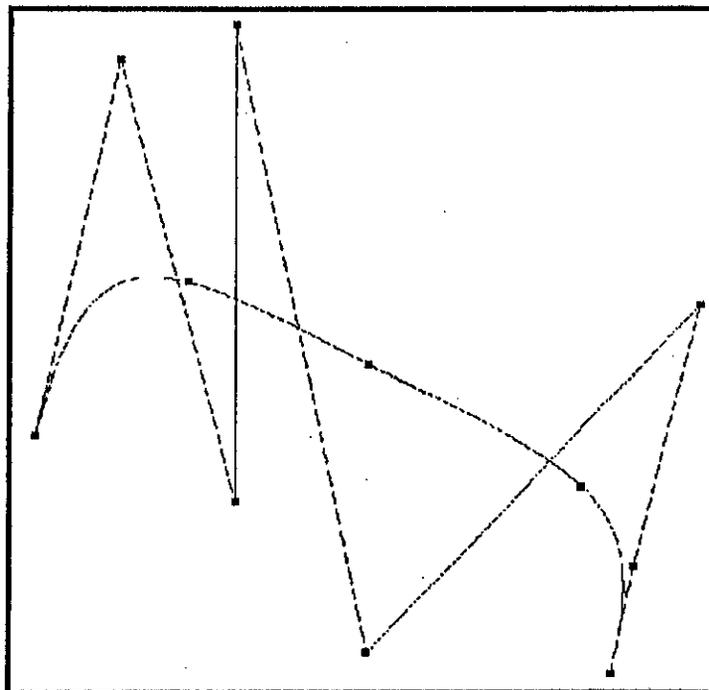


Figure 4.3. Sélection : répartition régulière

Par exemple l'étude de la variation de courbure (fig.4.2) dans le cas de la courbe plane (fig.1) indique trois pics de courbures. La courbe ne pourra être interpolée précisément

qu'avec une courbe de degré supérieur ou égal à quatre et les points correspondants aux pics de courbure sur la courbe pourront être utilisés pour l'interpolation. Il est supposé que toutes les variations de courbure et de torsion sont pertinentes, ce qui n'est pas toujours exact. Il sera alors nécessaire de 'lisser' les variations de la courbure. Dans le cas général des courbes, il faudra aussi tenir compte des variations de la torsion.

4.2.2. Le paramétrage

Le problème du paramétrage est trop souvent négligé dans les méthodes de réduction de degré, fréquemment on utilise le paramétrage d'origine qui est considéré à tort comme suffisant. C'est notamment le cas des méthodes basées sur l'inversion du procédé d'augmentation de degré (cet algorithme ne modifie pas le paramétrage de la courbe).

Pour analyser le paramétrage d'une courbe nous utilisons un outil graphique : Soit $f(s) = t$, la fonction appelée 'loi de paramétrage' qui retourne le paramètre d'un point de la courbe en fonction de son abscisse curviligne. Le tracé de cette fonction permet de visualiser et d'étudier aisément les différentes formes de paramétrage. Le paramétrage linéaire $f(s) = s$ (aussi appelé 'par longueur de corde') est utilisé comme référence. Dans un premier temps, nous allons étudier la pertinence de l'emploi du paramétrage d'origine. Nous évoquerons ensuite les différentes formes de paramétrage connues.

4.2.2.1 Le paramétrage d'origine

Une première expérience montre deux courbes identiques de degrés différents et leurs lois de paramétrage. Les différences sont visiblement importantes ce qui va à l'encontre des méthodes qui utilisent les caractéristiques géométriques de la courbe pour déterminer le paramétrage.

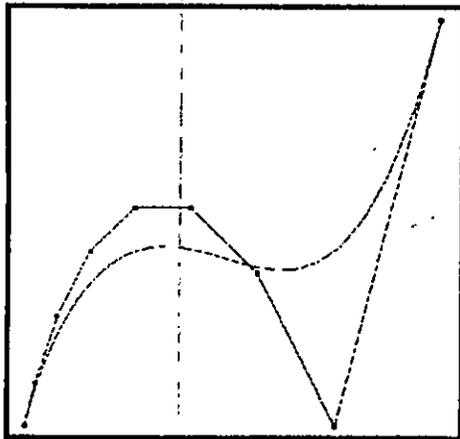


figure.4.4. degré 8

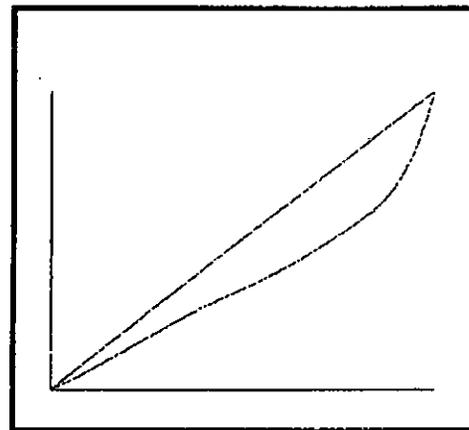


figure.4.5. Paramétrage

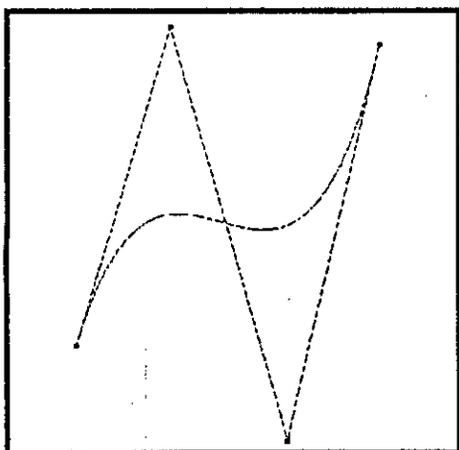


figure. 4.6. degré3

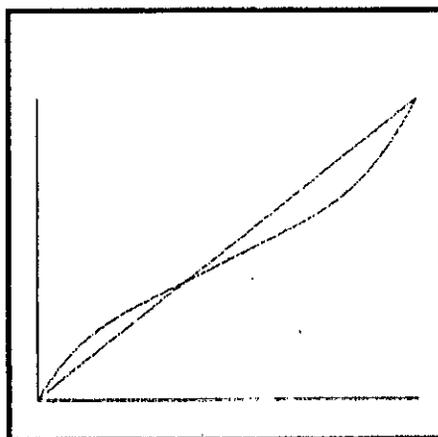


figure. 4.7. paramétrage

Une seconde expérience montre une réduction de degré sur une courbe de Bézier (fig. 8) par la méthode de Eck (fig. 4.10). L'algorithme de Eck calcul la meilleure approximation au sens des moindres carrés en minimisant la distance

$$d_2(Y_n - \bar{Y}_{n-1}) = \sqrt{\int_0^1 \|X_n(t) - \bar{X}_{n-1}(t)\|^2 dt}$$

pourtant les résultats obtenu est perfectible est

une rapide modification manuelle des pôles donne le résultats (figure 4.12) qui est visiblement bien meilleure qualité .

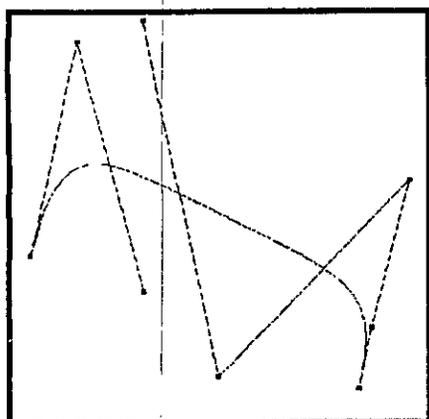


Figure. 4.8. courbe d'origine

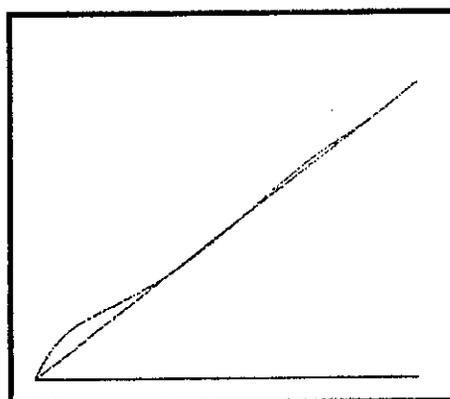


Figure. 4.9. paramétrage d'origine

Pour étudier les modifications apportées à la courbe de Eck, nous comparons les deux courbes obtenues après la réduction de degré et la courbe d'origine.

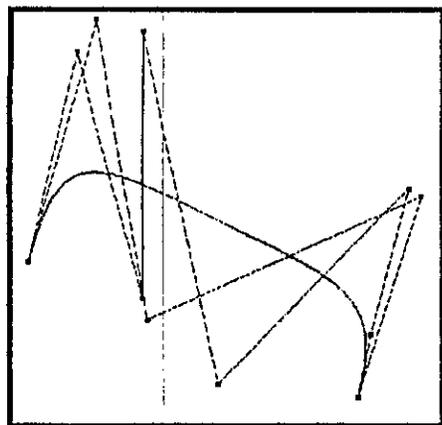


Figure.4.10.Méthode de Eck

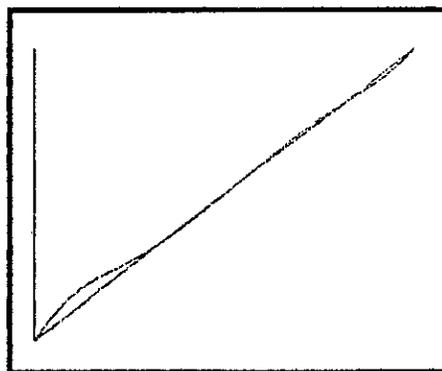


figure.4.11.Paramétrage de Eck

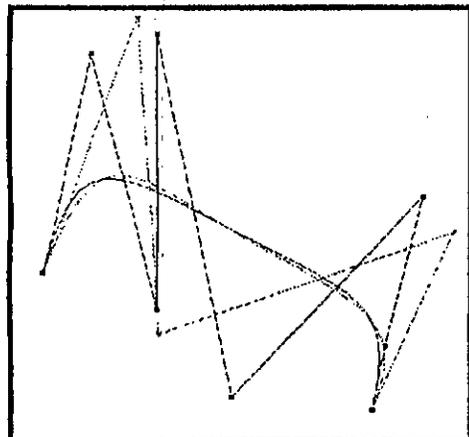


Figure 4.12. Modification manuelle

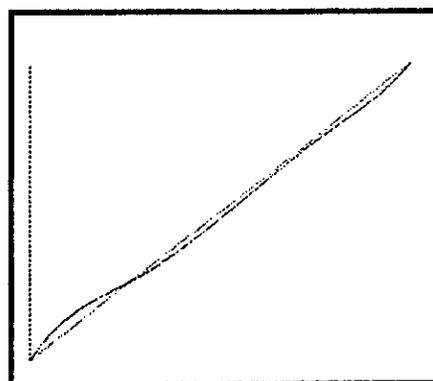


figure.4.13. Paramétrage manuelle

La loi de paramétrage n'a pas évolué dans le cas de la méthode de Eck (figure 4.11) alors qu'elle varie significativement lorsque nous modifions les pôles manuellement (figure 4.13) ce qui est cohérent avec notre première expérience. De toute évidence, le paramétrage d'origine n'est pas une bonne solution et un reparamétrage s'avère nécessaire.

4.2.2.2 Comparaisons de différents paramétrages

La problématique du paramétrage a été longuement étudiée et les propositions nombreuses. Hormis les schémas les plus simples (uniforme, linéaire), nous citerons les propositions de Lee [Lee'84] et Foley.

Lee propose une variation centripète du paramètre alors que Foley propose une méthode qui tient compte des variations de courbures en ralentissant la progression du paramétrage lorsque la courbure est forte. Ces propositions fournissent que rarement des résultats acceptables car elles ne tiennent pas compte du degré de la courbe à interpoler. Dans une étude précédente [JBA'95], nous proposons d'approcher la loi de paramétrage optimale par une courbe. Ces paramétrages 'paraboliques' et 'cubiques' sont calculés en fonction des conditions aux limites de la courbe. La démarche est intéressante, mais la méthode utilisée était lourde en calcul et finalement limitée à des lois de paramétrage de degré 3, ce qui est insuffisant. Une autre méthode, proposée par [Hoschek'88], mérite un peu d'attention. Hoschek introduit une décomposition de l'erreur entre un point à interpoler et son équivalent sur la courbe en erreur normale et tangentielle.

L'erreur tangentielle Δc , est alors interprétée comme une erreur de paramétrage qui peut être minimisée par approximations successives. Cette méthode nécessite de calculer l'interpolation de la courbe à chaque modification du paramétrage.

4.2.3 Conclusions

Tout d'abord, nous constatons que le paramétrage d'une même courbe varie selon le degré et que le paramétrage d'origine ne peut donc pas être considéré comme une bonne approximation du paramétrage optimal. Il est nécessaire de recalculer le paramétrage d'une courbe lors de la réduction de degré. Lors d'une étude précédente [Wah99], nous avons remarqué que le paramétrage obtenu en construisant manuellement la courbe était la meilleure approximation du paramétrage optimal, les résultats obtenus avec ce paramétrage 'manuel' étaient de loin les plus précis. De plus nous avons constaté que le choix des points à interpoler n'avait plus d'influence dès lors que l'on utilisait un bon paramétrage. Enfin, il est notable que les différentes formes de paramétrage existantes n'arrivent pas à approcher le paramétrage optimal. La définition du paramétrage optimal proposée en introduction doit être modifiée en conséquence : « *Pour un degré définit, on qualifie d'optimal le paramétrage qui permet d'interpoler la forme la plus proche de la forme d'origine.* ».

4.3 .Recherche d'un paramétrage optimal

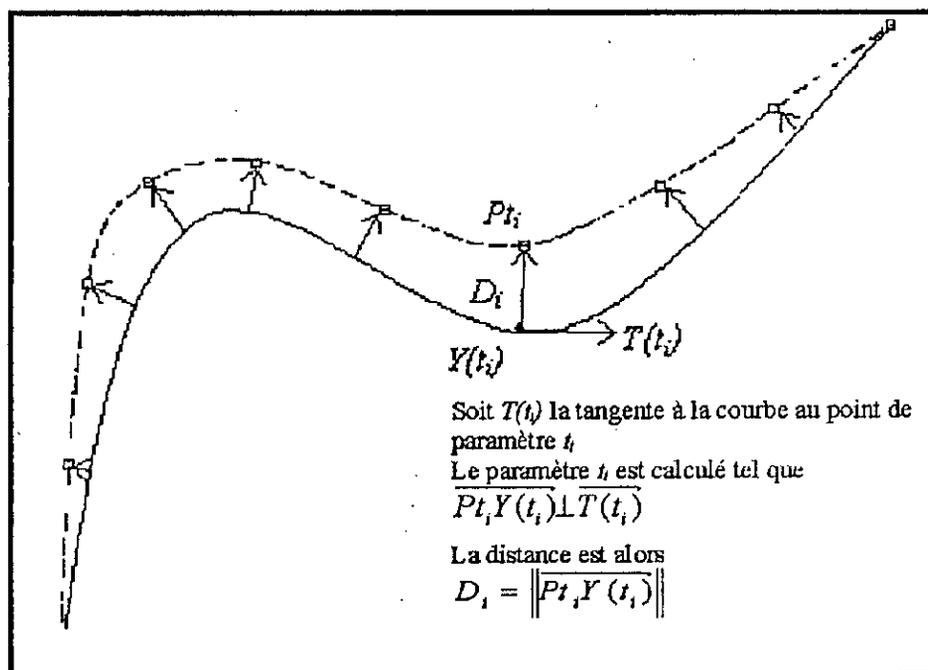
4.3.1 Principe

Nous avons vu que la meilleure approximation du paramétrage optimal était obtenue en construisant manuellement une courbe du degré désiré. Pour simuler cette construction,

nous utilisons une méthode itérative de minimisation des écarts normaux entre deux courbes. Une première courbe est grossièrement construite puis les écarts sont progressivement réduits par modification de son réseau de pôles.

4.3.2 Définition et recherche de l'écart entre deux courbes 'proches'

L'écart normal entre les deux courbes est défini en un certain nombre de points de mesure $\{P_{t_i}\}$ sélectionnés sur la courbe de référence.



On recherche sur la courbe à modifier les points dont la normale passe par un point de mesure.

On obtient ainsi une mesure de l'écart selon la normale à la courbe indépendante du paramétrage.

4.3.3 Minimisation de l'écart par modification du réseau de pôles

Pour minimiser les écarts, nous utilisons une méthode itérative basée sur le déplacement des pôles.

Pour une courbe de Bézier non rationnelle, le déplacement d'un pôle influe sur la quasi-totalité de la courbe. Mais cette influence est variable et une courbe réagira le plus

fortement à un déplacement de l'un de ses pôles P_i au voisinage du paramètre $t_i = \frac{i}{n}$ ($n =$ degré de la courbe).

Le processus de minimisation d'un écart en un point de paramètre t_j devra donc modifier tous les pôles en tenant compte de l'influence relative de chacun en cet endroit de la courbe. Pour cela nous utilisons le coefficient de Bernstein $B_i^n(t)$ qui atteint son maximum

lorsque $t = \frac{i}{n}$ le déplacement de chacun de ces pôles pour minimiser un écart ε_j sera donc $\delta P_{ij} = \varepsilon_j \cdot B_i^n(t_j)$. Pour réduire l'erreur globale entre les deux courbes, une première possibilité est de minimiser les écarts maximums, le déplacement de chaque pôle P_i sera alors $\delta P_i = \sum \max(\delta P_{ij}) = \sum [\max(\varepsilon_j) \cdot B_i^n(t_j)]$.

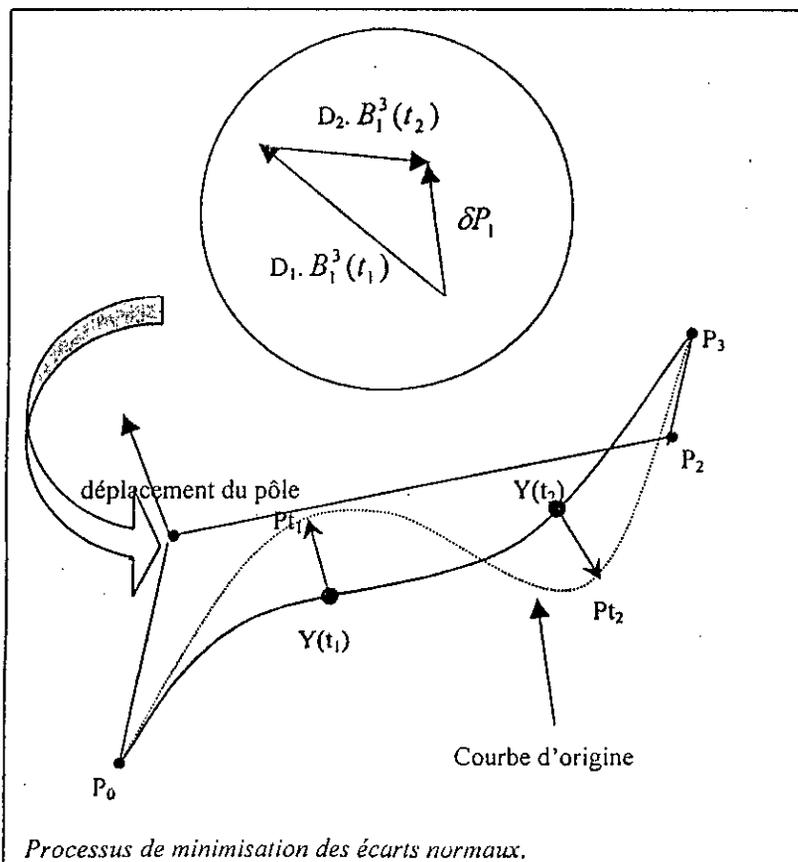
Cette minimisation est plus chère en calcul et converge lentement vers une solution stable. Une seconde solution est de minimiser l'ensemble des écarts de la courbe, le déplacement de chaque pôle P_i est alors $\delta P_i = \sum \delta P_{ij} = \sum [\varepsilon_j \cdot B_i^n(t_j)]$.

Cette solution est plus rapide mais aussi plus instable et l'on est souvent obligé de recourir aux techniques de relaxation pour converger.

Le paramètre t_j associé à un écart est recalculé à chaque itération, le paramétrage évolue donc de manière continue.

4.3.4 Algorithme

L'algorithme utilise comme paramètres d'entrée un ensemble de points $\{Pt_j\}$ extraits de la courbe de référence et une courbe définie par un réseau de pôle $\{P_i\}$. Le nombre de points Pt_j n'est pas limité.



Algorithme

Pour chaque point Pt_j

Rechercher le point $Y(t_j)$ de la courbe de paramètre t_j et dont la normale passe par le point Pt_j

Calculer l'écart ε_j entre $Y(t_j)$ et Pt_j

Pour chaque pôle P_i

$$\delta P_i = \sum \varepsilon_j \cdot B_i^n(t_j)$$

Déplacer les pôles : $\{P_i\} = \{P_i\} + \{\delta P_i\}$

Vérifier la convergence.

4.4 .Réduction de courbes et surfaces

La méthode construit, en plus du paramétrage optimal, une courbe dont la précision est égale ou meilleure à celle obtenue après interpolation. Il est donc inutile de calculer l'interpolation et elle sera utilisée directement comme méthode de réduction de degré. Ceci évitera de se soucier du paramétrage qui évoluera sans aucune intervention pour s'approcher du paramétrage optimal. Nous proposons une série de compléments permettant d'adapter l'algorithme à cette nouvelle utilisation.

4.4.1. Recherche du degré minimal par élévation de degré

Le plus souvent, le problème de la réduction de degré est abordé de façon descendante c'est à dire que le degré est réduit successivement et que le processus de réduction est utilisé plusieurs fois dans son intégralité. En procédant ainsi, il n'y a aucune garantie qu'il n'est pas de meilleure solution. Il peut pourtant être primordial de trouver le degré minimal car cela permet d'alléger considérablement la taille des modèles ainsi que les calculs. Nous proposons une méthode moins conventionnelle qui s'adapte mieux à notre procédé pour calculer la courbe de degré minimal. Le calcul est initialisé avec une courbe du degré le plus faible possible. Le degré de cette courbe peut être choisi après analyse des caractéristiques de la courbe ou, pour plus de sûreté, fixé au seuil minimal de 3 qui permet d'assurer la continuité de tangente. Si la précision de l'approximation est inférieure à ce qui est souhaité, il suffit

alors d'augmenter le degré et de relancer un nouveau calcul de minimisation. En procédant ainsi nous avons la certitude d'obtenir la courbe du degré le plus faible correspondant à notre attente. De plus les résultats précédents sont réutilisés ce qui évite de recommencer les calculs dans leur intégralité.

L'exemple suivant montre les différentes étapes de la réduction de degré maximale d'une courbe de Bézier non rationnelle de degré 10 (figure 4.14) avec une précision relative inférieure au millième. La courbe initiale est calculée à l'aide de la méthode de Eck (figure 4.15). La courbe (figure 4.16) est l'approximation obtenue après application de l'algorithme de minimisation des écarts, la précision étant insuffisante, la courbe est élevée en degré. Une nouvelle application de l'algorithme donne alors le résultat (figure 4.17). Les figures (18)(19)(20)(21) montrent l'évolution de la loi de paramétrage à chaque étape.

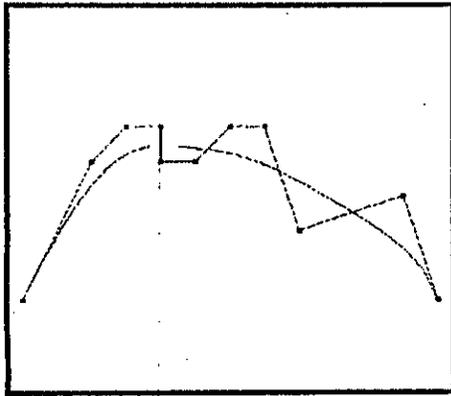


Figure 4.14. degré 10 initial

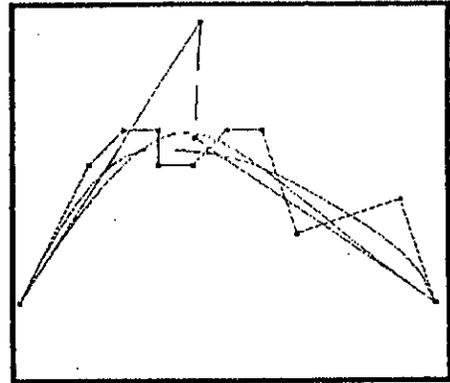


figure.4.15.Méthode de Eck

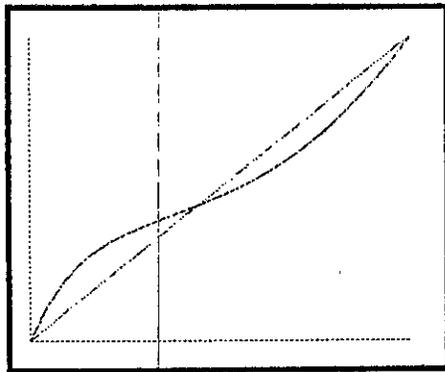


Figure4.18.

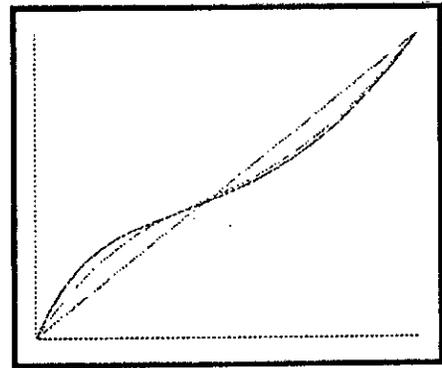


figure 4.19.

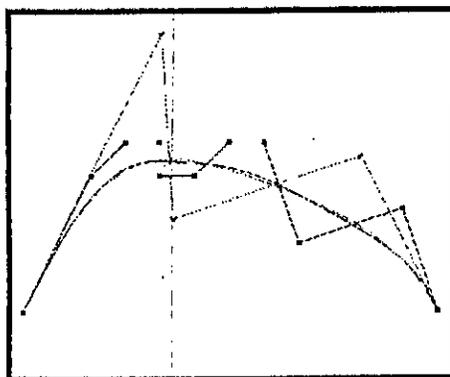


Figure 4.16. Degré 3
Précision 0.0055

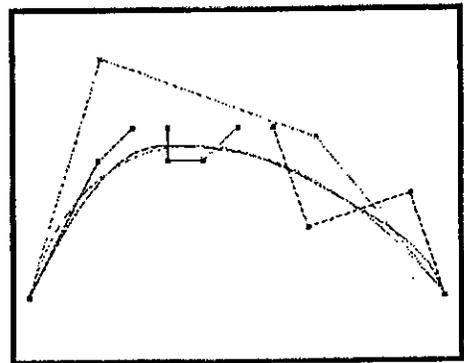


Figure 4.17. Degré 4
Précision 0.0009

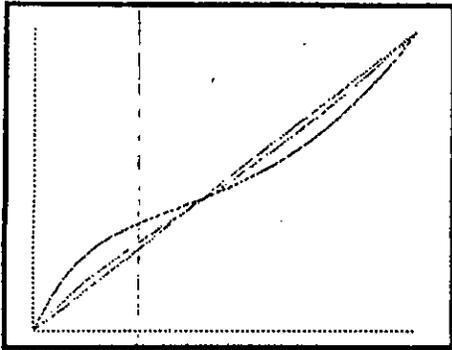


Figure 4.20.

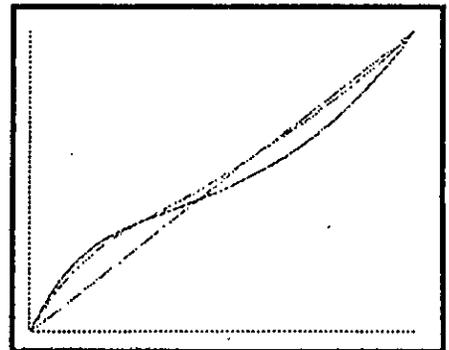


Figure 4.21.

4.4.2 Respect des tangentes aux extrémités

Pour respecter la cohérence du modèle, les contraintes de continuité doivent le plus souvent être respectées. Il est facile de modifier l'algorithme de manière à imposer les tangentes aux extrémités.

Pour cela, il suffit de projeter systématiquement les pôles P_1 et P_{n-1} sur les tangentes après chaque itération. Cette contrainte n'est pas sans effet sur le paramétrage. L'exemple suivant montre un cas où le respect des tangentes réduit fortement la précision de calcul. L'étude de variations de la courbure nous indique que le degré minimal pour cette courbe est 4. La courbe (figure 4.23), obtenue en respectant les tangentes aux extrémités, n'est pas suffisamment précise. On peut remarquer un écart important vers l'une de ces extrémités. En relâchant la contrainte, on obtient la courbe (figure 4.24) qui est bien plus précise.

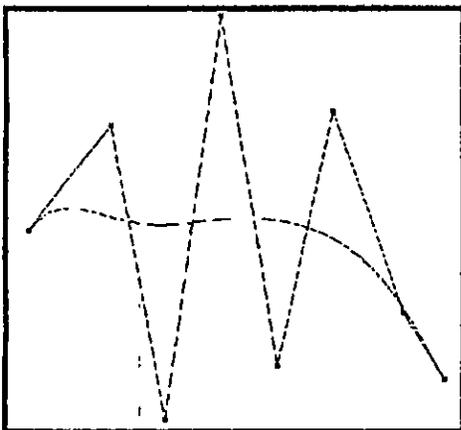


Figure 4.22. Degré 7 initial
 G^1

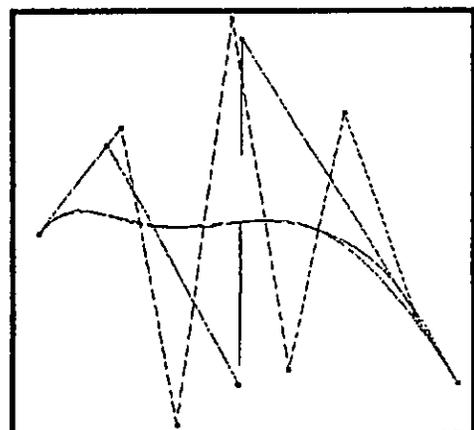


figure 4.23. Degré .4. continuité

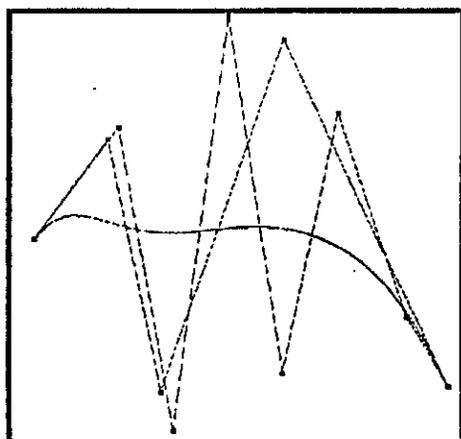


Figure 4.24. Degré 4. continuité G^0

4.5. Résultats

4.5.1. courbe

L'exemple suivant montre une courbe tridimensionnelle de degré 9 (figure 4.25). L'utilisation de la méthode avec 25 points de mesure et le respect des tangentes donne la courbe (figure 4.28) qui est une approximation de (figure 4.25) avec une erreur relative moyenne de 0,00245. La courbe (figure 4.31) représente l'évolution de l'erreur normale entre les deux courbes. On peut remarquer que la courbe d'erreur ondule presque régulièrement et que l'erreur maximale est proche de l'erreur moyenne (0,0042/0,0025). En fait les irrégularités que l'on peut voir sont amplifiées par le respect des contraintes de tangence.

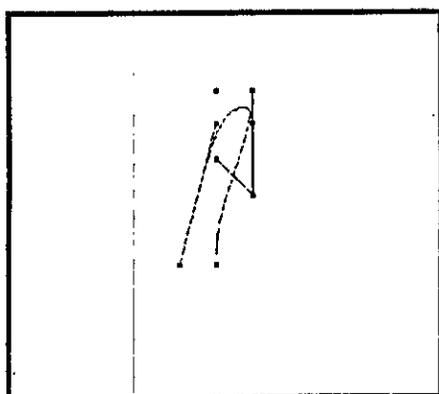


Figure 4.25. Vue de face

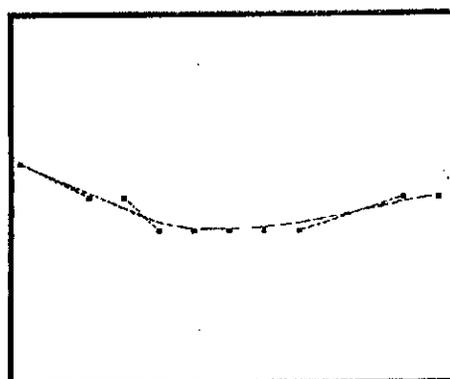


figure 4.26. Vue de coté

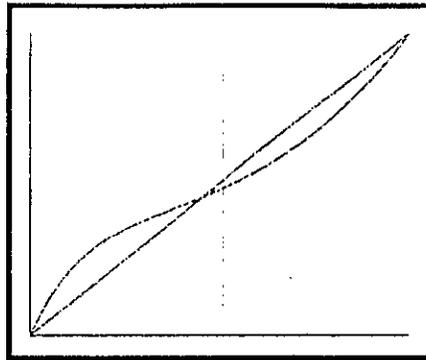


Figure 4.27. Paramétrage

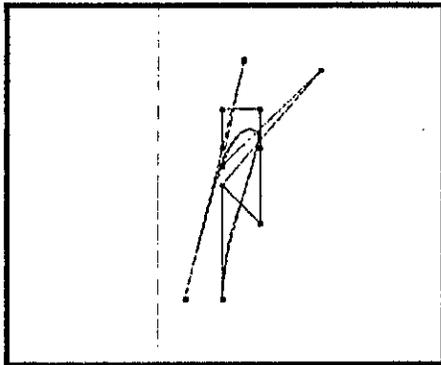


Figure 4.28. Vue de face

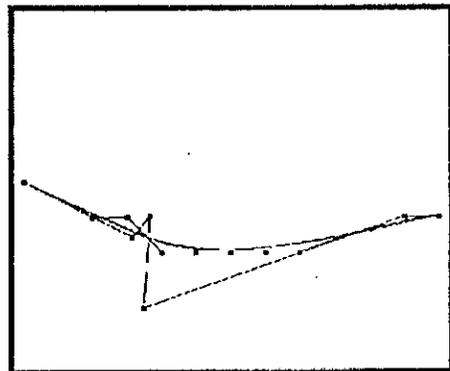


Figure 4.29. Vue de coté

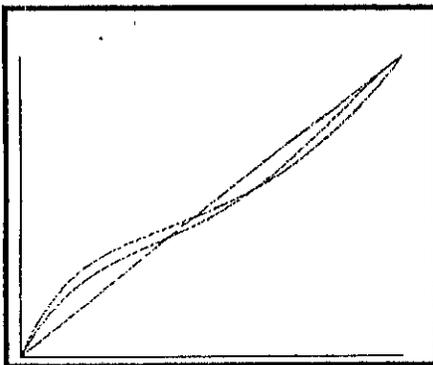


Figure 4.30. Paramétrage

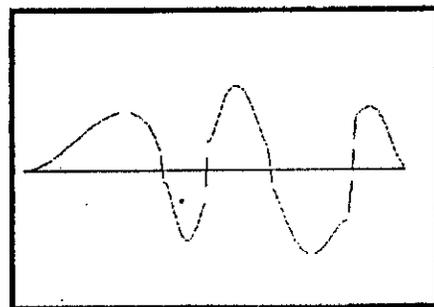


Figure 4.31. écart

4.5.2. Surface

L'extension de l'algorithme de base aux surfaces de Bézier non rationnelle repose sur les mêmes principes. Le déplacement de chaque pôle est alors calculé par

$$\delta P_{ik} = \sum \varepsilon_j \cdot B_i^n(u_j) \cdot B_k^m(u_j)$$

Les calculs dans le cas de surfaces sont bien plus importants, aussi il est important de soigner la précision de la surface initiale. Cet exemple est tiré d'une surface B-spline (figure 4.32) de degré 3*3, définie par un réseau de 23*23 pôles (figure 4.33).

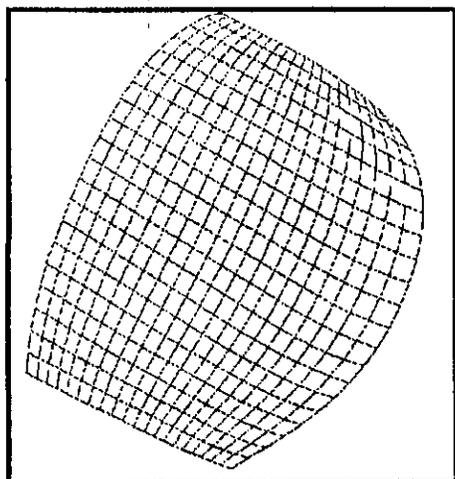


Figure 4.32. B-spline

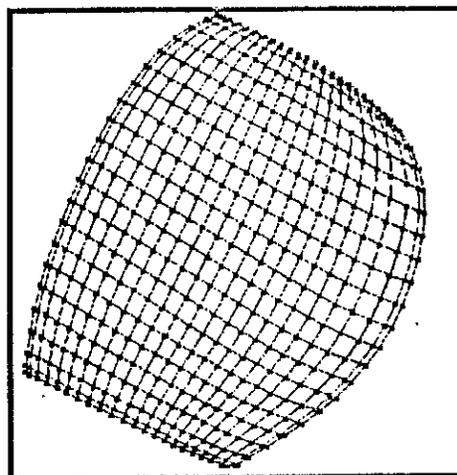


Figure 4.33. Poles

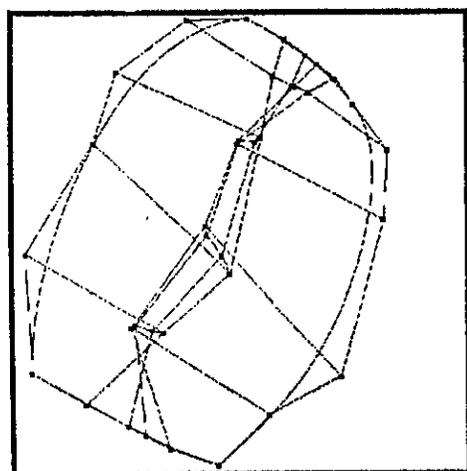


Figure 4.34. Bézier 5.5

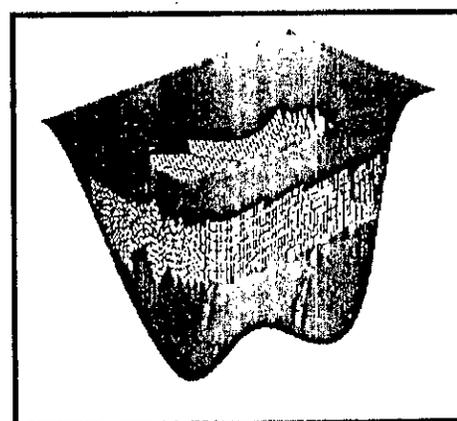


Figure 4.35. Ecart

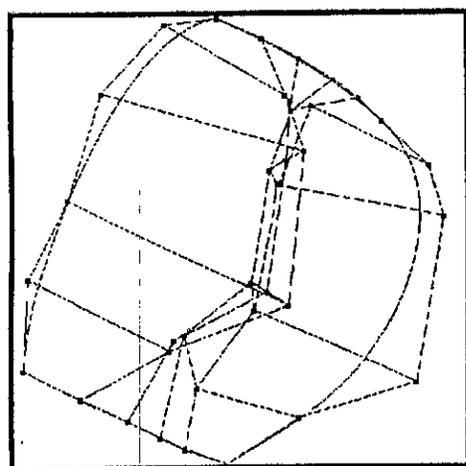


Figure 4.36. Bézier 6*6

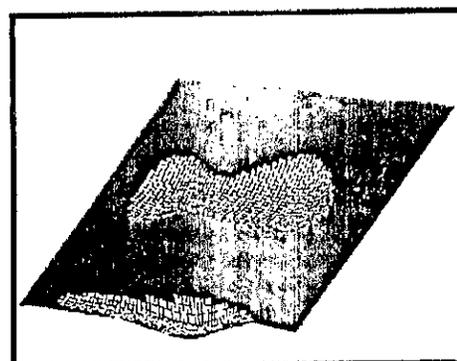


Figure 4.37. Ecart

4.6. Comparaison analytique

La méthode proposée est une méthode effective de réduction de degré des courbes et des surfaces en vue de l'échange de données entre systèmes, qui à l'avantage d'être plus rapide à celle des méthodes de conversion par approximation cité au chapitre précédents, car le processus de conversion par approximation nécessite plus temps de calcul qui dépend des courbes obtenue après un nombre fini de subdivision, l'algorithme présenté est facile à implémenter mais reste le problème de continuité qui est limité à l'ordre un (continuité de tangente).

4.7. Conclusion

Nous avons pu constater qu'il existait un paramétrage optimal permettant de calculer la meilleure approximation pour un degré définit. En cherchant à approcher ce paramétrage optimal, nous avons développé une méthode simple basée sur la minimisation progressive des écarts entre deux courbes par la modification du réseau de pôles. En réalité il s'agit d'une interpolation itérative où le paramétrage des points est modifié lors de chaque itération. Finalement la qualité des résultats est telle qu'il n'est pas nécessaire d'utiliser le paramétrage optimal puisque l'on dispose déjà de la courbe modifiée par l'algorithme de minimisation des écarts. Quelques perfectionnements ont alors été proposés pour tenir compte de contraintes propres à l'échange de données telles que la recherche du degré minimal et le respect des contraintes de continuité. Enfin, nous avons vu que la transposition au cas des surfaces était aisée mais il faudra fournir une surface initiale la plus proche possible de la surface d'origine pour que le procédé soit utilisable car les calculs sont beaucoup plus lourds que pour les courbes.

Enfin, la nature même du procédé laisse entrevoir d'autres utilisations comme l'approximation de courbes et surfaces à partir d'un nuage de points avec une évaluation automatique du paramétrage ou la transformation de surfaces quelconques en surfaces de Bézier.

CHAPITRE 5

Application à l'échange de données entre systèmes de CAO

5. Application à l'échange de données entre systèmes de CFAO .

5.1. problématique

Les systèmes de CFAO ont commencé à se répandre dans l'industrie automobile dans les années 70. A ce jour, tout le monde ou presque, s'accorde à reconnaître que la CFAO est beaucoup plus qu'un système électronique de dessin.

Il existe actuellement un nombre important de systèmes de CFAO, chacun faisant ressortir sa spécificité. Le choix d'un système CFAO obéit à des raisons stratégiques, économiques et technologiques qui varient d'une entreprise à l'autre.

Les systèmes de CAO stockent différemment leurs données et ont des concepts de modélisation différents. Il en résulte des problèmes d'échange de données. Ceci induit des coûts à prendre en compte dans tous les projets qui mettent en œuvre des échanges de données CAO entre systèmes hétérogènes.

5.2. Transmission de données

L'un des objectifs essentiels du système proposé concerne le transfert de données d'un logiciel (système de CFAO) à un autre de façon à ce qu'elles soient reçues dans la forme définie par le logiciel émetteur. Une première solution revient à développer des programmes d'interfaces spécifiques pour faire communiquer les systèmes deux à deux. Ce peut être une solution si cela résulte d'un accord entre deux constructeurs ayant une clientèle commune. Cette méthode devient vite coûteuse dès lors que le nombre de logiciels augmente : N systèmes à interfacer demandant $2.N(N-1)$ traducteurs. De plus il faut prendre en compte les problèmes de développement et de maintenance informatique des programmes en fonction de la normalisation et du logiciel auquel il se rapportent. Une deuxième solution consiste à définir un fichier intermédiaire standardisé (fichier neutre) dans lequel le système émetteur écrit et où le système récepteur vient de lire.

Le fichier neutre a un format et une définition selon la norme qu'il utilise. Dans ce cas, N systèmes à interfacer demanderont $2.N$ traducteurs.

Pour lire et écrire un fichier neutre, il faut que le logiciel soit doté de deux traducteurs. Un postprocesseur capable de lire le format d'un fichier neutre et de le traduire dans le « dialecte » de la base de données du système, et un préprocesseur capable de créer un fichier neutre à partir de cette même base de données.

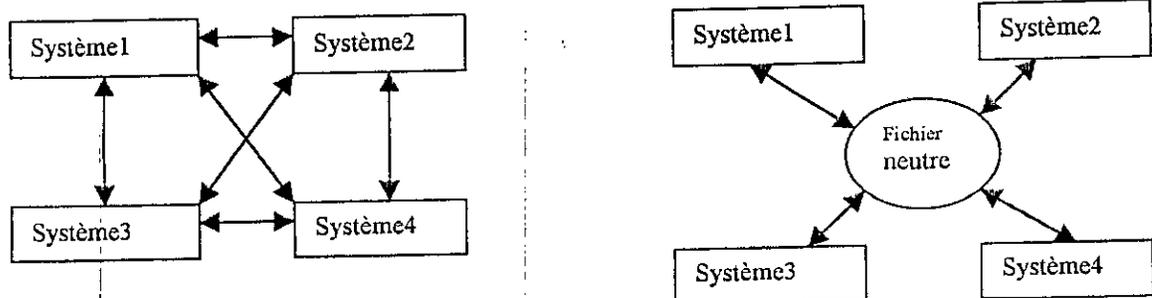


Figure 5.1. Modes de transfert de données

5.3.les principaux standards d'échanges

5.3.1.IGES

IGES, a été la première tentative réelle de définition d'un standard d'échanges de données. Elle est devenue une norme ANSI (Y14.26M) en 1981. IGES (International Graphics Exchange Specification)est actuellement la norme la plus utilisée pour le transfert des données en CFAO. IGES se veut d'un usage très général pouvant s'adapter à de multiples applications, pouvant d'écrire l'essentiel des caractéristiques géométriques et physiques d'objets manufacturés. Des améliorations y sont continuellement apportées pour ajouter de nouvelles entités en liaison avec l'évolution des logiciels. Un des inconvénients est la taille des fichiers créés. Ses avantages, consistent en une grande diversité de fonctionnalités, d'où un minimum de perte d'informations dans la plupart des applications.

5.3.1.1. Architecture

Un fichier IGES est un fichier séquentiel au format ASCII constitué d'enregistrements de 80 caractères de longueur. Il est structuré en cinq sections.

- **START SECTION.** Contient des commentaires qui peuvent être imprimés en clair.
- **GLOBAL SECTION.** Contient les informations générales sur le fichier (date de création, référence du fichier et du système émetteur, format des variables, échelle, unités), paramètres valides pour l'ensemble du fichier (épaisseur, type de traits).
- **DIRECTORY ENTRY SECTION.** Chaque entité y est définie par deux enregistrements de quatre-vingt caractères chacun. Chaque enregistrement est subdivisé en dix champs de huit caractères. Chaque champ contient une information sur la nature, les attributs de l'entité ou un pointeur sur un enregistrement de la section "parameter data" ou de la section "directory entry".
- **PARAMETER DATA SECTION.** Chaque enregistrement de cette section contient, selon le type d'entité, des caractéristiques géométriques (coordonnées, angles, coefficients de matrices, etc...), des pointeurs sur d'autres enregistrements définissant des listes de propriétés ou d'autres entités.
- **TERMINATE SECTION.** Cet enregistrement unique indique la fin du fichier. Il donne le nombre d'enregistrements de chaque section.

Pour l'ensemble des sections, un numéro de séquence permet l'utilisation de pointeurs entre les enregistrements des sections "directory entry" et "parameter data". Ce numéro figure à la fin de chaque enregistrement. Les colonnes 1 à 72 contiennent les données de l'enregistrement, la colonne 73 indique la section de l'enregistrement, les colonnes 74 à 80 contiennent le numéro de séquence.

5.3.1.2. performances et limitations

1. ne transfère que des surfaces.
2. ne transfère pas les informations sur les connexités.
3. traite les géométries filaire, surfaciques 2D et 3D.
4. traite le dessin technique.
5. traite la modélisation par éléments finis.
6. traite la schématique fluide et tuyauterie.
7. traite la schématique électrique.
8. Génère des fichiers ASCII (lignes de 80 caractères maxi).

5.3.2.VDA

VDA (Verband Der Automobilindustrie) recouvre des travaux pour l'échange de données dans le monde de l'automobile allemande. Il est ainsi plus spécifiquement dédié à l'échange de surfaces.

VDA-FS (Flächen Schnittstelle) représente la norme d'origine (DIN 66301) - courbes et surfaces, VDA-IS (Iges Subset) est un sous ensemble de IGES. Nous nous intéressons plus particulièrement à la norme VDA-FS.

C'est l'une des interfaces les plus facile à utiliser. VDA-FS contient seulement les données de surface. Par exemple, les matrices de transformation et les points sont définis, les géométries 2D et les dimensions ne le sont pas. En limitant l'étendue des données aux surfaces, VDA-FS peut éviter beaucoup de problèmes liée aux échanges. Par exemple, la source majeure de problèmes avec des données transféré est le texte et la cotation. VDA-FS ne soutient pas l'une ou l'autre de ces entités, le problème est donc soigneusement évité. LE VDA-FS est similaire dans le concept à IGES. Les deux interfaces créés des fichiers textes ASCII qui contiennent l'information (les formules mathématiques) des données CAO. Cependant, l'étendue des informations incluses est très limitée. Par exemple, tandis que l'information géométrique est incluse, l'information telle que la couche ou la couleur ne l'est pas.

5.3.2.1.Architecture

Le format des données entre VDA et IGES est très différent. IGES est une norme très rigide définie et très encombrante. Le VDA-FS est quand à lui très libre. La norme contient très peu des règles pour définir les entités

Il y a deux sections dans un fichier VDA-FS:

- **HEADER.** Elle contient l'information sur l'expéditeur (nom, numéro de téléphone, etc.), le récepteur (entreprise, individu, etc.) et sur la pièce (nom de projet, date d'entrée en vigueur, etc.). Cette information ressemble étroitement l'information contenu dans global section de IGES.
- **PART INFORMATION.** Les entités sont toutes définies de la même façon:
 - Nom.
 - Type d'entité.

<i>Type d'entité Géométrique</i>	
POINT	point
PSET	séquence de points
MDI	séquence de vecteur
CIRCLE	cercle / Arc
CURVE	Courbe
SURF	Surface
CONS	Courbe de surface
FACE	frontière de surface
TOP	Topologie

entités non-Géométrique
HEADER tête de fichier
\$\$ Commentaire
BEGINSET début de set
ENDSET fin de set
GROUP groupe
TMAT Matrice de transformation
TLIST Liste de Transformation
END FIN

- Paramètre.

L'information de paramètre est libre. Il peut prendre autant de ligne qu'il a besoin. Les paramètres individuels sont séparés par des virgules.

5.3.3.SET

SET (Standard d'Echange et de Transfert). SET est une norme AFNOR (Z68-300). Elle a été développée initialement par l'Aérospatial en 1983. Les principes sur lesquels repose son architecture sont : le fichier le plus compact possible pour une transmission rapide, une structure suffisamment simple pour diminuer les coûts de développements et faciliter la maintenance.

5.2.3.1. Architecture

Un fichier SET est un fichier séquentiel dont les enregistrements sont formés d'une suite de codes ASCII. Il est structuré sous forme d'ensembles pouvant contenir des sous-ensembles disjoints.

Chaque ensemble ou sous-ensemble est formé de blocs. Ceux-ci définissent une information de façon globale. Chaque bloc contient des sous-blocs qui décrivent avec précision une partie de l'information contenue dans le bloc auquel ils appartiennent.

Un fichier SET pourra donc avoir l'architecture suivante :

- Bloc début de SET
- Bloc début d'ensemble n° séquence 1
- suite de L blocs
- Bloc début de sous-ensemble n° séquence L + 2
- suite de N blocs
- Bloc fin de sous-ensemble n° séquence L + N + 3
- suite de M blocs
- Bloc fin d'ensemble n° séquence L + N + M + 4
- Bloc fin de SET

Les blocs sont numérotés dans un ordre chronologique (numéro de séquence). Il peut-être fait référence à tout moment à un bloc donné à l'aide d'un pointeur. Les données contenues dans les blocs sont de nature et d'un format équivalents à celles du FORTRAN. Les caractères d'en-tête d'informations utilisés sont :

@ pour les blocs

pour les sous-blocs

! pour les pointeurs sur d'autres blocs

: pour les paramètres

Le dictionnaire des paramètres contient les spécifications des attributs de définition et de représentation des entités décrites dans les blocs et les sous-blocs. Chaque paramètre est repéré par un numéro dont la signification et les valeurs possibles sont décrites dans la norme AFNOR.

5.3.3.2. performances et limitations

1. traite les géométries filaires, surfaciques 2D et 3D.
2. traite les géométries volumiques CSG (Constructive Solid Geometry) et B-Rep (Boundary Representation).
3. traite le dessin technique.
4. traite les données scientifiques.
5. traite les éléments finis : modélisation et analyse.
6. traite la schématique électrique.
7. normalisé (Z68-300) en août 85 puis mis à jour en juin 89.

5.3.4. STEP

STEP (Standard for the Exchange of Product Model Data) a pour but de fédérer l'ensemble des approches, avec des objectifs plus larges que le simple échange de données : en effet, STEP cherche à fournir un mécanisme pour représenter toutes les données associées à un produit, en tenant compte de la totalité de son cycle de vie.

STEP est organisé en trois couches :

La couche logique : elle doit contenir le modèle conceptuel des données du produit. décrit à partir d'un langage de définition des données (EXPRESS). Un tel modèle comprend :

- les données "fondamentales" : définition, décomposition, structure ...,
- les données géométriques et topologiques,
- cession des représentations : liaison avec un modeleur géométrique ...,
- structure : gestion nomenclature ...,
- caractéristiques de formes....
- tolérancements ...

La couche physique : c'est le niveau mise en œuvre, avec un format neutre de fichier d'échanges. le SGBD et l'interface programmation.

La couche application : elle définit des protocoles d'application dépendant du domaine. Parmi les protocoles en cours de définition, on trouve un protocole pour le dessin, un protocole pour l'analyse par éléments finis ...

L'initiative AIT (Advanced Information Technology in Design & Manufacturing) regroupe les principales entreprises européennes des secteurs aéronautique, automobile et informatique. Elle a pour but de développer des méthodes, des systèmes et des outils intégrés pour utiliser pleinement et efficacement les technologies de l'information dans la production et l'organisation industrielles. Elle s'appuie notamment sur STEP et renforce son essort.

STEP est toujours en cours de définition et représente le standard de l'avenir.

5.4.DXF

DXF(Drawing Exchange File) est un format d'échange en ASCII , à partir de sa définition par Autodesk comme format de fichier pour Autocad . Il est devenue un standard .

5.4.1.Architecture

chaque fichier DXF (à partir de R12) comporte 3 parties dont la première désignée par HEADER représente l'en tête qui comporte l'état et le type du logiciel ayant créé le fichier, on trouvera la version de format et l'initialisation des variables externes du logiciel notamment les types de traits , couleurs , dimension de la fenêtre de travail , nombre de couche , le style et les dimensions des caractères , etc., la seconde partie désignée par TABLES définit les fenêtres , et tous les paramètres format à activer (comme le type de trait ,d'écriture , etc.), la troisième partie comporte les entités géométriques du dessin . dans la partie BLOCKS sont alors définies les types d'entités , dans la quatrième partie appelé ENTITEES sont données les caractéristiques définissant entièrement l'entité . ce qui donne la structure globale suivante

Sections	Sub-sections	objets	Groupes
EN TETE		variables	Groupes
LISTES Groupes	TYPE	Entrées	Groupes
	COUCHE	Entrées	Groupes
	STYLE	Entrées	Groupes
BLOCKS	...	Entités	Groupes
ENTITEES		Entités	Groupes

FIN

Chaque fichier DXF commence par les caractères suivant

0

SECTION

2 {début de fichier et de section}

Chaîne1 {chaîne de caractère définissant la section . Peut prendre les affectations :
HEADER , TABLES, BLOCKS , et ENTITIES}

Et se termine par

0

EOF {fin de fichier}

Chaque section se termine par

0

ENDSEC

Pour la définition des entités , la chaîne1 prend l'affectation ENTITEES et sera suivie de

0

chaîne2 {chaîne de caractères définissant l'entité. Peut prendre les affectations :
POINT, LINE CIRCLE ARC POLYLIGNE , ... }

à la fin de la définition de chaque entité on trouve

SEQEND

Pour définir une courbe B-spline uniforme non rationnelle , on affecte POLYLIGNE à la chaîne2 qui sera suivie de

8 {valeur de début de groupe suivie de la largeur du trait }

66 { définition du type de trait }

10 {suivi de la valeur de la coordonnée suivant X du premier point }

20 {suivi de la valeur de la coordonnée suivant Y du premier point }

30 {suivi de la valeur de la coordonnée suivant Z du premier point }

70 {suivi de « 4 » pour définir le type polyligne , de « 8 » pour un point , ou de
« 16 » pour un pôles }

75 {suivi du degré quand il s'agit d'une courbe ou de la valeur 0 quand il s'agit
d'un polygone }

0 {fin de groupe }

suivi immédiatement par les points et les pôles de la courbe définis comme suit

VERTEX

8

10 X

20 Y

30 Z

70 suivi de « 8 » pour les points et de « 16 » pour les pôles

0

5.5.Application

Le format DXF-R14 est le plus approprié pour un échange de données ,mais dans le but de faire une application large pour l'échange de données entre Autocad , ConverCurves(application autonome), nous avons opté le format neutre DXF-R12 commun aux deux

logiciels . Ce qui permis de réaliser un pré-processeur limité aux courbes(extensibles aux surfaces) permettant de transformer le format DXF-R12 en format DES du logiciel ConverCurves , ainsi qu'un post-processeur limité aux courbes (extensibles aux surfaces) permetant de transformer le format DES en format DXF-R12 . nous obtenons ainsi la structure de changement de format ci-dessous (fig.5.2)

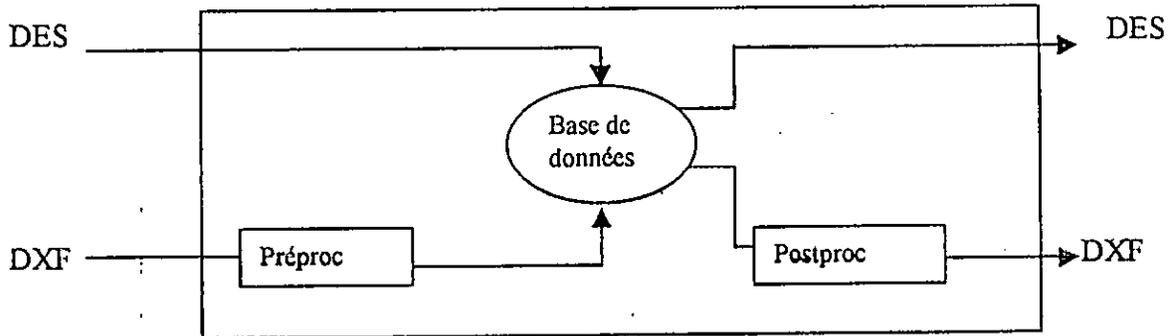


Figure 5.2. Structure de changement de format pour ConverCurves

Qui permet de réaliser la transmission selon le schéma directeur ci-dessous (figure 5.3)

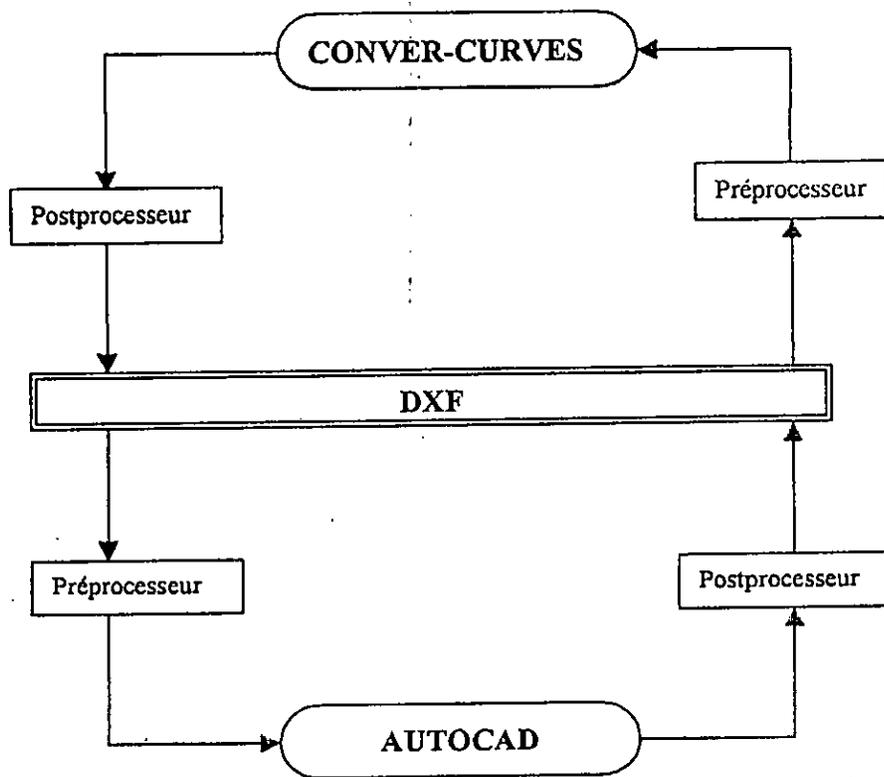


Figure 5.3. Possibilités de transfert de données à l'aide du système élaboré CONVER-CURVES

Dans le but d'une transmission de données entre AUTOCAD ↔ CONVER-CURVES, on peut transmettre une courbe de degré 25 exécutée par exemple sous CONVER-CURVES, qui supporte des courbes plus de 40 degrés, vers AUTOCAD en utilisant le module de conversion par approximation, sachant que logiciel AUTOCAD ne supporte pas les courbes B-spline et Bézier de degré supérieure à 5.

Les possibilités de passage AUTOCAD ↔ CONVER-CURVES pour les courbes est possible dans les deux sens, les transformations suivantes (Tableau 5) sont alors possibles

CONVER-CURVES	TRANSFORMATION	AUTOCAD
Bézier de degré >2	Réduction de degré, conversion par approximation.	Bézier de degré 3
B-spline de degré >2	Réduction de degré, conversion par approximation.	B-spline de degré 3
B-spline non uniforme de degré >2	Insertion de nœuds. Suppression de nœuds	B-spline uniforme de degré 3
B-spline ou Bézier de degré >2	Réduction de degré. Conversion par approximation. Insertion de nœuds. Suppression de nœuds. Subdivision. Transformation de base	B-spline ou Bézier transformée de degré 3

Tableau 5. Transformation possibles pour un transfert de données.

5. 6. Programme mis en œuvre

5.6.1. OpenGL : Présentation

1) Qu'est-ce qu'OpenGL ?

OpenGL est une bibliothèque graphique 3D très évoluée et totalement portable offrant de nombreuses ressources aux programmeurs cherchant à élaborer un moteur 3D (pour les jeux par exemples).

2) OpenGL et Direct 3D

Voici quelques comparaisons choisies à tout hasard entre les deux produits de Microsoft , OpenGL et Direct 3D :

OpenGL est totalement portable et peut tourner sous tous les systèmes d'exploitations dignes de ce nom	Direct3D ne tourne que sous Windows .
OpenGL est 10 fois plus rapide que Direct3D .	Direct3D est 10 fois plus lent que OpenGL .
OpenGL fournit des outils de rendu très évolué (mapping d'environnement, effets atmosphériques...)	Les fonctions de Direct3D sont très primitives (pour preuve, Direct3D ne fournit le mapping environnemental qu'à partir de la version 7.0, contre 1.0 pour OpenGL)
OpenGL est d'une utilisation très simple et intuitive, basée sur un système de primitives, donc très pratique pour le programmeur	Direct3D est exagérément orienté objet, avec des pointeurs dans tous les sens, et est donc une torture pour le programmeur
OpenGL est tout à fait stable .	Direct3D est instable .
OpenGL est disponible partout, gratuit, et dispose d'une bonne documentation	pour programmer avec Direct3D, il faut télécharger le SDK, qui fait quand même 125Mo, ou bien l'acheter

et OpenGL est le seul vrai bon standard en matière de 3D, puisqu'il permet des effets complexes et d'une grande facilité pour le programmeur , c'est pourquoi nous avons choisir de programmer avec OpenGL avec Visuel C++ .

OpenGL est axé selon un principe de primitives : chaque objet est composé d'éléments basiques comme les sommets, les faces, etc..., appelés primitives. Pour créer un objet, il suffit donc de lui indiquer toutes ses primitives. Par exemple, une face carrée sera générée de cette façon :

```
glBegin(GL_QUADS);
glVertex3f(1.0,1.0,0.0);
glVertex3f(0.0,1.0,0.0);
glVertex3f(0.0,0.0,0.0);
glVertex3f(1.0,0.0,0.0);
glEnd();
```

5.6.1.3. OpenGL : Récapitulation

Nous pouvons dire qu'OpenGL est carrément portable est rapide, que l'on crée des objets à partir de leurs primitives, et qu'un gros un programme dessinant en OpenGL se structure à peu près comme ça :

- Création d'une fenêtre
- Récupération d'un Device Context pour cette fenêtre
- Récupération d'un Rendering Context pour ce Device Context
- Initialisation d'OpenGL à l'aide de ce Rendering Context
- ...
- Création d'objets à partir de primitives simples
- Affichage des objets
- ...
- Libération du Rendering Context
- Libération du Device Context
- Fermeture du programme

5.6.2. Les différents modules du système élaboré

le programme élaboré se constitue d'une interface graphique en deux dimensions, Réalisé à l'aide des fonctions OpenGL.

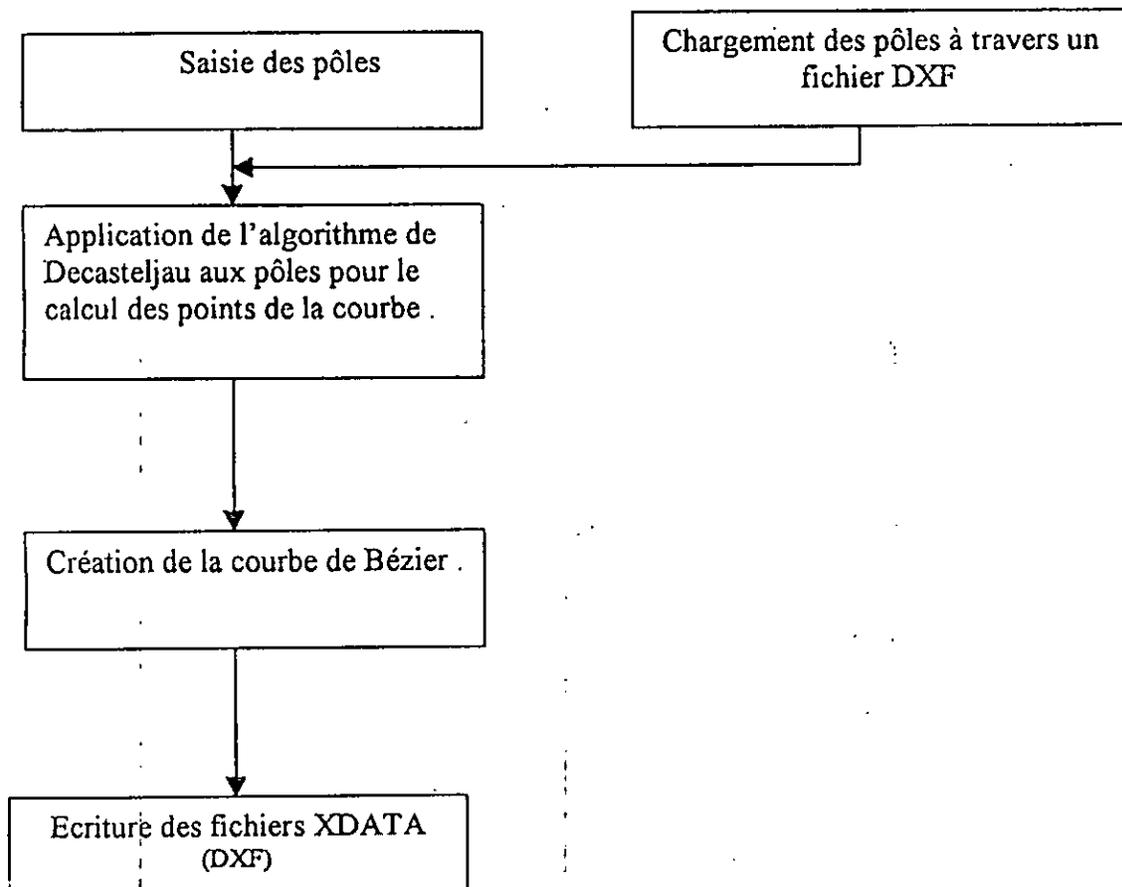
le programme et une application sous Windows, réalisé à l'aide du langage de programmation C++.

on a une souplesse de manipulation, car on peut dessiner un courbe de Bézier de degré quelconque, avec différent transformation: subdivision, réduction(avec les méthodes présentés), élévation de degré.

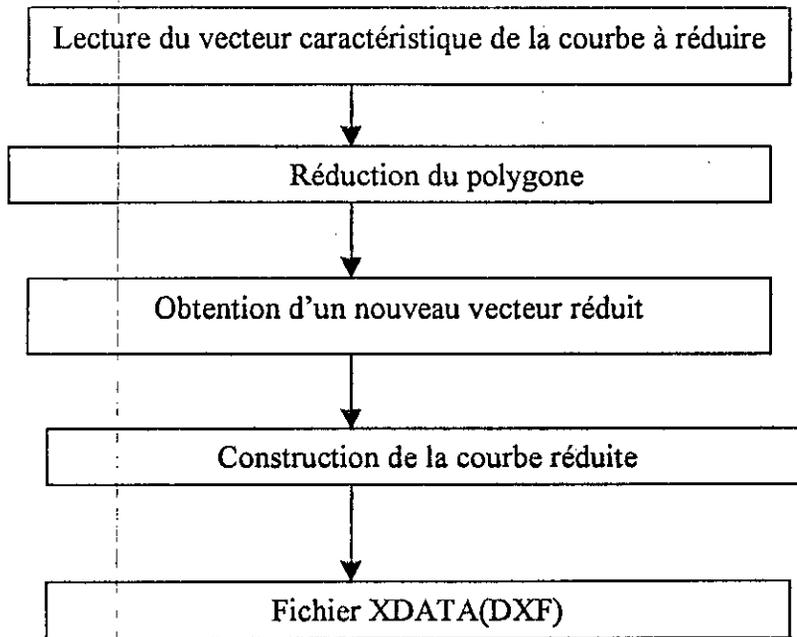
aux niveaux d'entrée et sortie I/O, l'application permet de charger ou bien de stoker des fichiers de format DXF.

les différents modules constituant le système proposé sont présentés comme suit :

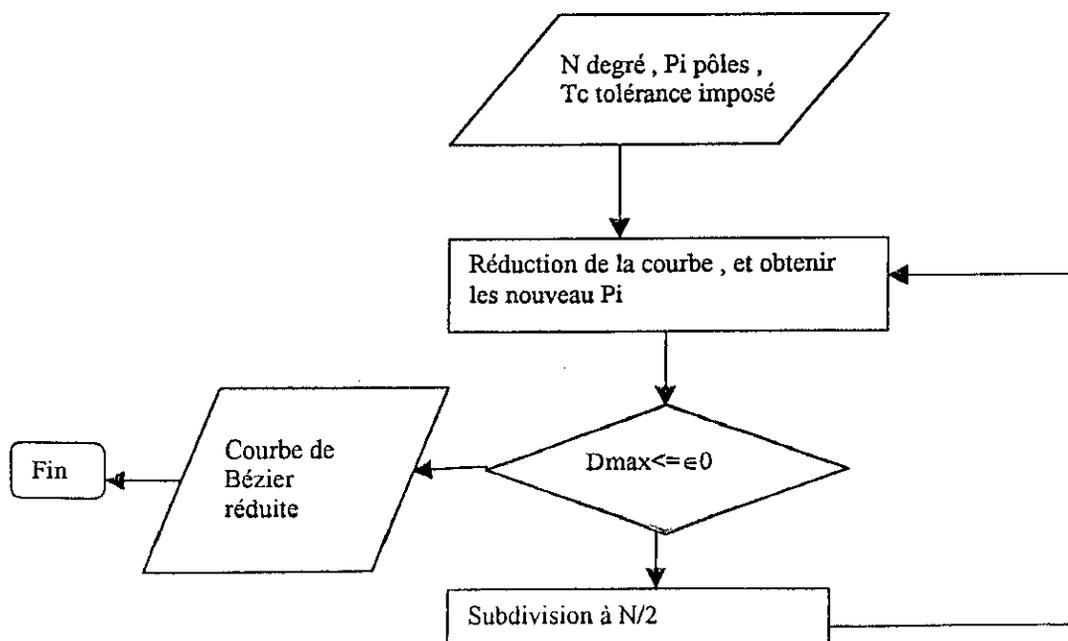
5.6.2.1. Module de construction de courbes



5.6.2.2. Module de réduction



5.6.2.3. Module de conversion



Conclusion

Conclusion

Ce travail nous a amené à établir une synthèse bibliographique très approfondie. Les fondements mathématiques des courbes et surfaces de Bézier et B-spline sont présentées de façon claire avec les différentes approches analytiques ainsi les algorithmes de base concernant leur construction.

Les différentes méthodes de réduction de degré des courbes de Bézier non rationnelles sont présentées en détail, en exposant pour chacune d'elle d'abord le principe général suivi des algorithmes correspondants sous forme de procédures très simple et prête à une implémentation adaptée au langage courant de programmation.

A la lumière de cette étude comparative effectuée sous forme algorithmique et expérimentale, nous avons conclu d'une part que le problème de la meilleure approximation demeure toujours posée, d'autre part les résultats obtenus de notre étude montre que la méthode de réduction de degré des courbes de Bézier non rationnelles proposée par Bensalah a l'avantage d'être rapide et présente une facilité de transformation remarquable pour le passage à la réduction de degré des surfaces de Bézier non rationnelles, mais produit une erreur très élevée et un nombre de courbe réduite très grand, sans assurer la continuité au-delà de l'ordre un, à l'opposé de l'algorithme de Forrest qui assure la continuité géométrique maximale aux extrémités.

Cependant cette méthode de réduction induit une erreur d'approximation très élevée qui donne par conséquent un nombre de courbe réduite très élevée. L'inconvénient majeur de la troisième méthode présentée par Watkins et Worsey réside au niveau de la continuité géométrique qui n'est assurée qu'à l'ordre zéro, et le temps de calcul est tout même raisonnable.

En se basant sur la notion de meilleure approximation inspirée du théorème de l'alternation, la méthode de Eck a été développée dans ce sens pour donner une meilleure approximation ainsi une continuité géométrique maximale, à cause de la contrainte de la continuité géométrique qui n'est pas respectée aux extrémités, l'auteur intervient sur la modification des pôles extrêmes d'où la notion de meilleure approximation est perdue, mais cette méthode reste la plus rapide, assurant une continuité géométrique imposée pour une erreur de position minimale, toutefois cette méthode nécessite une approximation en plus qui fait intervenir un algorithme spécifique pour le calcul des coefficients de pondération qui déterminent les nouveaux pôles de la courbe réduite.

L'approximation utilisée par la méthode de Hoschek est une approximation au sens des moindres carrées, en se basant sur la technique de Paramétrisation intrinsèque qui donne une bonne approximation de la courbe réduite, l'algorithme de réduction de degré nous amène à la résolution d'un système d'équations non linéaire dont la résolution nécessite un grand temps de calcul. L'avantage de cette méthode réside dans la possibilité qu'elle offre d'assurer un

ordre de continuité géométrique imposé à priori par l'utilisateur et la facilité de passage aux surfaces et d'autres modèles rationnels.

Les résultats de cette étude montrent que les méthodes de réduction de degré des courbes de Bézier non rationnelles comportent encore des imperfections qu'il est possible d'améliorer, ce qui nous a conduit à proposer une méthode simple et efficace basée sur la minimisation des écarts normaux entre deux courbes (l'une est la courbe originale et l'autre est la courbe réduite en utilisant l'algorithme de Eck qui donne une première approximation de la courbe), afin d'approcher du paramétrage optimal.

L'avantage de cette méthode réside dans le temps de calcul ainsi que la précision obtenue après l'application du processus de minimisation, car cette méthode n'utilise pas le principe de conversion par approximation dont le temps de calcul dépend de l'erreur commise lors d'une réduction de degré, ainsi le nombre de courbes obtenues après chaque subdivision qui sont à leurs tours réduites.

L'inconvénient de cette méthode réside dans la continuité géométrique, par conséquent on peut envisager une étude ultérieure en mettant en considération la continuité géométrique ainsi le degré de réduction maximal pour la réduction de degré des courbes de Bézier et B-spline rationnelles et non rationnelles en vue de l'échange de données entre systèmes de C.F.A.O..

Les algorithmes présentés dans ce travail peuvent être combinés et généralisés à des applications pratiques en C.F.A.O., telles que le passage d'une forme de Bézier de degré élevé à une forme B-spline de degré faible, passage d'une forme B-spline non uniforme à une forme uniforme et inversement..., ce qui constituera une richesse des fonctionnalités des logiciels industriels au niveau de la flexibilité de conception de formes et la communication entre systèmes basés sur des standards d'échange plus performants et normalisés.

Referencias Bibliográficas

Références bibliographiques

- [Asma'99] : F.ASMA
Etude comparative des méthodes réduction de degré des courbes gauches, mémoire de magister, IGM 1999.
- [GJA'99] : G.WAHU, J.M.BRUN, A.BOURAS
Réduction de degré de courbes et surfaces de Bézier en vue de l'échange de données, LGIM, LYON, 1999.
- [JSA'95] J.M.BRUN, S.FOUFOU, A.BOURAS
Study of iterative processes in computer graphic's problems, LIGIA, LYON 1995.
- [JSA'96] J.M.BRUN, S.FOUFOU, A.BOURAS
In search of an optimal parametrization of curves, In Proc. Of the fourth Int'l conf. On Computational Graphics and visualization techniques, Lisbon, 11-15 Dec 1995.
- [Farin'92] : G.FARIN
Courbes et surfaces pour la CGAO, édition MASSON 1992.
- [TRAN'97] M. TRANCNAM, Hybrid Symbolic-Numerical Methods for Computer Aided Geomtric Design. (Lecture Notes SS'97), Johannes Kepler University, A-4040Linz, Austria 1997.
- [GBB'96] T. Schreiber G. Brunnet and J. Braun.
The geometry of optimal degree reduction of Bézier curves.
C.A.G.D., 13:773-788, 1996.
- [Hos'88] J. Hoschek. Intrinsic parametrization for approximation. C.A.G.D., 5:27{31, 1988.
- [JBA'95] A. Bouras J.M. Brun, S. Foufou and P. Arthaud.
In search of an optimal parametrization of curves.
In Proc. of the fourth Int'l Conf. on computational graphics and visalization techniques, Lisbon, 11-15 Dec 1995.
- [Lac'88] M. Lachance. Chebyshev economization for approximation.
C.A.G.D., 5:195-208, 1988.
- [Lee'84] E. Lee. Choosing nodes in parametric curve interpolation.
C.A.D., 21(6):363-370, 1984.
- [PBY'96] E.W. Stanley P. Bogacki and X. Yuesheng.
Degree reduction of Bézier curves by uniform approximation with endpoint interpolation.
C.A.D., 27(9):27{31, 1996.
- [Wah'99] G. Wahu. Critical criteria for data reduction in curve and surface models conversion.
In Proc. of the first Swiss Conf. of CAD/CAM, Neuchâtel, 22-24 Feb 1999.
- [WW'88] M.A. Watkins and A.J. Worsey.
Degree reduction of Bézier curves.
C.A.D., 20(7):398-405, 1988.

- [Colm'99] : Colm Mulcachy ,the Basic Curves and Surfaces of Computer Aided Design ,Depeatment of Mathematics ,Spelman College ,Atlanta ,USA.(1998).
- [De Boor 20] Carl De Boor
B(asic)-spline Basics,United States Army 2000 .
- [GJ'98] : G.DEMENGEL et J.P.POUGET
Modèles de Bézier des B-splines et des NURBS, Mathématiques des courbes et des surfaces , Edition ellipses 1998.
- [Eck'95] Mathias Eck and Jan Handenfeld
Local Energy Fairing of B-spline curves ,
Department of Computer Science and Engineering ,1995,Fr-35,University of Waschington, USA.
- [Handen '95] Jan Handenfeld .
Fairing of B-spline curves and surfaces ,Technische Hochschule Darmstadt,1995.
- [Boehm&Hoschek&Hans] Wolfgang Boehm ,Josef Hoschek, Hans-Peter Seidel,
Mathematical Aspects for Computer Aided Geometric Design , Tchnische hochschule Darmstadt.
- [Pierre'72] P. J. Laurent, Approximation et optimisation, edition HERMAN Paris1972.
- [Eck'93] M.ECK
Degree rdution of Bézier curves ,Computer Aided Geometric Design, N°10,
1993 pp237-251
- [Eck'95]M .ECK
Least squares degree reduction of Bézier curves , Computer Aided Geometric Design, Vol27 N°11 Nov 1995.
- [Forrest'72]A.R.Forrest
Interactive interpolation and approximation by Bézier polynomial , Computer Journal, Vol 15, 1972 pp 71-79.
- [Forrest'90]A.R.Forrest.
Interactive interpolation and approximation by Bézier polynomial, Computer Aided Design, Vol22, N°9 Nov1990 pp 527-537
- [Hos'88]J.HOSCHEK
Intrinsic parametrization for approximation, Computer Aided Geometric Design N°5 1988 pp 27-31
- [Hoschek & Wissel'88]J.HOSCHEK and F.J.WISSEL
Optimal approximate conversion of B-spline surfaces, Computer Aided Geometric Design, N°6 1989 pp 293-306
- [Hoschek'90].J.HOSCHEK
Exact approximate conversion of spline curves and spline surfaces , Computation of curves and surfaces , W.Dahmen et al 1990, pp 73-116
- [Lachance'88]M.A.LACHANCE
Chebychev economisation for parametric surfaces , Computer Aided Geometric Design N°5 1988 pp 195-208
- [Watkins & Worsey'88] M.A.WATKINS and A.J.WORSEY
Degree reduction of Bézier curves . Computer Aided Geometric Design ,Vol 20, N°7 Sept 1988 pp 398-405
- [Piegl & Tiller'95]L .PIEGL and W.TILLER
Algorithm for degree reduction of B-spline curves , Computer Aided Geometric Design, Vol 27 N°2 Feb 1995 pp 101-110

[Salbonnière '87] P.SALBONNIERE

courbes B-splines , Mathématiques , informatique et applications , CNRS INSA de RENNES Mai 1987 pp 19-47

[Colm'99] : Colm Mulcachy ,the Basic Curves and Surfaces of Computer Aided Design ,Deptement of Mathematics ,Spelman College ,Atlanta ,USA.(1998) .

[Eck'95] : Mathias Eck and Jan Handenfeld, Local Energy Fairing of B-spline curves ,Department of Computer Science and Engeneering ,1995,Fr-35,University of Waschington, USA.

[Hand '95] :Jan Handenfeld , Fairing of B-spline curves and surfaces ,Technische Hochschule Darmstadt,1995.

[Boehm&Hoschek&Hans] : Wolfgang Boehm ,Josef Hoschek, Hans-Peter Seidel, Mathematical Aspects for Computer Aided Geometric Design , Technische hochschule Darmstadt.

[Pierre'72] :Pierre – Jean Laurent , Approximation et optimisation , Edition HERMAN Paris1972.

[Farin'2000] :G.FARIN

Least squares approximation, computer science and engeneering, Arizona State University, October 20, 2000.

[Farin'99] G.FARIN

Least squares B-spline approximation, computer science and engeneering, Arizona State University, October 27, 1999.

[WJH'96] W.BOEHM and J. HOSCHEK and H.P.SEIDEL

Mathematical Aspects of Computer Aided Geometric Design
Fachbereich Mathematik ,Technische Hochschule Darmstadt 1996.

[Hoschek&Pottmann'95]J.HOSCHEK and H.POTTMANN

Interpolation and Approximation With Developable B-spline surfaces, Mathematical Methods for Curves and Surfaces, pp 255-261 1996.

ANNEXES

A. Notions de base

1. Espace affine

L'espace dans lequel nous décrivons notre objet ne possède pas un système de coordonnées privilégié – nous avons à le définir nous même. Notre intérêt est dans l'objet et non dans sa relation à quelques systèmes de coordonnées arbitraire ; les méthodes que nous développerons doivent donc être indépendantes du choix particulier d'un système de coordonnées .

Les éléments de l'espace affine E^3 peuvent seulement être soustraits le résultat de cette opération est un vecteur . ils ne peuvent pas être additionnés . cette opération n'est pas définie pour des points (mais elle définit pour des vecteurs).

2. Points et vecteurs

bien que les points et les vecteurs soient tout décrit comme des triplet de nombres réels , nous insistons ici sur la distinction claire entre ces deux notions : pour chaque couple de points a, b il existe un vecteur v unique allant de a à b . il est calculé par soustraction des coordonnées :

$$v = b - a; \quad a, b \in E^3 \quad v \in \mathcal{R}^3.$$

d'un autre côté , étant donné un vecteur v , il y a une infinité de couples a, b tels que $v = b - a$. car si a, b un de ces couples et w est un vecteur arbitraire , alors $a+w, b+w$ est un autre de ces couples puisque $v = (b+w) - (a+w)$ également .

3. Enveloppe convexe

3.1 Combinaison convexe

définition : Ce sont des combinaisons barycentriques ou les coefficients α_i des points sont non seulement égaux à 1, mais sont également tous positifs , une combinaison convexe de points est toujours à l'intérieur de ces points .

3.2 Enveloppe convexe

définition : c'est l'ensemble qui est formé par toutes les combinaisons convexe d'un ensemble de points.

4. Cartes affines

la plupart des transformations qui sont utilisées pour modifier la position ou l'échelle d'un objet dans le graphisme informatique ou dans un environnement de CAO sont appelées des cartes affines .

définition : une carte ϕ qui applique E^3 sur lui même est appelé carte affine si elle laisse les combinaisons barycentriques invariantes.

Alors si $x = \sum \alpha_j a_j; \quad x, \alpha_j \in E^3$

Et si ϕ est une carte affine , alors

$$\phi x = \sum \alpha_j \phi a_j; \quad \phi x, \phi a_j \in E^3$$

avec $\phi x = A.x + v \quad v \in \mathcal{R}^3$ (vecteur) A (matrice)

5. interpolation linéaire par morceaux

soit $b_0, b_1, \dots, b_n \in E^3$ formant un polygone B . un polygone consiste en une suite de segments de droite, chacun étant généré par l'interpolation entre deux points b_i, b_{i+1} . il est par conséquent également appelé l'interpolant linéaire par morceau $P\ell$ aux points b_i . Si les points b_i sont sur une courbe c , alors B est dite un interpolant linéaire par morceau de c , et nous écrivons

$$B = P\ell c$$

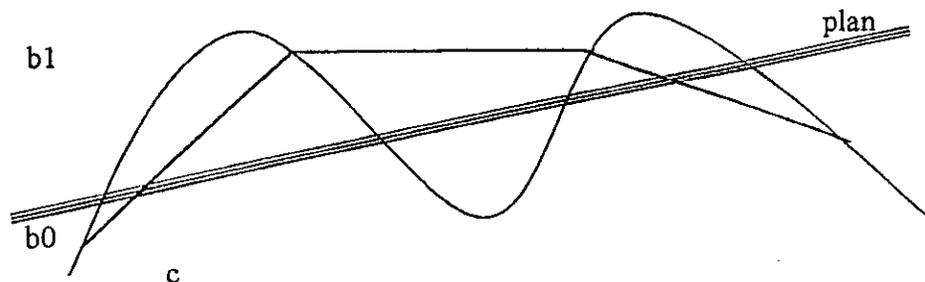
Une des propriétés importantes de l'interpolation linéaire par morceaux est l'invariance affine.

Une autre propriété est la propriété de diminution de variation : considérons une courbe continue c et un interpolant linéaire par morceaux $P\ell c$, et également un plan arbitraire. Soit $\text{crois}(c)$ le nombre d'intersection que c a avec ce plan, et soit $\text{crois}(P\ell c)$ le nombre d'intersections que l'interpolant linéaire par morceaux a avec ce plan :

Alors nous avons toujours

$$\text{crois}(P\ell c) \leq \text{crois}(c)$$

cette propriété découle d'une simple observation : considérons deux points b_i, b_{i+1} . Le segment de droite qui les réunit ne peut traverser un plan donné qu'en un point au plus, tandis que le segment de courbe c qui les relie peut traverser le même plan en un nombre arbitrairement grand de points. cette propriété de diminution de variation est illustrée figure 1.



B.1. continuité géométrique des courbes

Cependant, au lieu de regarder la continuité d'une paramétrisation spécifique (continuité paramétrique) nous devons regarder la continuité de la forme de la courbe qui est indépendante d'une reparamétrisation donnée. Donc nous disons que deux courbes joignent avec une continuité géométrique G^k , si les deux courbes sont paramétriquement continues sous la reparamétrisation.

Dans d'autres mots, deux courbes F et G sont G^k continûment à un paramètre s s'il existe une reparamétrisation ϕ (à laquelle conserve l'orientation donnée de la courbe) tel que $F(\phi(u))$ et $G(u)$ sont C^k continûment à s .

on utilisons la règle des dérivées composées, il est vu facilement que la continuité géométrique G^k est équivalent aux β_k contraintes suivantes :

$$G'(s) = \beta_1 \cdot F'(s)$$

$$G''(s) = \beta_1^2 \cdot F''(s) + \beta_2 \cdot F'(s)$$

$$G'''(s) = \beta_1^3 \cdot F'''(s) + 3 \cdot \beta_1 \cdot \beta_2 \cdot F''(s) + \beta_3 \cdot F'(s)$$

$$G^{(k)}(s) = \sum_{j=0}^k \sum_{\substack{i_1+i_2+\dots+i_k=j \\ i_1+2i_2+\dots+ki_k=k}} \frac{k!}{i_1!(1!)^{i_1} \dots i_k!(k!)^{i_k}} \cdot \beta_1^{i_1} \dots \beta_k^{i_k} \cdot F^{(j)}(s)$$

où

$\beta_k = \phi^{(k)}(s)$ et $\beta_1 > 0$ puisque ϕ préserve une orientation donnée, les paramètres β_i sont des paramètres de forme.

B.2. Théorème d'approximation de Weierstrass

Voici une version de ce théorème dans le contexte des courbes paramétrées.

Soit C une courbe continue définie sur $[0,1]$

Pour n fixé, nous pouvons échantillonner C pour les valeurs de paramètre i/n . Les points $C(i/n)$ peuvent maintenant être interprétés comme le polygone de Bézier d'une courbe polynomiale X_n :

$$X_n = \sum_{i=0}^n C\left(\frac{i}{n}\right) \cdot B_i^n(t)$$

On dit que X_n est l'approximation de Bernstein-Bézier de degré n à C .

Si nous augmentons la densité de nos échantillons c'est à dire, si nous augmentons n , nous pouvons alors former une suite d'approximation X_n, X_{n+1}, \dots . Le théorème d'approximation de Weierstrass affirme que cette suite de polynômes converge vers la courbe C , c'est à dire que :

$$\lim_{n \rightarrow +\infty} X_n(t) = C(t)$$

Ce théorème qui permet d'approcher une courbe donnée par des polynômes est cependant difficile à appliquer dans la pratique car pour avoir une bonne approximation, il faut

Annexe B

choisir n autour des milliers ou même des millions pour certaines fonctions. L'intérêt de ce théorème est donc d'une nature plus théorique. Il montre que chaque courbe peut-être approchée d'une manière arbitrairement proche d'une courbe polynomiale.

k=3 →

(59.c)

$$Q_3 = P_0 + (P_3 - P_2) \cdot \lambda_1^3 \cdot \omega_2 \\ + (P_2 - P_1) \cdot \left[\lambda_1^3 \cdot \left(-2 \cdot \omega_2 + 3 \cdot \omega_1^2 \cdot \frac{m-1}{m-2} \right) + \lambda_1^2 \cdot \omega_1 \cdot \left(3 - 6 \cdot \frac{m-1}{m-2} \right) \right. \\ \left. + 3 \cdot \lambda_1 \cdot \lambda_2 \cdot \omega_1 \cdot \frac{m-1}{m-2} \right] + (P_1 - P_0) \cdot \lambda_3.$$

$$Q_{m-3} = P_n + (P_{n-3} - P_{n-2}) \cdot \mu_1^3 \cdot \omega_2 \\ + (P_{n-2} - P_{n-1}) \cdot \left[\mu_1^3 \cdot \left(-2 \cdot \omega_2 + 3 \cdot \omega_1^2 \cdot \frac{m-1}{m-2} \right) + \mu_1^2 \cdot \omega_1 \cdot \left(3 - 6 \cdot \frac{m-1}{m-2} \right) \right. \\ \left. + 3 \cdot \mu_1 \cdot \mu_2 \cdot \omega_1 \cdot \frac{m-1}{m-2} \right] + (P_{n-1} - P_n) \cdot \mu_3.$$

k=4 →

(59.d)

$$Q_4 = P_0 + (P_4 - P_3) \cdot \lambda_1^4 \cdot \omega_3 + (P_3 - P_2) \cdot \{ \lambda_1^4 \cdot a_{40} + \lambda_1^3 \cdot a_{30} + \lambda_1^2 \cdot \lambda_2 \cdot a_{21} \} \\ + (P_2 - P_1) \cdot \{ \lambda_1^4 \cdot a_{400} + \lambda_1^3 \cdot a_{300} + \lambda_1^2 \cdot b_{200} + \lambda_2^2 \cdot b_{020} + \lambda_1^2 \cdot \lambda_2 \cdot b_{210} \\ + \lambda_1 \cdot \lambda_2 \cdot b_{110} + \lambda_1 \cdot \lambda_3 \cdot b_{101} \} + (P_1 - P_0) \cdot \lambda_4$$

 $Q_{m-4} =$

$$P_n + (P_{n-4} - P_{n-3}) \cdot \mu_1^4 \cdot \omega_3 + (P_{n-3} - P_{n-2}) \cdot \{ \mu_1^4 \cdot a_{40} + \mu_1^3 \cdot a_{30} + \mu_1^2 \cdot \mu_2 \cdot a_{21} \} \\ + (P_{n-2} - P_{n-1}) \cdot \{ \mu_1^4 \cdot a_{400} + \mu_1^3 \cdot a_{300} + \mu_1^2 \cdot b_{200} + \mu_2^2 \cdot b_{020} + \mu_1^2 \cdot \mu_2 \cdot b_{210} \\ + \mu_1 \cdot \mu_2 \cdot b_{110} + \mu_1 \cdot \mu_3 \cdot b_{101} \} + (P_{n-1} - P_n) \cdot \mu_4$$

$$\text{avec } \omega_1 = \frac{m \cdot (n-1)}{n \cdot (m-1)}, \quad \omega_2 = \frac{m^2 \cdot (n-1) \cdot (n-2)}{n^2 \cdot (m-1) \cdot (m-2)}, \quad \omega_3 = \frac{m^3 \cdot (n-1) \cdot (n-2) \cdot (n-3)}{n^3 \cdot (m-1) \cdot (m-2) \cdot (m-3)}$$

$$a_{40} = 6 \cdot \frac{m-1}{m-3} \cdot \omega_1 \cdot \omega_2 - 3 \cdot \omega_3, \quad a_{30} = \left(4 - 12 \cdot \frac{m-1}{m-3} \right) \cdot \omega_2, \quad a_{21} = 6 \cdot \frac{m-1}{m-3} \cdot \omega_2$$

$$b_{400} = 3 \cdot \omega_1 - 16 \cdot \frac{m-1}{m-3} \cdot \omega_1 \cdot \omega_2 + 15 \cdot \frac{(m-1)^2}{(m-2) \cdot (m-3)} \cdot \omega_1^3$$

$$b_{300} = 12 \cdot \frac{m-1}{m-2} \cdot \omega_1^2 - 24 \cdot \frac{m-1}{m-3} \cdot \omega_2 - 8 \omega_2 - 36 \cdot \frac{(m-1)^2}{(m-2) \cdot (m-3)} \cdot \omega_1^2$$

$$b_{200} = \omega_1 \cdot \left(6 + 12 \cdot \frac{m-1}{m-3} - 24 \cdot \frac{m-1}{m-2} + 12 \cdot \frac{(m-1)^2}{(m-2) \cdot (m-3)} \right)$$

$$b_{020} = 3 \cdot \frac{(m-1)^2}{(m-2) \cdot (m-3)} \cdot \omega_1, \quad b_{210} = 18 \cdot \frac{(m-1)^2}{(m-2) \cdot (m-3)} \cdot \omega_1^2 - 12 \cdot \frac{m-1}{m-3} \cdot \omega_2$$

$$b_{110} = 12 \cdot \omega_1 \cdot \left(\frac{m-1}{m-2} - \frac{m-1}{m-3} - \frac{(m-1)^2}{(m-2) \cdot (m-3)} \right), \quad \text{et } b_{101} = 4 \cdot \frac{m-1}{m-3} \cdot \omega_1$$

C. Quelques valeurs des matrices de changement de base Bézier - Tchebycheff d'après Watkins et Worsey

n=2:

$$T_{BM} \cdot T_{MT} = \frac{1}{8} \begin{pmatrix} 3 & 4 & 1 \\ 2 & 0 & -2 \\ 3 & 4 & 1 \end{pmatrix} \text{ et } T_{MT}^{-1} \cdot T_{BM}^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & -3 & 1 \end{pmatrix}$$

n=3:

$$T_{BM} \cdot T_{MT} = \frac{1}{32} \begin{pmatrix} 10 & -15 & 6 & -1 \\ 6 & -3 & -6 & 3 \\ 6 & 3 & -6 & -3 \\ 10 & 15 & 6 & 1 \end{pmatrix} \text{ et } T_{MT}^{-1} \cdot T_{BM}^{-1} = \frac{1}{3} \begin{pmatrix} 3 & 3 & 3 & 3 \\ -3 & -1 & 1 & 3 \\ 3 & -5 & -5 & 3 \\ -3 & 15 & -15 & 3 \end{pmatrix}$$

n=4:

$$T_{BM} \cdot T_{MT} = \frac{1}{128} \begin{pmatrix} 35 & -56 & 28 & -8 & 1 \\ 20 & -16 & -16 & 16 & -4 \\ 18 & 0 & -24 & 0 & 6 \\ 20 & 16 & -16 & -16 & -4 \\ 35 & 56 & 28 & 8 & 1 \end{pmatrix} \text{ et}$$

$$T_{MT}^{-1} \cdot T_{BM}^{-1} = \frac{1}{6} \begin{pmatrix} 6 & 6 & 6 & 6 & 6 \\ -6 & -3 & 0 & 3 & 6 \\ 6 & -6 & -10 & -6 & 6 \\ -6 & 21 & 0 & -21 & 6 \\ 6 & -42 & 70 & -42 & 6 \end{pmatrix}$$

n=5:

$$T_{BM} \cdot T_{MT} = \frac{1}{512} \begin{pmatrix} 126 & -210 & 120 & -45 & 10 & -1 \\ 70 & -70 & -40 & 65 & -30 & 5 \\ 60 & -20 & -80 & 30 & 20 & -10 \\ 60 & 20 & -80 & -30 & 20 & 10 \\ 70 & 70 & -40 & -65 & -30 & -5 \\ 126 & 210 & 120 & 45 & 10 & 1 \end{pmatrix} \text{ et}$$

$$T_{MT}^{-1} \cdot T_{BM}^{-1} = \frac{1}{5} \begin{pmatrix} 5 & 5 & 5 & 5 & 5 & 5 \\ -5 & -3 & -1 & 1 & 3 & 5 \\ 5 & -3 & -7 & -7 & -3 & 5 \\ -5 & 13 & 7 & -7 & -13 & 5 \\ 5 & -27 & 21 & 21 & -27 & 5 \\ -5 & 45 & -105 & 105 & -45 & 5 \end{pmatrix}$$