Code : Université Boumerdès-FSI-DPTMéca-CFAO-1999-01

Mémoire de Magister

Spécialité : Génie-mécanique
Option : Mécanique

REDUCTION DE DEGRE PAR APPROXIMATION DES COURBES DE BEZIER
Application à l'échange de données entre systèmes

Par

Farid ASMA

Soutenu devant le jury suivant le 10-07-1999

M. RECHAK Saïd	Professeur à l'ENP Alger, Président du jury
M. BELAIDI Idir	Docteur NT, Directeur du mémoire
M. MOHAMMEDI Kamal	Maître de Conférences à l'U.B, Examinateur
M. NOUR Abdelkader	Maître de Conférences à l'U.B, Examinateur
M. RACHEDI Kamel	Chargé de Cours à l'U.Boumerdès, Examinateur
M. BELAIDI Abderrahmane	Chargé de Cours à l'U.Boumerdès, Examinateur

Laboratoire de Modélisation des procédés de Fabrication et CFAO
Département de Mécanique
Faculté des sciences de l'Ingénieur
Université de Boumerdès

Remerciements

Je tiens à exprimer mes remerciements et ma sincère reconnaissance à tous ceux qui ont de près ou de loin manifesté leur intérêt, et apporté leur collaboration à ce présent travail, et en particulier à:

Monsieur I. BELAIDI, Docteur en C.F.A.O à l'Université de Boumerdès, d'avoir proposé et dirigé ce mémoire et de m'avoir permis de l'accomplir grâce à ses conseils précieux et sa rigueur dans le travail,

Monsieur S. RECHAK, Professeur à l'Ecole Nationale Polytechnique d'Alger, d'avoir bien voulu présider le jury de soutenance,

Monsieur K. MOHAMMEDI, Maître de Conférences à l'Université de Boumerdès, d'avoir bien voulu participer au jury de soutenance en qualité d'examinateur,

Monsieur A. NOUR, Maître de Conférences à l'Université de Boumerdès, d'avoir bien voulu participer au jury de soutenance en qualité d'examinateur,

Monsieur K. RACHEDI, Chargé de recherche à l'Université de Boumerdès, d'avoir bien voulu être membre du jury de soutenance, et pour ces conseils qui ont été d'un grand apport,

Monsieur A. BELAIDI Chargé de recherche à l'Université de Boumerdès, d'avoir bien voulu être membre du jury de soutenance, et pour ces conseils qui ont été d'un grand apport,

Je tiens également à exprimer ma reconnaissance à tous les enseignants de l'université de Boumerdès, particulièrement R. HABACHOU, SLIMANI, Y. TALAMALI, A. DJERRADA, B. BRACHEMI, T. KIRATI, K. RACHEDI, A. BELAIDI et I. BELAIDI.

Mes vifs remerciements vont aussi à mes amis et collègues de l'Université de Tizi-ouzou, M. DJELLIL, M. REZAOUI, H. AIT ABDERRAHMANE, M. LAHLOUH et A. AHMED ALI, pour leur soutient moral, combien précieux.

Sans oublier mes chers amis, R. BOURAS, S. BOUCIF, H. METAHRI, A. MEZENER, M^{elle} B. MADANI, M. LOUNICI, B. HASSAID, M. CHALLALI et A. ABOUTALEB, de m'avoir compris dans les moments difficiles ainsi que pour leurs avis et conseils qui m'ont été d'une aide incontestable.

Dédicaces

Je dédie ce modeste travail à

la mémoire de ma mère Fatima ASMA,

mon père Chabane,

ma belle-mère

mes grand-prents maternels

la mémoire de mes grand-parents paternels

mes frères et sœurs,

ma bien aimée Saliha AIT MEBAREK,

toute la famille,

mes amis.

Table des matières

Remerciements
Dédicace
Table des matières
Table des illustrations
Liste des abréviations - Notations

	Introduction	1
I.	**COURBES ET SURFACES : Modèles de Bézier et B-spline**	
I.2.	**Courbes et surfaces de Bézier**	4
I.2.1.	Fonction de base Bernstein	4
I.2.2.	Courbes de Bézier non rationnelles	5
I.2.3.	Courbes de Bézier rationnelles	5
I.2.4.	Surfaces de Bézier non rationnelles	8
I.2.5.	Surfaces de Bézier rationnelles	8
I.3.	**Courbes et surfaces B-Splines**	9
I.3.1.	Fonction de base B-spline	9
I.3.2.	Courbes B-spline non rationnelles	11
I.3.3.	Courbes B-spline rationnelles	12
I.3.4.	Surfaces B-spline non rationnelles	15
I.3.5.	Surfaces B-spline rationnelles	16
II.	**Algorithmique des transformations**	
II.1.	**Transformations dans le modèle de Bézier**	18
II.1.1.	Subdivision	18
II.1.1.1.	Courbes non rationnelles	18
II.1.1.2.	Courbes rationnelles	18
II.1.2.	Elévation de degré	20
II.1.2.1.	Courbes non rationnelles	20
II.1.2.2.	Courbes rationnelles	21
II.1.3.	Réduction de degré	21
II.1.3.1.	Courbes non rationnelles	21
II.1.3.2.	Courbes rationnelles	22
II.1.4.	Raccordement des courbes	23
II.1.4.1.	Courbes non rationnelles	23
II.1.4.2.	Courbes rationnelles	24

II.2. Transformations dans le modèle B-spline 24

II.2.1. Insertion de nœuds 24
II.2.1.1. Courbes non rationnelles 24
II.2.1.2. Courbes rationnelles 25
II.2.2. Suppression de nœuds 27
II.2.2.1. Courbes non rationnelles 27
II.2.2.2. Courbes rationnelles 29
II.2.3. Subdivision et l'algorithme d'Oslo 30
II.2.3.1. Courbes non rationnelles 30
II.2.3.2. Courbes rationnelles 32

II.3. Changement de bases 34

II.3.1. B-spline- Bézier 34
II.3.1.1. Le polygone local de Bézier d'une courbe B-spline 34
II.3.1.1. Les polygones de Bézier d'une courbe B-spline 35
II.3.2. Bézier- B-spline 35
II.3.2.1. Polygone B-spline d'une courbe de Bézier 35
II.3.2.2. Polygone B-spline des courbes locales de Bézier 36
II.3.2.3. Algorithme WM 36

III. Réduction de degré des courbes de Bézier non rationnelles

III.1. Algorithmique de réduction de degré 39

III.1.1. Méthode de Forrest 39
III.1.1.1. Algorithme de réduction 39
III.1.1.2. Algorithme de sub-réduction 44
III.1.2. Méthode de Bensalah 47
III.1.2.1. Algorithme de réduction 47
III.1.2.2. Algorithme de sub-réduction 49
III.1.3. Méthode de Hoschek 53
III.1.3.1. Paramétrisation intrinsèque d'une courbe de Bézier 53
III.1.3.2. Réduction de degré 55
III.1.4. Méthode de Watkins et Worsey 62
III.1.4.1. Algorithme de réduction 62
III.1.4.2. Algorithme de sub-réduction 64
III.1.4.3. Conversion de base 68
III.1.5. Méthode de Eck 70
III.1.5.1. Réduction de degré et continuité géométrique 70
III.1.5.2. Algorithme de sub-réduction 74

III.2. Analyse comparative des méthodes de Réduction 78

III.2.1. Etude critique 78
III.2.2. Comparaison analytique 79

III.2.3.	Comparaison expérimentale	80
III.2.4.	Méthode proposée	90
III.2.4.1.	Méthode d'approximation	90
III.2.4.2.	Algorithme de réduction	91
III.2.4.3.	Algorithme de sub-réduction	95
III.2.4.4.	Comparaison avec les autres méthodes	97
III.2.4.5	Conclusion	98
IV	**Application à l'échange de données entre systèmes**	
IV.1.	**Extension de la méthode proposée**	99
IV.1.1.	Courbes de Bézier rationnelles	99
IV.1.1.1.	Algorithme de subdivision	99
IV.1.1.2.	Algorithme de réduction	100
IV.1.1.3.	Algorithme de sub-réduction	105
IV.1.2.	Surfaces de Bézier non rationnelles	108
IV.1.3.	Surfaces de Bézier rationnelles	109
IV.1.4.	Courbes B-spline non rationnelles	111
IV.1.5.	Courbes B-spline rationnelles	112
IV.2.	**Transmission de données**	113
	Conclusion	114
	Références bibliographiques	
	Annexes	
A	Logiciel CONVER : 1. Structure des données	I
	2. Architecture et Manipulations	V
B	Equations de continuité géométriques d'après Hoschek	IX
C	Changement de base Bézier–Tchebycheff d'après Watkins et Worsey	X
D	Tableaux : Etude comparative	XI

Table des illustrations

Chapitre I

I.1. Polynômes de Bernstein de degré 5 — 5

I.2. Courbe de Bézier non rationnelle de degré 5 — 6

I.3. (a) Polynômes rationnels de Bernstein de degré 5 avec (W_0, ..., W_5)=(1, 3, 1, 1, 1, 1); (b) Polynômes de Bernstein de degré 5 avec (W_0, ..., W_5)=(1, 1, 1, 0.3, 1, 1) — 7

I.4. Courbes de Bézier rationnelles de degré 5 obtenues en appliquant différentes valeurs à (a) w_1 (b) w_3. — 8

I.5. (a) Surface de Bézier non rationnelles de degrés 3×4; (b) Surface de Bézier rationnelle de degrés 3×4 avec $w_{12}=w_{22}=20$ — 9

I.6. a)fonction de base B-spline quadratique définie par le vecteur nœuds {0, 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} — 10

I.7. Courbe B-spline cubique non rationnelle, avec son polygone de contrôle associé aux fonctions de base montrée en I.6. (b) — 12

I.8. Fonctions de base rationnelles quadratiques: (a) T={0, 0, 0, 1/3, 2/3, 1, 1, 1} et (w_0, ..., w_4)=(1, 4, 1, 1, 1); (b) T={0, 0, 0, 1/3, 2/3, 1, 1, 1} et (w_0, ..., w_4)=(1, 1, 0.3, 1, 1) — 13

I.9. Fonctions de base B-spline rationnelles cubiques: (a) T={0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1} et (w_0, ..., w_6)=(1, 2, 1, 1, 1, 1, 1); (b) T={0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1} et (w_0, ..., w_6)=(1, 1, 1, 3, 1, 1, 1) — 14

I.10. Courbes B-spline rationnelles cubiques — 15

I.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} et V={0, 0, 0, 0, 1, 1, 1} — 16

Chapitre II

I.12. Surface B-spline rationnelle de même réseau de contrôle et mêmes vecteurs nœuds que la I.11 avec $w_{11}=3$. — 18

II.1. Courbe de Bézier non rationnelle P(t) de degré 5 subdivisée en deux courbes R(t) et Q(t) de même degré — 20

II.2. Courbe de Bézier rationnelle de degré 5 (de poids w_i) subdivisée en deux courbes rationnelles de degré 5 (de poids α_j et β_k) — 20

II.3. Courbe de Bézier non rationnelle de degré 3 élevée au degré 5 — 21

II.4.a. Courbe de Bézier non rationnelle de degré 3 réduite au degré 2 par les deux méthodes polygone 1 et 2 — 23

II.4.b. Courbe précédente de degré 3 réduite par la méthode du polygone moyen (première approximation) — 23

II.5. Courbe de Bézier rationnelle de degré 6 réduite au degré 5 par la méthode du polygone moyen W={1,1.2,0.8,0.5,0.8,1.2,1}; W'={1,1.304,0.516,0.516,1.304,1} — 24

II.6. Insertion du nœud 0.5 pour une courbe B-spline non rationnelle avec T={0,0,0,0,0.2,0.8,1,1,1,1} — 26

II.7. Insertion du 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'={1,1.3,0.7,0.7,1.15,1.15,1} — 27

II.8.	Suppression du 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}	29
II.9.	Suppression du 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}; W'={1,1.2,0.72,1.648,1.6,0.8,1.2,1}	30
II.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} et T1={0,0,0,0,0.6,1,1,1,1}, T2={0,0,0,0,0.2,1,1,1,1}	33
II.11.	Courbe B-spline rationnelle de degré 3 subdivisée en deux courbes à t=0.5, avec T={0,0,0,0,0.2,0.6,0.8,1,1,1,1}, T1={0,0,0,0,0.4,1,1,1,1} et T2={0,0,0,0,0.2,0.6,1,1,1,1}; et w={1,1.2,0.8,1,0.8,1.2,1}, w1={1,1.2,0.87,0.92,0.92} et w2={0.92,0.93,0.93,0.8,1.2}	34
II.12.	Courbe B-spline non rationnelle de degré 3 transformée en 3 courbes de Bézier non rationnelles de degré3 avec T={0,0,0,0,0.3,0.7,1,1,1,1}	36

Chapitre III

III.1.a.	Courbe de Bézier de degré 6 réduite au degré 5	39
III.1.b	Erreur de position de la courbe précédente	39
III.1.c.	Variation de la courbure courbe précédente et sa courbe réduite.	40
III.2.a..	Courbe de Bézier de degré 9 réduite au degré 8	42
III.2.b.	Erreur de position de la courbe précédente	42
III.2.c.	Variation de la courbure courbe précédente et sa courbe réduite.	43
III.3.a.	Courbe de degré 16 réduite en 5 courbes de degré 6	44
III.3.b.	Erreur de position le long de la courbe de degré 16	45
III.3.c.	Variation de l'erreur su la courbure	45
III.4.	Courbe de degré 5 réduite au degré 4 par BEN1 et BEN2	47
III.5.	Courbe de degré 5 précédente réduite au degré 4 par BENm	47
III.6.a.	Courbe de degré 10 réduite en 8 courbes de degré 6 par l'algorithme SUB-BEN	49
III.6.b.	Erreur de position pour la courbe précédente	50
III.6.c.	Variation de l'erreur sur la tangente	50
III.6.d.	Variation de l'erreur sur la courbure après ajustement	51
III.7.a	Erreurs ponctuelles d'approximation	53
III.7.b	Correction de l'erreur par paramétrisation	53
III.8.a.	Courbe de Bézier de degré 7 réduite au degré 5	57
III.8.b.	Variation de l'erreur d'approximation	57
III.8.c.	Courbure des deux courbes	58
III.9.a.	Courbe de Bézier de degré 8 réduite en 4 courbes de degré 7 (k=3) par l'algorithme SUB-HOS avec ε_0=0.01	59
III.9.b.	Variation de l'erreur d'approximation pour la courbe sub-réduite précédente	60
III.9.c.	Variation de l'erreur sur la courbure pour la courbe précédente	60
III.10.a	Courbe de degré 6 réduite au degré 5 par la méthode de Watkins & Worsey	61
III.10.b	Variation de l'erreur sur la tangente	61
III.11.b	Variation de l'erreur de position	66
III.11.c.	Variation de l'erreur sur la tangente	66
III.12.a	Courbe de Bézier de degré 8 réduite au degré 7 par l'algorithme EC en respectant la continuité de la courbure (α=3)	71
III.12.b	Variation de l'erreur de position	71

III.12.c	Variation des courbures	72
III.13.a.	Courbe de Bézier de degré 8 réduite en deux courbes de degré 7 par l'algorithme SUB-EC	74
III.13..	Variation de l'erreur de position	75
III.13.c	Variation de l'erreur sur la courbure	75
III.14.a	Variation du temps de calcul pour le test 1	79
III.14.b	Variation de l'erreur pour le test 1	79
III.15.a	Variation du temps de calcul pour le test 2	80
III.15.b	Variation de l'erreur pour le test 2	80
III.16.a	Variation du temps de calcul pour le test 3	81
III.16.b	Variation de l'erreur pour le test 3	81
III.17.a	Variation du temps de calcul pour le test 4	82
III.17.b	Variation de l'erreur pour le test 4	82
III.18.a	Variation du temps de calcul pour le test 5	83
III.18.b	Variation de l'erreur pour le test 5	83
III.19.a	Variation du temps de calcul pour le test 6	84
III.19.b	Variation de l'erreur pour le test 6	84
III.20.a	Courbe de Bézier non rationnelle de degré 9 réduite au degré 7 avec une continuité C^2	92
II.20..b.	Courbure des deux courbes précédentes	92
III.21.a.	Courbe de Bézier non rationnelle de degré 12 réduite en deux courbes de degré 8 par l'algorithme proposé	93
III.21.b.	Variation de l'erreur de courbure	93

Chapitre IV

IV.1.	Courbe de Bézier rationnelle de degré 8 réduite au degré 6 par l'algorithme PRO#3 avec w={1,1.2,0.8,1,0.5,1,0.8,1.2,1}, w'={1,1.27,0.41,1.22,0.41,1.27,1}	103
IV.2.a	Courbe de Bézier rationnelle de degré 8 réduite en 3 courbe de degré 6 pour une continuité d'ordre3 et une tolérance de 0.01 mm	104
IV.2.b	Erreur de position pour la réduction précédente	105
IV.2.c	Variation des courbures pour la réduction précédentes	105
IV.3.	Surface de Bézier non rationnelle de degré (8,8) réduite au degré (6,6).	107
IV.4.	Surface de Bézier rationnelle de degré (8,8) réduite au degré (7,7).	108
IV.5.	Courbe B-spline non rationnelle de degré 6 réduite au degré 5.	109
IV.6.	Courbe B-spline rationnelle de degré 6 réduite au degré 4.	110
IV.7.	Modes de transfert de données	111
IV.8.	Structure de changement de format pour CONVER	112
IV.9.	Possibilité d'échange de données avec CONVER	112

Annexes

1.	Liste linéaire des entités	II
2.	Logiciel CONVER	III
3.	Structure du logiciel CONVER	IV

Notations

$B_{i,n}(t)$	$i^{ème}$ fonction Bernstein non rationnelle de degré n.
$B_{i,m;j,n}(u,v)$	fonction Bernstein non rationnelle bi-variable.
C^{α}	continuité géométrique d'ordre α.
$G_{i,n}(t)$	$i^{ème}$ fonction Bernstein rationnelle de degré n.
$G_{i,m;j,n}(u,v)$	fonction Bernstein rationnelle bi-variable.
H	projection homogène de l'espace 3D en 4D.
M_i	point d'une courbe paramétrique.
$N_{i,p}(t)$	$i^{ème}$ fonction de base B-spline non rationnelle de degré p.
$N_{i,m;j,n}(u,v)$	fonction de base B-spline non rationnelle bi-variable.
P_i, Q_i, R_i, ...	pôles d'une courbe paramétrique.
P_{ij}, Q_{ij}, R_{ij}, S_{ij} ...	pôles d'une surface paramétrique.
P(t), Q(t), R(t), ...	courbe paramétrique , modèle de Bézier ou B-Spline.
$R_{i,p}(t)$	$i^{ème}$ fonction de base B-spline rationnelle de degré p.
$R_{i,m;j,n}(u,v)$	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-splines.
$T_i(x)$	polynôme de Tchebycheff de degré i.
t	paramètre curviligne d'une courbe, généralement $t \in [0, 1]$.
u,v	paramètres curvilignes d'une courbe, généralement $(u,v) \in [0, 1]^2$.
w_i	poids d'une courbe rationnelle.

Liste des abréviations

C.A.O	Conception Assistée par Ordinateur
C.F.A.O	Conception et Fabrication Assistées par Ordinateur
CN	Commande numérique
DES	Format des fichier de CONVER
DXF	Data eXchange Format
F.A.O	Fabrication Assistée par Ordinateur
IGES	Internatinal Graphic Exchange Specification
MAS	Fichier masque du logiciel CONVER
MNU	Fichier menu du logiciel CONVER
NURBS	Non Uniform Rational B-Spline
R 12	Release 12 (Version 12) du format DXF d'AUTOCAD
SET	Système d'Echange et de Transfert de données géométriques
STEP	STandard for Exchange of Product model data
VDA/FS	Verband der Deutschen Ingénieure/ FlächenSchnittstellen exchange in des Automobil-Industrie

ملخص

بقصد تطبيقه في تبادل المعطيات بين جملة برامج التصميم و الصناعة بمساعدة الكمبيوتر(ت ص م ك)، يعالج هذا المرجع مشكلة تخفيظ الدرجة بالتقريب لمنحنيات بيزيي الغير منطقية من درجة م إلى درجة ن حيث ن < م بتقدير الإستمرارية الهندسية برتبة α>(ن-1)/2 على نهاية المنحن. هدفنا الخاص هو اختيار و استخدام طريقة التخفيض مكيفة مع تبادل المعطيات بين جملة برامج التصميم و الصناعة بمساعدة الكمبيوتر ذوي أسس المنحنيات و درجات مختلفة. في البداية أعطينا لمحة شاملة للمعلومات الرياضية للخصائص الهندسية لمنحنيات و سطائح بيزيي و ب-سبلين. كل المحولات الرياضية الضرورية للتحويل بالتقريب، لتطبيق و توسيع تخفيض الجردة على النموذج المنطقي و أسس ب-سبلين قدمت باختصار على شكل ألغورتمات. قدمنا حصيلة بيبليوغرافية عميقة لوسائل و طرق تخفيض الدرجة لمنحنيات بيزيي الغير منطقية مقدما المبدأ و ألغورتمات خفظ الدرجة، أتبعت ببحث مقارن مكننا من إختيار أفضل و أمثل طريقة.

هذا البحث المقارن أقيم على الهيئة الألغورتمية و النتائج التطبيقية و أضهر فوائد طريقة تقريب هوشاك لخفض درجة منحنيات بيزيي الغير منطقية. غير أن هذه الأخيرة سجلت ثقل في الحساب الذي قادنا لإقتراح طريقة الخفض المقللة للمعيار المربعي في أساس كثير الحدود برنستين باستعمال مفهوم فصل المتغيرات لإجتناب الحسابات الغير خطية.

التحولات الخامة (تخفيض و رفع الدرجة، إدماج و إلغاء العقد، التجزئة، تغيير الأسس) المتدخلة في إستبدال المعلومات بين أنضمة ت ص م ك قدمت بالتفاصيل و أدمجت في التطبيق المقترح. هذا الأخير مكننا من إنجاز تبادل المعلومات بين البرنامج المنجز "Conver" و برنامج "Autocad" و "Surfcam" في مخبر ت ص م ك لمعهد الهندسة الميكانكية لجامعة بومرداس بمتوسط المقاس المعياري لتبادل المعلومات المرسومة "DXF".

و مكننا برفع إجبار تحديد درجة المنحنيات المفروض على البرامج الآلية الذي يخفض مرونة الصيانة و التبديل على منحنيات بيزيي و ب-سبلين.

النتائج المحرزة من المقارنة بين منهجيات تخفيض الدرجة تبين قدرات الطريقة المقترحة، في المجال الألغورتمي و النتائج التطبيقية بدون تقليل في مجال منضوريات التوسيع و الإدماج الصناعي.

الكلمات الرئيسية

بيزيي، ب-سبلين، منحنيات و سطائح، خفض الدرجة، تقريب، تبادل المعلومات، تجزئة.

Résumé

Dans le cadre d'une application à l'échange de données entre systèmes de CFAO, ce présent travail propose une étude comparative des méthodes de réduction des courbes de Bézier non rationnelles du degré n au degré m<n, en respectant une continuité géométrique d'ordre $\alpha<(m-1)/2$ aux extrémités. Cette étude a abouti au développement d'une méthode de réduction de degré présentant les caractéristiques de précision et de rapidité requises, facilement extensible aux autres modèles de courbes et surfaces et adaptée au transfert de données entre systèmes de bases et degrés différents.

Après avoir donné un aperçu général des fondements mathématiques et des caractéristiques géométriques des courbes et surfaces de Bézier et B-spline, toutes les transformations mathématiques inhérentes à la conversion par approximation et à l'extension de la réduction de degré aux modèles rationnels et à la base B-spline, ont été présentées et exprimées sous forme d'algorithmes.

Une synthèse bibliographique approfondie des méthodes de réduction de degré des courbes de Bézier non rationnelles a été établie, présentant le principe et les algorithmes de réduction de degré, parachevée par une étude comparative qui a permis de choisir une méthode de réduction optimale. Cette étude comparative, basée sur les structures algorithmiques ainsi que sur les résultats pratiques, montre l'avantage de l'approximation de Hoschek pour la réduction de degré des courbes non rationnelles.

Toutefois, cette dernière présente une lourdeur de calcul qui nous a amenés à proposer une méthode de réduction qui minimise la norme carrée dans la base des polynômes de Bernstein faisant appel à la notion de séparation des variables, pour éviter les méthodes non linéaires.

Les transformations principales (réduction et élévation de degré, insertion et suppression de nœuds, subdivision, changement de base) intervenant dans la communication de données entre systèmes, ont été présentées dans le détail et intégrées dans l'application proposée. Cette dernière permet en effet un échange de données entre le programme élaboré « Conver » et les logiciels « Autocad » et « Surfcam », établie au laboratoire de CFAO de l'IGM de l'Université de Boumerdès par l'intermédiaire du format standard d'échange de données graphiques « DXF », levant ainsi la contrainte de limitation du degré des courbes imposée à ces logiciels qui réduisait la flexibilité de construction et de modifications des courbes de Bézier et B-spline.

Les résultats obtenus à partir de la comparaison établie entre les méthodes de réduction mettent en valeur la méthode de réduction proposée, aussi bien du point de vue algorithmique et résultats pratiques que des perspectives d'extension et d'intégration industrielle.

Mots clés:

Bézier, B-splines, courbes et surfaces, réduction de degré, approximation, échange de données, subdivision.

Abstract

In view of its application to data exchange between systems, this present work treats the problem of reducing the degree of Bézier curve by approximation from n to a prescribed terget degree m < n whereby geometric continuity of any order $\alpha<(m-1)/2$ can be preserved at the two endpoints. The set objective returns to the choice and to the stake in opens one method of reduction adapted to the transfer of datas between CAD/CAM systems with different degrees and bases.

We gave a general preview first some mathematical foundations and the geometric features of Bézier and B-spline curves and surfaces. All the mathematical necessary for the transformations to degree conversion by approximation and its application and extension to the rational patterns and B-spline basis, was presented and expressed under shape of algorithms. A bibliographic deepened synthesis some methods of non rational Bézier curves degree reduction was settled presenting the principle and the degree reduction algorithms , well attended one comparative survey who allowed us to choose a method of optimal reduction. This comparative survey, based on the algorithmiques structures thus that on the convenient results, shows the interest of Hoschek approximation for degree reduction of non rational Bézier curves. However, this method present an oppressiveness of count who brought us to suggest a method of reduction who understates the square norm in the Bernstein using the notion of separation variables, in order to avoid the non linear methods. The main transformations (degree reduction and elevation, insertion and Knot removal, subdivision, basis transformation) intervening in the data communication between systems, was presented in the detail and embedded in the suggested application. This last permits an exchange of datas between the formulated program indeed « Conver » and the software « Autocad » and « Surfcam » settled in the CAD/CAM laboratory of The IGM of the University of Boumerdès aided by the standard data graphic exchange format « DXF », rising thus the constraint of degree limitation curves imposed to these software and who reduced the construction and modifications pliability of the curves of Bézier and B-spline.

The results obtained starting from the comparison between the methods of settled reduction puts the suggested reduction method in value, also well of algorithmique dawns and convenient result of view that some perspectives of extension and industrial integration.

Keys words:

Bézier, B-splines, curves and surfaces, degree reduction, approximation, data exchange, subdivision.

Introduction

Introduction

L'un des problèmes essentiels rencontrés en Conception et Fabrication Assistées par Ordinateur ou CFAO est la communication et l'échange de données entre systèmes. En effet, les systèmes mis sur le marché utilisent une grande variété de modèles de définition des formes de bases et de degrés différents. La communication entre ces systèmes est devenue aujourd'hui une nécessité au sein d'une entreprise en raison de la généralisation des applications sur support informatique et de l'intérêt de la communication avec le monde extérieur dans le cadre des relations entre commanditaires et sous-traitants. Par conséquent, cela permettra d'éviter le risque de demeurer dépendant d'un fournisseur, de renoncer à des actions de partenariat, de choisir le sous-traitant en fonction de son système ou lui imposer son propre système et d'être contraint de développer des interfaces spécifiques.

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 posent 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. Toutefois, l'utilisation de ces derniers, en matière de modèles géométriques, n'offre pas une satisfaction totale en raison des problèmes qui subsistent encore et qui sont liés à la perte d'informations, à l'interprétation de définitions standards (cas de surfaces évolutives: Bézier, B-spline, Nurbs), au mode de modélisation en amont (surfacique, volumique, associatif ou variationnel) dont la mise en œuvre ne se limite pas aux données géométriques mais impliquant des procédures spécifiques. De plus, les systèmes de CAO des courbes et des surfaces reposent sur plusieurs modèles mathématiques. Ces différents modèles (Bézier, B-spline, Box-splines, Coons, N.U.R.B.S,...) sont fondés sur des fonctions 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 C.A.O exige souvent des modèles polynomiaux de degré élevé. Cela se justifie aisément par la flexibilité recherchée par l'utilisateur dans la conception de formes complexes et le respect des contraintes de conception géométriques, le calcul de courbes définies par l'intersection de deux surfaces et des courbes Offset etc. L'utilisateur peut en effet souhaiter pouvoir agir sur la forme de la courbe sans modifier ses caractéristiques géométriques.

Dès lors, l'importance et l'intérêt que présente la réduction de degré réside dans la possibilité de préserver ces avantages tout en évitant les problèmes d'instabilité liés

au degré élevé et d'incompatibilité lors d'un transfert de données vers des systèmes de F.A.O qui fonctionnent généralement avec des degrés faibles.

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 système de CFAO, basée sur une étude comparative des méthodes de réduction de degré des courbes de Bézier non rationnelles qui a abouti au développement d'une méthode de réduction de degré présentant les caractéristiques de précision et de rapidité requises, facilement extensible aux autres modèles de courbes et surfaces et adaptée au transfert de données entre systèmes de bases et degrés différents. Ainsi, nous avons structuré ce document en quatre chapitres organisés de la manière suivante :

Dans le chapitre 1, nous avons d'abord donné un aperçu général des fondements mathématiques et les caractéristiques géométriques des courbes et surfaces de Bézier et B-spline.

Toutes les transformations mathématiques nécessaires à la conversion par approximation, à l'application, et à l'extension de la réduction de degré aux modèles rationnels et à la base B-spline, ont été présentées dans le chapitre 2 et exprimées sous forme d'algorithmes.

La première partie du chapitre 3 est une synthèse bibliographique approfondie des méthodes de réduction de degré des courbes de Bézier non rationnelles, où sont présentés le principe et les algorithmes de réduction de degré. Dans la seconde partie, une étude comparative a été établie et nous a permis de choisir une méthode de réduction optimale. Cette étude comparative, basée sur les structures algorithmiques ainsi que sur les résultats pratiques, montre l'intérêt de l'approximation de Hoschek pour la réduction de degré des courbes non rationnelles. Toutefois, cette dernière présente une lourdeur de calcul qui nous a amenés à proposer, dans la dernière partie de ce chapitre, une méthode de réduction qui minimise la norme carrée dans la base des polynômes de Bernstein faisant appel à la notion de séparation des variables, pour éviter les méthodes non linéaires.

L'application à l'échange de données entre systèmes de CFAO fait l'objet du chapitre 4. Ce dernier traite de l'extension de la méthode de réduction de degré proposée aux courbes rationnelles de Bézier, aux surfaces rationnelles et non rationnelles ainsi qu'aux courbe B-splines rationnelles et non rationnelles et est parachevé par le développement et la mise en œuvre d'un système d'échange de données avec les logiciels de CAO « Autocad » et de CFAO « Surfcam » .

Une conclusion générale clôture notre étude en présentant de façon synthétique les résultats obtenus à partir de la comparaison entre les méthodes de réduction établie et met en valeur la méthode de réduction proposée, aussi bien du point de vue algorithmique et résultats pratiques que des perspectives d'extension et d'intégration industrielle.

COURBES ET SURFACES : Modèles de Bézier et B-spline

I.2.	**Courbes et surfaces de Bézier**	4
I.2.1.	Fonction de base Bernstein	4
I.2.2.	Courbes de Bézier non rationnelles	5
I.2.3.	Courbes de Bézier rationnelles	5
I.2.4.	Surfaces de Bézier non rationnelles	8
I.2.5.	Surfaces de Bézier rationnelles	8
I.3.	**Courbes et surfaces B-Splines**	9
I.3.1.	Fonction de base B-spline	9
I.3.2.	Courbes B-spline non rationnelles	11
I.3.3.	Courbes B-spline rationnelles	12
I.3.4.	Surfaces B-spline non rationnelles	15
I.3.5.	Surfaces B-spline rationnelles	16

I. COURBES ET SURFACES
Modèles Bézier- B-spline

I.1. Introduction

Les polynômes, les fonctions trigonométriques, les fonctions exponentielles ou logarithmiques sont autant de formes mathématiques permettant la description de courbes et de surfaces. En outre, les polynômes représentent un bon compromis entre:

- la diversité des formes des objets à modéliser,
- la simplicité des traitements qui leur sont appliqués au cours du processus de construction,
- la rapidité avec laquelle doivent être réalisées la plupart des fonctions d'un modeleur surfacique pour bénéficier d'une interactivité suffisante.

De plus les polynômes permettent de représenter des fonctions continues sur un intervalle [a, b] donné avec une précision arbitrairement fixée.
Les travaux de Schoenberg ont contribué à mettre en place des techniques de modélisation, basées sur des polynômes, issues des outils et méthodes de tracés utilisés dans les secteurs de l'automobile, de la construction navale,...
Parmi les polynômes, il reste à considérer les formes:

- explicites,
- implicites,
- paramétriques.

Les formes polynomiales explicites, $y=f(x)$ ou $z=g(x,y)$ ne satisfont pas les conditions de diversité puisqu'elles ne permettent pas de représenter des courbes ou des surfaces fermées, dont l'utilisation est fréquente, pas plus que des traitements tels que translation ou rotation ne peuvent être effectués simplement.
Quant aux formes polynomiales implicites $f_1(x,y,z)=0$, $f_2(x,y,z)=0$, pour une courbe définie par l'intersection de deux surfaces, ou $g(x,y,z)=0$ pour une surface, elles donnent accès à une diversité de formes importante. La complexité de certains traitements, intersection en particulier, se voit réduite par rapport à celle de formes polynomiales paramétriques.
Les formes paramétriques, P(u) ou Q(u,v) sont à la fois capables de représenter une grande diversité de courbes et de surfaces et disposent d'algorithmes simples pour les traitements les plus fréquents (visualisation, déplacement, limitation de courbes ou de surfaces,...)
Etant donné les moyens informatiques disponibles, les formes polynomiales paramétriques représentent le meilleur compromis entre les trois critères de sélection mentionnés précédemment par rapport aux formes explicites et implicites.
La grande majorité des systèmes de C.F.A.O utilisent les formes polynomiales paramétriques. Néanmoins, les schémas de représentation de ces systèmes diffèrent dans la base de polynômes utilisés. Les fonctions de base fréquemment utilisées dans les différents systèmes 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 les plus utilisés, Bézier et B-spline.

L'utilisation de ces modèles sous leur forme rationnelle dans les applications de la C.F.A.O devient de plus en plus répandue. Les courbes et les surfaces B-spline rationnelles ont été introduites dans le standard d'échange Initial Graphics Exchange Specification (IGES) depuis 1983, et un nombre important de système de C.A.O\ F.A.O basés sur les représentations rationnelles de Bézier et B-spline existent, on peut citer à titre d'exemple: le modeleur SDRC, Geomod, Euclid,...
Cette popularité des modèles Bézier et B-spline rationnels est due aux facteurs suivant: [*Léon '91*]

- Ils donnent une forme mathématique unique pour toutes les formes géométriques standard (lignes, coniques, cercle, plans et surfaces quadratiques) autant que les formes libres de courbes et de surfaces.
- Ils offrent un grand degré de liberté qui permet de générer une large variété de formes.
- Possèdent la propriété importante de l'invariance affine.

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

I.2. Courbes et surfaces de Bézier

I.2.1. Fonction de base Bernstein

La i^{eme} fonction Bernstein de degré n , notée $B_{i,n}(t)$, est définie par:

$$B_{i,n}(t) = \binom{n}{i}.(1-t)^{(n-i)}.t^i \quad \text{avec} \quad \binom{n}{i} = \frac{n!}{i!.(n-i)!} \qquad (1)$$

Qu'on peut écrire sous la forme récurrente suivante:

pour i allant de 1 à n-1 : $B_{i,n}(t)=(1-t).B_{i,n-1}(t)+t.B_{i-1,n-1}(t)$,

$B_{0,n}(t)=(1-t).B_{0,n-1}(t)$ et $B_{n,n}(t)=t.B_{n-1,n-1}(t)$.

En fait, on peut écrire la relation de récurrence pour i=0,n avec la convention $B_{i,n}(t)=0$ pour i<0 ou i>n, que nous adaptons par la suite.

Les fonctions $B_{i,n}(t)$ possèdent les propriétés importantes suivantes dans l'intervalle I=[0, 1]:

- Non négativité: $B_{i,n}(t)\geq 0$ pour toutes les valeurs de i, n et t.
- Partition unité: $\sum_{i=0}^{n} B_{i,n}(t) = 1$. pour tout $t \in [0, 1]$
- Symétrie: $B_{i,n}(t)=B_{n-i,n}(1-t)$ pour tout $t \in [0, 1]$.

Extremum: $B_{i,n}(t)$ atteint exactement un seul maximum à t=i/n, c'est à dire: $\underset{t\in I}{\text{Max}}\, B_{i,n}(t) = B_{i,n}(i/n)$

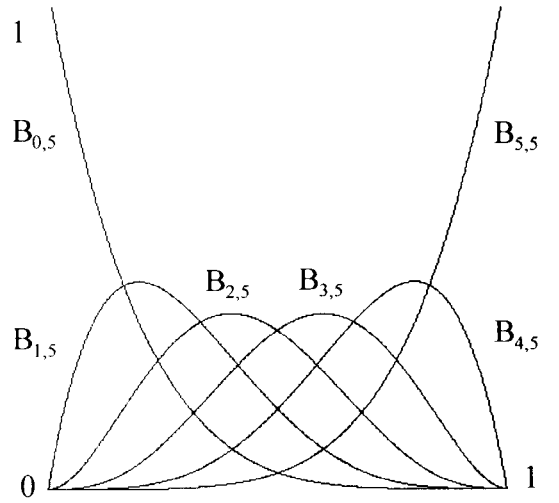

Figure I.1. Polynômes de Bernstein de degré 5

I.2.2. Courbes de Bézier non rationnelles

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

$$C(t) = \sum_{i=0}^{n} B_{i,n}(t).P_i \quad 0 \le t \le 1 \qquad (2)$$

où P_i sont les pôles de la courbe et $B_{i,n}(t)$ sont les fonctions de Bernstein de degré n définies par l'équation (1). Le degré de la courbe est égal au nombre de pôles moins un.

Les courbes de Bézier non rationnelles possèdent les propriétés géométriques principales suivantes: [*Fiorot '87*]

• Propriétés des frontières: La courbe passe par le premier et le dernier pôles en étant tangente au polygone avec: $P'(0)=n.(P_1-P_0)$ et $P'(1)=n.(P_n-P_{n-1})$.

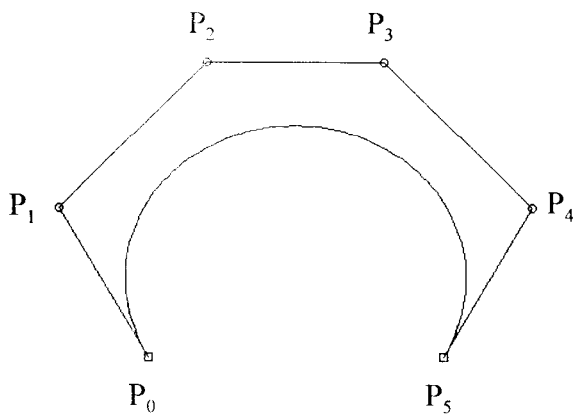

Figure I.2. Courbe de Bézier non rationnelle de degré 5

• Invariance affine: Une transformation affine est appliquée à la courbe si elle est appliquée aux pôles.

• Enveloppe convexe: La courbe de Bézier est toujours à l'intérieur du polygone convexe formé par les pôles.

• Comportement global: Le déplacement d'un pôle change la forme globale de toute la courbe. Le polygone formé par les P_i représente une approximation globale de la courbe.

• Propriété de la variation décroissante: Aucun plan n'a plus d'intersections avec la courbe qu'avec le polygone de contrôle.

• Propriété de symétrie: Lorsque nous changeons t en t-1, la courbe est parcourue dans le sens inverse.

• Propriété de l'hodographe: Si l'hodographe d'ordre r d'une courbe C(t) de degré n (r<n) est un point, alors il existe une courbe P(t) de degré r équivalente à C(t).

I.2.3. Courbes de Bézier rationnelles

Les coordonnées homogènes sont utilisées pour représenter les points de l'espace 3D en terme d'espace 4D. Un point non à l'infini dans l'espace 4D peut être représenté par les quatre nombres suivants: (wx,wy,wz,w), w>0, qui représente un point en coordonnées cartésiennes (x,y,z) qu'on représente par (x,y,z,1). Cette représentation est projection homogène qu'on peut définir par:

$$H\{(wx, wy, wz, w)\} = \begin{cases} (wx/w, wy/w, wz/w) & \text{si } w \neq 0 \\ \text{un point à l'infini sur} \\ \quad \text{la ligne partant} & \text{si } w = 0 \\ \text{vers le point } (x, y, z) \end{cases}$$

Les points de l'espace 4D sont notés P^w, et $P=H\{P^w\}$. En outre (x,y,z) et (x,y,z,1) sont considérés comme étant identiques.
Pour l'ensemble des pôles de l'espace 4D on a:

$$P_i^w = (w_i.x_i, w_i.y_i, w_i.z_i, w_i), \quad i = 0,n$$

On définit la courbe de Bézier non rationnelle 4D qui aura comme projective 3D ce qu'on appelle la courbe de Bézier rationnelle: [*Piegl '85, '86*]

$$C(t) = H\{C^w(t)\} = H\left\{\sum_{i=0}^{n} P_i^w . B_{i,n}(t)\right\} = \frac{\sum_{i=0}^{n} B_{i,n}(t).w_i.P_i}{\sum_{j=0}^{n} B_{j,n}(t).w_j}$$

$$= \sum_{i=0}^{n} G_{i,n}(t).P_i \qquad (3)$$

où $P_i = H\{P_i^w\} = (x_i, y_i, z_i)$ et $G_{i,n}(t) = \frac{B_{i,n}(t).w_i}{\sum_{j=0}^{n} B_{j,n}(t).w_j}$

où les $G_{i,n}(t)$ sont les fonctions de Bernstein rationnelles, les $w_i \geq 0$ pour i=0,n.

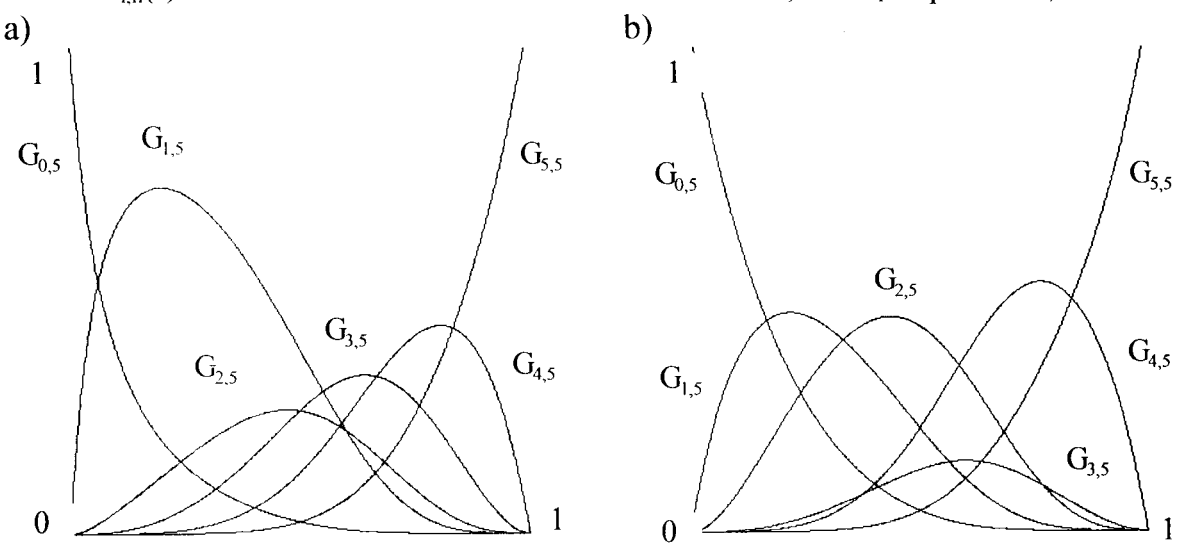

Figure I.3. (a) Polynômes rationnels de Bernstein de degré 5 avec (W_0, ..., W_5)=(1, 3, 1, 1, 1, 1); (b) Polynômes de Bernstein de degré 5 avec (W_0, ..., W_5)=(1, 1, 1, 0.3, 1, 1);

Les propriétés analytiques des fonctions $G_{i,n}(t)$ sont:

- Non négativité: $G_{i,n}(t) \geq 0$ pour toutes les valeurs de i, n et t.
- Partition unité: $\sum_{i=0}^{n} G_{i,n}(t) = 1$ pour tout $t \in [0, 1]$.
- Symétrie: $G_{i,n}(t) = G_{n-i,n}(1-t)$ pour tout $t \in [0, 1]$.

• Extremum: $G_{i,n}(t)$ atteint exactement un seul maximum à t=i/n, c'est à dire: $\underset{t\in I}{\text{Max}}\, G_{i,n}(t) = G_{i,n}(i/n)$.

• Les fonctions $G_{i,n}(t)$ sont généralisation des fonctions $B_{i,n}(t)$: $G_{i,n}(t)=B_{i,n}(t)$ si tous les $w_i=1$.

Ces propriétés proviennent de la définition des fonctions $G_{i,n}(t)$ et des propriétés analogues des fonctions $B_{i,n}(t)$.

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

• Propriétés des frontières: La courbe passe par le premier et le dernier pôles en étant tangente au polygone avec: $P'(0)=n.w_1.(P_1-P_0)/w_0$ et $P'(1)=n.w_{n-1}(P_n-P_{n-1})./w_n$

• Invariance affine: Une transformation affine est appliquée à la courbe si elle est appliquée aux pôles.

• Enveloppe convexe: La courbe de Bézier rationnelle est toujours à l'intérieur du polygone convexe formé par les pôles.

• Comportement global: Le déplacement d'un pôle change la forme globale de toute la courbe. Le polygone formé par les P_i représente une approximation globale de la courbe.

• Propriété de la variation décroissante: Aucun plan n'a plus d'intersections avec la courbe qu'avec le polygone de contrôle.

• Propriété de symétrie: Lorsque nous changeons t en t-1, la courbe est parcourue dans le sens inverse.

• Propriété de l'hodographe: Si l'hodographe d'ordre r d'une courbe C(t) de degré n (r<n) est un point, alors il existe une courbe P(t) de degré r équivalente à C(t).

• Si w_i croit la courbe s'approche du pôle P_i, si w_i décroît la courbe s'éloigne du pôle P_i, le rapprochement et l'éloignement d'un point C(t) en fonction de w_i est une droite passant par P_i.

a) b)

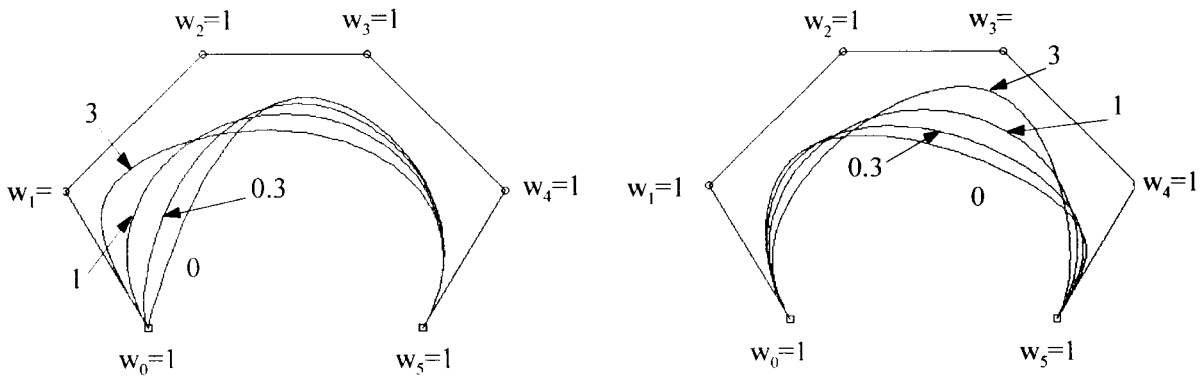

Figure I.4. Courbes de Bézier rationnelles de degré 5 obtenues en appliquant différentes valeurs à (a) w_1 (b) w_3.

I.2.4. Surfaces de Bézier non rationnelles

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

$$S(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u).B_{j,n}(v).P_{ij} \qquad (4)$$

où $0 \le u, v \le 1$ sont les paramètres curvilignes suivant les deux directions.
Les pôles P_{ij} sont arrangés dans une matrice rectangulaire appelée réseau de contrôle. Les fonctions $B_{i,m}(u)$et $B_{j,n}(v)$ les mêmes Polynômes de Bernstein définis par l'équation (1).

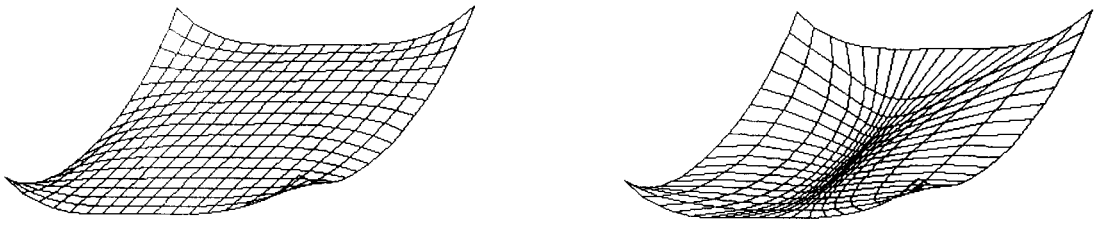

Figure I.5. (a) Surface de Bézier non rationnelles de degrés 3×4; (b) Surface de Bézier rationnelle de degrés 3×4 avec $w_{12}=w_{22}=20$.

La surface (4) représente un schéma de courbe bidirectionnel. On fixe un des paramètres, soit $u=u_0$, $0 \le u_0 \le 1$. Alors en variant v de 0 à 1 on construit une courbe de Bézier la surface, c'est à dire: [*Sablonnière '87-3*]

$$C(v) = S(u_0, v) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u_0).B_{j,n}(v).P_{ij}$$

$$= \sum_{j=0}^{n} B_{j,n}(v).\left(\sum_{i=0}^{m} B_{i,m}(u_0).P_{ij}\right) = \sum_{j=0}^{n} B_{j,n}(v).Q_j(u_0)$$

Les $Q_j(u_0)$ sont les n+1 pôles de la courbe C(v). En particulier, S(0,v), S(1,v), S(u,1), S(u,0) les quatre courbes délimitant la surface.

I.2.5. Surfaces de Bézier rationnelles

Une surface de Bézier rationnelle de degré (m, n) 3D est définie en 4D par:

$$S(u,v) = H\{S^w(u,v)\} = H\left\{\sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u).B_{j,n}(v).P_{ij}\right\}$$

$$= \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u).B_{j,n}(v).w_{ij}.P_{ij}}{\sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u).B_{j,n}(v).w_{ij}} = \sum_{i=0}^{m}\sum_{j=0}^{n} G_{i,m;j,n}(u,v).P_{ij} \qquad (5)$$

où les P_{ij} sont les pôles de la surface. Ce qui donne [*Piegl '85, '86*]

$$G_{i,m;j,n}(u,v) = \frac{B_{i,m}(u).B_{j,n}(v).w_{ij}}{\sum_{r=0}^{m}\sum_{s=0}^{n} B_{r,m}(u).B_{s,n}(v).w_{rs}}$$

Les poids $w_{ij} \geq 0$, pour toutes les valeurs de i et j.
Les $G_{i,m;j,n}(u,v)$ sont les fonctions de base rationnelle Bernstein bi-variables. Par convention, on omettra d'indiquer m et n, et la fonction s'écrira $R_{i,j}(u, v)$. On note que cette fonction n'est pas le produit des fonctions rationnelles pour les courbes, comme c'est le cas des courbes non rationnelles.

I.3. Courbes et surfaces B-spline

I.3.1. Fonction de base B-spline

On adopte ici la définition récursive connue sous le nom de l'algorithme de Cox- DeBoor: Soit $T = \{t_0, \ldots, t_i, t_{i+1}, \ldots, t_m\}$ une séquence croissante de nombres réels. La i^{ieme} fonction B-spline de degré p (d'ordre p+1) notée $N_{i,p}(t)$, est définie par:

$$N_{i,0}(t) = \begin{cases} 1 \text{ si } t_i \leq t < t_{i+1} \\ 0 \text{ sinon} \end{cases} \qquad (6)$$

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i}.N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}}.N_{i+1,p-1}(t)$$

avec la convention 0/0=0. [*Piegl '91*]
Les t_i sont appelés nœuds, et forment le vecteur nœuds T. Pour les formes non-uniformes et non- périodiques, le vecteur nœuds T prend la forme $T=\{\alpha, \alpha, \ldots, \alpha, t_{p+1}, \ldots, t_{m-p-1}, \beta, \beta, \ldots, \beta\}$, en pratique, et dans toutes les applications $\alpha=0$, $\beta=1$.

Les fonctions $N_{i,p}(t)$ ont les propriétés importantes suivantes: [*Piegl & Tiller '87*]

- Non négativité: $N_{i,p}(t) \geq 0$ quelque soient i, p et t.
- Partition unité: $\sum_{i=0}^{n} N_{i,p}(t) = 1 \ \forall \ t \in [t_0, t_m]$, n est le nombre total de fonctions B-spline de degré p pour un vecteur nœuds donné (n dépend de p et m).

- Support local: $N_{i,p}(t)=0$ si t est à l'extérieur de l'intervalle $[t_i,t_{i+p+1}]$.
- Dérivabilité: Toutes les dérivées des fonctions B-spline existent (par leur forme polynomiale). Pour un nœud de multiplicité k elles sont (p-k) continûment dérivable.
- Extremum: Excepté le cas p=0, les fonctions B-spline ont un seul maximum.

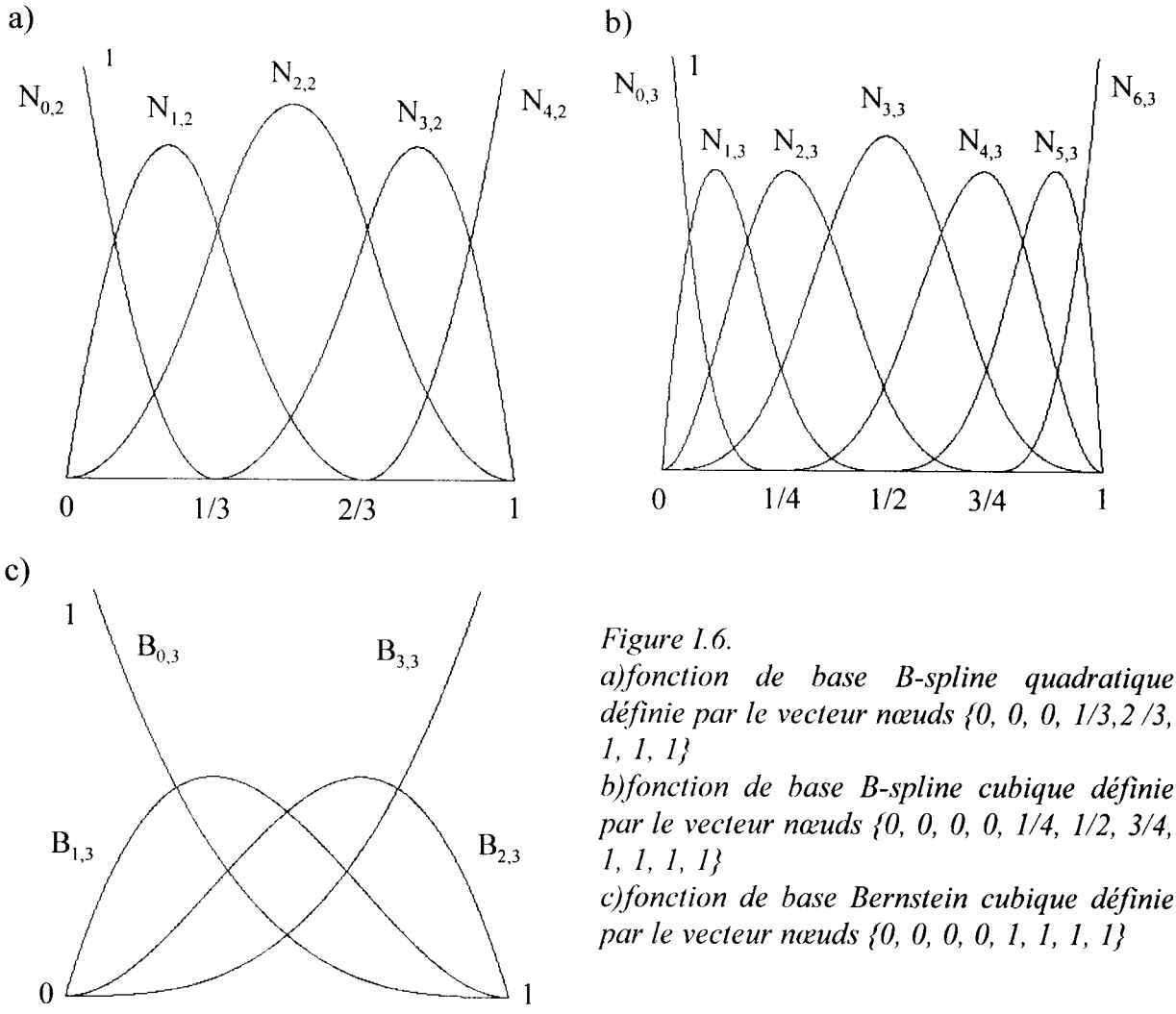

Figure I.6.
a)fonction de base B-spline quadratique définie par le vecteur nœuds {0, 0, 0, 1/3,2 /3, 1, 1, 1}
b)fonction de base B-spline cubique définie par le vecteur nœuds {0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1}
c)fonction de base Bernstein cubique définie par le vecteur nœuds {0, 0, 0, 0, 1, 1, 1, 1}

Le choix du vecteur nœuds T influe directement sur la forme de la fonction définie par l'équation (6). On considère plusieurs types de vecteurs nœuds. On suppose que p est fixé à l'avance, le vecteur nœuds $T=\{t_0,...,t_m\}$ est dit non périodique si le premier et le dernier nœuds ont une multiplicité p+1, c'est à dire
que $t_0=t_1=...=t_p$ et $t_{m-p}=t_{m-p+1}=...=t_m$. On considère pour la suite que $t_0=0$, et $t_m=1$ et que le vecteur nœuds est non périodique.
Les vecteurs nœuds ont la forme suivante:

$$T=\left\{\underbrace{0,\ldots,0}_{p+1},t_{p+1},\ldots,t_{m-p-1},\underbrace{1,\ldots,1}_{p+1}\right\}.$$

S'il existe un nombre réel positif d, tel que $t_{i+1}=t_i=d$ pour $p\le i\le m-p-1$ (nœuds proportionnellement espacés) alors T est un vecteur nœuds uniforme, sinon T est dit

non uniforme. L'utilisation des vecteurs nœuds non uniformes répond à un contrôle meilleur des formes et à une modélisation d'une large classe de formes que les vecteurs nœuds uniformes.

Si $T = \left\{ \underbrace{0,\ldots,0}_{p+1}, \underbrace{1,\ldots,1}_{p+1} \right\}$, les fonctions de base B-spline deviennent des fonctions de base Bernstein: $N_{i,p}(t)=B_{i,p}(t)$. (voir Figure I.6.c).

I.3.2. Courbes B-spline non rationnelles

Une courbe B-spline non rationnelle de degré p est définie par:

$$C(t) = \sum_{i=0}^{n} P_i . N_{i,p}(t) \quad 0 \le t \le 1 \tag{7}$$

où P_i sont les pôles de la courbe et $N_{i,p}(t)$ sont les fonctions de base B-spline de degré p définies par l'équation (6) et le vecteur nœuds (non périodique) $T=\{t_0,\ldots,t_m\}$. Le nombre de nœuds, le degré, et le nombre de pôles sont liés par la relation: $m=n+p+1$.
Les courbes B-spline possèdent des propriétés géométriques très utiles, qui découlent de la définition (7) et des propriétés analytiques des fonctions $N_{i,p}(t)$ décrites précédemment: [*Piegl '91*], [*Piegl & Tiller '87*]

- La multiplicité (p+1) des nœuds extrêmes engendre ces conditions aux extrémités de la courbe: $C(0)=P_0$, $C(1)=P_n$, $C'(0)=(P_1-P_0)/t_{p+1}$, et $C'(1)=p.(P_n-P_{n-1})/(1-t_{m-p-1})$.
- Invariance affine: Une transformation affine est appliquée à la courbe si elle est appliquée aux pôles.
- Enveloppe convexe locale: La courbe est contenue dans une enveloppe convexe des pôles, c'est à dire si $t\in[t_j,t_{j+1}]$ et $p\le j<m-p-1$, alors C(t) est dans l'enveloppe convexe des pôles P_{j-p}, P_j.
- Le polygone formé par les pôles P_j représente une approximation linéaire par morceaux de la courbe.

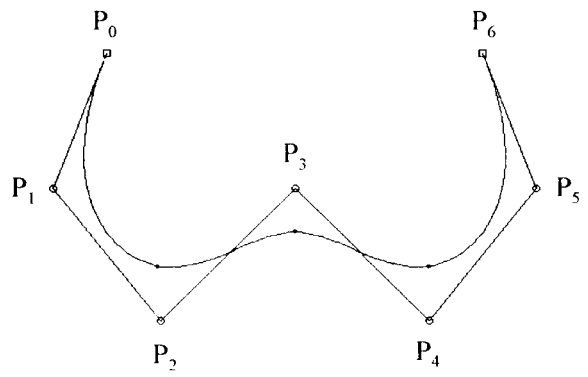

Figure I.7. Courbe B-spline cubique non rationnelle, avec son polygone de contrôle associé aux fonctions de base montrée en Figure I.6. (b).

- Comportement local: Si on déplace un pôle, la courbe sera localement déformée et affecte la courbe dans p+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 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 k, elle est p-k fois continûment dérivable.

- Propriété de la variance décroissante: Aucun plan n'a plus d'intersections avec la courbe qu'avec le polygone de contrôle.
- Type Bézier: Une courbe B-spline sans nœuds internes est une courbe de Bézier.

I.3.3. Courbes B-spline rationnelles

Pour l'ensemble des pôles de l'espace 4D on a:

$$P_i^w = (w_i . x_i, w_i . y_i, w_i . z_i), \quad i = 0, n$$

On définit la courbe B-spline non rationnelle 4D qui aura comme projective 3D ce qu'on appelle la courbe B-spline rationnelle: [*Piegl '91*], [*Piegl & Tiller '87*]

$$C(t) = H\{C^w(t)\} = H\left\{\sum_{i=0}^{n} P_i^w . N_{i,p}(t)\right\} = \frac{\sum_{i=0}^{n} N_{i,p}(t).w_i.P_i}{\sum_{j=0}^{n} N_{j,p}(t).w_j}$$

$$= \sum_{i=0}^{n} R_{i,p}(t).P_i \qquad (8)$$

où $P_i = H\{P_i^w\} = (x_i, y_i, z_i)$ et $R_{i,p}(t) = \dfrac{N_{i,p}(t).w_i}{\sum_{j=0}^{n} N_{j,p}(t).w_j}$

où les $R_{i,p}(t)$ sont les fonctions de base des courbes B-spline rationnelles, les $w_i \geq 0$ pour i=0,n.

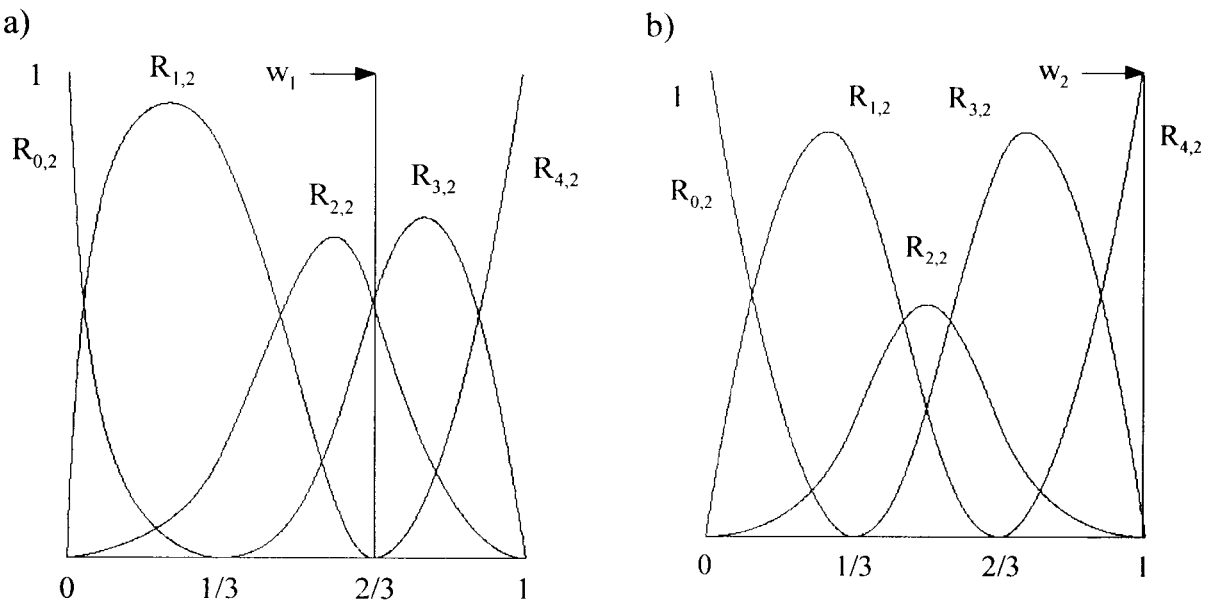

Figure I.8. Fonctions de base rationnelles quadratiques: (a) T={0, 0, 0, 1/3, 2/3, 1, 1, 1} et (w_0, ..., w_4)=(1, 4, 1, 1, 1); (b) T={0, 0, 0, 1/3, 2/3, 1, 1, 1} et (w_0, ..., w_4)=(1, 1, 0.3, 1, 1)

Les propriétés analytiques des fonctions $R_{i,p}(t)$ sont:

- Non négativité: $R_{i,p}(t) \geq 0$ quelque soient i,p et t.
- Partition unité: $\sum_{i=0}^{n} R_{i,p}(t) = 1 \ \forall \ t \in [t_0, t_m]$
- Support local: $R_{i,p}(t)=0$ si t est à l'extérieur de l'intervalle $[t_i, t_{i+p+1}]$.
- Dérivabilité: Toutes les dérivées des fonctions $R_{i,p}(t)$ existent. Pour un nœud de multiplicité k elles sont (p-k) continûment dérivable.
- Extremum: Excepté le cas p=0, les fonctions $R_{i,p}(t)$ ont un seul maximum.

Les fonctions $R_{i,p}(t)$ sont une généralisation des fonctions non rationnelles B-spline et des fonctions rationnelles et non rationnelles Bernstein: $R_{i,p}(t)=N_{i,p}(t)$ si tous les $w_i=1$,

$R_{i,p}(t)=B_{i,p}(t)$ si $T = \left\{ \underbrace{0,\ldots,0}_{p+1}, \underbrace{1,\ldots,1}_{p+1} \right\}$

Les preuves de ces propriétés découlent de la définition de la fonction $R_{i,p}(t)$ et des propriétés analogues de la fonction $N_{i,p}(t)$

Comme les fonctions $R_{i,p}(t)$ ont globalement les mêmes propriétés que les fonctions $N_{i,p}(t)$, les courbes B-spline rationnelles doivent particulièrement avoir les mêmes propriétés que les courbes B-spline non rationnelles, et, en effet cela est le cas.
Ces propriétés sont (se référer aux paragraphes précédents et aux figures I.9 et I.10):

- Les conditions aux extrémités de la courbe sont:

$$C(0) = P_0 = P_0^w / w_0 \,, C(1) = P_n = P_n^w / w_n$$
$$C'(0) = \left(P_1^w . (P_1 - P_0)\right) / \left(w_0 t_{p+1}\right)$$
$$C'(1) = \left(P_{n-1}^w \left(P_n - P_{n-1}\right)\right) / \left(w_n \left(1 - t_{m-p-1}\right)\right)$$

- Invariance affine: Une transformation affine est appliquée à la courbe si elle est appliquée aux pôles.

a) b)

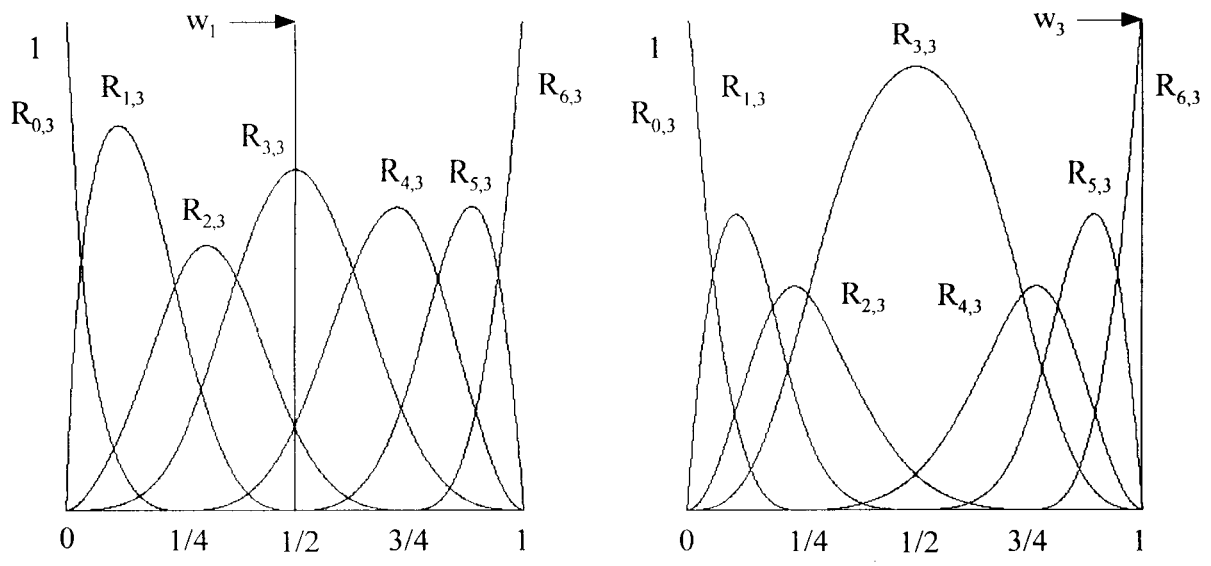

Figure I.9. Fonctions de base B-spline rationnelles cubiques: (a) T={0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1} et (w_0, ..., w_6)=(1, 2, 1, 1, 1, 1, 1); (b) T={0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1} et (w_0, ..., w_6)=(1, 1, 1, 3, 1, 1, 1)

• Enveloppe convexe locale: La courbe est contenue dans une enveloppe convexe des pôles, c'est à dire si $t\in[t_j,t_{j+1}]$ et $p\leq j<m-p-1$, alors C(t) est dans l'enveloppe convexe des pôles P_{j-p},P_j.

• Le polygone formé par les pôles P_j représente une approximation linéaire par morceaux de la courbe.

• Comportement local: Si on déplace un pôle, la courbe sera localement déformée et affecte la courbe dans les p+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 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 k, elle est p-k fois (non nécessairement continûment) dérivable.

• Propriété de la variance décroissante: Aucun plan n'a plus d'intersections avec la courbe qu'avec le polygone de contrôle.

• Type Bézier: Une courbe B-spline rationnelle sans nœuds internes est une courbe de Bézier rationnelle. Les courbes B-spline rationnelles sont une généralisation des courbes B-spline non rationnelles et des courbes de Bézier rationnelles et non rationnelles.

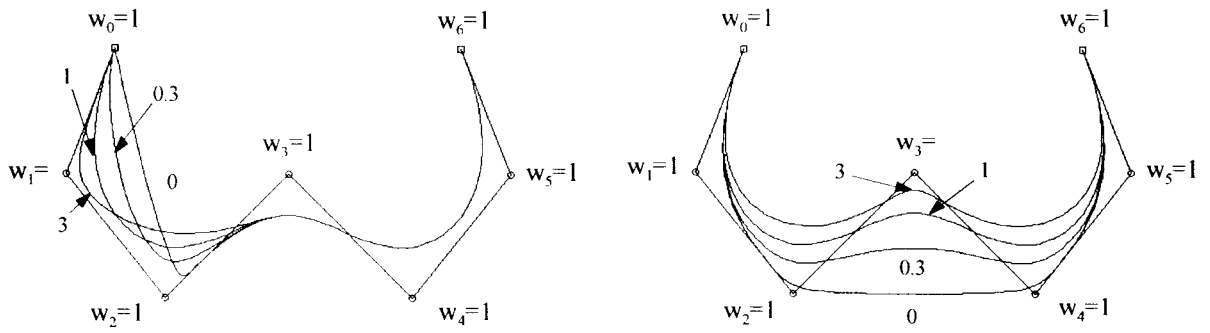

Figure I.10. Courbes B-spline rationnelles cubiques

On constate que les courbes B-spline rationnelles et non rationnelles possèdent globalement les mêmes propriétés géométriques. De plus, la théorie et les algorithmes de calcul des courbes non rationnelles sont facilement extensibles à ceux des courbes rationnelles.

Les w_i dans l'équation (8) sont appelés poids. La figure I.11 montre que les poids sont des paramètres de changement de forme. Ceci est remarquable dans les figures I.9 et I.10: [*Piegl '91*], [*Piegl & Tiller '87*]

• Les w_i affectent la courbe localement. plus précisément fixons les w_j pour tout $j\neq i$, le changement de w_i n'affectera la courbe que pour $t\in[t_i,t_{i+p+1}]$.

• Pour une valeur de t fixée, $0<t<1$, si le w_i croit alors $R_{i,p}(t)$ croit et les $R_{j,p}(t)$, $j\neq i$ décroissent (comparer les Figures I.7.(a), (b), I.9.(a), (b) et I.10.(a), (b)). De plus la courbe tend vers le pôle P_i. Si w_i décroît la courbe s'éloigne du pôle P_i.

Les poids ont un sens géométrique (3D) clair, et leur effet de rapprochement-éloignement sur la courbe peut être quantifié avec précision. Pour $0<i<n$ et w_j fixés pour tout $j\neq i$, on note par $C[w_i](t)$ une courbe parmi une famille paramétrique des courbes B-spline générées par les w_i variant ente $0<w_i<\infty$, soit $t\in[t_i,t_{i+p+1}]$. Alors pour toutes valeurs des w_i, le point $C[w_i](t)$ est lié par un segment passant par $P_i.C[w_i](t)$ quand $w_i\to\infty$. Alors quand les w_i varient, le point $C[w_i](t)$ balaye le segment passant par P_i.

I.3.4. Surfaces B-spline non rationnelles

Une surface B-spline non rationnelle de degré (p,q) est définie par:

$$S(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u).N_{j,q}(v).P_{ij} \qquad (9)$$

où $0\le u,v\le 1$ sont les paramètres curvilignes suivant les deux directions.
Les pôles P_{ij} sont arrangés dans une matrice rectangulaire appelée réseau de contrôle. Les fonctions $N_{i,p}(u)$ et $N_{j,q}(v)$ sont les mêmes fonctions de base B-spline définies par l'équation (1). U et V forment les vecteurs nœuds non uniformes et non périodiques relatifs aux fonctions $N_{i,p}(u)$ et $N_{j,q}(v)$ respectivement. la Figure I.12 représente une surface B-spline non rationnelle de degré (2,3).
La surface (9) représente un schéma de courbe bidirectionnel. On fixe un des paramètres, soit $u=u_0$, $0\le u_0\le 1$. Alors en variant v de 0 à 1 on construit une courbe B-spline sur la surface, c'est à dire: [*Sablonnière '87-3*]

$$C(v)=S(u_0,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u_0).N_{j,q}(v).P_{ij}$$

$$=\sum_{j=0}^{n}N_{j,q}(v).\left(\sum_{i=0}^{m}N_{i,p}(u_0).P_{ij}\right)=\sum_{j=0}^{n}N_{j,q}(v).Q_j(u_0)$$

Les $Q_j(u_0)$ sont les n+1 pôles de la courbe C(v). En particulier, S(0,v), S(1,v), S(u,1), S(u,0) les quatre courbes délimitant la surface.
Les (m+1)(n+1) produits de fonctions $N_{i,p}(u).N_{j,q}(v)$, de l'équation (9) sont appelés fonctions de base B-spline bi-variable de degré (p,q).

Ces fonctions ont les propriétés analytiques suivantes: [*Piegl '91*], [Piegl & Tiller '87]

- Non négativité: $N_{i,p}(u).N_{j,q}(v)\ge 0$ pour toutes valeurs de i, j, p, q, u et v.
- Partition unité: $\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u).N_{j,q}(v)=1$ pour tout couple $(u,v)\in[0,1]\times[0,1]$.
- Support local: $N_{i,p}(u).N_{j,q}(v)=0$ si (u,v) est à l'extérieur du rectangle $[u_i, u_{i+p+1}]\times[v_j, v_{j+q+1}]$.
- Dérivabilité: à l'intérieur des rectangles formés par u et v lignes nodales, toutes les dérivées partielles de $N_{i,p}(u).N_{j,q}(v)$ existent. Au nœud u (nœud v) elle est p-k (q-k) fois continûment dérivable suivant la direction u (direction v), où k est l'ordre de multiplicité du nœud.
- Extremum: si $p>0$ et $q>0$, alors $N_{i,p}(u).N_{j,q}(v)$ atteint exactement un seul maximum.

Excepté la propriété de la variation décroissante les propriétés des fonctions surfaciques de base B-spline non rationnelles sont analogues à celles des courbes.

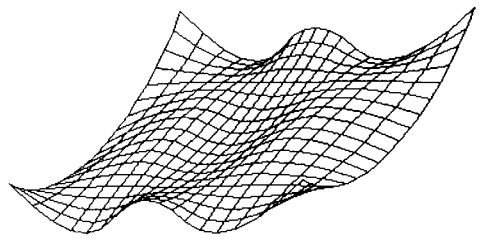

Figure I.11. Surface B-spline non rationnelle de degré (2,3). Les vecteurs nœuds sont: U={0, 0, 0, 1/3, 2/3, 1, 1, 1} et V={0, 0, 0, 0, 1, 1, 1}

- Les quatre pôles extrêmes du réseau de contrôle coïncident avec les quatre points extrêmes de la surface, par exemple $S(0,0)=P_{00}$. De plus, les dérivées partielles par rapport à u ou v (mais pas les dérivée mixtes) coïncident avec les dérivées des points de la courbe, par exemple $S_u(0,0)=p.(P_{10}-P_{00})/u_{p+1}$.
- Invariance affine.
- Enveloppe convexe solide: Si $u\in[u_r, u_{r+1}]$ et $v\in[v_s, v_{s+1}]$, alors S(u,v) est contenue dans l'enveloppe convexe formée par les pôles P_{ij}, i=r-p,...,r et j=s-q,...,s.
- Approximation locale: Le déplacement d'un pôle P_{ij} n'affecte la surface que dans le rectangle $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$.
- S(u,v) est p-k (q-k) fois continûment dérivable suivant la direction u (direction v) pour nœud suivant u (suivant v) de multiplicité k.
- Type Bézier: Une surface B-spline non rationnelle sans nœuds internes est une surface de Bézier non rationnelle.

I.3.5. Surfaces B-spline rationnelles

Une surface B-spline rationnelle de degré (p,q) 3D est définie en 4D par:

$$S(u,v) = H\left\{S^w(u,v)\right\} = H\left\{\sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,p}(u).N_{j,q}(v).P_{ij}\right\}$$

$$= \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,p}(u).N_{j,q}(v).w_{ij}.P_{ij}}{\sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,p}(u).N_{j,q}(v).w_{ij}} = \sum_{i=0}^{m}\sum_{j=0}^{n} R_{i,p;j,q}(u,v).Pij \qquad (10)$$

où les P_{ij} sont les pôles de la surface. Ce qui donne

$$R_{i,p;j,q}(u,v) = \frac{N_{i,p}(u).N_{j,q}(v).w_{ij}}{\sum_{r=0}^{m}\sum_{s=0}^{n} N_{r,p}(u).N_{s,q}(v).w_{rs}}$$

Les $R_{i,p;j,q}(u,v)$ sont les fonctions de base rationnelle B-spline bi-variables. Par convention, on omettra d'indiquer p et q, et la fonction s'écrira $R_{i,j}(u,v)$. On note que cette fonction n'est pas le produit des fonctions rationnelles pour les courbes, comme c'est le cas des courbes non rationnelles.

Les fonctions $R_{i,j}(u,v)$ ont une forme similaire aux fonctions $N_{i,p}(u).N_{j,q}(v)$ et ont les mêmes propriétés analytiques:

- Non négativité: $R_{i,j}(u,v) \geq 0$ pour toutes valeurs de i, j, u et v.
- Partition unité: $\sum_{i=0}^{m}\sum_{j=0}^{n} R_{i,j}(u,v) = 1$ pour tout couple $(u, v) \in [0, 1]\times[0, 1]$.
- Support local: $R_{i,j}(u,v)=0$ si (u, v) est à l'extérieur du rectangle $[u_i, u_{i+p+1}]\times[v_j, v_{j+q+1}]$.
- Dérivabilité: à l'intérieur des rectangles formés par u et v lignes nodales, toutes les dérivées partielles de $R_{j}(u, v)$ existent. Au nœud u (nœud v) elle est p-k (q-k) fois continûment dérivable suivant la direction u (direction v), où k est l'ordre de multiplicité du nœud.
- Extremum: si $p>0$ et $q>0$, alors $R_{i,j}(u, v)$ atteint exactement un seul maximum.
- Les fonctions $R_{i,j}(u,v)$ sont une généralisation des fonctions de base B-spline bi-variables, c'est à dire $R_{i,j}(u, v)=N_{i,p}(u).N_{j,q}(v)$ si tous les $w_{ij}=1$.

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

- Les quatre pôles extrêmes du réseau de contrôle coïncident avec les quatre points extrêmes de la surface. De plus, les dérivées partielles par rapport à u ou v coïncident avec les dérivées des points de la courbe.
- Invariance affine.
- Enveloppe convexe solide identique au cas non rationnel.

- Approximation locale identique au cas non rationnel.
- S(u, v) est p-k (q-k) fois continûment dérivable suivant la direction u (direction v) pour un nœud suivant u (suivant v) de multiplicité k.

Une surface B-spline rationnelle sans nœuds internes est une surface de Bézier rationnelle. Les surfaces B-spline rationnelles sont une généralisation des surfaces B-spline non rationnelles et des surfaces de Bézier rationnelles et non rationnelles.

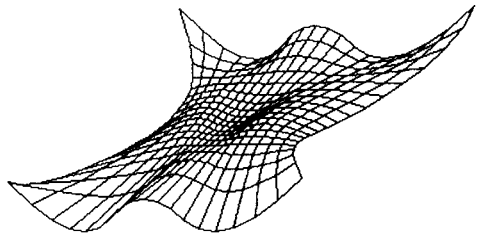

Figure I.12. Surface B-spline rationnelle de même réseau de contrôle et mêmes vecteurs nœuds que la Figure I.11 avec $w_{il}=3$.

Comme dans le cas des courbes, les poids w_{ij} sont des paramètres de modification de la forme. Ils affectent la surface localement et ont un effet de rapprochement- éloignement qui peut être quantifié.

Algorithmique des transformations

II.1.	**Transformations dans le modèle de Bézier**	18
II.1.1.	Subdivision	18
II.1.1.1.	Courbes non rationnelles	18
II.1.1.2.	Courbes rationnelles	18
II.1.2.	Elévation de degré	20
II.1.2.1.	Courbes non rationnelles	20
II.1.2.2.	Courbes rationnelles	21
II.1.3.	Réduction de degré	21
II.1.3.1.	Courbes non rationnelles	21
II.1.3.2.	Courbes rationnelles	22
II.1.4.	Raccordement des courbes	23
II.1.4.1.	Courbes non rationnelles	23
II.1.4.2.	Courbes rationnelles	24
II.2.	**Transformations dans le modèle B-spline**	24
II.2.1.	Insertion de nœuds	24
II.2.1.1.	Courbes non rationnelles	24
II.2.1.2.	Courbes rationnelles	25
II.2.2.	Suppression de nœuds	27
II.2.2.1.	Courbes non rationnelles	27
II.2.2.2.	Courbes rationnelles	29
II.2.3.	Subdivision et l'algorithme d'Oslo	30
II.2.3.1.	Courbes non rationnelles	30
II.2.3.2.	Courbes rationnelles	32
II.3.	**Changement de bases**	34
II.3.1.	B-spline- Bézier	34
II.3.1.1.	Le polygone local de Bézier d'une courbe B-spline	34
II.3.1.1.	Les polygones de Bézier d'une courbe B-spline	35
II.3.2.	Bézier- B-spline	35
II.3.2.1.	Polygone B-spline d'une courbe de Bézier	35
II.3.2.2.	Polygone B-spline des courbes locales de Bézier	36
II.3.2.3.	Algorithme WM	36

II. Algorithmique des transformations

II.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 subréduction.

II.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.

II.1.1.1. Subdivision des courbes non rationnelles

Une courbe de Bézier C(t) de degré n définie par ses (n+1) pôles $[P_0, ..., P_n]$, 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 $R(1)=Q(0)=C(t_0)$. En utilisant le polygone $[P_0, ..., P_n]$ et l'algorithme de récurrence: [*Bensalah '90*], [*Farouki & Neef '90*]

en posant $C_i^0(t) = P_0$

$$C_i^k(t) = (1-t).C_i^{k-1}(t) + t.C_{i+1}^{k-1}(t) \qquad (11)$$

les deux polygones $[R_0, ..., R_n]$ et $[Q_0, ..., Q_n]$ seront:

$R_i = C_0^i(t_0)$ $\quad i = 0, \ldots, n$

Exemple pour n=4

$C_0^0 = R_0$

$C_1^0 \rightarrow C_0^1 = R_1$

$C_2^0 \rightarrow C_1^1 \rightarrow C_0^2 = R_2$

$C_3^0 \rightarrow C_2^1 \rightarrow C_1^2 \rightarrow C_0^3 = R_3$

$C_4^0 \rightarrow C_3^1 \rightarrow C_2^2 \rightarrow C_1^3 \rightarrow C_0^4 = R_4 = Q_0$

$C_3^0 \rightarrow C_2^1 \rightarrow C_1^2 \rightarrow C_0^3 = Q_1$

$C_2^0 \rightarrow C_1^1 \rightarrow C_0^2 = Q_2$

$C_1^0 \rightarrow C_0^1 = Q_3$

$C_0^0 = Q_4$

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

Les $C_p^q(t_0)$ seront calculés en utilisant l'équation de récurrence (11).

II.1.1.2. Subdivision des courbes rationnelles

Une courbe de Bézier rationnelle C(t) de degré n définie par ses (n+1) pôles $[P_0, ..., P_n]$ chargés respectivement par les poids $(w_0, ..., w_n)$ peut être subdivisée à t_0 en deux courbes de Bézier rationnelles R(t) et Q(t) toutes les deux de degré n et ayant le point commun $R(1)=Q(0)=C(t_0)$. En utilisant le polygone $[P_0, ..., P_n]$ les charges $(w_0, ..., w_n)$ et l'algorithme de récurrence (13): [*Bensalah '90*]

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

avec $w_i^r(t) = (1-t).w_i^{r-1}(t) + t.w_{i+1}^{r-1}(t)$ (12)

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

$$C_i^r(t) = (1-t).\frac{w_i^{r-1}(t)}{w_i^r(t)}.C_i^{r-1}(t) + t.\frac{w_{i+1}^{r-1}(t)}{w_i^r(t)}.C_{i+1}^{r-1}(t) \quad (13)$$

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

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

$R_i = C_0^i(t_0)$, et $\alpha_i = w_0^i(t_0)$ i=0, ...,n

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

Les $C_p^q(t_0)$ seront calculés par l'équation (13) et les $w_p^q(t_0)$ par l'équation (12).

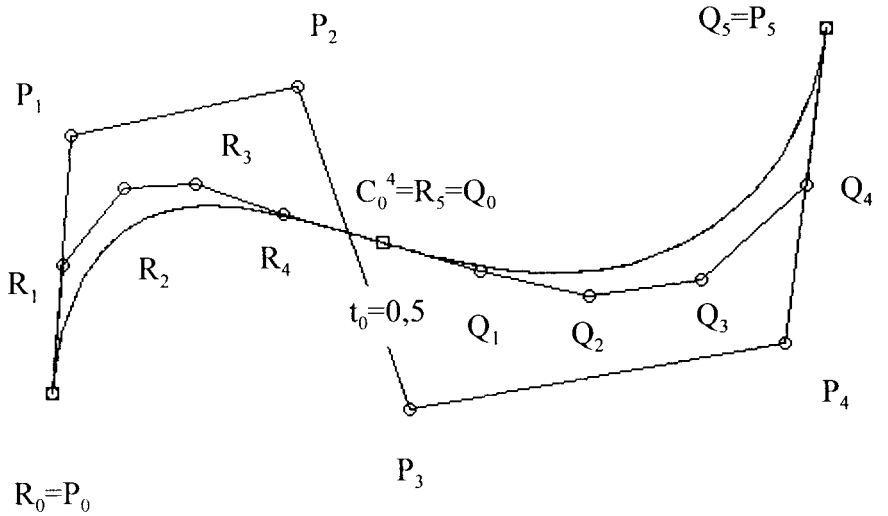

Figure II.1. Courbe de Bézier non rationnelle P(t) de degré 5 subdivisée en deux courbes R(t) et Q(t) de même degré

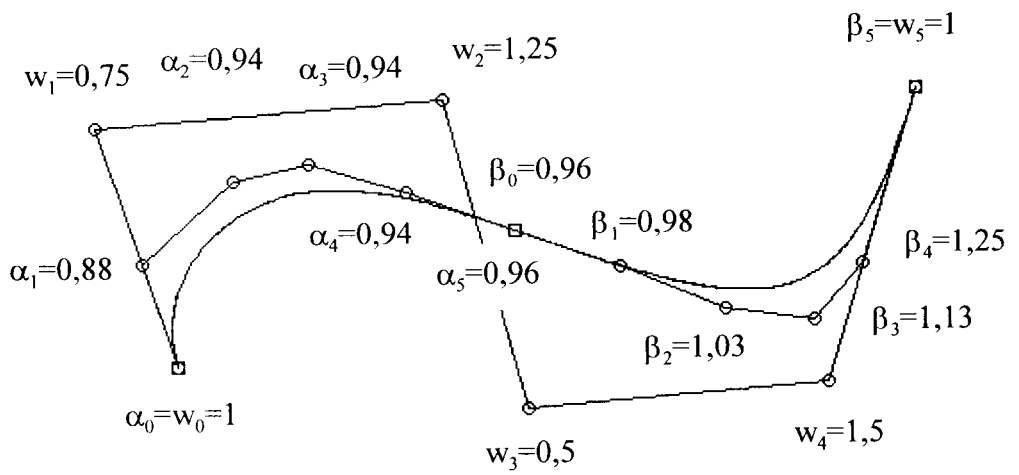

Figure II.2. Courbe de Bézier rationnelle de degré 5 (de poids w_i) subdivisée en deux courbes rationnelles de degré 5 (de poids α_j et β_k)

II.1.2. Elévation de degré

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

II.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,...,n). La courbe Q(t) de degré (n+1) est définie par ses (n+2) pôles Q_j (j=0,...,n+1) déterminés par la formule récurrente suivante: [*Bensalah '90*], [*Piegl & Tiller '94*]

$$Q_i = (1-\alpha_i).P_i + \alpha_i.P_{i-1}$$

$$\text{avec } \alpha_i = i/(n+1) \ (i=0,...,n+1) \qquad (14)$$

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 à n+2 en appliquant une nouvelle fois l'algorithme.
l'algorithme d'élévation de degré de n à n+p s'écrit alors de la manière suivante

(i=0,...,n)
    $Q_i=P_i$;
(j=1,...,p)
    $Q_{n+j}=Q_{n+j-1}$;
    (k=n+j-1,...,1 par -1)
        $Q_k=\alpha_{j,k}.Q_{k-1}+(1-\alpha_{j,k}).Q_k$;
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 un fichier.

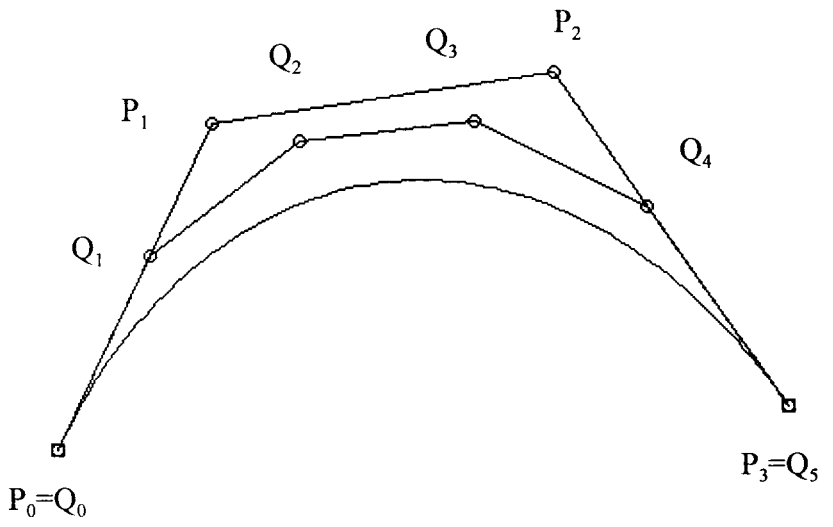

Figure II.3. Courbe de Bézier non rationnelle de degré 3 élevée au degré 5

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

$$Q_i = C^1_{i-1}\left(\frac{n+1-i}{n+1}\right); \quad i=0,...,n+1 \tag{15}$$

$$\beta_i = w^1_{i-1}\left(\frac{n+1-i}{n+1}\right); \quad i=0,...,n+1 \tag{16}$$

Les $C^q_p(t_0)$ seront calculés par l'équation (13) et les $w^q_p(t_0)$ par l'équation (12).
L'élévation de degré à l'ordre supérieur se fait par application répétée de cet algorithme.

II.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.

II.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,...,n+1), est mathématiquement identique à une courbe Q(t) de degré n définie par ses (n+1) pôles Q_j (j=0,...,n) alors la propriété de l'hodographe est vérifiée. Les pôles Q_j peuvent être calculés par la résolution inverse de l'équation d'élévation de degré (14). Cette résolution peut se faire de deux façons (17.a) et (17.b) selon que la résolution commence par Q_0 ou par Q_n: [*Farin '83*], [*Forrest '72*]

$$R_i = (1+\beta_i).P_i - \beta_i.R_{i-1}$$
$$\text{avec } \beta_i = i/(n+1-i) \ (i=0,...,n) \tag{17.a}$$
$$S_{i-1} = (1+\gamma_i).P_i - \gamma_i.S_i$$
$$\text{avec } \gamma_i = 1/\beta_i = (n+1-i)/i \ (i=n,...,0) \tag{17.b}$$

Si la propriété de l'hodographe n'est pas vérifiée alors (17.a) et (17.b) donnent deux polygones R_j et S_j différents.
Une première approximation à été proposée par Farin [*Farin '83*] et consiste à prendre pour la courbe réduite Q(t) la moyenne pondérée des deux polygones R et S:

$$Q_i = \frac{n-i}{n}.R_i + \frac{i}{n}.S_i; \quad i=0,...,n \tag{18}$$

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.

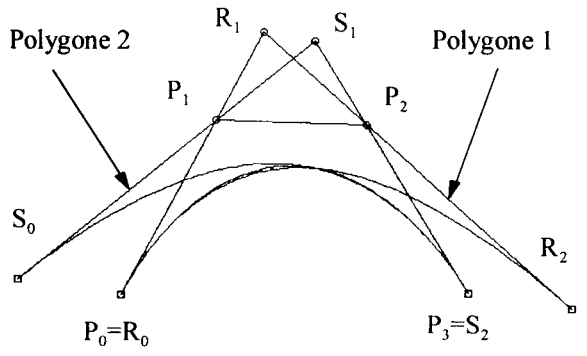

Figure II.4.a. Courbe de Bézier non rationnelle de degré 3 réduite au degré 2 par les deux méthodes polygone 1 et 2

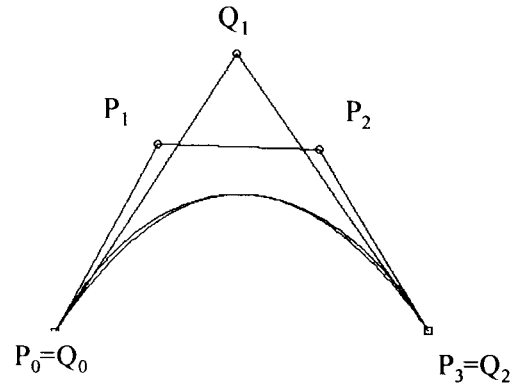

Figure II.4.b. Courbe précédente de degré 3 réduite par la méthode du polygone moyen (première approximation)

II.1.3.2. Réduction de degré des courbes rationnelles

En reprenant le même raisonnement que pour les courbes non rationnelles, les équations (15) et (16) peuvent être résolues de deux manières: [*Farin '83*]

i=0,...,n

$$R_i = \frac{n+1}{n+1-i}.\frac{w_i}{\alpha_i}.P_i - \frac{\alpha_{i-1}}{\alpha_i}.\frac{i}{n+1-i}.R_{i-1} \qquad (19.a)$$

$$\alpha_i = \frac{n+1}{n+1-i}.w_i - \frac{i}{n+1-i}.\alpha_{i-1} \qquad (20.a)$$

i=n,...,0

$$S_{i-1} = \frac{n+1}{i}.\frac{w_i}{\beta_{i-1}}.P_i - \frac{\beta_i}{\beta_{i-1}}.\frac{n+1-i}{i}.S_i \qquad (19.b)$$

$$\beta_{i-1} = \frac{n+1}{i}.w_i - \frac{n+1-i}{i}.\beta_i \qquad (20.b)$$

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

$$Q_i = \frac{n-i}{n}.\frac{\alpha_i}{\mu_i}.R_i - \frac{\beta_i}{\mu_i}.\frac{i}{n}.S_i \qquad (21.a)$$

$$\mu_i = \frac{n-i}{n}.\alpha_i - \frac{i}{n}.\beta_i \qquad (21.b)$$

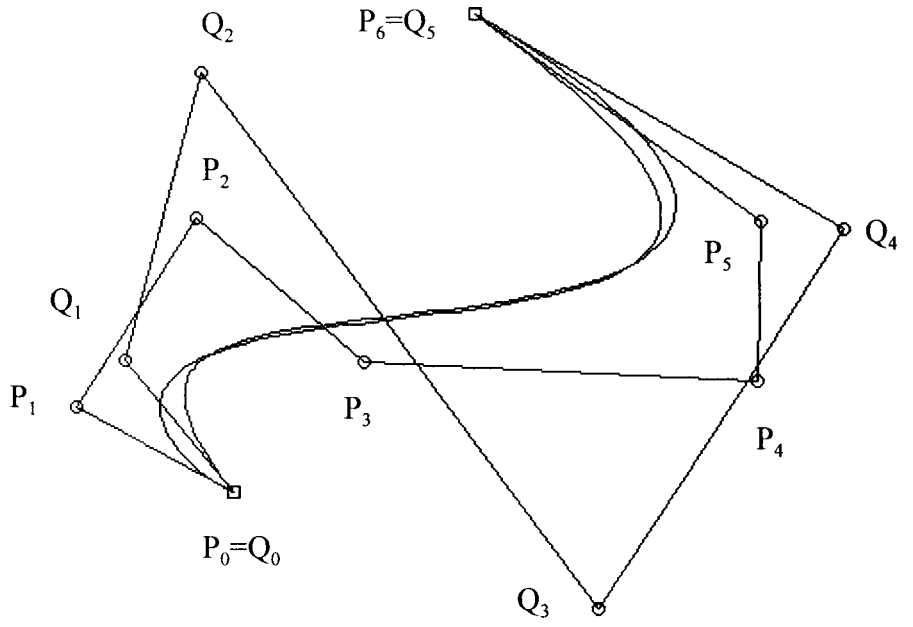

Figure II.5. Courbe de Bézier rationnelle de degré 6 réduite au degré 5 par la méthode du polygone moyen W={1,1.2,0.8,0.5,0.8,1.2,1}; W'={1,1.304,0.516,0.516,1.304,1}

II.1.4. Raccordement des courbes

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

II.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 à satisfaire. Pour avoir une continuité géométrique C^k (d'ordre k) entre deux courbes R(t) définie par le polygone (R_0, ...,R_n) et S(t) définie par le polygone (S_0,...,S_n) , il suffit de satisfaire les (k+1) conditions:

$$\left[\frac{d}{dt}\right]^i R(1) = \left[\frac{d}{dt}\right]^i S(0); \quad i=0,...,k\leq n \tag{22}$$

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) premiers pôles, il suffit seulement de calculer les (k+1) premiers pôles de S(t) en fonction 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

$$S_i = R^i_{n-i}(2); \; i=0,...,k \tag{23}$$

Il s'agit ici 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 ordre, l'extrapolation s'appliquera après les avoir mise au 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 rationnelle de Bézier R(t) définie par ses pôles R_i, et ses 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 point 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,...,k

$$S_i = R^i_{n-i}(2) \qquad (24.a)$$

$$\beta_i = \alpha^i_{n-i}(2) \qquad (24.b)$$

II.2. Transformations dans le modèle B-spline

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

II.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.

II.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:

$$C(t) = \sum_{i=0}^{n} P_i . N_{i,p}(t) \quad 0 \le t \le 1$$

où P_i sont les pôles de la courbe et $N_{i,p}(t)$ sont les fonctions de base B-spline de degré p définies par l'équation (6) et le vecteur nœuds (non périodique) $T=\{t_0,...,t_m\}$. Le nombre de nœuds, le degré, et le nombre de pôles sont liés par la relation: m=n+p+1.
Soit $x_r \le ... \le x_s$ une séquence de nœuds à insérer dans l'intervalle $[t_a, t_b]$ c'est à dire $t_a \le x_r$ et $x_s \le t_b$. Les nœuds de multiplicité k seront insérés k fois (la multiplicité ne doit pas dépasser n-1).
Pour insérer ces nœuds les pôles P_{a-p}, ...,P_{b-1} doivent être connus. Le nouveau vecteur nœuds sera: $U=\{u_0, ..., u_{m+s-r}\}$, 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 & Prautzch '85*]

Algorithme 1

i=b+p-1; l=b+s+p-r
pour j=b, b-1, ..., a
(*) $P_{l-p}=P_{i-p}$
si $x_j \le t_i$ et i>a
alors $t_l=t_i$
i=i-1; l=l-1
aller à (*)
sinon pour k=1, 2, ..., p
$\beta=t_{l+k}-x_j$
si $\beta \le 0$
alors $P_{l-p+k-1}=P_{l-p+k}$
sinon $\beta=\beta/(t_{l-k}-t_{i-p+k})$
$P_{l-p+k-1}= \beta.P_{l-p+k-1}+(1-\beta).P_{l-p+k}$
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.

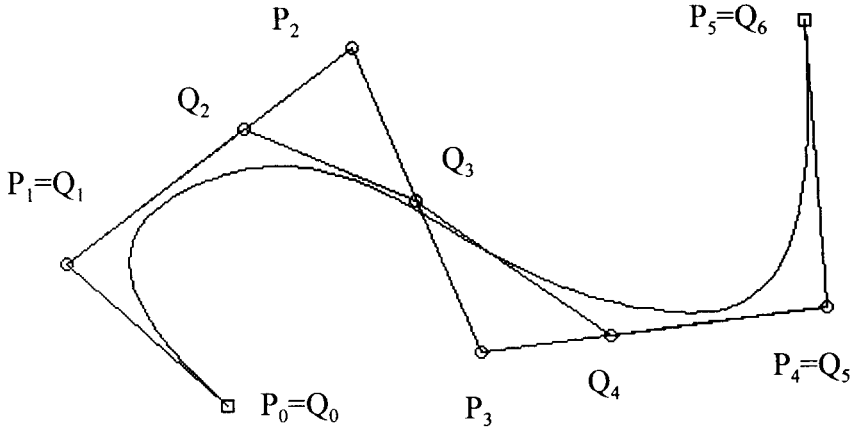

Figure II.6. Insertion du nœud 0.5 pour une courbe B-spline non rationnelle avec T={0,0,0,0,0.2,0.8,1,1,1,1}

II.2.1.2. Insertion de nœuds pour les courbes rationnelles

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

$$C(t)=\frac{\sum_{i=0}^{n} N_{i,p}(t).w_i.P_i}{\sum_{j=0}^{n} N_{j,p}(t).w_j}$$

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) $T=\{t_0,...,t_m\}$.

Soit $x_r \leq ... \leq x_s$ une séquence de nœuds à insérer dans l'intervalle $[t_a, t_b]$.
Dans le cas rationnel l'algorithme 1 précédent devient [*Böehm & Prautzch '85*] l'algorithme 2

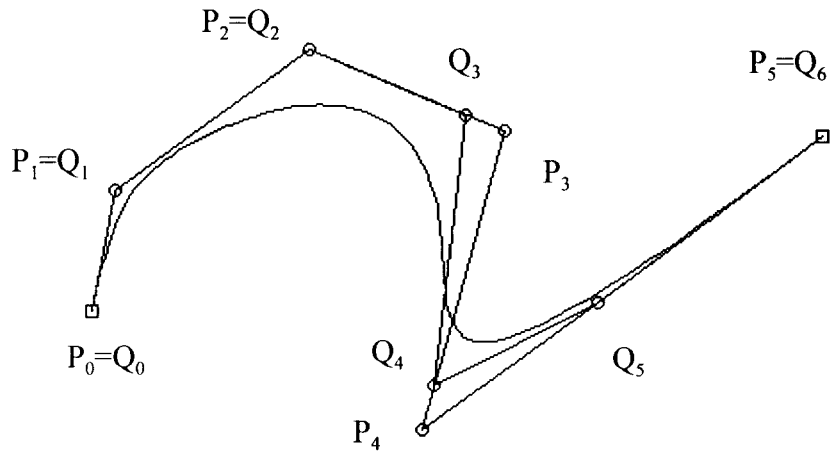

Figure II.7. Insertion du nœud 0.8 pour une courbe B-spline rationnelle de degré 3 avec T={0,0,0,0,0.2,0.6,1,1,1,1} et W={1,1.3,0.7,0.7,1.3,1}; W'={1,1.3,0.7,0.7,1.15,1.15,1}

Algorithme 2

i=b+p-1
l=b+s+p-r
pour j=b, b-1, ..., 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, ..., 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 $\beta=\beta/(t_{l-k}-t_{i-p+k})$
$w_{l-p+k-1}=\beta.w_{l-p+k-1}+(1-\beta).w_{l-p+k}$
$P_{l-p+k-1}=(\beta.P_{l-p+k-1}+(1-\beta).P_{l-p+k})/w_{l-p+k}$
Fin pour
$t_l=x_j$
l=l-1
Fin pour
Fin

II.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 nœud est une transformation par approximation.

II.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 (nœud surabondant).

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

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 :

$$P_3^0 = \alpha_3 . P_3^1 + (1-\alpha_3).P_2^1, \text{ avec } \alpha_3 = (t-t_3)/(t_7-t_3) \qquad (25)$$

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:

$$P_2^1 = \alpha_2 . P_2^2 + (1-\alpha_2).P_1^2, \text{ et } P_3^1 = \alpha_3 . P_3^2 + (1-\alpha_3).P_2^2$$

$$\alpha_i = \frac{t-t_i}{t_{i+p+2}-t_i}, \quad i=1,2. \qquad (26)$$

P_2^2 peut être tiré à partir des équations (26)

$$P_2^2 = \frac{P_2^1 - (1-\alpha_2).P_1^2}{\alpha_2}; \quad P_2^2 = \frac{P_3^1 - \alpha_3 . P_3^2}{1-\alpha_3} \qquad (27)$$

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

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

$$\alpha_i = \frac{t-t_i}{t_{i+p+3}-t_i}, \quad i=1,2,3. \qquad (28)$$

de la première et troisième équations (28) nous avons

$$P_1^3 = \frac{P_1^2 - (1-\alpha_1).P_0^3}{\alpha_1}; \quad P_2^3 = \frac{P_3^2 - \alpha_3 . P_3^3}{1-\alpha_3} \qquad (29)$$

en les remplaçant dans la deuxième équation de (28). Les formules générales sont

$$P_i^1 = \frac{P_i^0 - (1-\alpha_i).P_{i-1}^1}{\alpha_i} \qquad r-p \le i \le (2.r-p-s-1)/2$$

$$P_j^1 = \frac{P_j^0 - \alpha_j . P_{j+1}^1}{1-\alpha_j} \qquad (2.r-p-s+2)/2 \le j \le r-s \qquad (30)$$

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

où $t=t_r$ est le nœud de multiplicité s à supprimer et $1 \leq s \leq p$, 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 décrémentés et les numéros exposant des pôles sont incrémentés.
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 > l-1) faire
    $\alpha_i = (t - t_i)/(t_{i+p+1} - t_i)$
    $\alpha_j = (t - t_{j-l+1})/(t_{j+p+1} - t_{j-l+1})$
    $P_i^l = (P_i^{l-1} - (1-\alpha_i).P_{i-1}^l)/\alpha_i$
    $P_j^l = (P_j^{l-1} - \alpha_j.P_{j+1}^l)/(1-\alpha_j)$
    i=i+1
    j=j-1
  Fin faire
Fin pour
Fin

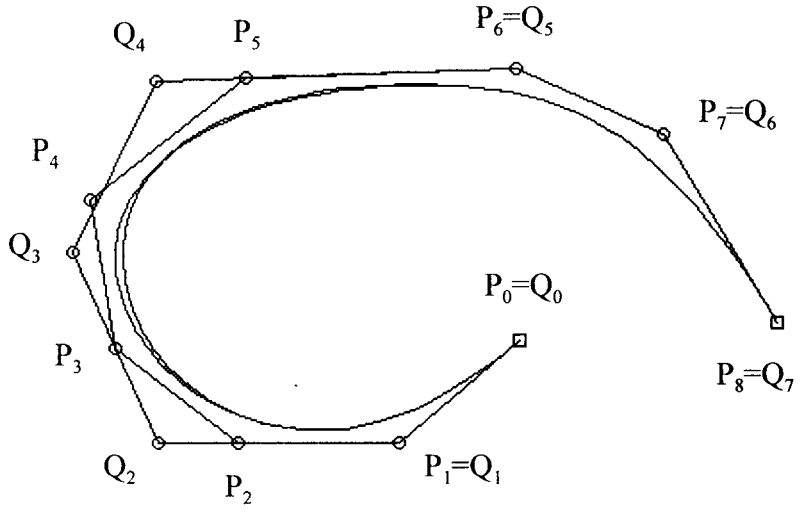

Figure II.8. Suppression du nœud 0.4 pour une courbe B-spline non rationnelle de degré 4 avec T={0,0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1,1}

II.2.2.2. Suppression de nœuds pour les courbes rationnelles

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

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

(2.r-p-s+2)/2 ≤ j ≤ r-s:

$$w_j^1 = \frac{w_j^0 - \alpha_j.w_{j+1}^1}{1-\alpha_j} \text{ et } P_j^1 = \frac{P_j^0 - \alpha_j.P_{j+1}^1}{(1-\alpha_j).w_j^1} \qquad (31)$$

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

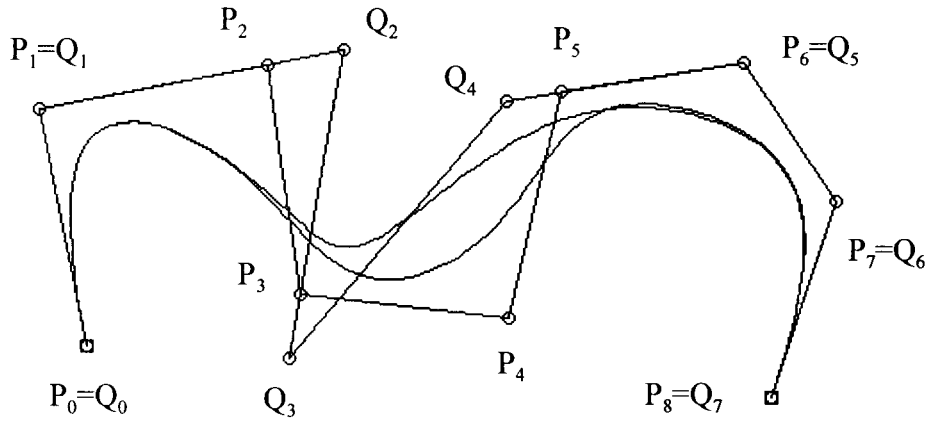

Figure II.9. Suppression du nœud 0.5 pour une courbe B-spline rationnelle de degré 4 avec T={0,0,0,0,0,0.2,0.5,0.6,0.8,1,1,1,1,1}, et W={1,1.2,0.8,1.3,0.7,1.3,0.8,1.2,1}; W'={1,1.2,0.72,1.648,1.6,0.8,1.2,1}

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

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

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

$$w_i^l = (w_i^{l-1} - (1-\alpha_i).w_{i-1}^l)/\alpha_i$$

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

$$w_j^l = (w_j^{l-1} - \alpha_j.w_{j+1}^l)/(1-\alpha_j)$$

$$P_j^l = \left(P_j^{l-1} - \alpha_j . P_{j+1}^l\right) \Big/ \left(\left(1-\alpha_j\right).w_j^l\right)$$

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

II.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.

II.2.3.1. Subdivision des courbes non rationnelles

Soit la courbe à base des fonctions B-spline $N'_{i,k}(t)$ de degré (k-1) définie par:

la fonction $C(t) = \sum_{i=0}^{n} P_i . N'_{i,k}(t)$ associée du vecteur nœuds $X=[X_0, ..., X_{n+k-1}]$

On désire additionner des nœuds supplémentaires $Y=[Y_1, ..., Y_1]$.
En posant m=n+1 la nouvelle répartition sera $Z=[Z_0, ..., Z_{m+k-1}]=X \cup Y$, répartie en ordre non décroissant.
Si $N_{i,k}(t)$ sont les fonctions de base B-spline sur cette nouvelle répartition, la courbe C(t) peut être écrite sous la forme (32) avec les inconnues Q_i :

$$C(t) = \sum_{i=0}^{n} Q_i . N_{i,k}(t) \text{ associée du vecteur nœuds } Z=[Z_0, ..., Z_{m+k-1}] \qquad (32)$$

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 (33)

$$Q_j = \sum_{i=1}^{n} \alpha_{i,k}(j).P_i \qquad (33)$$

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 $N'_{i,k}$ et les $N_{i,k}$ est donnée par :

Pour tout t :

$$N'_{i,k}(t) = \sum_{j=0}^{m} \alpha_{i,k}(j).N_{i,k}(t) \tag{34}$$

avec: $\alpha_{i,k}(j)=(X_{i+k}- X_i).[X_i, ..., X_{i+k}].\phi_{j,k}(t)$ (35)

et $$\phi_{j,k}(t) = (t - a_j)_+^0 . \prod_{r=1}^{k-1}(t - Z_{j+r}) \tag{36}$$

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

a_j est un nœud de $[Z_j, Z_{j+k}[$, et $[X_i, ..., X_{i+k}]$. $\phi_{j,k}(t)$ 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 $\alpha_{i,k}(j)$ vient de l'équation (35)

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

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

$$\alpha_{i,k}(j) = (Z_{j+k-1} - X_i).\beta_{i,k-1}(j) + (X_{i+k} - Z_{j+k-1}).\beta_{i+1,k-1}(j) \tag{37}$$

avec $$\beta_{i,k}(j) = \begin{cases} \alpha_{i,k}(j)/(X_{i+k} - X_i) & \text{si } X_{i+k} > X_i \\ 0 & \text{ailleurs} \end{cases} \tag{38}$$

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

Algorithme 5

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

$X_{s+2-k}, ..., X_{s+k-1}$ et $Z_{j+1}, ..., Z_{j+k-1}$

avec $X_{s+2-k} \le ... \le X_s < X_{s+1} \le ... \le X_{s+k-1}$

$X_s \le Z_j \le X_{s+1}$

$\alpha(s,1)=1$

$s_2=s$

Pour r=1,2, ..., k-1

$\beta_1=0$

$Z_j=Z_{j+r}$

Pour i=s_2, s_2+1, ..., s

$d_1=Z_j-X_i$

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

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

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

$\beta_1=d_1.\beta$

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 $Q_j = \sum_{i=s-k+1}^{s} \alpha_{i,k}(t).P_i$ et k, j, s, X_i ainsi que Z_i définis suivant l'algorithme 5 [*Bensalah '90*], [*Sablonnière '87-1*]

Algorithme 6

s_2=s-k+1
Pour i=s_2, s_2+1, ..., s
$P_{i,1}=P_i$
Fin pour
Pour r=1, 2, ..., k-1
$s_2=s_2-1$
k_r=k-r
$Z_j=Z_{j+kr}$
Pour i=s, s-1, ..., s_2
$d_1=Z_j-X_i$
$d_2=X_{i+kr}-Z_j$
$P_{i,r+1}=(d_1.P_{i,r}+d_2.P_{i+1,r})/(d_1+d_2)$
Fin pour
Fin pour
$Q_j=P_{s,k}$
Fin

$(d_1+d_2)>0$ tant que $X_s<X_{s+1}$ [*Bensalah '90*]

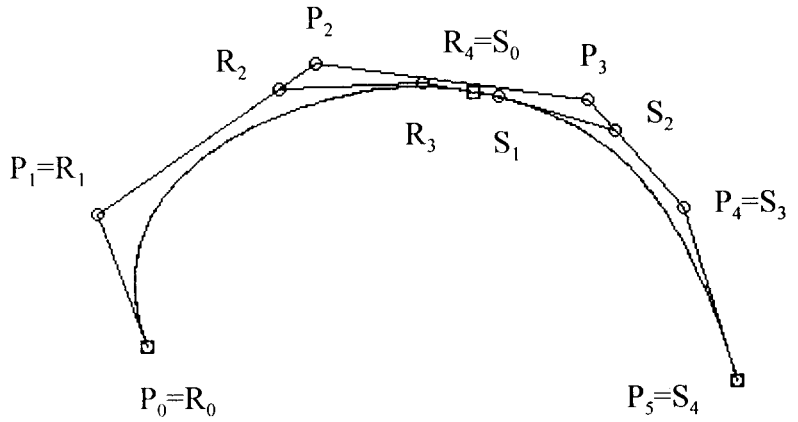

Figure II.10. Subdivision d'une courbe B-spline de degré 3 au point t=0.5 avec T={0,0,0,0,0.3,0.6,1,1,1,1} et T1={0,0,0,0,0.6,1,1,1,1}, T2={0,0,0,0,0.2,1,1,1,1}

II.2.3.2. Subdivision des courbes rationnelles

Soit la courbe rationnelle B-spline C(t) à base $N'_{i,k}(t)$:

$$C(t) = \frac{\sum_{i=0}^{n} w_i.P_i.N'_{i,k}(t)}{\sum_{i=0}^{n} w_i.N'_{i,k}(t)} \quad \text{dans } X=[X_0, ..., X_{n+k-1}] \qquad (39)$$

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

$$C(t) = \frac{\sum_{i=0}^{n} \sigma_i . P_i . N_{i,k}(t)}{\sum_{i=0}^{n} \sigma_i . N_{i,k}(t)} \quad \text{dans } Z=[Z_0, ..., Z_{m+k-1}] \qquad (40)$$

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 $Q_j = \sum_{i=s-k+1}^{s} (\alpha_{i,k}(t).P_i)/\sigma_i$, $\sigma_j = \sum_{i=s-k+1}^{s} \alpha_{i,k}(t).w_i$ et k, j, s, X_i ainsi que Z_i définis suivant l'algorithme 5 [*Bensalah '90*].

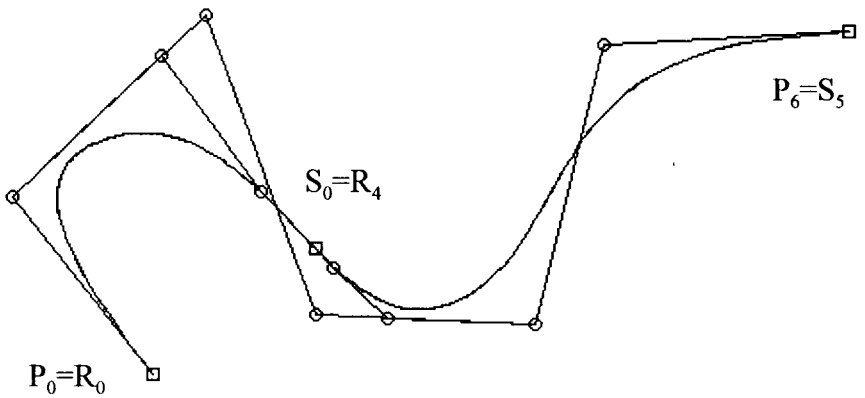

Figure II.11. Courbe B-spline rationnelle de degré 3 subdivisée en deux courbes à t=0.5, avec T={0,0,0,0,0.2,0.6,0.8,1,1,1,1}, T1={0,0,0,0,0.4,1,1,1,1} et T2={0,0,0,0,0.2,0.6,1,1,1,1}; et w={1,1.2,0.8,1,0.8,1.2,1}, w1={1,1.2,0.87,0.92,0.92} et w2={0.92,0.93,0.93,0.8,1.2}

Algorithme 7

$s_2=s-k+1$
Pour $i=s_2, s_2+1, ..., s$
  $P_{i,1}=P_i$
  $w_{i,1}=w_i$
Fin pour
Pour $r=1, 2, ..., k-1$
  $s_2=s_2-1$
  $k_r=k-r$
  $Z_j=Z_{j+kr}$
  Pour $i=s, s-1, ..., s_2$
    $d_1=Z_j-X_i$
    $d_2=X_{i+kr} + -Z_j$
    $P_{i,r+1}=(d_1.P_{i,r}+d_2.P_{i+1,r})/(d_1+d_2)$

$w_{i,r+1}=(d_1.w_{i,r}+d_2.w_{i+1,r})/(d_1+d_2)$

Fin pour

Fin pour

$Q_j=P_{s,k}$

$\sigma_j=w_{s,k}$

Fin

$(d_1+d_2)>0$ tant que $X_s<X_{s+1}$ [*Bensalah '90*]

II.3. Changement de base

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

II.3.1. B-spline- Bézier

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

II.3.1.1. Le polygone local de Bézier d'une courbe B-spline:

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

$$C(t) = \sum_{i=l-k}^{l} P_i . N_{i,k+1}(t) \qquad (41)$$

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 $P^0_{i,0} = P_i$ pour $l-k \le i \le l$ et

Pour $0 \le r \le k$, $1 \le j \le k-r$, et $l+j+r-k \le i \le l$

$$\alpha = \frac{t_l - t_i}{t_{i+k+l-j-r} - t_i} \qquad (42.a)$$

$$P^r_{i,j} = \alpha . P^r_{i,j-1} + (1-\alpha) . P^r_{i-1,j-1}$$

Pour $0 \le j \le k$, $1 \le r \le k-1$, et $l+j+r-k \le i \le l$

$$\beta = \frac{t_{l+1} - t_i}{t_{i+k+l-j-r} - t_i} \qquad (42.b)$$

$$P^r_{i,j} = \beta . P^{r-1}_{i,j} + (1-\beta) . P^{r-1}_{i-1,j}$$

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

$$Q_{l,j} = P^j_{l,k-j} \text{ pour } j=0, 1, ..., k \qquad (43)$$

alors les k+1 pôles $Q_{l,j}$ (j=0, 1, ..., k) de la courbe de Bézier locale (44) sont identique à la l-ième portion de la courbe B-spline définie par son support local (41):

$$Q_l(t) = \sum_{i=0}^{k} Q_{l,i} . B_{i,k}(t) \qquad (44)$$

Les deux algorithmes (42.a) et (42.b) doivent fonctionner simultanément. tous les polygones locaux de Bézier de toute la courbe B-spline peuvent donc être calculés.

II.3.1.1. Les polygones de Bézier d'une courbe B-spline:

La courbe B-spline définie par ses pôles P_i (i=0, ..., n), son ordre k, et son vecteur nœuds $T=\{t_0, ..., t_m\}$ 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: [*Bensalah '90*], [*Eck & Lasser '92*]

Pour $t_l<t_{l+1}=t_{k-1}$, ...,$t_l<t_{l+1}=t_m$
Si t_{l+1} est de multiplicité s<k-1
Calculer $A_{l,p}(t_{l+1})$ 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, ...,p remplacer
les t_{l+i} par t_{l+1}
les P_{l-p+i} par $A_{l-p+i,i}$
les P_{l+i} par $A_{l,p-i}$

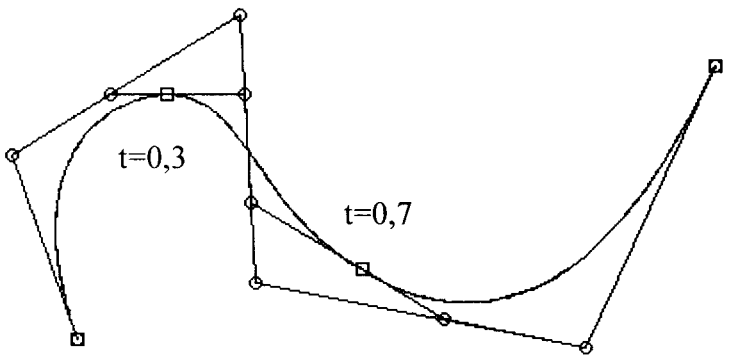

Figure II.12. Courbe B-spline non rationnelle de degré 3 transformée en 3 courbes de Bézier non rationnelles de degré3 avec T={0,0,0,0,0.3,0.7,1,1,1,1}

II.3.2. Bézier- B-spline

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

II.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, ..., n), d'après la propriété (§ I.3.1) des fonctions de base B-spline, la courbe B-spline définie par le vecteur nœuds $T=\left\{\underbrace{0,\ldots,0}_{n+1},\underbrace{1,\ldots,1}_{n+1}\right\}$ et les mêmes pôles est la courbe de Bézier C(t).

II.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 $Q=\{Q_{i,0}, ..., Q_{i,m}, ..., Q_{j,0}, ..., Q_{j,m}\}$, le vecteur nœuds est calculé en assignant une multiplicité (m+1) au premier et au dernier nœuds, et en mettant des intervalles proportionnels aux valeurs limites du paramètre t de chaque courbe de Bézier par rapport à la somme des intervalles. Pour chaque polygone local, les P_i locaux sont calculés par (45.a) et (45.b):[*Bensalah '90*], [*Joe, Wang & Cheng '94*]

en posant $P^0_{i,0} = Q_{l,m-l+i}$ pour l-m ≤ i ≤ l et

Pour 0 ≤ k ≤ m, l ≤ j ≤ m-k, et l+j+k-m ≤ i ≤ l

$$\alpha = \frac{t_{l+1-j} - t_l}{t_{l+1} - t_l} \qquad (45.a)$$

$$P^k_{i,j} = \alpha . P^k_{i,j-1} + (1-\alpha) . P^k_{i-1,j-1}$$

Pour 0 ≤ j ≤ m, l ≤ k ≤ m-j, et l+j+k-m ≤ i ≤ l

$$\beta = \frac{t_{l+k} - t_l}{t_{l+1} - t_l} \qquad (45.b)$$

$$P^k_{i,j} = \beta . P^{k-1}_{i,j} + (1-\beta) . P^{k-1}_{i-1,j}$$

Les pôles de la courbe B-spline seront:

$$P_{l-j} = P^{m-j}_{l,j} \text{ pour } j=0, 1, ..., m \qquad (46)$$

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

II.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*]

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

$$C(t) = \sum_{j=i-p}^{i} P_j . N_{j,p}(t)$$

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

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

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:

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

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

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

La courbe B-spline donc s'exprime:

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

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

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:

$$\sum_{j=i-p}^{i} \varphi_{k,p,i} . P_j - Q_{k,i} = 0 \quad \text{pour } k=0, 1, ..., p$$

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 $t \in [t_i, t_{i+1}]$ et

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

Ce qui peut s'écrire:

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

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 le calcul des coefficients $\varphi_{k,p,j}$ de la façon suivante: [*Meyer '87*]

$$\varphi_{k,p,j} = \frac{k}{p}\cdot\left\{\varphi_{k-1,p-1,j}\left(\frac{t_{i+1}-t_i}{t_{j+p}-t_j}\right)+\varphi_{k-1,p-1,j+1}\left(\frac{t_{i+p+1}-t_{i+1}}{t_{j+p+1}-t_{j+1}}\right)\right\}+$$
$$+\frac{p-k}{p}\cdot\left\{\varphi_{k,p-1,j}\left(\frac{t_i-t_j}{t_{j+p}-t_j}\right)+\varphi_{k,p-1,j+1}\left(\frac{t_{j+p+1}-t_i}{t_{j+p+1}-t_{j+1}}\right)\right\} \quad (47)$$

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}, ..., 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 WM 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 partie la plus importante dans cet algorithme est la définition des coefficients entre 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.

Réduction de degré des courbes de Bézier non rationnelles

III.1.	**Algorithmique de réduction de degré**	**39**
III.1.1.	Méthode de Forrest	39
III.1.1.1.	Algorithme de réduction	39
III.1.1.2.	Algorithme de sub-réduction	44
III.1.2.	Méthode de Bensalah	47
III.1.2.1.	Algorithme de réduction	47
III.1.2.2.	Algorithme de sub-réduction	49
III.1.3.	Méthode de Hoschek	53
III.1.3.1.	Paramétrisation intrinsèque d'une courbe de Bézier	53
III.1.3.2.	Réduction de degré	55
III.1.4.	Méthode de Watkins et Worsey	62
III.1.4.1.	Algorithme de réduction	62
III.1.4.2.	Algorithme de sub-réduction	64
III.1.4.3.	Conversion de base	68
III.1.5.	Méthode de Eck	70
III.1.5.1.	Réduction de degré et continuité géométrique	70
III.1.5.2.	Algorithme de sub-réduction	74
III.2.	**Analyse comparative des méthodes de Réduction**	**78**
III.2.1.	Etude critique	78
III.2.2.	Comparaison analytique	79
III.2.3.	Comparaison expérimentale	80
III.2.4.	Méthode proposée	90
III.2.4.1.	Méthode d'approximation	90
III.2.4.2.	Algorithme de réduction	91
III.2.4.3.	Algorithme de sub-réduction	95
III.2.4.4.	Comparaison avec les autres méthodes	97

III. Réduction de degré des courbes de Bézier non rationnelles

Nous constatons dans l'algorithme proposé dans le chapitre II que la réduction de degré produit une erreur d'approximation qu'on ne peut contrôler dépendant du degré et de la forme de la courbe. En pratique, et suivant l'application associée à cette réduction, l'approximation se pose de la manière suivante :

Réduire le degré d'une courbe de n à m<n en respectant une ou plusieurs tolérances géométriques 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 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 combinant 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 réduction actuelles, appelées aussi sub-réduction.

Nous présenterons dans ce chapitre d'autres méthodes de réduction de degré faisant des approximations raisonnables de façon à surmonter les inconvénients rencontrés dans le processus de transmission de données entre systèmes de C.F.A.O.

III.1. Algorithmique de réduction de degré

III.1.1. Méthode de Forrest

En inversant l'algorithme d'élévation de degré, Forrest a proposé un algorithme de réduction [*Forrest '72*]. En vue de son application à la réduction de degré Piegl et Tiller, l'ont développé explicitement et l'ont appliqué sur les courbes B-spline non rationnelles [*Piegl & Tiller '95*].
La méthode présentée ici est celle de Piegl et Tiller basée sur les fondements de Forrest.

III.1.1.1. Algorithme de réduction

Soit une courbe P(t) de degré (n+1) définie par ces n+2 pôles P_i à réduire en une courbe Q(t) de degré n définie par ses n+1 pôles Q_i.

La réduction de degré des courbes de Bézier non rationnelles proposée dans le paragraphe II.1.3.1. donne deux solutions généralement différentes R(t) (17.a) produisant une bonne approximation au voisinage de t=0 et S(t) (17.b) produisant une bonne approximation au voisinage de t=1.

L'approximation proposée par Forrest consiste à prendre pour la courbe réduite Q(t) les (n+1)/2 premiers pôles de la courbe R(t) et les (n+1)/2 derniers pôles de la courbe S(t).
Selon que n est pair ou impair, l'algorithme de Forrest s'écrit

Pour n pair: (48)

Pour i=0,...,(n-1)/2

$Q_i=R_i$

Fin pour

Pour i=(n+1)/2,...,n-1

$Q_i=S_j$

Fin pour

Fin

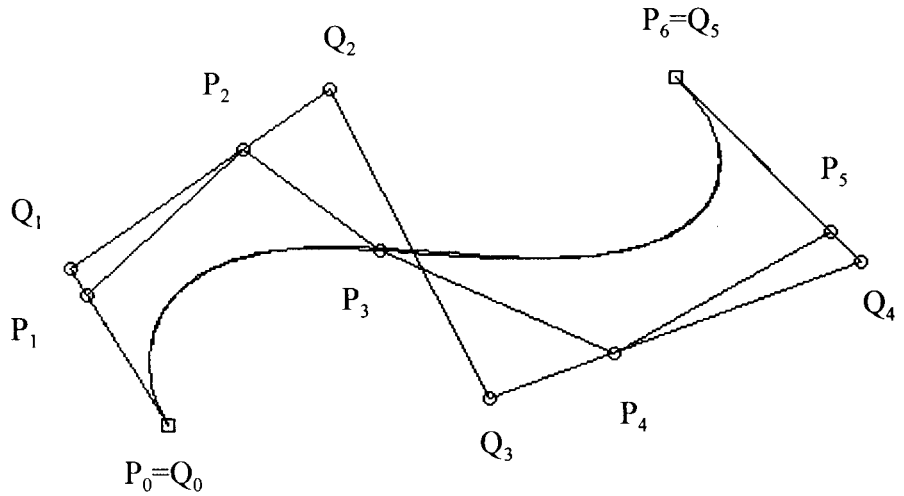

Figure III.1.a. Courbe de Bézier de degré 6 réduite au degré 5

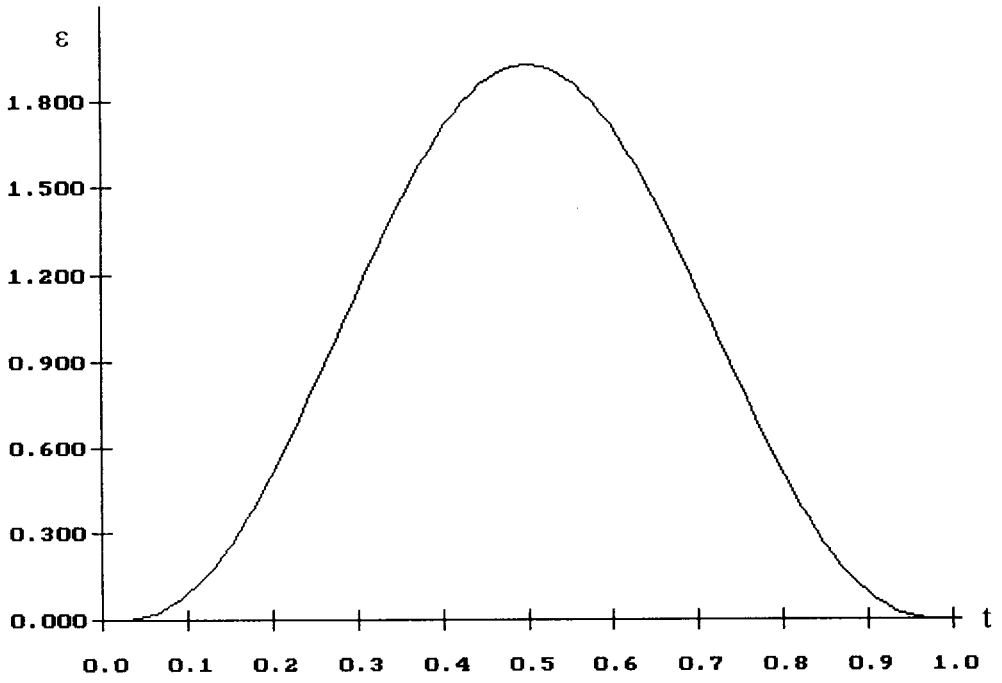

Figure III.1.b Erreur de position de la courbe précédente

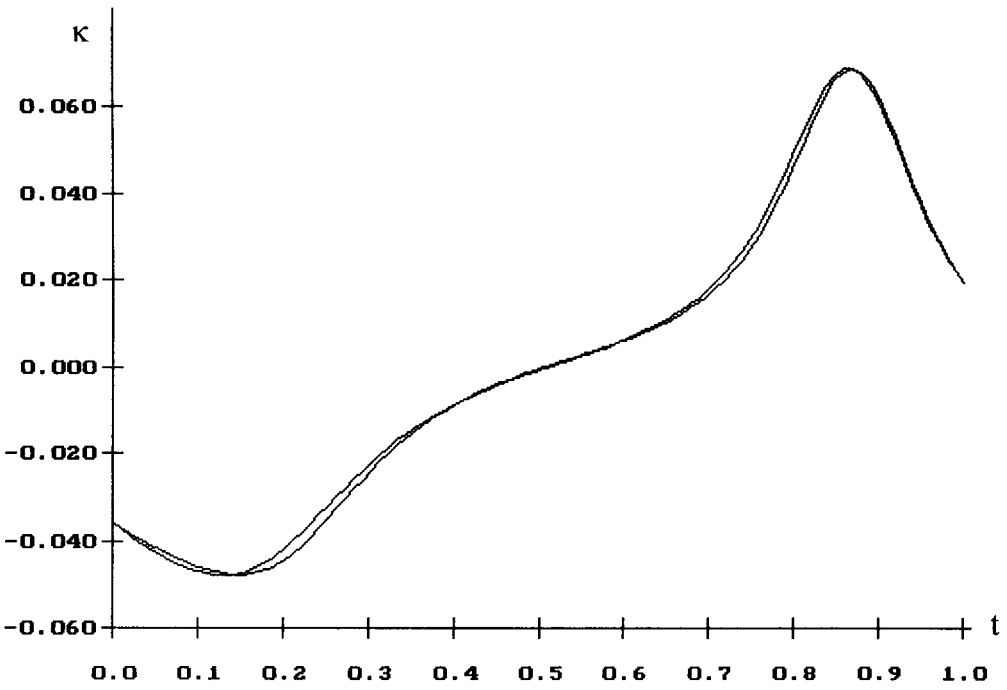

Figure III.1.c. Variation de la courbure de la courbe précédente (avant et après réduction)

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

$$\left|P(t)-Q(t)\right| = \left|B_{(n+1)/2,n}(t).\left[P_{(n+1)/2} - \frac{1}{2}.\left(Q_{(n-1)/2} - Q_{(n+1)/2}\right)\right]\right| \qquad (49)$$

Il est intéressant de noter que l'erreur maximale est obtenue à t=1/2 (fig III.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: (50)

Pour i=0,...,(n-3)/2
$Q_i=R_i$
Fin pour
Pour i=(n+1)/2,...,n-1
$Q_i=S_i$
Fin pour
$Q_{(n-1)/2}=(R_{(n-1)/2}+S_{(n-1)/2})/2$
Fin.

L'erreur d'approximation est calculée par [*Piegl & Tiller '95*]

$$\left|P(t)-Q(t)\right| = \frac{1}{2}.\left(1-\frac{n-1}{2.n}\right).\left|\left(B_{(n-1)/2,n}(t) - B_{(n+1)/2,n}(t)\right).\left(R_{(n-1)/2} - S_{(n-1)/2}\right)\right| \qquad (51)$$

Il est aussi intéressant de 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, que l'erreur possède deux pics (figure III.2.b) calculés par l'équation quadratique suivante:

$$t^2 - t + \frac{n}{4.(n+1)} = 0 \qquad (52)$$

En considérant la formule précédente de l'erreur, nous constatons que
- Cette formule a un contrôle paramétrique de l'erreur. Le maximum géométrique n'est pas nécessairement atteint à la même valeur paramétrique de 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 d'une réduction de degré.

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. 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, 1, ...,n) polygone de la courbe de degré n
k ordre de la réduction

Pour i=0, 1, ..., 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
$\beta_i=i/(n-i)$
$R_i = (1+\beta_i).W_i - \beta_i.R_{i-1}$
$W_i=R_i$
Fin pour
Pour i=n-1, n-2, ..., 1
$\gamma_i=(n-i)/i$
$S_{i-1} = (1+\gamma_i).W_i - \gamma_i.S_i$
$W_i=S_i$
Fin pour
n=n-1
Fin pour
Si n-k est pair alors
Pour r=0, 1, ..., (n-k)/2
$Q_i=R_i$;
Fin pour
Pour i=(n-k)/2,...,n
$Q_i=S_j$
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+5)/2, ..., n-k
$Q_i=S_i$
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 la courbe de degré (n-k)
Fin

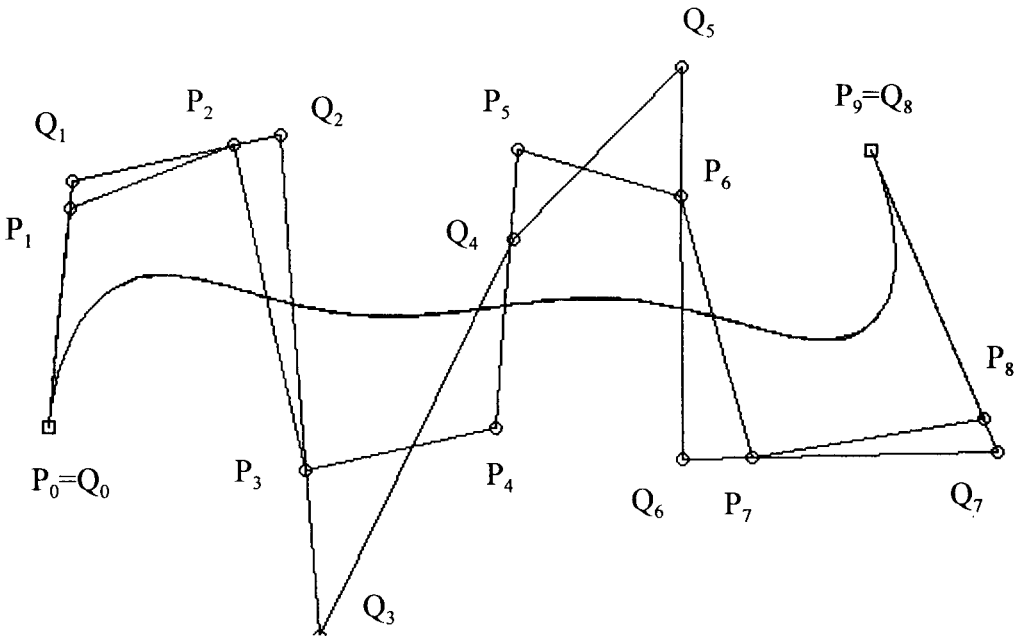

Figure III.2.a.. Courbe de Bézier de degré 9 réduite au degré 8

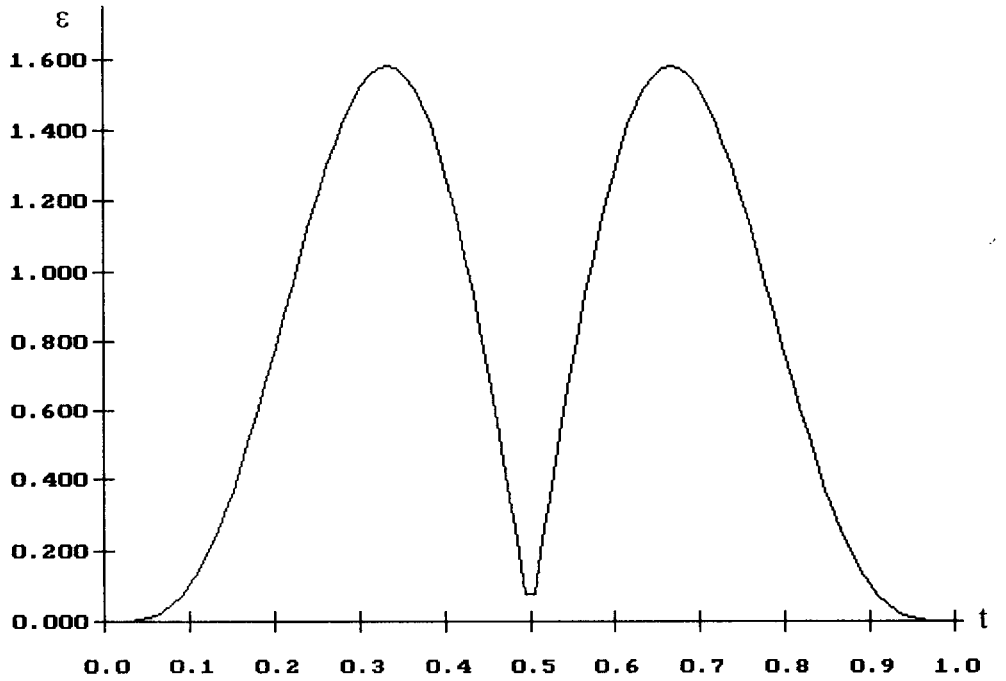

Figure III.2.b. Erreur de position de la courbe précédente

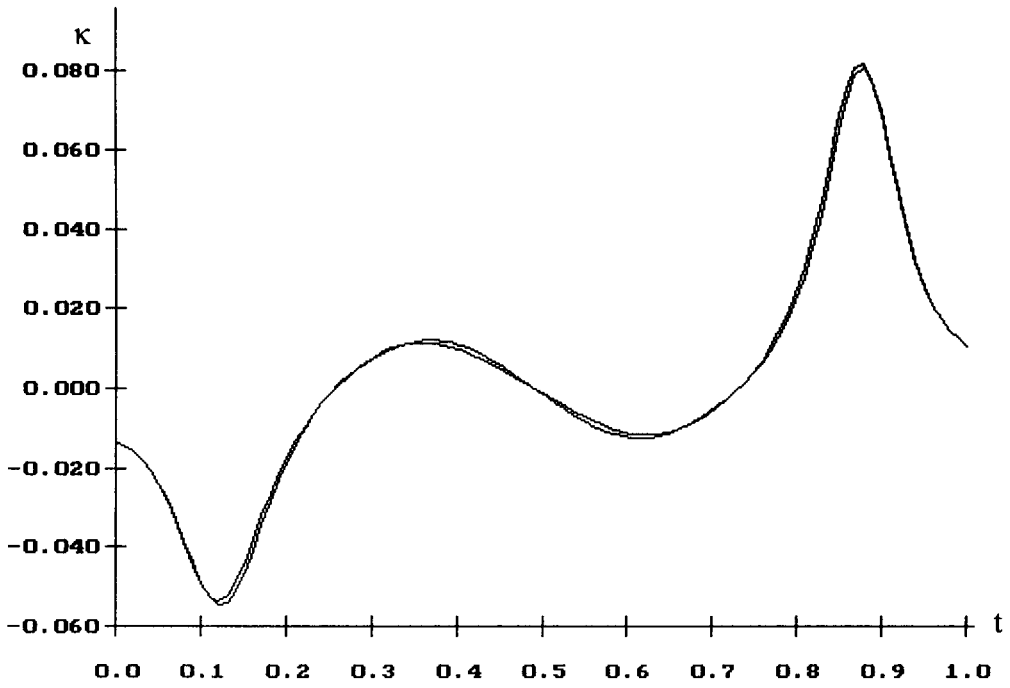

Figure III.2.c. Variation de la courbure de la courbe précédente (avant et après réduction)

III.1.1.2. Algorithme de sub-réduction

Nous constatons que l'algorithme FOR produit une 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 SUB-FOR permettant de respecter une tolérance ε_0.
L'algorithme de subdivision qui sera utilisé dans toutes les méthodes de sub-réduction est l'algorithme SUB suivant

Algorithme SUB

Données: C_i (i=0, 1, ..., n) polygone de la courbe de degré n
t_0 : abscisse curviligne du point de subdivision

Pour i=0, 1, ..., n
$C_i^0(t_0) = C_i$
Fin pour
Pour k=1, 2, ..., n
Pour i=0, 1, ..., n
$C_i^k(t_0) = (1-t_0).C_i^{k-1}(t_0) + t_0.C_{i+1}^{k-1}(t_0)$
Fin pour
Fin pour
Pour i=0, 1, ..., n
$P_i = C_0^i(t_0)$; $Q_i = C_i^{n-i}(t_0)$
Fin pour

Résultats: P_i et Q_i (i=0, 1, ..., n)
Fin

Algorithme SUB-FOR

Données : P_i^0 (i=0, 1, ..., n) polygone de la courbe de degré n.
k: Ordre de la réduction
ε_0: tolérance de position
N: nombre de pas.

n1=0; n2=0
Pour j=0, 1, ..., n1
FOR (P^j, k → 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 → P0, P1)
n1=n1+2; N=N div 2
Pour i=0, 1, ..., n
$P_i^{n1-1} = P0_i$; $P_i^{n1} = P1_i$
Fin pour
Sinon faire
Pour j=0, 1, ..., n-k
$Q_i^{n2} = Q_i$
Fin pour
n2=n2+1
Fin Si
Fin pour

Résultats : Q_i^j (i=0, 1, ..., n-k) (j=0, 2, ..., n2-1)
n2 : nombre de courbes de degré (n-k).
Fin.

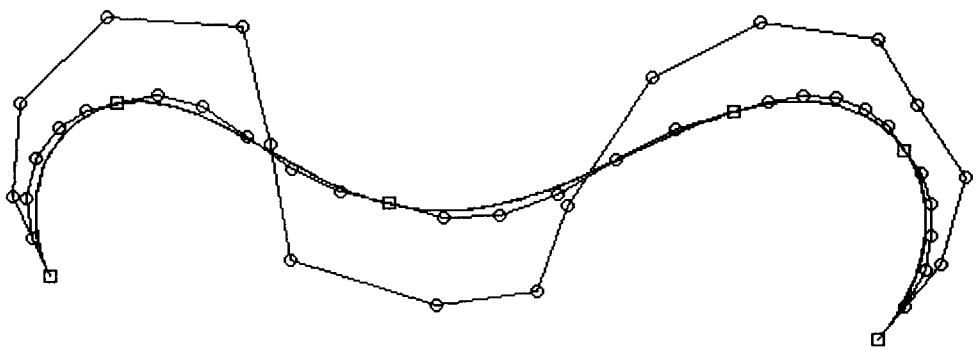

Figure III.3.a. Courbe de degré 16 réduite en 5 courbes de degré 6

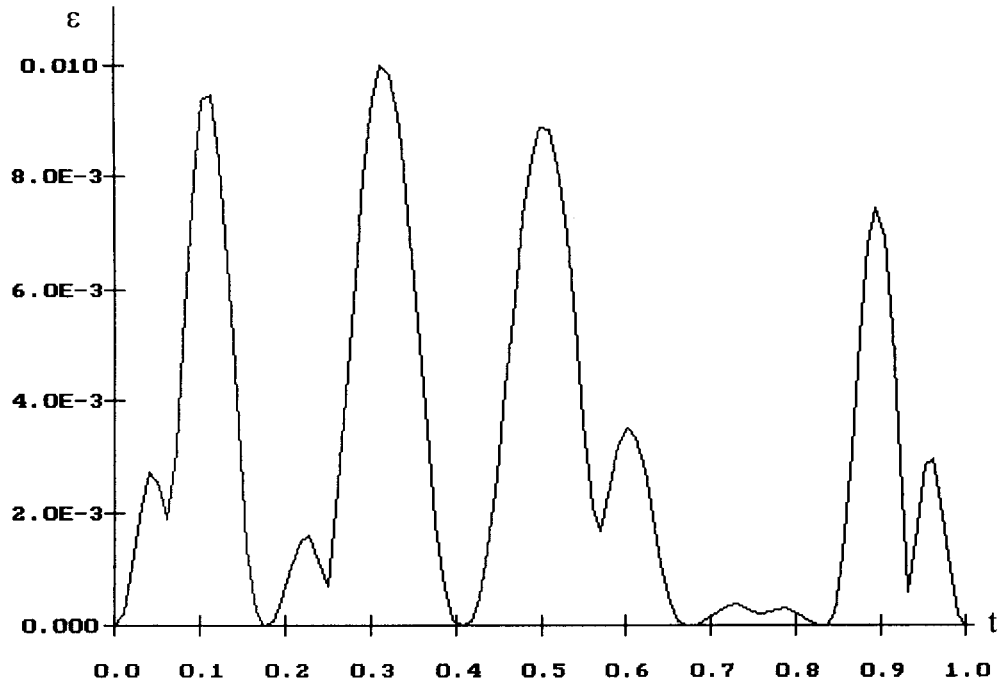

Figure III.3.b. Erreur de position le long de la courbe de degré 16

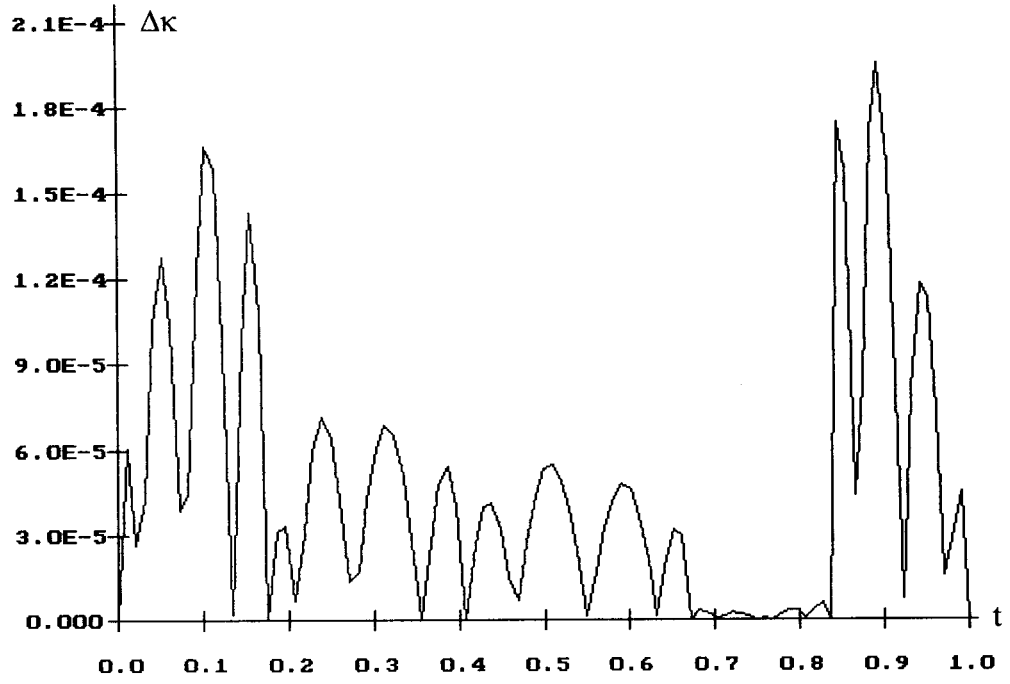

Figure III.3.c. Variation de l'erreur sur la courbure

La figure III.3.a représente une courbe de Bézier non rationnelle de degré 16 réduite en 5 courbes de degré 6 par l'algorithme SUB-FOR avec ε_0=0.01 et k=10. La figure III.3.b qui 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 aux points communs, tandis que la figure III.3.c exprime la continuité de courbure.

III.1.2. Méthode de Bensalah

III.1.2.1 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 III.4). La meilleure courbe est celle donnée par Q(t) (polygone moyen (18)) qui est une moyenne pondérée des deux (figure III.5). Ces trois solutions forment les algorithmes de base de la méthode de Bensalah que nous reprenons sous la forme suivante [*Farin '83*], [*Bensalah '90*].

Algorithme BEN1

Données: P_i (i=0, 1, ...,n) polygone de la courbe de degré n
k ordre de la réduction

Pour r=1, 2, ..., k
$R_0=P_0$
Pour i=1, 2, ..., n-1
$\beta_i=i/(n-i)$
$R_i = (1+\beta_i).P_i - \beta_i.R_{i-1}$
$P_i=R_i$
Fin pour
n=n-1
Fin pour

Résultats: R_i (i=0, 1, ..., n-k) polygone de la courbe de degré (n-k)
Fin

Algorithme BEN2

Données: P_i (i=0, 1, ...,n) polygone de la courbe de degré n
k ordre de la réduction

Pour r=1, 2, ..., k
$S_{n-1}=P_n$
Pour i=n-1, n-2, ..., 0
$\gamma_i=(n-i)/i$
$S_{i-1} = (1+\gamma_i).P_i - \gamma_i.S_i$
$P_i=S_i$
Fin pour
n=n-1
Fin pour

Résultats: S_i (i=0, 1, ..., n-k) polygone de la courbe de degré (n-k)
Fin

Algorithme BENm

Données: P_i (i=0, 1, ...,n) polygone de la courbe de degré n
k ordre de la réduction

BEN1 (P, k $\rightarrow$ R)
BEN2 (P, k $\rightarrow$ S)
Pour i=0, 1, ..., n-k

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

Fin pour

Résultats: Q_i (i=0, 1, ..., n-k) polygone de la courbe de degré (n-k)
Fin

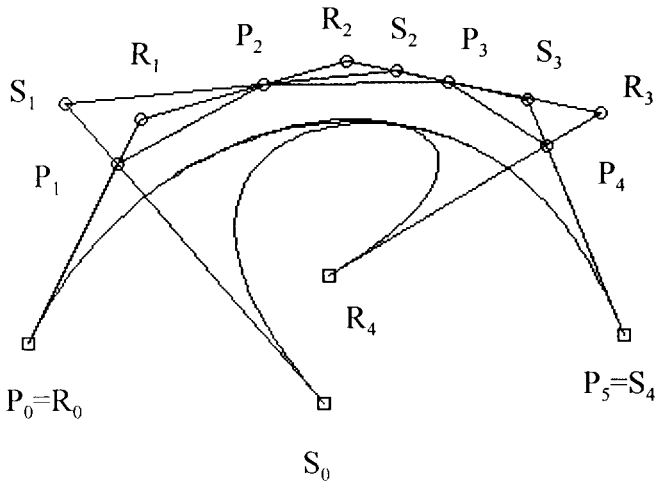

Figure III.4.courbe de degré 5 réduite au degré 4 par BEN1 et BEN2

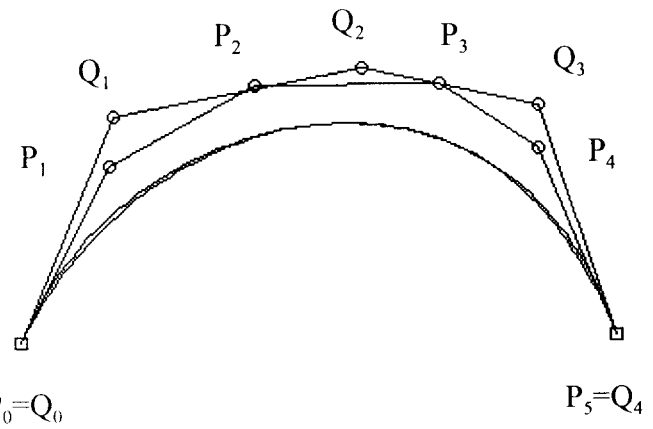

Figure III.5. courbe de degré 5 précédente réduite au degré 4 par BENm

III.1.2.2. Algorithme de sub-réduction

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 ou 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 SUB-BEN. Nous remarquons que pour chaque itération on retrouve 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 coté d'où la partie 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. [*Bensalah '90*]

Algorithme SUB-BEN

Données : P_i (i=0, 1, ..., n) polygone de la courbe de degré n.
k: Ordre de la réduction
ε_0: tolérance de position
N: nombre de pas.

(*) **BEN1** (P, k → Q)
s=0; $d = |P_n - Q_{n-k}|$
Si (d>ε_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 → Q0, Q1) ; **SUB** (P, t_0 → P0, P1)
Pour i=0, 1, ..., n-k
$Q_i^s = Q0_i$
Fin pour
Pour i=0, 1, ..., n
$P_i = P1_i$
Fin pour
N=N-(1/t)
BEN2 (P, k → Q)
s=0; $d = |P_0 - Q_0|$
Si (d>ε_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 → Q0, Q1)
SUB (P, t_0 → 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
Ajustement des tangentes des Q_i^s

Résultats : Q_i^j (i=0, 1, ..., n-k) (j=1, 2, ..., 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.

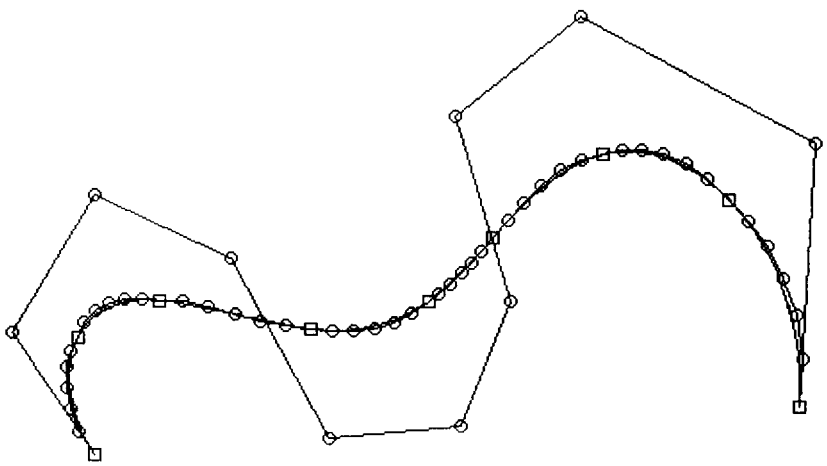

Figure III.6.a. Courbe de degré 10 réduite en 8 courbes de degré 6 par l'algorithme SUB-BEN

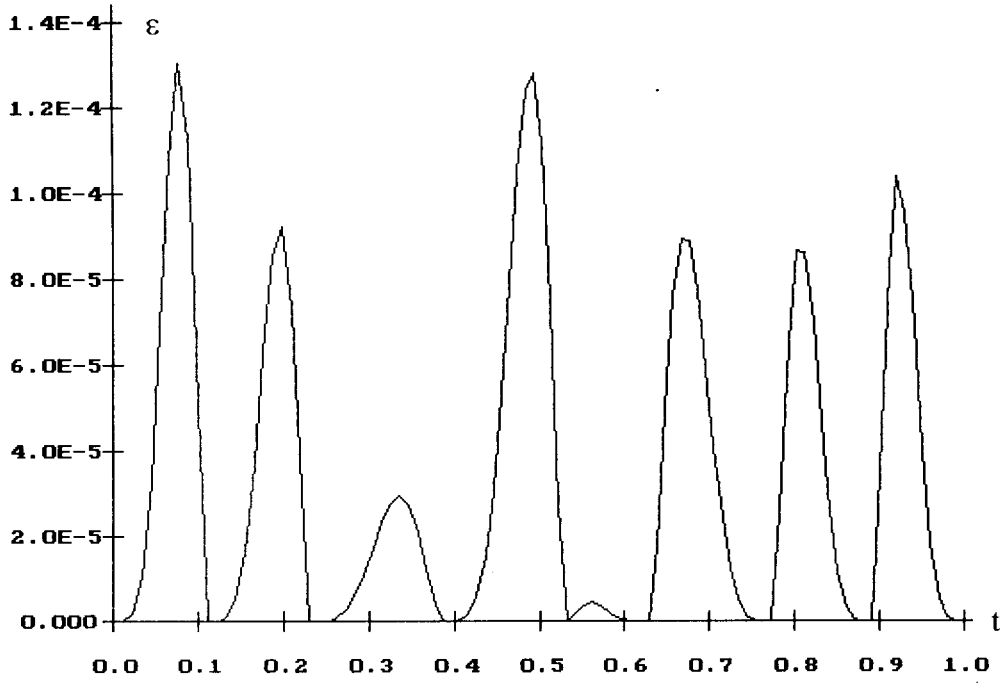

Figure III.6.b. Erreur de position pour la courbe précédente

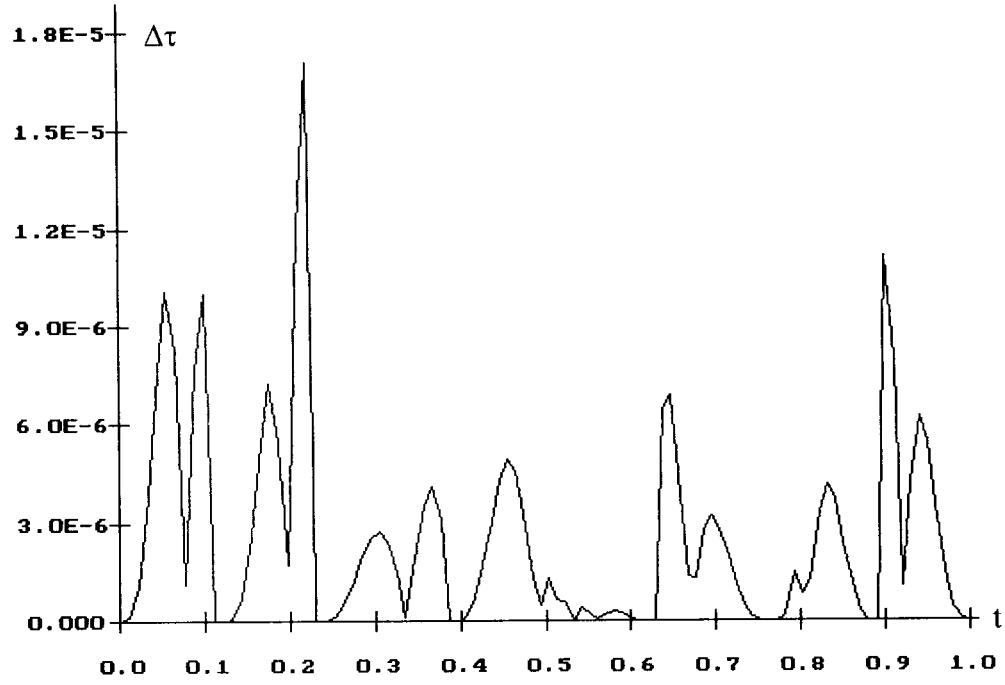

Figure III.6.c. Variation de l'erreur sur la tangente

La figure III.6.a représente une courbe de Bézier non rationnelle de degré 10 réduite en 8 courbes de degré 6 par l'algorithme SUB-BEN avec ε_0=0.01. Nous pouvons constater que la continuité de position sur la figure III.6.b, et la continuité des tangentes sur la figure III.6.c sont bien respectées grâce à l'ajustement des tangentes, mais la figure III.6.d montre qu'il y a une discontinuité de la courbure. La méthode ne respecte donc que la continuité de la tangente.

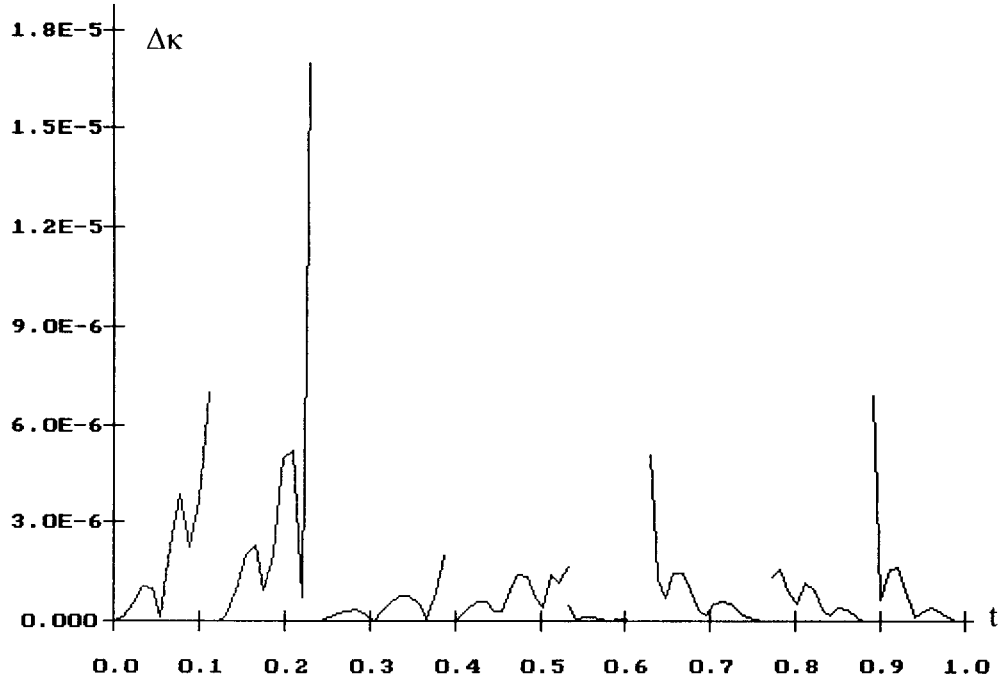

Figure III.6.d. Variation de l'erreur sur la courbure après ajustement

III.1.3. Méthode de Hoschek

La subréduction étudiée dans cette partie, a trouvé son application dans le cas des courbes planes pour lesquelles elle donne de bons résultats.
De cet algorithme découle deux applications: [*Hoschek & Wissel '88*], [*Hoschek '90*]

- 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 M en une courbe de degré N>M (élévation de degré) en respectant une tolérance 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 de points changera, si on change les valeurs du paramètre t des points durant le processus d'approximation.

III.1.3.1. Paramétrisation intrinsèque

Le but de la paramétrisation est de trouver une bonne répartition des paramètres t_i sur les points M_i permettant une approximation optimale.

Soit donné un ensemble np de points M_i, La paramétrisation permet d'affecter un coefficient t_i à chaque point. L'approximation de ces points par une courbe de Bézier non rationnelle donne

$$M_j \approx P(t_j) = \sum_{i=0}^{n} P_i . B_{i,n}(t_j), \text{ n degré choisi} \tag{53}$$

L'approximation est optimale si la distance entre deux points consécutifs de la courbe P(t) est minimisée, le problème se ramène donc à la minimisation suivante

$$\sum_{j=0}^{np-1} \min_{t_j \in [0,1]} \left| P(t_j) - M_j \right| \rightarrow \min \tag{54}$$

Ce problème de minimisation non linéaire peut être réduit à un processus linéaire en introduisant l'erreur vectorielle d_j. La minimisation devient alors

$$D = \sum_{j=0}^{np-1} d_j = \sum_{j=0}^{np-1} \left(M_j - P(t_j) \right) \rightarrow \min \quad [\textit{Hoschek '88}] \tag{55}$$

la méthode des moindres carrés permet d'écrire $\dfrac{\partial D}{\partial P_i} = 0$ (56)

Généralement, cette erreur vectorielle n'est pas orthogonale à l'approximation P(t). La résolution est linéaire, mais la distance entre les points M_j et leur approximation par la courbe $P(t_j)$ n'est pas minimisée. Cas de la distance décrite Figure III.7.a.
Pour une approximation optimale, il devient nécessaire de corriger la paramétrisation en faisant une approximation en plus sur cette distance. Il faut donc construire de nouveaux paramètres $\tilde{t}_j$ pour avoir une nouvelle erreur vectorielle $\tilde{D}_j$ qui converge à la normale à la courbe P(t). Pour cela, il faut remplacer la courbe à chaque point $P(t_j)$ par la tangente (approximation linéaire de P(t) au voisinage de $P(t_j)$), puis projeter

l'erreur vectorielle D_j sur la tangente, ce qui permet d'obtenir la correction Δc_j comme mesure de changement du paramètre t_j qui converge vers la perpendiculaire M_j à P(t) (Figure III.7.b).

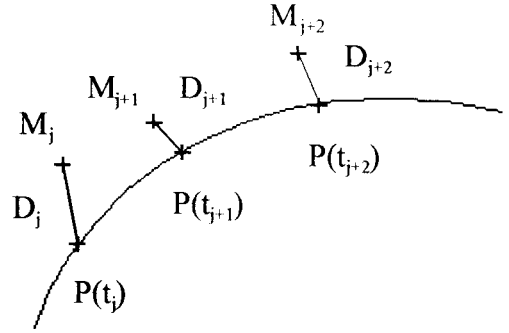

Figure III.7.a

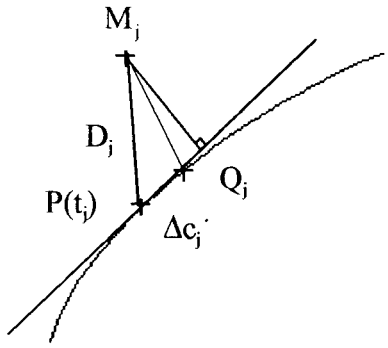

Figure III.7.b

Δc_j doit être mesurée en respectant la longueur de la courbe et celle de l'intervalle I=[a, b] ici a=0, b=1, et la longueur de la courbe μ est approximée par celle du polygone obtenu par les points M_j. Ce qui donne la correction du paramètre t_j suivante

$$\tilde{t}_j = t_j + \Delta c_j . \frac{b-a}{\mu}, \text{ avec } \Delta c_j = D_j . \frac{P'(t_j)}{|P'(t_j)|} \qquad (57)$$

la correction (57) est admissible si $|P(\tilde{t}_j) - M_j| \le |P(t_j) - M_j|$, sinon la correction aura dépassé la position du point Q_j (Figure III.7.b) de la perpendiculaire à P(t) passant par M_j, donc t_j sera corrigé seulement par $\Delta c_j/2$.
La méthode des moindres carrés (55) est de nouveau appliquée en utilisant les valeurs obtenues en (57). Cette étape est répétée jusqu'à ce que l'erreur vectorielle $\tilde{D}_j$ soit orthogonale (approximée en respectant une tolérance ε_0) à la courbe P(t).
Cette paramétrisation sera la paramétrisation optimale du problème considéré.
L'algorithme de la paramétrisation intrinsèque s'écrit [*Hoschek '88*]

Algorithme 8

Données: M_j j=0, 1, ..., np
n : degré de l'approximation

Pour j=0, 1, ...,np
t_j=j/np
Fin pour
L=4; ε_0=0.01; r=1
Calculer μ (longueur du polygone $\{M_j\}$)
(*) Déterminer P(t) par la méthode des moindres carrés (en respectant (56))
Pour j=0, 1, ...,np
Calculer Δc_j
(**) $\tilde{t}_j = t_j + \Delta c_j . \frac{b-a}{\mu}$

$|D_j| = |P(t_j) - M_j|$

$|\tilde{D}_j| = |P(\tilde{t}_j) - M_j|$

Si $D_j > \tilde{D}_j$ alors aller à (***)

Sinon $\Delta c_j = \Delta c_j/2$ aller à (**)

(***) $\alpha_j = \dfrac{D_j . P'(t_j)}{|D_j| . |P'(t_j)|}$

Fin pour

$\alpha = 0$

Pour j=0, 1, ...,np

si $\alpha_j > \alpha$ alors $\alpha = \alpha_j$

Fin pour

r=r+1

Si $\left|\frac{\pi}{2} - \alpha\right| \geq \varepsilon_0$ alors

Si r<L alors aller à (*)

FinSi

FinSi

Résultats: Les pôles P_i (i=0, 1, ..., n)

Fin.

Cette méthode a été utilisée par Hoschek pour l'approximation de la conversion de courbes, et pour l'approximation des offsets et donne des résultats pratiques très satisfaisants.

III.1.3.2. 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(t) et Q(t) ont une continuité géométrique d'ordre k si les conditions suivantes sont vérifiées pour le point commun entre P(t) et Q(t). [*Hoschek & Wissel '88*], [Hoschek '86]

$$\begin{aligned} k=1 &\rightarrow \quad P' = \lambda_1 . Q' \\ k=2 &\rightarrow \quad P'' = \lambda_1^2 . Q'' + \lambda_2 . Q' \\ k=3 &\rightarrow \quad P''' = \lambda_1^3 . Q''' + 3.\lambda_1 . \lambda_2 . Q'' + \lambda_3 . Q' \\ k=4 &\rightarrow \quad P^{IV} = \lambda_1^4 . Q^{IV} + 6.\lambda_1^2 . \lambda_2 . Q''' + (3.\lambda_2^2 + 4\lambda_1 . \lambda_3) . Q'' + \lambda_4 . Q' \end{aligned} \qquad (58)$$

Pour un choix arbitraire des paramètres λ_i ces conditions peuvent être développées en utilisant une droite ou une conique ou aussi une cubique. Le modèle utilisé pour le développement de l'algorithme est le modèle de Bézier.
Les conditions de continuité dans le modèle de Bézier deviennent

$$k=1 \rightarrow \quad Q_0 = P_0$$
$$Q_1 = P_0 + \lambda_1.(P_1 - P_0) \qquad (59.a)$$
$$Q_m = P_n$$
$$Q_{m-1} = P_n + \mu_1.(P_n - P_{n-1})$$
$$k=2 \rightarrow \quad Q_2 = P_1 + \lambda_1^2.\omega_1.(P_2 - P_1) + \lambda_2.(P_1 - P_0)$$
$$Q_{m-2} = P_{n-1} + \mu_1^2.\omega_1.(P_{n-2} - P_{n-1}) + \mu_2.(P_n - P_{n-1}) \qquad (59.b)$$

avec $\omega_1 = \dfrac{m.(n-1)}{n.(m-1)}$

Pour k=3 et k=4 (*Cf. Annexe C*)

III.1.3.2.1 Algorithme de réduction

Nous développerons dans cette partie les deux cas de continuité k=1 et k=2.

Continuité d'ordre 1

k=1 et m=3. Les équations (59.a) sont les quatre conditions pour les pôles de la courbe de Bézier à déterminer Q_0, Q_1, Q_2 et Q_3, ces pôles sont obtenus par:
- les paramètres λ_1 et μ_1.
- la paramétrisation de la courbe de Bézier Q(t).

Pour évaluer le couple optimal (λ_1, μ_1), on choisit sur la courbe donnée P, (n+1) points M_i avec un paramètre curviligne équidistant $t_i = \dfrac{i}{n}$; $(i = 0,...,n)$ et on considère que Q(t) est approximée par ces points: $M_i = \sum_{j=0}^{3} Q_j.B_{j,3}(t) + \delta_i$ (60)

$$\delta = \sum_{i=0}^{n} |\delta_i|^2 \qquad (61)$$

Le minimum de δ est obtenu par les conditions suivantes

$$\frac{\partial \delta}{\partial \lambda_1} = 0 \quad \text{et} \quad \frac{\partial \delta}{\partial \mu_1} = 0 \qquad (62)$$

qui conduit à un système linéaire en λ_1 et μ_1, mais le résultat dépend de la paramétrisation des points M_i. Il faut donc résoudre le système linéaire obtenu à partir de (62) et insérer la solution avec (59.a) dans (60), on obtient alors la première approximation de la courbe Q(t). Si maintenant on insère les valeurs des paramètres donnés t_i dans Q(t), on obtient l'erreur vectorielle: $\delta_i = Q(t_i) - M_i$
Les vecteurs sont généralement non orthogonaux à Q(t) dans $Q(t_i)$, mais seule l'erreur vectorielle conduit à une approximation optimale. Donc, on doit changer la

paramétrisation des points M_i pour avoir une meilleure approximation en utilisant la paramétrisation intrinsèque précédente.
Sachant que la projection Δc_i est normée par μ (approximation de la longueur de la courbe Q(t) par la longueur de son polygone de contrôle).

Dans nos exemples nous avons à rechercher ces positions orthogonales seulement après 3 itérations, la courbe obtenue est l'approximation optimale de l'ensemble des points et leurs polynômes choisis (Bernstein) comme fonctions de base. On peut évaluer l'erreur de position maximale à :

$$\delta = \mathrm{Max}\left|\delta_j\right| = \mathrm{Max}\left|P(t_i) - Q(t_i)\right|$$

de la courbe P donnée et la courbe Q(t) obtenue pour quelques valeurs du paramètre t_j, les t_j sont choisis suivant:

(a) Les valeurs du paramètre corrigé de t_j des points $M_i \in P(t_i)$.

(b) En incrémentant les valeurs t_l entre t_i et t_{i+1} en respectant (a). Le nombre de paramètres t_l dépend de la distance M_iM_{i+1}.

Cette approche respecte l'erreur δ_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 de la courbe P(t) au point $P(t_i)$.

Si δ est inférieure à la tolérance fixée ε_0, la procédure s'arrête ($\delta<\varepsilon_0$), si δ dépasse ε_0 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 $P_1(t)$ et $P_2(t)$.
Si la tolérance ε_0 est supérieure à δ, la procédure s'arrête pour la courbe considérée, sinon la courbe sera subdivisée elle aussi de la même manière.

Continuité d'ordre 2

Pour une continuité d'ordre 2, les équations (59.a) et (59.b) devront toutes les deux être satisfaites. De même, il faut choisir (n+1) points M_i de la courbe P(t) donnée pour un paramètre équidistant $t_i=i/n$ (i=0,...,n) et les insérer dans la représentation paramétrée de la courbe Q(t).

Dans ce cas m=5, l'erreur $\delta = \sum_{i=0}^{n} \delta_i^2$ devient une fonction non linéaire en λ_1 et μ_1 mais linéaire en λ_2 et μ_2. Pour éviter les méthodes non linéaires, l'algorithme est partagé en deux parties:
D'abord, choisir λ_1 et μ_1, puis minimiser l'erreur δ (pour les λ_1 et μ_1 constant) par la méthode des moindres carrés en s'aidant des conditions suivantes

$$\frac{\partial \delta}{\partial \lambda_2} = 0 \quad \text{et} \quad \frac{\partial \delta}{\partial \mu_2} = 0 \qquad (63)$$

les équations (63) conduisent à un système linéaire en λ_2 et μ_2, la solution de (63) sera minimisée par l'optimisation des paramètres dans (61).

Si la tolérance ε_0 est inférieure à δ, la procédure s'arrête, si $\varepsilon_0>\delta$, alors λ_1 et μ_1 doivent être changés par une optimisation non linéaire dont le but de réduire l'erreur δ.

La généralisation 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 (58).

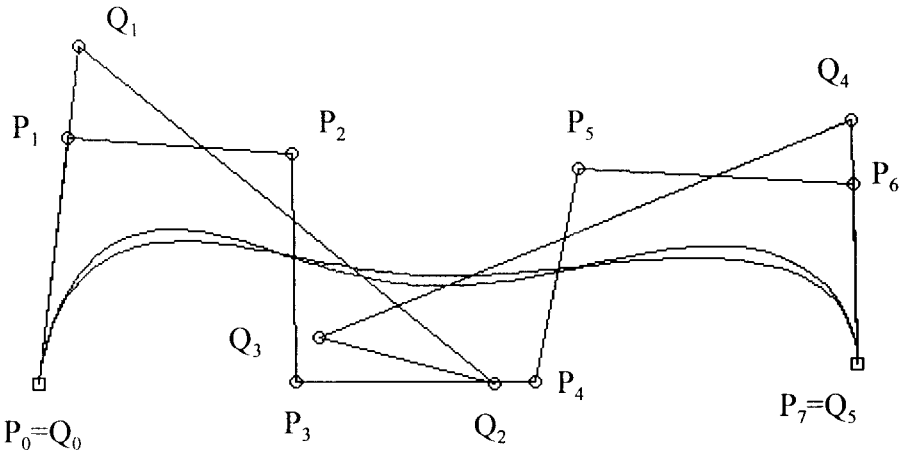

Figure III.8.a. Courbe de Bézier de degré 7 réduite au degré 5

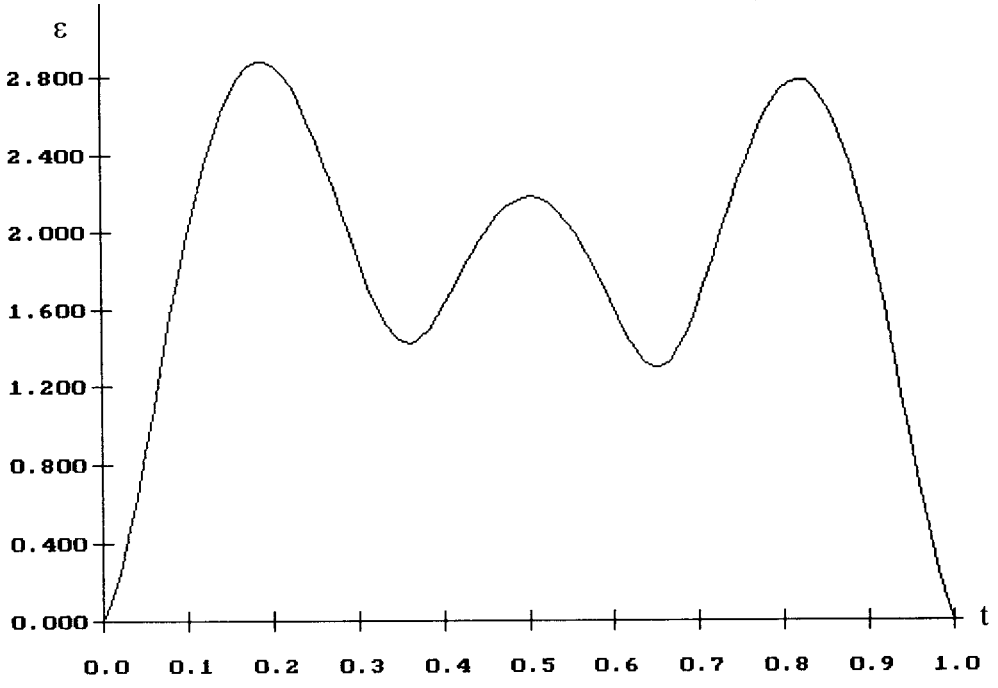

Figure III.8.b. Variation de l'erreur d'approximation

Algorithme HOS

Données: np nombre de points de la courbe
P_i (i=0, 1, ...,n) pôles de la courbe
k ordre de continuité
ε_0 tolérance

m=2.k+1
Pour j=0, 1, ...,np
t_j=j/np
M_j=P(t_j)

Fin pour
$L=4$; $\varepsilon_l=0$; $r=1$
Calculer μ (longueur du polygone $\{M_j\}$)
(*) Calculer les λ_l et μ_l ($l=1, 2, ..., k$) par un algorithme de résolution non linéaire
Calculer les pôles Q_l ($l=1, 2, ..., m$) par (96)
Pour $j=0, 1, ...,np$
Calculer Δc_j

(**) $$\tilde{t}_j = t_j + \frac{\Delta c_j}{\mu}$$

$$D_j = \left|Q(t_j) - M_j\right|$$

$$\tilde{D}_j = \left|Q(\tilde{t}_j) - M_j\right|$$

Si $D_j > \tilde{D}_j$ alors aller à (***)
Sinon $\Delta c_j = \Delta c_j/2$ aller à (**)
(***) Si $D_j > \varepsilon_l$ alors $\varepsilon_l = D_j$ (ε_l maximum des D_j)
$t_j = \tilde{t}_j$
Fin pour
$r=r+1$
Si $\varepsilon_l \geq \varepsilon_0$ alors
Si $r<L$ alors aller à (*)
FinSi
FinSi

Résultats: Les pôles Q_i ($i=0, 1, ..., m$)
ε_l erreur de l'approximation
Fin.

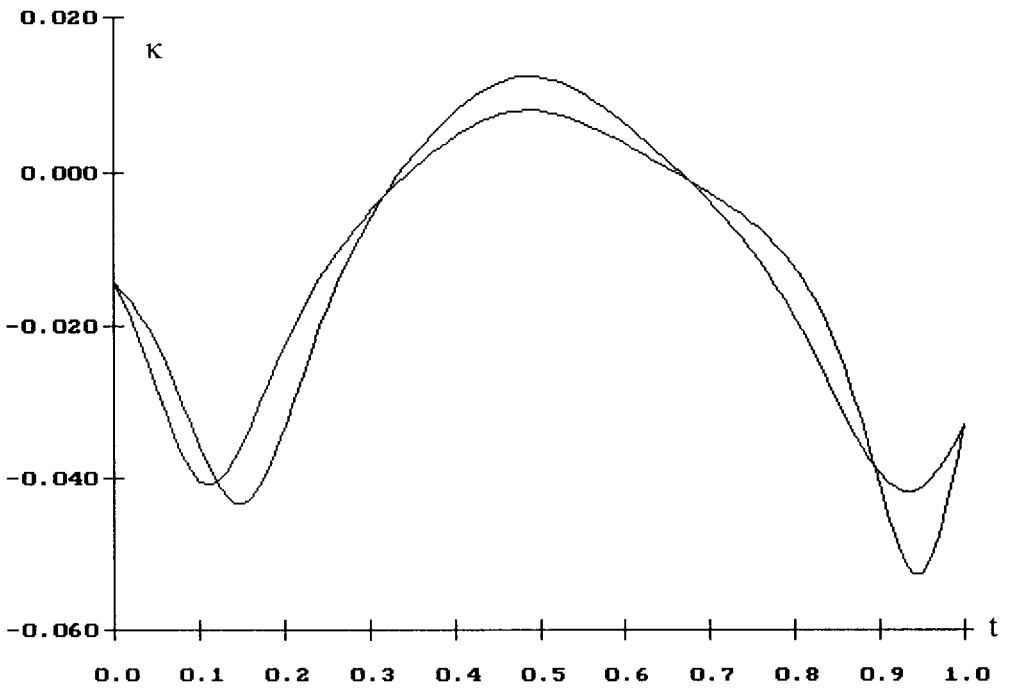

Figure III.8.c. Courbure des deux courbes

III.1.3.2.2. Algorithme de sub-réduction

L'algorithme global de subréduction 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'écart entre celle ci et la courbe obtenue. Si l'erreur est inférieure à la tolérance fixée, la procédure s'arrête sinon la courbe sera subdivisée en deux courbes sur lesquelles le processus s'appliquera.

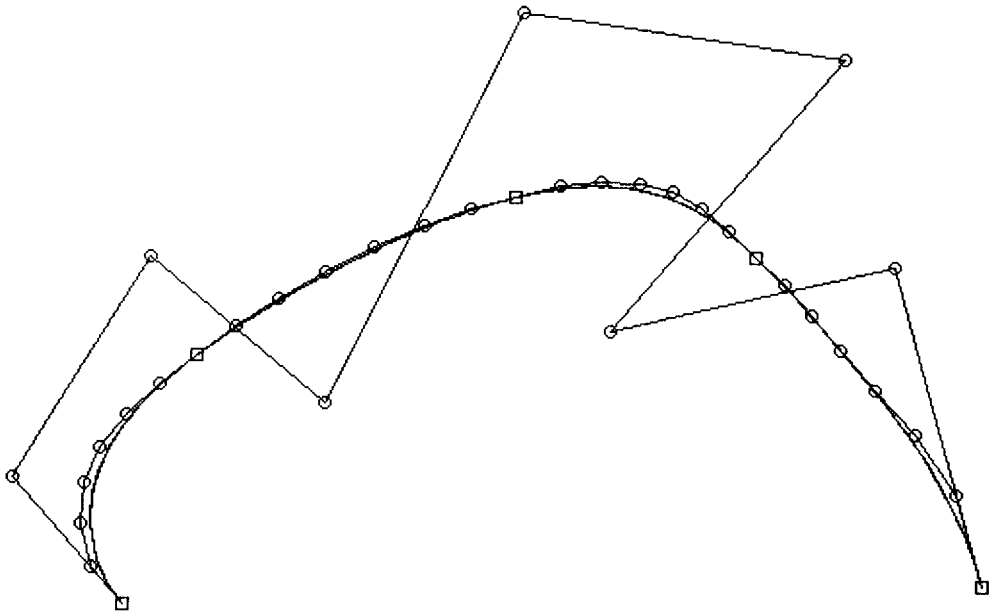

Figure III.9.a. Courbe de Bézier de degré 8 réduite en 4 courbes de degré 7 (k=3) par l'algorithme SUB-HOS avec ε_0=0.01

Algorithme SUB-HOS

Données : np nombre de points de la courbe

P_i^0 (i=0, 1, ..., n) polygone de la courbe de degré n.

k: Ordre de continuité

ε_0: tolérance de position

n1=0 ; n2=0
Pour j=0, 1, ..., n1
HOS (np, P^j, k, $\varepsilon_0 \rightarrow Q, \varepsilon_1$)
Si $\varepsilon_1 > \varepsilon_0$ alors
SUB ($P^j \rightarrow$ P0, P1)
n1=n1+2
Pour i=0, 1, ..., n
$P_i^{n1-1} = P0_i$; $P_i^{n1} = P1_i$
Fin pour
Sinon faire
Pour j=0, 1, ..., n-k
$Q_i^{n2} = Q_i$
Fin pour
n2=n2+1

Fin Si
Fin pour

Résultats : Q_i^j (i=0, 1, ..., n-k) (j=0, 2, ..., n2-1)
n2 : nombre de courbes de degré (m=2.k+1).
Fin.

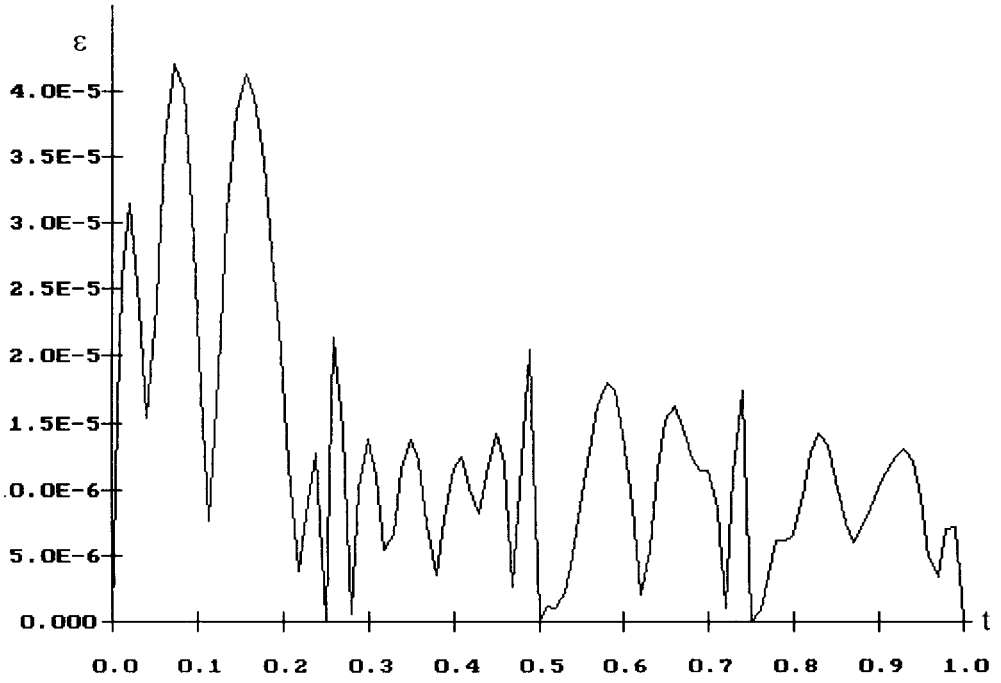

Figure III.9.b. Variation de l'erreur d'approximation pour la courbe sub-réduite précédente

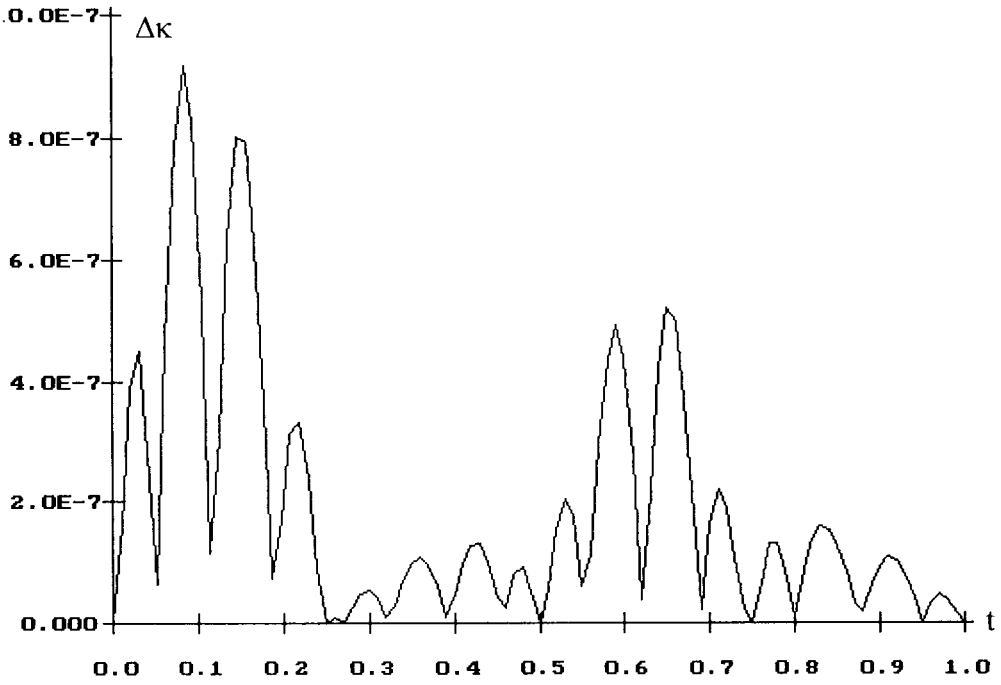

Figure III.9.c. Variation de l'erreur sur la courbure pour la courbe précédente

III.1.4. Méthode de Watkins et Worsey

Se basant sur le théorème de l'alternation [*Watkins & Worsey '88*] Watkins et Worsey ont proposé un algorithme de réduction de degré des courbes de Bézier non rationnelle qui 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.

III.1.4.1. Algorithme de réduction

Un des principaux résultats dans le domaine des meilleures approximations est le théorème de l'alternation, repris sous cette forme:

Soit $T_i(x)$ est le polynôme de Tchebycheff de degré i.
Ces polynômes sont déterminés par:

$$T_i(y_k) = (-1)^k \quad (k=0,...,i) \quad \text{où} \quad y_k = \cos\left(\frac{k.\pi}{i}\right) \tag{64}$$

T_i étant de degré i. Pour $t\in[-1, 1]$, on doit écrire:

$$T_i(t) = \cos(i.\arccos(t)) \Rightarrow \|T_i\| = 1, \quad \forall i$$

Si on considère que $Q_{n-1}(t)$ est une approximation de $P_n(t)$ alors l'erreur e est donnée par:

$$e(x)=P_n(t)-Q_{n-1}(t)=\alpha_n.T_n(t) \tag{65}$$

Donc $\|e\| = |\alpha_n|$

Alors $Q_{n-1}(t)$ est la meilleure approximation polynomiale de degré n-1 de $P_n(t)$ dans l'intervalle [-1, 1] d'après le théorème de l'alternation [*Watkins & Worsey '88*]

Si l'intervalle considéré est un intervalle [a, b] inclus dans [-1,1], alors les polynômes de Tchebycheff seront transformés en T_i^*

$$T_i^*(t) = T_i\left(\frac{2.t-a-b}{b-a}\right); \quad a \le t \le b \tag{66}$$

Comme dans le cas des courbes de Bézier l'intervalle est [0,1] les polynômes de Tchebycheff transformés s'écriront sous la forme suivante:

$$T_i^*(t) = T_i(2.t-1); \quad 0 \le t \le 1 \tag{67}$$

Dans ce qui va suivre les polynômes de Tchebycheff à utiliser sont des polynômes transformés.

Il est important de noter qu'en général, $Q_{n-1}(t)$ ne respecte pas la continuité aux extrémités par rapport à $P_n(t)$. 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 les premier et dernier pôles. Après cette modification, il est à noter que la courbe obtenue n'est pas la meilleure approximation dans la norme uniforme.

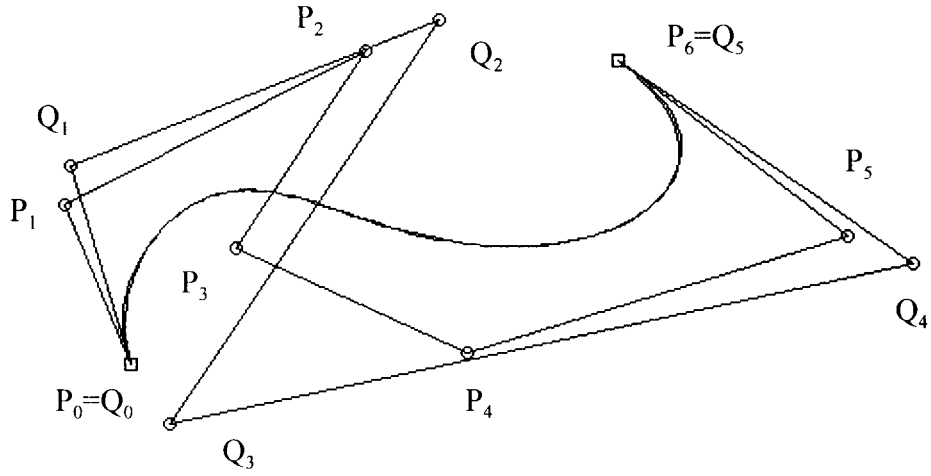

Figure III.10.a Courbe de degré 6 réduite au degré 5 par la méthode de Watkins & Worsey

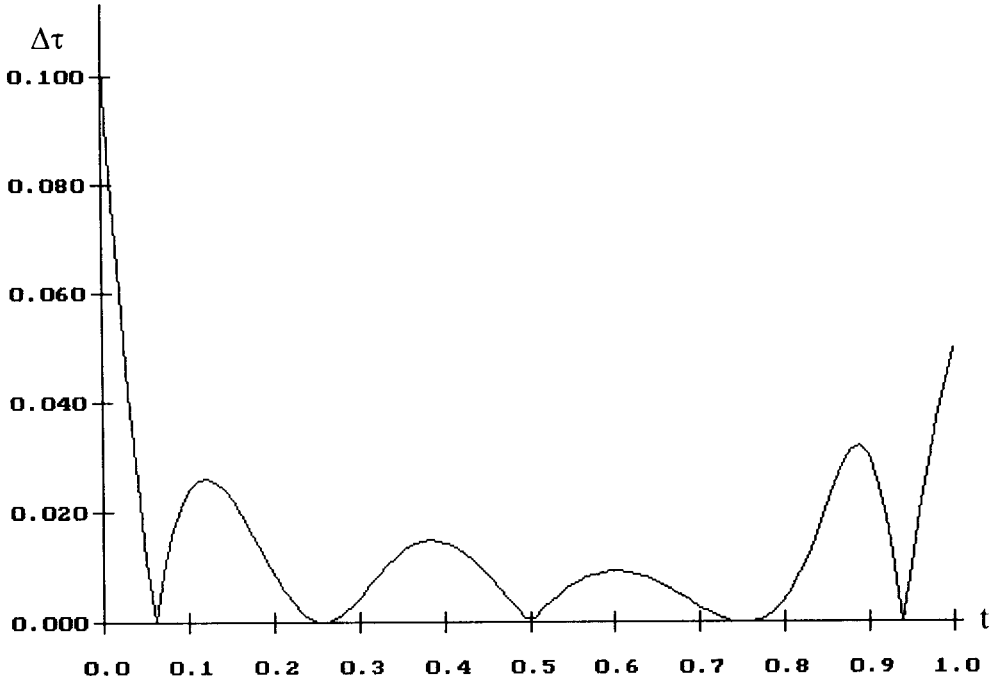

Figure III.10.b Variation de l'erreur sur la tangente

La figure III.10.a représente une courbe de Bézier non rationnelle P(t) de degré 6 réduite par l'algorithme WW en une courbe Q(t) de degré 5 et met en évidence la continuité de position aux extrémités, par contre la figure III.10.b montre une discontinuité de la tangente aux extrémités.

Pour réduire une courbe P(t) de degré n en courbe Q(t) de degré m=n-k, il suffit d'appliquer k fois le processus décrit.

Algorithme WW

Données: P_i (i=0, 1, ..., n) pôles de la courbe de degré n
k ordre de la réduction

Pour i=0, 1, ..., n
$V_P(i+1)=P_i$
Fin pour
Pour i=1, 2, ..., n
$V_a(i)=0$
Pour j=1, 2, ..., n
$V_a(i)=V_a(i)+(T_{BM}.T_{MT})_{ij}.V_P(j)$
Fin pour
Fin pour
m=n-k
Pour i=1, 2, ..., m+1
$V_P(i)=0$
Pour j=1, 2, ..., m+1
$$V_P(i) = V_P(i) + \left(T_{MT}^{-1}.T_{BM}^{-1}\right)_{ij}.V_a(j)$$
Fin pour
Fin pour
Pour i=1, 2, ..., m-1
$Q_i=V_P(i+1)$
Fin pour
$Q_0=P_0$
$Q_m=P_n$

Résultats: Q_j (j=0, 1, ..., m) courbe réduite de degré m=n-k
Fin.

III.1.4.2. Algorithme de sub-réduction

On se propose de déterminer l'erreur commise dans le processus de réduction ainsi que pour l'interpolation forcée. On décrira alors un nouvel algorithme de sub-réduction 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(t) de degré n, il est possible de déterminer a priori combien de fois doit on subdiviser P(t) 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 contrôler la linéarité après chaque subdivision.

On suppose que l'erreur commise pour la réduction d'une courbe P(t) est ε. Alors, puisque l'erreur est donnée par $\alpha_n.T_n(t)$ et les valeurs extrémales de $T_n(t)$ 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(t) et la courbe réduite avec interpolation forcée Q'(t), alors on peut calculer $\|P - Q'\|$ par l'inégalité suivante:

$$\|P-Q'\| \le \|P-Q\| + \|Q-Q'\| \qquad (68)$$

La fonction Q-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 ±ε. Donc, d'après la propriété de l'enveloppe convexe des courbes de Bézier:

$$\|Q - Q'\| \leq \varepsilon \tag{69}$$

Les inégalités (68) et (69) 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.ε.

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.

De plus on peut montrer que l'erreur due à la réduction de degré diminue par multiplication au facteur $\frac{1}{2^n}$ à chaque subdivision [*Watkins & Worsey '88*], ce qui donne après s subdivision une erreur égale à $\left(\frac{1}{2^{ns}}\right).\alpha_n$. 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:

$$s > \frac{1}{n}.\log_2\left(\frac{\alpha_n}{\varepsilon}\right) \tag{70}$$

Ces résultats conduisent au schéma de sub-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*]

Algorithme SUB-WW

Données: P_i (i=0, 1, ..., n) pôles 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, ..., 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
Calculer α_m
$ns = \frac{1}{m}.\mathrm{Log}_2\left(\frac{k.\alpha_m}{2.\varepsilon_0}\right)$
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;
WW (Q0, 1 → Q)
Pour j=0, 1, ..., m-1
$R_j^c = Q_j$
Fin pour
Fin pour
c=c+1;
WW (S, 1 → Q)
Pour j=0, 1, m-1
$R_j^c = Q_j$
Fin pour
Fin pour
a=b+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, ..., m) et (i=1, 2, ..., nb) nb courbe de degré m=n-k
Fin.

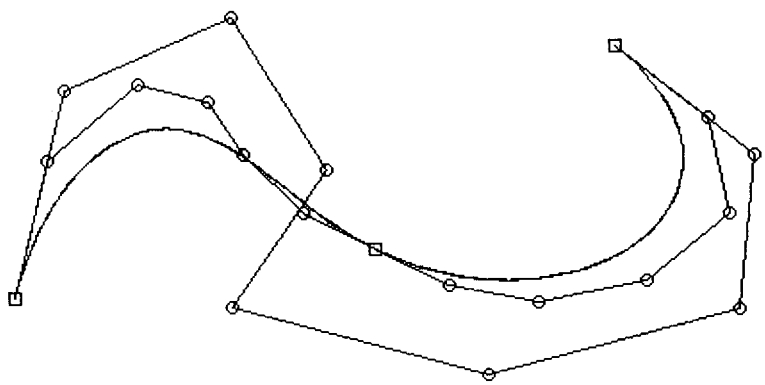

Figure III.11.a. Courbe de degré 8 réduite en deux courbes de degré 6 par l'algorithme SUB-WW

La figure III.11.a représente une courbe de Bézier non rationnelle de degré 8 réduite en deux courbes de degré 6 par l'algorithme SUB-WW avec ε_0=0.01 dont la variation de l'erreur de position est représentée en figure III.11.b, celle ci nous permet de constater que la

tolérance fixée est respectée et que l'erreur de position est nulle aux extrémités des courbes. L'approximation obtenue a un comportement symétrique. La figure III.11.c qui représente la variation de l'erreur sur la tangente montre une discontinuité de la tangente aux extrémités.

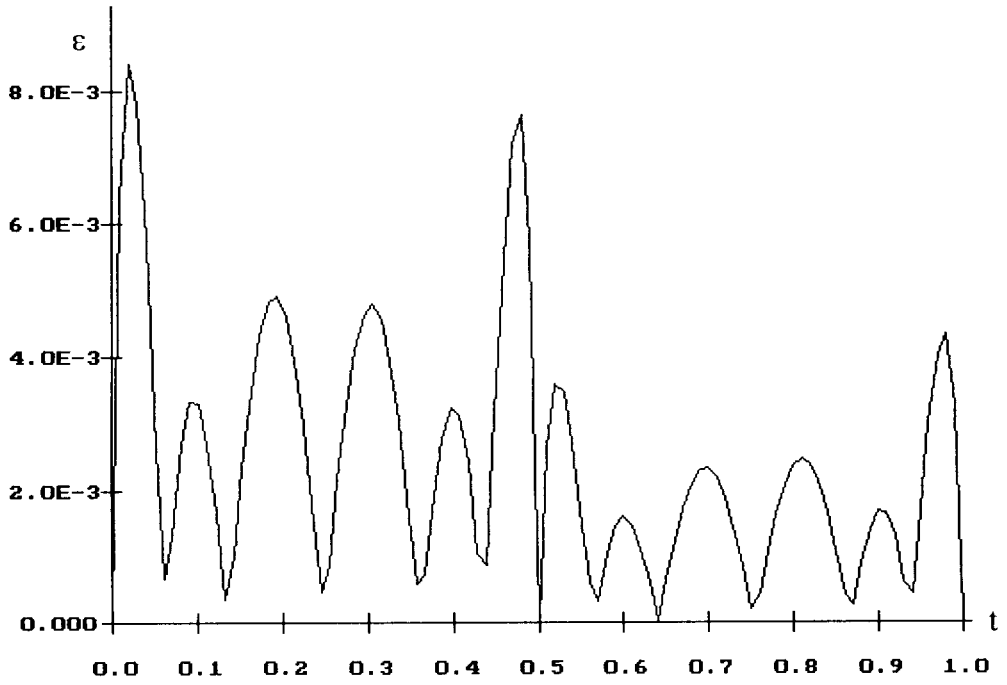

Figure III.11.b. Variation de l'erreur de position

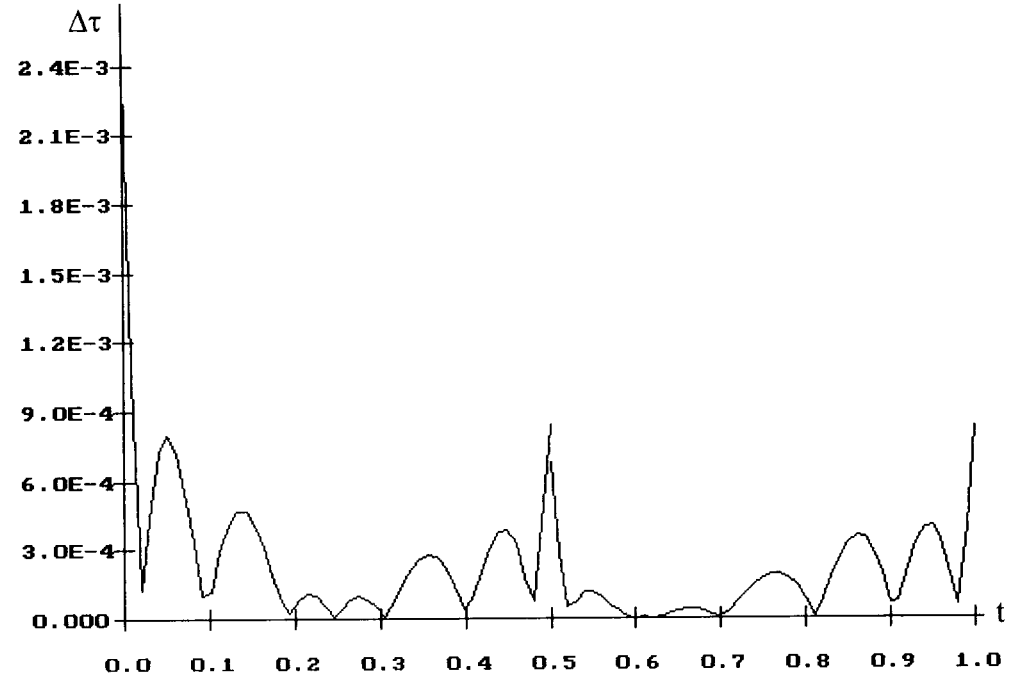

Figure III.11.c. Variation de l'erreur sur la tangente

III.1.4.3. Conversion de base

Puisque la conversion de bases polynomiales est une transformation linéaire, la conversion de base entre les formes 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 Tchebycheff, on utilise l'équation suivante: $V_a=V_P.T_{BM}.T_{MT}$
Pour la conversion inverse on écrit: $V_P = V_a . T_{MT}^{-1} . T_{BM}^{-1}$
où T_{BM} est la matrice qui convertit les coefficients de la base de Bézier à la base monomiale, T_{MT} est la matrice qui convertit les coefficients de la base monomiale à la base de Tchebycheff, T_{BM}^{-1} et T_{MT}^{-1} sont leurs inverses respectives.

Les matrices T_{BM} et T_{BM}^{-1} sont connues explicitement, la matrice T_{MT}^{-1} peut être calculée par une relation récursive pour les polynômes de Tchebycheff transformés. T_{MT} est aussi déterminée 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 base de Bernstein 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 $T_{BM}.T_{MT}$ et $T_{MT}^{-1}.T_{BM}^{-1}$ 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'estimation 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}: i=1, 2, ..., n+1 et j=i, i+1, ..., n+1

$$\left(T_{BM}\right)_{ij} = (-1)^{i+j}.\binom{n}{i-1}.\binom{n+1-i}{n+1-j} \tag{71}$$

Toutes les autres composantes sont nulles.

T_{BM}^{-1} : i=1, 2, ..., n+1 et j=i, i+1, ..., n+1

$$\left(T_{BM}^{-1}\right)_{ij} = \binom{j-1}{j-i} \Big/ \binom{n}{i-1} \tag{72}$$

Toutes les autres composantes sont nulles.

T_{MT}:

$$\left(T_{MT}\right)_{11} = 1,\ \left(T_{MT}\right)_{21} = \frac{1}{2},\ \text{et}\ \left(T_{MT}\right)_{22} = \frac{1}{2}$$

si i≥3 et j=1 alors: $\left(T_{MT}\right)_{ij} = \binom{2.i-3}{i-3} \Big/ 2^{2.i-3}$ (73)

si i≥3 et j≥2 alors: $\left(T_{MT}\right)_{ij} = \binom{2.i-3}{i-j} \Big/ 2^{2.i-3}$

Toutes les autres composantes sont nulles.

T_{MT}^{-1}:

$\left(T_{MT}^{-1}\right)_{11} = 1,\ \left(T_{MT}^{-1}\right)_{21} = 1,\ \left(T_{MT}^{-1}\right)_{22} = 2$

si j≤i alors: $\left(T_{MT}^{-1}\right)_{ij} = 4.\left(T_{MT}^{-1}\right)_{i-1,j-1} - 2.\left(T_{MT}^{-1}\right)_{i-1,j} - \left(T_{MT}^{-1}\right)_{i-2,j}$ (74)

si j>i alors: $\left(T_{MT}^{-1}\right)_{ij} = 0$

Toutes les autres composantes sont nulles.

Quelques produits vectoriels $T_{BM}.T_{MT}$ et $T_{MT}^{-1}.T_{BM}^{-1}$ pour quelques valeurs de n sont donnés en (*Cf Annexe D*).

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 le mieux numériquement que les bases de puissance. C'est à dire, qu'elles augmentent la stabilité de l'algorithme, puisque le conditionnement de ces bases est meilleur. 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 entre les bases de Bernstein et Tchebycheff, du point de vue 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 virgules flottantes est modestement réduit.

III.1.5. Méthode de Eck

Des équations (14) dérivent deux formules d'extrapolation récursives distinctes pour la détermination des Q_i à partir des P_i (17.a) polygonel et (17.b) polygone2 respectivement R_i et S_i.
Ces deux formules sont simplement des approximations, et on peut immédiatement constater que R(t) produit une approximation raisonnable au voisinage de P_0 et S(t) d'une manière régressive au voisinage de P_n.
Forrest, a proposé de combiner ces deux formules en prenant la moitié gauche des pôles à partir de R et la moitié droite à partir de S, qui est le principe de l'algorithme **FOR**.

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:

$$Q_i = (1-\lambda_{i,n}).R_i + \lambda_{i,n}.S_i \quad (i=0,\ldots,n-1) \tag{75}$$

pour définir les nouveaux pôles Q_i (i=0, 1, ..., n-1).
les facteurs $\{\lambda_{i,n}\}_{i=0}^{n-1}$ sont déterminés par

$$\lambda_{i,n} = 2^{1-2.n}.\sum_{j=0}^{i}\binom{2.n}{2.j} \quad (i=0,\ 1,\ \ldots,\ n-1) \tag{76}$$

de plus l'erreur d'approximation est donnée par

$$d(P,Q) = 2^{1-2.n}.\left|\Delta^n P_0\right|. \tag{77}$$

Remarques:

(1) Si $\Delta^n P_0=0$ alors les facteurs $\lambda_{i,n}$ peuvent être choisis arbitrairement puisque $R_i=S_i$ pour tout i.

(2) On a $0<\lambda_{0,n}<\lambda_{1,n}<\ldots<\lambda_{n-1,n}<1$.

Un autre avantage de la formule (76) 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 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.

III.1.5.1.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 tout simplement obtenue en définissant $\lambda_{0,n}=0$ et $\lambda_{n-1,n}=0$ et les autres facteurs sont toujours déterminés par l'équation (76). Il

est évident que le 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 $C^{\alpha-1}$ 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.α-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 $\alpha \geq 2$ ne sont pas connues à l'exception de:

$$T_{m,0}(t) = T_m(t) \quad \text{et} \quad T_{m,1}(t) = T_m\left(t.\cos\frac{\pi}{2.m}\right). \tag{78}$$

Donc, ces polynômes doivent être déterminés numériquement par l'intermédiaire de l'algorithme *« Type Remes»* [*Eck '93*], [*Bogaki, Weinstein & Xu '95*], [*Lachance '88*] 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

$$T_{m,\alpha}(t) = \sum_{i=0}^{m} c_{i,m,\alpha}.B_{i,m}\left(\frac{t+1}{2}\right), \quad t \in [-1,\ 1] \tag{79}$$

En calculant numériquement les coefficients $c_{i,m,\alpha}$. Ces coefficients obéissent évidemment à:

$$c_{i,m,\alpha} = c_{m-i,m,\alpha} = 0 \quad (i=0,\ 1,\ \ldots,\ \alpha) \tag{80}$$

Donc, puisque $T_{m,\alpha}$ est d'égale oscillation, les signes des coefficients non nuls sont alternés, c'est à dire:

$$c_{i,m,\alpha} = (-1)^{m+i}.\left|c_{i,m,\alpha}\right| \quad i=\alpha,\ \alpha+1,\ \ldots,\ m-\alpha \tag{81}$$

Le calcul des nouveaux coefficients précédants ne change pas le principe. Il suffit simplement de modifier la formule (75) par:

$$Q_i = \left(1-\lambda_{i,n,\alpha}\right).R_i + \lambda_{i,n,\alpha}.S_i \quad (i=0,\ 1,\ \ldots,\ n-1) \tag{82}$$

les facteurs $\left\{\lambda_{i,n,\alpha}\right\}_{i=0}^{n-1}$ sont déterminés par:

$$\lambda_{i,n,\alpha} = \begin{cases} 0 & (i=0,\ldots,\alpha-1) \\ \dfrac{1}{\gamma_{n,\alpha}}.\displaystyle\sum_{j=\alpha}^{i}\binom{n}{j}.\left|c_{j,n,\alpha}\right| & (i=\alpha,\ldots,n-\alpha-1) \\ 1 & (i=n-\alpha,\ldots,n-1) \end{cases} \tag{83}$$

où

$$\gamma_{n,\alpha} = \sum_{j=\alpha}^{n-\alpha}\binom{n}{j}.\left|c_{j,n,\alpha}\right|. \tag{84}$$

De plus, en utilisant le facteur de turbulence $s_{n,\alpha} = 2^{2.n-1}/\gamma_{n,\alpha}$ l'erreur d'approximation est donnée par: $d(P, Q)=s_{n,\alpha}.2^{1-2.n}.|\Delta^n P_0|$ (85)

Remarques:

(1) 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 $s_{n,0}=1$.

(2) On a $0=\lambda_{0,n,\alpha}= \ldots = \lambda_{\alpha-1,n,\alpha} < \lambda_{\alpha,n,\alpha} < \ldots < \lambda_{n-\alpha-1,n,\alpha} < \lambda_{n-\alpha,n,\alpha} = \ldots = \lambda_{n-1,n,\alpha} = 1$.

(3) Le cas n=2.α (exclu dans le théorème [*Eck '95*]) n'est pas intéressant puisqu'il représente simplement le schéma d'interpolation d'Hermite (construction de Forest), [*Sederberg & Farouki '92*].

Dans la pratique on ne calcule $\lambda_{i,n,\alpha}$ et $\gamma_{n,\alpha}$ qu'une seule fois pour chaque combinaison de n et $\alpha \geq 1$ (la solution pour $\alpha=0$ est donnée dans (76)). Alors les résultats obtenus pour $\{\lambda_{i,n,\alpha}\}_{i=0}^{n-1}$ et $s_{n,\alpha}$ doivent être enregistrées dans un fichier. Pour une application, on les rappelle et on calcule les nouveaux coefficients par (82), [*Eck '93*].

i	α 0	1	2	3
0	0.00003	0	0	0
1	0.00369	0.00287	0	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: Facteurs $\lambda_{i,8,\alpha}$ pour α = 0, 1, 2, 3

n	α 0	1	2	3
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 EC

Données: P_i (i=0, 1, ..., n) pôles de la courbe de degré n
k ordre de la réduction
α ordre de continuité

m=n
Pour i=0, 1, ...,n
$Q_i=P_i$
Fin pour
Pour j=1, 2, ..., k

BEN1 (Q, 1 → R)
BEN2 (Q, 1 → S)
Lecture ($\lambda_{i,m,\alpha}$)
Pour i=0, 1, ..., m-1
$Q_i=(1-\lambda_{i,m,\alpha}).R_i+\lambda_{i,m,\alpha}.S_i$
Fin pour
m=m-1
Fin pour

Résultats: Q_i (i=0, 1, ..., m) courbe de degré m=n-k
Fin.

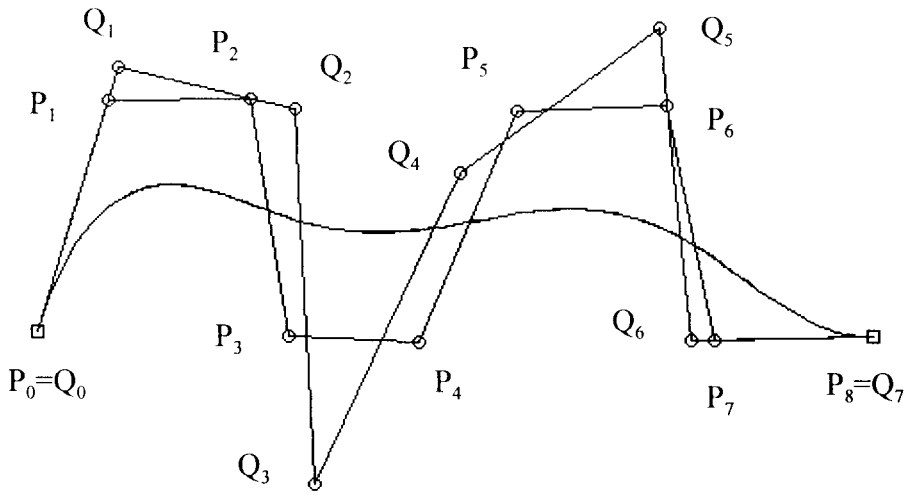

Figure III.12.a Courbe de Bézier de degré 8 réduite au degré 7 par l'algorithme EC en respectant la continuité de la courbure (α=3)

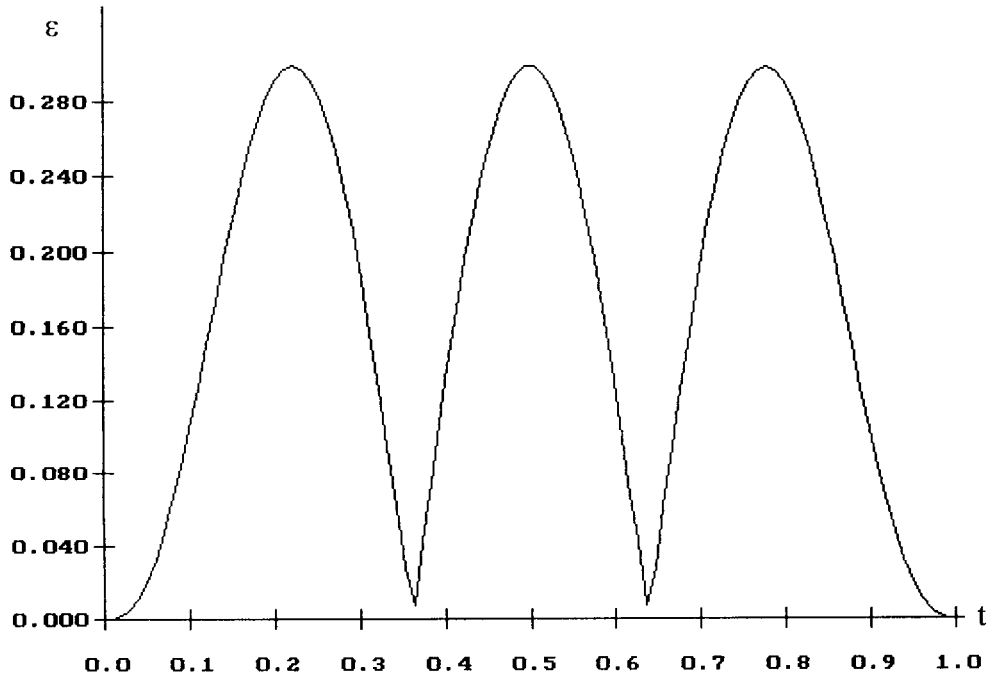

Figure III.12.b. Variation de l'erreur de position

La figure III.12.a représente une courbe de Bézier non rationnelle de degré 8 réduite au degré 7 par l'algorithme de Eck en respectant une continuité de la courbure que nous pouvons remarquer sur la figure III.12.c. La figure III.12.b montre la nature uniforme de l'approximation de Eck.

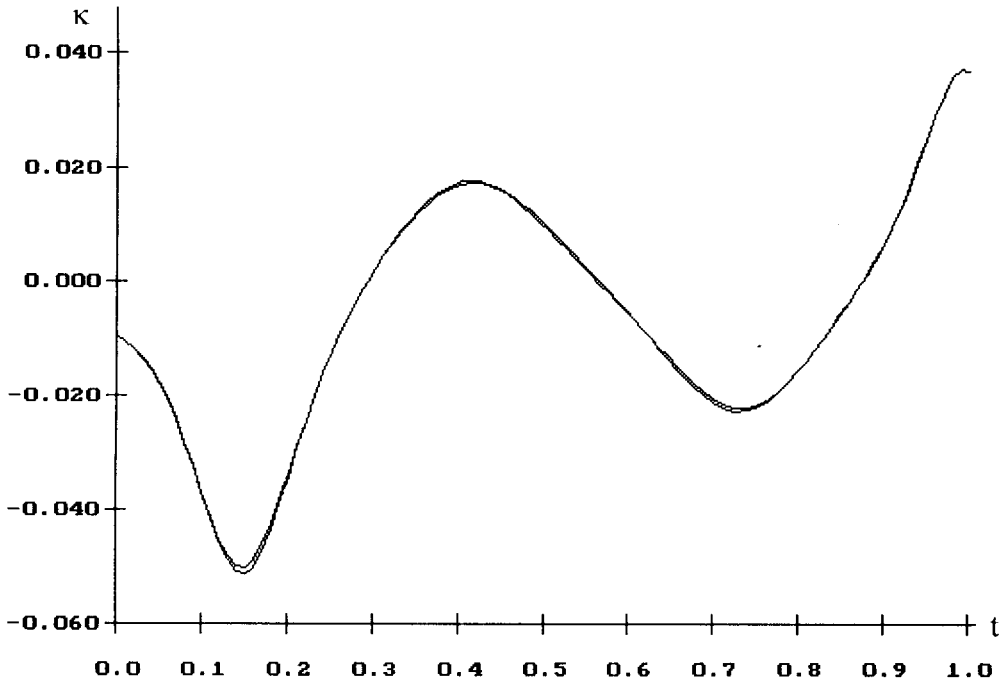

Figure III.12.c Variation des courbures

III.1.5.2. Algorithme de sub-réduction

Maintenant, on peut soulever la question sur le ce schéma de réduction de degré en respectant une tolérance de position ε fixée au préalable. a priori l'erreur estimée (77) 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, ..., ns) alors les n différences de pôles sur chacun des ns+1 segment de courbe est $1/(ns+1)^n$ fois la différence originelle $\Delta^n P_0$ 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 (85) est

$$ns = \left[\frac{1}{4} \cdot \left(\frac{2 \cdot s_{n,\alpha} \cdot \left| \Delta^n P_0 \right|}{\varepsilon} \right)^{1/n} \right]. \qquad (86)$$

Evidement, 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 degré 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 par 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 [*Brunett, Screiber & Braun '96*]

$$\left(\int_0^1 \frac{(P(t)-Q(t))^2}{\sqrt{t.(1-t)}} . dt \right)^{1/2} \tag{87}$$

puisque les polynômes classiques 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 peut valider ces arguments 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 $\varepsilon/(n-m)$ à chaque réduction comme précédemment en subdivisant k-fois si nécessaire.

Ce qui nous conduit à l'algorithme de sub-réduction suivant: [*Eck '93*]

Algorithme SUB-EC

Données: P_i (i=0, 1, ..., n) pôles de la courbe de degré n
k ordre de la réduction
ε_0 tolérance de position
α ordre de continuité

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 $\Delta^m P_0$

$$ns = \left\lfloor \frac{1}{4} . \left(\frac{2.k.s_{m,\alpha} . \left| \Delta^m P_0 \right|}{\varepsilon_0} \right)^{1/m} \right\rfloor .$$

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
EC (Q0, 1, $\alpha \rightarrow$ Q)
Pour j=0, 1, ..., m-1
$R_j^c = Q_j$
Fin pour
Fin pour
c=c+1
EC (S, 1, $\alpha \rightarrow$ Q)
Pour j=0, 1, m-1
$R_j^c = Q_j$
Fin pour
Fin pour
a=b+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, ..., m) et (i=1, 2, ..., nb) nb courbe de degré m=n-k
Fin.

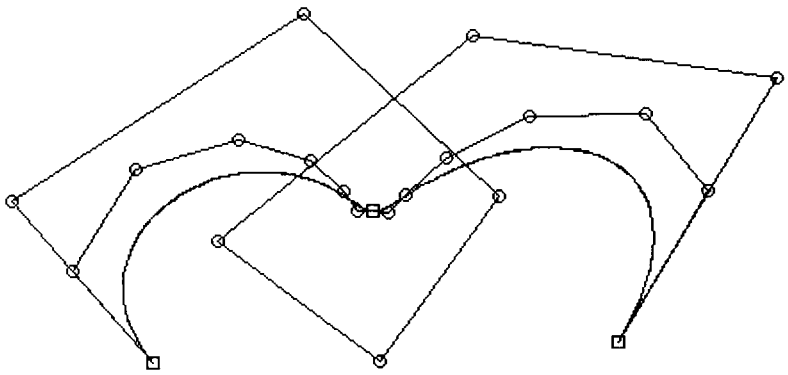

Figure III.13.a. Courbe de Bézier de degré 8 réduite en deux courbes de degré 7 par l'algorithme SUB-EC

La figure III.13.a représente une courbe de Bézier non rationnelle de degré 8 réduite en deux courbes de degré 7 par l'algorithme SUB-EC avec α=3 et ε_0=0.01. L'approximation uniforme symétrique obtenue est représentée par la variation de l'erreur de position sur la

figure III.13.b, alors que la continuité de la courbure est montrée sur la figure III.13.c qui représente la variation de l'erreur sur la courbure.

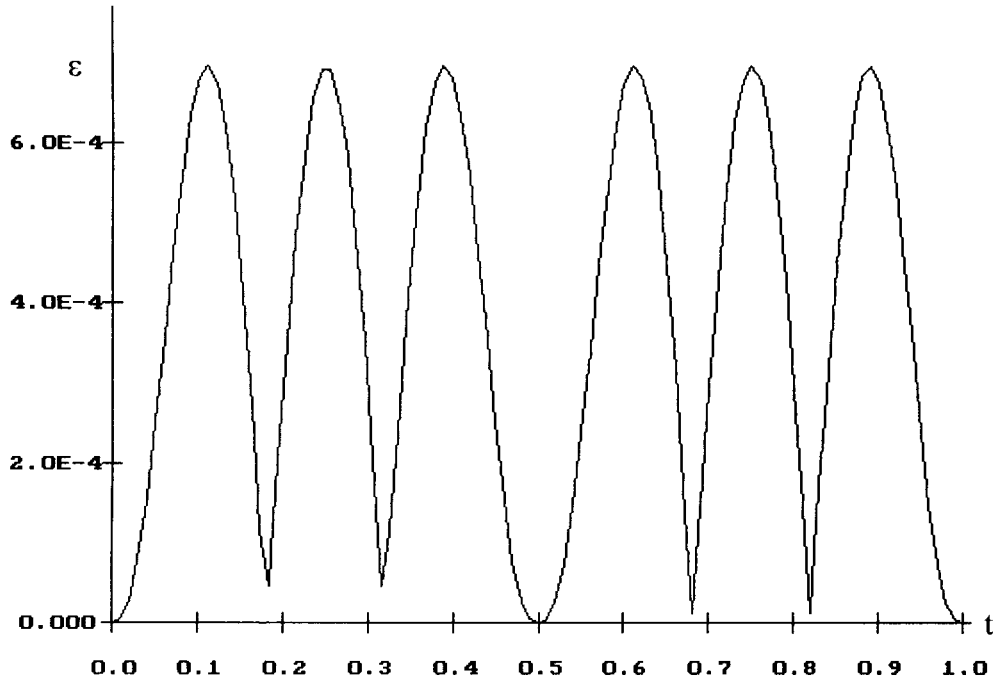

Figure III.13.b. Variation de l'erreur de position

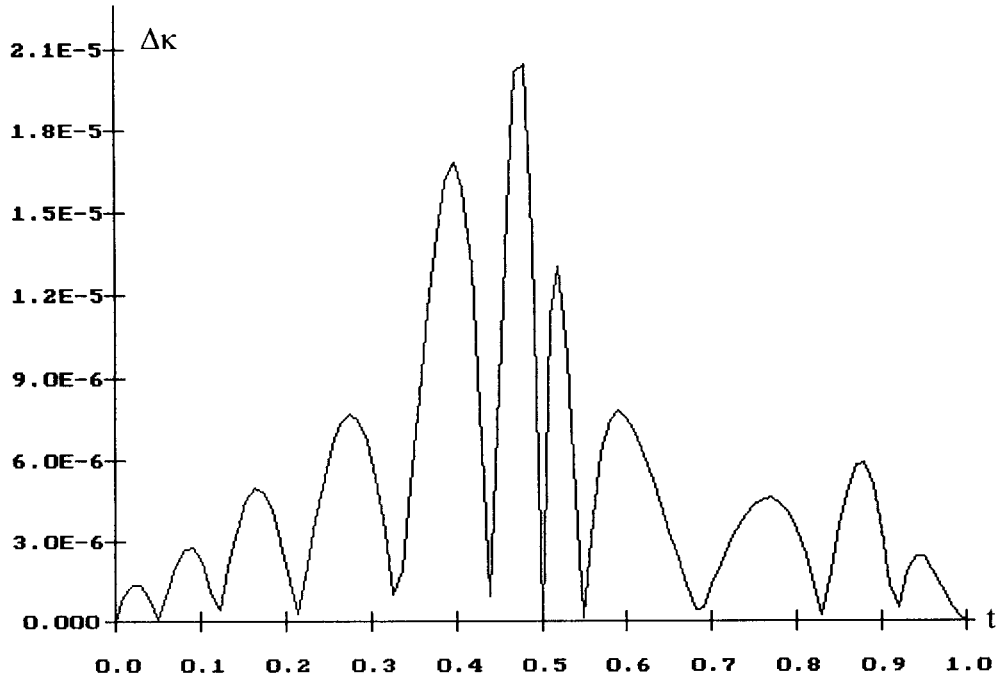

Figure III.13.c Variation de l'erreur sur la courbure

III.2. Etude comparative des méthodes de réduction

III.2.1. Etude critique

La réduction de degré étant une approximation, nous devons prendre en considération dans notre étude critique les critères s'y afférant qui sont la nature, la norme et l'erreur d'approximation, sans 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 général a 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 d'équations non linéaires, qui engendrent un temps de calcul très lourd [*Hoschek, Schneider & Wassum '89*]. 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 direct 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 Watkins et Worsey:*** Grâce aux polynômes de Tchebycheff, en utilisant la conversion de base nous retrouvons dans l'algorithme de Watkins et Worsey une notion très importante concernant la meilleure approximation, utilisant la norme de maximisation. 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 ainsi une continuité de position. L'algorithme global est d'une facilité d'implémentation mais nécessite des matrices de passage de la base Bernstein à la base de Tchebycheff et inversement qui doivent être calculées par un algorithme externe. Les résultats pratiques obtenus par l'auteur donnent une erreur d'approximation et un nombre de segments corrects.

- ***Algorithme de ECK:*** De la même manière que la méthode précédente 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 ce problème est maîtrisé grâce à une transformation des polynômes de Tchebycheff appelés alors contraints. Cette méthode se base sur la réduction par les polygones moyens dans laquelle l'auteur introduit les facteurs de pondération $\lambda_{i,n,\alpha}$ (pour une continuité d'ordre α-1: $C^{\alpha-1}$) qui dépendent 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. La nature de l'approximation est uniforme produisant une erreur jugée intéressante.

III.2.2. Comparaison 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 la 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 si on veut réaliser un transfert de données sans perte entre systèmes. 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'elles sont inclues dans la méthode de Eck. En effet, si on réduit dans l'algorithme de Eck les $\lambda_{i,n,\alpha}$=i/n, c'est à dire α=0 on retombe dans l'algorithme de Bensalah, de même si on choisit la valeur maximale pour α (m/2) suivant le nombre de pôles à obtenir après réduction, on retombe sur l'algorithme de Forrest; et finalement si on prend α=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.
Ne connaissant pas les résultats pratiques de l'algorithme de Eck, nous ne pouvons comparer qu'analytiquement les avantages et les inconvénients de cette méthode.

L'avantage principal de l'algorithme de Hoschek est la paramétrisation, mais sans utiliser la paramétrisation on ne risque pas de produire d'erreur supplémentaire. L'avantage principal de l'algorithme de Eck est la détermination du nombre de subdivision nécessaire pour respecter une tolérance donnée.
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, le respect d'un ordre de continuité élevé pour réduire dans certains cas d'un degré (exemple: de n=10 à m=9, ce qui donne k=4), et qu'on ne peut avoir de courbes réduites de degré $m \neq 2.k+1$ ou $m > 2.k+1$.
Les inconvénients 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.

III.2.3. Comparaison expérimentale

Considérons 14 courbes de Bézier non rationnelles du degré 4 au degré 18 choisies arbitrairement (définies en trois dimensions) sur lesquelles nous faisons subir des tests pratiques aux méthodes décrites dans ce chapitre

Pour l'étude comparative la méthode de Bensalah sera omise car elle donne en erreur d'approximation des valeurs très élevées (de l'ordre du mètre) incomparable aux autres méthodes. Son temps de calcul est pour la réduction seule identique au temps obtenu par la méthode de Forrest.

Pour la méthode de Eck nous ne pouvons faire de comparaison sur la qualité de l'approximation car nous ne 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 α=0, 1, 2, 3.

Les temps de calcul pour la méthode de Hoschek sont minimisés 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.

Les algorithmes sont implémentés dans l'environnement d'un logiciel de C.F.A.O (CONVER de l'IGM de l'université de Boumerdès) 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 6 Tests relatifs aux manipulations les plus rencontrées.
Les tests sont appliqués sur une machine P.C. 486/DX2 66 MHz avec coprocesseur mathématique. En fonction de la méthode le temps CPU est calculé après exécution d'un nombre défini de fois l'algorithme.

Test 1: n → 3 (Cf. Tableau 6 Annexe E)

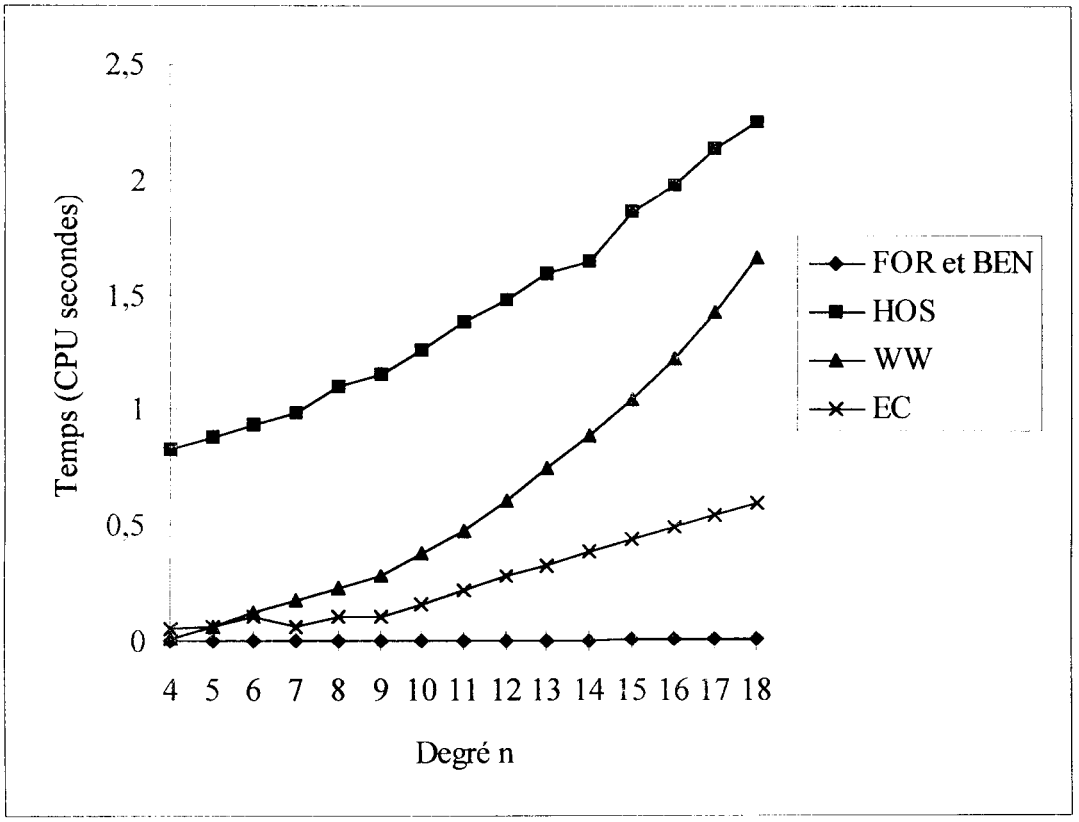

Figure III.14.a

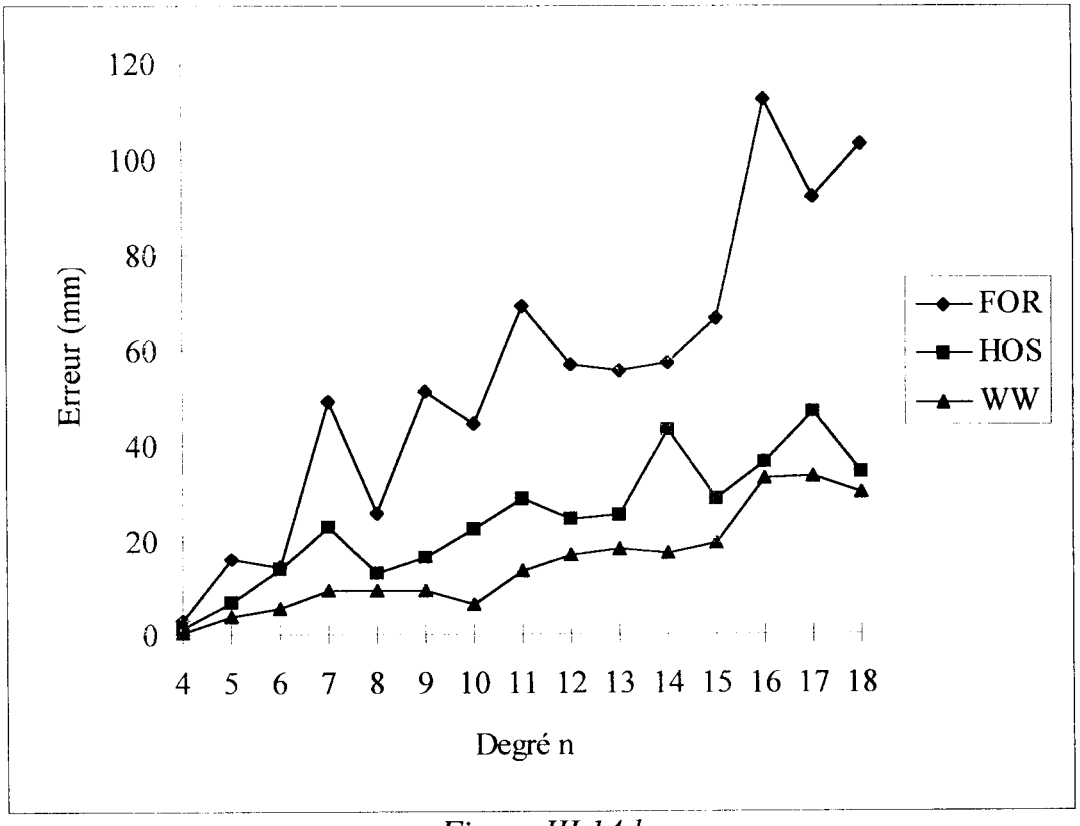

Figure III.14.b

Test 2: n → 5 (Cf. Tableau 7 Annexe E)

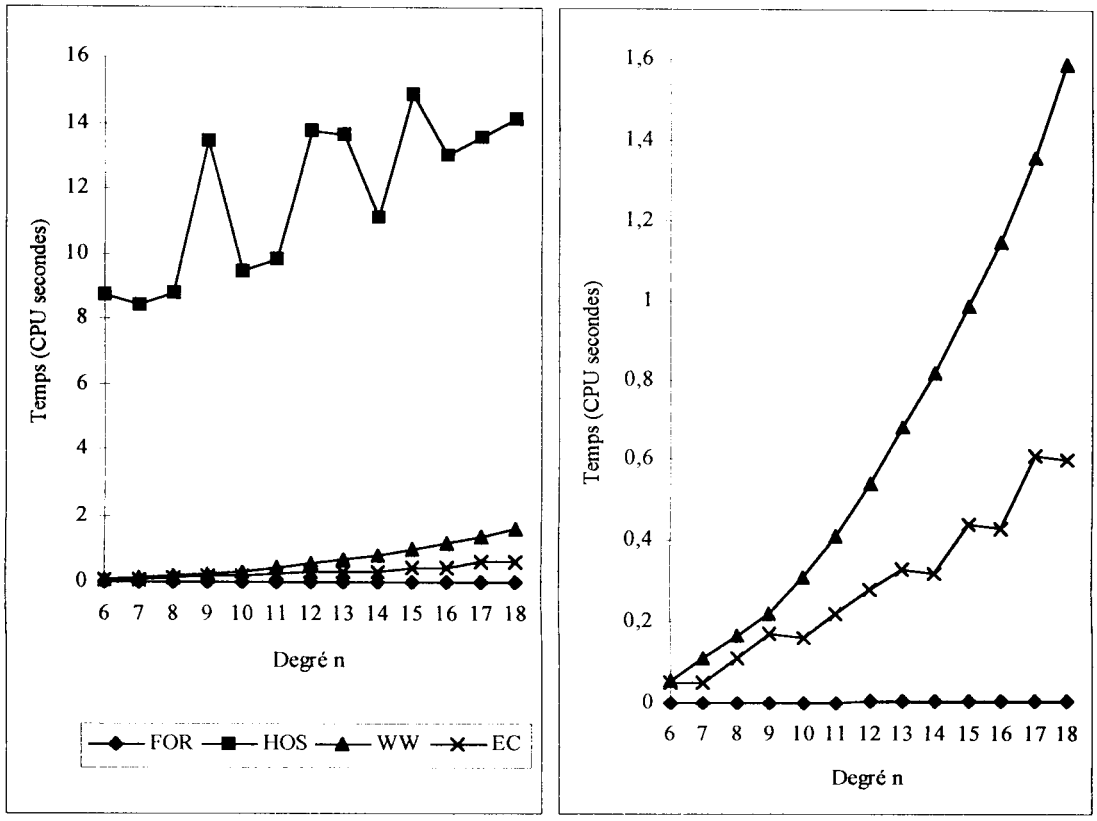

Figure III.15.a

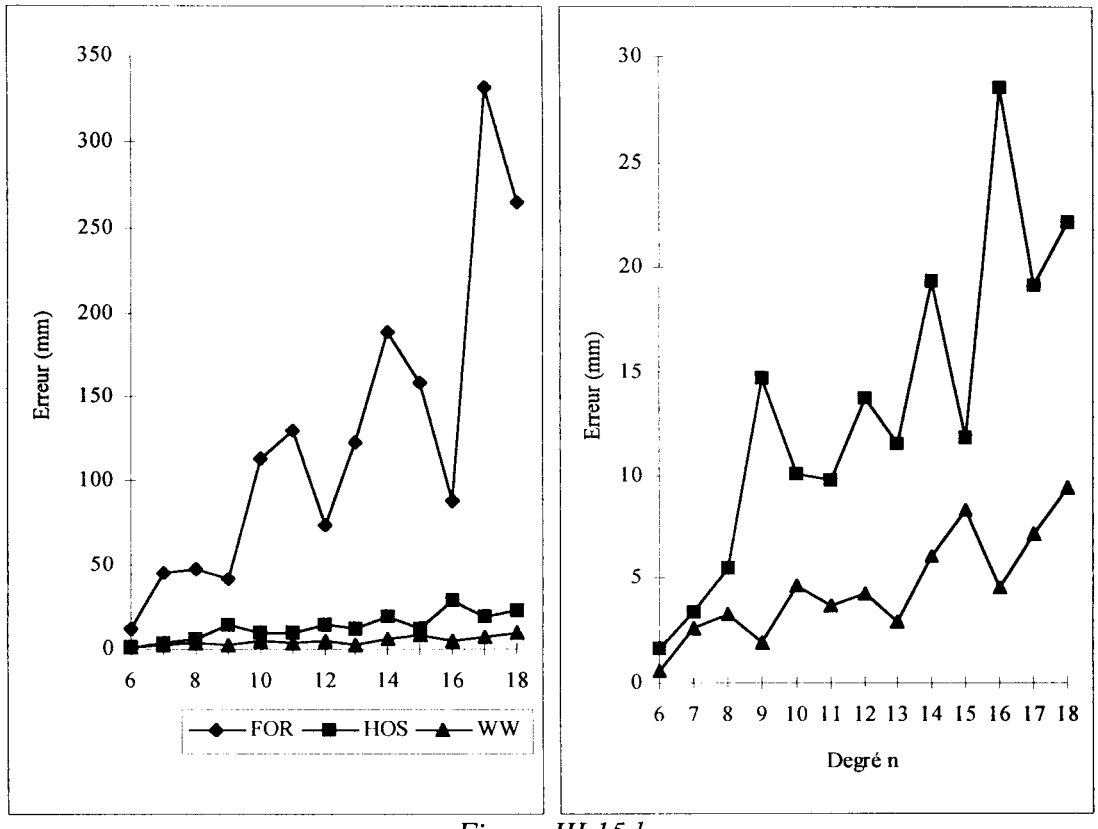

Figure III.15.b

Test 3: n → 7 (Cf. Tableau 8 Annexe E)

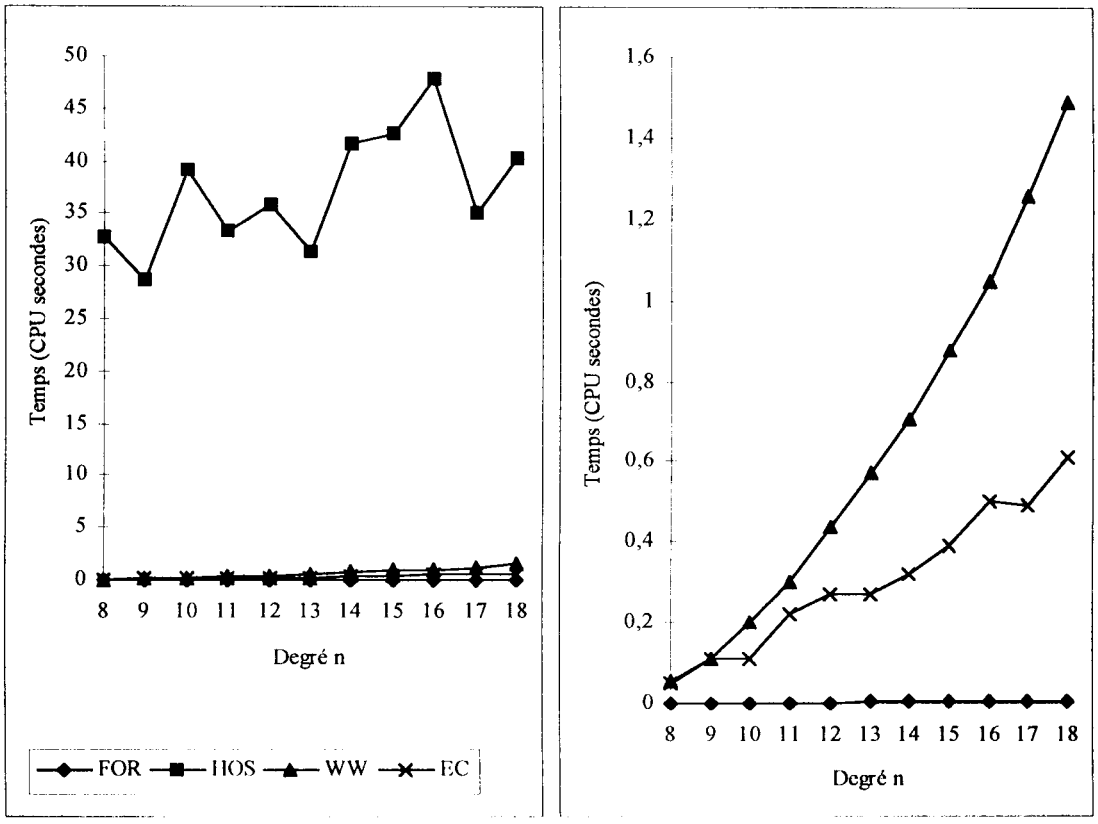

Figure III.16.a

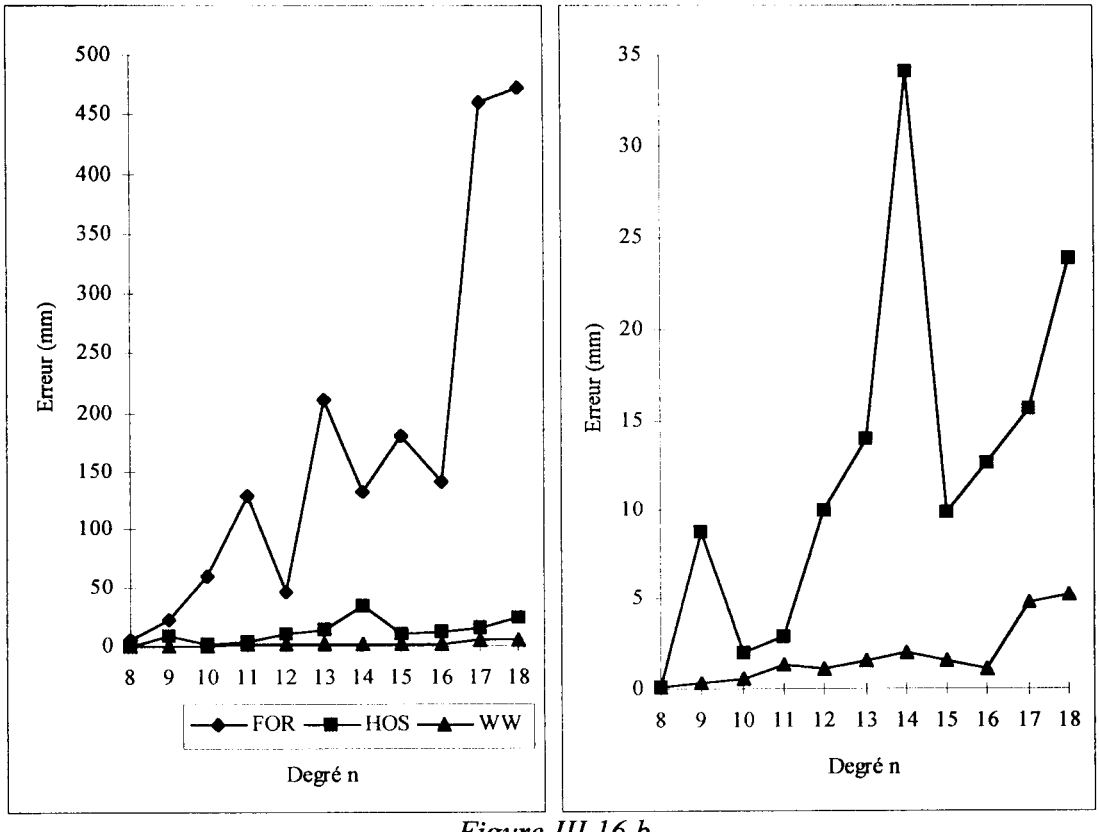

Figure III.16.b

Test 4: n → n-1 (Cf. Tableau 9 Annexe E)

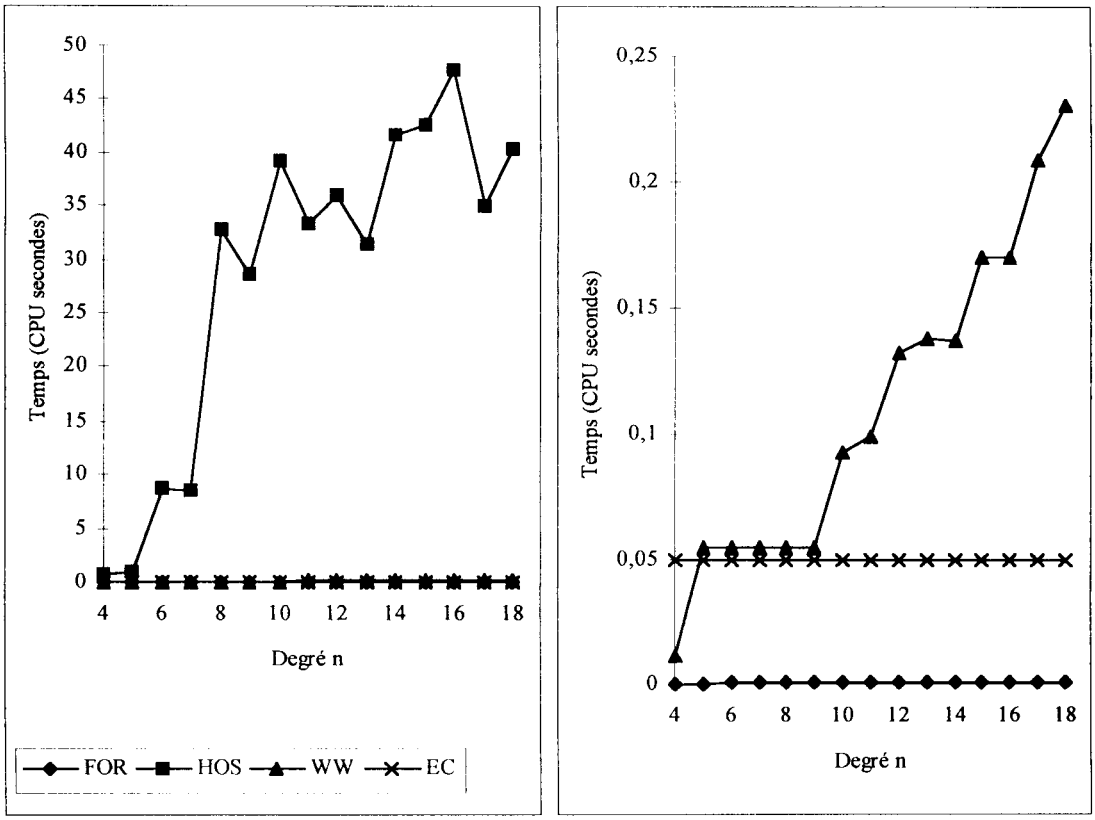

Figure III.17.a

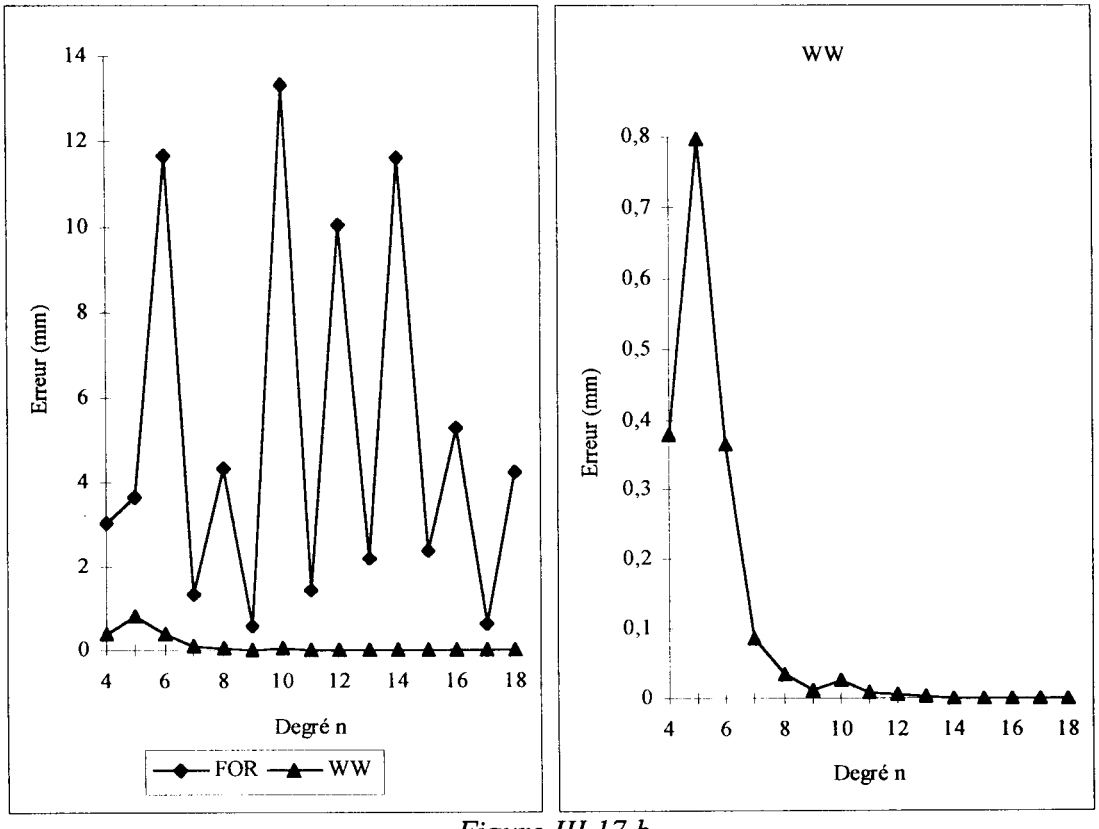

Figure III.17.b

Test 5: n → n-2 (Cf. Tableau 10 Annexe E)

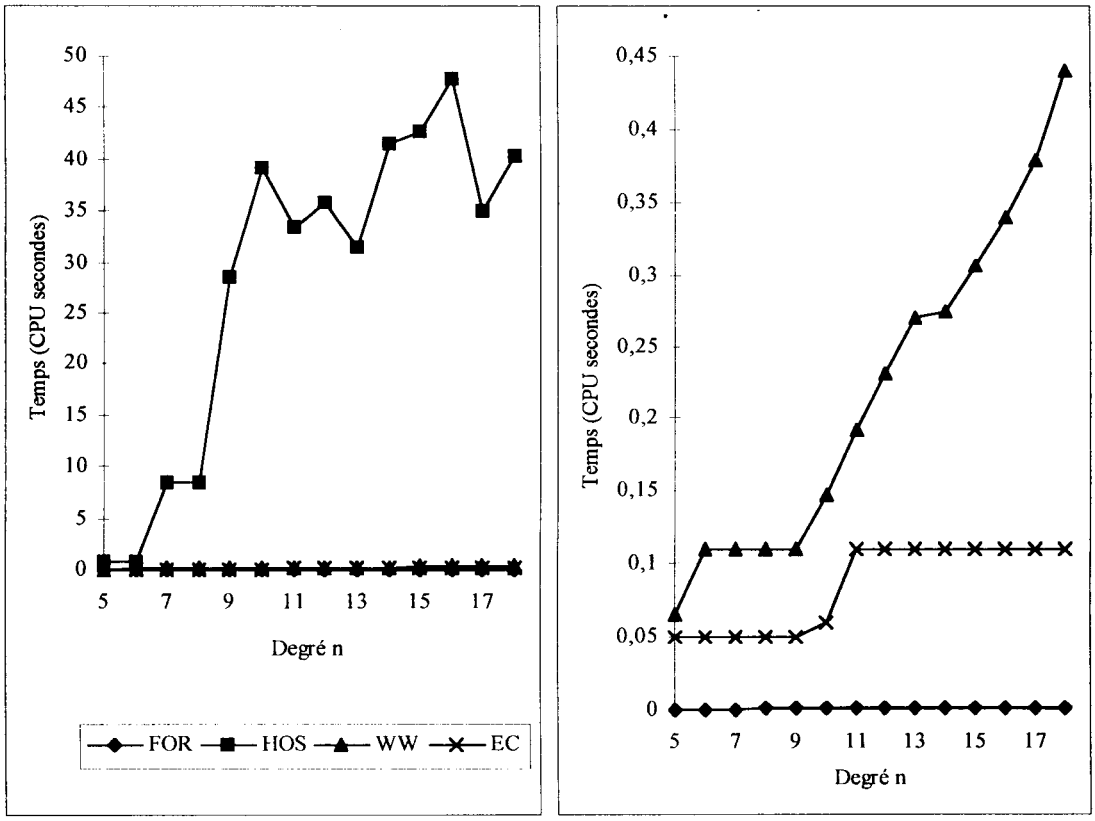

Figure III.18.a

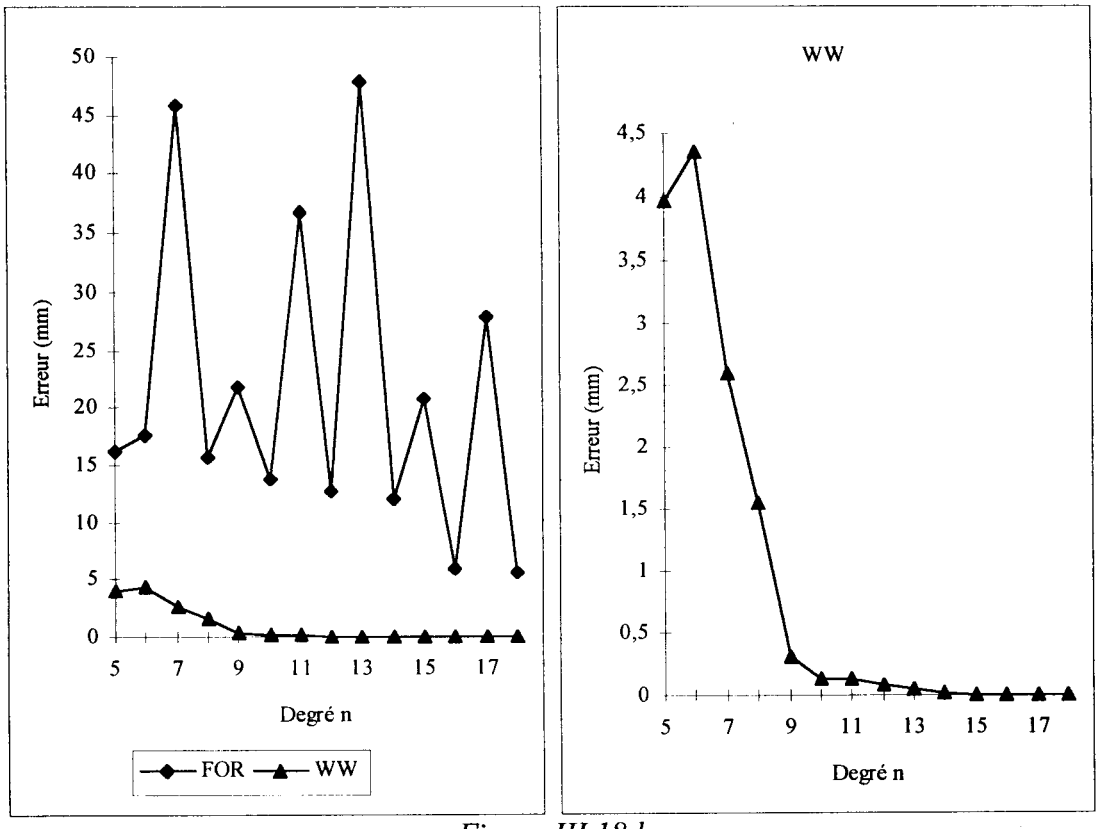

Figure III.18.b

Test 6: n → n-3 (Cf. Tableau 11 Annexe E)

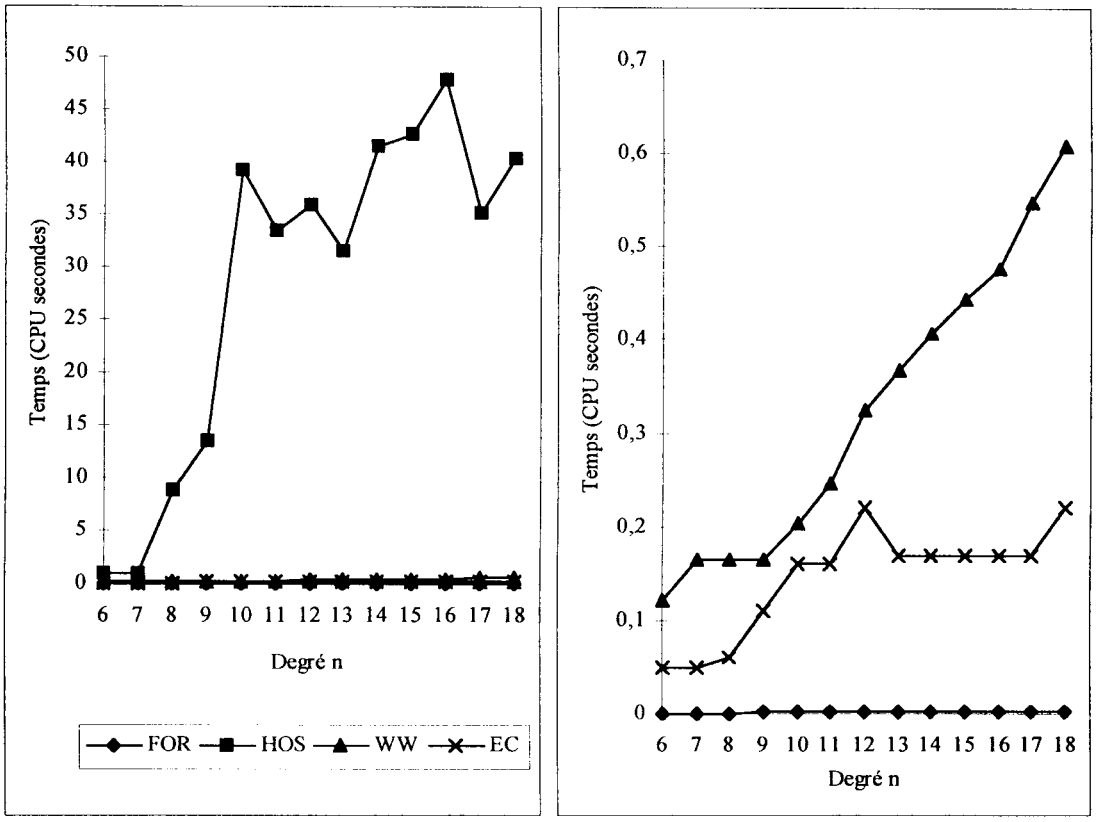

Figure III.19.a

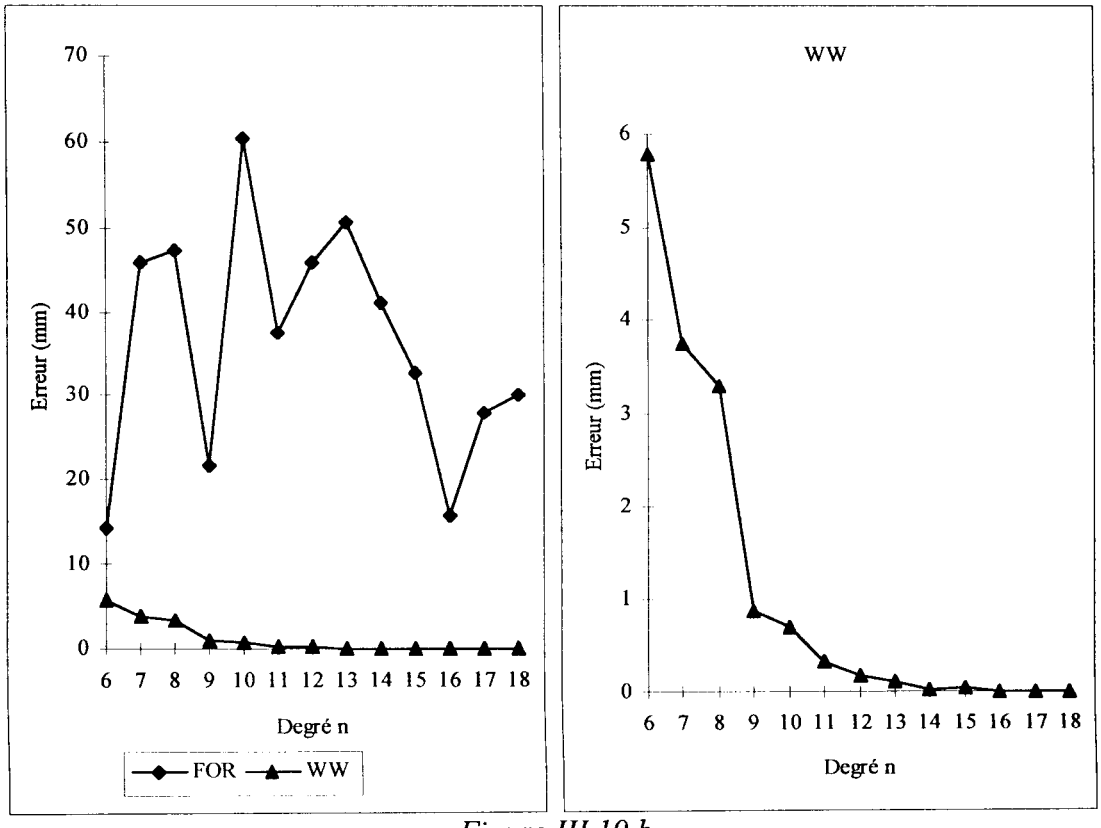

Figure III.19.b

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. A l'opposé, la méthode de Forrest respecte la continuité maximale et donne évidement l'erreur d'approximation la plus élevée qui est souvent indésirable. L'erreur est encore plus grande dans le cas de l'algorithme de Bensalah. Ces trois méthodes ne répandant pas au critère de la continuité géométrique seront donc rejetées.

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 courbes obtenues	Très Elevé	Elevé	Réduit	Réduit	Réduit
Temps de calcul	Très court	Très court	Court	Court	Très élevé
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=2k+1.)
Continuité	Continuité de position et de tangence.	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 Tchebycheff.	Nécessite un algorithme de passage du modèle voulu à la base Tchebycheff.	Possible en modifiant la base Bernstein, par la base du modèle voulu
Améliorations possibles	On peut prévoir une amélioration de l'algorithme de façon à réduire le nombre de courbes	L'amélioration de cet algorithme nous conduit à l'algorithme proposé	L'amélioration 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

Tableau 3.Tableau récapitulatif de la comparaison entre les méthodes

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 courbes réduites élevé. Le passage aux surfaces est facile à implémenter.

L'inconvénient majeur de la méthode de Watkins et Worsey est la continuité géométrique qui n'est respectée qu'à l'ordre 0. Cette méthode est basée sur un développement analytique 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, la méthode n'est opérationnelle que pour les réductions de courbes du degré 8 au degré 7.

Bien que la méthode de Hoschek utilise une méthode de résolution de système 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 de données entre systèmes. Elle utilise une paramétrisation intrinsèque et assure la continuité désirée. Son temps de calcul est très élevé, mais on se propose d'apporter une contribution pour le réduire.

III.2.4. Méthode proposée

L'application de la méthode d'approximation au sens des moindres carrés utilisée dans la méthode de Hoschek pose un problème de résolution de système d'équations non linéaire dû à la nature analytique des conditions de continuité géométrique.

L'idée essentielle de la méthode proposée est d'appliquer la méthode d'approximation au sens des moindres carrés et d'éviter la résolution de système non linéaire.

Pour cela nous travaillons à variables séparées; C'est à dire que les pôles influant sur la continuité géométrique désirée $C^{\alpha-1}$ seront déterminés par une autre méthode. L'application de l'approximation des moindres carrés aux autres pôles donne un système d'équation linéaire à résoudre.

III.2.4.1. Méthode d'approximation

D'une manière générale, dans le cas d'approximation d'un ensemble de points donnés, on se propose de construire une courbe de type et de degré choisis a priori, et passant au plus près des points M_j donnés. Cela permet par exemple de lisser des points expérimentaux par une courbe relativement régulière, et de faible degré malgré un nombre important de points donnés.

Le critère d'approximation classiquement retenu est celui des moindres carrés [*Léon '91*], [*Marty, Cassagnes & Marin '83*], [Sablonnière '87-2], qui consiste à minimiser la somme des carrés des distances entre les points M_j et les points associés sur la courbe. Soit $P(t_j)$ le point associé à M_j, et $d_j=P(t_j)-M_j$, la valeur à minimiser est

$$D = \sum_{j=0}^{np} \left\| \delta_j \right\|^2 = \sum_{j=0}^{np} \left(\delta_{jx}^2 + \delta_{jy}^2 + \delta_{jz}^2 \right) \qquad (88)$$

Ce problème se ramène alors à la minimisation des trois composantes de la forme

$$D_x = \sum_{j=0}^{np} \delta_{jx}^2 \qquad (89)$$

Les variables d'optimisation sont les composantes (P_{ix}, P_{iy}, P_{iz}) des pôles P_i du polygone caractéristique recherché, et le minimum de D_x est obtenu en résolvant les équations de type

$$\frac{\partial D_x}{\partial P_{ix}} = 0 \text{ pour } i=0, \ldots,n \qquad (90)$$

ce qui fournit donc un système d'équations vectorielles linéaires de dimension n+1.

Le choix des paramètres t_j se fait généralement soit par répartition uniforme ($t_j=j/np$), soit par répartition proportionnelle à la distance entre les points donnés (91).

La répartition proportionnelle est à préférer lorsque la distance entre deux points de passage successifs varie de façon importante, ceci afin de limiter les oscillations de la courbe de part et d'autre des points de passage, nous optons alors pour

$$t_0=0 \text{ et } t_{j+1} = t_j + \frac{\|M_{j+1} - M_j\|}{\sum_{i=0}^{np-1} \|M_{i+1} - M_i\|} \tag{91}$$

III.2.4.2. Algorithme de réduction

Le problème de réduction de degré dans la communication entre système de C.F.A.O se pose de la manière suivante: Réduire le degré d'une courbe P(t) de degré n en une courbe Q(t) de degré m<n en respectant une tolérance de position ε_0 imposée et une continuité $C^{\alpha-1}$ aux extrémités (α=0 veut dire pas de continuité).

Détermination des pôles influant sur la continuité géométrique

L'approximation de Forrest donne les pôles produisant une continuité géométrique maximale. Alors pour avoir une continuité d'ordre $C^{\alpha-1}$, il suffit de prendre les α premiers pôles R_i et les α derniers pôles S_i de Forrest. C'est à dire

$i=0, 1, ..., \alpha-1 \quad Q_i=R_i$ (92)

$i=n-1, n-2, ..., n-\alpha \quad Q_i=S_i$ (93)

Détermination des pôles intermédiaires

Considérons la courbe de degré n, $P(t) = \sum_{i=0}^{n} P_i . B_{i,n}(t)$ à réduire en une courbe de degré n-1, $Q(t) = \sum_{i=0}^{n-1} Q_i . B_{i,n-1}(t)$ en respectant la continuité $C^{\alpha-1}$ (α-1<(n-1)/2).

L'objectif est de minimiser la fonction δ suivante

$$\delta = \sum_{j=1}^{np} \left[P(t_j) - Q(t_j)\right]^2 = \sum_{j=1}^{np} e_j^2 \tag{94}$$

avec $e_j=P(t_j)-Q(t_j)$ et np étant le nombre de points de la courbe.

$$e_j = \sum_{i=0}^{n} P_i . B_{i,n}(t_j) - \sum_{i=0}^{n-1} Q_i . B_{i,n-1}(t_j)$$

$$\frac{\partial e_j}{\partial Q_h} = \frac{\partial}{\partial Q_h}\left[P(t_j) - Q(t_j)\right] \qquad \text{pour } h=\alpha, \alpha-1, ..., n-\alpha-1$$

$$\frac{\partial e_j}{\partial Q_h} = \frac{\partial P(t_j)}{\partial Q_h} - \frac{\partial Q(t_j)}{\partial Q_h} \text{ avec } \frac{\partial P(t_j)}{\partial Q_h} = 0$$

$$\frac{\partial e_j}{\partial Q_h} = -\frac{\partial Q(t_j)}{\partial Q_h} = -\sum_{i=0}^{n-1} \frac{\partial Q_i}{\partial Q_h} . B_{i,n-1}(t_j) \text{ avec } \frac{\partial Q_i}{\partial Q_h} = \begin{cases} 0 \text{ si } i \neq h \\ 1 \text{ si } i = h \end{cases}$$

$$\Rightarrow \frac{\partial e_j}{\partial Q_h} = -B_{h,n-1}(t_j) \tag{95}$$

$$\frac{\partial \delta}{\partial Q_h} = \sum_{j=1}^{np} \frac{\partial (e_j)^2}{\partial Q_h} = 2.\sum_{j=1}^{np} e_j . \frac{\partial e_j}{\partial Q_h}$$

Nous obtenons en intégrant (95)

$$\frac{\partial \delta}{\partial Q_h} = -2.\sum_{j=1}^{np} e_j . B_{h,n-1}(t_j) \tag{96}$$

$$\frac{\partial \delta}{\partial Q_h} = -2.\sum_{j=1}^{np}\left[\sum_{i=0}^{n} P_i . B_{i,n}(t_j) - \sum_{i=0}^{n-1} Q_i . B_{i,n-1}(t_j)\right] . B_{h,n-1}(t_j)$$

$$\frac{\partial \delta}{\partial Q_h} = -2.\sum_{j=1}^{np}\left[\sum_{i=0}^{n} P_i . B_{i,n}(t_j) . B_{h,n-1}(t_j) - \sum_{i=0}^{n-1} Q_i . B_{i,n-1}(t_j) . B_{h,n-1}(t_j)\right] \tag{97}$$

L'application de l'approximation au sens des moindres carrés revient alors à résoudre les équations de la forme

$$\frac{\partial \delta}{\partial Q_h} = 0 \quad \text{pour } h=\alpha,\ \alpha-1,\ n-\alpha-1 \tag{98}$$

$$\sum_{j=1}^{np}\left[\sum_{i=0}^{n} P_i . B_{i,n}(t_j) . B_{h,n-1}(t_j) - \sum_{i=0}^{n-1} Q_i . B_{i,n-1}(t_j) . B_{h,n-1}(t_j)\right] = 0$$

$$\sum_{j=1}^{np}\sum_{i=0}^{n} P_i . B_{i,n}(t_j) . B_{h,n-1}(t_j) = \sum_{j=1}^{np}\sum_{i=0}^{n-1} Q_i . B_{i,n-1}(t_j) . B_{h,n-1}(t_j)$$

$$\sum_{i=0}^{n} P_i . \sum_{j=1}^{np} B_{i,n}(t_j) . B_{h,n-1}(t_j) = \sum_{i=0}^{n-1} Q_i . \sum_{j=1}^{np} B_{i,n-1}(t_j) . B_{h,n-1}(t_j)$$

Ce qui donne le système final suivant à résoudre

$$\begin{aligned}&\sum_{i=0}^{n} P_i . \sum_{j=1}^{np} B_{i,n}(t_j) . B_{h,n-1}(t_j) - \sum_{i=0}^{\alpha-1} Q_i . \sum_{j=1}^{np} B_{i,n-1}(t_j) . B_{h,n-1}(t_j) - \\ &\sum_{i=n-\alpha}^{n-1} Q_i . \sum_{j=1}^{np} B_{i,n-1}(t_j) . B_{h,n-1}(t_j) = \sum_{i=\alpha}^{n-\alpha-1} Q_i . \sum_{j=1}^{np} B_{i,n-1}(t_j) . B_{h,n-1}(t_j)\end{aligned} \tag{99}$$

Il s'agit donc, d'un système d'équation d'ordre $n-2.\alpha$ de la forme: $A_{ih}.Q_h=b_i$
On note:

$$\gamma_{ih} = \sum_{j=1}^{np} B_{i,n}(t_j) . B_{h,n-1}(t_j) \tag{100}$$

$$\beta_{ih} = \sum_{j=1}^{np} B_{i,n-1}(t_j) . B_{h,n-1}(t_j) \text{ et} \tag{101}$$

$$b_h = \sum_{i=0}^{n} P_i . \gamma_{ih} - \sum_{i=0}^{\alpha-1} Q_i . \beta_{ih} - \sum_{i=n-\alpha}^{n-1} Q_i . \beta_{ih} \tag{102}$$

et le système s'écrit en fin

$$\sum_{i=\alpha}^{n-\alpha} Q_i.\beta_{ih} = b_h \qquad (103)$$

La méthode se ramène aux trois étapes suivantes

Pour i=0, 1, ..., α-1 $Q_i=R_i$
Pour i=n-1, n-2, ..., n-α $Q_i=S_i$
Pour i=α, α+1, ..., n-α-1 Résoudre le système $\sum_{i=\alpha}^{n-\alpha} Q_i.\beta_{ih} = b_h$

L'application successive de cette méthode produit une réduction d'un degré n à un autre degré m<n. Ce qui nous conduit à l'algorithme de réduction suivant

Algorithme PRO#1

Données: P_i (i=0, 1, ...,n) Polygone de la courbe de degré n
k ordre de la réduction
α facteur de continuité

Pour i=0, 1, ..., n
$W_i=P_i$
Fin pour
Pour r=1, 2, ..., k
Si α>0 alors
$Q_0=W_0$, $Q_{n-1}=W_n$
Pour i=1, 2, ..., α-1

$$Q_i = \left(1+\frac{i}{n-i}\right).W_i - \frac{i}{n-i}.Q_{i-1}$$

Fin pour
Pour i=n-1, n-2, ..., n-α+1

$$Q_{i-1} = \left(1+\frac{n-i}{i}\right).W_i - \frac{n-i}{i}.Q_i$$

Fin pour

Calculer $\gamma_{ih} = \sum_{j=1}^{np} B_{i,n}(t_j).B_{h,n-1}(t_j)$

Calculer $\beta_{ih} = \sum_{j=1}^{np} B_{i,n-1}(t_j).B_{h,n-1}(t_j)$

Calculer $b_h = \sum_{i=0}^{n} W_i.\gamma_{ih} - \sum_{i=0}^{\alpha-1} Q_i.\beta_{ih} - \sum_{i=n-\alpha}^{n-1} Q_i.\beta_{ih}$

Résoudre le système $\sum_{i=\alpha}^{n-\alpha} Q_i.\beta_{ih} = b_h$

Fin Si
Pour i=0, 1, ..., n-1
$W_i=Q_i$

Fin pour
n=n-1
Fin pour

Résultats: Q_i (i=0, 1, ..., n-k) courbe de degré n-k
Fin.

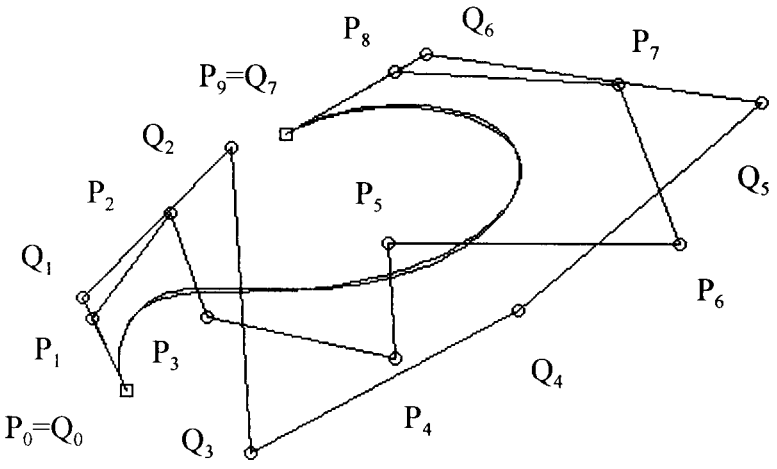

Figure III.20.a. Courbe de Bézier non rationnelle de degré 9 réduite au degré 7 avec une continuité C^2

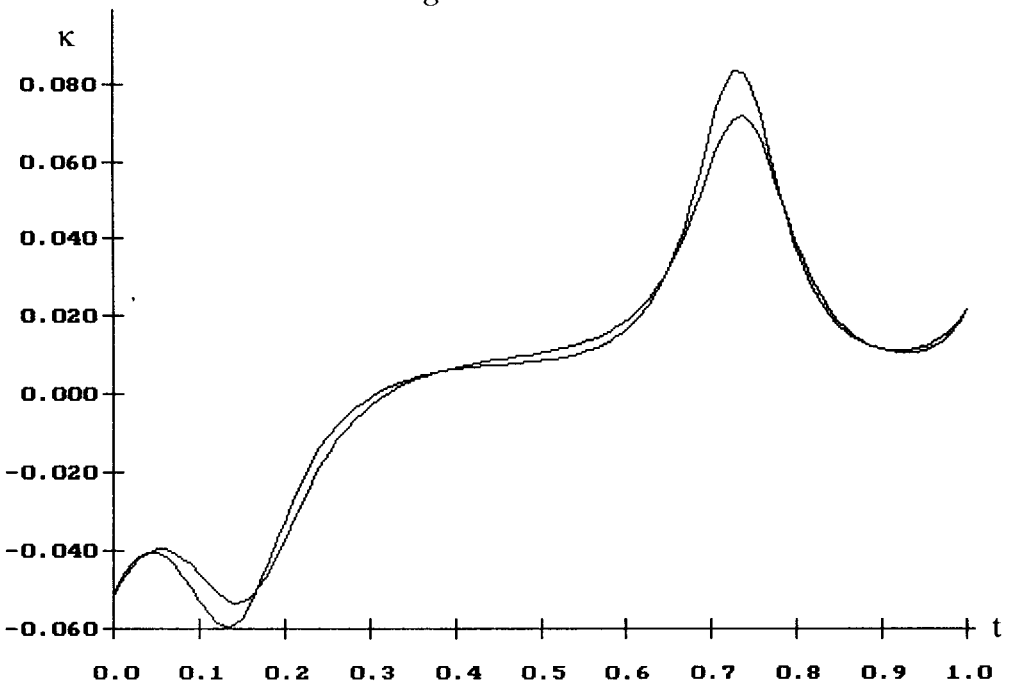

Figure III.20.b. Courbure des deux courbes précédentes

La figure III.20.a représente une courbe de Bézier non rationnelle P(t) de degré 9 réduite en une courbe Q(t) de degré 7 par l'algorithme PRO#1 en respectant une continuité C^2, les deux polygones caractéristiques représentés montrent que ces deux courbes ont une continuité C^1 aux extrémités. La figure III.20.b représente sur un même graphe la variation des courbures des deux courbes, exprimant ainsi une continuité aux extrémités de la courbe.

III.2.4.3. Algorithme de sub-réduction

Dans les méthodes précédentes aucune stratégie de choix du point de subdivision n'est utilisée, puisqu'on subdivise chaque fois que l'erreur est supérieure à la tolérance fixée au point d'abscisse curviligne t_0=0,5. La subdivision en ce point risque de donner une ou deux courbes de plus.
La stratégie proposée basée sur la subdivision de l'intervalle [0, 1] par Dichotomie consiste à chercher le point de subdivision qui donne pour la courbe réduite l'erreur maximale inférieure à la tolérance ε_0. Ce qui donne un nombre minimal de courbes de degré réduit.
Cette stratégie conduit à l'algorithme de sub-réduction PRO#2

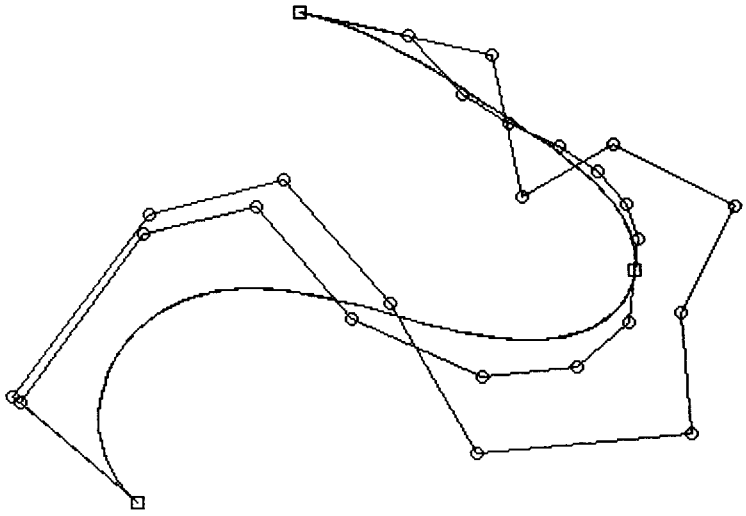

Figure III.21.a. Courbe de Bézier non rationnelle de degré 12 convertie en deux courbes de degré 8 par l'algorithme proposé

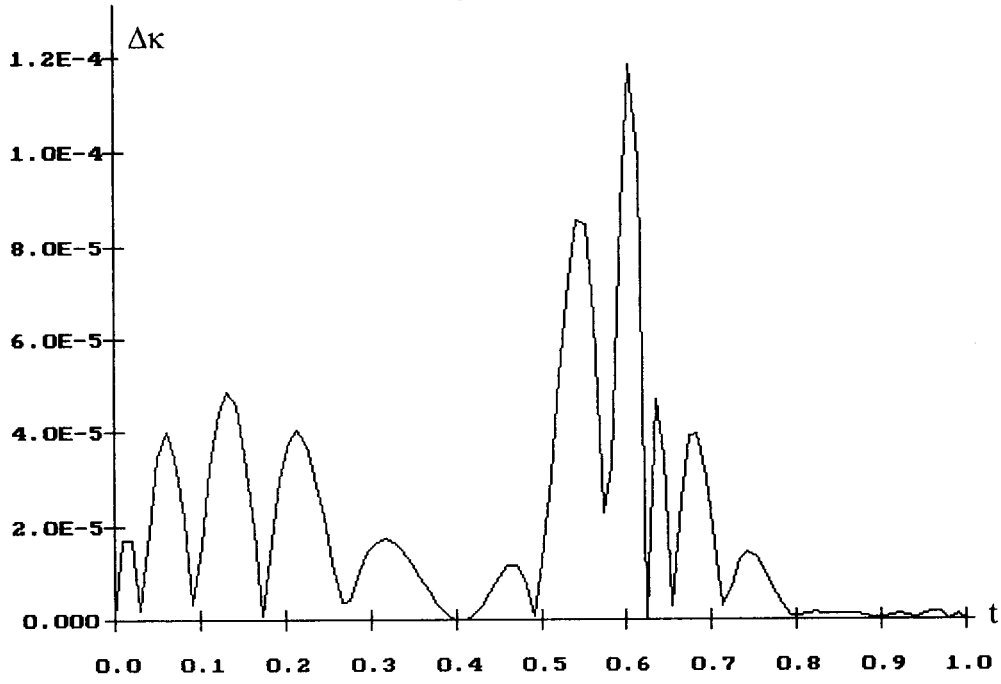

Figure III.21.b. Variation de l'erreur de courbure

La Figure III.21.a représente une courbe de Bézier non rationnelle de degré 12 réduite en deux courbes de degré 8 en respectant une continuité C^2 mise en évidence dans la figure III.21.b.

Algorithme PRO#2

Données: P_i (i=0, 1, ..., n) Pôles de la courbe de degré n
α degré de continuité à respecter
k ordre de la réduction
N nombre de points de la courbe

l=0
(*) PRO#1 (P, k, $\alpha \rightarrow$ Q)
EPS = Max$|P(t) - Q(t)|$
Si EPS >ε_0 Alors
t_1=0; t_2=1
Répéter
Répéter
$t_0=(t_1+t_2)/2$
SUB (P, $t_0 \rightarrow$ P1, P2)
PRO#1 (P1, k, $\alpha \rightarrow$ Q1)
EPS = Max$|P1(t) - Q1(t)|$
Si EPS >ε_0 Alors $t_2=t_0$
Jusqu'à ce que EPS $\leq \varepsilon_0$
PRO#1 (P2, k, $\alpha \rightarrow$ Q2)
EPS = Max$|P2(t) - Q2(t)|$
Si EPS > ε_0 Alors
$t_1=t_0$
Si $(t_2-t_1) < (1/(N-1))$ Alors
l=l+1;
Pour i=0, 1, ..., n-k
$Q_i^l = Q1_i$
Fin pour
Pour i=0, 1, ..., n
$P_i = P2_i$
Fin pour
N=N2; Aller à (*)
Fin Si
Fin Si
Jusqu'à ce que EPS $\leq \varepsilon_0$
l=l+2
Pour i=0, 1, ..., n-k
$Q_i^{l-1} = Q1_i$; $Q_i^l = Q2_i$
Fin pour
Sinon

l=l+1
Pour i=0, 1, ..., n-k
$Q_i^l = Q_i$
Fin pour
Fin Si

Résultats: Q_i^j (j=1, 2, ..., l; i=0, 1, ..., n-k) l courbes de degré n-k
Fin.

III.2.4.4. Comparaison avec les autres méthodes

L'algorithme proposé donne une approximation très proche de celle de Eck et possède les mêmes caractéristiques.

La méthode proposée comme celle de Eck a l'avantage de respecter le problème de la réduction de degré à savoir réduire le degré n d'une courbe en un degré m<n en respectant une continuité géométrique $C^{\alpha-1}$. La séparation du problème de la continuité géométrique et de l'approximation réduit le champ de l'approximation. Pour le respect de la continuité géométrique on fait appel aux polygones 1 et 2 et pour l'approximation on fait appel à celle des moindres carrés. Les résultats pratiques sont jugés intéressant puisque la stratégie de subdivision donne le nombre minimal de courbes compte tenu de la méthode.

Le temps de calcul est important mais beaucoup plus faible que celui de la méthode de Hoschek et produit une assez bonne approximation, proche de la méthode de Hoschek (Cf. Tableaux 3 à 9 Annexe E).

Pour aboutir à une conclusion finale, il est indispensable de voir l'influence de l'erreur d'approximation sur le nombre de courbes et le temps de calcul obtenus après réduction de degré pour les deux méthodes retenues (principalement pour leur respect de la continuité géométrique désirée). Pour cela, faisons subir aux algorithmes SUB-HOS et PRO#2 les tests suivants.

Test 7: Réduction du degré n au degré m=5 avec ε_0=0,01 mm
Test 8: Réduction du degré n au degré m=5 avec ε_0=0,005 mm
Test 9: Réduction du degré n au degré m=5 avec ε_0=0,001 mm

k=2 pour la méthode de Hoschek et α=3 pour la méthode proposée, ce qui donne la même continuité géométrique $C^k=C^{\alpha-1}=C^2$ qui est une maximale.

Nous remarquons d'après les résultats expérimentaux (tableau suivant) que la méthode proposée PRO#2 donne à l'unité près le même nombre de courbes que la méthode de Hoschek pour un temps de calcul beaucoup plus petit.

Test	Test 7 (ε_0=1e-5)				Test 8 (ε_0=5e-6)				Test 9 (ε_0=1e-6)			
Méthode	Sub-Hos		Pro#2		Sub-Hos		Pro#2		Sub-Hos		Pro#2	
n	t	nc	t	nc	t	nc	t	nc	t	nc	t	nc
6	29,71	4	1,43	4	29,66	4	1,27	4	29,72	4	1,87	5
7	37,07	4	1,98	5	36,31	4	1,59	5	67,07	7	2,92	7
8	49,87	4	2,58	6	96,73	7	2,63	6	107,32	8	3,46	8
9	125,45	6	2,52	6	138,58	7	2,31	6	162,63	8	4,18	8
10	153,9	7	2,91	6	177,19	8	3,19	7	189,5	9	4,62	9
11	161,48	6	3,47	7	172,52	7	3,41	8	224,64	9	5,27	10
12	229,2	6	3,68	7	231,29	6	4,29	8	355,7	10	5,61	10
13	892,21	6	4,01	7	920,77	7	4,34	8	1019,64	10	6,09	11
14	350,59	6	4,67	7	527,45	9	4,61	8	600,06	11	6,86	11
15	708,65	10	5,54	8	708,87	10	5,1	9	829,82	13	6,86	12
16	613,14	8	5,05	8	619,17	8	5,33	9	767,8	12	7,09	11
17	1031,33	10	6,7	9	1039,19	11	6,15	10	1040,4	11	9,28	13
18	1091,92	9	6,98	9	1091,81	9	6,87	10	1223,19	13	9,45	13
Moy	**421,12**	7	**3,96**	7	**445,35**	7	**3,93**	8	**509,04**	**10**	**5,66**	**10**

Tableau 4. Comparaison sur les nombres de courbes obtenues après sub-réduction

III.2.4.5. Conclusion

L'approximation proposée basée sur la méthode des moindres carrés a un comportement uniforme produisant ainsi une erreur moyenne acceptable. Du point de vue algorithmique la méthode PRO#2 ayant un temps de calcul très appréciable est d'une facilité d'implémentation remarquable sans pour cela nécessiter de grand moyens mémoire. Celui ci possède les qualités d'extensibilité aux autres modèles et d'intégration dans l'environnement des logiciels industriels. Du point de vue application à l'échange de données, la stratégie de subdivision utilisée donne un nombre de courbes réduites du même ordre que la méthode de Hoschek et permet de réduire d'un degré quelconque à un autre degré inférieur en respectant la continuité désirée. Par conséquent l'algorithme PRO#2 est la méthode qui répond à toutes les exigences du transfert de données entre systèmes (continuité géométrique et degré) en donnant le nombre de courbes minimal pour un temps de calcul moyen.

Nous pouvons dire qu'entre les méthodes présentées dans ce chapitre l'algorithme PRO#2 aura le plus d'application en C.F.A.O. Pour cela cette méthode est retenue pour notre application à l'échange de données entre systèmes et pour son extension aux autres modèles de courbes et aux surfaces.

Application à l'échange de données entre systèmes

IV.1.	**Extension de la méthode proposée**	99
IV.1.1.	Courbes de Bézier rationnelles	99
IV.1.1.1.	Algorithme de subdivision	99
IV.1.1.2.	Algorithme de réduction	100
IV.1.1.3.	Algorithme de sub-réduction	105
IV.1.2.	Surfaces de Bézier non rationnelles	108
IV.1.3.	Surfaces de Bézier rationnelles	109
IV.1.4.	Courbes B-spline non rationnelles	111
IV.1.5.	Courbes B-spline rationnelles	112
IV.2.	**Transmission de données**	113

IV. Application à l'échange de données entre systèmes

IV.1. Extension de la méthode proposée

Dans le but de l'appliquer à un échange de données entre systèmes, nous développerons dans cette partie les algorithmes de réduction obtenus par extension de la méthode proposée aux courbes de Bézier rationnelles et aux surfaces rationnelles et non rationnelles ainsi qu'aux courbes B-spline rationnelles et non rationnelles. Pour cela nous faisons appel aux algorithmes développés dans le chapitre II, pour l'insertion et la suppression de nœuds des courbes B-splines, la subdivision et le passage réversible entre les modèles Bézier et B-spline.
Pour ce faire, plusieurs possibilités d'extension sont envisageables mais nous nous limitons à l'application intégrale de l'algorithme.

IV.1.1. Courbes de Bézier rationnelles

IV.1.1.1. Algorithme de subdivision

La méthode de subdivision des courbes de Bézier rationnelles est semblable à celle des courbes non rationnelles.
Soit la courbe C(t) de degré n ayant pour pôles C_i chargés par w_i à subdiviser au point t_0 en deux courbes P(t) et Q(t) de degré n ayant respectivement les pôles P_i chargés par ξ_i et Q_i chargés par ψ_i .
suivant le § II.1.1.2. l'algorithme s'écrit

Algorithme SUB

Données: C_i (i=0, 1, ..., n) polygone de la courbe de degré n
w_i (i=0, 1, ..., n) charges des pôles
t_0 : abscisse curviligne du point de subdivision

Pour i=0, 1, ..., n
$C_i^0(t_0) = C_i$
$w_i^0(t_0) = w_i$
Fin pour
Pour k=1, 2, ..., n
Pour i=0, 1, ..., n

$$w_i^k(t_0) = (1-t_0).w_i^{k-1}(t_0) + t_0.w_{i+1}^{k-1}(t_0)$$

$$C_i^k(t_0) = (1-t_0).\frac{w_i^{k-1}(t_0)}{w_i^k(t_0)}.C_i^{k-1}(t_0) + t_0.\frac{w_{i+1}^{k-1}(t_0)}{w_i^k(t_0)}.C_{i+1}^{k-1}(t_0)$$

Fin pour
Fin pour
Pour i=0, 1, ..., n
$P_i = C_0^i(t_0)$; $Q_i = C_i^{n-i}(t_0)$
$\xi_i = w_0^i(t_0)$; $\psi_i = w_i^{n-i}(t_0)$

Fin pour

Résultats : P_i, ξ_i et Q_i, ψ_i (i=0, 1, ..., n)
Fin

IV.1.1.2. Réduction de degré

La méthode de réduction proposée se base sur l'approximation au sens des moindres carrés. Cette approximation pour les courbes rationnelles pose un problème non linéaire à résoudre indépendamment de l'ordre de continuité à respecter.
De la même manière que pour l'algorithme PRO#1, nous travaillons à variables séparées pour éviter l'approximation non linéaire. La méthode se déroule alors en deux étapes à savoir:

- détermination séparée des pôles et de leurs poids pour éviter l'approximation non linéaire due au modèle rationnel,
- détermination séparée des pôles influant sur la continuité géométrique pour éviter la résolution d'équations non linéaires générées par l'ordre de continuité à respecter,
- détermination des poids influant sur la continuité géométrique.

Ainsi pour une courbe de degré n, $P(t) = \sum_{i=0}^{n} P_i . G_{i,n}(t)$ ayant pour charges w_i à réduire en une courbe de degré n-1, $Q(t) = \sum_{i=0}^{n-1} Q_i . G_{i,n-1}(t)$ ayant pour charges ψ_i en respectant la continuité $C^{\alpha-1}$, $G_{i,n}(t)$ étant les fonctions de Bernstein rationnelles avec

$$G_{i,n}(t) = \frac{B_{i,n}(t).w_i}{\sum_{j=0}^{n} B_{j,n}(t).w_j},$$

l'application de notre méthode permet alors :

a) la détermination des pôles et des charges influant sur la continuité

En effet, de la même manière que pour le cas non rationnel, pour avoir une continuité géométrique $C^{\alpha-1}$, il suffit de prendre les α premiers pôles R_i de (19.a) ainsi que les α premières charges α_i de (20.a) et les α derniers pôles S_i de (19.b) ainsi que les α dernières charges β_i de (20.b). Ce qui donne:

$Q_0=P_0$ et $\psi_0= w_0$

i=1, 2, ..., α-1 $\quad \psi_i = \frac{n}{n-i}.w_i - \frac{i}{n-i}.\psi_{i-1}$ (104.a)

$$Q_i = \frac{n}{n-i}.\frac{w_i}{\psi_i}.P_i - \frac{\psi_{i-1}}{\psi_i}.\frac{i}{n-i}.Q_{i-1} \quad (105.a)$$

$Q_{n-1}=P_n$ et $\psi_{n-1}= w_n$

$$i=n, n-2, ..., n-\alpha+1 \quad \psi_{i-1} = \frac{n}{i}.w_i - \frac{n-i}{i}.\psi_i \tag{104.b}$$

$$Q_{i-1} = \frac{n}{i}.\frac{w_i}{\psi_{i-1}}.P_i - \frac{\psi_i}{\psi_{i-1}}.\frac{n-i}{i}.Q_i \tag{105.b}$$

b) la détermination des charges restantes

Les courbes rationnelles étant une généralisation des courbes non rationnelles obtenue par transformation projective, on peut considérer les deux fonctions suivantes:

$$w(t) = \sum_{i=0}^{n} w_i.B_{i,n}(t) \text{ et } \psi(t) = \sum_{i=0}^{n-1} \psi_i.B_{i,n-1}(t) \tag{106}$$

$\psi(t)$ étant une approximation de degré n-1 de la fonction scalaire w(t) de degré n.
Dans ce cas l'approximation des moindres carrés revient à minimiser la fonction δ suivante:

$$\delta = \sum_{j=1}^{np}\left[w(t_j) - \psi(t_j)\right]^2 = \sum_{j=1}^{np} e_j^2 \tag{107}$$

avec $e_j=w(t_j)-\psi(t_j)$ et np étant le nombre de points de la courbe.

$$e_j = \sum_{i=0}^{n} w_i.B_{i,n}(t_j) - \sum_{i=0}^{n-1} \psi_i.B_{i,n-1}(t_j)$$

$$\frac{\partial e_j}{\partial \psi_h} = \frac{\partial}{\partial \psi_h}\left[w(t_j) - \psi(t_j)\right] \qquad \text{pour } h=\alpha, \alpha-1, ..., n-\alpha-1$$

$$\frac{\partial e_j}{\partial \psi_h} = \frac{\partial w(t_j)}{\partial \psi_h} - \frac{\partial \psi(t_j)}{\partial \psi_h} \text{ avec } \frac{\partial w(t_j)}{\partial \psi_h} = 0$$

$$\frac{\partial e_j}{\partial \psi_h} = -\frac{\partial \psi(t_j)}{\partial \psi_h} = -\sum_{i=0}^{n-1} \frac{\partial \psi_i}{\partial \psi_h}.B_{i,n-1}(t_j) \text{ avec } \frac{\partial \psi_i}{\partial \psi_h} = \begin{cases} 0 \text{ si } i \neq h \\ 1 \text{ si } i = h \end{cases}$$

$$\Rightarrow \frac{\partial e_j}{\partial \psi_h} = -B_{h,n-1}(t_j) \tag{108}$$

$$\frac{\partial \delta}{\partial \psi_h} = \sum_{j=1}^{np} \frac{\partial (e_j)^2}{\partial \psi_h} = 2.\sum_{j=1}^{np} e_j.\frac{\partial e_j}{\partial \psi_h}$$

Ce qui donne un système d'équations d'ordre n-2.α de la forme: $A_{ih}.\psi_h=b_i$
On note:

$$\gamma_{ih} = \sum_{j=1}^{np} B_{i,n}(t_j).B_{h,n-1}(t_j) \tag{109}$$

$$\beta_{ih} = \sum_{j=1}^{np} B_{i,n-1}(t_j).B_{h,n-1}(t_j) \text{ et} \tag{110}$$

$$b_h = \sum_{i=0}^{n} w_i \cdot \gamma_{ih} - \sum_{i=0}^{\alpha-1} \psi_i \cdot \beta_{ih} - \sum_{i=n-\alpha}^{n-1} \psi_i \cdot \beta_{ih} \qquad (111)$$

et le système s'écrit alors:

$$\sum_{i=\alpha}^{n-\alpha} \psi_i \cdot \beta_{ih} = b_h \qquad (112)$$

c) la détermination des pôles intermédiaires

L'objectif est de minimiser la fonction δ définie par

$$\delta = \sum_{j=1}^{np} \left[P(t_j) - Q(t_j) \right]^2 = \sum_{j=1}^{np} e_j^2 \qquad (113)$$

avec $e_j=P(t_j)-Q(t_j)$ et np le nombre de points de la courbe.

$$e_j = \sum_{i=0}^{n} P_i \cdot G_{i,n}(t_j) - \sum_{i=0}^{n-1} Q_i \cdot G_{i,n-1}(t_j)$$

sachant que : $\dfrac{\partial e_j}{\partial Q_h} = \dfrac{\partial}{\partial Q_h}\left[P(t_j) - Q(t_j) \right]$ pour h=α, α-1, ..., n-α-1

$$\frac{\partial e_j}{\partial Q_h} = \frac{\partial P(t_j)}{\partial Q_h} - \frac{\partial Q(t_j)}{\partial Q_h} \text{ avec } \frac{\partial P(t_j)}{\partial Q_h} = 0$$

$$\frac{\partial e_j}{\partial Q_h} = -\frac{\partial Q(t_j)}{\partial Q_h} = -\sum_{i=0}^{n-1} \frac{\partial Q_i}{\partial Q_h} \cdot G_{i,n-1}(t_j)$$

avec $\dfrac{\partial Q_i}{\partial Q_h} = \begin{cases} 0 \text{ si } i \neq h \\ 1 \text{ si } i = h \end{cases}$, ce qui donne

$$\Rightarrow \frac{\partial e_j}{\partial Q_h} = -G_{h,n-1}(t_j) \qquad (114)$$

$$\frac{\partial \delta}{\partial Q_h} = \sum_{j=1}^{np} \frac{\partial (e_j)^2}{\partial Q_h} = 2 \cdot \sum_{j=1}^{np} e_j \cdot \frac{\partial e_j}{\partial Q_h}$$

en y insérant (114) , nous obtenons

$$\frac{\partial \delta}{\partial Q_h} = -2 \cdot \sum_{j=1}^{np} e_j \cdot G_{h,n-1}(t_j) \qquad (115)$$

Ce qui donne le système final suivant à résoudre:

$$\sum_{i=0}^{n} P_i \cdot \sum_{j=1}^{np} G_{i,n}(t_j) \cdot G_{h,n-1}(t_j) - \sum_{i=0}^{\alpha-1} Q_i \cdot \sum_{j=1}^{np} G_{i,n-1}(t_j) \cdot G_{h,n-1}(t_j) - \sum_{i=n-\alpha}^{n-1} Q_i \cdot \sum_{j=1}^{np} G_{i,n-1}(t_j) \cdot G_{h,n-1}(t_j) = \sum_{i=\alpha}^{n-\alpha-1} Q_i \cdot \sum_{j=1}^{np} G_{i,n-1}(t_j) \cdot G_{h,n-1}(t_j) \qquad (116)$$

Ce système d'équations d'ordre n-2.α est de la forme: $U_{ih} \cdot Q_h = v_i$
Notons:

$$\rho_{ih} = \sum_{j=1}^{np} G_{i,n}(t_j).G_{h,n-1}(t_j) \tag{117}$$

$$\sigma_{ih} = \sum_{j=1}^{np} G_{i,n-1}(t_j).G_{h,n-1}(t_j) \text{ et} \tag{118}$$

$$v_h = \sum_{i=0}^{n} P_i.\rho_{ih} - \sum_{i=0}^{\alpha-1} Q_i.\sigma_{ih} - \sum_{i=n-\alpha}^{n-1} Q_i.\sigma_{ih} \tag{119}$$

et le système (116) s'écrit alors:

$$\sum_{i=\alpha}^{n-\alpha} Q_i.\sigma_{ih} = v_h \tag{120}$$

La méthode se ramène alors aux étapes suivantes:

$Q_0=P_0$ et $\psi_0= w_0$

i=1, 2, ..., α-1 $\psi_i = \frac{n}{n-i}.w_i - \frac{i}{n-i}.\psi_{i-1}$

$$Q_i = \frac{n}{n-i}.\frac{w_i}{\psi_i}.P_i - \frac{\psi_{i-1}}{\psi_i}.\frac{i}{n-i}.Q_{i-1}$$

avec $Q_{n-1}=P_n$ et $\psi_{n-1}= w_n$

i=n-1, n-2, ..., n-α+1 $\psi_{i-1} = \frac{n}{i}.w_i - \frac{n-i}{i}.\psi_i$

$$Q_{i-1} = \frac{n}{i}.\frac{w_i}{\psi_{i-1}}.P_i - \frac{\psi_i}{\psi_{i-1}}.\frac{n-i}{i}.Q_i$$

Pour i=α, α+1, ..., n-α-1 Résoudre le système $\sum_{i=\alpha}^{n-\alpha} \psi_i.\beta_{ih} = b_h$

Pour i=α, α+1, ..., n-α-1 Résoudre le système $\sum_{i=\alpha}^{n-\alpha} Q_i.\sigma_{ih} = v_h$

L'application successive de cette méthode produit une réduction d'un degré n à un autre degré m<n. Ce qui nous conduit à l'algorithme de résolution suivant:

Algorithme PRO#3

Données: P_i (i=0, 1, ..., n) Polygone de la courbe de degré n
w_i (i=0, 1, ..., n) charges des pôles
k ordre de la réduction
α facteur de continuité

Pour i=0, 1, ..., n
$R_i=P_i$, $\xi_i=w_i$
Fin pour
Pour r=1, 2, ..., k
Si α>0 alors
$Q_0=R_0$

$\psi_0=\xi_0$
$Q_{n-1}=R_n$
$\psi_{n-1}=\xi_n$
Pour i=1, 2, ..., α-1

$$\psi_i = \frac{n}{n-i}.\xi_i - \frac{i}{n-i}.\psi_{i-1}$$

$$Q_i = \frac{n}{n-i}.\frac{\xi_i}{\psi_i}.R_i - \frac{\psi_{i-1}}{\psi_i}.\frac{i}{n-i}.Q_{i-1}$$

Fin pour
Pour i=n-1, n-2, ..., n-α+1

$$\psi_{i-1} = \frac{n}{i}.\xi_i - \frac{n-i}{i}.\psi_i$$

$$Q_{i-1} = \frac{n}{i}.\frac{\xi_i}{\psi_{i-1}}.R_i - \frac{\psi_i}{\psi_{i-1}}.\frac{n-i}{i}.Q_i$$

Fin pour

Calculer $\gamma_{ih} = \sum_{j=1}^{np} B_{i,n}(t_j).B_{h,n-1}(t_j)$

Calculer $\beta_{ih} = \sum_{j=1}^{np} B_{i,n-1}(t_j).B_{h,n-1}(t_j)$

Calculer $b_h = \sum_{i=0}^{n} \xi_i.\gamma_{ih} - \sum_{i=0}^{\alpha-1} \psi_i.\beta_{ih} - \sum_{i=n-\alpha}^{n-1} \psi_i.\beta_{ih}$

Résoudre le système $\sum_{i=\alpha}^{n-\alpha} \psi_i.\beta_{ih} = b_h$

Calculer $\rho_{ih} = \sum_{j=1}^{np} G_{i,n}(t_j).G_{h,n-1}(t_j)$

Calculer $\sigma_{ih} = \sum_{j=1}^{np} G_{i,n-1}(t_j).G_{h,n-1}(t_j)$

Calculer $v_h = \sum_{i=0}^{n} P_i.\rho_{ih} - \sum_{i=0}^{\alpha-1} Q_i.\sigma_{ih} - \sum_{i=n-\alpha}^{n-1} Q_i.\sigma_{ih}$

Résoudre le système $\sum_{i=\alpha}^{n-\alpha} Q_i.\sigma_{ih} = v_h$

Fin Si
Pour i=0, 1, ..., n-1
$R_i=Q_i$
$\xi_i=\psi_i$
Fin pour
n=n-1
Fin pour

Résultats: Q_i (i=0, 1, ..., n-k) courbe de degré n-k
ψ_i (i=0, 1, ..., n-k) charges des pôles Q_i
Fin.

La mise en œuvre de cet algorithme nous permet de l'appliquer sur la courbes P(t) de degré 8 que nous réduisons au degré 6 en respectant une continuité géométrique C^2 [Figure IV.1], la continuité de la tangente aux extrémités est exprimée par la visualisation des deux polygones caractéristiques.

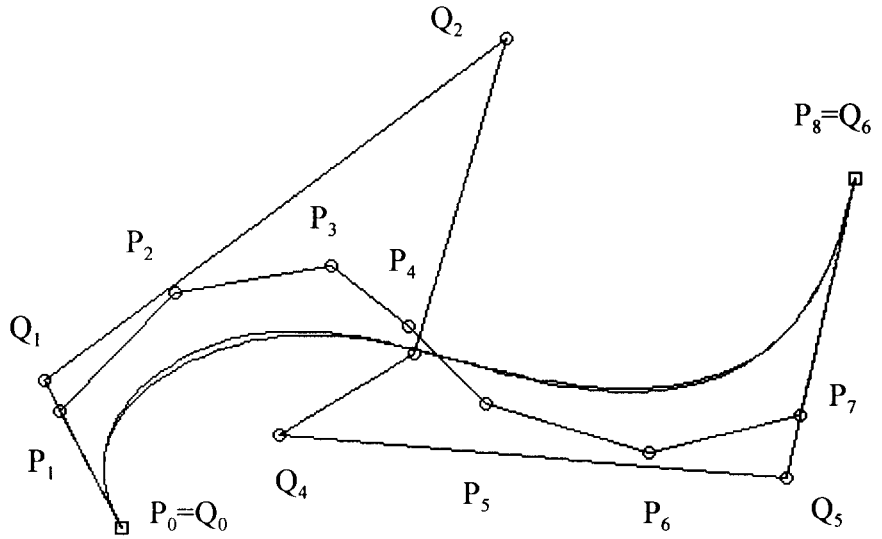

Figure IV.1. Courbe de Bézier rationnelle de degré 8 réduite au degré 6 par l'algorithme PRO#3 avec w={1,1.2,0.8,1,0.5,1,0.8,1.2,1}, w'={1,1.27,0.41,1.22,0.41,1.27,1}

IV.1.1.3. Algorithme de sub-réduction

L'algorithme de sub-réduction développé pour les courbes rationnelles est similaire à l'algorithme PRO#2 [Cf. §III.2.2.3]:

Algorithme PRO#4

Données: P_i (i=0, 1, ..., n) Pôles de la courbe de degré n
w_i (i=0, 1, ..., n) Les charges des pôles
α degré de continuité à respecter
k ordre de la réduction
N nombre de points de la courbe

l=0
(*) PRO#3 (P, w, k, $\alpha \rightarrow$ Q, ψ)
EPS = Max|P(t) – Q(t)|
Si EPS >ε_0 Alors
    t_1=0
    t_2=1
    Répéter
        Répéter
            $t_0=(t_1+t_2)/2$
            SUB (P, w, t_0 $\rightarrow$ P1, w1, P2, w2)
            PRO#3 (P1, w1, k, $\alpha \rightarrow$ Q1, ψ1)

$EPS = Max|P1(t) - Q1(t)|$

Si EPS $>\varepsilon_0$ Alors $t_2=t_0$

Jusqu'à ce que EPS $\leq \varepsilon_0$

PRO#3 (P2, w, k, $\alpha \rightarrow$ Q2, ψ2)

$EPS = Max|P2(t) - Q2(t)|$

Si EPS $> \varepsilon_0$ Alors

$t_1=t_0$

Si $(t_2-t_1) < (1/(N-1))$ Alors

l=l+1;

Pour i=0, 1, ..., n-k

$Q_i^l = Q1_i$

$\psi_i^l = \psi 1_i$

Fin pour

Pour i=0, 1, ..., n

$P_i = P2_i$

$w_i = w2_i$

Fin pour

N=N2, Aller à **(*)**

Fin Si

Fin Si

Jusqu'à ce que EPS $\leq \varepsilon_0$

l=l+2

Pour i=0, 1, ..., n-k

$Q_i^{l-1} = Q1_i$;

$\psi_i^{l-1} = \psi 1_i$

$Q_i^l = Q2_i$;

$\psi_i^l = \psi 2_i$

Fin pour

Sinon

l=l+1

Pour i=0, 1, ..., n-k

$Q_i^l = Q_i$, $\psi_i^l = \psi_i$

Fin pour

Fin Si

Résultats: Q_i^j (j=1, 2, ..., l; i=0, 1, ..., n-k) l courbes de degré n-k

ψ_i^j (j=1, 2, ..., l; i=0, 1, ..., n-k) les charges respectives des pôles

Fin.

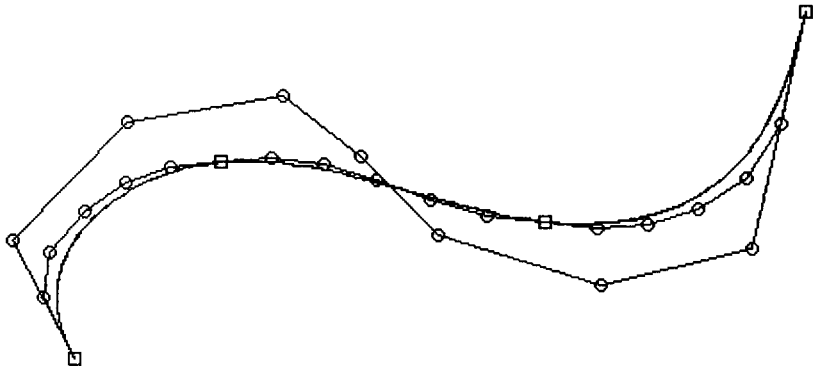

Figure IV.2.a Courbe de Bézier rationnelle de degré 8 réduite en 3 courbe de degré 6 pour une continuité d'ordre3 et une tolérance de 0.01 mm

La figure IV.2.a ci dessus représente une courbe de Bézier rationnelle de degré 8 réduite en 3 courbes de degré 6 en respectant une continuité géométrique C^3 obtenue par l'application de cet algorithme, alors que les figures IV.2.b et IV.2.c mettent en évidence le respect de la tolérance de position ε_0=0.01 ainsi que la continuité de position aux extrémités des courbes, et respectivement l'erreur de la courbure et la continuité de courbure.

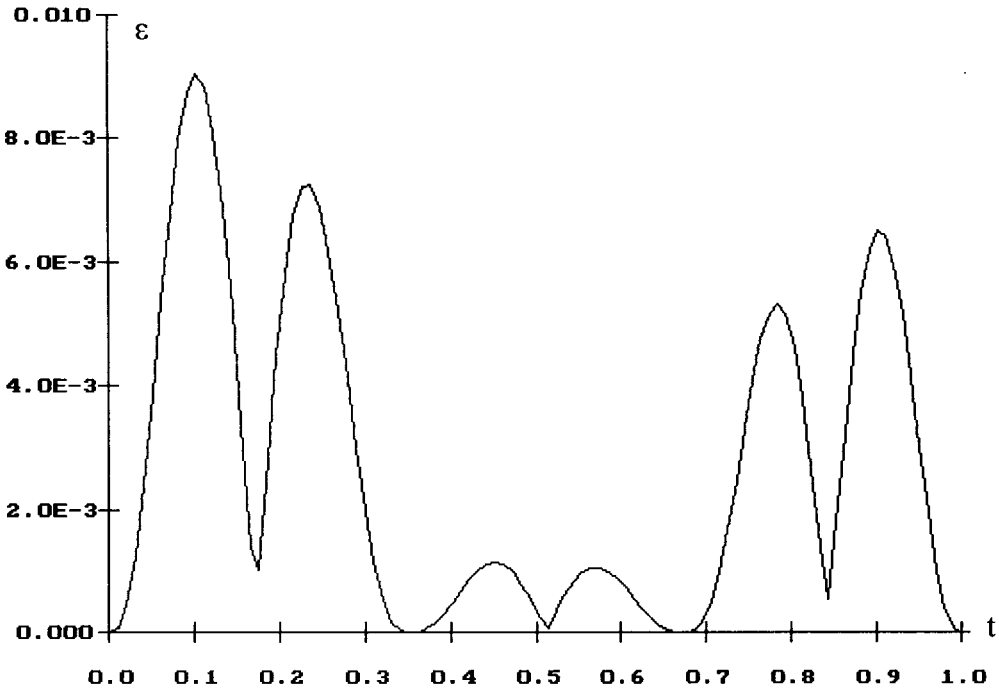

Figure IV.2.b Erreur de position pour la réduction précédente

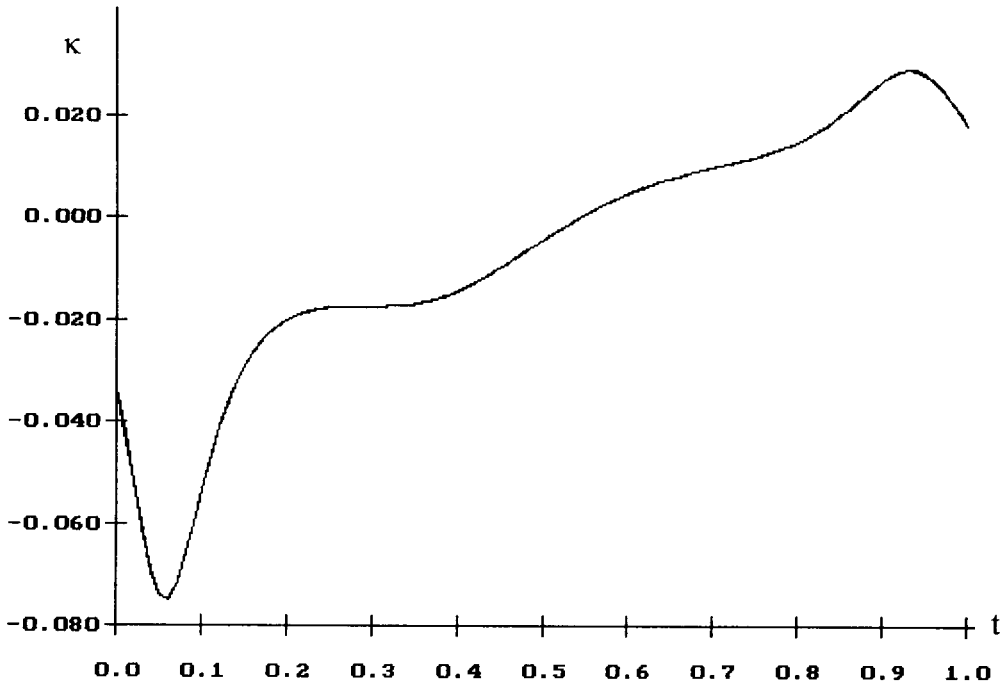

Figure IV.2.c Variation des courbures pour la réduction précédente

IV.1.2. Surfaces de Bézier non rationnelles

La réduction de degré d'une surface de Bézier non rationnelle S(u, v) de degré (m, n) définie par le carreau [S_{00}, S_{10}, ..., S_{m0}, S_{01}, S_{11}, ..., S_{mn}] suivant le paramètre u ou v revient à appliquer pour chaque rangée de pôles suivant ce paramètre (u ou v) l'algorithme de réduction de degré des courbes non rationnelles PRO#1. Ce qui donne l'algorithme PRO#5 suivant pour la réduction du degré (m, n) au degré ($m-k_1$, $n-k_2$) avec une continuité α_1 suivant la direction u et α_2 suivant v.

Algorithme PRO#5
Données: S_{ij} (i=0, 1, ..., m ; j=0, 1, ..., n) Pôles du réseau caractéristique de la surface S(m, n)

k_1, k_2 ordres de la réduction
α_1 facteur de continuité suivant u
α_2 facteur de continuité v

Pour j=1, 2, ...,n
  Pour i=0, 1, ..., m
    $P_i=S_{ij}$
  Fin pour
  PRO#1(P, k_1, α_1 → Q)
  Pour i=0, 1, ..., $m-k_1$
    $C_{ij}=Q_i$
  Fin pour
Fin pour
Pour i=1, 2, ..., $m-k_1$
  Pour j=0, 1, ..., n

$P_i=C_{ij}$
Fin pour
PRO#1(P, k_2, $\alpha_2 \rightarrow$ Q)
Pour j=0, 1, ..., n-k_2
$R_{ij}=Q_j$
Fin pour
Fin pour

Résultats: R_{ij} (i=0, 1, ..., m-k_1; j=0, 1, ..., n-k_2) Réseau caractéristique R(m-k_1, n-k_2)
Fin.

Le résultat obtenu en appliquant cet algorithme est illustré par la figure IV.3 qui représente une surface de Bézier non rationnelle de degré (8, 8) dessinée en trait continu réduite au degré (6,6) désignée en trait interrompu en respectant une continuité géométrique d'ordre 2 dans les deux directions u et v. L'approximation obtenue est comparable à la méthode de Hoschek appliquée aux surfaces, et une évaluation du temps de calcul pour quelques exemples pratiques, a montré une réduction significative.

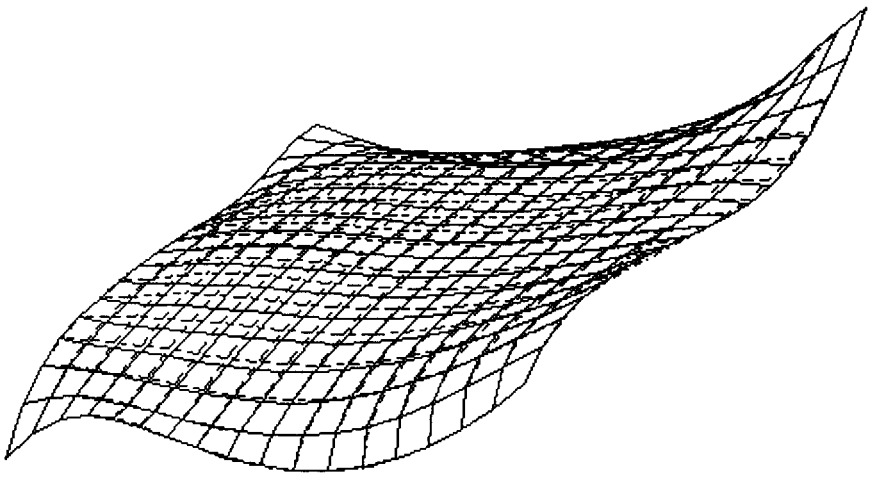

Figure IV.3. Surface de Bézier non rationnelle de degré (8,8) réduite au degré (6,6).

IV.1.3. Surfaces de Bézier rationnelles

La réduction de degré d'une surface de Bézier rationnelle S(u, v) de degré (m, n) définie par le carreau [S_{00}, S_{10}, ..., S_{m0}, S_{01}, S_{11}, ..., S_{mn}] chargée par [w_{00}, w_{10}, ..., w_{m0}, w_{01}, w_{11}, ..., w_{mn}] suivant le paramètre u ou v revient à appliquer pour chaque rangée de pôles suivant ce paramètre (u ou v) l'algorithme de réduction de degré des courbes rationnelles PRO#3. Ce qui donne l'algorithme suivant.

Algorithme PRO#6
Données: S_{ij} (i=0, 1, ..., m; j=0, 1, ..., n) Pôles du réseau caractéristique de la surface S(m, n)

w_{ij} (i=0, 1, ..., m; j=0, 1, ..., n) charges des pôles
k_1, k_2 ordres de la réduction
α_1, α_2 facteurs de continuité

Pour j=1, 2, ...,n
  Pour i=0, 1, ..., m
    $P_i=S_{ij}$
    $\xi_i=w_{ij}$
  Fin pour
  PRO#3(P, ξ, k_1, $\alpha_1 \rightarrow$ Q, ψ)
  Pour i=0, 1, ..., m-k_1
    $C_{ij}=Q_i$
    $\gamma_{ij}=\psi_{ij}$
  Fin pour
Fin pour
Pour i=1, 2, ..., m-k_1
  Pour j=0, 1, ..., n
    $P_i=C_{ij}$
    $\xi_j=\gamma_{ij}$
  Fin pour
  PRO#3(P, ξ, k_2, $\alpha_2 \rightarrow$ Q, ψ)
  Pour j=0, 1, ..., n-k_2
    $R_{ij}=Q_j$
    $\rho_{ij}=\psi_{ij}$
  Fin pour
Fin pour

Résultats: R_{ij} (i=0, 1, ..., m-k_1; j=0, 1, ..., n-k_2) Réseau caractéristique R(m-k_1, n-k_2)
  ρ_{ij} (i=0, 1, ..., m-k_1; j=0, 1, ..., n-k_2) charges des pôles
Fin.

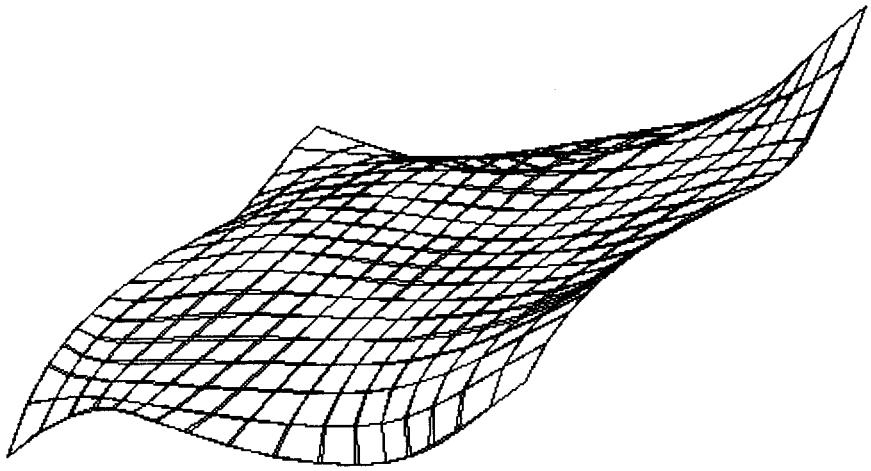

Figure IV.4. Surface de Bézier rationnelle de degré (8,8) réduite au degré (7,7).

Le résultat obtenu en appliquant cet algorithme est illustré par la figure IV.4 qui représente une surface de Bézier non rationnelle de degré (8, 8) réduite au degré (7, 7) en respectant une continuité géométrique d'ordre 2 dans les deux directions u et v.

IV.1.4. Courbes B-spline non rationnelles

Pour la réduction des courbes B-spline, il existe aussi d'autres méthodes spécifiques à ce modèle [*Dannenberg & Nowacki '85*], [*Patrikalakis '89*], [*Hoschek & Schneider '90*], [*Eck & Hadenfeld '94*], [*Piegl & Tiller '95*]. Pour ce qui concerne notre travail, nous désirons appliquer la méthode retenue au modèle B-spline, pour cela nous utiliserons les transformations réciproques Bézier- B-spline présentées dans le chapitre II.
L'application de la méthode proposée aux courbes B-spline non rationnelles se décompose alors en ces trois étapes qui consiste à:
- décomposer la courbe B-spline en un ensemble de courbes de Bézier de même degré,
- réduire toutes les courbes de Bézier obtenues au degré voulu par l'algorithme PRO#1,
- composer la courbe B-spline réduite à partir des courbes de Bézier réduites obtenues et supprimer les nœuds surabondants [*Hölzle '83*].

Ce qui nous conduit à l'algorithme suivant:

Algorithme PRO#7
Données: P_i (i=0, 1, ..., n) pôles de la courbe B-spline de degré p
t_i (i=0, 1, ..., m=n+p+1) vecteur nœuds
k ordre de la réduction
α ordre de continuité

DECP (P, t → Q, nb)
Pour i=1, 2, ..., nb
  Pour j=0, 1, ..., p
    $R_i=Q_{ij}$
  Fin pour
  PRO#1 (R, k → S)
  Pour j=0, 1, ..., p-k
    $NQ_{ij}=S_j$
  Fin pour
Fin pour
COMP (NQ, nb → NP, Nt)

Résultats: NP_i (i=0, 1, ...,n-k) courbe B-spline de degré p-k
Nt (i=0, 1, ..., m=n+p-k+1) vecteur nœuds
Fin.

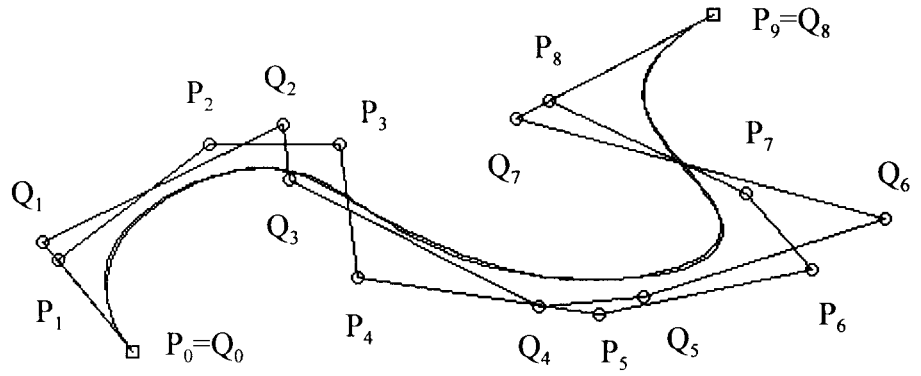

Figure IV.5. Courbe B-spline non rationnelle de degré 6 réduite au degré 5.

La figure IV.5 ci dessus représente la courbe B-spline non rationnelle P(t) de degré 9 ayant pour vecteur nodal T={0,0,0,0,0,0,0,0,0.2,0.4,0.6,1,1,1,1,1,1,1,1} réduite en courbe Q(t) de degré 8 ayant pour vecteur nodal T'={0,0,0,0,0,0,0,0.2,0.4,0.6,1,1,1,1,1,1,1}par l'algorithme proposé PRO#7 en respectant une continuité d'ordre 2. Les deux courbes ont des nœuds internes identiques, la répartition nodale est donc respectée après réduction.

IV.1.5. Courbes B-spline rationnelles

De façon analogue aux courbes non rationnelles, l'application intégrale de la méthode proposée aux courbes B-splines rationnelles se décompose en ces trois étapes qui consiste à:

- décomposer la courbe B-spline en un ensemble de courbes de Bézier de même degré,
- réduire toutes les courbes de Bézier obtenues au degré voulu par l'algorithme PRO#3,
- composer la courbe B-spline réduite à partir des courbes de Bézier réduites obtenues.

Ce qui nous conduit également à l'algorithme suivant:

Algorithme PRO#8
Données: P_i (i=0, 1, ..., n) pôles de la courbe B-spline de degré p
t_i (i=0, 1, ..., m=n+p+1) vecteur nœuds
w_i (i=0, 1, ...,n) charges des pôles
k ordre de la réduction
α ordre de continuité

DECP_R (P, w, t → Q, ξ, nb)
Pour i=1, 2, ..., nb
Pour j=0, 1, ..., p
$R_i=Q_{ij}$, $\psi_j=\xi_{ij}$
Fin pour
PRO#3 (R, ψ, k → S, γ)

Pour j=0, 1, ..., p-k
$NQ_{ij} = S_j$, $N\xi_{ij} = \psi_j$
Fin pour
Fin pour
COMP_R (NQ, Nξ, nb → NP, Nw, Nt)

Résultats: NP_i (i=0, 1, ...,n-k) courbe B-spline de degré p-k
Nt (i=0, 1, ..., m=n+p-k+1) vecteur nœuds
Nw (i=0, 1, ..., n-k) charges des pôles
Fin.

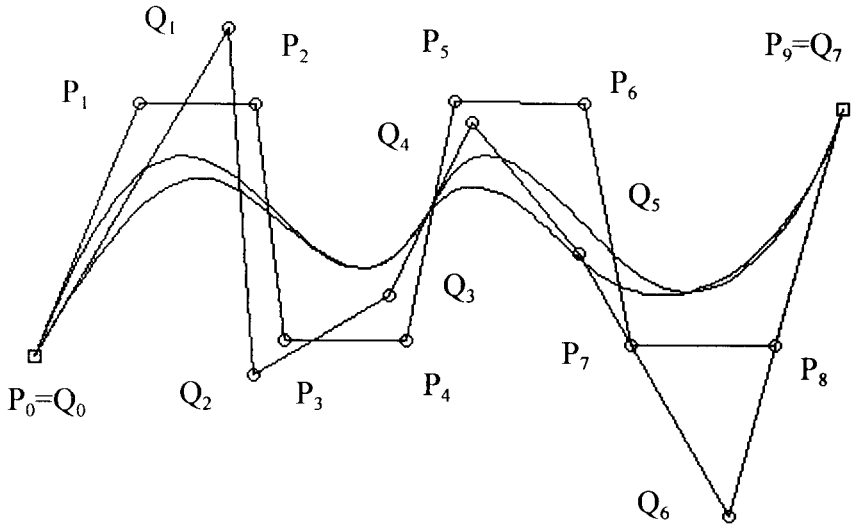

Figure IV.6. Courbe B-spline rationnelle de degré 6 réduite au degré 4.

La figure IV.6 précédente montre une courbe B-spline rationnelle P(t) de degré 9 dont le vecteur nodal T={0,0,0,0,0,0,0,0,0.4,0.5,0.6,1,1,1,1,1,1,1,1} et les charges sont w={1,1.2,0.8,0.5,0.8,1.2,0.5,1.2,0.8,0.5,0.8,1} réduite par l'algorithme proposé en une courbe rationnelle Q(t) de degré 7 ayant T'={0,0,0,0,0,0,0.4,0.5,0.6,1,1,1,1,1,1} pour vecteur nodal et les charges w'={1,0.956,0.458,0.788,0.975,1.95,0.7,1} en respectant une continuité géométrique d'ordre 2 aux extrémités.

IV.2.Transmissions de données

L'un des objectifs essentiels du système proposé concerne le transfert de données d'un logiciel à un autre de façon à ce qu'elles soient reçues dans la forme définie par le logiciel émetteur. Le moyen le plus simple est de créer des interfaces entre logiciels, mais cette solution demande un travail conséquent et est d'un coût élevé car le nombre d'interfaces est donné par la relation N(N-1)/2, N étant le nombre de logiciels à interfacer. Il parait donc plus satisfaisant de songer à des formats standards de données susceptibles d'être crées ou lus par les divers logiciels d'application [*Marty, Cassagnes & Marin '93*] (*Figure IV.7*).

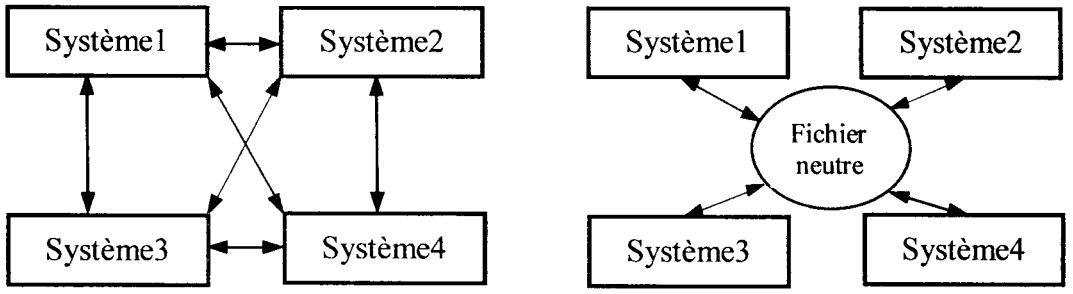

Figure IV.7. Modes de transfert de données

Les principaux standards d'échange les plus couramment utilisés [*Lewis '81*], [*Owen & Bloor '87*] sont représentés par le format natif d'AUTOCAD DXF, le format IGES retenu par la norme ANSI qui est un standard d'échange graphique, le système d'Echange et de Transfert SET, développé autour du projet Airbus et normalisé par l'AFNOR, le système allemand pour l'industrie automobile VDA-FS, pour l'échange de surfaces. Le standard d'échange de données pour les modèles de produits STEP, qui devient un standard international car il ne se limite pas aux entités géométriques mais il traite aussi des entités fonctionnelles ou features et il devait à terme intégrer des données technologiques. Le format neutre DXF- R14 est le plus approprié pour un échange de données au sein de notre laboratoire, mais dans le but de faire une application large pour l'échange de données entre AUTOCAD, CONVER, et SURFCAM, nous avons opté pour le format neutre DXF R12 commun aux deux logiciels. Ce qui a permis de réaliser un pré- processeur limité aux courbes permettant de transformer le format DXF R12 en format DES du logiciel CONVER, ainsi qu'un post- processeur limité aux courbes permettant de transformer le format DES en format DXF. Nous obtenons ainsi la structure de changement de format ci-dessous (Fig.IV.8)

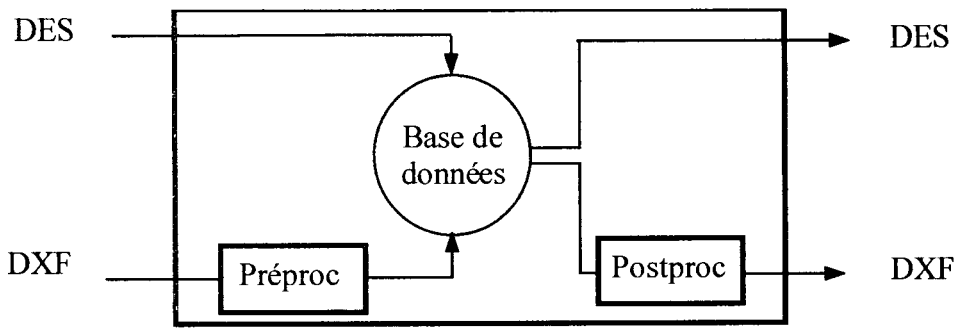

Figure IV.8. Structure de changement de format pour CONVER

qui permet de réaliser la transmission selon le schéma directeur ci-dessous (Figure IV.9).

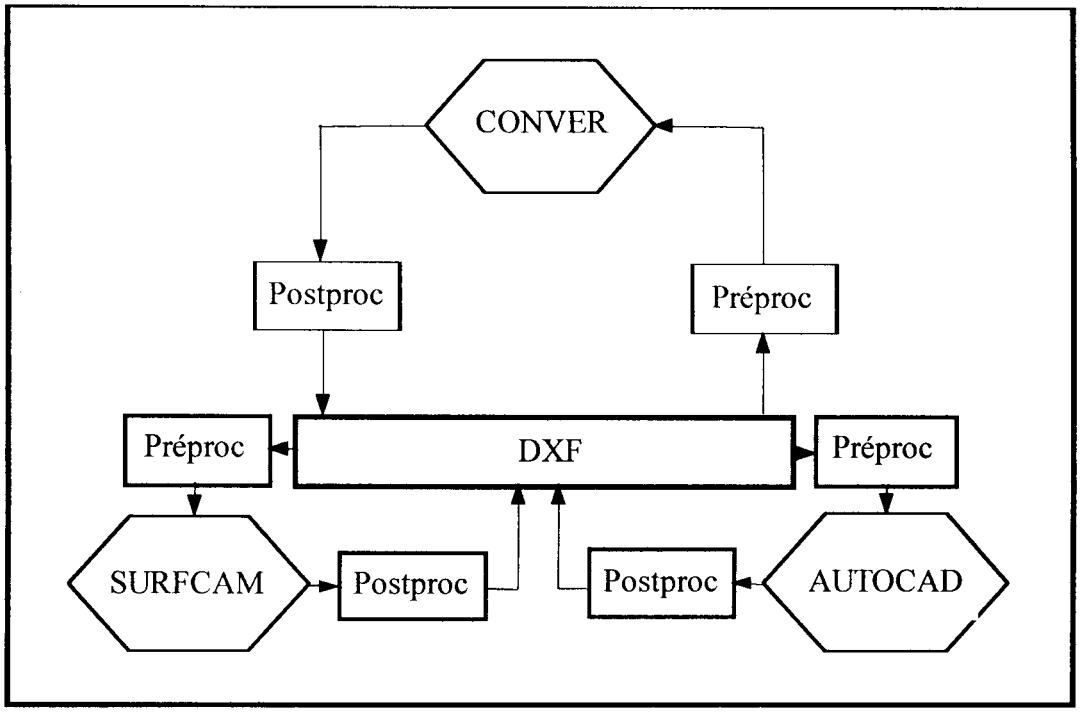

Figure IV.9. Possibilités de transfert de données à l'aide du système élaboré CONVER

D'une part SURFCAM ne dispose pas de format DXF pour les courbes, les transmissions dans un sens: SURFCAM → AUTOCAD et SURFCAM → CONVER ne sont pas réalisables. D'autre part AUTOCAD ne supporte pas les courbes B-spline et Bézier de degré supérieur à 5, on peut transmettre une courbe de degré 18 exécutée par exemple sous CONVER vers AUTOCAD en utilisant le module de réduction de degré. Par contre, La possibilité de passage *CONVER ↔ AUTOCAD* pour les courbes est possible dans les deux sens, les transformations suivantes (*Tableau 5*) sont alors possibles.

CONVER	TRANSFORMATION	AUTOCAD
Bézier de degré ≥ 3	Réduction de degré	Bézier de degré 3
B-spline de degré ≥ 3	Réduction de degré	B-spline de degré 3
B-spline non uniforme de degré ≥ 3	Insertion de nœuds Suppression de nœuds	B-spline uniforme de degré 3
B-spline ou Bézier de degré ≥ 3	Réduction de degré Insertion de nœuds Suppression de nœuds Subdivision Transformation de base	B-spline ou Bézier transformée de degré 3

Tableau 5. Transformations possibles pour un transfert de données

La transmission AUTOCAD → CONVER se fait sans aucune modification puisque CONVER supporte les courbes Bézier et B-spline jusqu'au degré 20.
La transmission AUTOCAD → SURFCAM et CONVER → SURFCAM transforme la courbe en forme polygonale dont le nombre de sommets est égal au nombre de points.

Les développements actuels du format DXF R14 nous permettront de définir d'une manière précise les entités géométriques et d'élargir notre application à toutes les entités courbes rationnelles et les surfaces Bézier, B-spline.
Quant aux anciens fichiers réalisés avec des logiciels se basant sur le format précédant le DXF R12, ils doivent être transformés manuellement suivant la structure globale des formats R12, 13 et 14.

Nous reproduisons dans ce qui suit les structures des fichiers DXF R12 et des fichiers DES, développés sous CONVER.

a) Structures des fichiers DXF R12
Chaque fichier DXF comporte 3 parties dont la première désignée par HEADER représente l'en tête qui comporte l'état et le type du logiciel ayant créée le fichier, on trouvera la version du 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 définie 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 appelée ENTITIES 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			
	TYPE		
		Entrées	
			Groupes
	COUCHE		
		Entrées	
			Groupes
	STYLE		
		Entrées	
			Groupes
	...		
BLOCS			
		Entitées	
			Groupes
ENTITEES			
		Entitées	
			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 {fin de section}

pour la définition des entités, la chaîne1 prend l'affectation ENTITIES et sera suivie de

0
Chaîne2 {chaîne de caractères définissant l'entité. Peut prendre les affectations: POINT, LINE, ARC, CIRCLE, POLYLINE, ...}

à la fin de la définition de chaque entité on trouve

SEQEND {fin de séquence}.

Pour définir une courbe B-spline uniforme non rationnelle, on affecte POLYLINE à la chaîne2 qui sera suivie de

8 {valeur du 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 polyline, de "8" pour un point, ou de "16" pour un pôle}
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

b) Structure des fichiers DES

Le format DES ne fait pas intervenir les chaînes de caractères, il utilise directement les valeurs numériques. Les entités se suivent sans qu'elles soient séparées. Chaque entité est définie par:

t	entier définissant l'entité {"1" pour une courbe de Bézier non rationnelle}
Color	couleur du trait
nbr_pts	nombre de paramètres
nbr_p	nombre de points
Parm1	Premier paramètre
Parm2	
Parm3	
.....	
ParmN	$N^{ième}$ paramètre

Cette structure permet d'insérer de nouvelles entités en les codifiant sans avoir à modifier la programmation du logiciel CONVER.

Conclusion

Conclusion

Ce travail nous a amené à établir une synthèse bibliographique très approfondie. Les différentes méthodes de réduction de degré des courbes de Bézier non rationnelles recensées sont présentées de façon claire et bien structurée, en exposant pour chacune d'elle d'abord le principe général suivi des algorithmes correspondants détaillés, présentés sous une forme procédurale très simple et prête à une implémentation adaptée aux langages courant de programmation.

A la lumière de l'étude comparative effectuée sous forme algorithmique et expérimentale, nous avons constaté d'une part que le problème de la meilleure approximation demeure toujours posé. Toutefois, la notion de qualité d'équivalence de CFAO introduite dans [*Belaidi '98*] a permis d'apporter des élements de réponse à ce problème. D'autre part, les résultats de notre étude montrent 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 de surfaces de Bézier non rationnelles, mais produit une erreur très élevée et un nombre de courbes réduites très grand, sans assurer de continuité géométrique au-delà de l'ordre1, à l'opposé de l'algorithme de Forrest qui assure la continuité géométrique maximale. Cependant, cette méthode de réduction de degré induit une erreur d'approximation très élevée qui donne par conséquent un nombre élevé de courbes réduites résultantes. 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 0 et les temps de calcul sont tout de même raisonnables, à l'instar des deux méthodes ci-dessus et de la méthode de Eck. Cette dernière peut être considérée comme une généralisation des méthodes précédentes, qui assure en outre un ordre de continuité souhaité pour une erreur de position minimale Cette méthode nécessite toutefois une approximation qui fait intervenir un algorithme spécifique pour le calcul des coefficients de pondération qui déterminent les nouveaux pôles de la courbe, ce qui constitue une restriction limitant l'application de cette méthode à des degrés faibles. La méthode développée par Hoschek est basée sur la technique d'approximation des courbes osculatrices pour la réduction de degré qui aboutit à un système d'équations non linéaire dont la résolution nécessite un grand effort de calcul et une certaine lenteur. L'avantage très appréciable 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 à d'autres modèles non rationnels.

Les résultats de cette étude comparative montrent que les méthodes de réduction de degré des courbes de Bézier présentées comportent encore des imperfections qu'il est possible d'améliorer. Ce qui nous a conduit à proposer une méthode de réduction de degré des courbes de Bézier non rationnelles, basée sur une notion de séparation des variables et une stratégie de subdivision, mieux adaptée aux applications potentielles

ou effectives de la C.F.A.O. Cette méthode aboutit en effet à un système d'équations linéaire dépendant simplement du nombre de points, du degré de la courbe initiale, du degré de la courbe finale ainsi que de l'ordre de continuité géométrique, ce qui par conséquent facilite la transformation du système d'équations en équations directes permettant de rendre le processus de réduction plus rapide. En outre, son degré de flexibilité et sa facilité de passage aussi bien aux modèles rationnels et aux surfaces de Bézier qu'aux modèles B-spline rationnels et non rationnels ont permis de développer des extensions à même d'élargir le domaine d'application de la réduction de degré en CFAO, notamment dans son application à l'échange de données entre systèmes.

Les transformations principales (réduction et élévation de degré, insertion et suppression de nœuds, subdivision, changement de base) intervenant dans la communication de données entre systèmes, ont été présentées dans le détail et intégrées dans l'application proposée. Cette dernière permet en effet un échange de données entre le programme élaboré « CONVER » et les logiciels « AUTOCAD » et « SURFCAM » par l'intermédiaire du format standard d'échange de données graphiques « DXF », levant ainsi la contrainte de limitation du degré des courbes imposée à ces logiciels et qui réduisait la flexibilité de construction et de modifications des courbes de Bézier et B-spline.

Les fondements de la méthode proposée pour la réduction de degré des courbes de modèle B-spline par transformation de la base des polynômes, en gardant la notion de séparation des variables, est à envisager pour aboutir à une méthode de réduction directe, qui peut être étendue aisément aux surfaces.
Les algorithmes présentés dans ce travail peuvent être combinés et généralisés à des applications pratique en CFAO, 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 B-spline uniforme et inversement..., ce qui constituera une richesse des fonctionnalités de logiciels industriels au niveau de la flexibilité de conception de formes et la communication entre systèmes basée sur des standards d'échange plus performants et normalisés.

Références bibliographiques

[Asma, Benouali & Belaidi '93] F.ASMA, M.BENOUALI & I.BELAIDI
Subdivision réduction des courbes et surfaces de Bézier et influence sur la qualité de C.F.A.O, Projet de fin d'études, INGM 1993.

[Asma, Belaidi, Ishiomin & Coffignal'99] F.ASMA, I.BELAIDI, G.ISHIOMIN & G.COFFIGNAL
Réduction de degré des courbes de Bézier de degré élevé : une étude comparative des méthodes publiées. A paraître.

[Asma, Belaidi, Ishiomin & Coffignal'99] F.ASMA, I.BELAIDI, G.ISHIOMIN & G.COFFIGNAL
Réduction de degré des courbes de Bézier de degré élevé : une solution par séparation des variables. A paraître.

[Bardis & Patrikalakis '89] L.BARDIS & N.M.PATRIKALAKIS
Approximate conversion of rational B-splines patches, Computer Aided Geometric Design, Vol 6 1989, pp 189-204

[Belaïdi & Asma '94] I.BELAIDI & F.ASMA
Méthodes de conversion par approximation des courbes de Bézier de degré élevé : Etude comparative ; Rapport interne N°1 du Laboratoire de CFAO INGM Boumrdès ; 1993-94.

[Belaïdi '98] I.BELAIDI
Réduction de degré des courbes de Bézier, Qualité d'équivalence en relation avec des paramètres d'usinage, Thèse de doctorat, LMS-ENSAM-CER de Paris 1998

[Bensalah '90] S.BENSALAH
Equivalence et algorithmique des transformations dans les modèles mathématiques pour la conception et la fabrication assistées par ordinateur des courbes et des surfaces, Thèse de doctorat, ENSAM 1990

[Böehm & Prautzch '85] W.BÖEHM and H.PRAUTZCH
The insertion algorithm, Computer Aided Design, Vol 17 N°2 Mars 1985 pp 58-59.

[Bogaki, Weinstein & Xu '95] P.BOGACKI, S.E.WEINSTEIN and Y.XU
Degree reduction of Bézier curves by uniform approximation with endpoint interpolation, Computer Aided Design, Vol 27 N°9 Sept 1995 pp 651-661

[Brunnett, Schreiber & Braun '96] G.BRUNNETT, T.SCHREIBER and J.BRAUN
The geometry of optimal degree reduction of Bézier curves, Computer Aided Geometric Design, N°13 1996 pp 773-788

[Dannenberg & Nowacki '85] L.DANNENBERG and H.NOWACKI
Approximate conversion of surfaces representations with polynomial bases, Computer Aided Geometric Design, N°2 1985 pp 123-131

[Eck & Lasser '92] M.ECK and D.LASSER
B-spline-Bézier representation of geometric spline curves : quartics and quintics, Computer Math Applic 1992 Vol 23 N° 11 pp 23-39

[Eck '93] M.ECK
Degree reduction of Bézier curves, Computer Aided Geometric Design, N°10 1993 pp 237-251

[Eck '94] M.ECK and J.HADENFELD
A septwise algorithm for converting B-splines, Curves and Surfaces in Geometric Design, 1994 pp 131-138

[Eck '95] M.ECK
Least squares degree reduction of Bézier curves, Computer Aided Design, Vol 27 N°11 Nov 1995

[Farin '83] G.FARIN
Algorithms for rational Bézier curves, Computer Aided Design, Vol 15 N°2 Mar 1983 pp 73-77

[Farouki & Neff '90] R.T.FAROUKI and C.A.NEFF
On the numerical condition of Bernstein-Bézier subdivision processes, Mathematics of Computation, Vol 55 N°192 Oct 1990 pp 637-647

[Fiorot '87] J.C.FIOROT
Courbes Bézier, Mathématiques, Informatique et Applications, CNRS INSA de RENNES Mai 1987 pp 4-18

[Forrest '72] A.R.FORREST
Interactive interpolation and approximation by Bézier polynomials, Computer .J 1972, Vol 15, pp 71-79

[Forrest '90] A.R.FORREST
Interactive interpolation and approximation by Bézier polynomials, Computer Aided Design, Vol22, N° 9 Nov 1990 pp 527-537

[Hölzle '83] G.E.HÖLZLE
Knot placement for piecewise polynomial approximation of curves, Computer Aided Design, Vol15 N°5 Sept 1983 pp 295-296

[Hoschek '86] J.HOSCHEK
Approximate conversion of spline curves, Computer Aided Geometric Design (4) 1987 pp 59-66

[Hoschek '88] J.HOSCHEK
Intrinsic parametrisation for approximation, Computer Aided Geometric Design N°5 1988 pp 27-31

[Hoschek & Wissel '88] J. HOSCHEK and N.WISSEL
Optimal approximate conversion of spline curves and spline approximation of offset curves, Computer Aided Design, Vol 20, N°8 Oct 1988 pp 475-482

[Hoschek, Schneider & Wassum '89] J.HOSCHEK, F.J. SCHNEIDER, P.WASSUM
Optimal approximate conversion of 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

[Hoschek & Schneider '90] J.HOSCHEK and F.J.SCHNEIDER
Spline conversion for trimmed rational Bézier and B-spline surfaces, Computer Aided Design, Vol 22 N°9 Nov 1990 pp 580-590

[Joe, Wang & Cheng '94] B.JOE, W.WANG and F.CHENG
Reduced-knot NURBS representations of rational G^1 composite Bézier curves, Computer Aided Design, Vol26 N°5 May 1994

[Lachance '88] M.A.LACHANCE
Chebyshev economization for parametric surfaces, Computer Aided Geometric Design N°5 1988 pp 195-208

[Léon '91] J.C LEON
Modelisation et construction de surfaces pour la C.F.A.O, Hermès 1991

[Lewis '81] J.W.LEWIS
Interchanging spline curves using IGES, Computer Aided Design, Vol 13, N°6 Nov 1981 pp 359-363

[Marty, Cassagnes & Marin '93] C.MARTY, C.CASSAGNES et P.MARIN
La pratique de la commande numérique des machines-outils, Lavoisier 1993

[Meyer '87] M.MEYER
Algorithme WM, Renault Automation N° E152, 1987

[Owen & Bloor '87] J.OWEN and M.S.BLOOR
Neutral format for product data exchange: the current situation, Computer Aided Design, Vol 19 N°8 Oct 1987 pp4364-43

[Patrikalakis '89] N.M.PATRIKALAKIS
Approximate conversion of rational splines, Computer Aided Geometric Design, N°6 1989 pp 155-165

[Piegl '85] L.PIEGL
Recursive algorithms for the representation of parametric curves and surfaces, Computer Aided Design, Vol 17 N°5 June 1985 pp 225-229

[Piegl '86] L.PIEGL
Representation of rational Bézier curves and surfaces by recursive algorithms, Computer Aided Design, Vol 18 N°7 Sept 1986 pp 361-366

[Piegl & Tiller '87] L.PIEGL and W.TILLER
B-splines, Computer Aided Design, Vol19 N°9 Nov 1987 pp 485-498

[Piegl '91] L.PIEGL
On NURBS: A Survey, IEEE Computer Graphics & Applications, Jan 1991 pp 55-71

[Piegl & Tiller '94] L.PIEGL and W.TILLER
Software-engineering approach to degree elevation of B-spline curves, Computer Aided Design, Vol 26 N°1 Jan 1994

[Piegl & Tiller '95] L.PIEGL and W.TILLER
Algorithm for degree reduction of B-spline curves, Computer Aided Design, Vol 27 N°2 FEB 1995 pp 101-110

[Sablonnière '87-1] P.SABLONNIERE
Courbes B-splines, Mathématiques, Informatique et Applications, CNRS INSA de RENNES Mai 1987 pp 19-47

[Sablonnière '87-2] P.SABLONNIERE
Approximation de formes (courbes), Mathématiques, Informatique et Applications, CNRS INSA de RENNES Mai 1987 pp 48-62

[Sablonnière '87-3] P.SABLONNIERE
Produits tensoriels Bézier/B-spline, Mathématiques, Informatique et Applications, CNRS INSA de RENNES Mai 1987 pp 63-88

[Sederberg & Farouki '92] T.W.SEDERBERG and R.FAROUKI
Approximate by interval Bézier curves, IEEE Computer Graphics & Application, Sept 1992 pp 87-95

[Watkins & worsey '88] M.A.WATKINS and A.J.WORSEY
Degree reduction of Bézier curves, Computer Aided Design, Vol 20, N°7 Sept 1988 pp 398-405

Annexes

A. Logiciel Conver
1. Structure de données

La structure de données étant destinée à être implémentée dans un logiciel de C.F.A.O, elle doit permettre un certain nombre de manipulations dont les plus importantes sont: [*Asma, Benouali & Belaïdi'93*]

- Mémoriser toutes les entités reconnues par le logiciel,
- permettre une mise à jour continuelle et rapide (insertion de nouvelles entités, suppression d'entités existantes, sélection d'entité, ...)
- permettre l'implantation de fonctions de manipulation efficace à fin de garantir une bonne convivialité du logiciel,
- permettre une bonne gestion de la mémoire.

Une base de données principale et des structures de données annexes sont prévue permettant l'implantation des fonctions de manipulation optimales. A titre d'exemple, dans un projet composé de plusieurs calques, la base principale contiendra la totalité des calques, par contre une structure annexe de manipulation ne contiendra que le ou les calques actifs susceptibles d'être manipulés (l'espace mémoire occupé n'en sera pas augmenté de beaucoup puisque la partie « données » (DATA) sera commune aux deux structures.

A partir des exigences définies au-dessus et de la démarche adoptée on commencera par définir la section DATA capable de représenter toutes les entités manipulées par notre logiciel. Ace stade à fin de gagner du temps on préfère s'inspirer d'un logiciel existant plutôt que de reprendre le travail à zéro. Les logiciels retenus sont DMT et CADKEY vu qu'ils existent au niveau du laboratoire. Le logiciel DMT étant le plus documenté en ce qui concerne sa structure de données internes, on adoptera une structure similaire.

Liste des différentes entités supportées

Bien qu'au stade actuel notre logiciel n'est définis que pour supporter quelques entités, on envisage dans le futur une extension pour toutes les entités graphiques supportées par les logiciels de C.F.A.O. La structure tiendra compte de cela et devra supporter les entités suivantes: Point, Droite, Segment, Arc de cercle, Ellipse, Arc d'ellipse, Courbe par pôles, Courbe par points, Surface, Profil, Repère, Profil C.N.

Définition de la section DATA commune à toutes les entités

Les structures statiques étant exclues pour ce type d'application, le choix se fera parmi les structures dynamiques; Parmi ces dernières la structure en liste chaînée est la plus adaptée puisqu'elle répond aux exigences requises.

Pour la réalisation de la liste chaînée, il faut rajouter au type de base d'une entité définissant un nœud de la liste ce qu'on appellera le lien et qui ne sera rien d'autre qu'un pointeur qui pointera sur l'entité suivante, afin de pouvoir initialiser la liste (liste vide). On définira deux éléments particuliers qui représenteront la tête et la fin de liste.

La section "données" commune à toutes les entités sera définie à travers un type enregistrement contenant les champs suivants:

```
P_Entite = ^Entite;

   Entite = record
              t              : integer;     { type d'entité }
              color          : integer;     { couleur }
              nbr_pts        : integer;     { nombre de paramètres }
              nbr_p          : integer;     { nombre de points }
              prem_pe        : PPoint ;     { premier point }
              dern_pe        : PPoint ;     { dernier point }
              selectionee    : integer;     { variable booléenne de sélection ou
d'effacement}
              E_S            : P_Entite;    { entité suivante }
            end;
```

Le type point sera défini comme un enregistrement:

```
PPoint = ^Point;
   Point  = record
              Val1      : double;   { coordonnée X du point }
              Val2      : double;   { coordonnée Y du point }
              Val3      : double;   { coordonnée Z du point }
              P_S       : Ppoint;   { point suivant }
            end;

   Liste_E=Record
              Une_E          : Entite;      { nœud }
              E_S            : ^Entité;     { lien }
            end;
```

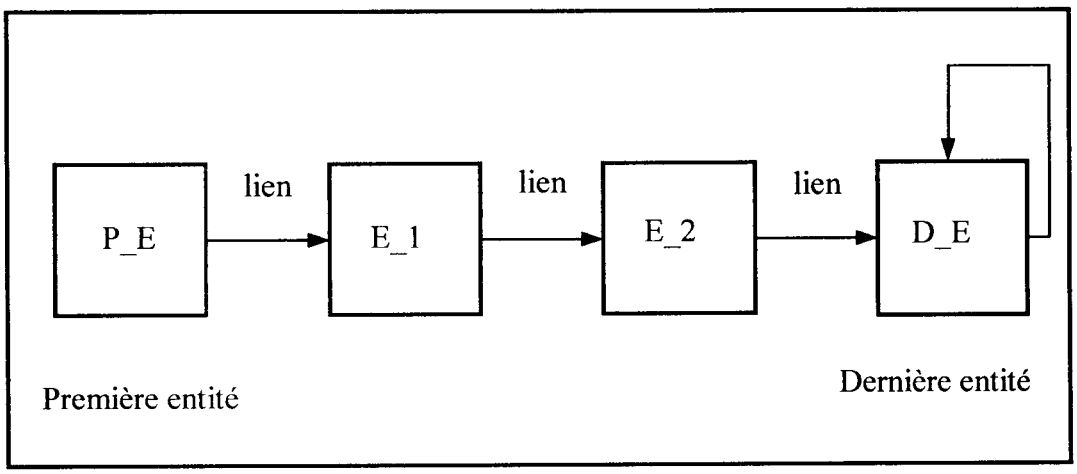

Figure 1. Liste linéaire des entités

Interface utilisateur

Lancement

Le lancement du logiciel ce fait sous DOS par l'appel CONVER et met à la disposition de l'utilisateur l'écran divisé en cinq parties (Figure 2):

(1) La partie au centre de l'écran est la fenêtre de travail. Tous les dessins et traitements seront visualisé dans cette partie,

(2) la partie de droite représente les menus et sous- menus. C'est l'outil de dialogue qui permet grâce à la souris de choisir le traitement voulu,

(3) la partie du bas représente le masque de saisie de paramètres, où on peut changer quelques variables du logiciel (ces variables sont dites externe),

(4) la partie de dessus affiche la position du curseur graphique dans l'espace et la mémoire vive restante. Cette partie est appelée zone message car elle affiche les messages courants pendant le traitement,

(5) la partie de dessous est appelée phrase d'exécution qui explique la nature du traitement exécuté.

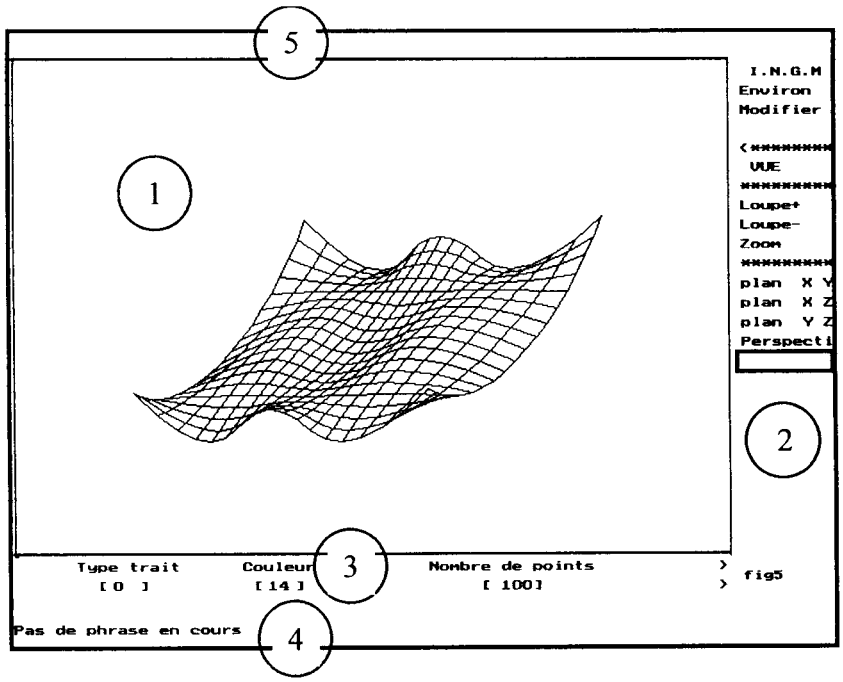

Figure 2. Logiciel CONVER

Structure des fichiers MNU

Une ligne d'un fichier MNU est formée comme suit:

exemple: I.N.G.M. 0 principa.mnu far.mnu
(1) (2) (3) (4)

(1) Représente la chaîne alphanumérique à afficher dans la zone menu,

(2) donne le code permettant de reconnaître le traitement et la compréhension de la manipulation,

(3) donne le nom du fichier menu à afficher après avoir fini le traitement (très utile pour le déroulement des menus)

(4) donne le nom du fichier masque correspondant à la manipulation.

On retrouve généralement dans chaque fichier menu les symboles suivant:

I.N.G.M. : permet de revenir au menu principal (début)
<<******* : permet de revenir au menu précédent
********> : permet de passer au menu suite s'il existe
<******* : permet de revenir au menu début après avoir visualisé le menu suite
C*P : donne accès rapide au masque de saisie de paramètres
Ce qui donne une structure arborescente de l'outil de dialogue.
Structure des fichiers MAS

Il est donné à chaque paramètre pouvant être modifié à l'aide du masque un numéro, il s'agit là de variables dites externes.
Les fichiers masques sont formés de trois lignes; Les deux premières sont des chaînes de caractères qu'il se charge de lire et d'écrire sans traitement, alors que la troisième comporte des codes numériques:

Type de trait Couleur Rayon >

exemple: [ ] [ ] [ ] >
100 6 140 12 268 18 1 2 6
(1) (2) (3)

(1) Donne le code permettant de détecter la position d'écriture en mode graphique de la valeur d'un paramètre
(2) donne le code permettant de détecter la position d'un paramètre en mode texte
(3) donne le numéro correspondant à la variable externe comprise dans le masque dans l'ordre d'écriture.
Cette structure des fichiers masque et menu nous permet d'intégrer un nouveau module dans le logiciel sans être obligé de modifier ni le programme principal, ni les autres modules existants. Ajouté à cela les méthodes de transformation et d'analyse des courbes et des surfaces, nous avons obtenu la structure finale suivante

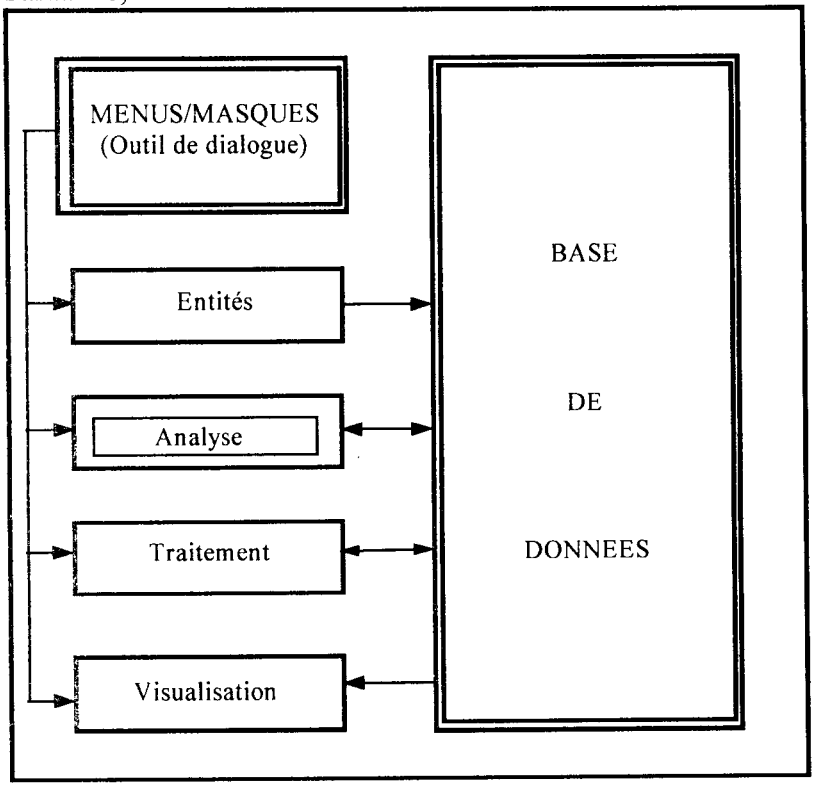

Figure 3. Structure du logiciel CONVER

2. Architecture et Manipulations

Il est très important de simplifier la manipulation pour principalement faciliter le travail pour l'utilisateur et lui permettre de générer le dessin d'une manière simple et aisée. Le temps de réponse requiert alors une importance primordiale.
Le lancement du programme se fait par la fonction CONVER

Entités géométriques

- Crée_Ent: Il permet de dessiner les entités de types courbes géométriques 3D Bézier et B-spline rationnelles et non rationnelles.
- Surface: Permet la construction de surfaces rationnelles et non rationnelles de Bézier et B-spline.

Fonctions de visualisation

- Plans XY, YZ, ZX: permet de visualiser le dessin dans un plan différent, ce qui nous permet d'obtenir les vues de face, de gauche et de dessus.
- Perspective: représentation du dessin suivant une perspective cavalière

  ACCES: Vue Plan XY
  Vue Plan YZ
  Vue Plan ZX
  Perspective
- Zoom: Donne la possibilité de représenter d'une manière agrandie les entités contenues dans le rectangle élastique que l'utilisateur défini par deux points diagonaux. La zone rectangulaire contenue dans le curseur élastique est affichée en plein écran

  ACCES: Vue Zoom
- Loupe+: Multiplie la valeur du grossissement de la fenêtre par un coefficient (égal à 2)

  ACCES: Vue Loupe+
- Loupe-: Divise la valeur du grossissement de la fenêtre par un coefficient (égal à 2)

  ACCES: Vue Loupe-
- Retracer: Affiche le contenu de la fenêtre même si celle ci a été déjà effacée

  ACCES: Retracer
- Effacer la fenêtre de travail: Permet d'effacer la fenêtre de travail pour éliminer les surcharges sur le dessin

  ACCES: Eff_fent

Création d'entités

- Courbe de Bézier non rationnelle

  ACCES: Cree_Ent Courbe C_Be_Cast
  ACCES: Cree_Ent Courbe C_Be_Ber
- Courbe de Bézier rationnelle

ACCES: Cree_Ent Courbe C_Be_Rati

- Courbe B-spline non rationnelle
  ACCES: Cree_Ent Courbe C_Bspl_NR
- Courbe B-spline rationnelle
  ACCES: Cree_Ent Courbe C_Bspl_RA
- Surface de Bézier non rationnelle
  ACCES: Surface bez_norat
- Surface de Bézier rationnelle
  ACCES: Surface bez_rat
- Surface B-spline non rationnelle
  ACCES: Surface Bsp_norat
- Surface B-spline rationnelle
  ACCES: Surface Bsp_rat

Caractéristiques des courbes

- Visualisation de la courbure pour les courbes de Bézier et B-spline rationnelles et non rationnelles
  ACCES: Cree_Ent Courbe Traiter Courbure
- Visualisation des points singuliers des courbes de Bézier et B-spline rationnelles et non rationnelles
  ACCES: Cree_Ent Courbe Traiter Pts_sing1
  ACCES: Cree_Ent Courbe Traiter Pts_sing2
  ACCES: Cree_Ent Courbe Traiter Pts_Infl
- Visualisation du polygone caractéristique pour les courbes de Bézier et B-spline rationnelles et non rationnelles
  ACCES: Cree_Ent Courbe Traiter Polygone
- Comparaison graphique des courbures de deux ou plusieurs courbes sélectionnées à la fois parmi les courbes de Bézier et B-spline rationnelles et non rationnelles
  ACCES: Cree_Ent Courbe Traiter Comp_cour

Fonction de modification

- Déplacer un pôle: Permet de déplacer un pôle pour une courbe de Bézier non rationnelle, ce qui modifie complètement la courbe et crée une nouvelle
  ACCES: Cree_Ent Courbe Modif_c Dep_pole
- Insérer un pôle: Permet d'insérer un pôle dans le polygone caractéristique d'une courbe de Bézier non rationnelle, la nouvelle courbe passera par ce pôle puisqu'il est considéré comme le dernier pôle du polygone, la courbe devient de degré (n+1)
  ACCES: Cree_Ent Courbe Modif_c Ins_pole
- Tirer sur un pôle: Permet de modifier l'allure d'une courbe de Bézier non rationnelle en la faisant tendre vers un pole; La courbe devient de degré (n+1)
  ACCES: Cree_Ent Courbe Modif_c Tir_pole

- Supprimer un pôle: Permet de supprimer un pôle du polygone caractéristique d'une courbe de Bézier non rationnelle, ce qui modifie l'allure de la courbe et devient de degré (n-1)

  ACCES: Cree_Ent Courbe Modif_c Eff_pole

- Réduction de degré: Permet de réduire le degré d'une courbe ou d'une surface; On obtient alors une courbe de degré (n-k) ou une surface de degré (m-k1, n-k2) plusieurs méthodes sont ainsi possibles:

ACCES:

Pour les courbes de Bézier rationnelles et non rationnelles:

Cree_Ent	Courbe	Traiter	Polygone1	
Cree_Ent	Courbe	Traiter	Polygone2	
Cree_Ent	Courbe	Traiter	Réduction	
Cree_Ent	Courbe	Traiter	Sub-Red	PRO#
Cree_Ent	Courbe	Traiter	Sub-Red	FOR-BEN
Cree_Ent	Courbe	Traiter	Sub-Red	FOREST
Cree_Ent	Courbe	Traiter	Sub-Red	BENSALAH1
Cree_Ent	Courbe	Traiter	Sub-Red	BENSALAH2
Cree_Ent	Courbe	Traiter	Sub-Red	ECK
Cree_Ent	Courbe	Traiter	Sub-Red	HOSCHEK
Cree_Ent	Courbe	Traiter	Sub-Red	WATKINS

Pour les courbes B-spline rationnelles et non rationnelles:

Cree_Ent	Courbe	Traiter	Sub-Red	PRO#

Pour les surfaces de Bézier rationnelles et non rationnelles:

Surface	PRO#
Surface	Polygone1
Surface	Polygone2
Surface	Polygonem

- Subdivision: Permet de subdiviser une courbe de Bézier ou B-spline rationnelles ou non rationnelle en deux courbes de même degré au point t=0.5

  ACCES: Cree_Ent Courbe Traiter Subdiv

- Insertion de nœuds: Permet d'insérer un nœud pour les courbes B-spline rationnelles et non rationnelles

  ACCES: Cree_Ent Courbe Convers Insertion

- Suppression de nœuds: Permet de supprimer un nœud pour les courbes B-spline rationnelles et non rationnelles

  ACCES: Cree_Ent Courbe Convers Suppression

- Transformation d'une courbe de Bézier non rationnelle en une courbe B-spline non rationnelle

  ACCES: Cree_Ent Courbe Convers 1BZnr 1BS

- Transformation d'une courbe de Bézier rationnelle en une courbe B-spline rationnelle

  ACCES: Cree_Ent Courbe Convers 1 BZrt 1BS

- Transformation d'une courbe B-spline non rationnelle en plusieurs courbes de Bézier non rationnelles

ACCES: Cree_Ent Courbe Convers 1BSnr MBZ

- Transformation d'une courbe B-spline rationnelle en plusieurs courbe de Bézier rationnelles
  ACCES: Cree_Ent Courbe Convers 1 BSrt MBZ

Analyse:

On ne peut accéder au module d'analyse qu'après avoir fait une subréduction. Il permet de faire une comparaison des caractéristiques de la courbe mère et des courbes réduites obtenues.

- Visualisation de la courbure de la courbe mère et des courbes réduites obtenues sur un même graphe.
  ACCES: Courbure
- Visualisation de l'erreur sur la courbure entre la courbe mère et les courbes réduites
  ACCES: Comp_cour
- Visualisation de l'erreur sur la tangente entre la courbe mère et les courbes réduites
  ACCES: tangente
- Visualisation de l'écart de position entre la courbe mère et les courbes réduites
  ACCES: position

Entrée - Sortie

- Recherche d'un fichier
  ACCES: Ouvrir
- Lecture d'un fichier de type DES (crée par CONVER ou MENU0)
  ACCES: Charger_f
- Lecture d'un fichier de type DXF (format d'AUTOCAD)
  ACCES: Format DXF_IN
- Affectation d'un nouveau nom à un fichier
  ACCES: nouv_nom
- Enregistrement d'un dessin sous le format DES
  ACCES: Sauver
- Enregistrement d'un dessin sous le format DXF
  ACCES: Format DXF_OUT
- Quitter le logiciel CONVER
  ACCES: Sortir

B. Equations de Continuité géométrique d'après Hoschek

k=3 → (59.c)

$$Q_3 = P_0 + (P_3 - P_2).\lambda_1^3.\omega_2 + (P_2 - P_1).\left[\lambda_1^3.\left(-2.\omega_2 + 3.\omega_1^2.\frac{m-1}{m-2}\right) + \lambda_1^2.\omega_1.\left(3 - 6.\frac{m-1}{m-2}\right) + 3.\lambda_1.\lambda_2.\omega_1.\frac{m-1}{m-2}\right] + (P_1 - P_0).\lambda_3.$$

$$Q_{m-3} = P_n + (P_{n-3} - P_{n-2}).\mu_1^3.\omega_2 + (P_{n-2} - P_{n-1}).\left[\mu_1^3.\left(-2.\omega_2 + 3.\omega_1^2.\frac{m-1}{m-2}\right) + \mu_1^2.\omega_1.\left(3 - 6.\frac{m-1}{m-2}\right) + 3.\mu_1.\mu_2.\omega_1.\frac{m-1}{m-2}\right] + (P_{n-1} - P_n).\mu_3.$$

k=4 → (59.d)

$$Q_4 = P_0 + (P_4 - P_3).\lambda_1^4.\omega_3 + (P_3 - P_2).\{\lambda_1^4.a_{40} + \lambda_1^3.a_{30} + \lambda_1^2.\lambda_2.a_{21}\} + (P_2 - P_1).\{\lambda_1^4.a_{400} + \lambda_1^3.a_{300} + \lambda_1^2.b_{200} + \lambda_2^2.b_{020} + \lambda_1^2.\lambda_2.b_{210} + \lambda_1.\lambda_2.b_{110} + \lambda_1.\lambda_3.b_{101}\} + (P_1 - P_0).\lambda_4$$

$$Q_{m-4} = P_n + (P_{n-4} - P_{n-3}).\mu_1^4.\omega_3 + (P_{n-3} - P_{n-2}).\{\mu_1^4.a_{40} + \mu_1^3.a_{30} + \mu_1^2.\mu_2.a_{21}\} + (P_{n-2} - P_{n-1}).\{\mu_1^4.a_{400} + \mu_1^3.a_{300} + \mu_1^2.b_{200} + \mu_2^2.b_{020} + \mu_1^2.\mu_2.b_{210} + \mu_1.\mu_2.b_{110} + \mu_1.\mu_3.b_{101}\} + (P_{n-1} - P_n).\mu_4$$

avec $\omega_1 = \frac{m.(n-1)}{n.(m-1)}$, $\omega_2 = \frac{m^2.(n-1).(n-2)}{n^2.(m-1).(m-2)}$, $\omega_2 = \frac{m^3.(n-1).(n-2).(n-3)}{n^3.(m-1).(m-2).(m-3)}$

$a_{40} = 6.\frac{m-1}{m-3}.\omega_1.\omega_2 - 3.\omega_3$, $a_{30} = \left(4 - 12.\frac{m-1}{m-3}\right).\omega_2$, $a_{21} = 6.\frac{m-1}{m-3}.\omega_2$

$b_{400} = 3.\omega_3 - 16.\frac{m-1}{m-3}.\omega_1.\omega_2 + 15.\frac{(m-1)^2}{(m-2).(m-3)}.\omega_1^3$

$b_{300} = 12.\frac{m-1}{m-2}.\omega_1^2 - 24.\frac{m-1}{m-3}.\omega_2 - 8\omega_2 - 36.\frac{(m-1)^2}{(m-2).(m-3)}.\omega_1^2$

$b_{200} = \omega_1.\left(6 + 12.\frac{m-1}{m-3} - 24.\frac{m-1}{m-2} + 12.\frac{(m-1)^2}{(m-2).(m-3)}\right)$

$b_{020} = 3.\frac{(m-1)^2}{(m-2).(m-3)}.\omega_1$, $b_{210} = 18.\frac{(m-1)^2}{(m-2).(m-3)}.\omega_1^2 - 12.\frac{m-1}{m-3}.\omega_2$

$b_{110} = 12.\omega_1.\left(\frac{m-1}{m-2} - \frac{m-1}{m-3} - \frac{(m-1)^2}{(m-2).(m-3)}\right)$, et $b_{101} = 4.\frac{m-1}{m-3}.\omega_1$

C. Quelques valeurs des matrices de changement de base Bézier - Tchebycheff d'après Watkins et Worsey

n=2:

$$T_{BM}.T_{MT} = \frac{1}{8}\begin{pmatrix} 3 & 4 & 1 \\ 2 & 0 & -2 \\ 3 & 4 & 1 \end{pmatrix} \text{ et } T_{MT}^{-1}.T_{BM}^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & -3 & 1 \end{pmatrix}$$

n=3:

$$T_{BM}.T_{MT} = \frac{1}{32}\begin{pmatrix} 10 & -15 & 6 & -1 \\ 6 & -3 & -6 & 3 \\ 6 & 3 & -6 & -3 \\ 10 & 15 & 6 & 1 \end{pmatrix} \text{ et } T_{MT}^{-1}.T_{BM}^{-1} = \frac{1}{3}\begin{pmatrix} 3 & 3 & 3 & 3 \\ -3 & -1 & 1 & 3 \\ 3 & -5 & -5 & 3 \\ -3 & 15 & -15 & 3 \end{pmatrix}$$

n=4:

$$T_{BM}.T_{MT} = \frac{1}{128}\begin{pmatrix} 35 & -56 & 28 & -8 & 1 \\ 20 & -16 & -16 & 16 & -4 \\ 18 & 0 & -24 & 0 & 6 \\ 20 & 16 & -16 & -16 & -4 \\ 35 & 56 & 28 & 8 & 1 \end{pmatrix} \text{ et}$$

$$T_{MT}^{-1}.T_{BM}^{-1} = \frac{1}{6}\begin{pmatrix} 6 & 6 & 6 & 6 & 6 \\ -6 & -3 & 0 & 3 & 6 \\ 6 & -6 & -10 & -6 & 6 \\ -6 & 21 & 0 & -21 & 6 \\ 6 & -42 & 70 & -42 & 6 \end{pmatrix}$$

n=5:

$$T_{BM}.T_{MT} = \frac{1}{512}\begin{pmatrix} 126 & -210 & 120 & -45 & 10 & -1 \\ 70 & -70 & -40 & 65 & -30 & 5 \\ 60 & -20 & -80 & 30 & 20 & -10 \\ 60 & 20 & -80 & -30 & 20 & 10 \\ 70 & 70 & -40 & -65 & -30 & -5 \\ 126 & 210 & 120 & 45 & 10 & 1 \end{pmatrix} \text{ et}$$

$$T_{MT}^{-1}.T_{BM}^{-1} = \frac{1}{5}\begin{pmatrix} 5 & 5 & 5 & 5 & 5 & 5 \\ -5 & -3 & -1 & 1 & 3 & 5 \\ 5 & -3 & -7 & -7 & -3 & 5 \\ -5 & 13 & 7 & -7 & -13 & 5 \\ 5 & -27 & 21 & 21 & -27 & 5 \\ -5 & 45 & -105 & 105 & -45 & 5 \end{pmatrix}$$

D. Tableaux de l'étude comparative

Algorithme	FOR		HOS		WW		EC		PRO#1	
n	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur
4	3,30E-04	3,028	8,30E-01	1,1721	1,10E-02	0,6218	5,00E-02	-	2,20E-01	0,5563
5	5,50E-04	16,1595	8,80E-01	6,824	6,60E-02	3,9625	6,00E-02	-	2,20E-01	3,5244
6	8,20E-04	14,3501	9,30E-01	13,9039	1,21E-01	5,6876	1,10E-01	-	3,30E-01	5,0368
7	1,04E-03	49,2448	9,90E-01	23,1321	1,76E-01	9,3302	6,00E-02	-	3,30E-01	8,4796
8	1,38E-03	26,1547	1,10E+00	13,201	2,31E-01	9,2725	1,10E-01	-	3,90E-01	8,3048
9	1,75E-03	51,2956	1,15E+00	16,5937	2,86E-01	9,3857	1,10E-01	-	4,40E-01	8,4351
10	2,19E-03	44,8415	1,26E+00	22,387	3,79E-01	6,5671	1,60E-01	-	4,90E-01	6,0329
11	2,64E-03	69,5007	1,38E+00	29,0721	4,78E-01	13,4318	2,20E-01	-	5,50E-01	11,985
12	3,19E-03	57,2141	1,48E+00	24,7739	6,10E-01	17,1395	2,80E-01	-	6,00E-01	15,1512
13	3,79E-03	55,8928	1,59E+00	25,6031	7,48E-01	18,3782	3,30E-01	-	6,60E-01	16,3303
14	4,39E-03	57,4174	1,65E+00	43,2255	8,85E-01	17,4584	3,90E-01	-	7,10E-01	15,0269
15	5,05E-03	66,63	1,87E+00	29,0044	1,05E+00	19,4277	4,40E-01	-	7,70E-01	17,1337
16	5,72E-03	112,7116	1,98E+00	36,4131	1,22E+00	33,0419	4,90E-01	-	8,20E-01	29,254
17	6,48E-03	92,5339	2,14E+00	47,3828	1,43E+00	33,8071	5,50E-01	-	9,30E-01	29,6043
18	7,25E-03	103,5637	2,25E+00	34,663	1,66E+00	30,1452	6,00E-01	-	9,90E-01	26,8432

Tableau 6 : Réduction du degré n au degré 3

Algorithme	FOR		HOS		WW		EC		PRO#1	
n	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur
6	3,80E-04	11,6514	8,74E+00	1,6556	5,50E-02	0,6262	5,00E-02	-	3,90E-01	1,2959
7	7,20E-04	45,7949	8,46E+00	3,4213	1,10E-01	2,5918	5,00E-02	-	4,40E-01	5,2934
8	1,04E-03	47,4074	8,79E+00	5,5196	1,65E-01	3,3001	1,10E-01	-	5,00E-01	6,4236
9	1,43E-03	42,0998	1,35E+01	14,6847	2,20E-01	1,9707	1,70E-01	-	5,50E-01	4,2389
10	1,87E-03	113,94	9,45E+00	10,0527	3,13E-01	4,6733	1,60E-01	-	6,00E-01	10,1922
11	2,36E-03	130,7944	9,84E+00	9,7595	4,12E-01	3,7077	2,20E-01	-	6,60E-01	9,9731
12	2,80E-03	73,978	1,38E+01	13,786	5,44E-01	4,2776	2,80E-01	-	7,10E-01	9,7396
13	3,40E-03	123,2511	1,37E+01	11,504	6,82E-01	2,8626	3,30E-01	-	7,70E-01	8,9982
14	4,01E-03	188,9369	1,12E+01	19,3746	8,19E-01	6,1171	3,20E-01	-	8,80E-01	14,6537
15	4,67E-03	158,4619	1,49E+01	11,7871	9,89E-01	8,337	4,40E-01	-	8,80E-01	17,9678
16	5,38E-03	88,4432	1,30E+01	28,5009	1,15E+00	4,5647	4,30E-01	-	9,90E-01	8,1727
17	6,16E-03	331,7069	1,36E+01	19,1847	1,36E+00	7,1804	6,10E-01	-	9,90E-01	22,0209
18	6,87E-03	265,2402	1,42E+01	22,1402	1,59E+00	9,3991	6,00E-01	-	1,10E+00	17,9974

Tableau 7 : Réduction du degré n au degré 5

Algorithme	FOR		HOS		WW		EC		PRO#1	
n	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur
8	4,90E-04	4,3451	3,27E+01	0,1091	5,50E-02	0,0592	5,00E-02	0,3332	6,10E-01	0,3476
9	8,80E-04	21,8386	2,86E+01	8,7929	1,10E-01	0,3134	1,10E-01	-	7,20E-01	1,7981
10	1,32E-03	60,4579	3,92E+01	2,0212	2,03E-01	0,5974	1,10E-01	-	7,20E-01	4,4029
11	1,76E-03	129,6834	3,34E+01	2,8634	3,02E-01	1,3402	2,20E-01	-	7,70E-01	9,1212
12	2,31E-03	45,5776	3,59E+01	9,9847	4,34E-01	1,1389	2,70E-01	-	8,80E-01	4,9236
13	2,85E-03	212,3522	3,15E+01	13,9685	5,72E-01	1,5659	2,70E-01	-	8,80E-01	11,8578
14	3,46E-03	133,6944	4,16E+01	34,0664	7,09E-01	2,033	3,20E-01	-	9,40E-01	12,3248
15	4,12E-03	181,3675	4,27E+01	9,8621	8,79E-01	1,5663	3,90E-01	-	9,80E-01	12,4103
16	4,83E-03	140,9005	4,78E+01	12,6409	1,05E+00	1,1261	5,00E-01	-	1,10E+00	9,1065
17	5,60E-03	460,9204	3,51E+01	15,6335	1,26E+00	4,7437	4,90E-01	-	1,15E+00	33,4961
18	6,31E-03	473,2132	4,03E+01	23,9018	1,49E+00	5,2679	6,10E-01	-	1,26E+00	31,4934

Tableau 8 : Réduction du degré n au degré 7

Algorithme	FOR		HOS		WW		EC		PRO#1	
n	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur
4	3,30E-04	3,028	8,30E-01	1,1721	1,10E-02	0,3786	5,00E-02	-	2,20E-01	0,5563
5	3,80E-04	3,6539	8,80E-01	-	5,50E-02	0,7981	5,00E-02	-	1,70E-01	3,6539
6	4,40E-04	11,6514	8,74E+00	1,6556	5,50E-02	0,3642	5,00E-02	-	4,40E-01	1,296
7	4,90E-04	1,3111	8,46E+00	-	5,50E-02	0,0861	5,00E-02	-	2,80E-01	1,3111
8	5,10E-04	4,3451	3,27E+01	0,1091	5,50E-02	0,034	5,00E-02	0,3332	6,10E-01	0,3476
9	5,70E-04	0,5951	2,86E+01	-	5,50E-02	0,0112	5,00E-02	-	1,05E+00	0,088
10	6,00E-04	13,3264	3,92E+01	-	9,30E-02	0,026	5,00E-02	-	1,59E+00	0,1702
11	6,60E-04	1,4081	3,34E+01	-	9,90E-02	0,0074	5,00E-02	-	2,25E+00	0,0417
12	7,10E-04	10,0632	3,59E+01	-	1,32E-01	0,0049	5,00E-02	-	3,02E+00	0,0249
13	7,70E-04	2,1695	3,15E+01	-	1,38E-01	0,0031	5,00E-02	-	4,01E+00	0,0144
14	7,60E-04	11,6417	4,16E+01	-	1,37E-01	0,0014	5,00E-02	-	5,05E+00	6,10E-03
15	8,80E-04	2,3668	4,27E+01	-	1,70E-01	9,00E-04	5,00E-02	-	6,26E+00	3,70E-03
16	8,80E-04	5,3033	4,78E+01	-	1,70E-01	2,00E-04	5,00E-02	-	7,69E+00	6,00E-04
17	9,80E-04	0,6361	3,51E+01	-	2,09E-01	6,00E-05	5,00E-02	-	9,33E+00	2,00E-04
18	9,80E-04	4,2151	4,03E+01	-	2,31E-01	3,00E-05	5,00E-02	-	1,10E+01	1,00E-04

Tableau 9 : Réduction du degré n au degré n-1

Algorithme	FOR		HOS		WW		EC		PRO#1	
n	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur
5	4,90E-04	16,1595	8,80E-01	6,8240	6,60E-02	3,9625	5,00E-02	-	2,20E-01	3,5244
6	6,00E-04	17,5508	8,80E-01	-	1,10E-01	4,3495	5,00E-02	-	4,90E-01	3,7915
7	6,60E-04	45,7949	8,46E+00	3,4213	1,10E-01	2,5918	5,00E-02	-	4,30E-01	3,2935
8	7,70E-04	15,5946	8,46E+00	-	1,10E-01	1,5497	5,00E-02	-	7,70E-01	2,7279
9	8,80E-04	21,8386	2,86E+01	8,7929	1,10E-01	0,3134	5,00E-02	-	6,60E-01	1,6962
10	9,90E-04	13,8416	3,92E+01	-	1,48E-01	0,1366	6,00E-02	-	1,15E+00	0,564
11	1,04E-03	36,6781	3,34E+01	-	1,92E-01	0,1367	1,10E-01	-	1,70E+00	0,49
12	1,15E-03	12,7476	3,59E+01	-	2,31E-01	0,0751	1,10E-01	-	2,42E+00	0,2322
13	1,26E-03	47,9398	3,15E+01	-	2,70E-01	0,0454	1,10E-01	-	3,24E+00	0,1302
14	1,38E-03	12,0433	4,16E+01	-	2,75E-01	9,90E-03	1,10E-01	-	4,18E+00	0,0255
15	1,42E-03	20,7478	4,27E+01	-	3,07E-01	6,20E-03	1,10E-01	-	5,33E+00	0,0143
16	1,54E-03	5,9229	4,78E+01	-	3,40E-01	2,20E-03	1,10E-01	-	6,59E+00	4,80E-03
17	1,65E-03	27,9518	3,51E+01	-	3,79E-01	1,80E-03	1,10E-01	-	8,07E+00	3,60E-03
18	1,76E-03	5,6448	4,03E+01	-	4,40E-01	6,60E-03	1,10E-01	-	9,78E+00	1,20E-03

Tableau 10 : Réduction du degré n au degré n-2

Algorithme	FOR		HOS		WW		EC		PRO#1	
n	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur	Temps	Erreur
6	7,70E-04	14,3501	9,30E-01	13,9039	1,21E-01	5,7732	5,00E-02	-	3,30E-01	5,0368
7	9,30E-04	45,7787	9,90E-01	-	1,65E-01	3,7496	5,00E-02	-	5,50E-01	3,267
8	1,05E-03	47,4074	8,79E+00	5,5196	1,65E-01	3,3001	6,00E-02	-	5,00E-01	6,4236
9	1,21E-03	21,8415	1,35E+01	-	1,65E-01	0,8742	1,10E-01	-	8,30E-01	1,541
10	1,27E-03	60,4579	3,92E+01	2,0212	2,03E-01	0,6824	1,60E-01	-	7,20E-01	4,4029
11	1,48E-03	37,4166	3,34E+01	-	2,47E-01	0,333	1,60E-01	-	1,21E+00	1,2501
12	1,60E-03	45,8296	3,59E+01	-	3,24E-01	0,1811	2,20E-01	-	1,81E+00	0,6489
13	1,76E-03	50,739	3,15E+01	-	3,69E-01	0,1181	1,70E-01	-	2,58E+00	0,3828
14	1,87E-03	41,0837	4,16E+01	-	4,07E-01	0,024	1,70E-01	-	3,46E+00	0,071
15	2,03E-03	32,7609	4,27E+01	-	4,45E-01	0,0371	1,70E-01	-	4,44E+00	0,1144
16	2,14E-03	15,7108	4,78E+01	-	4,77E-01	8,00E-03	1,70E-01	-	5,60E+00	0,0176
17	2,31E-03	27,9819	3,51E+01	-	5,49E-01	2,70E-03	1,70E-01	-	7,03E+00	5,40E-03
18	2,47E-03	30,2161	4,03E+01	-	6,10E-01	6,80E-03	2,20E-01	-	8,51E+00	3,20E-03

Tableau 11 : Réduction du degré n au degré n-3