



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITE M'HAMED BOUGARA BOUMERDES

Faculté des Sciences de l'Ingénieur

Département Génie Mécanique

GRUPE MODELISATION EN MECANIQUE ET PRODUCTIQUE – LMMC

MEMOIRE

Présenté par

M. MAHIDDINI Brahim

en vue de l'obtention du diplôme de

MAGISTER en GENIE MECANIQUE - PRODUCTIQUE

OPTION : SYSTEME DE PRODUCTION

**Echange des données entre systèmes de CFAO :
Application à la conversion par approximation des
courbes et surfaces de Bézier de degré élevé**

Soutenu le 30/10/2006 devant le jury suivant :

Pr. N. NECIB	MC EMP	Président
Dr. I. BELAIDI	CC UMBB	Encadreur
Dr. K. MOHAMMEDI	MC UMBB	Examineur
Dr A. ZERIZER	Professeur UMBB	Examineur
Dr. N. HANOUNE	MC USTHB	Examineur

Année universitaire 2005/2006

Remerciement

Je remercie le Dr. I.Belaidi pour avoir proposé ce sujet et accepté de m'encadrer. Ses conseils, ses directives et son soutien aux moments difficiles, m'ont toujours été d'un très grand appui.

J'ai le plaisir d'exprimer ma reconnaissance à tout le groupe de recherche de CFAO, sous la direction du Dr.K.Mohammed.

Mes vifs remerciements vont aussi à ma promotion et à tous mes amis et collègues.

Dédicace

*A mon père, ma mère,
ma fiancé,
mes sœur,
mes frères
et tous mes proches.*

Notations

$B_{i,n}(t)$	Fonction de Bernstein non rationnelle de degré n et d'indice i .
$G_{i,n}(t)$	Fonction de Bernstein rationnelle de degré n et d'indice i .
$U_i(t)$	Polynôme de Tchebychev de degré i .
C^α	Continuité géométrique d'ordre α .
t	Paramètre curviligne d'une courbe.
u, v	Deux paramètres curvilignes d'une surface.
P_i, Q_i	Pôles d'une courbe de Bézier.
$P(t)$	Courbe de Bézier.
w_i	Poids d'une courbe de Bézier rationnelle.
$S(u, v)$	Surface de Bézier.
$S_{i,j}$	Réseau caractéristique d'une surface de Bézier.

Liste des abréviations

SGDT	Système de Gestion de Données Techniques.
STEP	Standard for Exchange of Product data model.
API	Access Programming Interface.
CAO	Conception Assistée par Ordinateur.
CFAO	Conception et Fabrication Assistée par Ordinateur.
NURBS	Non Uniform Rational B-Spline.
FAO	Fabrication Assistée par Ordinateur.
IGES	Initial graphical Exchange System.
NBS	National Bureau of Standard.
SET	Standard d'Echange et de Transfert.
VDA-FS	Verband der Automobilindustrie Flaechen-Schnittstelle- Grafikstandard.
XAO	Désigne tout ce qui peut être assisté par ordinateur (AO).
GOSET	Groupe opérationnel pour but de développer la norme SET.

SGBD	Système de Gestion de Bases de Données.
SDAI	Standard Data Access Interface.
OpenGL	Open Graphic library.

Liste des figures et des tableaux

ChapitreI: Introduction à l'échange de données en CFAO

Figure I.1 Echange de données entre deux systèmes [19].....	06
Figure I.2 Le partage de données entre les systèmes [19].....	07
FigureI.3 Echange les données entre les systèmes de CAO par des interfaces directes [05]	08
FigureI.4 Echange les données entre les systèmes de CAO par des formats neutres [05].....	08
Figure I.5 Structure de la norme STEP (Source : [23]).....	11
Figure I.6 L'architecture de STEP(source :[31]).....	12
Figure I.7 Un exemple de diagramme EXPRESS-G (Source : [13]).....	15
Figure I.8 Exemple de schéma EXPRESS et du fichier neutre associé (Source : [13]).....	16
Figure I.9 Format d'un fichier STEP en langage EXPRESS[05].....	17
Figure I.10 Echange de données dans STEP [12].....	18
FigureI.11Les différentes étapes d'analyse d'un fichier d'échange STEP [12].....	19
Figure I.12 Schéma de transfert entre deux systèmes interactifs de CFAO.	21

ChapitreII: Méthode de réductions de degré de courbes et surfaces de Bézier

Figure II.1 Fonction de Bernstein pour $n = 6$	25
Figure II.2 La courbe de Bézier et son polygone caractéristique selon P. Bézier.....	26
Figure II.3 Surface de Bézier non rationnelle.....	30
Figure II.4 Organigramme général du principe de la conversion des courbes de Bézier par approximation [47].....	36

ChapitreIII: Programme de conversion par approximation implémenté

Figure III.1 Organigramme général de fonctionnement du logiciel élaboré.....	53
Figure III.2 Courbe de Bézier de degré 9 réduite en courbe de degré 6 avec une continuité $\alpha = 3$	63

Figure III.3	Variation de l'erreur de position de la courbe de la figure III.2.....	63
Figure III.4	Variation de la courbure de la courbe de la figure III.2.....	64
Figure III.5	Courbe de Bézier de degré 10 réduite en courbe de degré 8 avec une continuité $\alpha = 3$	64
Figure III.6	Variation de l'erreur de position de la courbe de la figure III.5.....	65
Figure III.7	Variation de courbure de la courbe de la figure III.5.....	65
Figure III.8	Courbe de Bézier de degré 10 convertit en 2 courbes de degré 8 avec une continuité $\alpha = 3$ et $\varepsilon = 0.01mm$	66
Figure III.9	Variation de l'erreur de position de la figure de la position III.8.....	66
Figure III.10	Variation de la courbure de la courbe de la figure III.8.....	67
Figure III.11	Courbe de Bézier de degré 10 convertit en 2 courbes de degré 8 avec une continuité $\alpha = 3$ et $\varepsilon = 0.01mm$	67
Figure III.12	Variation de l'erreur de position de la courbe de la figure III.11.....	68
Figure III.13	Variation de la courbure de la courbe de la figure III.11.....	68
Figure III.14	Surface de Bézier de degré (9,9)	69
Figure III.15	Les 4 surfaces de Bézier de degré (7,7) après la conversion.....	69
Figure III.16	Variation de la courbure gaussienne de la surface de la figure III.14.....	70
Figure III.17	Variation de la courbure moyenne de la surface de la figure III.14.....	70
Figure III.18	Surface de Bézier de degré (4,4).....	70
Figure III.19	Les 10 surfaces de Bézier de degré (3,3) après la conversion.....	71
Figure III.20	Variation de la courbure gaussienne de la surface de la figure III.18.....	71
Figure III.21	Variation de la courbure moyenne de la surface de la figure III.18...	71
Figure III.22	Réseau caractéristique de la surface de la figure III.18 après la modification locale.....	72
Figure III.23	Surface de Bézier de degré (4,4).....	72
Figure III.24	Les surfaces de Bézier de degré (3,3) après la conversion.....	73
Figure III.25	Variation de la courbure gaussienne de la surface de la figure III.23.....	73
Figure III.26	Variation de la courbure moyenne de la surface de la figure III.23.....	73

ChapitreIV: Résultats et interprétations

Figure IV.1	Courbe de Bézier de degré 18.....	76
Tableau IV.1	Réduction du degré n au degré $n - k$	77

Figure IV.2	Variation de nombre des courbes en fonction du degré n	77
Figure IV.3	Variation de Temps de calcul en fonction du degré n	77
Figure IV.4	Variation d'erreur de position en fonction du degré n	78
Figure IV.5	Variation de courbure en fonction du degré n	78
Tableau IV.2	Réduction du degré n au degré $n - 3$	79
Figure IV.6	Variation de nombre des courbes en fonction du degré n	79
Figure IV.7	Variation de temps de calcul en fonction du degré n	80
Figure IV.8	Variation d'erreur de position en fonction du degré n	80
Figure IV.9	Variation de courbure en fonction du degré n	80
Figure IV.10	Courbe de Bézier de degré 10 réduite en deux courbes de degré 7 avec $\alpha = 3$ et $\varepsilon = 0.01mm$	81
Figure IV.11	Courbe de Bézier de degré 10 réduite en deux courbes de degré 7 avec $\alpha = 3$ et $\varepsilon = 0.01mm$ après modification locale.....	82
Figure IV.12	Variation de l'erreur de position de la courbe de la figure IV.10.....	82
Figure IV.13	Variation de l'erreur de position de la courbe de la figure IV.11.....	83
Figure IV.14	Variation de courbure de la courbe de la figure IV.10	83
Figure IV.15	Variation de courbure de la courbe de la figure IV.11.....	84

Annexe A

Figure A.1	Interface pour dessiner la courbe de Bézier.....	II
Figure A.2	Fenêtre de choix entre les deux méthodes.....	III
Figure A.3	Variation de l'erreur de position et la courbure de la courbe précédente.....	III
Figure A.4	Lecteur du réseau caractéristique à partir d'un fichier Txt.....	IV
Figure A.5	Surface de Bézier de degré (8,8).....	IV
Figure A.6	Les surfaces de Bézier de degré (6,6) après la Sub-réduction.....	V
Figure A.7	Les deux courbures (Gaussienne et Moyenne) pour les 4 surfaces obtenues.....	V
Figure A.8	Historique des normes d'échanges de données de produit[24].....	VI

SOMMAIRE

Introduction générale.....01

ChapitreI: Introduction à l'échange de données en CFAO

I.1 Données techniques de produits.....	04
I.1.1 Eléments géométriques simples.....	04
I.1.2 Eléments surfaciques.....	05
I.1.3 Eléments volumiques.....	05
I.2 Echange et partage de données techniques.....	06
I.2.1 Echange de données.....	06
I.2.2 Partage de données.....	06
I.3 Différentes méthodes de transfert des données.....	07
I.3.1 Méthode d'interface directe.....	07
I.3.2 Méthode de formats neutres.....	08
I.4 Développement de standards d'échange.....	08
I.4.1 La norme internationale STEP.....	09
I.4.1.1 Structure de STEP.....	10
I.4.1.2 Interface standard d'accès aux données : SDAI.....	14
I.4.1.3 Echange de données dans STEP.....	17
I.4.2 La norme IGES.....	19
I.4.2.1 Formats d'un fichier IGES.....	20
I.4.2.2 Les transferts en lecture et écriture.....	21
I.4.2.3 Possibilités de transfert.....	22
I.4.3 La Norme SET.....	23
I.4.4 La Norme VDA-FS.....	23
I.4.5 Standard DXF.....	23
I.5 Conclusion.....	23

ChapitreII: Méthode de réductions de degré de courbes et surfaces de Bézier

II.1 Courbes et surfaces de Bézier.....	25
II.1.1 Polynôme de Bernstein.....	25
II.1.2 Courbe de Bézier non rationnelle.....	26
II.1.2.1 Evaluation d'un point de la courbe de Bézier.....	27

II.1.2.2	Forme matricielle d'une courbe de Bézier.....	27
II.1.2.3	Dérivée d'une courbe de Bézier.....	27
II.1.2.4	Hodographe d'une courbe de Bézier.....	28
II.1.3	Courbe de Bézier rationnelle.....	28
II.1.4	Subdivision d'une courbe de Bézier.....	28
II.1.5	Élévation de degré des courbes de Bézier.....	29
II.1.6	Surfaces de Bézier non rationnelles.....	30
II.1.6.1	Évaluation d'un point de la surface de Bézier.....	31
II.1.6.2	Forme matricielle d'une surface de Bézier.....	31
II.1.6.3	Dérivée d'une surface de Bézier.....	32
II.1.7	Surfaces de Bézier rationnelles.....	32
II.1.8	Courbure gaussienne et courbure moyenne.....	33
II.1.9	Subdivision d'une surface de Bézier.....	34
II.1.10	Élévation de degré des surfaces de Bézier.....	34
II.2	Réduction de degré des courbes de Bézier.....	34
II.3	Conversion des courbes de Bézier non rationnelles.....	36
II.3.1	Principe général de la conversion.....	36
II.3.2	Méthodes de réduction de degré.....	37
II.3.2.1	Méthode de Forrest.....	37
II.3.2.2	Méthode de Hoschek.....	37
II.3.2.3	Méthode de séparation des variables	38
II.3.2.3.1	Réduction de degré à (n-1).....	38
II.3.2.3.2	Détermination des pôles influant sur la continuité.....	38
II.3.2.3.3	Détermination des pôles intermédiaires.....	38
II.3.2.4	Méthode de Wathins et Worsey.....	41
II.3.2.5	Méthode de Eck.....	42
II.3.2.6	Méthode d'approximation L^1 par l'interpolation des points Extrêmes.....	43
II.3.2.7	Méthode de transformation entre la base de Legendre et la base de Bernstein pour élévation et réduction de degré.....	45
II.3.2.8	Méthode de représentation matricielle unifiée pour réduction des courbes de Bézier.....	46
II.3.2.9	Utilisation des polynômes de Jacobi pour la réduction de degré des courbes de Bézier avec des contraintes de continuité C^α	47
II.3.2.10	Multi réduction optimale de degré des courbes de Bézier avec des contraintes de continuité des points extrêmes.....	49
II.4	Conclusion.....	51

Chapitre III: Programme de conversion par approximation Implémenté

III.1 Architecture générale.....	52
III.2 Algorithmiques pour la sub-réduction de courbes mise en œuvre.....	53
III.2.1 Procédures algorithmique de la méthode de Eck.....	54
III.2.2 Procédures algorithmique de la méthode de séparation des variables...56	
III.3 Méthode de sub-réduction de surfaces mise en œuvre.....	59
III.3.1 Réduction de surface de Bézier.....	59
III.3.2 Algorithme de Sub-réduction de la surface.....	59
III.4 Mise en œuvre du programme implémente.....	63
III.4.1 Cas des courbes.....	63
III.4.1.1 Réduction de courbes de Bézier non rationnelles.....	63
III.4.1.2 Sub-réduction de courbes de Bézier non rationnelles.....	66
III.4.2 Cas des surfaces.....	69
III.5 Modification locale de la surface de Bézier.....	72
III.6 Conclusion.....	74

Chapitre IV: Résultats et interprétations

IV.1 Etude de l'influence du facteur de réduction.....	75
IV.2 Etude influence du degré de la courbe.....	77
IV.3 Influence de pseudo-modification locale.....	80
IV.4 Interprétation des résultats.....	83

Conclusion générale.....84

Références bibliographiques.....i

Annexe A.....I

Introduction Générale

Introduction Générale

La CFAO a débuté à la fin des années soixante avec la modélisation des surfaces complexes ou formes libres. Tous les grands constructeurs automobiles et aérospatiaux ont développé, à cette époque, des logiciels internes de modélisation de surfaces dans le but:

- De faire de la simulation d'usinage, c'est-à-dire, générer la trajectoire des outils d'usinage de la surface (code commande numérique CN d'une machine-outil), d'estampage, de montage,
- D'effectuer des calculs d'aérodynamique,
- De juger de l'esthétique à partir d'images de synthèse.

Les modèles paramétriques sont la base de nombreux logiciels de CAO. Parmi les modèles les plus utilisés, nous distinguons les modèles de Coons, les modèles de Bézier, les B-splines et les NURBS. De tous les modèles polynomiales actuellement utilisées dans les systèmes de CAO, les courbes de Bézier sont les plus stables du point de vue numérique, malgré leur manque de flexibilité dans la conception de forme, par rapport aux modèles rationnels et aux modèles NURBS et B-spline.

Les contributions de Ferguson, de Bézier et de De Casteljau dans les années 1960 se sont attaché plus à mettre en œuvre des modèles de description des courbes et des surfaces dont le comportement serait contrôlé à partir des grandeurs géométriques simples et faciles d'accès au personnel de bureau d'études, ce qui a conduit à l'introduction de la notion de polygone et de réseau caractéristiques associés à une courbe et respectivement à une surface.

Pour une meilleure intégration industrielle dans un contexte d'ingénieries virtuelle et concourante, les entreprises ont dû faire appel à des techniques de communication entre leurs différents systèmes d'applications, telles que les échanges de données et le partage d'informations par interfaces graphiques spécifiques, accompagnés de la mise en place d'un modèle de référence fédérateur, supporté par le **S**ystème de **G**estion des **D**onnées **T**echniques (SGDT). Diverses actions, notamment normatives, ont été menées dans le domaine des échanges de données, partage et intégration de données techniques depuis les années 1980.

Ainsi, les premières normes proposées ciblaient des domaines d'application particuliers à l'instar de la norme allemande *VDA/FS : échange de surfaces de Bézier*, dans l'industrie automobile [VDA, 86], ou des méthodologies particulières telle que la norme américaine *IGES* [IGES, 80] créée au départ pour l'échange de données géométriques. Par la suite, au début des années 1990, l'ensemble de ces actions normatives ont été unifiées dans une norme internationale de l'ISO : ISO 10303 [ISO, 94a], connue sous le nom de la '**S**tandard for **E**xchange of **P**roduct **M**odel **D**ata', notée '*STEP*'.

Les nombreux travaux de conversion par approximation des courbes et surfaces de Bézier de degré élevé [32], [33], [07], [44], [20], [25], [17], [18], [38], [39],[14],[48],[01] ont permis d'engager une réflexion dans la perspective de leur intégration dans les systèmes de CFAO visant la simplification de la complexité géométrique et l'augmentation de la flexibilité des modèles de Bézier lors de l'étape de conception (modèles à degré élevé), mais également pour faciliter et uniformiser les échanges de données entre les systèmes de CFAO à base de modèles différents ou utilisant des modèles identiques mais de degré différents et pour une meilleure compatibilité 'outil-pièce' et une bonne stabilité numérique lors de la génération de trajectoires d'usinage, basée sur les modèles de Bézier à faible degré.

Dans le contexte de notre travail, nous proposons une approche algorithmique pour la conversion par approximation des courbes et surfaces de Bézier de degré élevé. Ainsi, nous proposons les développements et la mise en œuvre informatique de deux méthodes comparées, à savoir la méthode de réduction de degré de courbes et surfaces de Bézier non rationnelles par séparation de variables et minimisation de l'erreur au sens des moindres carrés (99) et la méthode de réduction de degré de courbes et surfaces de Bézier par application algorithme inverse d'élévation de degré et minimisation de l'erreur au sens des polynômes orthogonaux [17].

A cet effet, nous avons structuré notre présent mémoire en cinq chapitres :

Le premier chapitre est consacré à une présentation générale des standards d'échange entre les systèmes de CFAO, et des produits traités dans les processus de transformation.

Dans le deuxième chapitre, nous présentons une synthèse bibliographique des méthodes de conversion par approximation des courbes et surfaces de Bézier non rationnelles, publiées durant ces dernière décennies.

Le chapitre trois traite des développements algorithmiques mis en oeuvre pour la réduction de degré et la subdivision des éléments (courbes et surfaces) de Bézier non rationnels, et la représentation détaillée des programmes informatiques implémentés dans le langage VisualC++.

De nombreux exemples de conversion par approximation d'éléments sont élaborés dans le quatrième, en utilisant les deux méthodes implémentées. Une interprétation et une discussion des résultats comparés clôturent ce chapitre.

Une conclusion générale et perspectives parachèvent ce présent mémoire.

***Chapitre I : Introduction
à l'échange de donnée en
CFAO***

Introduction

Le développement des produits passe par plusieurs phases et nécessite l'intervention de plusieurs métiers. La perception du produit dans chaque phase de son cycle de vie est différente. Les informations sur les produits sont représentées sous différents formats selon les outils informatiques que l'on utilise lors de chaque phase. Elles permettent, entre autres, de répondre à certaines questions que l'on se pose sur le produit (quelle est la forme du produit, quel est le modèle de calcul de sa structure, quelles sont les étapes de sa fabrication, ...). Chaque vue de produit est, en effet, perçue à travers un certain nombre de représentations. Chacune de ces représentations est créée en utilisant des outils spécifiques.

I.1 Données techniques de produits

Les données techniques de produits sont les données produites par les systèmes informatiques spécifiques des études, de l'ingénierie, de la fabrication, de la gestion des projets, de la gestion de qualité etc... Elles présentent la caractéristique très particulière d'être chacune associée à un outil particulier (exemple : système CAO, FAO, code de calcul éléments finis, ...). Les données techniques d'un produit peuvent être définies de la manière suivante : il s'agit d'une structure arborescente représentant la composition d'un produit et servant de support à l'organisation de toutes les données de produit [19].

I.1.1 Eléments géométriques simples

Les courbes ou plus particulièrement les arcs de courbes, sont très utilisés en CAO. En raison de leur nombre de degrés de liberté théoriquement infini, ces entités seront surtout utilisées par des designers à des fines esthétiques, mais elles peuvent aussi représenter des entités fonctionnelles.

Les courbes sont modélisées par des courbes paramétriques de degrés finis. Les modèles couramment employés sont les modèles de la famille des NURBS tels que les courbes de Bézières.

Ces modèles sont utilisés en raison de leurs propriétés géométriques simples exploitables par l'utilisateur. En effet, le comportement de ces courbes est contrôlé à partir de grandeurs géométriques faciles d'accès comme le *polygone caractéristique*.

I.1.2 Eléments surfaciques

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

- Le modèle doit appartenir à une classe de fonctions continues, suffisamment dérivables pour autoriser une manipulation mathématique.
- Le modèle doit être facile à manipuler et à stocker dans un ordinateur.
- L'évaluation d'un point courant par ce modèle doit se faire avec un nombre restreint d'opérations.
- Le modèle doit permettre de mettre au point des algorithmes d'intersection à coût de calcul faible.
- La manipulation des paramètres du modèle par le biais d'une interface doit pouvoir se faire de façon intuitive.
- Le modèle doit permettre de couvrir une grande variété de formes.

I.1.3 Eléments volumiques

Le modèle surfacique permet de définir une forme par un ensemble de surfaces, mais ces surfaces restent indépendantes les unes des autres comme des *peaux* infiniment fines dont on ne verrait pas qu'elles délimitent un volume. Ainsi, le modèle surfacique ne permet pas de distinguer de notion d'intérieur et d'extérieur ; il n'y a donc, par conséquent, pas de matière et on ne peut pas encore :

- Effectuer un calcul de masse.
- Déterminer le centre et la matrice d'inertie nécessaires à une simulation dynamique multi-corps.
- Effectuer un calcul de structures nécessitant de passer par le maillage du volume.
- Bénéficier de la fonctionnalité si importante de détection d'interpénétration de pièces lors d'une simulation cinématique.
- Générer des plans 2D avec des hachures automatiques.

I.2 Echange et partage de données techniques

La représentation des données CAO, par exemple, diffère d'un système à un autre. L'échange de données entre différents systèmes pose des problèmes de compatibilité à différents niveaux: géométrie, sémantiques, ... [12]. Cet échange devient d'autant plus complexe que les modèles mathématiques utilisés pour représenter des entités géométriques sont différents.

I.2.1 Echange de données

L'échange de données consiste à échanger des fichiers contenant des données codées selon différents formats. Souvent, les sous-traitants sont obligés d'acquiescer le même système que leurs donneurs d'ordre uniquement pour l'utiliser comme un système intermédiaire, pour pouvoir passer ensuite à leurs propres systèmes. L'utilisation des données d'un système dans un autre nécessite un passage par un pré-processeur qui joue le rôle du traducteur. C'est la raison pour laquelle des normes d'échange ont été développées.

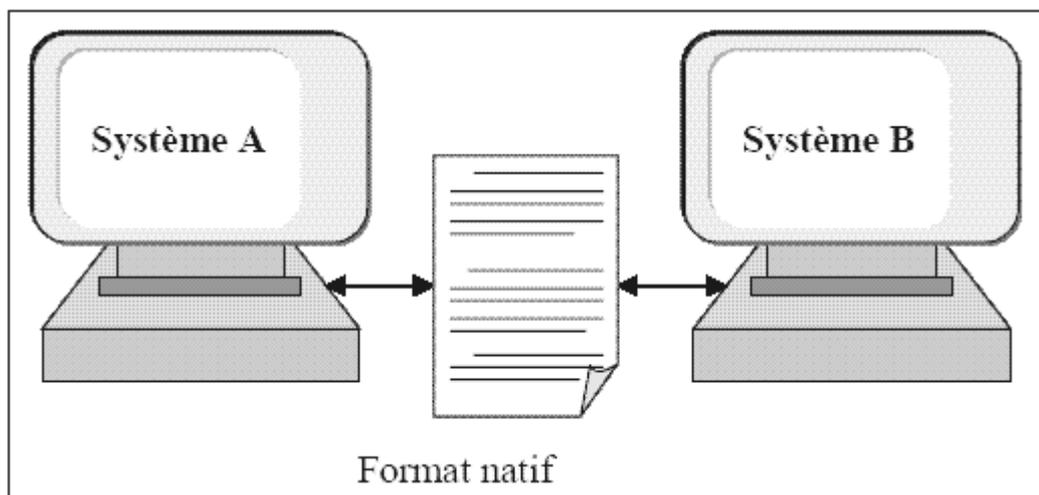


Figure I.1 Echange de données entre deux systèmes [19].

I.2.2 Partage de données

Le stockage des données dans une base commune sous un format permettant de pouvoir les utiliser sur différents systèmes. Cela permet de faciliter l'implémentation des systèmes qui gèrent les accès aux données dans un contexte d'intervenants multiples. De plus, il conduit à une fédération des données afin

qu'elles soient disponibles pour chacun de ces intervenants et sans se soucier des applications qu'ils utilisent.

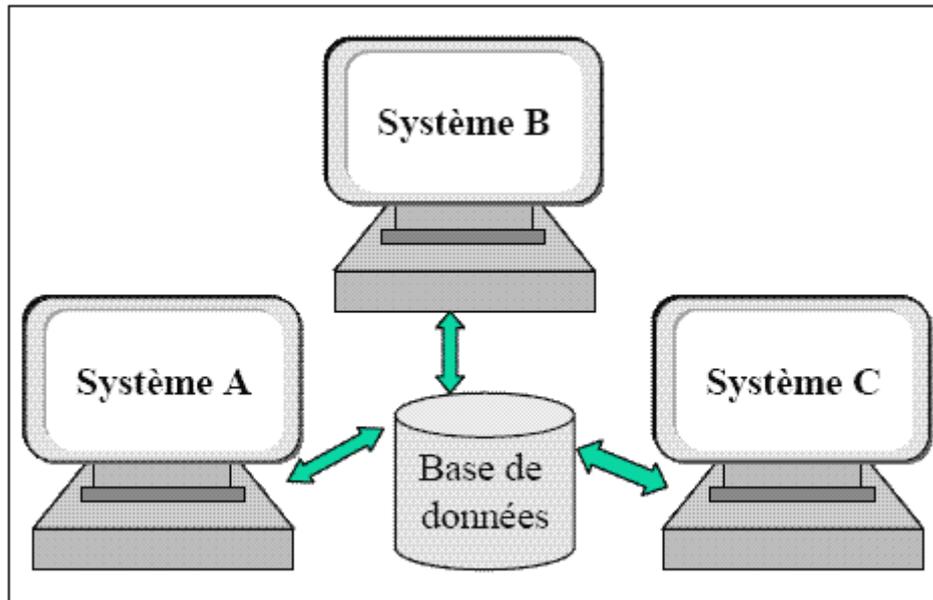


Figure I.2 Le partage de données entre les systèmes [19].

I.3 Différentes méthodes de transfert des données

I.3.1 Méthode d'interface directe

Dédiées à un couple de logiciels seulement (ex: AB, BC, AC), parmi les avantages et les inconvénients de cette méthode, nous distinguons :

Avantage :

-La meilleure adaptation possible entre les deux logiciels.

Inconvénients:

-Nécessitent une mise à jour à chaque fois qu'un des deux logiciels est mis à jour.

-Implique une interface directe pour chaque logiciel avec lequel des échanges sont envisagés.

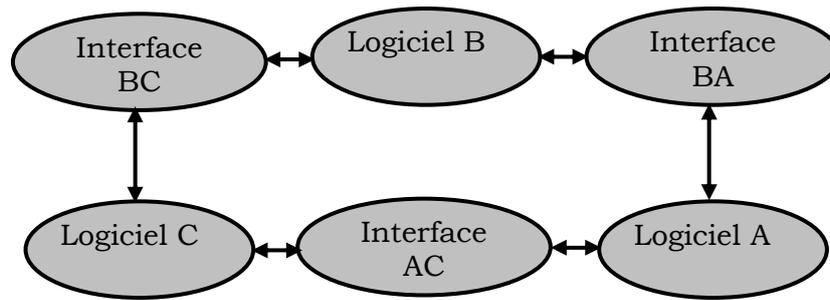


Figure I.3 Echange des données entre les systèmes de CAO par des interfaces directes [05].

Remarque : Si n systèmes alors, nous avons $n(n-1)$ interfaces requises.

I.3.2 Méthode de formats neutres

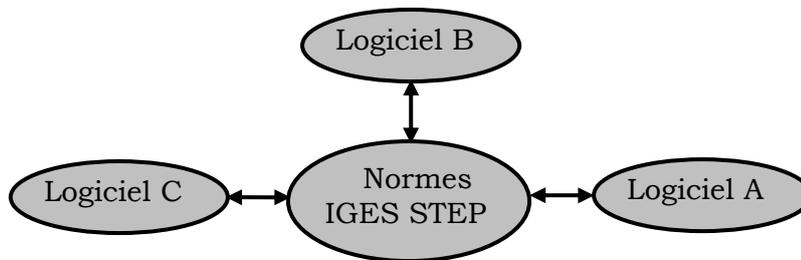


Figure I.4 Echange des données entre les systèmes de CAO par des formats neutres [05].

Avantage:

1. Permet l'échange de données avec tous les logiciels munis de la même interface normalisée.

Inconvénients:

1. Les spécificités des logiciels qui ne sont pas supportés par la norme ne pourront être transférées.

Remarque : Si n systèmes alors, nous avons $2n$ interfaces requises.

I.4 Développement de standards d'échange

La recherche d'un format commun pour les données issues de la XAO est un souci récurrent des industriels dans le monde entier depuis les années 70. La mise en œuvre d'échange et de partage des données a amené les industriels, surtout

ceux qui utilisent beaucoup la sous-traitance (l'industrie automobile, aérospatiale), à créer des standards.

Ces standards ont beaucoup servi aux différents partenaires pour mener à bien leurs projets. La plupart d'entre eux sont des standards nationaux; on peut citer la norme SET (Standard d'Echange et de Transfert), IGES (Initial graphical Exchange System), la norme allemande VDA-FS, ...[19].

1.4.1 La norme internationale STEP

Les normes nationales et standard industriel ont permis la résolution de certains problèmes d'échange et de partage de données. Toutefois, la création d'une norme internationale plus élaborée s'avérait intéressante. La communauté internationale voulait regrouper dans cette norme tous les avantages des standards nationaux (IGES, PDDI, SET et VDA-FS) tout en palliant les inconvénients et limitations de ces derniers. Elle a été conçue de telle manière à être facilement extensible et réutilisable. Après un certain nombre de rencontres des acteurs internationaux, un consensus a été établi sur les grandes lignes de ce standard. L'objectif principal de ce standard est de représenter, sans aucune ambiguïté les données liées à l'ingénierie et aux produits, dans un format neutre, interprétable par tout système informatique, sans perdre cependant l'intégrité des données durant tout le cycle de vie d'un produit [37] [46]. Ce standard international est connu sous l'acronyme STEP (Standard for Exchange of Product model data) et est enregistré au sein de l'Organisation Internationale de Normalisation ISO sous la référence ISO 10303 [ISO 10303-1:1994].

Pour atteindre son objectif, STEP a :

- développé une nouvelle méthodologie de modélisation de données, pour assurer leur indépendance de tout système informatique. Cette approche comporte [46]:

1. la définition d'un langage de spécification de données, le langage EXPRESS [ISO 10303-11:1994].
2. la définition d'un format neutre d'échange et de stockage des données décrites dans ce langage [ISO 10303-21:1994].

- établi des procédures permettant aux experts de chaque domaine technique de définir des modèles propres à leur domaine de spécialisation. Un certain nombre de modèles relevant d'aspects communs ont été également définis. Ainsi, plusieurs protocoles d'application, basés sur le langage EXPRESS, ont été définis au sein de STEP. Un protocole d'application définit toutes les données requises dans un domaine d'application particulier. L'approche suivie pour l'élaboration d'un protocole d'application permet ainsi que celui-ci soit une véritable capitalisation du

savoir-faire des experts du domaine d'application visé. Il décrit les informations pertinentes dans un domaine technique donné ainsi que la structuration de ces données. Il spécifie également les différents sous-ensembles implantables « classe de conformité » ainsi que la manière de les mettre en oeuvre. Pour être conformes à STEP, les systèmes logiciels doivent être en mesure d'interpréter toutes les données définies dans une classe de conformité d'un protocole d'application. Cela assure que toutes les données peuvent être traduites à l'entrée ou à la sortie de cette structure sans aucune perte.

I.4.1.1 Structure de STEP

L'architecture de STEP s'appuie sur un ensemble de composantes décrivant, soit une méthode, soit une ressource, définies chacune par un ou plusieurs documents standard. Ces différents composants sont :

- La méthode de description constituée du langage de modélisation EXPRESS.
- La méthode d'implémentation, constituée du format d'échange et de l'interface de manipulation des données STEP.
- Les ressources intégrées, composées des ressources générales (*Generic Resources*) indépendantes de tout contexte, et des ressources plus spécifiques à un groupe de domaines d'applications (*Application Resources*).
- Une liste, qui croît au fil du temps, de ce qu'on appelle les protocoles d'application, qui sont des cas d'utilisation des ressources intégrées pour des applications spécifiques.
- La méthode de tests de conformité des implantations des protocoles d'application.

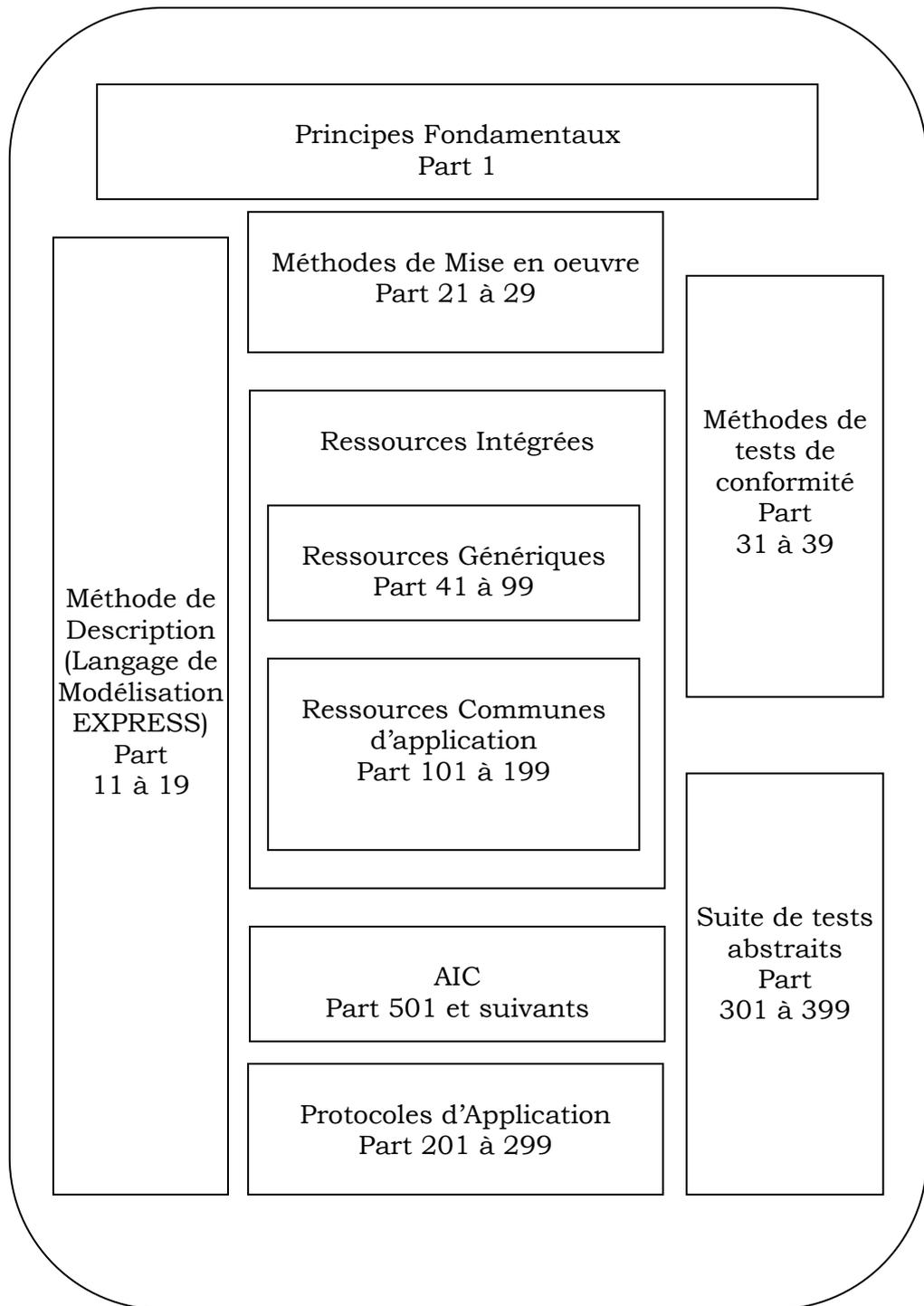


Figure I.5 Structure de la norme STEP (Source : [23])

Par ailleurs, elle est basée sur la structure à trois niveaux d'abstraction (externe, conceptuel, interne) préconisée dans le rapport de standardisation des architectures des SGBD [ANSI75], comme l'indique la figure ci-dessous :

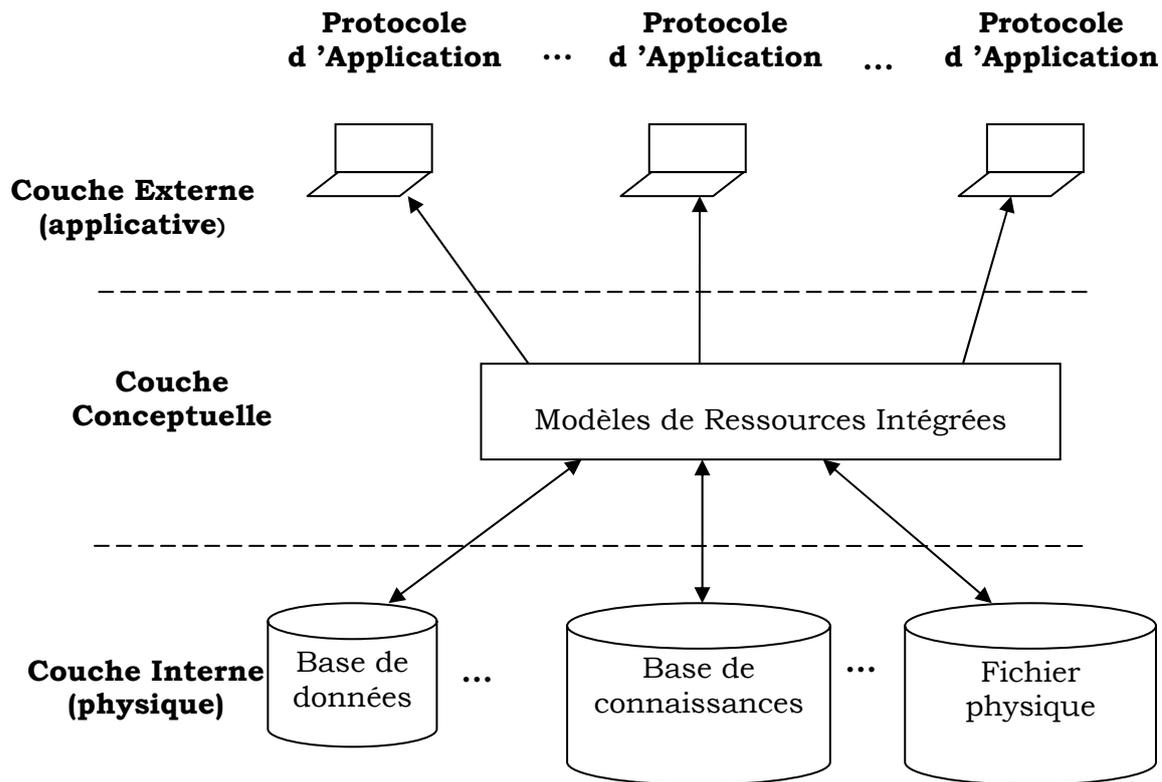


Figure I.6 L'architecture de STEP (source : [31]).

Le niveau conceptuel

Il est constitué par :

- Ressources Intégrées - IR (part 41 à 99 et 101 à 199). Elles sont composées de Ressources Génériques indépendantes de tout contexte et de Ressources Communes d'application plus spécifiques à un groupe de domaines d'application. L'ensemble constitue une bibliothèque de définitions d'entités qui permettent de construire les descriptions des données d'un produit dans un Protocole d'Application.
- Construction Interprétée par Application - AIC (part 501 et suivants) ce sont des bibliothèques de modèles, complémentaires aux Ressources Intégrées et qui sont utilisables par plusieurs Protocoles d'Application.

Le niveau externe

Il est matérialisé par les Protocoles d'Application - AP (part 201 à 299). Ces AP résultent de la sélection de certaines Ressources Intégrées pour des utilisations

dans un contexte donné.

Chaque AP comprend une expression des besoins informationnels (donc un modèle de données) pour un domaine donné. Un domaine correspond à un type de produit, à certaines phases de son cycle de vie. A ce titre, l'AP 203 se focalise sur la conception 3D de produits mécaniques avec gestion de configuration.

Les modèles informationnels de STEP se localisent dans les niveaux conceptuel et externe, au niveau des Ressources Intégrées, des AIC et des Protocoles d'Application. Ces modèles sont décrits à l'aide des *Méthodes de description* de STEP (part 11 à 19), constitués de langages normalisés pour la représentation de données dont essentiellement le langage de modélisation EXPRESS (norme ISO 10303-11), ce dernier est un langage de modélisation formelle, interprétable informatiquement. Il utilise le formalisme entité-association étendu, en intégrant des concepts pour spécifier la généralisation, l'agrégation et les contraintes. Un des principaux concepts sur lesquels se base ce langage est le concept de schéma.

EXPRESS permet la définition de modèles de données modulaires en partitionnant le modèle complet en schémas. Chaque schéma correspond à un domaine d'intérêt particulier caractérisé par un ensemble d'objets.

Le niveau interne

Est constitué par les Méthodes de Mise en œuvre (part 21 à 29) qui permettent de traduire et de manipuler les données d'un domaine d'application sous forme informatique. Deux formes de mise en œuvre sont disponibles aujourd'hui : l'échange par fichier neutre et le partage au moyen de requêtes normalisées [23].

- Le fichier neutre STEP est analogue aux fichiers d'échange de données classiques. C'est un fichier ASCII contenant des données. La structure des données est définie en EXPRESS. Le format de ce fichier est défini par la partie 21 de STEP. Un fichier STEP contient une section '*header*' qui indique entre autres le nom du schéma utilisé, et une section '*data*' qui contient des données correspondant au schéma.

- Pour le partage de données, le protocole SDAI (partie 22 à 27 de STEP) constitue une interface fonctionnelle ensemble de routines - entre une application et les données qu'elle manipule. La structure de ces données doit être décrite en

EXPRESS. SDAI est indépendant du format de stockage des données. Les parties 22, 23, 24 et 26 de STEP définissent SDAI et sa mise en oeuvre avec un langage de programmation.

Les tests de conformité

Instruites par les difficultés rencontrées dans les interfaces de l'autre génération (telle que IGES) où il était très difficile, en cas d'anomalie dans un transfert de données, d'identifier les responsabilités entre l'émetteur, le receveur, voire les erreurs ou imprécisions de la spécification elle-même, la norme STEP définit pour chaque AP :

- Comment doivent être testées les interfaces ?

Sur quels critères établir leur conformité ?

à l'ISO, des projets de normes définissant les tests de référence sont en cours de développement pour les AP202, 203, 204.

D'autres tests de référence doivent être progressivement développés pour tous les protocoles d'application.

I.4.1.2 Interface standard d'accès aux données : SDAI

Une opération SDAI est caractérisée par une liste de paramètres d'entrée et une autre de sortie. Pour toute opération standard SDAI, on définit son équivalent dans un certain nombre de langages d'implémentation (C, C++, Fortran). S'il n'y a pas homomorphisme entre les opérations SDAI et celles d'un langage d'implémentation, il est souhaitable de définir des extensions dans ce langage, pour qu'en plus des opérations standard, on trouve d'autres fonctionnalités utiles.

Cette interface permet en particulier :

- L'accès et la manipulation par programme d'instances d'entités définies en EXPRESS.
- L'accès simultané à plusieurs bases de données par plusieurs applications.
- L'accès à la définition EXPRESS des éléments de données qui peuvent être manipulés par une application.
- La vérification des contraintes définies dans le modèle de données EXPRESS.

Une notation graphique est associée au langage EXPRESS. Il s'agit de l'EXPRESS-G qui est basée sur le notation IDEF-1X. Elle supporte la notion de schéma et les liens inter-schémas et à un niveau d'abstraction plus bas, les

concepts d'entité, type, attribut, relation et cardinalités. La Figure 1.7 illustre le diagramme EXPRESS-G associé au schéma EXPRESS.

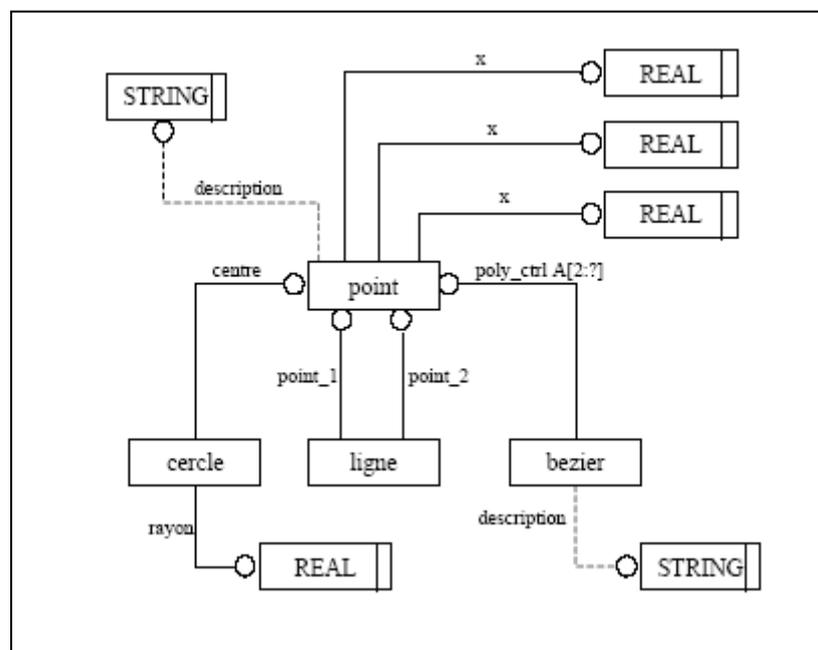


Figure I.7 Un exemple de diagramme EXPRESS-G
(Source : [13])

<pre> SCHEMA Exemple; ENTITY point x : REAL; y : REAL; z : REAL; description: STRING; END_ENTITY; ENTITY ligne point_1 : point; point_2 : point; END_ENTITY ENTITY cercle centre : point; rayon : REAL; DERIVE diametre : REAL := rayon*2; WHERE rayon>0.0; END_ENTITY; ENTITY bezier poly_ctrl : ARRAY [2:?] OF point; description: OPTIONAL STRING; END_ENTITY END_SCHEMA </pre>	<pre> ISO-10303-21; HEADER; FILE_DESCRIPTION ((' ', ' 1 '); FILE_NAME (' unspecified ', ' 1999- 02- 08T15:50:22-0 ', ('), (' ', ' ', 'Fred '); FILE_SCHEMA ((' EXEMPLE ')); ENDSEC ; DATA ; #1 = POINT (0.0, 0.0, 0.0, 'Origine '); #2 = POINT (1.0, 0.0, 0.0, 'Axe X '); #3 = POINT (0.0, 1.0, 0.0, 'Axe Y '); #4 = POINT (0.0, 0.0, 1.0, 'Axe Z '); #5 = POINT (1.0, 1.0, 1.0, 'Centre '); #6 = LIGNE (#1, #2); #6 = LIGNE (#1, #3); #6 = LIGNE (#1, #4); #7 = CERCLE (#5, 1.0); #8 = BEZIER ((#1, #2, #3, #4, #5), \$); END-ISO-10303-21 : </pre>
Un schéma EXPRESS	Exemple de fichier neutre

Figure I.8 Exemple de schéma EXPRESS et du fichier neutre associé (Source : [13]).

Exemple de format du fichier STEP en langage EXPRESS

```

SCHEMA geometrie;
ENTITY point;
x: REAL;
y: REAL;
z: REAL;
END_ENTITY;
ENTITY ligne;
debut: point;
fin : point;
END_ENTITY;
ENTITY cercle_xy;
Centre: point;
Rayon: REAL;
END_ENTITY;
END_SCHEMA;
#10=point (0.,0.,0.);
#11=point (-5.,0.,0.);
#12=point (5.,0.,0.);
#13=ligne (#11,#12);
#14=cercle_xy(#10,2.);

```

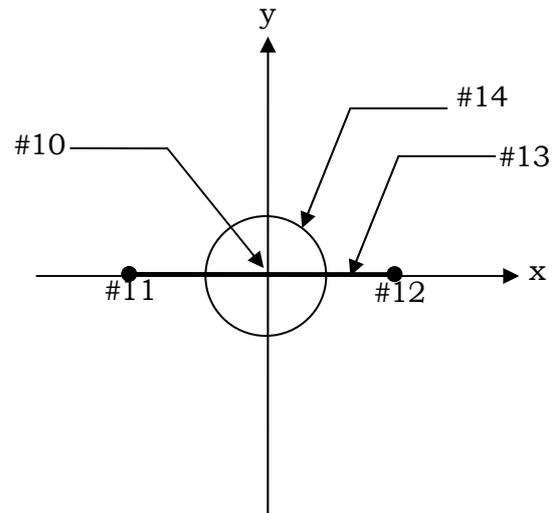


Figure I.9 Format d'un fichier STEP en langage EXPRESS[05].

I.4.1.3 Echange de données dans STEP

Un fichier d'échange STEP comporte les instances d'une partie ou de la totalité des entités définies dans un modèle EXPRESS. L'échange se déroule correctement si l'interprétation des données transférées est la même pour l'émetteur que pour le récepteur. Dans le cas de la norme STEP, cette propriété est garantie par l'unicité du modèle EXPRESS implémenté. Autrement dit, il ne peut y avoir d'échange de données STEP si les systèmes qui souhaitent échanger des données ne partagent pas un même modèle EXPRESS.

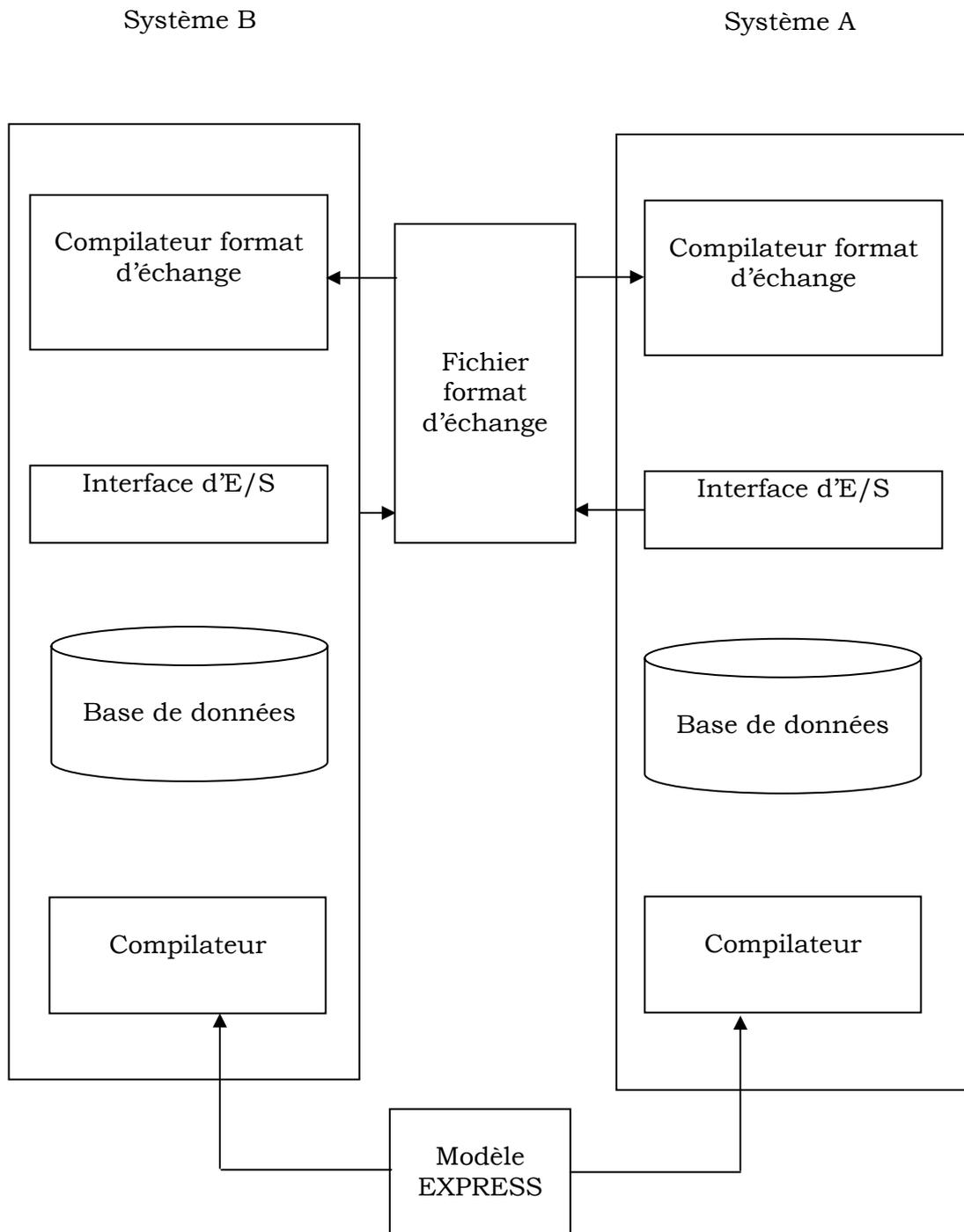


Figure I.10 Echange de données dans STEP [12].

La phase de vérification de la validité des données au format d'échange par rapport à un modèle EXPRESS se base sur l'application des règles de correspondance définies dans le document du standard STEP-21.

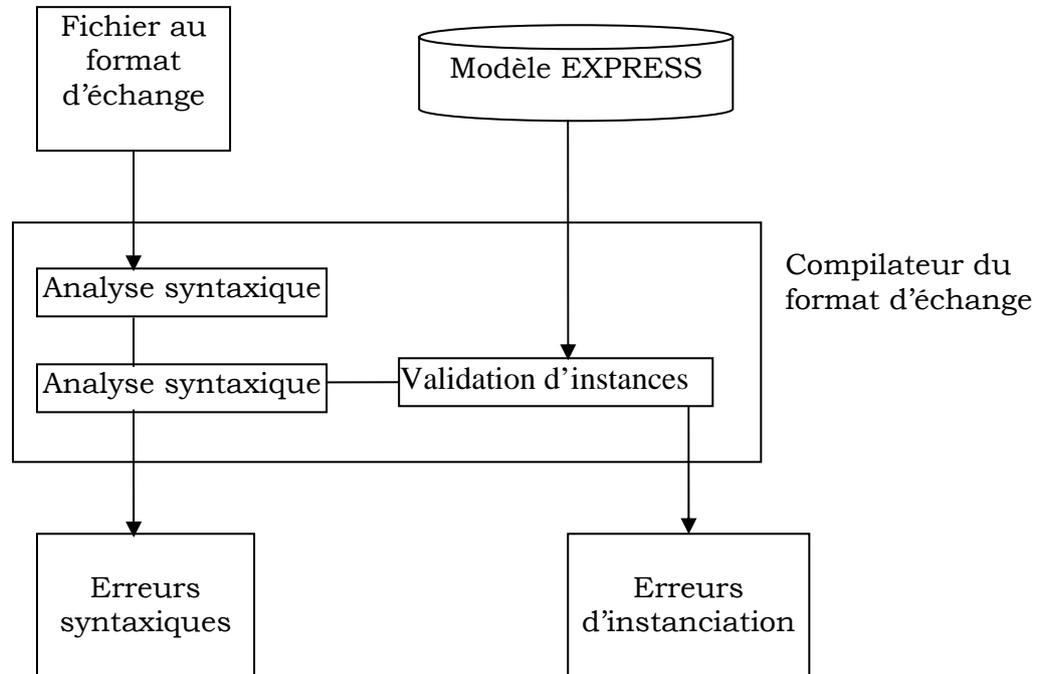


Figure I.11 Les différentes étapes d'analyse d'un fichier d'échange STEP [12].

I.4.2 La norme IGES

En 1979, aux Etats Unis, les utilisateurs et les éditeurs de systèmes CAO se sont mobilisés pour créer le premier standard d'échange de données CAO. Au printemps 1980 le NBS (National Bureau of Standard) avait formé l'organisation IGES (Initial Graphics Exchange Specification)[37]. IGES est un standard d'échange, dont leur principal objectif est l'échange des dessins techniques. Il est utilisé principalement dans le domaine de mécanique et d'aéronautique. Au début, il contenait des entités de base telle que la géométrie, l'annotation et la structure. Les nouvelles versions ont intégré les méthodes éléments finis et les systèmes électriques. Le transfert des modèles solides a été ajouté dans la version 4.0 [28].

Elle permet d'échanger des informations de type courbes, textes, notes, surfaces, solides, maillages, schématiques.

I.4.2.1 Format d'un fichier IGES

Le format d'un fichier IGES est composé de 5 sections distinctes

1. Section départ « Start Section » .
2. Section générale « Global Section » .
3. Section donnée « Directory Entry Section » .
4. Section paramètre « Parameter Data Section » .
5. Section terminale « Terminate section » .

La section « départ » contient des commentaires.

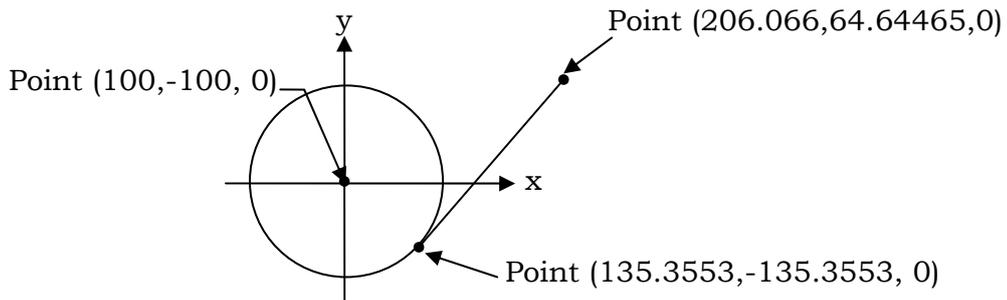
La section « générale » contient les paramètres généraux du transfert.

La section « donnée » contient les données à transférer (types d'entités,...).

La section « paramètre » contient les informations propres à chaque type d'entité.

La section « terminale » ne contient qu'une seule ligne, marquant la fin de fichier.

Exemple d'un format IGES



Section départ

Example IGES FILE	S0000001
-------------------	----------

Section générale

1H,,1H ;,,9HMASTERCAM,1H1,16,8,24,8,56,,1.,1,4HINCH,1,0.01,	G0000001
13H850101.010000,0.,100.,,,;	G0000002

Section donnée

116	1	1	1	1			0	00000000	D0000001
116	0	3	1						D0000002
124	2	1	1	1			0	00000000	D0000003
124	0	3	1						D0000004

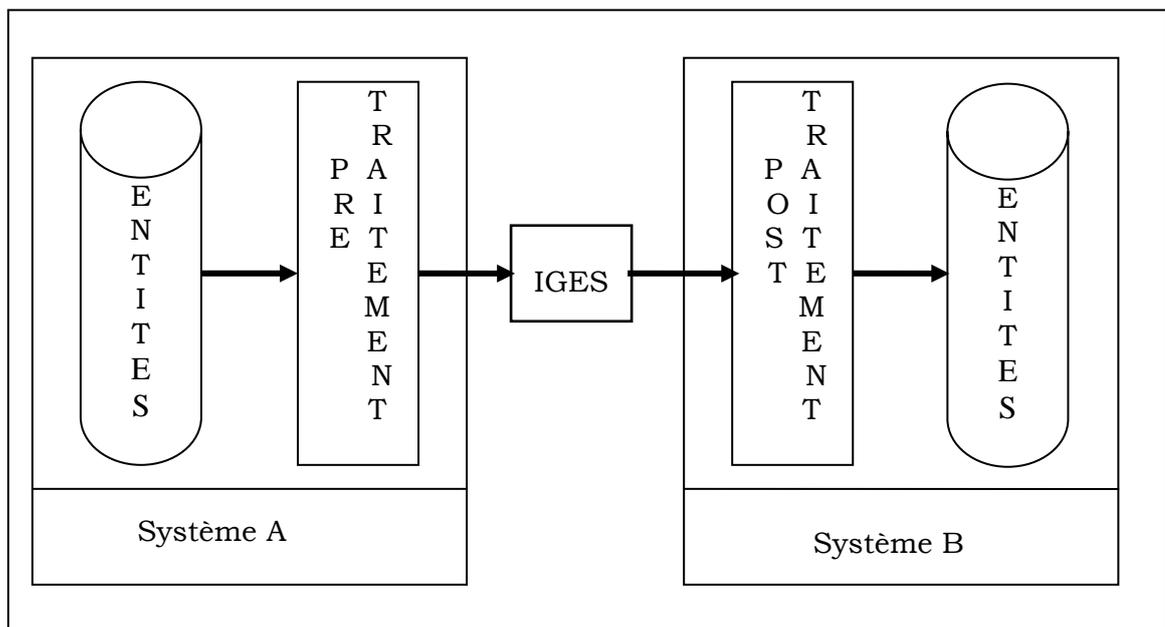
100	3	1	1	1			3	00000000	D0000005
100	0	3	1						D0000006
110	4	1	1	1			0	00000000	D0000007
110	0	3	1						D0000008

Section paramètre

116, 100., -100.,0. ;	1P0000001
124,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0. ;	3P0000002
100, 0., 100.,-100.,150.,-100.,150.,-99,99999 ;	5P0000003
110,135.3553,-135.3553, 0.,206.066, -64.64465,0. ;	7P0000004

Section terminale

S0000001G0000002D0000008P0000004	T0000001
----------------------------------	----------

I.4.2.2 Les transferts en lecture et écriture**Figure I.12** Schéma de transfert entre deux systèmes interactifs de CFAO.

L'échange d'information entre deux systèmes à l'aide d'un fichier IGES se fait en deux étapes :

1. Ecriture : un pré-traitement réalise sur le système émetteur et permettant de générer un fichier IGES.
2. Lecture : un post-traitement effectuée sur le système récepteur qui consiste

à traduire le fichier IGES selon le langage propre à ce système.

I.4.2.3 Possibilités de transfert

En écriture, les fichiers générés par l'interface IGES sont formatés selon les spécifications de la norme IGES.

Quelque soit le sens de transfert (écriture ou lecture), l'interface transfère les données suivantes :

1. Géométrie (à l'exception des solides CSG).
2. Structure, figures.
3. Cotation et habillage (texte, cotes, hachures).
4. Dessins et vues.
5. Attributs graphiques (types de lignes, couleurs et épaisseurs).
6. Niveaux.

Avantages et inconvénients de la norme IGES

Avantages:

1. Interface neutre d'échanges de données pièces.
2. Conserve la géométrie sous forme de modèle 3D surfacique.

Inconvénients:

1. Pas d'échange de modèle 3D solide.
2. Seul les modèles mathématiques normalisés sont échangés.
3. Bon pour l'échange DAO.
4. Format volumineux et vieux (peu de support dans le futur)
5. Pas d'échanges d'information produit.
6. Interfaces de lecture et écriture difficiles à supporter.

I.4.3 La Norme SET

SET (Standard d'Echange et de transfert) est un standard français lancé en 1983 par Aérospatiale. Cette dernière avait besoin d'une base de données commune pour des données issues de différents systèmes de CAO. Le standard représente une solution aux exigences relatives à l'échange de données entre différents systèmes de CFAO, et au besoin d'archiver ces données. La première version normalisée est apparue en 1985 (norme AFNOR expérimentale Z68.300.85.08). Elle contient:

- des spécifications détaillées du domaine de la mécanique.

- des informations supplémentaires concernant la structure des données et les concepts utilisés.
- des règles et des recommandations concernant les spécifications qui assurent la cohérence dans le futur développement.

En 1987 est apparu le groupe opérationnel GOSET dont les principaux objectifs étaient de développer SET, d'apporter une aide technique aux utilisateurs de la norme, de valider les interfaces entre les logiciels de CFAO et la norme et de participer aux travaux de normalisation nationaux et internationaux dans le domaine des échanges de données CFAO.

I.4.4 la Norme VDA-FS

Le standard allemand, "Verband der Automobilindustrie Flaechen-Schnittstelle - Grafikstandard", (VDA-FS) a été développé dans l'industrie automobile pour l'échange des modèles surfaciques. Il a été normalisé en 1985 par DIN (L'organisme de normalisation allemande). Cette norme était basée sur IGES mais a ajouté en particulier un format d'échange de fichiers de calcul à celle-ci[37].

I.4.5 Standard DXF

DXF est un standard développé par la société AUTODESK qui édite le logiciel AUTOCAD. Ce standard a pour objectif de sauvegarder les modélisations de DAO (Dessin Assisté par Ordinateur) d'AUTOCAD dans des fichiers ASCII. Le succès qu'a eu le logiciel AUTOCAD a rendu ce standard largement utilisable pour échanger des modélisations de DAO principalement dans le monde de la micro-informatique. Ce standard permet d'échanger sous forme numérique des plans de dessin industriel c'est-à-dire des entités géométriques bidimensionnelles, des annotations de dessin et des hachures.

I.5 Conclusion

Les données de produit ont été visées par des travaux de normalisation dans les instances nationales et internationales de standardisation. Le standard STEP vise la représentation sous forme de données de produits conçus et/ou fabriqués dans l'entreprise (appelé "données de produits").

La norme STEP a été conçue pour répondre à des besoins bien précis dans le domaines de la CFAO , la possibilité d'une utilisation partielle des méthodes STEP est une nouvelle approche qui couvrir plusieurs phases d'une technologie de

l'information dans un domaine très éloigné de la CFAO. Toutefois des outils existent et prouvent chaque jour son utilité. Pour assurer une réelle communication par les données par une interopérabilité entre les différents outils-logiciels métiers intervenant à chaque instant de la vie du produit, il est clair que c'est ce type d'outil qu'il faut déployer. Il ne constitue peut être pas la solution optimale mais a déjà le mérite d'exister et de fonctionner.

***Chapitre II : Méthodes de
réduction de degré de
courbes et surfaces de
Bézier***

Introduction

Dans cette partie, nous allons faire un rappel général sur les courbes et les surfaces de Bézier, et les différentes méthodes pour la conversion par approximation des courbes de Bézier de degré élevé.

II.1 Courbes et surfaces de Bézier

II.1.1 Polynôme de Bernstein

Ces polynômes ont été utilisés par Bernstein, pour l'approximation polynomiale des fonctions. Les propriétés essentielles de ces polynômes proviennent de leur propriété relative au binôme de Newton et de la relation de récurrence $(1 = (t + (1-t))^n)$.

Définition

Polynôme de Bernstein de degré n et d'indice i , définit par la formule suivante :

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}.$$

C_n^i : est le coefficient classique du binôme $\frac{n!}{i!(n-i)!}$.

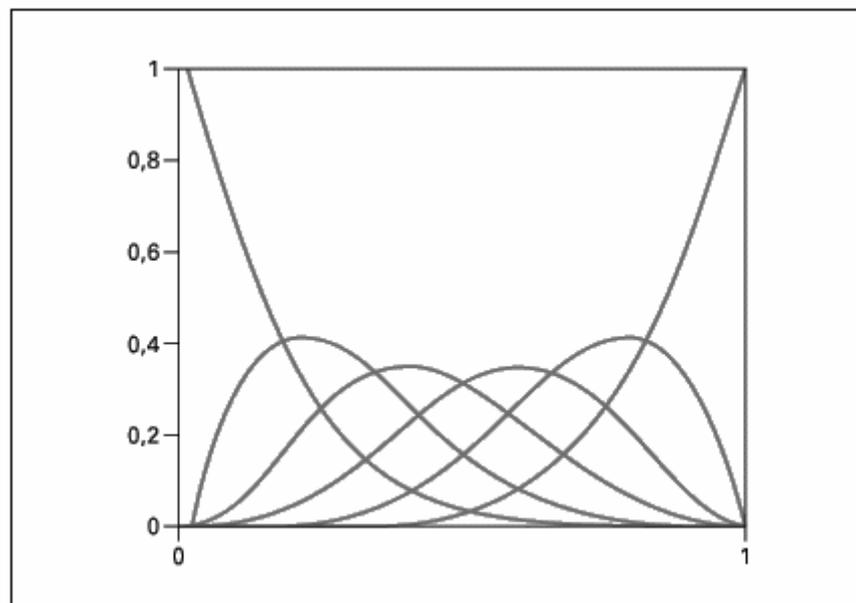


Figure II.1 Fonction de Bernstein pour $n = 6$.

Propriétés

- Partition de l'unité $\sum_{i=0}^n B_{i,n}(t) = 1$.
- Positivité $\forall t \in [0,1], B_{i,n}(t) \geq 0$.
- Récursivité $\forall i \in [0, n-1], B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t)$.
- Invariance affine : les combinaisons barycentriques sont invariantes par cartes affines.
- Invariance par transformations affines du paramètre : algébriquement, cette propriété s'écrit

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

- Propriété d'enveloppe convexe : elle est due à ce que pour $t \in [0,1]$, les polynômes de Bernstein sont positifs.
- Symétrie : $\sum_{i=0}^n P_i \cdot B_{i,n}(t) = \sum_{i=0}^n P_{n-i} \cdot B_{i,n}(1-t)$.

II.1.2 Courbe de Bézier non rationnelle

Du nom d'un ingénieur de Renault (Pierre Bézier) qui développa ce modèle de courbes pour la conception de carrosseries de voitures. La relation s'exprime par :

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

Les points P_i sont appelés points de contrôles de la courbe $P(t)$.

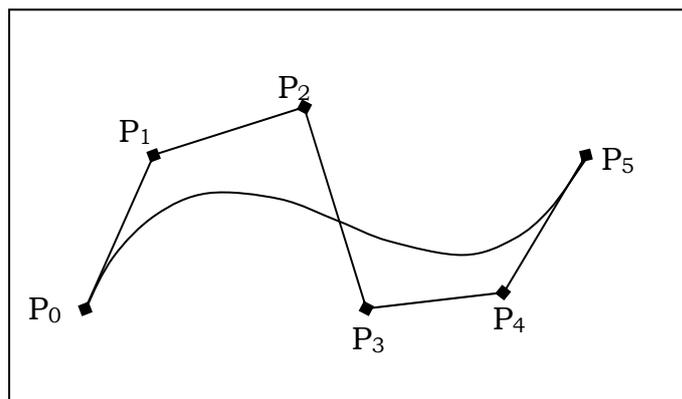


Figure II.2 La courbe de Bézier et son polygone caractéristique selon P. Bézier.

II.1.2.1 Evaluation d'un point courant de la courbe de Bézier

Pour évaluer un point de la courbe pour un paramètre t donné, on utilise l'algorithme de De Casteljaou suivant:

$$\begin{aligned}
 P_n^i(t) &= P_i, 0 \leq i \leq n \\
 P_i^{r+1}(t) &= (1-t)P_{i-1}^r + tP_i^r \\
 0 \leq r &\leq n-1, r+1 \leq i \leq n \\
 P_n^n &= P(t).
 \end{aligned}
 \tag{II.2}$$

Les points P_i^r , sont appelés les points auxiliaires de la courbe de Bézier $P(t)$.

II.1.2.2 Forme matricielle d'une courbe de Bézier

Quelque auteurs Faut et Pratt[22], Mortenson[42], préfèrent écrire les courbes de Bézier et autres courbes polynomiales sous la forme matricielle :

$$P(t) = \sum_{i=0}^n P_i \cdot B_{i,n}(t).$$

avec :

$$P(t) = [P_0 \dots P_n] \cdot \begin{bmatrix} m_{00} & \dots & m_{0n} \\ \vdots & & \vdots \\ m_{n0} & \dots & m_{nn} \end{bmatrix} \cdot \begin{bmatrix} t^0 \\ \vdots \\ t^n \end{bmatrix}$$

avec les éléments $m_{ij} = (-1)^{j-i} \binom{n}{j} \binom{j}{i}$.

II.1.2.3 Dérivée d'une courbe de Bézier

La dérivée à l'ordre k d'une courbe de Bézier est donnée par :

$$D^k P(t) = \frac{n!}{(n-k)!} \sum_{i=0}^{(n-k)} B_{i,n-k} D^k P_i \tag{II.3}$$

En prenant $k=1$ avec des valeurs de $t=0$ et $t=1$, les vecteurs tangents de la courbe sur les deux extrémités sont donnés par :

$$n\Delta P_0 = n(P_1 - P_0).$$

$$n\Delta P_{(n-1)} = n(P_n - P_{(n-1)}).$$

Remarque :

La dérivée k^{ieme} d'une courbe de Bézier en ses extrémités dépend uniquement des $k+1$ points de contrôle les plus proches de cette extrémité.

II.1.2.4 Hodographe d'une courbe de Bézier

L'hodographe d'une courbe de Bézier $P(t)$, est une courbe de Bézier illustrant l'évolution d'une dérivée de $P(t)$.

Remarque :

L'hodographe d'ordre k est une dérivée d'ordre k de la courbe de Bézier $P(t)$ de degré n , donc est une courbe de Bézier de degré $(n - k)$.

II.1.3 Courbe de Bézier rationnelle

Une courbe de Bézier rationnelle est définie par :

$$P(t) = \frac{\sum_{i=0}^n B_{i,n}(t) \cdot w_i \cdot P_i}{\sum_{j=0}^n B_{j,n}(t) \cdot w_j} \tag{II.4}$$

où

$$\frac{B_{i,n}(t) \cdot w_i}{\sum_{j=0}^n B_{j,n}(t) \cdot w_j} : \text{ sont les fonctions de Bernstein rationnelles.}$$

w_i : sont les poids associés aux P_i .

II.1.4 Subdivision d'une courbe de Bézier

Pour tout $t \in [0,1]$, le point $P(t)$ subdivise la courbe en deux courbes de Bézier de même degré. Les deux courbes $P1(t), P2(t)$ sont exprimées en fonction des points auxiliaires par :

$$P_0^0, \dots, P_n^n .$$

$$P_n^n, \dots, P_0^0 .$$

on peut aussi écrire les deux courbes de Bézier en fonction de $P(t)$ comme suit :

$$P1(t) = \sum_{i=0}^n P_i^i B1_{i,n}(t), \text{ pour } 0 \leq t \leq t_0$$

$$B1_{i,n}(t) = B_{i,n}\left(\frac{t}{t_0}\right).$$

$$P2(t) = \sum_{i=0}^n P_n^n B2_{i,n}(t), \text{ pour } t_0 \leq t \leq 1,$$

$$B_{2_{i,n}}(t) = B_{i,n}\left(\frac{t-t_0}{1-t_0}\right).$$

Exemple :

D'après l'équation de récurrence (II.2).

pour $n=4$

$$\begin{array}{ccccccc}
 P_0^0 = P1_0 & & & & & & \\
 & \longrightarrow & P_0^1 = P1_1 & & & & \\
 P_1^0 & & & \longrightarrow & P_0^2 = P1_2 & & \\
 & \longrightarrow & P_1^1 & & & \longrightarrow & P_0^3 = P1_3 \\
 P_2^0 & & & \longrightarrow & P_1^2 & & \longrightarrow & P_0^4 = P1_4 = P2_0 \\
 & \longrightarrow & P_2^1 & & & \longrightarrow & P_1^3 = P2_1 \\
 P_3^0 & & & \longrightarrow & P_2^2 = P2_2 & & \\
 & \longrightarrow & P_3^1 = P2_3 & & & & \\
 P_4^0 = P2_4 & & & & & &
 \end{array}$$

II.1.5 Elévation de degré des courbes de Bézier

Soit P_i ($i = 0, \dots, n$), le polygone caractéristique d'une courbe de Bézier non rationnelle de degré n exprimer par l'équation :

$$P(t) = \sum_{i=0}^n P_i B_{i,n}(t) \quad t \in [0,1] . \quad (II.5)$$

Il s'agit de construire une courbe de Bézier $Q(t)$ non rationnelle correspondant exactement à $P(t)$, et admettant pour polygone caractéristique Q_i ($i = 0, \dots, n+1$), $Q(t)$ et $P(t)$, liées par l'équation :

$$Q(t) = \sum_{i=0}^{n+1} Q_i B_{i,n+1}(t) = \sum_{i=0}^n P_i B_{i,n}(t) \quad t \in [0,1] . \quad (II.6)$$

a) Cas des courbes non rationnelles

Dans le cas des courbes non rationnelles, les sommets Q_i et P_i sont liés par la relation :

$$\begin{aligned}
 Q_0 &= P_0 . \\
 Q_i &= P_{i-1} + \frac{(n+1-i)}{(n+1)}(P_i - P_{i-1}), i \in \{1, \dots, n\} . \\
 Q_{n+1} &= P_n .
 \end{aligned} \quad (II.7)$$

b) Cas des courbes rationnelles

Soit une courbe de Bézier $P(t)$ de degré n définie par ses $(n+1)$ pôles P_i ($i=0, \dots, n$) chargés par les poids β_i ($i=0, \dots, n$). La courbe de Bézier $Q(t)$ de degré $(n+1)$ est définie par ses $(n+2)$ pôles Q_i ($i=0, \dots, n+1$) déterminés par Bensalah [07], Piegl et Tiller [45].

$$Q_0 = P_0.$$

$$\mu_0 = \beta_0.$$

$$Q_i = P_{i-1}^1 \left(\frac{n+1-i}{n+1} \right), \quad i \in \{1, \dots, n+1\} \quad (II.8)$$

$$\mu_i = \beta_{i-1}^1 \left(\frac{n+1-i}{n+1} \right), \quad i \in \{1, \dots, n+1\} \quad (II.9)$$

les P_i^1 et β_i^1 seront calculés par les équations suivantes :

$$\beta_i^1(t) = (1-t)\beta_i^0(t) + t\beta_{i+1}^0(t)$$

$$P_i^1(t) = (1-t) \frac{\beta_i^0(t)}{\beta_i^1(t)} P_i^0 + t \frac{\beta_{i+1}^0(t)}{\beta_i^1(t)} P_{i+1}^0(t)$$

avec $P_i^0(t) = P_i$; $\beta_i^0(t) = \beta_i$.

II.1.6 Surfaces de Bézier non rationnelles

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

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) \cdot B_{j,m}(v) \cdot P_{i,j} \quad (II.10)$$

avec $(u, v) \in [0, 1]^2$

$P_{i,j}$: sont les pôles de la surface où le réseau caractéristique.

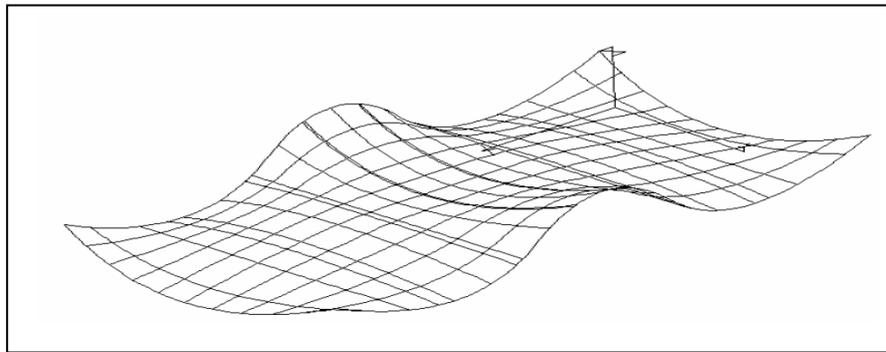


Figure II.3 Surface de Bézier non rationnelle.

II.1.6.1 Evaluation d'un point courant de la surface de Bézier

Pour une surface $S(u, v)$ défini par un réseau caractéristique $P_{i,j}$, l'algorithme consiste à appliquer, à chaque rangée de sommets dans la direction u , l'algorithme de P. de Casteljau élaboré pour les courbes, et pour chaque courbe isoparamétrique u on applique l'algorithme enfin on obtient le sommet $P_{0,0}^{n,m}$ correspondant à $S(u, v)$.

Les étapes composant l'algorithme de calcul de $S(u, v)$ s'énoncent :

Pour ($j = 0, \dots, m$)

Pour ($k = 1, \dots, n$)

Pour ($i = 0, \dots, n - k$)

$$P_{i,j}^k = (1 - u_0) \cdot P_{i,j}^{k-1} + u_0 \cdot P_{i+1,j}^{k-1}$$

Fin pour

Fin pour

Fin pour

Pour ($l = 1, \dots, m$)

Pour ($j = 0, \dots, m - l$)

$$P_{0,j}^{n,l} = (1 - v_0) \cdot P_{0,j}^{n,l-1} + v_0 \cdot P_{0,j+1}^{n,l-1}$$

Fin pour

Fin pour

Le point cherché est donné par : $S(u_0, v_0) = P_{0,0}^{n,m}$

II.1.6.2 Forme matricielle d'une surface de Bézier

Nous avons formulé une expression matricielle des courbes de Bézier. Cette approche se transpose bien aux facettes produites tensorielles. Nous pouvons écrire :

$$S(u, v) = [B_{0,n}(u) \cdots B_{n,n}(u)] \cdot \begin{bmatrix} P_{0,0} & \cdots & P_{0,m} \\ \vdots & & \vdots \\ P_{n,0} & \cdots & P_{n,m} \end{bmatrix} \begin{bmatrix} B_{0,m}(v) \\ \vdots \\ B_{m,m}(v) \end{bmatrix}$$

si nous écrivons les polynômes de Bernstein sous forme de monômes, nous obtenons :

$$S(u, v) = [u^0 \cdots u^n] \cdot N^T \cdot \begin{bmatrix} P_{0,0} & \cdots & P_{0,m} \\ \vdots & & \vdots \\ P_{n,0} & \cdots & P_{n,m} \end{bmatrix} \cdot M \cdot \begin{bmatrix} v^0 \\ \vdots \\ v^m \end{bmatrix}$$

Les matrices carrées M et N sont données par :

$$m_{i,j} = (-1)^{j-i} \binom{m}{j} \binom{j}{i}, \quad n_{i,j} = (-1)^{j-i} \binom{n}{j} \binom{j}{i}$$

II.1.6.3 Dérivée d'une surface de Bézier

Les dérivées que nous considérerons sont les dérivées partielles $\partial/\partial u$ ou $\partial/\partial v$ et Les dérivées partielles croisées $\partial^2/\partial u \partial v$. Une dérivée partielle est le vecteur tangent d'une courbe isoparamétrique .

Les dérivées partielles en u :

$$\frac{\partial}{\partial u} S(u, v) = n \cdot \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} (P_{i+1,j} - P_{i,j}) \cdot B_{i,n-1}(u) \cdot B_{j,m}(v)$$

Les dérivées partielles en v :

$$\frac{\partial}{\partial v} S(u, v) = m \cdot \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (P_{i,j+1} - P_{i,j}) \cdot B_{i,n}(u) \cdot B_{j,m-1}(v)$$

Les dérivées partielles croisées en u et v :

$$\frac{\partial^2}{\partial u \partial v} S(u, v) = n \cdot m \cdot \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (P_{i+1,j+1} - P_{i+1,j} - P_{i,j+1} + P_{i,j}) \cdot B_{i,n-1}(u) \cdot B_{j,m-1}(v)$$

II.1.7 Surfaces de Bézier rationnelles

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

$$\begin{aligned} S(u, v) &= \frac{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) \cdot B_{j,m}(v) \cdot w_{i,j} \cdot P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) \cdot B_{j,m}(v) \cdot w_{i,j}} \\ &= \sum_{i=0}^n \sum_{j=0}^m G_{i,n,j,m}(u, v) \cdot P_{i,j} \end{aligned} \quad (II.11)$$

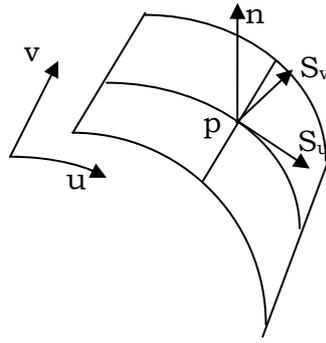
avec :

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

$P_{i,j}$: sont les pôles de la surface où le réseau caractéristique.

$w_{i,j}$: sont les poids associés aux $P_{i,j}$.

II.1.8 Courbure gaussienne et courbure moyenne



$n = \frac{S_u \times S_v}{\|S_u \wedge S_v\|}$: C'est Le vecteur unitaire normal à la surface en point $p(u, v)$.

La courbure k_0 d'une telle courbe est appelée courbure normale de la surface en p dans la directions $t = du/dv$. La courbure est donnée par :

$$k_0(p, t) = \frac{L + 2 \cdot M \cdot \lambda + N \cdot \lambda^2}{E + 2 \cdot F \cdot \lambda + G \cdot \lambda^2} \quad (II.12)$$

avec

$$L = L(u, v) = -S_u n_u = n S_{uu} \cdot$$

$$M = M(u, v) = -\frac{1}{2}(S_u n_v + S_v n_u) = n S_{uv} \cdot$$

$$N = N(u, v) = -S_v n_v = n S_{vv} \cdot$$

$$E = E(u, v) = S_u S_u \cdot$$

$$F = F(u, v) = S_u S_v \cdot$$

$$G = G(u, v) = S_v S_v \cdot$$

Remarque : cette formule simple a une interprétation intéressante et importante, connue sous le nom de théorème de Meusnier.

Les valeurs extrêmes k_1 et k_2 de (II.12) correspondent aux racines λ_1, λ_2 de

$$\det \begin{vmatrix} \lambda^2 & -\lambda & 1 \\ E & F & G \\ L & M & N \end{vmatrix} = 0 \quad (II.13)$$

λ_1, λ_2 sont toujours réels donc les valeurs extrêmes k_1 et k_2 elles -mêmes sont les racines de

$$\det \begin{vmatrix} k_0 E - L & k_0 F - M \\ k_0 F - M & k_0 G - N \end{vmatrix} = 0 \quad (II.14)$$

Un calcul de (II.14) avec $k_0^2 - (k_1 + k_2)k_0 + k_1 k_2 = 0$ conduit à

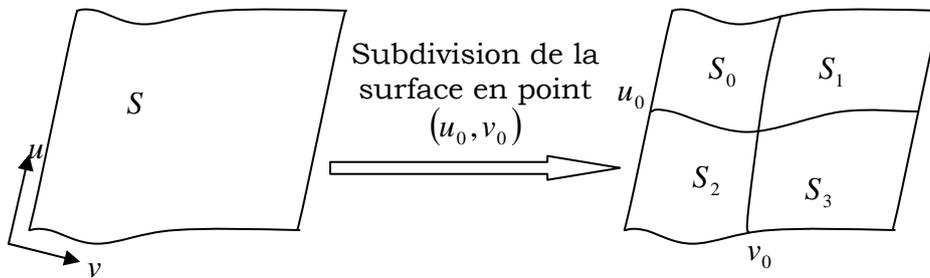
$$K = k_1 k_2 = \frac{LN - M^2}{EG - F^2} \quad \text{et} \quad H = \frac{1}{2}(k_1 + k_2) = \frac{NE - 2MF + LG}{2(EG - F^2)}.$$

Le terme K est appelé courbure gaussienne.

Le terme H est appelé courbure moyenne.

II.1.9 Subdivision d'une surface de Bézier

Le procédé de subdivision le plus simple pour une surface de Bézier rationnelle ou non rationnelle $S(u, v)$ consiste à appliquer l'algorithme de De Casteljau à une surface au point (u_0, v_0) , cela se traduit par l'application de l'algorithme de De Casteljau pour les courbes aux $(n+1)$ polygones dans la direction u au paramètre u_0 ensuite aux $2(m+1)$ polygones ainsi obtenus dans la direction v au paramètre v_0 . On obtient ainsi quatre surfaces à chaque subdivision de même type possédant chacune son propre réseau caractéristique.



II.1.10 Élévation de degré des surfaces de Bézier

Si nous voulons élever le degré à la fois dans les directions u et v , nous pouvons effectuer la procédure d'élévation de degré des courbes de Bézier dans la direction u puis dans la direction v . Les deux approches génèrent une surface de degré $(n+1, m+1)$, son réseau peut être trouvé en une étape par :

$$Q_{i,j} = \begin{bmatrix} \frac{i}{n+1} & \left(1 - \frac{i}{n+1}\right) \end{bmatrix} \cdot \begin{bmatrix} P_{i-1,j-1} & P_{i-1,j} \\ P_{i,j-1} & P_{i,j} \end{bmatrix} \cdot \begin{bmatrix} \frac{j}{m+1} \\ \left(1 - \frac{j}{m+1}\right) \end{bmatrix} \text{ avec } \begin{cases} i = 0, \dots, n+1 \\ j = 0, \dots, m+1 \end{cases}$$

II.2 Réduction de degré des courbes de Bézier

Le principe de la construction est basé sur l'algorithme d'élévation de degré (premier algorithme de réduction de degré [Farin]) : on ne peut pas trouver une courbe de Bézier de degré $(n-1)$ identique. Autrement c'est possible pour des cas particulier si on vérifie la propriété de l'hodographe).

Soit P_n le polygone caractéristique d'une courbe de Bézier de degré n , il s'agit de construire le polygone caractéristique Q_{n-1} d'une courbe de Bézier de degré $(n-1)$ telle que :

$$Q(t) = \sum_{i=0}^{n-1} Q_i B_{i,n-1}(t) \approx P(t). \quad (II.15)$$

Les pôles Q_i peuvent être calculés par la résolution inverse de l'équation d'élévation de degré (II.7), cette équation peut être résolue de deux façons (II.16) et (II.17), cela dépend si l'on commence par P_0 ou par P_n :

a) Cas des courbes non rationnelles

$$R_i = \frac{n}{(n-i)} P_i - \frac{i}{(n-i)} R_{i-1} \quad i \in \{1, \dots, n-1\}. \quad (II.16)$$

$$S_{i-1} = \frac{n}{i} P_i - \frac{(n-i)}{i} S_i \quad i \in \{n-1, \dots, 1\}. \quad (II.17)$$

Les polygones S_{n-1} et R_{n-1} sont différents, mais donnent chacun une approximation à son début, et le meilleur polygone est celui donné par (II.18), qui est une moyenne pondérée des deux polygones:

$$Q_i = \frac{n-i}{n} R_i + \frac{i}{n} S_i. \quad (II.18)$$

a) Cas des courbes rationnelles

$$R_i = \frac{n}{n-i} \cdot \frac{w_i}{\alpha_i} P_i - \frac{\alpha_{i-1}}{\alpha_i} \frac{i}{n-i} R_{i-1} \quad i \in \{1, \dots, n-1\} \quad (II.19)$$

$$\alpha_i = \frac{n}{n-i} w_i - \frac{i}{n-i} \alpha_{i-1} \quad (II.20)$$

$$S_{i-1} = \frac{n}{i} \cdot \frac{w_i}{\beta_{i-1}} P_i - \frac{\beta_i}{\beta_{i-1}} \frac{n-i}{i} S_i \quad i \in \{1, \dots, n-1\} \quad (II.21)$$

$$\beta_{i-1} = \frac{n}{i} w_i - \frac{n-i}{i} \beta_i \quad (II.22)$$

L'approximation de Farin pour les courbes rationnelles devient :

$$Q_i = \frac{n-i}{n} \frac{\alpha_i}{\mu_i} R_i - \frac{\beta_i}{\mu_i} \frac{i}{n} S_i \quad (II.23)$$

$$\mu_i = \frac{n-i}{n} \alpha_i - \frac{i}{n} \beta_i \quad (II.24)$$

II.3 Conversion des courbes de Bézier non rationnelles

La réduction de degré d'une courbe de Bézier, produit une erreur d'approximation que l'on ne peut pas contrôler et dépend du degré et de la forme de la courbe.

Suivant l'application associée à cette réduction, l'approximation se présente de la manière suivante :

- Réduction le degré d'une courbe en respectant une ou plusieurs contraintes (position, tangente, courbure, torsion,...).
- L'utilisation de l'algorithme de subdivision d'une courbe de degré n , permet de respecter la tolérance de position donnée.

Plus le nombre de subdivisions ou ensemble de portions de courbes réduites est plus grand, plus l'erreur de l'approximation sera plus petite.

II.3.1 Principe général de la conversion

La conversion par approximation est donnée par l'organigramme ci-dessous :

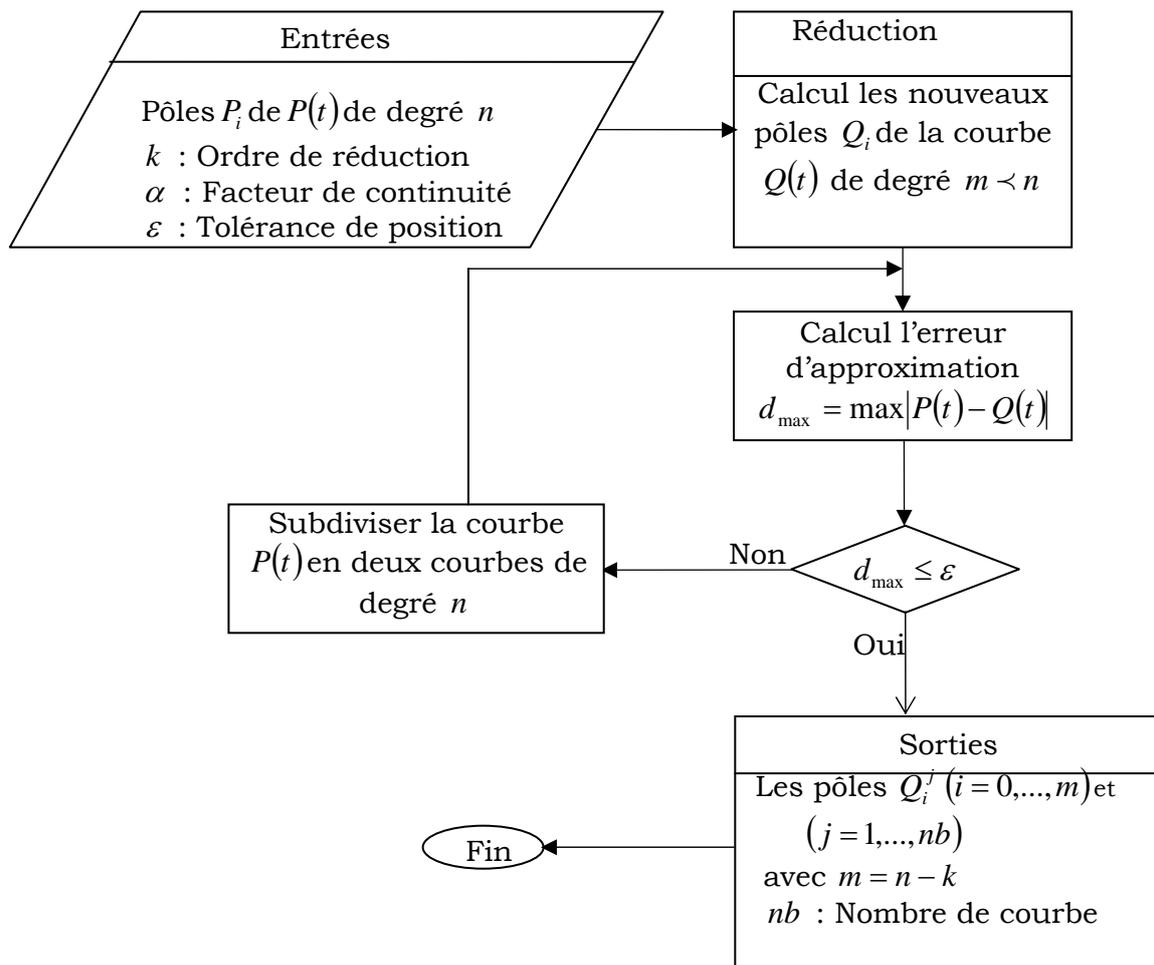


Figure II.4 Organigramme général du principe de la conversion des courbes de Bézier par approximation [47]

II.3.2 Méthodes de réduction de degré

II.3.2.1 Méthode de Forrest

La méthode de réduction de degré des courbes de Bézier donne deux solutions généralement différentes $R(t)$ (II.16) produisant une bonne approximation au voisinage de $t=0$, et $S(t)$ (II.17) produisant une bonne approximation au voisinage de $t=1$. La méthode de Forrest consiste à prendre pour la courbe réduite $Q(t)$ les $n/2$ premiers pôles de la courbe $R(t)$ et les $n/2$ derniers pôles de la courbe $S(t)$, nous remarquons que cette méthode prend deux formes pour n pair ou impair.

L'erreur maximale est poussée pour $t=1/2$. Si on applique k fois successive de cette méthode on obtient une courbe réduite de degré $m = n - k$.

II.3.2.2 Méthode de Hoschek

Cette méthode basée sur le principe de la minimisation par la méthode des moindres carrés, en effectuant une valeur des paramètres t_j durant le processus d'approximation ; le but de la paramétrisation est de trouver une bonne répartition des paramètres t_j sur les points M_j permettant une approximation optimale :

$$M_j \approx P(t_j) = \sum_{i=0}^n P_i \cdot B_{i,n}(t_j) \quad (II.25)$$

Le problème donc se ramène à minimiser la quantité suivante :

$$\sum_{j=0}^{np} (M_j - P(t_j)) \quad (np : \text{Nombre de points}).$$

Cet algorithme découle des deux applications : Hoschek et Wissel [35], Hoschek[33] qui consiste à :

- Subdiviser une courbe de degré n donnée en une ou plusieurs courbes de degré $m < n$ en respectant une tolérance de position fixée ε .
- Elévation de degré d'une courbe de degré m en une courbe de degré $n > m$ en respectant une tolérance de position fixée ε .

Parmi les avantages de cette méthode, nous citerons les résultats satisfaisant dans le cas des courbes planes, et pour l'approximation des offsets et donne des résultats pratiques très satisfaisants.

II.3.2.3 Méthode de séparation des variables

II.3.2.3.1 Réduction de degré à $(n-1)$

L'originalité de l'approche proposée réside dans la stratégie de résolution du problème par de séparation des variables : le processus de réduction s'effectue en deux étapes : les pôles influant sur la continuité souhaitée $C^{\alpha-1}$ sont d'abord déterminés par approximation de Forrest [25] en utilisant l'algorithme inverse d'élévation de degré [17], [18], tandis que les pôles intermédiaires sont calculés par minimisation du vecteur erreur par l'application de la méthode d'approximation au sens des moindres carrés [32], [33]. Le problème se simplifie alors et se ramène à la résolution d'un système d'équations linéaires. La conversion de la courbe de Bézier de degré n définie par l'équation suivante:

$$C(t) = \sum_{i=0}^n P_i \cdot B_i^n(t) \quad ; \quad i = 0, \dots, n \quad \text{et} \quad t \in [0, 1] \quad (II.26)$$

et la courbe de Bézier équivalente de degré $m < n$, définie par l'équation :

$$Q(t) = \sum_{i=0}^m Q_i \cdot B_i^m(t) \quad ; \quad i = 0, \dots, m \quad \text{et} \quad t \in [0, 1] \quad (II.27)$$

II.3.2.3.2 Détermination des pôles influant sur la continuité

Considérons la courbe de Bézier $C(t)$ de degré n , telle que définie par l'équation (II.26) que nous voulons réduire pour obtenir la courbe de Bézier équivalente $Q(t)$ de degré $m = (n-1)$, définie par l'équation (II.27). L'approximation de Forrest [24], [25] permet alors de déterminer les pôles respectant une continuité maximale aux points communs extrêmes. Ainsi, pour obtenir une continuité d'ordre $C^{\alpha-1}$, avec $(\alpha - 1 < (n-1)/2)$, il suffit d'impliquer les α premiers pôles R_i et les α derniers pôles S_i , à savoir :

$$\begin{aligned} \text{Pour } i &= (0, \dots, \alpha - 1) \\ &Q_i = R_i \\ \text{Pour } i &= (n - 1, \dots, n - \alpha) \\ &Q_i = S_i \end{aligned} \quad (II.28)$$

II.3.2.3.3 Détermination des pôles intermédiaires

Les pôles intermédiaires de la courbe équivalente $Q(t)$ sont les pôles qui n'ont pas d'influence sur la continuité et sont déterminés de la façon suivante :

Les points courants np de la courbe équivalente $Q(t)$ sont déterminés par une optimisation paramétrique [32], [33], basée sur une approximation au sens des moindres carrés. En choisissant sur la courbe donnée $P(t)$ initiale, $(n+1)$ points M_j répartis selon le paramètre curviligne équidistant $t_j = j/n$; ($i = 0, \dots, n$) tels que :

$$M_j = \sum_{i=0}^n P_i \cdot B_{i,n}(t_j) + e_j$$

où e_j est l'erreur vectorielle qui s'écrit, si les $P(t_j)$ sont les points associés à M_j :

$$e_j = P(t_j) - M_j$$

Le problème d'optimisation se ramène ainsi à la minimisation de l'erreur vectorielle e_j . Soit alors :

$$D = \sum_{j=0}^{np} e_j = \sum_{j=0}^{np} [M_j - P(t_j)] \rightarrow \min$$

La minimisation de la somme des carrés des distances entre les points M_j et les points associés sur la courbe $C(t)$ revient à minimiser la fonction :

$$D = \sum_{i=0}^{np} \|e_j\|^2 = \sum_{j=0}^{np} (e_{jx}^2 + e_{jy}^2 + e_{jz}^2) \quad (II.29)$$

qui se concrétise donc par la minimisation des trois composantes :

$$e_x = \sum_{j=0}^{np} (e_{jx}^2) \quad ; \quad e_y = \sum_{j=0}^{np} (e_{jy}^2) \quad ; \quad e_z = \sum_{j=0}^{np} (e_{jz}^2)$$

dont les variables d'optimisation sont les composantes (P_{ix}, P_{iy}, P_{iz}) des pôles du polygone caractéristique recherché. Les valeurs minimales sont obtenues en résolvant alors les équations :

$$\frac{\partial D_x}{\partial P_{ix}} = 0 \quad ; \quad \frac{\partial D_{yx}}{\partial P_{iy}} = 0 \quad ; \quad \frac{\partial D_z}{\partial P_{iz}} = 0 \quad ; \quad \text{pour } (i = 0, \dots, n)$$

ce qui fournit un système d'équations vectorielles linéaire de dimensions $(n+1)$.

Soit donc :

$$\delta = \sum_{j=1}^{np} [P(t_j) - Q(t_j)]^2 = \sum_{j=1}^{np} e_j^2 \quad (II.30)$$

$$\text{où : } e_i = P(t_j) - Q(t_j) \quad (II.31)$$

est l'erreur vectorielle. En insérant (II.26) et (II.27) dans (II.31), il vient :

$$e_j = \sum_{i=0}^n P_i \cdot B_{i,n}(t_j) - \sum_{i=0}^{n-1} Q_i \cdot B_{i,n-1}(t_j) \quad (II.32a)$$

La dérivée partielle de (II.32a) par rapport à Q_h pour $h = (\alpha, \alpha - 1, \dots, n - \alpha - 1)$ donne :

$$\frac{\partial e_j}{\partial Q_h} = \frac{\partial}{\partial Q_h} [P(t_j) - Q(t_j)] \quad (II.32b)$$

En développant l'équation (II.32b), on obtient : $\frac{\partial e_j}{\partial Q_h} = \frac{\partial P(t_j)}{\partial Q_h} - \frac{\partial Q(t_j)}{\partial Q_h}$;

$$\text{avec : } \frac{\partial P(t_j)}{\partial Q_h} = 0, \quad \frac{\partial e_j}{\partial Q_h} = -\frac{\partial Q(t_j)}{\partial Q_h} = -\sum_{i=0}^{n-1} \frac{\partial Q_i}{\partial Q_h} \cdot B_{i,n-1}(t_j)$$

$$\frac{\partial Q_i}{\partial Q_h} = \begin{cases} 0 & \text{si } i \neq h \\ 1 & \dots \text{si } i = h \end{cases}, \quad \text{ce qui donne :}$$

$$\frac{\partial e_j}{\partial Q_h} = -B_{h,n-1}(t_j) \quad (II.33)$$

D'autre part, avec :

$$\frac{\partial \delta}{\partial Q_h} = \sum_{j=1}^{np} \frac{\partial (e_j)^2}{\partial Q_h} = 2 \cdot \sum_{j=1}^{np} e_j \cdot \frac{\partial e_j}{\partial Q_h} \quad (II.34)$$

et l'insertion de (II.32a) et de (II.32b) dans (II.34), on peut écrire:

$$\frac{\partial \delta}{\partial Q_h} = -2 \cdot \sum_{j=1}^{np} e_j \cdot B_{h,n-1}(t_j) \quad (II.35)$$

$$\frac{\partial \delta}{\partial Q_h} = -2 \cdot \sum_{j=1}^{np} \left[\sum_{i=0}^n P_i \cdot B_{i,n}(t_j) \cdot B_{h,n-1}(t_j) - \sum_{i=0}^{n-1} Q_i \cdot B_{i,n-1}(t_j) \cdot B_{h,n-1}(t_j) \right] \quad (II.36)$$

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

$$\frac{\partial \delta}{\partial Q_h} = 0 \quad h = (\alpha, \alpha - 1, \dots, n - \alpha - 1) \quad (II.37)$$

c'est-à-dire d'après (II.36) :

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

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

Nous obtenons ainsi le système d'équations d'ordre $(n - 2\alpha)$ de la forme:

$A_{ih} \cdot Q_h = b_i$, que l'on peut exprimer par :

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

$$- \sum_{i=n-\alpha}^{n-1} Q_i \cdot \sum_{j=1}^{np} B_{i,n-1}(t_j) \cdot B_{h,n-1}(t_j) = \sum_{i=\alpha}^{n-\alpha-1} Q_i \cdot \sum_{j=1}^{np} B_{i,n-1}(t_j) \cdot B_{h,n-1}(t_j) \quad (II.38)$$

En posant :

$$\gamma_{ih} = \sum_{j=1}^{np} B_{i,n}(t_j) B_{h,n-1}(t_j) \quad (II.39)$$

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

$$b_h = \sum_{i=0}^n P_i \cdot \gamma_{ih} - \sum_{i=0}^{\alpha-1} Q_i \cdot \beta_{ih} - \sum_{i=n-\alpha}^{n-1} Q_i \cdot \beta_{ih} \quad (II.40)$$

L'expression (II.38) se ramène alors à :

$$\sum_{i=\alpha}^{n-\alpha} Q_i \cdot \beta_{ih} = b_h \quad (II.41)$$

L'algorithme de réduction d'un degré n à un degré $(n-1)$ s'exprimera ainsi comme suit :

Pour $i = (0, 1, \dots, \alpha - 1)$

$$Q_i = R_i$$

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

$$Q_i = S_i$$

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

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

II.3.2.4 Méthode de Wathins et Worsey

Cette méthode est basée sur le théorème de l'alternation (± 1) établi par Watkins et Worsey [49], donc pour la réduction des courbes de Bézier non rationnelles, Wathins et Worsey ont utilisés le polynôme de Tchebychev pour minimiser l'erreur d'approximation. On peut résumer cette méthode par les trois étapes suivantes :

-Exprimer la courbe dans la base de Tchebychev.

-Tronquer le terme de plus haut degré.

-Reconvertir la courbe dans la base de Bézier –Bernstein.

Le polynôme de Tchebychev de degré i est donné par :

$$U_i(t) = \cos(i \cdot \arccos(t)) \quad \forall i. \quad (II.42)$$

Si $P(t)$ la courbe de Bézier et $Q(t)$ la courbe de Bézier réduite, alors l'erreur d'approximation est donnée par :

$$e(t) = P(t) - Q(t) = \alpha_n \cdot U_n(t) \quad (II.43)$$

On peut montrer que l'erreur due à la réduction de degré diminue par multiplication au facteur $\frac{1}{2^n}$ à chaque subdivision, ce qui donne après s subdivision une erreur égale à $\frac{1}{2^n} \cdot \alpha_n$. donc pour obtenir une approximation de degré $(n-1)$ à une courbe de degré n en respectant une tolérance ε donnée, le nombre de subdivision s nécessaire est :

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

II.3.2.5 Méthode de Eck

Le principe général de la méthode est basé sur l'application de l'algorithme inverse de l'élévation de degré au sens de Forrest [24]. La résolution d'algorithme de réduction de degré de courbe de Bézier donne deux courbes différentes $R(t)$ et $S(t)$, la première courbe produit une meilleure approximation au voisinage de $t=0$, et la deuxième courbe produit une meilleure approximation au voisinage de $t=1$. 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 rassemble les coefficients R_i et S_i entre eux.

$$Q_i = (1 - \lambda_{i,n}) \cdot R_i + \lambda_{i,n} \cdot S_i \quad (i = 0, \dots, n-1) \quad (II.44)$$

avec

$$0 \leq \lambda_{i,n} \leq 1$$

Et l'introduction des polynômes de Tchebychev contraints $U_{n,\alpha}(t)$ qui permet de déterminer la meilleure approximation forcée pour une continuité $C^{\alpha-1}$ aux deux extrémités. Ces polynômes sont donnés en terme du polynôme de Bernstein par la forme suivante :

$$U_{n,\alpha}(2t-1) = \sum_{i=0}^n C_{i,n,\alpha} B_{i,n}(t) \quad t \in [0,1]$$

$$C_{i,n,\alpha} = C_{n-i,n,\alpha} = 0 \quad (i = 0, \dots, \alpha)$$

$$C_{i,n,\alpha} = (-1)^{n+i} |C_{i,n,\alpha}| \quad (i = \alpha, \dots, n-\alpha)$$

$$C_{i,n,\alpha} = (-1)^{n+i} \binom{n}{i}^{-1} \binom{n}{i-\alpha} \binom{n}{j+\alpha}$$

les nouveaux facteurs $\lambda_{i,n,\alpha}$ sont déterminés par la relation suivante [18]:

$$\lambda_{i,n,\alpha} = \frac{1}{\binom{2n}{n+2\alpha}} \sum_{j=\alpha}^i \binom{n}{j-\alpha} \binom{n}{j+\alpha}.$$

donc l'équation (II.28) devient :

$$Q_i = (1 - \lambda_{i,n,\alpha}) \cdot R_i + \lambda_{i,n,\alpha} \cdot S_i$$

II.3.2.6 Méthode d'approximation L^1 par l'interpolation des points extrêmes

La méthode d'approximation L^1 a été proposée par Kim et Moon, pour résoudre le problème de réduction de degré n en degré $(n-1)$ avec une interpolation des points extrêmes par la norme L^1 . La méthode proposée offre une meilleure réduction pour les courbes de Bézier de degré $n \leq 5$ par l'interpolation des points extrêmes en utilisant des splines réelles. Dans le cas général, ils ont obtenus une bonne réduction de degré avec l'utilisation appropriée des polynômes $U_n(t)$ de Tchebychev de second type et de degré n .

L'algorithme de subdivision nécessite l'application de la réduction de degré $(n-1)$ avec une contrainte de position ε [38].

Réduction de degré avec des splines réelles

Une spline réelle de degré n et de vecteur nœud $0 < u_1 < \dots < u_r < 1$ dans l'intervalle $[0,1]$, est définie par la forme suivante :

$$p_n(t) = c \left(t^n + 2 \sum_{i=0}^r (-1)^i (t - u_i)_+^i \right) + \sum_{j=0}^{n-1} a_j t^j \quad (II.45)$$

avec c, a_0, \dots, a_{n-1} constantes réels.

Exemple :

Nous prenons dans cet exemple $n = 2$, la courbe de Bézier est définie comme suit :

$$P_2(t) = \sum_{i=0}^2 P_i B_{i,2}(t)$$

si $Q_1(t)$ est la courbe réduite, donc l'erreur est donnée par :

$$U_2 = P_2 - Q_1$$

la fonction d'erreur est définie par :

$$U_2(t) = \Delta^2 P_0 t(t-1)$$

La spline réelle $p_2(t)$ correspond à $U_2(t)$ est définie par la forme suivante :

$$P_2(t) = \begin{cases} -\frac{1}{2}t^2 + \frac{1}{2}t, & \text{si } \Delta^2 P_0 > 0 \\ \frac{1}{2}t^2 - \frac{1}{2}t & \text{si } \Delta^2 P_0 < 0 \end{cases}$$

donc la courbe réduite est donnée par :

$$\begin{aligned} Q_1(t) &= (\Delta^2 P_0 + 2\Delta^1 P_0)t + P_0 \\ &= \sum_{i=0}^1 Q_i B_{i,1}(t) \end{aligned}$$

qui implique : $Q_0 = P_0$ et $Q_1 = P_2$

Réduction de degré avec des polynômes de Tchebychev

Le polynôme de Tchebychev $U_n(t)$ de deuxième espèce et de degré n est définie par :

$$U_n(t) = \frac{\sin((n+1)\arccos(t))}{\sin(\arccos(t))} \quad -1 \leq t \leq 1 \quad (II.46)$$

$$U_i^*(t) = U_n\left(\cos\left(\frac{\pi}{i+1}\right)(2t-1)\right) \quad 0 \leq t \leq 1 \text{ pour } (i = 2, \dots, n)$$

$$U_n^*(0) = 0 \text{ et } U_n^*(1) = 0$$

$$P_n(t) = Q_{n-1}(t) - \frac{\Delta^n P_0}{2^{2n} (\cos(\pi/(n+1)))^n} U_n^*(t)$$

l'erreur de réduction de la courbe de Bézier $P_n(t)$ avec la norme L^1 est donné par la relation suivante :

$$e = \frac{|\Delta^n P_0| (n-1)}{2^{2n} (n+1)} \cdot \frac{1}{(\cos(\pi/(n+1)))^{n+1}}$$

Algorithme de subdivision

Données les pôles $P_i (i = 0, \dots, n)$.

ε : Tolérance de position.

Calculer le nombre de subdivision.

$$k = \left(\text{int} \left(\frac{|\Delta^n P_0|}{2^{2n}} \cdot \frac{n-1}{n+1} \cdot \frac{1}{\cos(\pi/(n+1))^{n+1}} \cdot \varepsilon \right)^{1/n+1} \right)$$

Subdiviser la courbe de Bézier $P_n(t)$ en $k+1$ courbes de Bézier

équidistance.

Calculer l'erreur de réduction.

$$e = \frac{|\Delta^n P_0|(n-1)}{2^{2n}(n+1)} \cdot \frac{1}{(\cos(\pi/n+1))^{n+1}}$$

Fin

avec
$$\Delta^n P_0 = \sum_{i=0}^n (-1)^i \binom{n}{i} P_{n-i}.$$

II.3.2.7 Méthode de transformation entre la base de Legendre et la base de Bernstein pour élévation et réduction de degré

Pour des réductions multiples et optimales de degré des courbes de Bézier, nous allons voir une méthode simple et efficace [39], qui concerne la transformation entre Legendre et la base de Bernstein, en respectant la norme L^2 .

La base de Bernstein et de Legendre

Le polynôme de Bernstein de degré n est donné par :

$$B_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} t^i \quad 0 \leq t \leq 1$$

le polynôme de Legendre $L_n(t)$ est donné par :

$$L_n(t) = \sqrt{2n+1} \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{i, i} (t^2 - t)^i (2t - 1)^{n-2i} \quad (II.47)$$

avec

$$\binom{n}{i, j} = \frac{n!}{i! j! (n-i-j)!}.$$

la relation liée entre le polynôme de Bernstein et Le polynôme de Legendre est donnée par :

$$L_k(t) = \sqrt{2k+1} \sum_{i=0}^k (-1)^{k+i} \binom{k}{i} B_{i,k}(t)$$

$$L_k(t) = \sum_{j=0}^n M_n(j, k) B_{j,n}(t).$$

$$M_n(j, k) = \frac{\sqrt{2k+1}}{\binom{n}{j}} \sum_{i=\max(0, j+k-n)}^{\min(j, k)} (-1)^{k+i} \binom{k}{i} \binom{k}{i} \binom{n-k}{j-i} \quad (II.48)$$

M_n : La matrice de passage entre la base de Legendre vers la base de Bernstein.

la norme L^2 pour une courbe de Bézier $P_n(t)$ est définie par:

$$\begin{aligned} \|P_n\|_2^2 &= \int_0^1 \left| \sum_{i=0}^n P_i B_{i,n}(t) \right|^2 dt \\ &= \frac{1}{2n+1} \sum_{i,j} P_i P_j \frac{\binom{n}{i} \binom{n}{j}}{\binom{2n}{i+j}} \\ \|P_n\|_2^2 &= P^t q_n P \text{ avec } q_n(i,j) = \frac{1}{2n+1} \frac{\binom{n}{i} \binom{n}{j}}{\binom{2n}{i+j}} \quad (i, j = 0, \dots, n) \end{aligned}$$

Soit la courbe de Bézier $P(t)$ de degré n réduite en courbe de Bézier $Q(t)$ de degré m , les pôles de la courbe $Q(t)$ sont déterminés par la relation suivante :

$$Q = (T_{m,r}^t q_n T_{m,r})^{-1} T_{m,r}^t q_n P \text{ avec } Q = (Q_0, \dots, Q_m)^t \text{ et } P = (P_0, \dots, P_n)^t$$

$T_{m,r}$: La matrice d'élévation de degré est défini par :

$$T_{m,r}(i,j) = \frac{\binom{m}{j} \binom{r}{i-j}}{\binom{m+r}{i}}, \quad i = 0, \dots, m+r \text{ et } j = 0, \dots, m.$$

II.3.2.8 Méthode de représentation matricielle unifiée pour réduction des courbes de Bézier

Hasik et Namyong ont représentés le problème de réduction de degré des courbes de Bézier sous forme d'une matrice unifiée. La plupart des algorithmes de réduction de degré sont basées sur le problème inverse de l'élévation de degré et ont été décrits comme ensemble d'équations récursives [48].

L'élévation de degré de la courbe de Bézier $Q(t)$ de degré $(n-1)$ à une courbe de Bézier $P(t)$ de degré n est donnée par la formule suivante :

$$P_i = \frac{1}{n} (i Q_{i-1} + (n-i) Q_i) \text{ avec } i = 0, \dots, n \quad (II.49)$$

pour la recherche des Q_i , l'expression (II.49) est résolue de deux façons (II.50) et (II.51) ; cela dépend si l'on commence par Q_0 ou par Q_{n-1} .

$$\bar{Q}_i^1 = \frac{1}{n-i} (n P_i - i \bar{Q}_{i-1}^1) \text{ avec } (i = 0, \dots, n-1) \quad (II.50)$$

$$\bar{Q}_{i-1}^2 = \frac{1}{i} \left(n.P_i - (n-i).\bar{Q}_i^2 \right) \text{ avec } (i = n, \dots, 1) \quad (II.51)$$

La solution finale est donnée par la combinaison linéaire entre les deux solutions, à savoir :

$$\bar{Q}_i = (1 - \lambda_i)\bar{Q}_i^1 + \lambda_i\bar{Q}_i^2 \text{ avec } (i = 0, \dots, n-1) \quad (II.52)$$

$\{\lambda_i\}_{i=0}^{n-1}$: Sont des facteurs introduire par Eck [17].

La représentation matricielle des équations précédentes donne les relations suivantes :

$$P = AQ \text{ avec } P = [P_0, \dots, P_n]^t \text{ et } Q = [Q_0, \dots, Q_{n-1}]^t$$

$$A = (a_{ij})$$

$$a_{ij} = \begin{cases} (n-i)/n & i = j \\ i/n & i = j+1 \\ 0 & \text{les autres cas} \end{cases}$$

$$\bar{Q}^1 = LP \text{ et } \bar{Q}^2 = UP \quad L = (l_{ij}) \text{ et } U = (u_{ij})$$

avec

$$l_{ij} = \begin{cases} (-1)^{i+j} \frac{\binom{n}{j}}{\binom{n-i}{j}} & \text{pour } i = 0, \dots, n-1 \text{ et } j = 0, \dots, i \\ 0 & \text{les autres cas} \end{cases}$$

$$u_{ij} = \begin{cases} (-1)^{i+j+1} \frac{\binom{n}{j}}{\binom{n-i}{j}} & \text{pour } i = 0, \dots, n-1 \text{ et } j = i+1, \dots, n \\ 0 & \text{les autres cas} \end{cases}$$

donc

$$\bar{Q} = (I - D)\bar{Q}^1 + D\bar{Q}^2$$

$$\bar{Q} = \{(I - D)L + DU\}P$$

I : Matrice identité

$$D = \text{diag}(\lambda_0, \dots, \lambda_{n-1})$$

II.3.2.9 Utilisation des polynômes de Jacobi pour la réduction de degré des courbes de Bézier avec des contraintes de continuité C^α

Cette méthode basée sur le polynôme contraint de Jacobi comme fonction d'erreur pour obtenir une bonne réduction de degré de la courbe de Bézier avec des

contrainte C^α aux les bornes ($\alpha = 2,3$). Cette technique fut proposée par Ahn, le résultat obtenu est une extension de la méthode proposée par Kim et Ahn (2000). La meilleure réduction de degré pour une contrainte C^α dans la norme L_∞ ne peut pas être obtenue sous une forme explicite et exige des calculs plus sophistiqués et complexes car basés sur l'algorithme de Remes, mais elle peut être représentée sous une forme explicite par la méthode de réduction de degré avec une contrainte C^α utilisant les polynômes contraints de Jacobi [01].

Le polynôme Contraint de Jacobi avec une continuité C^α est définie par :

$$J_n^\alpha(t) = \frac{t^{\alpha+1}(t-1)^{\alpha+1} P_{n-2\alpha-2}^{(2\alpha+1, 2\alpha+1)}(2t-1)}{\binom{2n-2}{n-2\alpha-2}}, \quad t \in [0,1] \quad (II.53)$$

avec $\alpha \geq 1$ et $n \geq 2\alpha + 2$

Le polynôme de Jacobi est définie par :

$$P_m^{(\alpha, \alpha)}(t) = \sum_{i=0}^m \binom{m+\alpha}{i} \binom{m+\alpha}{m-i} \left(\frac{t-1}{2}\right)^i \left(\frac{t+1}{2}\right)^{m-i}, \quad t \in [-1,1] \quad (II.54)$$

Le polynôme Contraint de Jacobi avec une continuité C^α dans la norme L_∞ est déterminé par la relation suivante :

$$\|J_n^\alpha(\cdot)\|_{L_\infty[0,1]} = \begin{cases} \max\{|J_n^\alpha(\tau_1/2)|, |J_n^\alpha(\tau_1)|\} & n \text{ impaire} \\ \max\left\{\frac{\binom{n-1}{n/2-\alpha-1}}{2^n \binom{2n-2}{n-2\alpha-2}}, |J_n^\alpha(\tau_1)|\right\} & n \text{ pair} \end{cases}$$

$\tau_1/2$ et τ_1 : Sont les plus proches zéros de la $\frac{d}{dt} J_n^\alpha(t)$, avec ($\tau_1 < 1$).

la forme de Bézier du polynôme contraint de Jacobi avec une contrainte C^α , pour tous $\alpha \geq 1$ et $n \geq 2\alpha + 2$.

$$J_n^\alpha(t) = \sum_{i=0}^n c_i B_{i,n}(t) \quad (II.55)$$

avec

$$c_i = \begin{cases} 0 & \text{si } i = 0, \dots, \alpha, n-\alpha, \dots, n \\ (-1)^{n-i} \binom{n-1}{i-\alpha-1} \binom{n-1}{n-\alpha-i-1} / \binom{n}{i} & \text{si non} \end{cases}$$

Soit $[P_0, \dots, P_n]$, le polygone de la courbe de Bézier $P(t)$, et $[Q_0, \dots, Q_{n-1}]$ le polygone de la courbe réduite $Q(t)$ est définie par :

$$Q_i^\alpha = \begin{cases} P_0 - \Delta^n P_0 c_0 & \text{si } i = 0 \\ \frac{n}{n-i} \left(P_i - \Delta^n P_0 c_i - \frac{i}{n} \bar{Q}_{i-1} \right) & \text{si } i = 1, \dots, n-1 \end{cases}$$

où

$$\Delta^n P_0 = \sum_{i=0}^n (-1)^i \binom{n}{i} P_{n-i}$$

II.3.2.10 Multi-réduction optimale de degré des courbes de Bézier avec contraintes de continuité aux pôles extrêmes

En utilisant des polynômes contraints orthogonaux de Jacobi dans cette méthode, la courbe de Bézier réduite est donnée sous sa forme explicite en garantissant la continuité des points extrêmes ($r, s \geq 0$) par l'application de la théorie d'approximation traditionnelle. Pour réduire la durée des calculs, cette méthode s'effectue en une seule étape [14].

Définition

La forme de Bézier d'un polynôme de Bernstein est définie par :

$$B_{i,m}(t) = \sum_{j=i}^{n+i-m} b_{i,j}^{(m,n)} B_{j,n}(t) \quad n > m, i = 0, \dots, m. \quad (II.56)$$

avec

$$b_{i,j}^{(m,n)} = \binom{m}{i} \binom{n-m}{j-i} / \binom{n}{j}.$$

Soit $P(t)$ la courbe de Bézier de degré n et $Q(t)$ la courbe de Bézier réduite de degré m , on peut écrire $Q(t)$ sous la forme suivante :

$$Q(t) = \sum_{i=0}^r Q_i B_{i,m}(t) + \sum_{i=r+1}^{n-s-1} P_i^1 B_{i,n}(t) + \sum_{i=m-s}^m Q_i B_{i,m}(t) \quad (II.57)$$

avec

$$\begin{cases} Q_0 = \frac{1}{b_{0,0}^{(m,n)}} P_0, Q_j = \frac{1}{b_{j,j}^{(m,n)}} \left(P_j - \sum_{i=\max(0, j-(n-m))}^{j-1} b_{i,j}^{(m,n)} Q_i \right), j = 1, \dots, r \\ Q_m = \frac{1}{b_{m,m}^{(m,n)}} P_n \\ Q_{m-j} = \frac{1}{b_{m-j, n-j}^{(m,n)}} \left(P_{n-j} - \sum_{i=\max(0, j-(n-m))}^{j-1} b_{m-i, n-j}^{(m,n)} Q_{m-i} \right), j = 1, \dots, s \end{cases}$$

Et pour P_i^I , nous distinguons deux cas :

-si $m - s > n + r - m$

$$\begin{cases} P_j^I = P_j - \sum_{i=0}^r b_{i,j}^{(m,n)} Q_i, & j = r+1, \dots, n+r-m \\ P_j^I = P_j, & m-s-1 > n+r-m, j = n+r-m+1, \dots, m-s-1 \\ P_j^I = P_j - \sum_{i=0}^s b_{m-i,j}^{(m,n)} Q_{m-i}, & j = m-s, \dots, n-s-1 \end{cases}$$

-si $m - s \leq n + r - m$

$$\begin{cases} P_j^I = P_j - \sum_{i=0}^r b_{i,j}^{(m,n)} Q_i, & m-s-1 > 1, j = r+1, \dots, m-s-1 \\ P_j^I = P_j - \sum_{i=0}^r b_{i,j}^{(m,n)} Q_i - \sum_{i=0}^s b_{m-i,j}^{(m,n)} Q_{m-i}, & j = m-s, \dots, n+r-m \\ P_j^I = P_j - \sum_{i=0}^s b_{m-i,j}^{(m,n)} Q_{m-i}, & m-s-1 > r, j = n+r-m+1, \dots, n-s-1 \end{cases}$$

le polynôme de Jacobi en fonction du polynôme de Bernstein est donné par:

$$J_n^{(r,s)}(2t-1) = \sum_{i=0}^n (-1)^{n+i} \left(\binom{n+r}{i} \binom{n+s}{n-i} \right) B_{i,n}(t) / \binom{n}{i} \quad n \geq 0 \text{ et } t \in [0,1] \quad (II.58)$$

le polygone $[Q_0, \dots, Q_m]$ de la courbe réduite est donné par :

$$Q_i = \tilde{P}_{i-r-1}^{IV} \left(\binom{M}{i-r-1} \right) / \binom{m}{i} \quad i = r+1, \dots, m-s-1, \quad M = m - (r+s+2). \quad (II.59)$$

$$\tilde{P}_M^{IV} = (\tilde{P}_0^{IV}, \dots, \tilde{P}_M^{IV}) = \tilde{P}_M^{III} L_{M \times M}^{(2r+2, 2s+2)}$$

$$L_{M \times M}^{(2r+2, 2s+2)} = \left(L_{k,j}^{2r+2, 2s+2} \right)_{(M+1) \times (M+1)}$$

$$L_{k,j}^{2r+2, 2s+2} = \sum_{i=\max(0, j+k-n)}^{\min(j,k)} (-1)^{k+i} b_{i,j}^{(k,n)} \left(\binom{k+2r+2}{i} \binom{k+2s+2}{k-i} \right) / \binom{k}{i}$$

$$\tilde{P}_M^{III} = (\tilde{P}_0^{III}, \dots, \tilde{P}_M^{III}) = \tilde{P}_M^{II} E_{M \times M}^{(2r+2, 2s+2)}$$

$$\tilde{P}_N^{II} = \sum_{i=0}^N P_i^{II} B_i^N(t) \quad \text{avec } N = n - (r+s+2)$$

$$P_i^{II} = P_{r+1+i}^I \left(\binom{n}{r+1+i} \right) / \binom{N}{i} \quad \text{et } i = 0, \dots, n - (r+s+2)$$

$$E_{M \times M}^{(2r+2, 2s+2)} = \left(L_{M \times M}^{(2r+2, 2s+2)} \right)^{-1}$$

II.4 Conclusion

Le choix d'une méthode d'approximation de courbes de Bézier doit essentiellement tenir compte des critères suivants :

- 1- la précision de calcul au sens position, tangente et courbure,
- 2- le temps de traitements,
3. la facilité de passage d'un modèle vers un autre,
4. l'indépendance du coefficient de réduction,
- 5- les exigences et restrictions technologiques, et la facilité de mise en œuvre dans la perspective d'une intégration dans les systèmes de CFAO,

En CFAO, la continuité géométrique est une caractéristique importante qui est indispensable lors de la réduction des courbes. Chaque méthode assure un ordre de continuité désirée et ayant des caractéristiques différentes par rapport à l'autre. Nous remarquons d'après ces algorithmes que l'erreur d'approximation varie selon le nombre des courbes obtenues lors de la de subdivision.

***Chapitre III : Programme
de conversion par
approximation
implémenté***

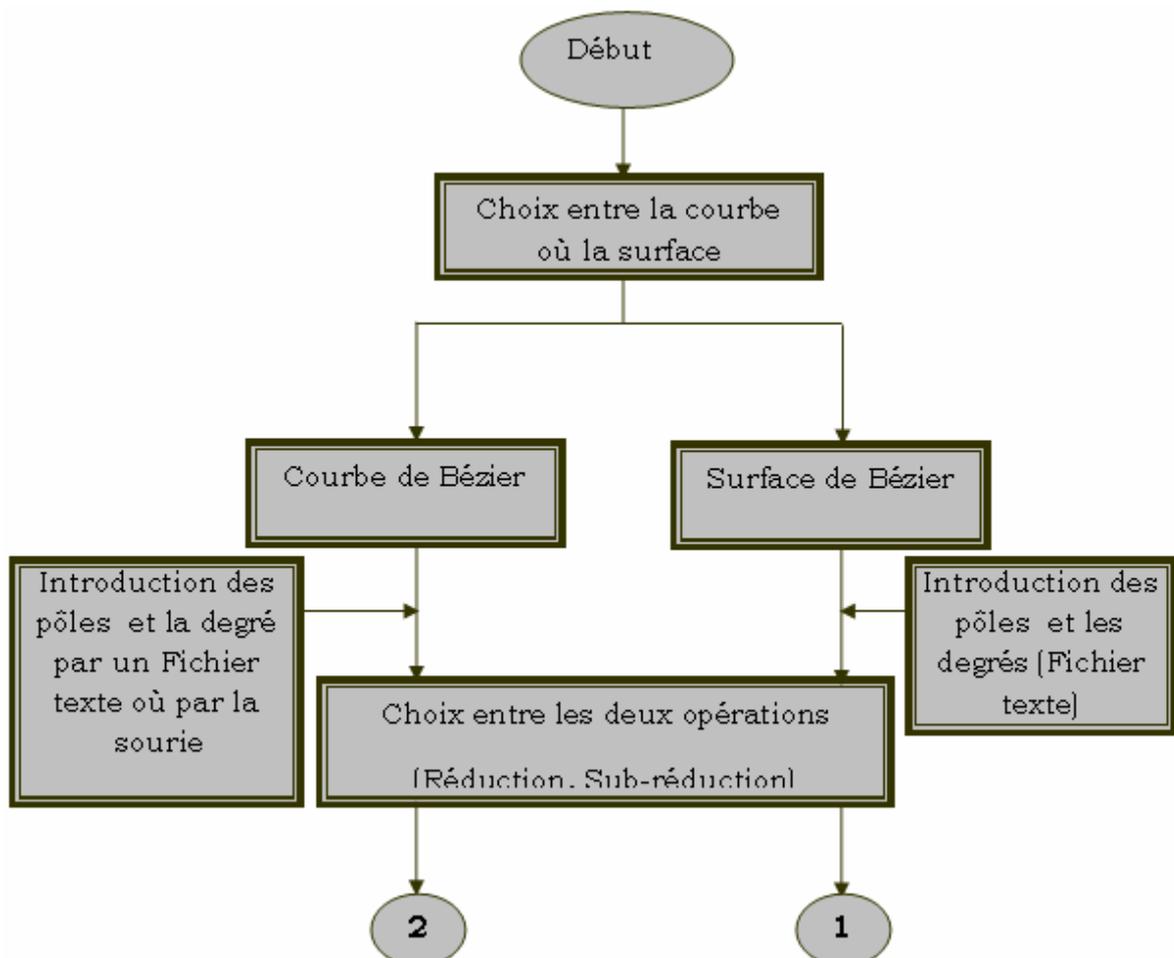
Introduction

Le programme développé est une application graphique sous Windows, en utilisant visuel C++ comme langage de programmation à l'aide d'une bibliothèque graphique OpenGL. Le programme permet d'effectuer des opérations graphiques sur l'écran, et est dédié à la conversion des courbes et des surfaces de Bézier non rationnelles. Dans le cas des courbes, deux méthodes de conversion ont été implémentées, à savoir :

1- La réduction de degré basée sur le principe inverse de l'algorithme d'élévation de degré et la minimisation des erreurs d'approximation au sens des polygones orthogonaux (méthode de Eck).

2- La réduction de degré basée sur la méthode des polygones moyens, avec résolutions par séparations des variables et minimisation des erreurs d'approximation aux sens des moindres carrés (méthode de séparation des variables). Cette dernière méthode a été étendue à la réduction de degré des surfaces de Bézier. L'architecture générale du programme globale est donnée ci-dessous (fig. III.1)

III.1 Architecture générale



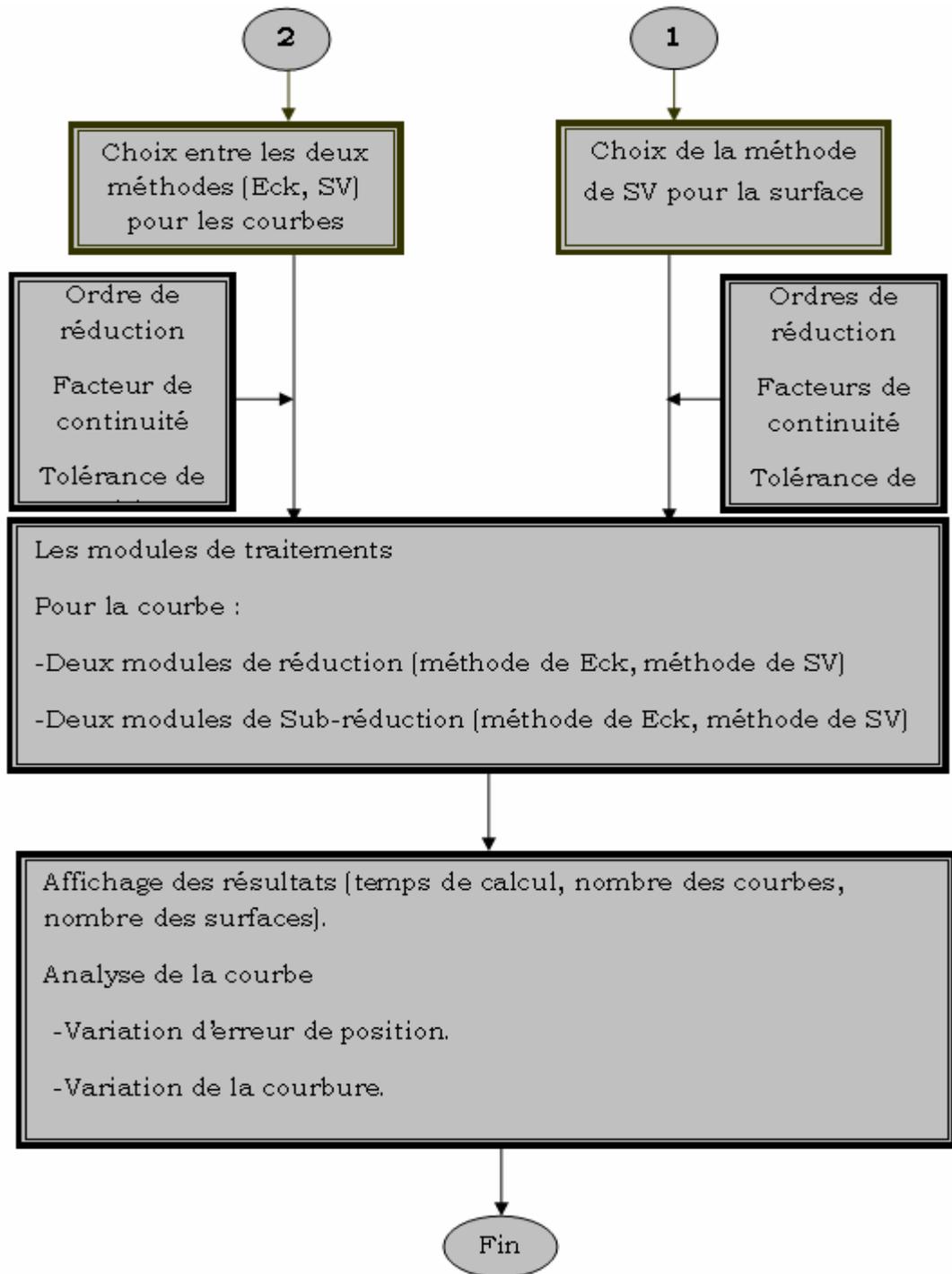


Figure III.1 Organigramme général de fonctionnement du logiciel élaboré.

III.2 Algorithmiques pour la sub-réduction de courbe mise en œuvre

Dans le cas des courbes de Bézier non rationnelles, nous utiliserons pour la conversion par approximation les deux méthodes sous-cotées. Nous donnons dans ce qui suit les procédures algorithmiques employées pour l'implémentation de notre programme.

III.2.1 Procédures algorithmique de la méthode de Eck **Pseudo-code pour la Réduction de degré « Red-Eck »**

Données P_i ($i = 0, \dots, n$) pôles de la courbe de degré n

k : Ordre de la réduction.

α : Ordre de continuité.

Début

$m = n$;

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

$Q_i = P_i$;

Fin pour

Pour ($j = 1, \dots, k$)

Algo-Eck1 ($Q, 1 \rightarrow R$) ;

Algo-Eck2 ($Q, 1 \rightarrow S$) ;

Calculer ($\lambda_{i,m,\alpha}$) ;

Pour ($i = 0, \dots, m-1$)

$Q_i = (1 - \lambda_{i,m,\alpha}) \cdot R_i + \lambda_{i,m,\alpha} \cdot S_i$;

Fin pour

$m = m - 1$;

Fin pour

Résultats : Q_i ($i = 0, \dots, m$) courbe de degré $m = n - k$;

Fin

Algo-Eck1

Données : P_i ($i = 0, \dots, n$) polygone de la courbe de degré n .

k : Ordre de la réduction.

Début

Pour ($j = 1, \dots, k$)

$R_0 = P_0$;

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

$R_i = \left(1 + \frac{i}{n-i}\right) \cdot P_i - \frac{i}{n-i} \cdot R_{i-1}$;

$P_i = R_i$;

Fin pour

$n = n - 1$;

Fin pour

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

Fin

Algo-Eck2

Données : P_i ($i = 0, \dots, n$) polygone de la courbe de degré n .

k : Ordre de la réduction.

Début

```

Pour (j = 1, ..., k)
  Sn-1 = Pn ;
  Pour (i = n - 1, ..., 1)
    Si-1 = (1 +  $\frac{n-i}{i}$ ) · Pi -  $\frac{n-i}{i}$  · Si ;
    Pi = Si ;
  Fin pour
n = n - 1 ;

Fin pour

```

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

Fin**Pseudo-code pour la Sub-réduction**

Données : P_i (i = 0, ..., n) polygone de la courbe de degré n.
 k : Ordre de la réduction.
 ε : Tolérance de position.
 α : Ordre de continuité.

Début

```

m = n ; a = 1 ; b = 1 ; c = 1 ;
Pour (i = 0, ..., n)
  Ri1 = Pi ;
Fin pour
Pour (r = 1, ..., k)
  Pour (f = a, ..., b)
    Pour (i = 0, ..., m)
      Si = Rif ;
    Fin pour
    Calculer (λi,m,α, sm,α) ;
    Calculer (ΔmS0) ;
    ns = ( 0.25 · (  $\frac{2 \cdot k \cdot s_{m,\alpha} \cdot |\Delta^m S_0|}{\varepsilon}$  )1/m ) ;
    Pour (d = 1, ..., ns)
      t0 = 1/(ns - d + 2) ;
      Sub-courbe (S, t → Q0, Q1)
      Pour (j = 0, ..., m)
        Sj = Q1j ;
      Fin pour
      c = c + 1 ;
      Red-Eck (Q0, 1, α → Q)
      Pour (j = 0, ..., m - 1)
        Rjc = Qj ;
      Fin pour

```

```

    Fin pour
    c = c + 1 ;
    Red-Eck (S,1,α → Q)
    Pour (j = 0,...,m-1)
        Rjc = Qj ;
    Fin pour

    Fin pour
    a = b + 1 ; b = c ; m = m - 1 ;
    Fin pour
    f = 0 ;
    Pour (i = a,...,b)
        f = f + 1 ;
    Pour (j = 0,...,m)
        Qjf = Rji ;
    Fin pour
    Fin pour
    nc = b - a ;
    Résultats : Qji (j = 0,...,m) et (i = 1,...,nc) nc nombre des courbes
    de degré (m = n - k) ;

```

Fin

Pour calculer les $(\lambda_{i,m,\alpha}, s_{m,\alpha})$ (Voir l'annexe).

III.2.2 Procédures algorithmique de la méthode de séparation des variables**Algorithme de réduction «S-variables »**

Données : P_i ($i = 0, \dots, n$) polygone de la courbe de degré n .

k : Ordre de la réduction.

α : Ordre de continuité.

Début

```

    Pour (i = 0,...,n)
        Wi = Pi ;
    Fin pour
    Pour (r = 1,...,k)
        Si α > 0 alors
            Q0 = W0 ; Qn-1 = Wn ;
            Pour (i = 1,...,α-1)
                Qi =  $\left(1 + \frac{i}{n-i}\right) \cdot W_i - \frac{i}{n-i} \cdot Q_{i-1}$  ;
            Fin pour
            Pour (i = n-1,...,n-α+1)
                Qi-1 =  $\left(1 + \frac{n-i}{i}\right) \cdot W_i - \frac{n-i}{i} \cdot Q_i$  ;
            Fin pour
        Calculer  $(\gamma_{i,h}, \beta_{i,h}, b_h)$  (voir Annexe A)

```

Résoudre le système linéaire $\sum_{i=\alpha}^{n-\alpha} Q_i \cdot \beta_{i,h} = b_h$

Fin Si

Pour ($i = 0, \dots, n-1$)

$W_i = Q_i$;

Fin pour

$n = n-1$;

Fin pour

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

Fin

Algorithme de Sub-réduction

Données : P_i ($i = 0, \dots, n$) pôles de la courbe de degré n .

α : Degré de continuité.

k : Ordre de la réduction.

N : Nombre de points de la courbe.

Début

$l = 0$;

(1) **S-variables**($P, k, \alpha \rightarrow Q$)

$eps = \text{Max}|P(t) - Q(t)|$

Si ($eps > \varepsilon$) alors

$t_1 = 0$; $t_2 = 1$;

Répéter

Répéter

Algo-S-variables1

Jusqu'à ce que ($eps \leq \varepsilon$)

S-variables($P2, k, \alpha \rightarrow Q2$)

$eps = \text{Max}|P2(t) - Q2(t)|$

Si ($eps > \varepsilon$) alors

Algo-S-variables2

Fin si

Jusqu'à ce que ($eps \leq \varepsilon$)

$l = l + 2$;

Pour ($i = 0, \dots, n-k$)

$Q_i^{l-1} = Q1_i$; $Q_i^l = Q2_i$;

Fin pour

Sinon

$l = l + 1$;

Pour ($i = 0, \dots, n-k$)

$Q_i^l = Q_i$;

Fin pour

Fin si

Résultats : Q_i^j ($j = 1, \dots, l$) ($i = 0, \dots, n-k$) l : nombre des courbes de degré ($m = n-k$)

Fin

Algo-S-variables1**Début**

```

 $t_0 = ((t_1 + t_2)/2) ;$ 
Sub-courbe ( $P, t_0 \rightarrow P1, P2$ )
S-variables ( $P1, k, \alpha \rightarrow Q1$ )
 $eps = \text{Max}|P1(t) - Q1(t)|$ 
Si ( $eps > \varepsilon$ ) alors ( $t_2 = t_0$ )

```

Fin**Algo-S-variables2****Début**

```

 $t_1 = t_0$ 
Si ( $t_2 - t_1 < (1/(N - 1))$ ) alors
   $l = l + 1 ;$ 
  Pour ( $i = 0, \dots, n - k$ )
     $Q_i^l = Q1_i ;$ 
  Fin pour
  Pour ( $i = 0, \dots, n$ )
     $P_i = P2_i ;$ 
  Fin pour
   $N = N \cdot (t_2 - t_1) ;$  aller à (1)
Fin si

```

Fin**Algorithme Sub-courbe**

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

Début

```

Pour ( $i = 0, \dots, n$ )
   $P_i^0(t_0) = P_i ;$ 
Fin pour
Pour ( $k = 1, \dots, n$ )
  Pour ( $i = 0, \dots, n - k$ )
     $P_i^k(t_0) = (1 - t_0)P_i^{k-1}(t_0) + t_0P_{i+1}^{k-1}(t_0) ;$ 
  Fin pour
Fin pour
Pour ( $i = 0, \dots, n$ )
   $P0_i = P_0^i(t_0), P1_i = P_i^{n-i}(t_0) ;$ 
Fin pour
Résultats :  $P0_i$  et  $P1_i (i = 0, \dots, n)$ 

```

Fin

III.3 Méthode de sub-réduction de surfaces mise en œuvre

III.3.1 Réduction de surface de Bézier

Soit une surface de Bézier $S(u, v)$ de degré n suivant u et m suivant v , si nous appliquons l'algorithme de réduction de courbe de Bézier selon les deux directions, nous obtiendrons l'algorithme de réduction d'une surface suivant :

Algorithme Red-surface

Données $S_{i,j}$ ($i = 0, \dots, n$) ($j = 0, \dots, m$) réseau caractéristique de la surface $S(u, v)$

k_1 : Ordre de la réduction suivant u .

k_2 : Ordre de la réduction suivant v .

α_1 : Facteur de continuité suivant u .

α_2 : Facteur de continuité suivant v .

Début

Pour ($j = 0, \dots, m$)

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

$P_i = S_{i,j}$;

Fin pour

S-variables($P, k_1, \alpha_1 \rightarrow Q$)

Pour ($i = 0, \dots, n - k_1$)

$C_{i,j} = Q_i$;

Fin pour

Fin pour

Pour ($i = 0, \dots, n - k_1$)

Pour ($j = 0, \dots, m$)

$P_j = C_{i,j}$;

Fin pour

S-variables($P, k_2, \alpha_2 \rightarrow Q$)

Pour ($j = 0, \dots, m - k_2$)

$C_{i,j} = Q_j$;

Fin pour

Fin pour

Résultats : $R_{i,j}$ ($i = 0, \dots, n - k_1$) ($j = 0, \dots, m - k_2$) réseau caractéristique de la surface réduite.

Fin

III.3.2 Algorithme de Sub-réduction de la surface

Données $S_{i,j}$ ($i = 0, \dots, n$) ($j = 0, \dots, m$) réseau caractéristique de la surface $S(u, v)$

k_1 : Ordre de la réduction suivant u .

k_2 : Ordre de la réduction suivant v .

α_1 : Facteur de continuité suivant u .

α_2 : Facteur de continuité suivant v .

ε : Tolérance de position.

N : Nombre de points de la courbe suivant u .

M : Nombre de points de la courbe suivant v .

Début

$L = 0$; $P = 0$; $iteration = 0$;

$Surface[0] = Surface_mer$;

Répéter

(1) **Red-surface**($Surface[iteration](u, v), k_1, k_2, \alpha_1, \alpha_2 \rightarrow Surface_reduite(u, v)$)

$eps = \max |Surface[iteration](u, v) - Surface_reduite(u, v)|$

Si ($eps > \varepsilon$) alors

$u_1 = 0$; $v_1 = 0$;

$u_2 = 1$; $v_2 = 1$;

Répéter

Répéter

$u_0 = (u_1 + u_2)/2$;

$v_0 = (v_1 + v_2)/2$;

Sub-surface($Surface[iteration](u, v), (u_0, v_0) \rightarrow Surface_sub[h](u, v), (h = 1, \dots, 4)$)

Red-surface ($Surface_sub[1](u, v), k_1, k_2, \alpha_1, \alpha_2 \rightarrow Surface_subreduit1(u, v)$)

$eps = \max |Surface_sub[1](u, v), Surface_subreduit1(u, v)|$

Si ($eps > \varepsilon$) alors

$u_2 = u_0$; $v_2 = v_0$;

Fin si

Jusqu'à ce que ($eps \leq \varepsilon$)

Red-surface($Surface_sub[2](u, v), k_1, k_2, \alpha_1, \alpha_2 \rightarrow Surface_subreduite2(u, v)$)

$eps = \max |Surface_sub[2](u, v), Surface_subreduite2(u, v)|$

Si ($eps > \varepsilon$) alors

$u_1 = u_0$; $v_1 = v_0$;

Si ($(u_2 - u_1) < 1/(N - 1) \& (v_2 - v_1) < 1/(M - 1)$) alors

$l = l + 1$;

Pour ($i = 0, \dots, n - k_1$)

Pour ($j = 0, \dots, m - k_2$)

$Surface_resultat[l]_{i,j} = Surface_subreduite[1]_{i,j}$

Fin pour

Fin pour

Pour ($w = 2, \dots, 4$)

$P = P + 1$;

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

Pour($j = 0, \dots, m$)

$Surface[P]_{i,j} = Surface_sub[w]_{i,j}$;

Fin pour

Fin pou

Fin pour

$N = N(u_2 - u_1)$; $M = M(v_2 - v_1)$;

$iteration = iteration + 1$;

Aller à (1)

Fin si

Fin si

```

    Jusqu'à ce que ( $\epsilon \leq \epsilon$ )
     $l = l + 1$  ;
    Pour ( $i = 0, \dots, n - k_1$ )
        Pour ( $j = 0, \dots, m - k_2$ )

             $Surface\_resultat[l]_{i,j} = Surface\_subreduite[1]_{i,j}$  ;

        Fin pour
    Fin pour
     $l = l + 1$  ;
    Pour ( $i = 0, \dots, n - k_1$ )
        Pour ( $j = 0, \dots, m - k_2$ )
             $Surface\_resultat[l]_{i,j} = Surface\_subreduite[2]_{i,j}$  ;
        Fin pour
    Fin pour
    Pour ( $w = 3, \dots, 4$ )
         $P = P + 1$  ;
        Pour ( $i = 0, \dots, n$ )
            Pour ( $j = 0, \dots, m$ )
                 $Surface[P]_{i,j} = Surface\_sub[w]_{i,j}$  ;
            Fin pour
        Fin pour
    Sinon
         $l = l + 1$  ;
        Pour ( $i = 0, \dots, n - k_1$ )
            Pour ( $j = 0, \dots, m - k_2$ )
                 $Surface\_resultat[l]_{i,j} = Surface\_reduite_{i,j}$  ;
            Fin pour
        Fin pour
    Fin si
     $iteration = iteration + 1$  ;
    Jusqu'à ce que ( $iteration \leq P$ )
    Résultats :  $Surface\_resultat[f]_{i,j}$  ( $i = 0, \dots, n - k_1$ ) ( $j = 0, \dots, m - k_2$ ) ( $f = 1, \dots, l$ )  $l$  :
    surfaces de degrés ( $n - k_1, m - k_2$ )
Fin

```

Algorithme de Sub-surface

Soit une surface de Bézier non rationnelle $S(u, v)$ de degré (n, m) , définie par son réseau caractéristique $S_{i,j}$ la subdivision selon les deux directions u et v donne l'algorithme suivante :

Début

```

Pour (i = 0, ..., n)
  Pour (k = 0, ..., m)
    C0,k = Si,k ;
  Fin pour
  Pour (k = 0, ..., m)
    Pour (h = 0, ..., m - k)
      Ck,h = (1 - v) · Ck-1,h + v · Ck-1,h+1 ;
    Fin pour
  Fin pour
  Pour (k = 0, ..., m)
    R1i,k = Ck,0 ;
    R2i,k = Ck,m-k ;
  Fin pour
Fin pour
Pour (j = 0, ..., m)
  Pour (k = 0, ..., n)
    C0,k = R1k,j ;
  Fin pour
  Pour (k = 0, ..., n)
    Pour (h = 0, ..., n - k)
      Ck,h = (1 - u) · Ck-1,h + u · Ck-1,h+1 ;
    Fin pour
  Fin pour
  Pour (k = 0, ..., n)
    S1k,j = Ck,0 ;
    S2k,j = Ck,n-k ;
  Fin pour
Fin pour
Pour (j = 0, ..., m)
  Pour (k = 0, ..., n)
    C0,k = R2k,j ;
  Fin pour
  Pour (k = 0, ..., n)
    Pour (h = 0, ..., n - k)
      Ck,h = (1 - u) · Ck-1,h + u · Ck-1,h+1 ;
    Fin pour
  Fin pour
  Pour (k = 0, ..., n)
    S3k,j = Ck,0 ;
    S4k,j = Ck,n-k ;
  Fin pour
Fin pour

```

Fin

III.4 Mise en œuvre du programme implémenté

Remarque : les résultats obtenus sont faites sur un Pentium 4 CPU 2.40GHz.

III.4.1 Cas des courbes

III.4.1.1 Réduction de courbes de Bézier non rationnelles

Soit une courbe de Bézier de degré $n=10$ réduite en degré $m=8$, avec une continuité C^2 , l'application du logiciel pour la réduction donne les résultats suivants :

- **Méthode de Eck**

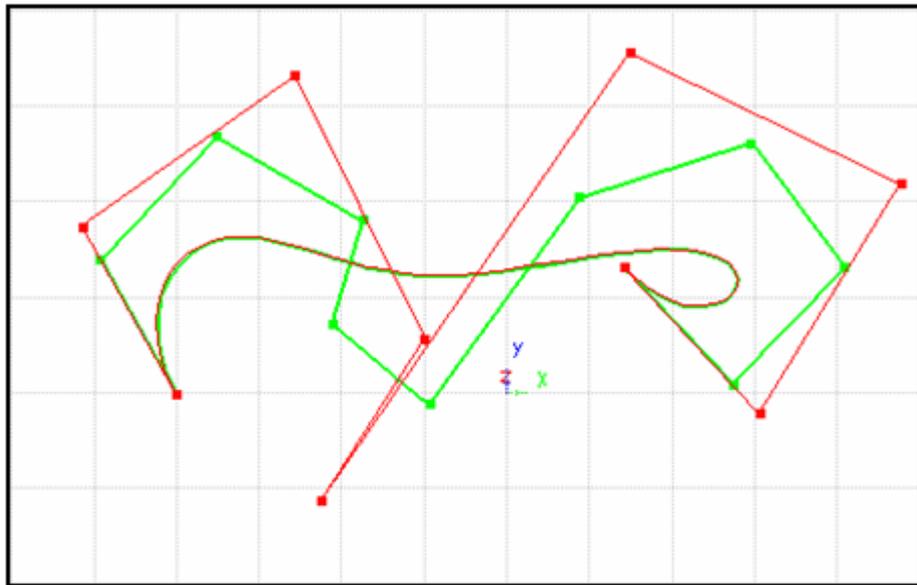


Figure III.2 Courbe de Bézier de degré 9 réduite en courbe de degré 6 avec une continuité $\alpha = 3$

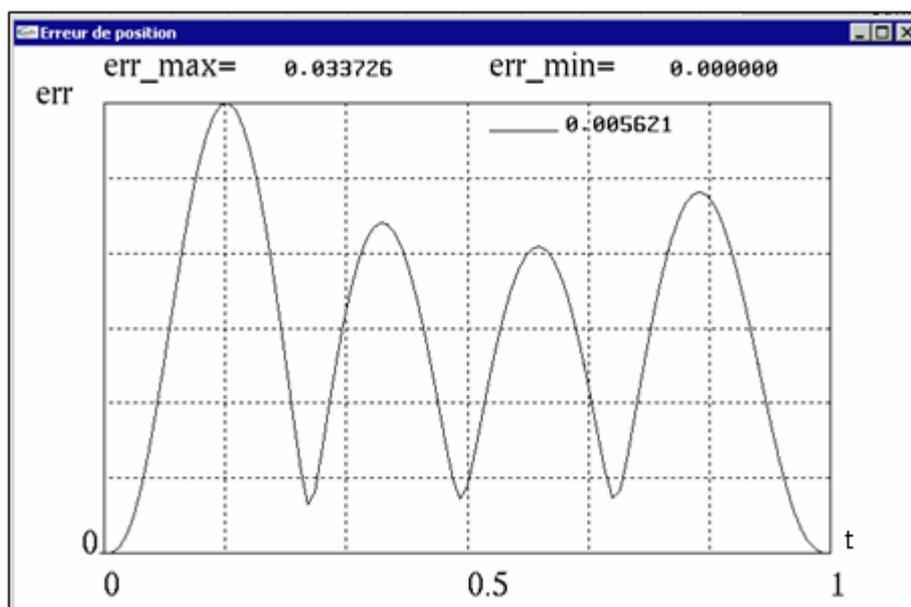


Figure III.3 Variation de l'erreur de position de la courbe de la figure III.2.

err_max= erreur de position maximale.

err_min= erreur de position minimale.

dk_max= variation de la courbure maximale.

dk_min= variation de la courbure minimale.

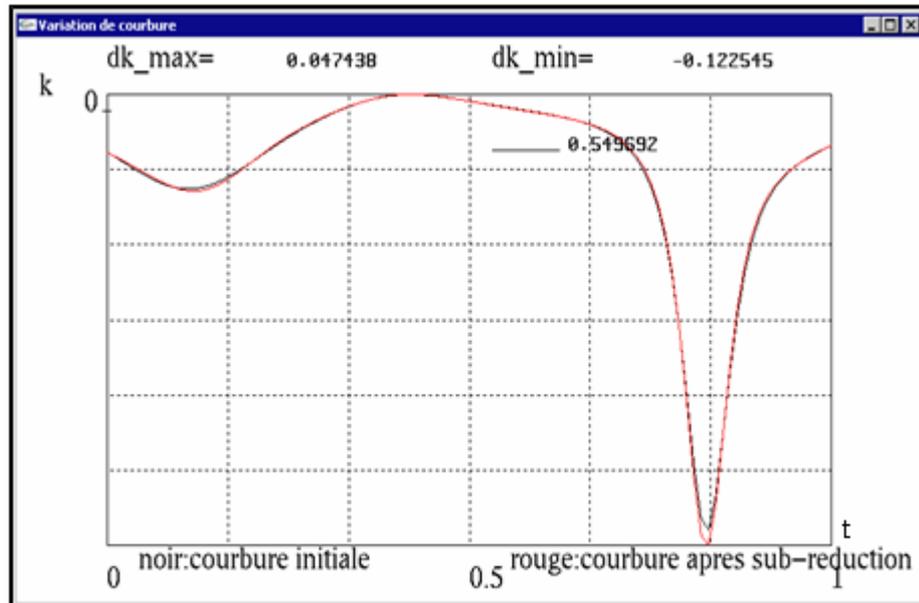


Figure III.4 Variation de la courbure de la courbe de la figure III.2.

- **Méthode de s-variables**

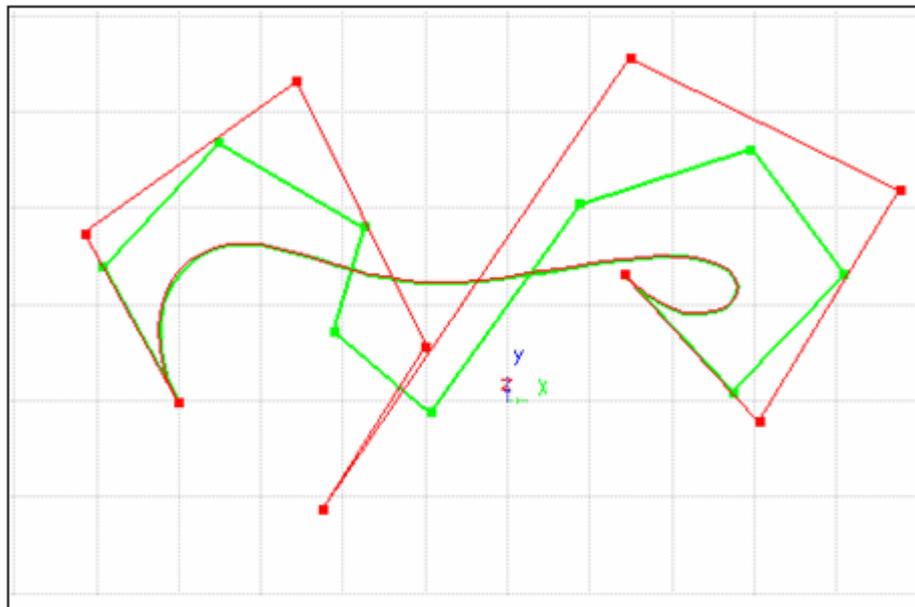


Figure III.5 Courbe de Bézier de degré 10 réduite en courbe de degré 8 avec une continuité $\alpha = 3$

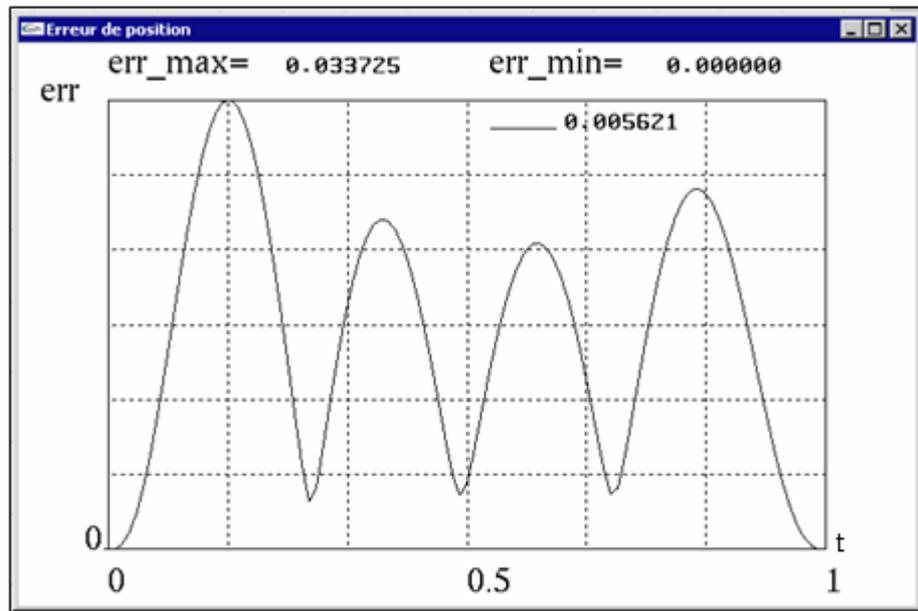


Figure III.6 Variation de l'erreur de position de la courbe de la figure III.5.

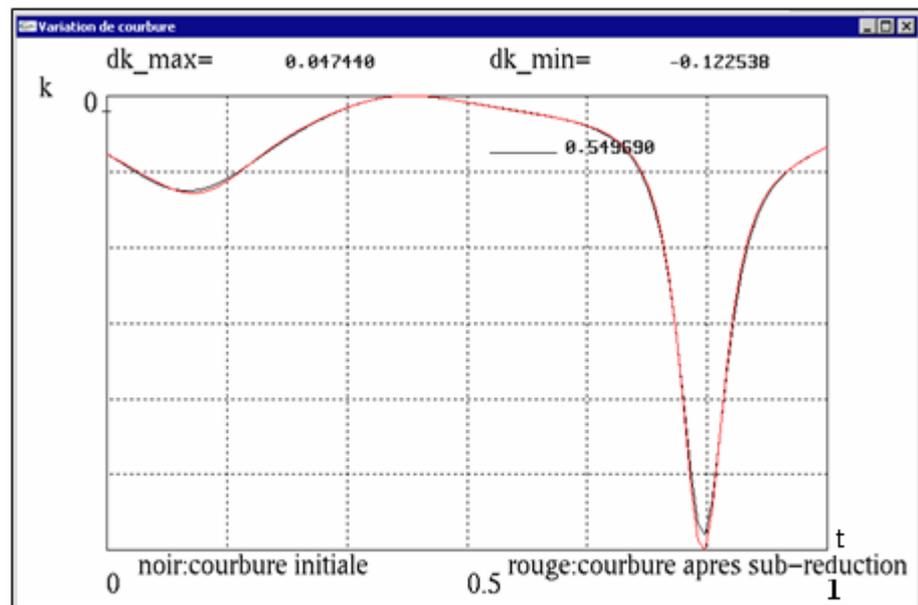


Figure III.7 Variation de courbure de la courbe de la figure III.5.

- **Tableau comparatif**

Méthode	Eck	S-Variables
Temps de calcul	0.016s	0.047s
Erreur de position	0.033756mm	0.033725mm
Variation de courbure	0.047438 (mm ⁻¹)	0.047440 (mm ⁻¹)

III.4.1.2 Sub-réduction de courbes de Bézier non rationnelles

Même courbe précédente, en fixant une tolérance de position $\varepsilon = 0.01mm$.

- **Méthode de Eck**

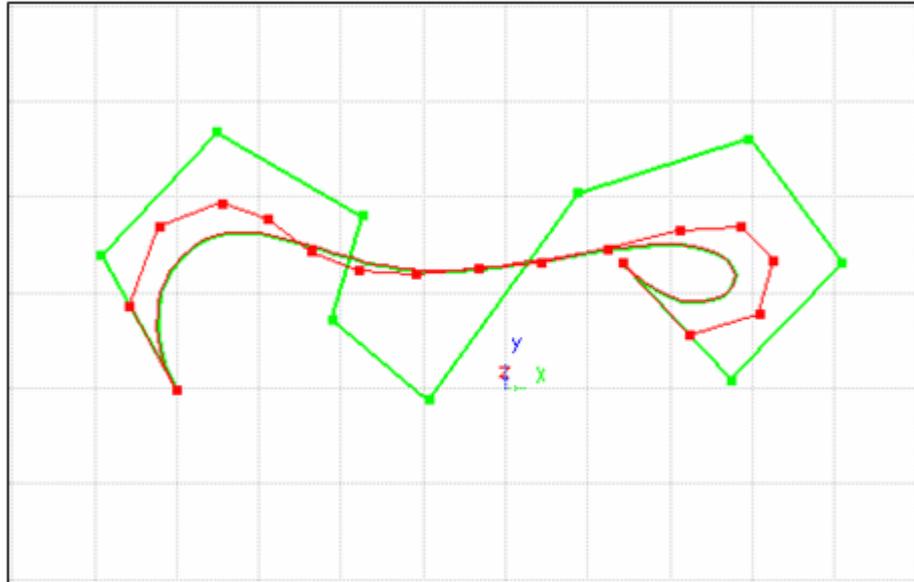


Figure III.8 Courbe de Bézier de degré 10 convertit en 2 courbes de degré 8 avec une continuité $\alpha = 3$ et $\varepsilon = 0.01mm$.

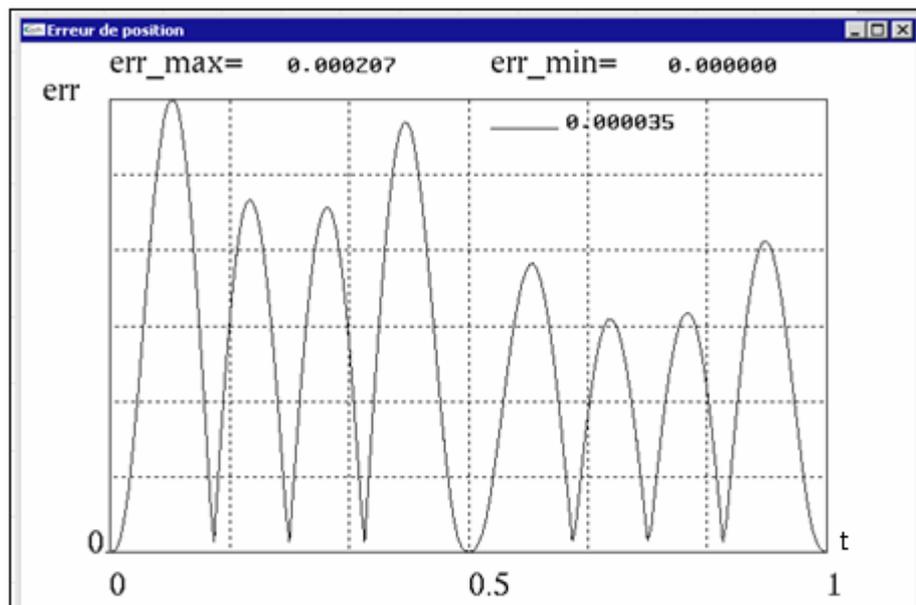


Figure III.9 Variation de l'erreur de position de la figure de la position III.8.

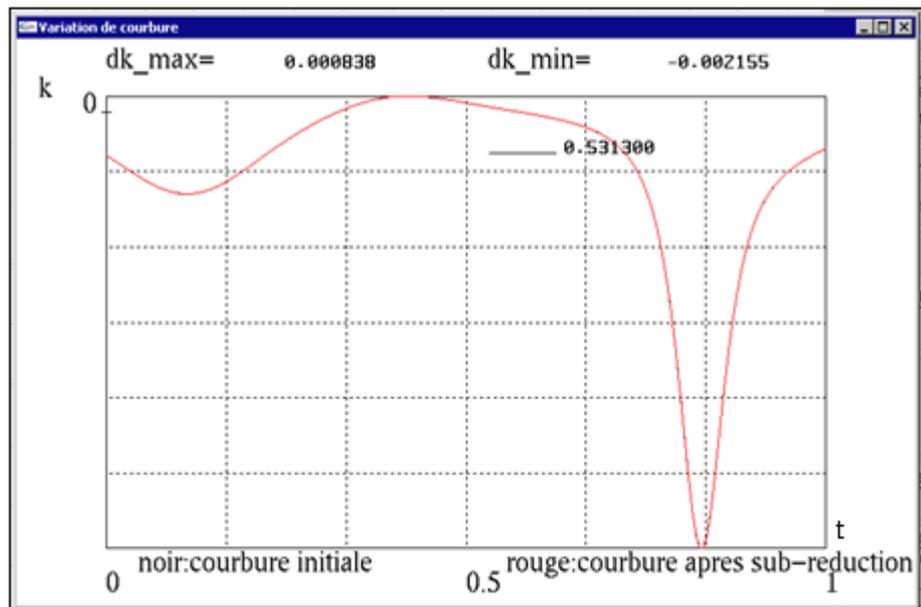


Figure III.10 Variation de la courbure de la courbe de la figure III.8.

- **Méthode de s-variables**

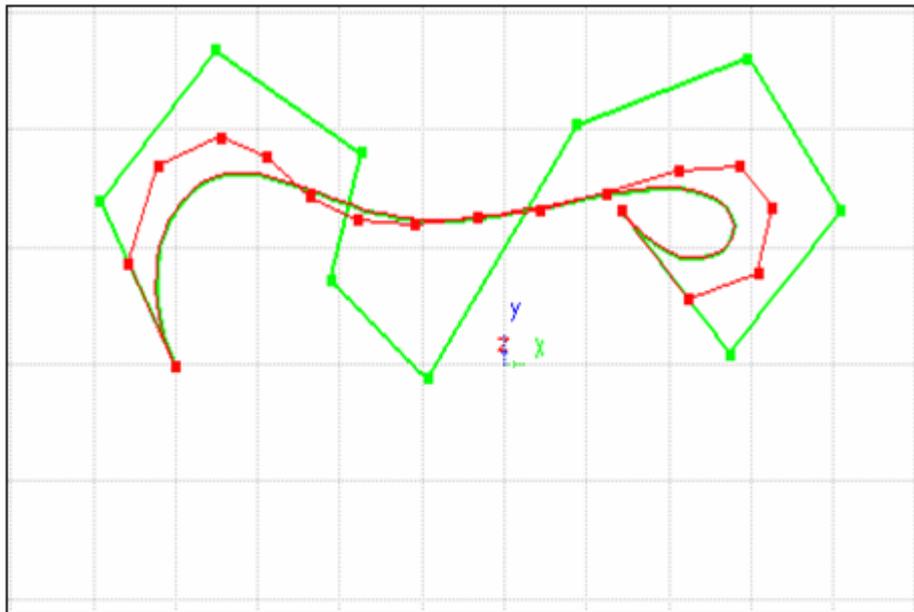


Figure III.11 Courbe de Bézier de degré 10 convertit en 2 courbes de degré 8 avec une continuité $\alpha = 3$ et $\varepsilon = 0.01mm$.

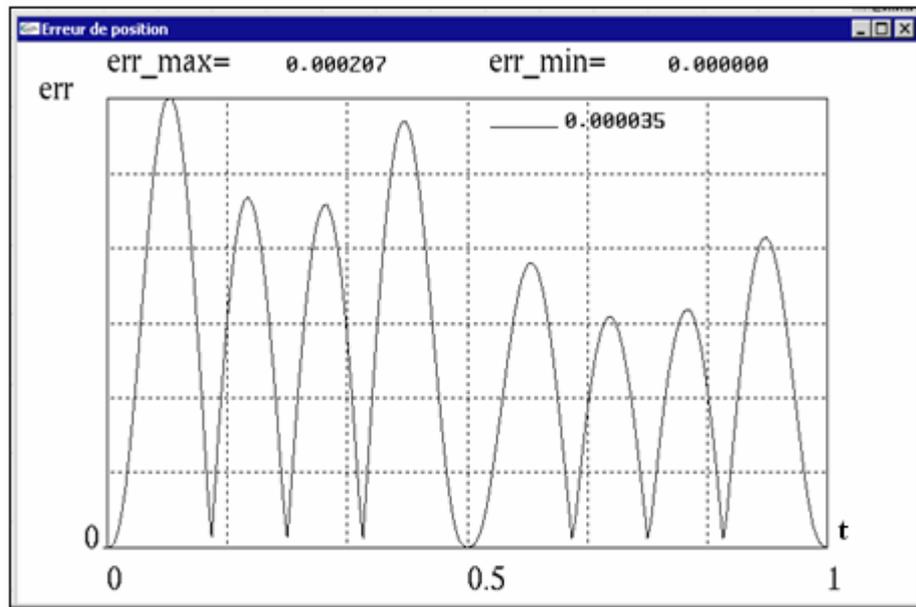


Figure III.12 Variation de l'erreur de position de la courbe de la figure III.11.

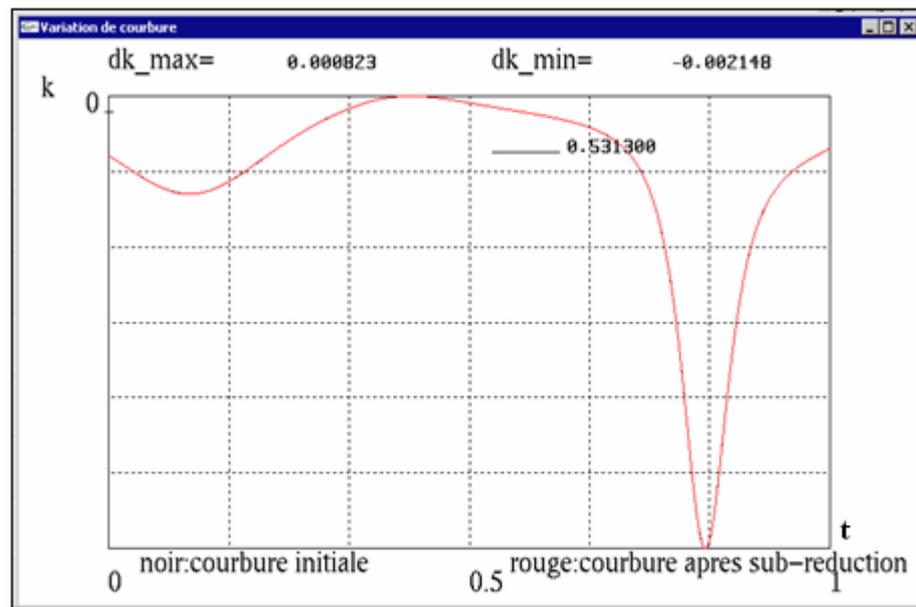


Figure III.13 Variation de la courbure de la courbe de la figure III.11.

- **Tableau comparatif**

Méthode	Eck	S-Variables
Nombre de courbe	2	2
Temps de calcul	0.047 s	0.172 s
Erreur de position	0.000207 mm	0.000207 mm
Variation de courbure	0.000838 (mm ⁻¹)	0.000823 (mm ⁻¹)

III.4.2 Cas des surfaces

- **Test01**

Soit une surface de Bézier de degré (9,9) convertit en quatre (04) surfaces de degré (7,7) pour une continuité C^1 suivant les deux directions et $\varepsilon = 0.03mm$.

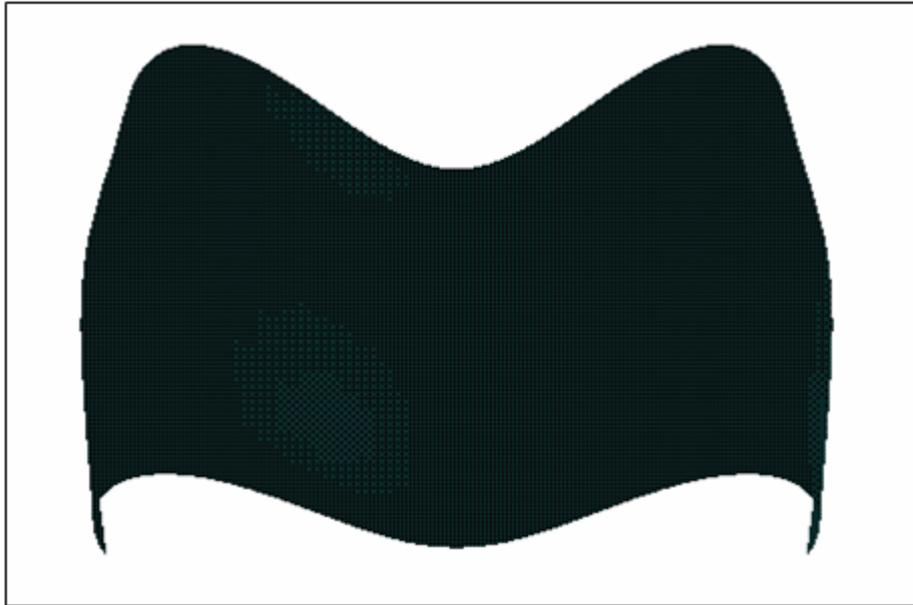


Figure III.14 Surface de Bézier de degré (9,9).

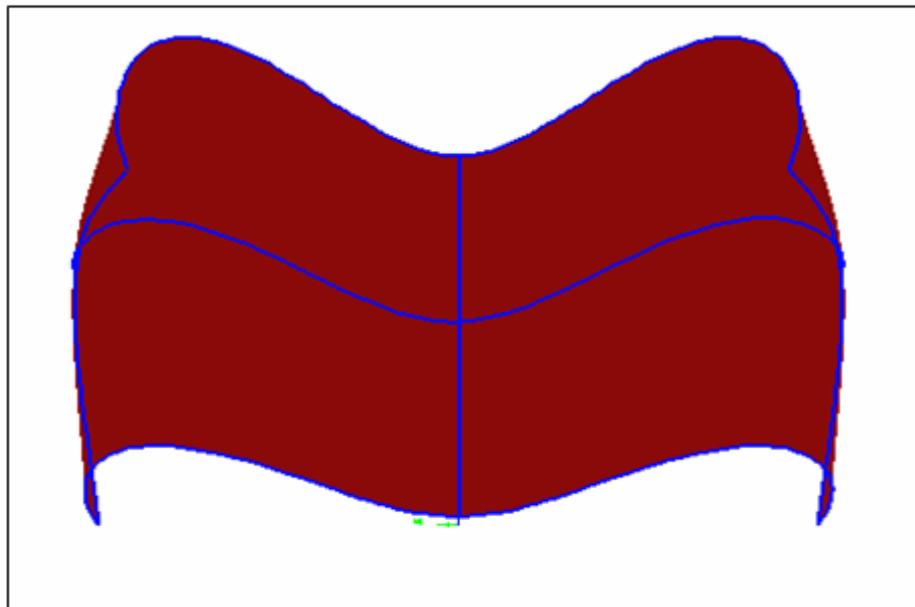


Figure III.15 Les 4 surfaces de Bézier de degré (7,7) après la conversion.

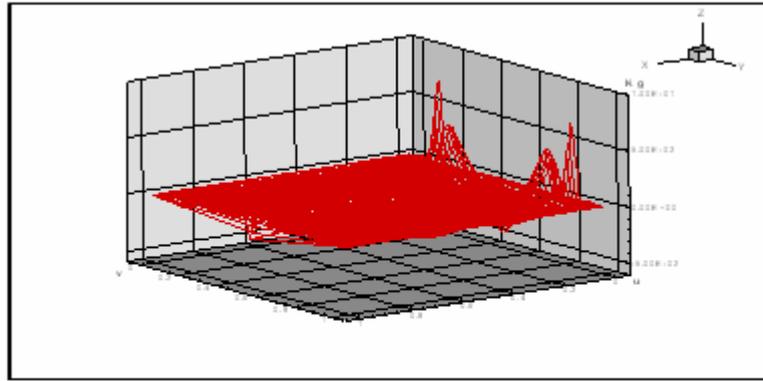


Figure III.16 Variation de la courbure gaussienne de la surface de la figure III.14.

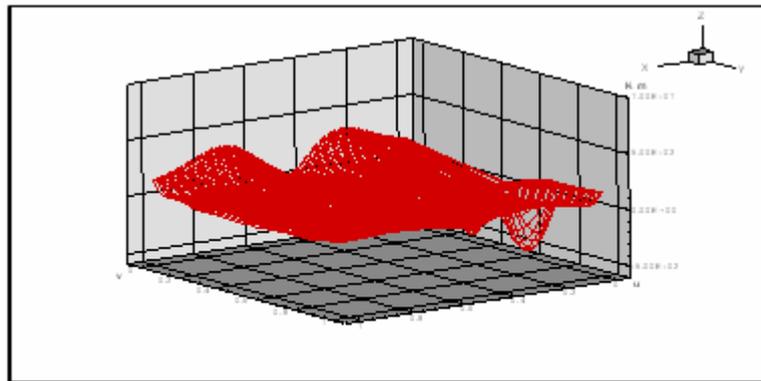


Figure III.17 Variation de la courbure moyenne de la surface de la figure III.14.

Le temps du calcul $T = 65.188s$ et le nombre des surfaces obtenues $ns = 4$.

- **Test02**

Soit une surface de Bézier de degré (4,4) convertit en des surfaces de degré (3,3), pour une continuité $\alpha = 1$ et $\varepsilon = 0.01mm$

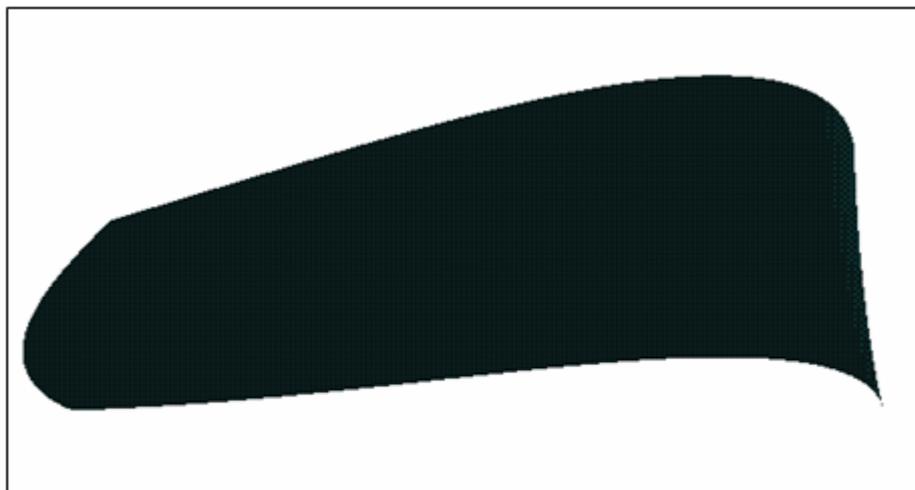


Figure III.18 Surface de Bézier de degré (4,4).

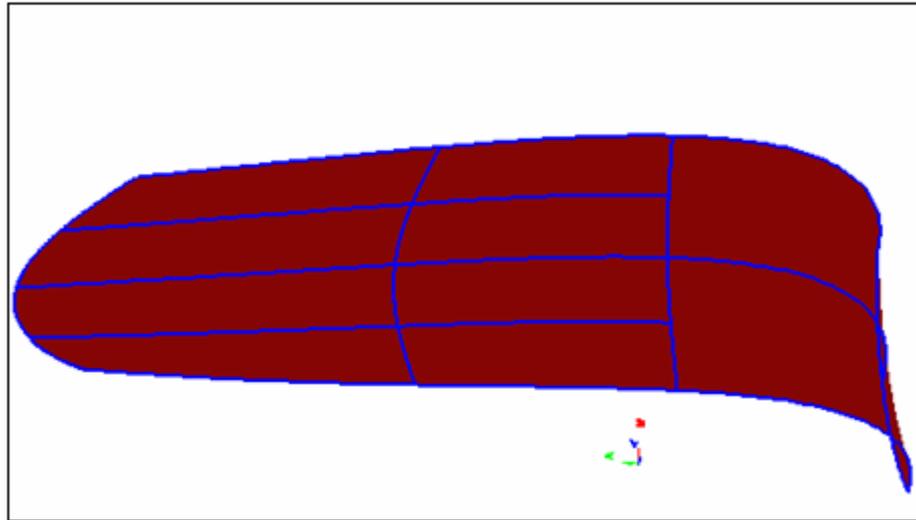


Figure III.19 Les 10 de surfaces de Bézier de degré (3,3) après la conversion.

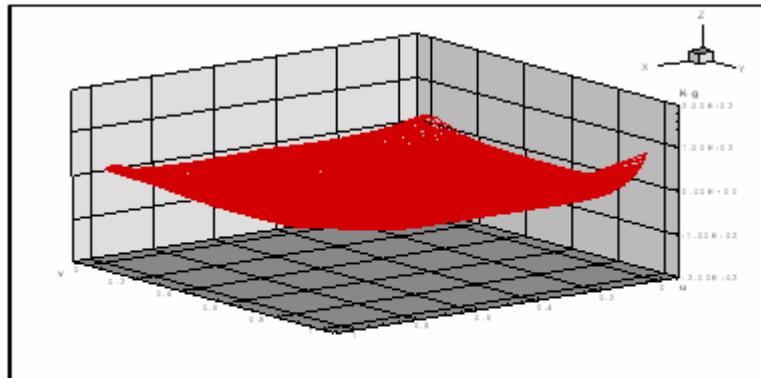


Figure III.20 Variation de la courbure gaussienne de la surface de la figure III.18.

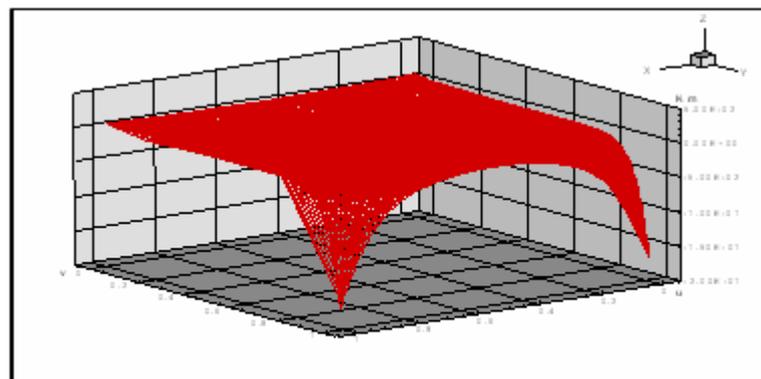


Figure III.21 Variation de la courbure moyenne de la surface de la figure III.18.

Le temps de calcul $T = 37.078s$.

Le nombre des surfaces obtenues $ns = 10$.

III.5 Modification locale de la surface de Bézier

Pour ce cas, nous prenons la surface de la Figure (III.18), avec une modification locale au niveau du pôle $P_{2,2}$, alors nous obtenons la surface suivante :

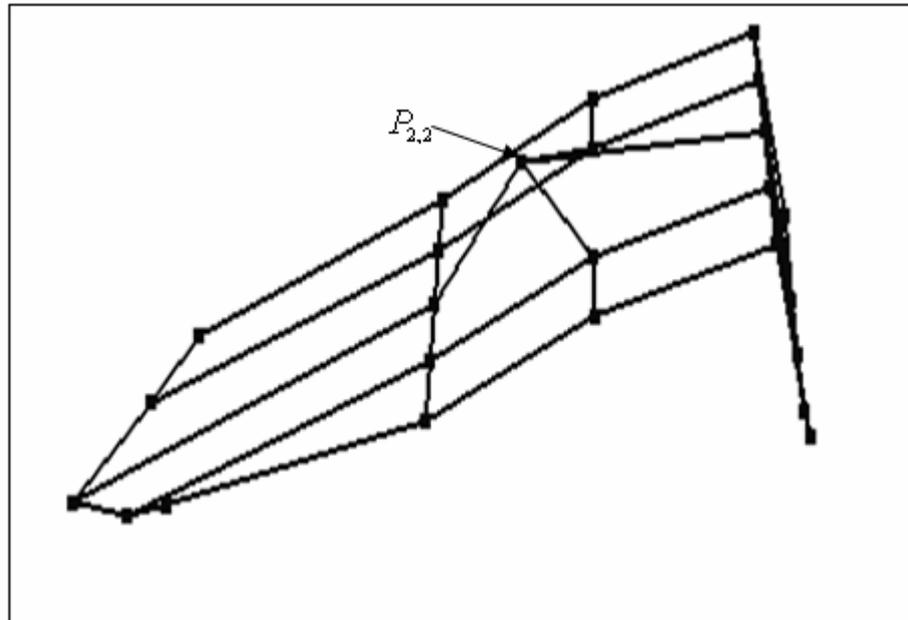


Figure III.22 Réseau caractéristique de la surface de la figure III.17 après la modification locale.

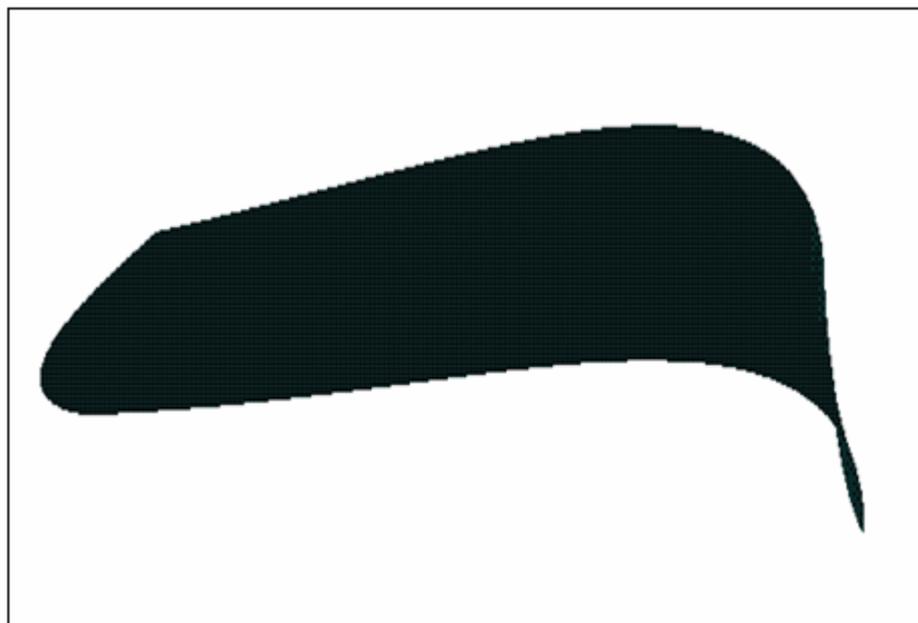


Figure III.23 Surface de Bézier de degré (4,4).

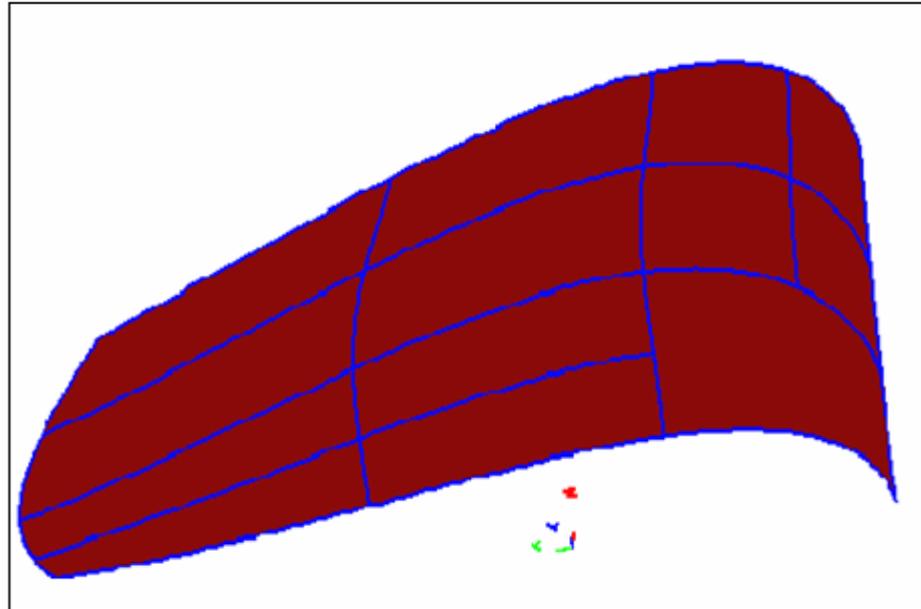


Figure III.24 Les surfaces de Bézier de degré (3,3) après la conversion.

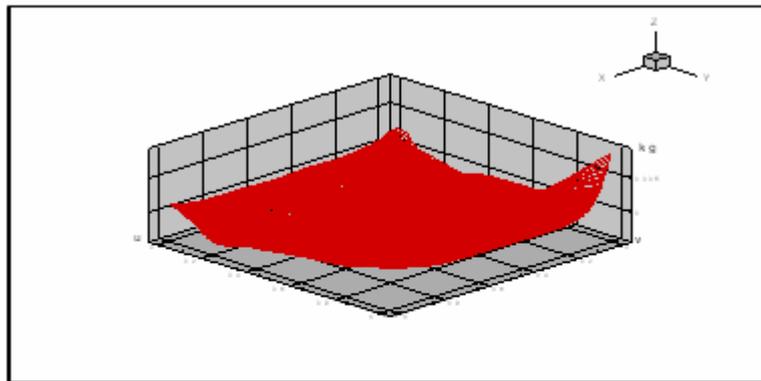


Figure III.25 Variation de la courbure gaussienne de la surface de la figure III.22.

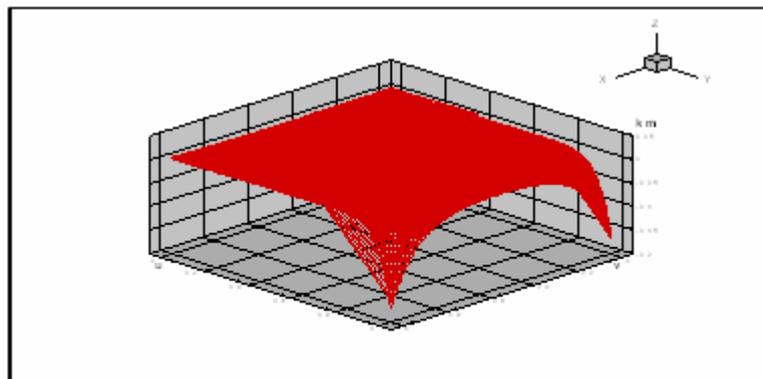


Figure III.26 Variation de la courbure moyenne de la surface de la figure III.22.

Temps de calcul =47,703s.

Nombres des surfaces obtenues =13.

III.6 Conclusion

D'après ces tests, la méthode de S-variables assure une erreur sur la courbure plus faible par rapport à la méthode de Eck. Cette propriété est plus intéressante en CFAO, en dépit d'un temps de calcul plus élevé. Dans le cas de surfaces, nous remarquons que la variation de deux grandeurs de la courbure gaussienne et de la courbure moyenne sur les courbes frontières d'une surface est nulle.

Chapitre IV : Résultats et interprétations

Dans ce chapitre nous allons étudier l'influence du facteur de réduction k et le degré de la courbe n sur la sub-réduction d'une courbe de Bézier non rationnelle.

IV.1 Etude de l'influence du facteur de réduction

Soit une courbe de Bézier de degré $n=18$ convertie en des courbes de degré $m = n - k$ tel que $k = 1, \dots, 11$, pour une continuité C^2 et $\varepsilon = 0.01\text{mm}$.

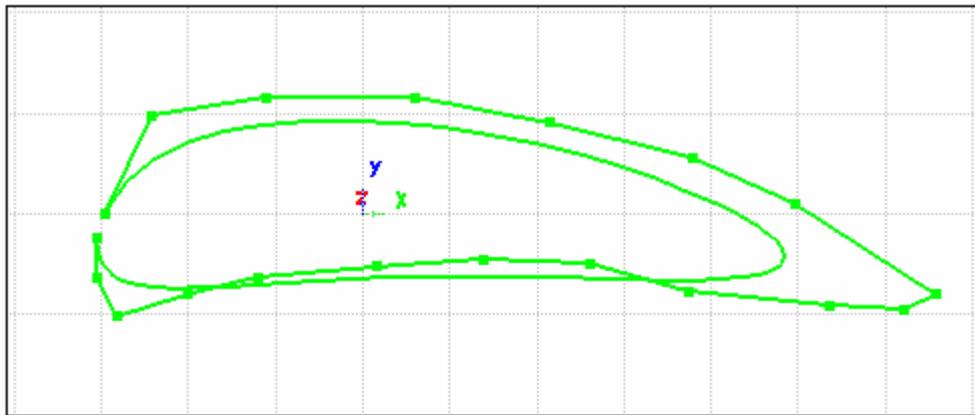


Figure IV.1 Courbe de Bézier de degré 18.

- **Tableau des résultats obtenus**

Méthode	Eck				s-variables			
	Nombre des courbes	Temps de calcul (s)	Erreur de position (mm)	Erreur de courbure (mm^{-1})	Nombre des courbes	Temps de calcul (s)	Erreur de position (mm)	Erreur de courbure (mm^{-1})
n-11	7	0.078	7.3E-05	8.43E-04	3	4.719	9.902E-03	0.012723
n-10	5	0.062	7.3E-05	8.51E-04	2	1.000	6.234E-03	0.033063
n-9	4	0.062	1.83E-04	4.479E-03	2	0.937	1.687E-03	0.011032
n-8	4	0.062	1.83E-04	4.551E-03	2	0.890	3.01E-04	2.683E-03
n-7	2	0.047	1.84E-04	4.555E-03	2	0.844	8E-05	3.102E-03
n-6	2	0.047	1.93E-04	3.424E-03	1	0.313	4.67E-03	0.042685

n-5	2	0.047	1.92E-04	3.497E-03	1	0.266	6.6E-04	0.016297
n-4	1	0.047	1.91E-04	3.496E-03	1	0.234	8.12E-04	7.711E-03
n-3	1	0.031	1.64E-04	1.9993E-03	1	0.219	1.173E-03	6.245E-03
n-2	1	0.032	3.7E-05	3.76E-04	1	0.188	3.769E-03	0.039293
n-1	1	0.016	7E-06	9.1E-05	1	0.140	9.07E-04	0.010686

Tableau IV.1 Réduction du degré n au degré $n - k$.

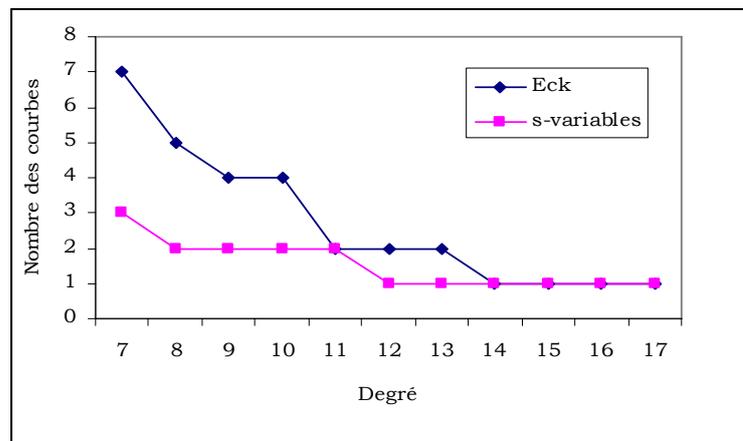


Figure IV.2 Variation de nombre des courbes en fonction du degré n .

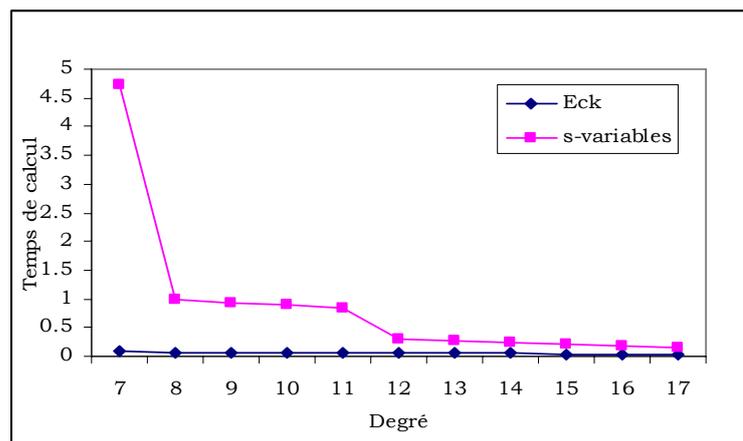


Figure IV.3 Variation de Temps de calcul en fonction du degré n .

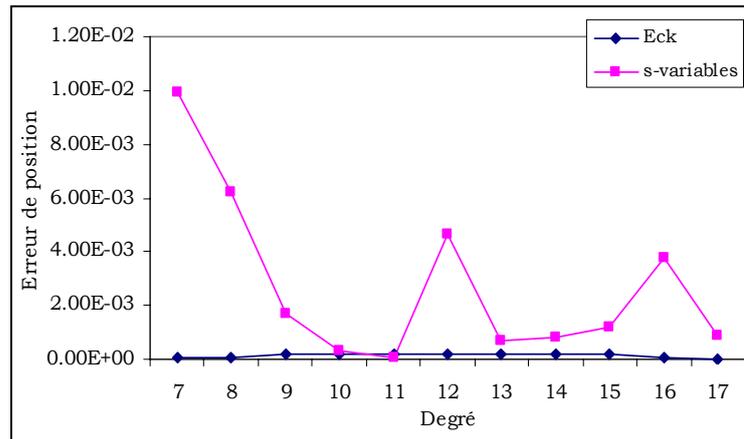


Figure IV.4 Variation d'erreur de position en fonction du degré n .

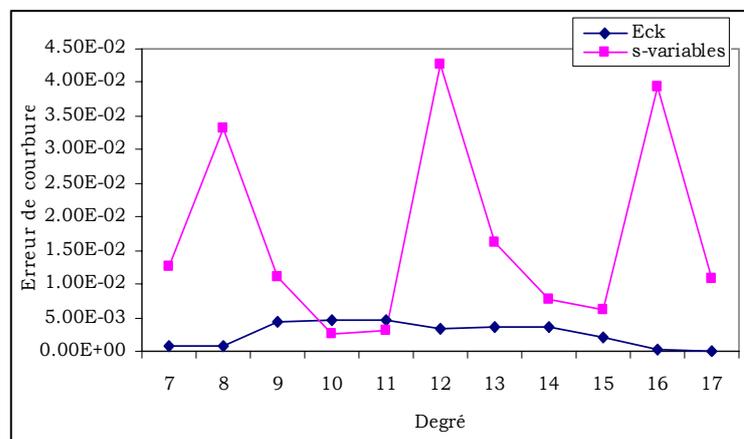


Figure IV.5 Variation de la courbure en fonction du degré n .

Les figures (IV.2) , (IV.3), (IV.4) et (IV.5) montrent les résultats comparés des temps de traitement et des erreurs d'approximation pour une tolérance de position de $\varepsilon = 0.01mm$ et une continuité géométrique d'ordre 2, en fonction d'un facteur de réduction $k = 1, \dots, 11$ pour une degré $n = 18$.

IV.2 Etude influence du degré de la courbe

Après la Sub-réduction des courbes de Bézier différentes (de degré variable) pour une continuité C^2 et $\varepsilon = 0.01mm$. Les résultats obtenus présenter dans le tableau suivant :

Méthode	Eck				S-Variables			
	Nombre des courbes	Temps de calcul (s)	Erreur de position (mm)	Erreur de courbure (mm^{-1})	Nombre des courbes	Temps de calcul (s)	Erreur de position (mm)	Erreur de courbure (mm^{-1})
18	1	0.031	1.3E-04	2.91E-04	1	0.203	1.354E-03	1.308E-03
17	1	0.016	1.96E-04	4.73E-04	1	0.203	5.6E-04	4.73E-04
16	2	0.031	3.29E-04	8.73E-04	1	0.187	1.441E-03	2.717E-03
15	2	0.031	4.5E-04	2.587E-03	1	0.172	1.563E-03	2.465E-03
14	2	0.031	4.1E-05	7.1E-05	1	0.156	8.886E-03	1.6445E-02
13	2	0.031	1.34E-04	1.92E-04	2	0.359	1.2E-04	2.92E-04
12	2	0.046	6.76E-04	8.93E-04	2	0.328	5E-04	1.73E-04
11	2	0.032	7.3E-04	1.57E-03	2	0.281	3.82E-04	1.98E-03
10	4	0.047	8.4E-05	8.21E-04	2	0.281	1.102E-03	1.177E-03
9	4	0.047	3.16E-04	6E-04	2	0.219	3.533E-03	1.5126E-02
8	4	0.031	4.26E-04	1.102E-03	3	0.641	9.951E-03	8.391E-03

Tableau IV.2 Réduction du degré n au degré $n - 3$.

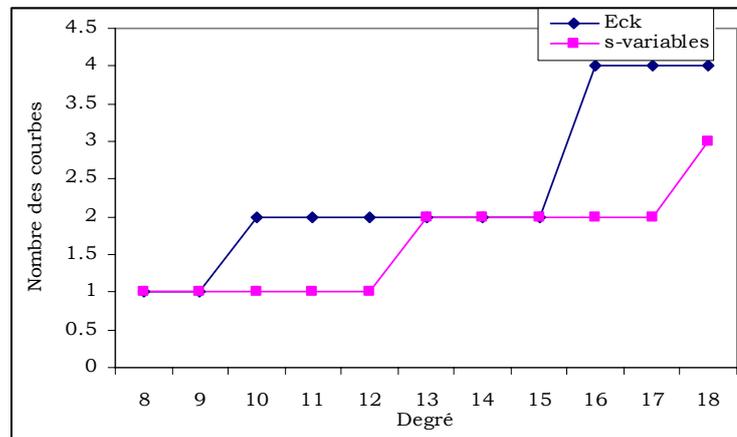


Figure IV.6 Variation de nombre des courbes en fonction du degré n .

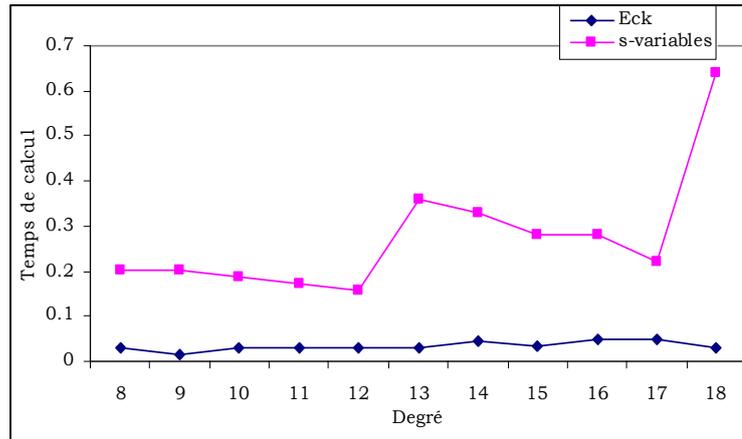


Figure IV.7 Variation de temps de calcul en fonction du degré n.

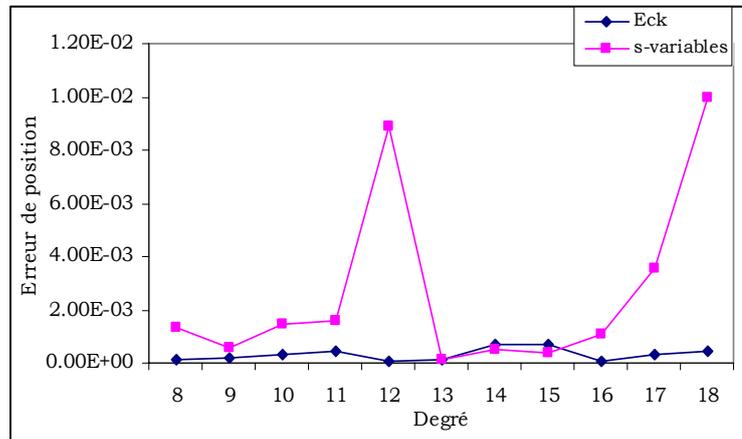


Figure IV.8 Variation d'erreur de position en fonction du degré n.

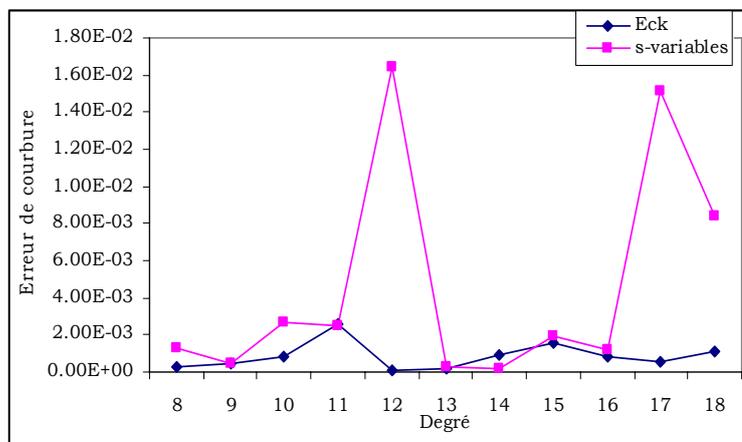


Figure IV.9 Variation de la courbure en fonction du degré n.

Les figures (IV.6), (IV.7), (IV.8) et (IV.9) montrent les résultats comparés des temps de traitement et des erreurs d'approximation pour une tolérance de position de $\varepsilon = 0.01mm$ et une continuité géométrique d'ordre 2, en fonction de degré $n = 8, \dots, 18$ pour un facteur de réduction $k = 3$.

IV.3 Influence de pseudo-modification locale

Si nous déplaçons uniquement un des pôles P_i du polygone de contrôle, alors la courbe est principalement affectée par ce changement dans la région autour des points où le paramètre prend la valeur i/n .

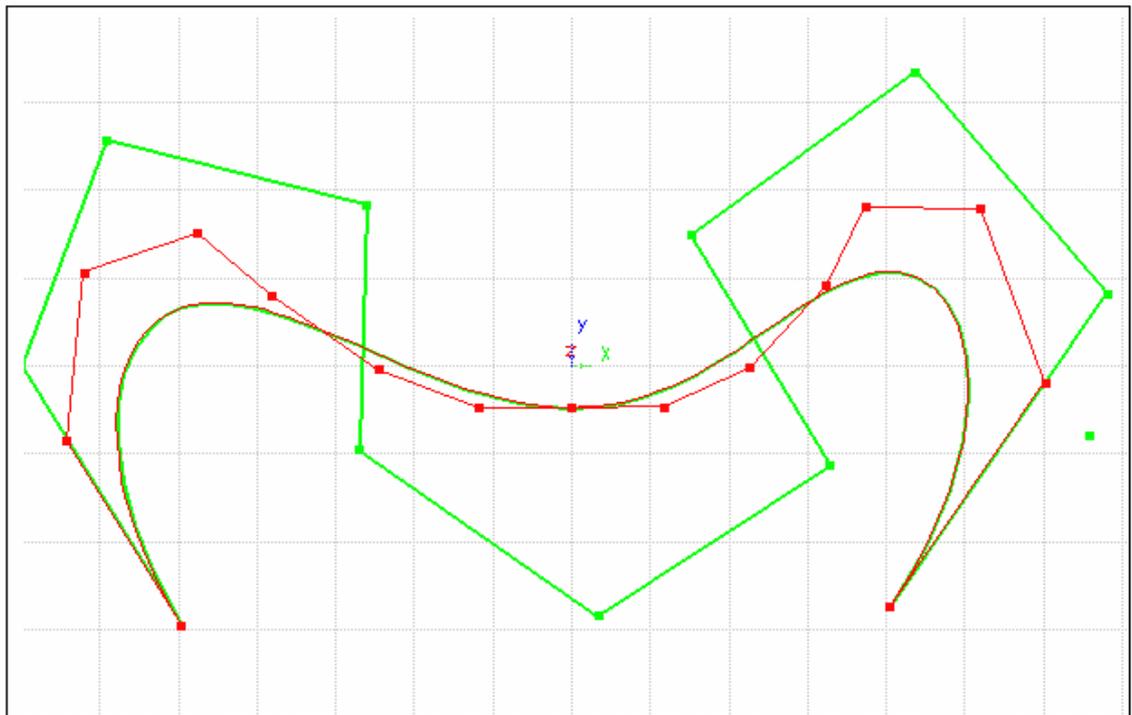


Figure IV.10 Courbe de Bézier de degré 10 réduite en deux courbes de degré 7 avec $\alpha = 3$ et $\varepsilon = 0.01mm$.

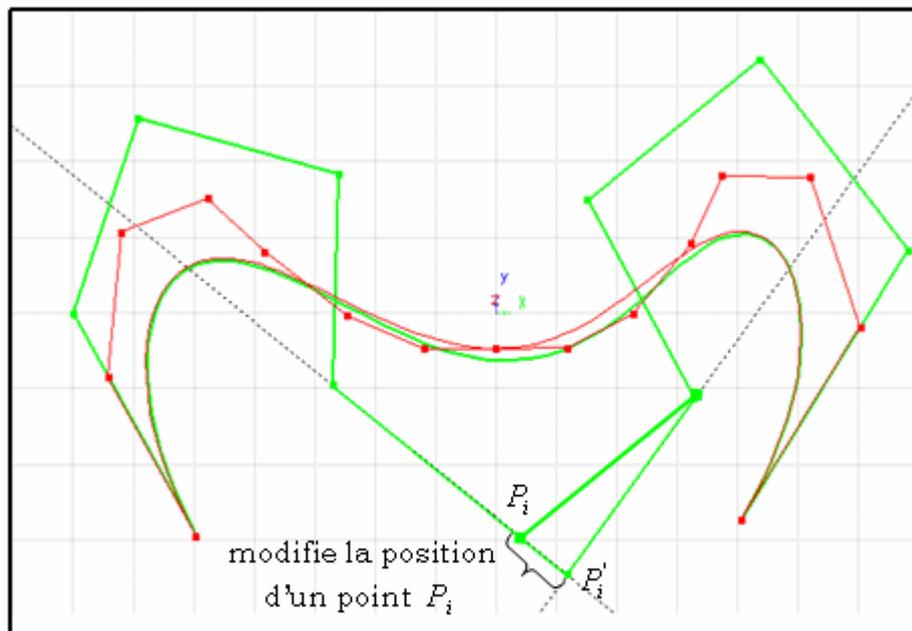


Figure IV.11 Courbe de Bézier de degré 10 réduite en deux courbes de degré 7 avec $\alpha=3$ et $\varepsilon=0.01mm$ après modification locale.

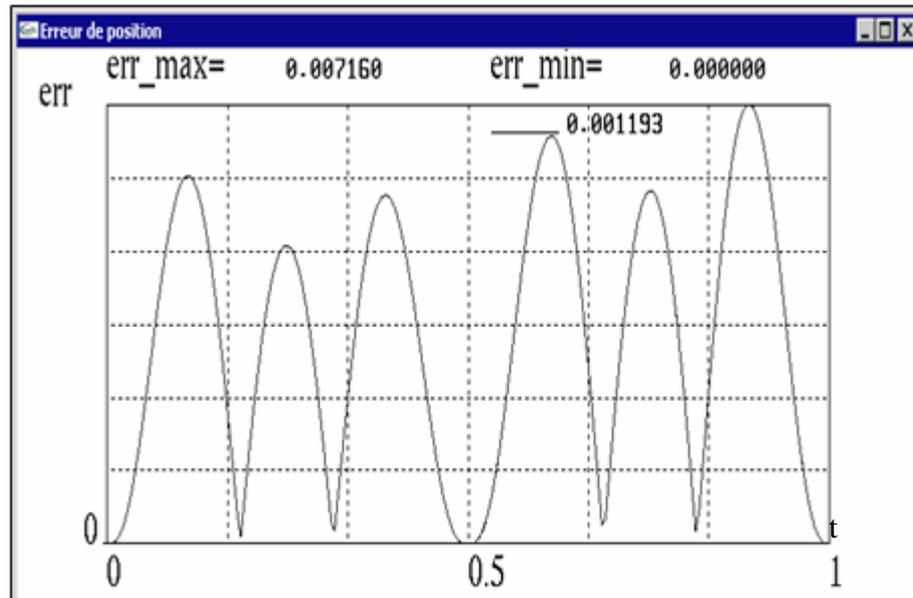


Figure IV.12 Variation de l'erreur de position de la courbe de la figure IV.10.

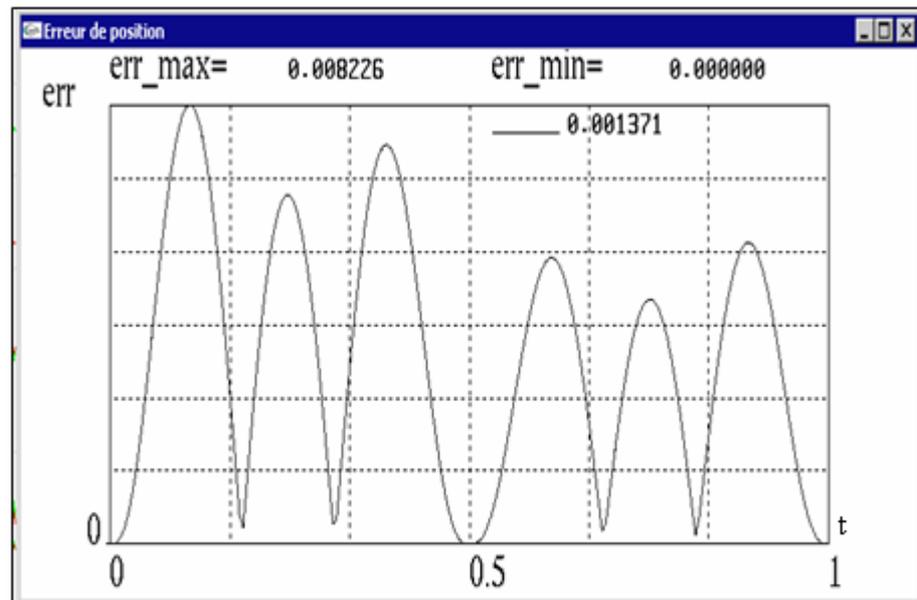


Figure IV.13 Variation de l'erreur de position de la courbe de la figure IV.11.

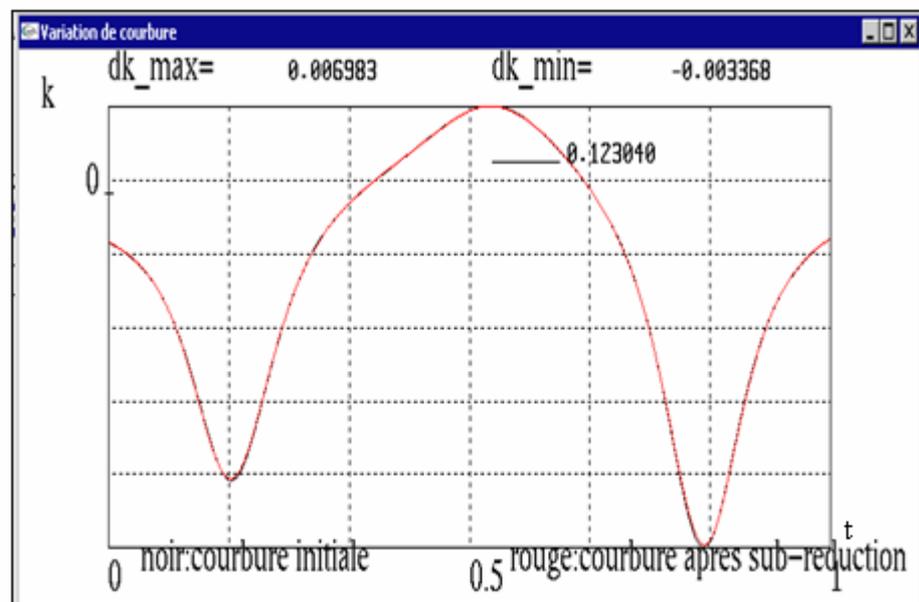


Figure IV.14 Variation de courbure de la courbe de la figure IV.10.

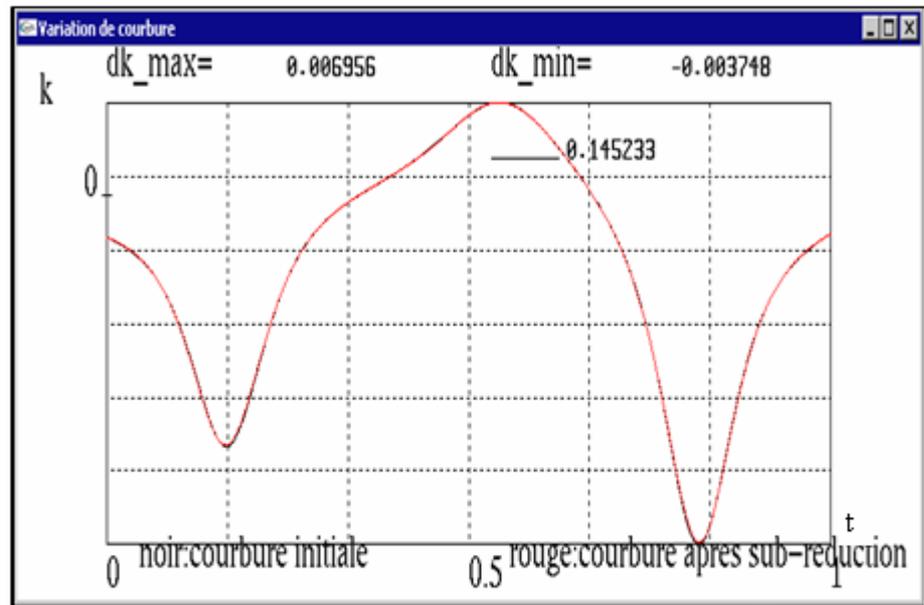


Figure IV.15 Variation de courbure de la courbe de la figure IV.11.

IV.4 Interprétation des résultats

D'après les résultats expérimentaux, nous remarquons que la méthode proposée par Asma [02] donne après l'opération de la Sub-réduction un nombre des courbes moins par rapport la méthode de Eck, mais le temps de calcul légèrement plus grand.

On obtient une erreur d'approximation minimale pour un nombre des courbes obtenues après la Sub-réduction plus grand, la même chose pour la courbure, d'autre part si l'erreur de position fixée est très faibles, ce qui donne par conséquent un nombre des courbes très élevé.

Le cas $n = 2 \cdot \alpha$ exclu dans la méthode de Eck et de séparation des variables. Donc on peut conclure que l'algorithme proposé donne une approximation très proche de celle de Eck et possède presque les mêmes caractéristiques. Pour aboutir à une conclusion finale, il est clair de voir l'influence de l'erreur de position proposée sur le nombre de courbes réduites et le temps de calcul pour respecter une continuité géométrique désirée.

Conclusion Générale

Conclusion Générale

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la recherche de méthodes mathématiques, algorithmiques et informatiques dans la perspective d'assurer la conversion par approximation de courbes et surfaces de degré et de modèles différents, tels qu'utilisés usuellement dans les systèmes de CFAO actuels. En effet, l'intérêt consiste à permettre à ces derniers une certaine universalité, voir une standardisation dans l'échange et la communication de données techniques.

La conversion par approximation est aussi envisageable pour les besoins d'élévation de la flexibilité des modèles de Bézier en particulier, pour les besoins de compatibilité entre outil et la pièce lors de la génération de trajectoires d'outils en FAO, pour les besoins de simplification des données géométriques des modèles de CAO, ...

Nous avons introduit dans ce mémoire en premier lieu des notions essentielles sur les problèmes d'échange de données techniques, suivi d'un aperçu général sur les développements mathématiques et algorithmiques des méthodes de conversion par approximations publiées.

Dans une deuxième étape, nous avons élaboré et implémenté en langage C++ un programme informatique, basé sur deux méthodes de réduction de degré des courbes et surfaces de Bézier, à savoir :

- Méthode basée sur le principe inverse de l'algorithme d'élévation de degré pour la réduction de degré, avec une détermination des pôles par séparation des variables, et la minimisation de l'erreur d'approximation au sens des moindres carré, désigné par « Méthode de réduction de degré par séparation des variables »,
- Méthode basée sur le principe inverse de l'algorithme d'élévation de degré pour la réduction et la minimisation de l'erreur par application des polygones orthogonaux au sens de Tchebychev, ou de Legendre , désignée par « Méthode de Eck » ,
- La validation a été entreprise sur des modèles différents de courbes et surfaces de Bézier choisies, où nous avons pu établir une comparaison entre les deux méthodes au sens précision, temps de calcul et nombre de subdivision obtenus, sous l'influence de certains critères déterminants, tels que le degré, l'erreur de position imposée, l'ordre de continuité, le degré de l'éléforme,....

A la lumière de nos résultats, nous avons constaté que la méthode de réduction de degré par séparation des variables est bien adaptée pour une éventuelle intégration dans les systèmes de CFAO, de part les propriétés mathématiques et technologiques qu'elle offre, à savoir : flexibilité et facilité de passage vers d'autres modèles, facilité de mise en œuvre informatique, indépendance du coefficient de réduction, réduction de degré en une seule étape. Toutefois, cela reste subordonné à une amélioration au sens optimisation des développements mathématiques et des algorithmes mis en œuvre.

Le travail que nous avons réalisé ne constitue en effet qu'une contribution initiale dans le domaine des échanges de données et communication entre systèmes de CFAO et il mérite d'être poursuivi dans la perspective d'une intégration effective, dans les systèmes de CFAO, des méthodes de conversion entre différents modèles de représentations des formes généralement utilisés. A cet effet, certaines extensions sont envisageables, en s'appuyant sur les développements que nous avons mis en œuvre dans ce mémoire, et relatifs :

- Au passage dans un sens et inversement aux autres modèles de courbes et surfaces (modèles Bézier rationnel, B-Splines, NURBS, ...), avec une possibilité d'optimisation des programmes au sens algorithmique et implémentation,
- A l'utilisation d'une plateforme de programmation, avec un noyau unifié et homogène (opencascade par exemple) ou utilisation des MFC (Microsoft foundation Classes) pour la conception de l'interface dans le but de réaliser une application aussi performante et fiable que possible.

*Références
Bibliographiques*

Références Bibliographiques

- [01]Ahn Y.J « Using Jacobi polynomials for degree reduction of Bezier curves with C^k -constraints» *Computer Aided Geometric Design* 20(2003) 423-434.
- [02]Asma F, « Méthodes de réduction de degré des courbes de Bézier. Etude comparative et Application à l'échange de données entre systèmes de CFAO», Mémoire de Magister, Département de Mécanique, Faculté des Sciences de l'Ingénieur, Université de Boumerdès, Algérie, Juil.1999.
- [03]Amirat S et Tahmi R « Modification locale des surfaces de Bézier non rationnelles » PFE, Promoteur : Dr. Idir BELAIDI, Boumerdes 2002.
- [04]Asma F, Belaidi I, Ishiomin G et Coffignal G « Réduction de degré des courbes de Bézier de degré élevé : une étude comparative des méthodes publiées» ,
- [05]Baron L « Cours 4-5 : Echange IGES-STEP»,Ecole Polytechnique Montréal, Hiver 2003.
- [06]BENAMAR A « Modélisation de courbes sur les surfaces splines » thèse de doctorat, Université PIERRE ET MARIE CURIE, Paris 1994.
- [07]BEN SALAH A « Equivalences et algorithmique des transformations dans les modèles mathématique pour la conception et la fabrication assistées par ordinateur des courbes et des surfaces » thèse de doctorat, laboratoire de CAO-Structures, Ecole Nationale Supérieure d'Arts et Metiers juillet 1990.
- [08]Belaidi I, Ishiomin G « Critères d'équivalences géométriques en relation avec des paramètres d'usinage », Actes *1^{er} colloque national sur la productique*, Université de Tizi-Ouzou (Algérie) et INI de Belfort (France), mai, 1998.
- [09]Belaidi I, Asma F, Ishiomin G, Coffignal G et Touratier M « Réduction de degré des courbes de Bézier. Applications potentielles en CFAO», *XV^{ème} Congrès Français de la Mécanique 2001*, Nancy, France, 3-7 septembre, 2001.
- [10]Belaidi I « Réduction de degré des courbes de Bézier. Qualité d'équivalence en relation avec des paramètres d'usinage » , Thèse de Doctorat, LMS-ENSAM, Paris, 1998.
- [11]Belaidi I et Asma F « Méthode de conversion par approximation des courbes de Bézier de degré élevé : Etude comparative » , Rapport interne N°1 du labo CFAO INGM Boumerdès ; 1993-94.
- [12]Bouazza M « La norme STEP » HERMES 1995.
- [13]Chambolle F : « Un modèle produit piloté par les processus d'élaboration : application au secteur automobile dans l'environnement STEP», Thèse de Doctorat de l'Ecole Centrale Paris, 1999.
- [14]Chen G D et Wang G J« Optimal multi-degree reduction of Bézier curves with constraints of endpoints» *continuity Computer Aided Geometric Design* 19 (2002)365-377.

- [15]Chouadria R et Omar T « Conversion par approximation des courbes de Bézier de degré élevé : Amélioration et mise en œuvre informatique de la méthode de HOSCHEK » PFE, Promoteur : Dr. Idir BELAIDI, Boumerdes 1995.
- [16]Demengel G et Pouget J P « Mathématiques des courbes et des surfaces, modèles de Bézier des B-SPLINES et des NURBS » ellipses 1998.
- [17]Eck M «Degree reduction of Bézier curves» ,Computer Aided Geometric Design, Vol.10, 1993, p.237-251.
- [18]Eck M « Least squares degree reduction of Bézier curves», Computer Aided Design, Vol.27, N°11, 1995, p.845-851.
- [19]El hadj mimoune M « Contribution à la modélisation explicite et à la représentation des données de composants industriels : application au modèle PLIB » thèse de doctorat, Facultés des sciences fondamentales et Appliquées, université de Poitiers Avril 2002.
- [20]Farin G « Curves and surfaces for CAGD », a practical guide, AcademicPress,Inc. 1991.
- [21]Farin.G « Courbes et surfaces pour la CGAO » MASSON 1992.
- [22]Faux I.D et Pratt M.J « Computational Geometry for Design and manufacture», Ellis Horwood, Chichester,1979,pp36,50,171,210,214,423,426,445,466.
- [23]Féru F et Viel C « Echanger avec le protocole d'application 203 de STEP, Echange et partage de données CAO et GDT», Association GOSET1998. ISBN 2-9513382-0-1.
- [24]Forrest A« Interactive interpolation and approximation by Bézier potynomials », Computer Aided Disign .J 1972 Vol 15, pp 71-79.
- [25]Forrest A « Interactive interpolation and approximation by Bézier polynomials » ,Computer Aided Design Vol 22, N°9 Nov 1990 pp 527-537.
- [26]Frederick M. P et Thomas R. K« A Feature-based Machining System using STEP», National Institute of Standards and Technology Building 220, Room B124 Gaithersburg, MD 20899-0001.
- [28]Foulard C « La modélisation en entreprise - CIMOSA et ingénierie simultanée», Ed.Hermès, Paris, 1994.
- [29]Gilberto F .M « Découpage transformationnel pour la conception des systèmes mixtes Logiciel/matériel » thèse de doctorat, Institut National Polytechnique de Grenoble Novembre 1998.
- [30]GRIVEL L « L'hypertexte comme mode d'exploitation des résultats d'outils et méthodes d'analyse de l'information scientifique et technique » thèse de doctorat, Facultés des sciences et technique de Saint Jérôme, Université de Droit et Economie, et des sciences d' Aix Marseille janvier 2000.
- [31]GZARA L « Les patterns pour l'ingénierie des systèmes d'information produit » thèse de doctorat, laboratoire GILCO, Institut National Polytechnique de Grenoble décembre 2000.

- [32]Hoschek J « Approximate conversion of spline curves », Computer Aided Geometric Design, Vol.4, 1987, p.59-66.
- [33]Hoschek J« Exact approximate conversion of spline curves and spline surfaces, Computation of Curves and Surfaces» , W.Dahmen et al. Editions, Kluwer. 1990, p.73-116.
- [34]Hoschek J « Intrinsic parametrisation for approximation »,Computer Aided Geometric Design N°5 1988 pp27-3.
- [35]Hoschek J et Wissel N « Optimal approximate conversion of spline curves and spline approximation of offset curves » , Computer Aided Design, Vol 20, N°8 Oct 1988 pp475-482.
- [36]Jean C.L « Modélisation et construction des surfaces pour la CFAO » HERMES 1991.
- [37]Kemmerer S « STEP: The Grand Experience», Editor of NIST Special Publication 939, National Institute of Standards and Technology, Gaithersburg, MD, 1999.
- [38]Kim H.O et Moon S.Y« Degree Reduction of Bézier curves by L^1 -Approximation with Endpoints Interpolation», Computers Math. Applic. Vol. 33, No.5, pp.67-77, 1997.
- [39]Lee B.G et Park Y et Yoo J « Application of Legendre-Bernstein basis transformations to degree elevation and degree reduction», Computer Aided Geometric Design 19 (2002)709-718.
- [40]Loffredo D « Fundamentals of STEP Implementation, STEP Tools», Inc., Rensselaer Technology Park, Troy, New York 12180.
- [41]Morris K.C et Mary J. M et Barnard A « Validating STEP Application Models at the National PDES Testbed», National Institute of Standards and Technology, Building 220, Room A127, Gaithersburg, MD, 20899.
- [42]Mortenson M.E « Geometric modeling», John Wiley,1985, pp17,411,423.
- [43]Nougal.F « Etude de paramètres de conversion par approximation des courbes de Bézier non rationnelles de degré élevé » PFE, Promoteur : Dr. Idir BELAIDI, Boumerdes2001.
- [44]Peterson J « Degree reduction of Bezier curves », Computer Aided Design, Vol.23,N°6, 1991, p.460-461.
- [45]Piegl Le et Tiller W. « Software engineering approach to degree elevation of B-spline curves»,Computer Aided Design, Vol.26, N°1, 1994, p.17-28.
- [46]Pierra G « Représentation et Echange de données techniques», Laboratoire d'Informatique Scientifique et Industrielle E.N.S.M.A 2000.
- [47]Rayane D et Noual R « Contribution à l'élaboration d'un système de CFAO de surfaces gauches de Bézier de degré quelconques. Implémentation d'un module de construction flexible de formes gauches», PFE. Labo CFAO université de Boumerdes 1999.

[48] Sunwoo H et Lee N « A unified matrix representation for degree reduction of Bézier curves », *Computer Aided Geometric Design* 21(2001) 151-164.

[49] Watkins M.A et Worsey A.J « Degree reduction of Bézier curves », *Computer Aided Design*, Vol.20, N°7, 1988, pp 398-40.

Annexe A

A.1 Détermination les fonctions $\lambda_{i,h}, \beta_{i,h}, b_h$

$$\begin{aligned}\gamma_{i,h} &= \sum_{j=1}^{np} B_{i,n}(t_j) \cdot B_{h,n-1}(t_j) \\ \beta_{i,h} &= \sum_{j=1}^{np} B_{i,n-1}(t_j) \cdot B_{h,n-1}(t_j) \\ b_h &= \sum_{i=0}^n W_i \cdot \gamma_{i,h} - \sum_{i=0}^{\alpha-1} Q_i \cdot \beta_{i,h} - \sum_{i=n-\alpha}^{n-1} Q_i \cdot \beta_{i,h}\end{aligned}$$

A.2 Continuité géométrique

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 $Q(t)$ [34].

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

$$k = 2 \rightarrow P'' = \lambda_1^2 \cdot Q'' + \lambda_2 \cdot Q'$$

$$k = 3 \rightarrow P''' = \lambda_1^3 \cdot Q''' + 3 \cdot \lambda_1 \cdot \lambda_2 \cdot Q'' + \lambda_3 \cdot Q'$$

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

Dans le cas des courbes de Bézier les équations précédentes deviennent :

$$k = 1 \rightarrow Q_0 = P_0$$

$$Q_1 = P_0 + \lambda_1 \cdot (P_1 - P_0)$$

$$Q_m = P_n$$

$$Q_{m-1} = P_n + \mu_1 \cdot (P_n - P_{n-1})$$

$$k = 2 \rightarrow Q_2 = P_1 + \lambda_1^2 \cdot \omega_1 \cdot (P_2 - P_1) + \lambda_2 \cdot (P_1 - P_0)$$

$$Q_{m-2} = P_{n-1} + \mu_1^2 \cdot \omega_1 \cdot (P_{n-2} - P_{n-1}) + \mu_2 \cdot (P_n - P_{n-1})$$

$$\text{avec } \omega_1 = \frac{m \cdot (n-1)}{n \cdot (m-1)}$$

A.3 Rappel sur les normes

La norme L^1 d'une courbe de Bézier $P(t)$ de degré n est donnée par la relation suivante :

$$\|P_n\|_1 = \int_0^1 |P_i \cdot B_{i,n}(t)| dt$$

La norme L^2 d'une courbe de Bézier $P(t)$ de degré n est donnée par la relation suivante :

$$\|P_n\|_2^2 = \int_0^1 \left| \sum_{i=0}^n P_i \cdot B_{i,n}(t) \right|^2 dt$$

La norme L_∞ d'une courbe de Bézier $P(t)$ de degré n est donnée par la forme suivante :

$$\|P_n\|_\infty = \max_{t \in [0,1]} \|P_i \cdot B_{i,n}(t)\|$$

Matrice d'élévation de degré T_n $(n+2) \times (n+1)$:

$$T_n = \frac{1}{n+1} \begin{pmatrix} n+1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & n & 0 & \dots & 0 & 0 & 0 \\ 0 & 2 & n-1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & n-1 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & n & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & n+1 \end{pmatrix}$$

A.4 Notice d'utilisation du programme élaboré

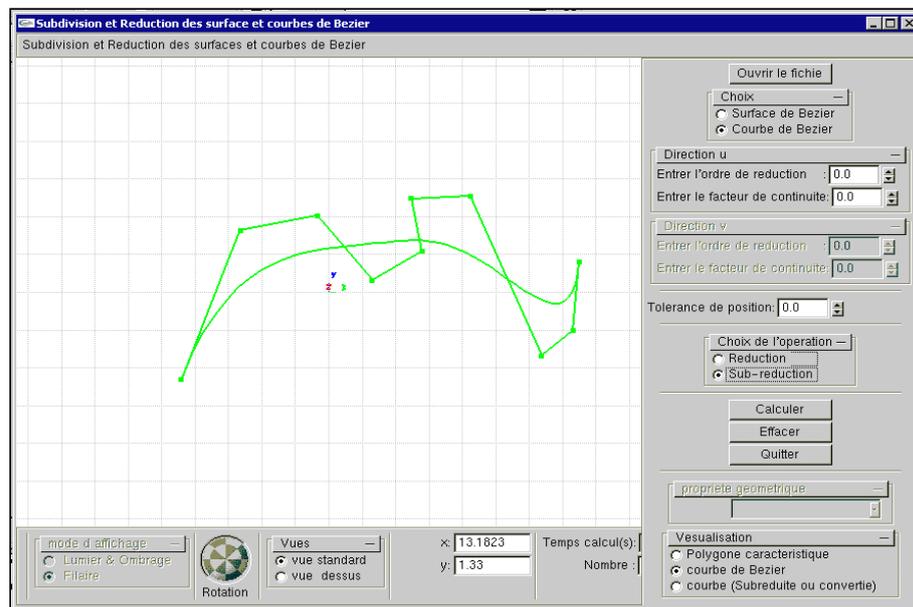


Figure A.1 Interface pour dessiner la courbe de Bézier.

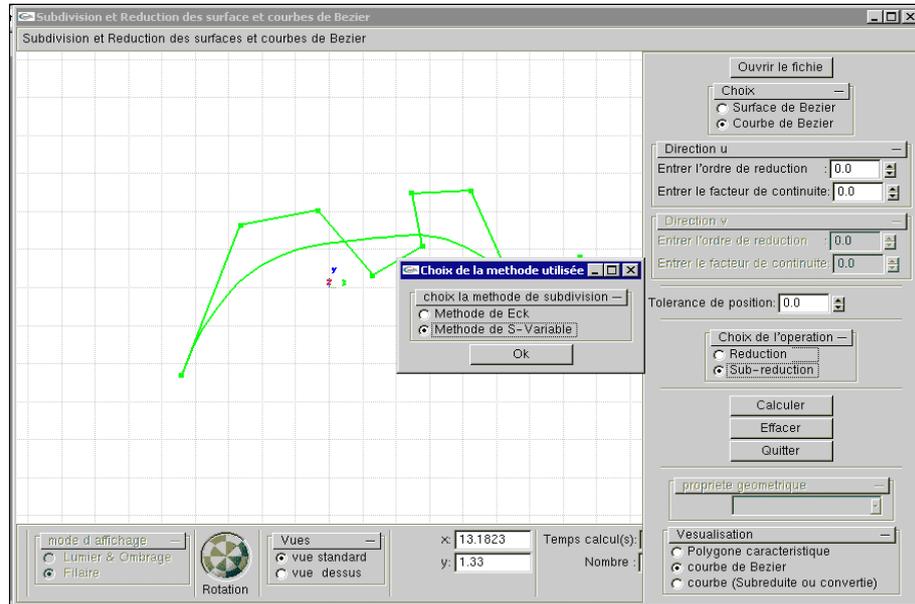


Figure A.2 Fenêtre de choix entre les deux méthodes.

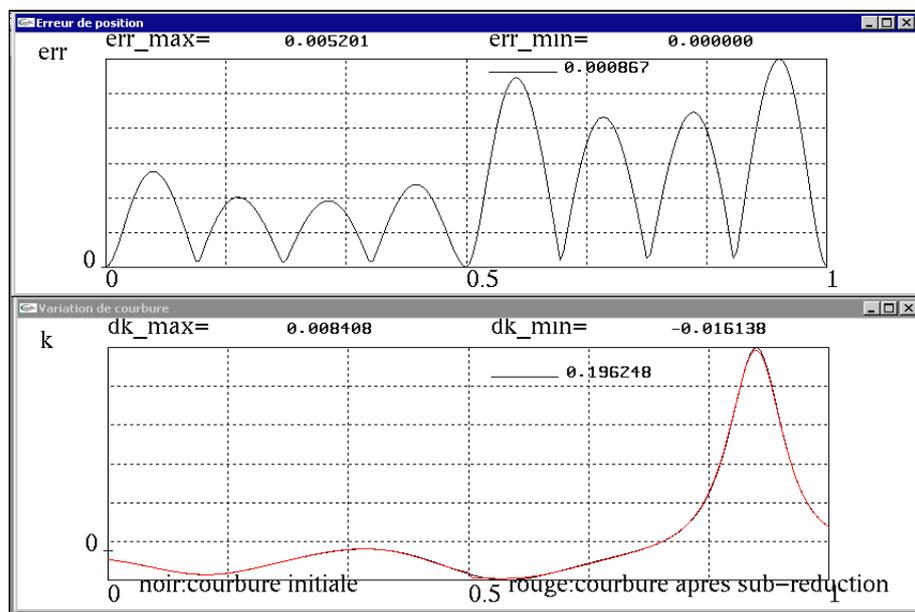


Figure A.3 Variation de l'erreur de position et la courbure de la courbe précédente.

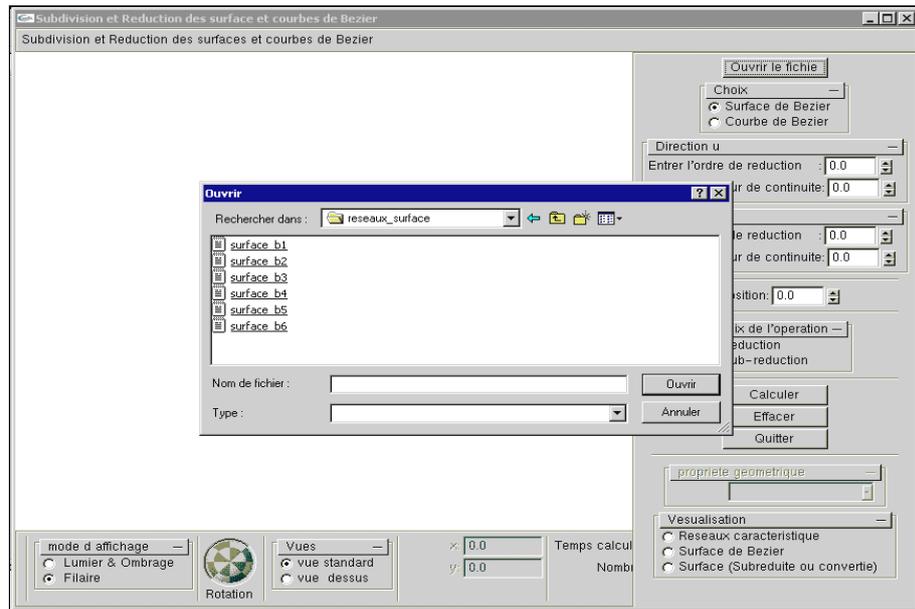


Figure A.4 Lecteur du réseau caractéristique à partir d'un fichier Txt.

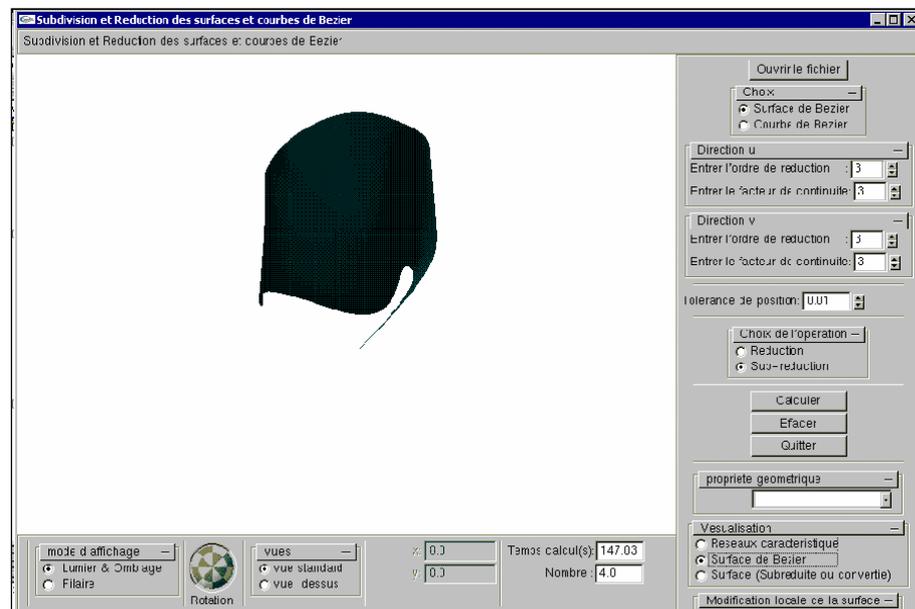


Figure A.5 Surface de Bézier de degré(14,14).

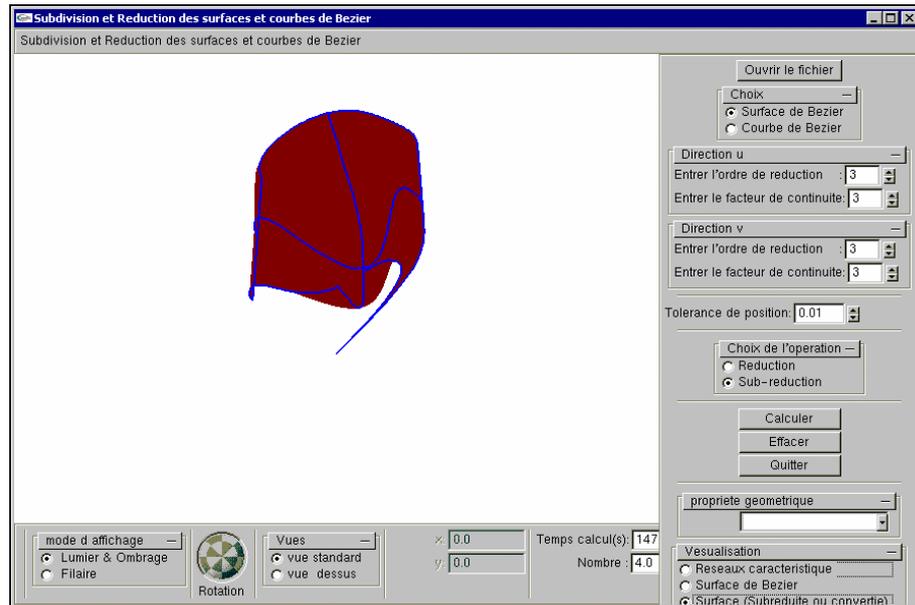


Figure A.6 Les surfaces de Bézier de degré (11,11) après la Sub-réduction.

Rayan_courbure_gaussienne - Bloc-notes			Rayan_courbure_moyenne - Bloc-notes		
Fichier	Edition	Format ?	Fichier	Edition	Format ?
0	0	0.0017423	0	0	0.00603892
0	0.0078125	0.00200137	0	0.0078125	0.00561475
0	0.015625	0.00221591	0	0.015625	0.00522381
0	0.0234375	0.00238235	0	0.0234375	0.00485951
0	0.03125	0.0024979	0	0.03125	0.00451609
0	0.0390625	0.00256069	0	0.0390625	0.00418883
0	0.046875	0.00256995	0	0.046875	0.00387409
0	0.0546875	0.00252616	0	0.0546875	0.00356939
0	0.0625	0.00243113	0	0.0625	0.00327346
0	0.0703125	0.00228799	0	0.0703125	0.00298623
0	0.078125	0.00210118	0	0.078125	0.00270877
0	0.0859375	0.00187629	0	0.0859375	0.00244325
0	0.09375	0.00161981	0	0.09375	0.00219258
0	0.101563	0.00133895	0	0.101563	0.00196034
0	0.109375	0.00104119	0	0.109375	0.00175047
0	0.117188	0.00073402	0	0.117188	0.00156686
0	0.125	0.00042461	0	0.125	0.00141329
0	0.132813	0.000119457	0	0.132813	0.00129291
0	0.140625	-0.000175764	0	0.140625	0.00120817
0	0.148438	-0.000456418	0	0.148438	0.00116061
0	0.15625	-0.000718955	0	0.15625	0.00115076
0	0.164063	-0.000960904	0	0.164063	0.00117811
0	0.171875	-0.00118029	0	0.171875	0.00114118

Figure A.7 Les deux courbures (Gaussienne et Moyenne) pour les 4 surfaces obtenues.

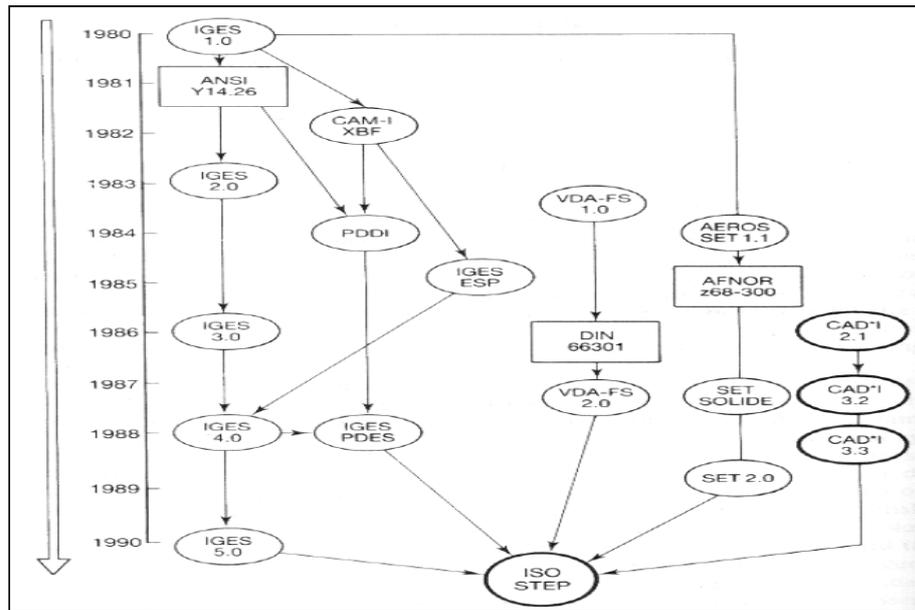


Figure A.8 Historique des normes d'échanges de données de produit[24].

Résumé

Ce présent travail s'inscrit dans le domaine d'échange de données entre systèmes de CFAO, et la modélisation géométrique des objets en utilisant des courbes/surfaces de Bézier de degré élevé, cette dernière présente plusieurs avantages tels que la souplesse de manipulation et la possibilité de modéliser des formes complexes. Mais elle engendre des comportements parasites et nécessitant un temps de calcul élevé. Pour résoudre ce problème ils y' a des méthodes de réduction de degré de courbes et surfaces de Bézier de degré élevé n en degré faible m , en respectant une continuité géométrique d'ordre $\alpha < (m-1)/2$ aux extrémités.

Notre travail consiste à réaliser une application qui permet la conversion par approximation des courbes/surfaces de Bézier non rationnelles de haut degré, et faire l'analyse des courbes et surfaces sub-réduites, pour les quelles deux (2) méthodes sont utilisées dans le cas des courbes :-1-Méthode de séparation des variables Asma [02].-2-Méthode de Eck. Et une seule méthode (séparation des variables) pour le cas des surfaces. Cette application nous permet de faire pour les deux méthodes citées précédemment la comparaison des paramètres tels que l'erreur de position, nombre des courbes obtenues après la sub-réduction et le temps de calcul.

Mots clés : Courbes de Bézier, approximation, réduction de degré, subdivision, conversion.

Abstract

This present work falls under the field of data exchange between systems of CAD-CAM, and the geometrical modelling of the objects by using curves/surfaces of Bézier of high degree, the latter has several advantages such as the flexibility of handling and the possibility of modelling complex forms. But it generates parasitic behaviours and requiring a high time calculating. To solve this problem they were methods of reduction of degree of curves and surfaces of Bézier of high degree n , in small degree m by respecting a geometrical continuity $\alpha < (m-1)/2$ at the endpoints.

Our work consists in carrying out an application which allows conversion by approximation of the curves and surfaces of Bézier non rational of high degree, and to make the analysis of the curves and surfaces sub-reduced, for which two (2) methods are used in the case of the curves:-1-Method of separation of the variables Asma [02].-2- Method of Eck. And only one method (separation of the variables) for the case of surfaces. This application enables us to make for the two methods quoted previously the comparison of the parameters such as the error of position, numbers curves obtained after the sub-reduction and the time calculating.

Keys words: curves and surfaces, approximation, degree reduction, data exchange, subdivision, conversion.

ملخص

إن هذا المشروع يندرج ضمن تبادل المعلومات بين انظمة و برنامج التصميم و الصناعة بمساعد الحاسوب لذلك قمنا بهذا العمل الذي يعالج مشكلة تخفيض درجة منحنيات و مسطحات بيبي من الدرجة العالية م إلى الدرجة المنخفضة ن باحترام الاستمرارية الهندسية التي تعتبر خاصية مهمة برتبة $(n-1/2)$ إن استعمال المنحنيات ذات الدرجة العالية يتميز بخاصية الليونة في التصميم وإنشاء الأشكال لكنها تستغرق وقتا كبير في الحساب لذلك قمنا بانجاز برنامج يخص تخفيض درجة منحنيات و هذا باستعمالنا لطريقتين الأولى طريقة الفصل بين المتغيرات والثانية طريقة أبك كما قمنا بالمقارنة بعد حساب الزمن عدد المنحنيات تغير في الوضعية تغير في المنحنى إما بنسبة للمسطحات فاستعملنا طريقة الفصل بين المتغيرات .

الكلمات المفتاحية: منحنيات بيبي; معطيات التبادل; تخفيض الدرجة.