

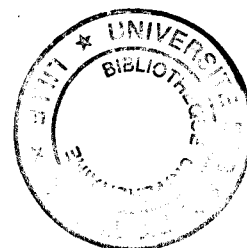
T

juin 2000 3365

50376
1996
240

NUMERO D'ORDRE :

ANNEE : 1996



THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le titre de

DOCTEUR en INFORMATIQUE

par

Maxime Froumentin

Modélisation à l'aide de surfaces quadriques dans le cadre de la synthèse d'images

Thèse soutenue le 27 Juin 1996, devant la commission d'examen :

- | | | |
|----------------------|---------------|-----------------------------------|
| Président : | V. Cordonnier | Université de Lille 1 |
| Directeur de Thèse : | C. Chaillou | Université de Lille 1 |
| Rapporteurs : | D. Bechmann | Université L. Pasteur, Strasbourg |
| | P. Chenin | Université J. Fourier, Grenoble |
| Examineurs : | M. Mériaux | Université de Poitiers |
| | R. Caubet | Université P. Sabatier, Toulouse |

UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE
 U.F.R. d'I.E.E.A. Bât M3. 59655 Villeneuve d'Ascq
 Tél. 20.43.47.24 Fax. 20.43.65.66

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M. H. LEFEBVRE, M. PARREAU

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT
ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PARREAU, J. LOMBARD, M. MIGEON, J. CORTOIS, A. DUBRULLE

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

M. P. LOUIS

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CHAMLEY Hervé	Géotechnique
M. CONSTANT Eugène	Electronique
M. ESCAIG Bertrand	Physique du solide
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. LABLACHE COMBIER Alain	Chimie
M. LOMBARD Jacques	Sociologie
M. MACKE Bruno	Physique moléculaire et rayonnements atmosphériques

M. MIGEON Michel
M. MONTREUIL Jean
M. PARREAU Michel
M. TRIDOT Gabriel

EUDIL
Biochimie
Analyse
Chimie appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre
M. BIAYS Pierre
M. BILLARD Jean
M. BOILLY Bénoni
M. BONNELLE Jean Pierre
M. BOSCOQ Denis
M. BOUGHON Pierre
M. BOURIQUET Robert
M. BRASSELET Jean Paul
M. BREZINSKI Claude
M. BRIDOUX Michel
M. BRUYELLE Pierre
M. CARREZ Christian
M. CELET Paul
M. COEURE Gérard
M. CORDONNIER Vincent
M. CROSNIER Yves
Mme DACHARRY Monique
M. DAUCHET Max
M. DEBOURSE Jean Pierre
M. DEBRABANT Pierre
M. DECLERCQ Roger
M. DEGAUQUE Pierre
M. DESCHEPPER Joseph
Mme DESSAUX Odile
M. DHAINAUT André
Mme DHAINAUT Nicole
M. DJAFARI Rouhani
M. DORMARD Serge
M. DOUKHAN Jean Claude
M. DUBRULLE Alain
M. DUPOUY Jean Paul
M. DYMENT Arthur
M. FOCT Jacques Jacques
M. FOUQUART Yves
M. FOURNET Bernard
M. FRONTIER Serge
M. GLORIEUX Pierre
M. GOSSELIN Gabriel
M. GOUDMAND Pierre
M. GRANELLE Jean Jacques
M. GRUSON Laurent
M. GUILBAULT Pierre
M. GUILLAUME Jean
M. HECTOR Joseph
M. HENRY Jean Pierre
M. HERMAN Maurice
M. LACOSTE Louis
M. LANGRAND Claude

Astronomie
Géographie
Physique du Solide
Biologie
Chimie-Physique
Probabilités
Algèbre
Biologie Végétale
Géométrie et topologie
Analyse numérique
Chimie Physique
Géographie
Informatique
Géologie générale
Analyse
Informatique
Electronique
Géographie
Informatique
Gestion des entreprises
Géologie appliquée
Sciences de gestion
Electronique
Sciences de gestion
Spectroscopie de la réactivité chimique
Biologie animale
Biologie animale
Physique
Sciences Economiques
Physique du solide
Spectroscopie hertzienne
Biologie
Mécanique
Métallurgie
Optique atmosphérique
Biochimie structurale
Ecologie numérique
Physique moléculaire et rayonnements atmosphériques
Sociologie
Chimie-Physique
Sciences Economiques
Algèbre
Physiologie animale
Microbiologie
Géométrie
Génie mécanique
Physique spatiale
Biologie Végétale
Probabilités et statistiques

M. LATTEUX Michel
M. LAVEINE Jean Pierre
Mme LECLERCQ Ginette
M. LEHMANN Daniel
Mme LENOBLE Jacqueline
M. LEROY Jean Marie
M. LHENAFF René
M. LHOMME Jean
M. LOUAGE François
M. LOUCHEUX Claude
M. LUCQUIN Michel
M. MAILLET Pierre
M. MAROUF Nadir
M. MICHEAU Pierre
M. PAQUET Jacques
M. PASZKOWSKI Stéfan
M. PETIT Francis
M. PORCHET Maurice
M. POUZET Pierre
M. POVY Lucien
M. PROUVOST Jean
M. RACZY Ladislas
M. RAMAN Jean Pierre
M. SALMER Georges
M. SCHAMPS Joël
Mme SCHWARZBACH Yvette
M. SEGUIER Guy
M. SIMON Michel
M. SLIWA Henri
M. SOMME Jean
Melle SPIK Geneviève
M. STANKIEWICZ François
M. THIEBAULT François
M. THOMAS Jean Claude
M. THUMERELLE Pierre
M. TILLIEU Jacques
M. TOULOTTE Jean Marc
M. TREANTON Jean René
M. TURRELL Georges
M. VANEECLOO Nicolas
M. VAST Pierre
M. VERBERT André
M. VERNET Philippe
M. VIDAL Pierre
M. WALLART François
M. WEINSTEIN Olivier
M. ZEYTOUNIAN Radyadour

Informatique
Paléontologie
Catalyse
Géométrie
Physique atomique et moléculaire
Spectrochimie
Géographie
Chimie organique biologique
Electronique
Chimie-Physique
Chimie physique
Sciences Economiques
Sociologie
Mécanique des fluides
Géologie générale
Mathématiques
Chimie organique
Biologie animale
Modélisation - calcul scientifique
Automatique
Minéralogie
Electronique
Sciences de gestion
Electronique
Spectroscopie moléculaire
Géométrie
Electrotechnique
Sociologie
Chimie organique
Géographie
Biochimie
Sciences Economiques
Sciences de la Terre
Géométrie - Topologie
Démographie - Géographie humaine
Physique théorique
Automatique
Sociologie du travail
Spectrochimie infrarouge et raman
Sciences Economiques
Chimie inorganique
Biochimie
Génétique
Automatique
Spectrochimie infrarouge et raman
Analyse économique de la recherche et développement
Mécanique

PROFESSEURS - 2ème CLASSE

M. ABRAHAM Francis	Composants électroniques
M. ALLAMANDO Etienne	Biologie des organismes
M. ANDRIES Jean Claude	Analyse
M. ANTOINE Philippe	Génétique
M. BALL Steven	Biologie animale
M. BART André	Génie des procédés et réactions chimiques
M. BASSERY Louis	Géographie
Mme BATTIAU Yvonne	Systèmes électroniques
M. BAUSIERE Robert	Mécanique
M. BEGUIN Paul	Physique atomique et moléculaire
M. BELLET Jean	Physique atomique, moléculaire et du rayonnement
M. BERNAGE Pascal	Sciences Economiques
M. BERTHOUD Arnaud	Sciences Economiques
M. BERTRAND Hugues	Analyse
M. BERZIN Robert	Physique de l'état condensé et cristallographie
M. BISKUPSKI Gérard	Algèbre
M. BKOUCHE Rudolphe	Biologie végétale
M. BODARD Marcel	Biochimie métabolique et cellulaire
M. BOHIN Jean Pierre	Mécanique
M. BOIS Pierre	Génie civil
M. BOISSIER Daniel	Spectrochimie
M. BOIVIN Jean Claude	Physique
M. BOUCHER Daniel	Biologie appliquée aux enzymes
M. BOUQUELET Stéphane	Gestion
M. BOUQUIN Henri	Chimie
M. BROCARD Jacques	Paléontologie
Mme BROUSMICHE Claudine	Mécanique
M. BUISINE Daniel	Biologie animale
M. CAPURON Alfred	Géographie humaine
M. CARRE François	Chimie organique
M. CATTEAU Jean Pierre	Sciences Economiques
M. CAYATTE Jean Louis	Electronique
M. CHAPOTON Alain	Biochimie structurale
M. CHARET Pierre	Composants électroniques optiques
M. CHIVE Maurice	Informatique théorique
M. COMYN Gérard	Composants électroniques et optiques
Mme CONSTANT Monique	Psychophysiologie
M. COQUERY Jean Marie	Sciences Economiques
M. CORIAT Benjamin	Paléontologie
Mme CORSIN Paule	Physique nucléaire et corpusculaire
M. CORTOIS Jean	Chimie organique
M. COUTURIER Daniel	Tectonique géodynamique
M. CRAMPON Norbert	Biologie
M. CURGY Jean Jacques	Physique théorique
M. DANGOISSE Didier	Analyse
M. DE PARIS Jean Claude	Composants électroniques et optiques
M. DECOSTER Didier	Electrochimie et Cinétique
M. DEJAEGER Roger	Informatique
M. DELAHA YE Jean Paul	Physiologie animale
M. DELORME Pierre	Sciences Economiques
M. DELORME Robert	Sociologie
M. DEMUNTER Paul	Physique atomique, moléculaire et du rayonnement
Mme DEMUYNCK Claire	Informatique
M. DENEL Jacques	Physique du solide - cristallographie
M. DEPREZ Gilbert	

M. DERIEUX Jean Claude
 M. DERYCKE Alain
 M. DESCAMPS Marc
 M. DEVRAINNE Pierre
 M. DEWAILLY Jean Michel
 M. DHAMELINCOURT Paul
 M. DI PERSIO Jean
 M. DUBAR Claude
 M. DUBOIS Henri
 M. DUBOIS Jean Jacques
 M. DUBUS Jean Paul
 M. DUPONT Christophe
 M. DUTHOIT Bruno
 Mme DUVAL Anne
 Mme EVRARD Micheline
 M. FAKIR Sabah
 M. FARVACQUE Jean Louis
 M. FAUQUEMBERGUE Renaud
 M. FELIX Yves
 M. FERRIERE Jacky
 M. FISCHER Jean Claude
 M. FONTAINE Hubert
 M. FORSE Michel
 M. GADREY Jean
 M. GAMBLIN André
 M. GOBLOT Rémi
 M. GOURIEROUX Christian
 M. GREGORY Pierre
 M. GREMY Jean Paul
 M. GREVET Patrice
 M. GRIMBLOT Jean
 M. GUELTON Michel
 M. GUICHAOUA André
 M. HAIMAN Georges
 M. HOUDART René
 M. HUEBSCHMANN Johannes
 M. HUTTNER Marc
 M. ISAERT Noël
 M. JACOB Gérard
 M. JACOB Pierre
 M. JEAN Raymond
 M. JOFFRE Patrick
 M. JOURNAL Gérard
 M. KOENIG Gérard
 M. KOSTRUBIEC Benjamin
 M. KREMBEL Jean
 Mme KRIFA Hadjila
 M. LANGEVIN Michel
 M. LASSALLE Bernard
 M. LE MEHAUTE Alain
 M. LEBFEVRE Yannic
 M. LECLERCQ Lucien
 M. LEFEBVRE Jacques
 M. LEFEBVRE Marc
 M. LEFEBVRE Christian
 Mlle LEGRAND Denise
 M. LEGRAND Michel
 M. LEGRAND Pierre
 Mme LEGRAND Solange
 Mme LEHMANN Josiane
 M. LEMAIRE Jean

Microbiologie
 Informatique
 Physique de l'état condensé et cristallographie
 Chimie minérale
 Géographie humaine
 Chimie physique
 Physique de l'état condensé et cristallographie
 Sociologie démographique
 Spectroscopie hertzienne
 Géographie
 Spectrométrie des solides
 Vie de la firme
 Génie civil
 Algèbre
 Génie des procédés et réactions chimiques
 Algèbre
 Physique de l'état condensé et cristallographie
 Composants électroniques
 Mathématiques
 Tectonique - Géodynamique
 Chimie organique, minérale et analytique
 Dynamique des cristaux
 Sociologie
 Sciences économiques
 Géographie urbaine, industrielle et démographie
 Algèbre
 Probabilités et statistiques
 I.A.E.
 Sociologie
 Sciences Economiques
 Chimie organique
 Chimie physique
 Sociologie
 Modélisation, calcul scientifique, statistiques
 Physique atomique
 Mathématiques
 Algèbre
 Physique de l'état condensé et cristallographie
 Informatique
 Probabilités et statistiques
 Biologie des populations végétales
 Vie de la firme
 Spectroscopie hertzienne
 Sciences de gestion
 Géographie
 Biochimie
 Sciences Economiques
 Algèbre
 Embryologie et biologie de la différenciation
 Modélisation, calcul scientifique, statistiques
 Physique atomique, moléculaire et du rayonnement
 Chimie physique
 Physique
 Composants électroniques et optiques
 Pétrologie
 Algèbre
 Astronomie - Météorologie
 Chimie
 Algèbre
 Analyse
 Spectroscopie hertzienne

M. LE MAROIS Henri	Vie de la firme
M. LEMOINE Yves	Biologie et physiologie végétales
M. LESCURE François	Algèbre
M. LESENNE Jacques	Systèmes électroniques
M. LOCQUENEUX Robert	Physique théorique
Mme LOPES Maria	Mathématiques
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU François	Sciences économiques
M. MAHIEU Jean Marie	Optique - Physique atomique
M. MAIZIERES Christian	Automatique
M. MANSY Jean Louis	Géologie
M. MAURISSON Patrick	Sciences Economiques
M. MERIAUX Michel	EUDIL
M. MERLIN Jean Claude	Chimie
M. MESMACQUE Gérard	Génie mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MOCHE Raymond	Modélisation,calcul scientifique,statistiques
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie organique
M. MORE Marcel	Physique de l'état condensé et cristallographie
M. MORTREUX André	Chimie organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
M. NIAY Pierre	Physique atomique,moléculaire et du rayonnement
M. NICOLE Jacques	Spectrochimie
M. NOTELET Francis	Systèmes électroniques
M. PALAVIT Gérard	Génie chimique
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. PERTUZON Emile	Physiologie animale
M. PETIT Daniel	Biologie des populations et écosystèmes
M. PLIHON Dominique	Sciences Economiques
M. PONSOLLE Louis	Chimie physique
M. POSTAIRE Jack	Informatique industrielle
M. RAMBOUR Serge	Biologie
M. RENARD Jean Pierre	Géographie humaine
M. RENARD Philippe	Sciences de gestion
M. RICHARD Alain	Biologie animale
M. RIETSCH François	Physique des polymères
M. ROBINET Jean Claude	EUDIL
M. ROGALSKI Marc	Analyse
M. ROLLAND Paul	Composants électroniques et optiques
M. ROLLET Philippe	Sciences Economiques
Mme ROUSSEL Isabelle	Géographie physique
M. ROUSSIGNOL Michel	Modélisation,calcul scientifique,statistiques
M. ROY Jean Claude	Psychophysiologie
M. SALERNO François	Sciences de gestion
M. SANCHOLLE Michel	Biologie et physiologie végétales
Mme SANDIG Anna Margarette	
M. SAWERYSYN Jean Pierre	Chimie physique
M. STAROSWIECKI Marcel	Informatique
M. STEEN Jean Pierre	Informatique
Mme STELLMACHER Irène	Astronomie - Météorologie
M. STERBOUL François	Informatique
M. TAILLIEZ Roger	Génie alimentaire
M. TANRE Daniel	Géométrie - Topologie
M. THERY Pierre	Systèmes électroniques
Mme TJOTTA Jacqueline	Mathématiques
M. TOURSEL Bernard	Informatique
M. TREANTON Jean René	Sociologie du travail

M. TUREL Georges
M. VANDIJK Hendrik
Mme VAN ISEGHEM Jeanine
M. VANDORPE Bernard
M. VASSEUR Christian
M. VASSEUR Jacques
Mme VIANO Marie Claude
M. WACRENIER Jean Marie
M. WARTEL Michel
M. WATERLOT Michel
M. WEICHERT Dieter
M. WERNER Georges
M. WIGNACOURT Jean Pierre
M. WOZNIAK Michel
Mme ZINN JUSTIN Nicole

Spectrochimie infrarouge et raman

Modélisation, calcul scientifique, statistiques
Chimie minérale
Automatique
Biologie

Electronique
Chimie inorganique
géologie générale
Génie mécanique
Informatique théorique

Spectrochimie
Algèbre

Modélisation à l'aide de surfaces quadriques dans le cadre de la synthèse d'images

Après une série de recherches au LIFL concernant l'affichage temps-réel des surfaces quadriques, nous considérons l'aspect "modélisation géométrique" du projet : la création de modèles composés de surfaces quadriques. Dans un premier temps, nous présentons l'aspect mathématique des quadriques : représentations classiques et nouvelles, ainsi que leurs propriétés pour la modélisation. Nous utilisons ensuite ces représentations de deux façons : dans un premier temps nous montrons les possibilités de création de surfaces complexes quadriques par morceaux par exemple par interpolation de données, ou par déformation à l'aide d'un polygone de contrôle. Nous introduisons également une nouvelle représentation, basée sur le formalisme B-spline, permettant un meilleur contrôle de la continuité des surfaces. Dans un second temps nous montrons, qu'à l'instar des facettes, les modèles complexes 3D composés de primitives de haut-niveau (Bézier, NURBS...) peuvent être approchés par des quadriques pour l'affichage. Nous présentons dans ce cadre un algorithme d'approximation quadrique de surfaces algébriques. Ces résultats nous permettent de conclure sur l'utilisation des quadriques comme primitives d'affichage en synthèse matérielle temps-réel.

Mots-clés : Synthèse d'Images, Modélisation géométrique, quadriques

Geometric modeling using quadric surfaces for image synthesis

Following the work at LIFL concerned with real-time display of quadric surfaces, we consider here the "geometric modeling" side of the project : the design of models made of quadric surfaces. We first describe the mathematical aspects of quadrics : well-known representations as well as more recent ones, and their properties for modeling. We use these representations in two ways : we first show the possibilities of complex piecewise quadric surface design using, for example point interpolation or free-form modeling using a control polygon. We also introduce a new representation, based on simplex B-splines, which gives a better control on surface continuity. Second, we show that complex 3D models made of high level primitives (Bézier, NURBS...) can be approximated by quadrics in order to be displayed. In this context, we introduce an algorithm for approximating algebraic surfaces by piecewise quadrics. These results help us to conclude about quadric surfaces as display primitives in hardware image synthesis.

Je tiens à remercier :

Dominique Bechmann, Carole Blanc, René Caubet, Christophe Chaillou, Patrick Chenin, Vincent Cordonnier, Samuel Degrande, Alain Fargue, Jean-Charles Fiorot, Laurent Fuchs, Henri Glanc, Insung Ihm, Sylvain Karpf, Hervé Laporte, Suresh Lodha, Pierre Macé, Jai Menon, Michel Mériaux, Charles Micchelli, Douglas Moore, Éric Nyiri, Alain Preux, Reinier Van Kleij, Martin Weiss, et tous les autres que j'oublie.

Table des matières

1 Synthèse d'images temps-réel	3
1.1 Modélisation géométrique tridimensionnelle – Primitives	4
1.1.1 Primitives volumiques	4
1.1.2 Primitives surfaciques	7
1.2 L'affichage temps-réel	9
1.2.1 Les accélérateurs graphiques	9
1.2.2 Facettisation	11
1.2.3 Problèmes	12
1.3 Alternatives aux facettes	12
1.3.1 Interpolation quadratique	12
1.3.2 Racine carrée	14
1.4 L'affichage matériel des quadriques	15
1.4.1 Limiter les quadriques	15
1.4.2 Surfaces/volumes quadriques	16
1.4.3 Opérations booléennes	16
1.5 Les problèmes logiciels	17
2 Courbes et surfaces de Bézier	19
2.1 Courbes de Bézier	20
2.1.1 Définition	20
2.1.2 Les polynômes de Bernstein sur $[0, 1]$	20
2.1.3 Propriétés des courbes de Bézier	22
2.1.4 L'algorithme de De Casteljaou	23
2.2 Surfaces de Bézier	24
2.2.1 Carreaux	25
2.2.2 Triangles	26
2.3 Courbes et surfaces de Bézier fonctionnelles	29
3 Les quadriques et leurs représentations	33
3.1 Représentation algébrique	34
3.1.1 Définition	34
3.1.2 Ecriture matricielle	34
3.1.3 Propriétés	35
3.1.4 Continuité	36
3.2 Représentation de Bernstein-Bézier	40
3.2.1 Représentation des coniques	40
3.2.2 Propriétés	43
3.2.3 Généralisation	43
3.2.4 Raccordement	45
3.3 Autres représentations	48
3.3.1 Le modèle géométrique	48
3.3.2 Représentation paramétrique	49

TABLE DES MATIÈRES

3.3.3	La représentation de Lodha-Warren	50
3.3.4	La représentation de Sederberg-Zundel	51
3.4	Bilan	53
4	Modélisation à l'aide de surfaces quadriques	55
4.1	Manipulation directe de la représentation	56
4.2	Modélisation traditionnelle	56
4.2.1	Quadriques naturelles	56
4.2.2	Quadriques spécifiques	57
4.2.3	Quadriques de révolution	57
4.3	Interpolation	60
4.3.1	Méthode de Pratt	61
4.3.2	Méthode de Bajaj et Ihm	63
4.3.3	Macropatchs quadriques	64
4.4	Nouvelle définition des macropatchs quadriques	67
4.4.1	Objectifs	67
4.4.2	Bilan sur les macropatchs	75
4.5	Surfaces quadriques définies par simplex B-splines	76
4.5.1	Volumes B-splines par produit tensoriel	77
4.5.2	Les simplex B-splines	78
4.5.3	Application aux quadriques	81
4.6	Bilan	85
5	Approximation	87
5.1	Problématique	88
5.1.1	État de l'art : la facettisation	89
5.2	Quadrisation par interpolation	90
5.2.1	Échantillonnage de surfaces	90
5.3	Subdivision spatiale et réduction de degré	92
5.3.1	Les algorithmes de réduction de degré pour les simplex de Bézier	92
5.3.2	Réduction de degré appliquée aux surfaces algébriques	95
5.3.3	Réduction par morceaux	97
5.3.4	Application aux quadriques	98
6	Bilan et Propositions	105
6.1	Synthèse	106
6.2	Perspectives	108

Introduction

Actuellement, le seul moyen de calculer une image de synthèse tridimensionnelle avec une vitesse suffisante pour une application interactive est l'utilisation d'accélérateurs graphiques matériels. Ces accélérateurs utilisent des circuits spécifiques pour effectuer les calculs de rendu d'objets ou de scènes composés de facettes planes.

Jusqu'à présent, la recherche en rendu temps-réel a progressé principalement au niveau de la rapidité d'affichage, du placage de textures sur les objets, et de la complexité des primitives affichables. Ce troisième aspect est à la base d'un des thèmes de recherche au LIFL : la conception d'un accélérateur pour l'affichage de surfaces quadriques. Ces surfaces, beaucoup plus variées au niveau de la forme que les facettes, incluent des surfaces bien connues comme les sphères ou les cylindres, et permettent ainsi de mieux représenter les objets courbes.

L'aspect matériel de ce projet a été principalement étudié : la conception de circuits, l'architecture de l'accélérateur, etc. À l'opposé, nous tentons d'explorer ici le point de vue logiciel, dans le but d'apporter une réponse à une question fondamentale du projet : les surfaces quadriques représentent-elles un progrès véritable par rapport aux facettes en tant que primitives d'affichage temps-réel ?

Nous évaluons donc ici les possibilités géométriques offertes par les surfaces quadriques et les méthodes de création d'objets à l'aide de celles-ci, dans le but de prouver leur utilité pour la synthèse d'images. Ce travail nous mène loin du domaine de l'architecture des ordinateurs, mais nous tentons de démontrer qu'il est nécessaire pour prouver la validité de l'idée de départ.

- Le premier chapitre présente en détail le contexte du document : les primitives utilisées en modélisation géométrique, la différence entre primitives de modélisation et primitives de visualisation, l'affichage matériel de facettes, et l'évolution dans ce domaine jusqu'à l'affichage de quadriques.
- Les chapitres 2 et 3 ont pour but d'exposer certains outils mathématiques de la théorie des surfaces, respectivement les surfaces de Bézier et les surfaces algébriques, afin, d'une part, de poser le formalisme utilisé par la suite et, d'autre part, d'établir certains résultats fondamentaux concernant la création de surfaces complexes à l'aide de quadriques.

Les principaux apports de cette thèse sont développés dans les chapitres 4 et 5, où la création de surfaces quadriques complexes est présentée sous deux aspects : la création directe ou l'approximation :

- Le premier aspect est traité au chapitre 4, que nous commençons en énumérant les méthodes classiques de création directe de surfaces quadriques par morceaux. Nous introduisons également une définition de "macropatches" quadriques, déformables à l'aide de polygones de contrôle. Après avoir établi que ces modèles sont tous fortement limités dès que l'on cherche

à créer des surfaces avec continuité de la tangente, nous cherchons à définir une représentation plus souple des surfaces par morceaux. À partir de la définition des simplex B-splines de Dahmen-Micchelli-Seidel, nous introduisons une nouvelle définition des surfaces quadriques par morceaux qui évite les limitations précédentes.

- La seconde perspective est étudiée au chapitre 5. Celle-ci consiste en la conversion de surfaces complexes en quadriques, pour l'affichage. Ce processus s'apparente à la facettisation où les surfaces complexes sont approximées par des facettes planes. Bien qu'à notre connaissance aucun algorithme de ce genre n'existe nous pouvons supposer qu'à l'instar des algorithmes de facettisation, les algorithmes d'approximation quadrique sont fondamentalement différents selon le type de surface à approximer. Nous tentons d'évaluer les méthodes possibles pour les définitions de surfaces les plus courantes, et nous proposons en particulier un algorithme de "réduction de degré" qui permet de calculer une approximation quadrique d'une surface algébrique quelconque.

Le chapitre 6 synthétise les résultats obtenus : nous récapitulons les différentes méthodes introduites et nous évaluons les possibilités d'amélioration de celles-ci. Finalement, nous abordons des perspectives plus éloignées, en particulier l'utilisation de surfaces de degré supérieur à deux comme primitives de visualisation.

Chapitre 1

Synthèse d'images temps-réel

La réalisation d'une image de synthèse s'obtient à partir de trois processus successifs : la *modélisation géométrique*, la *modélisation photométrique* et le *rendu*. La modélisation géométrique consiste à représenter la géométrie d'un objet ou d'une scène en mémoire. Cette représentation est un ensemble de surfaces ou volumes mathématiques, qui décrivent une approximation de l'objet réel ou qui répondent à des contraintes géométriques, physiques ou esthétiques. D'autres informations, susceptibles d'intervenir dans la réalisation de l'image, peuvent également être stockées, principalement d'ordre topologique. La modélisation photométrique a pour but de définir les propriétés visuelles du modèle géométrique, par exemple les couleurs, les matières ou les textures des objets. Le rendu est le calcul de l'image proprement dit, à partir du modèle géométrique et photométrique. Ce calcul consiste à simuler les effets lumineux (diffusion, ombres, réflexions, etc.) au voisinage de l'objet, en fonction du point de vue.

Bien entendu, le réalisme d'une image de synthèse est variable, en fonction des méthodes utilisées à la modélisation et au rendu. Une image très réaliste sera réalisée en créant un modèle géométrique et photométrique très proche de l'objet réel original, et en utilisant des algorithmes de rendu complexes, reproduisant de la façon la plus fidèle possible les modèles physiques de la lumière. Le réalisme et la qualité d'image s'obtiennent bien entendu au détriment du temps de calcul, et la création d'une seule image peut, encore actuellement, demander plusieurs heures, même sur des machines puissantes.

Toutefois, il existe de nombreuses applications pour lesquelles, il faut sacrifier le réalisme au profit de la rapidité du calcul. En particulier, dans le cas d'applications temps-réel, comme la visualisation scientifique (par exemple, pour l'étude d'un système de particules : feu, écoulement des fluides..) ou les simulateurs (véhicules, médicaux...), il est nécessaire de générer les images avec une fréquence élevée, typiquement 25 par seconde. Les problèmes de la *synthèse d'images temps-réel* diffèrent beaucoup de ceux de la synthèse réaliste, aussi bien au niveau de la modélisation qu'au niveau du rendu.

Ce chapitre a pour but de décrire certains de ces problèmes, en introduisant le contexte particulier de cette thèse. Nous décrivons d'abord les notions de primitives géométriques. En présentant l'affichage matériel temps-réel, nous décrivons la différence entre primitives de modélisation et primitives de visualisation et les problèmes qu'entraîne cette différence. Dans le but de résoudre partiellement celle-ci, nous proposons ensuite une alternative aux systèmes actuels, développée au LIFL : l'utilisation des surfaces quadriques. Les conséquences de ce choix sont ensuite évaluées, du point de vue de la modélisation.

1.1 Modélisation géométrique tridimensionnelle – Primitives

La première étape de la création d'une image tridimensionnelle (3D) consiste à décrire la scène ou l'objet à représenter, de façon analytique, à l'aide de surfaces ou de volumes élémentaires, appelés *primitives*. Ces primitives sont stockées en mémoire, ainsi que d'autres informations géométriques : voisinage, inclusion, topologie... Les primitives géométriques sont en général définies dans le cadre de la théorie mathématique des courbes et surfaces [Farin92, Bohm et al.84, Fiorot et al.92]. C'est ainsi que l'on peut les classer en familles selon leur type (surface, volume) et leur définition mathématique (paramétrique, implicite). Cette classification n'est évidemment pas exhaustive. La théorie des courbes et surfaces a connu avec l'informatique d'importants développements et énormément de possibilités ont été étudiées. Nous nous contentons ici d'en énoncer les grandes lignes.

1.1.1 Primitives volumiques

Un objet courant peut toujours être considéré comme un volume : à un niveau de description macroscopique, tout objet est un amas de matière et peut donc être représenté comme un sous-ensemble *volumique* de points de l'espace \mathbb{R}^3 . Représenter un objet sous forme volumique, c'est donc se donner un moyen de différencier les points de l'espace appartenant ou n'appartenant pas à l'objet. Un volume complexe est en général représenté par un ensemble de volumes plus simples : les primitives volumiques.

L'assemblage des primitives volumiques se fait de façon booléenne : l'*union* consiste simplement à assembler les primitives. Cet opérateur est à la base des géométries *voxel* où un objet est l'union d'un ensemble de cubes élémentaires alignés (figure 1.1).

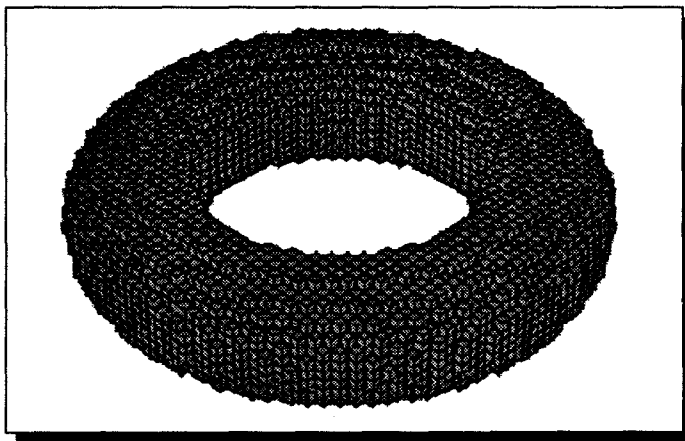


FIG. 1.1 - Tore représenté par primitives voxel

Les opérateurs d'*intersection* ou de *différence*, utilisés de manière hiérarchique, sont à la base de la *géométrie solide constructive* (CSG). La figure 1.2 montre un volume simple défini à l'aide de CSG.

On peut classer les primitives utilisées par ces schémas selon trois familles :

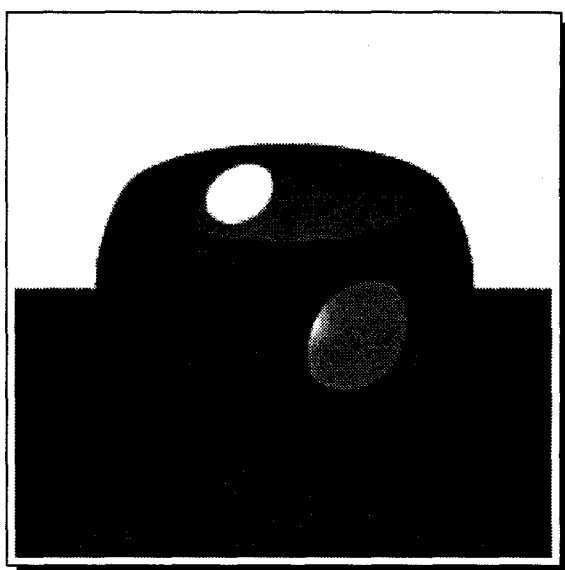


FIG. 1.2 - Objet construit à l'aide d'une structure CSG utilisant l'opérateur de différence

Instanciation

Le type de primitive volumique le plus simple s'obtient en *instanciant* certaines primitives volumiques élémentaires, par exemple des cubes pour les voxels, ou des cylindres, boules, etc. pour les schémas plus complexes. L'instanciation se fait en choisissant une primitive et en lui donnant certains attributs simples comme la taille (ou les dimensions suivant les trois axes), la position et l'orientation (figure 1.3). Dans le cas simple des voxels, chaque cube est instancié uniquement en donnant sa position dans l'espace. La plupart des modèles CSG, bien que non limitatifs au niveau de la complexité des primitives, utilisent des primitives instanciées parce que la limitation du type de primitives rend d'autant plus simples certains algorithmes ou traitements comme l'affichage, le calcul de valeurs physiques (masse, moment d'inertie) ou la fabrication (i.e. trous, encoches, chanfreins avec une machine à commande numérique, cf. Gardan [Gardan91], page 245).

Primitives volumiques paramétriques

Avec la multiplication des applications en modélisation géométrique, apparaît la nécessité de créer des objets plus complexes, de forme variable (répondant à des contraintes précises ou à des demandes plus esthétiques) qui sortent du cadre limité des modèles instanciables. Par exemple, les formes complexes que peuvent prendre certains objets tels que les carrosseries de voiture, ont une diversité telle qu'un modèle instanciable ne suffit pas. Il est par conséquent nécessaire de faire intervenir des formalismes géométriques plus complexes.

Un volume peut être défini de manière paramétrique par une équation du type :

$$\begin{aligned}
 f: \quad \mathbb{R}^3 &\rightarrow \mathbb{R}^3 \\
 (s, t, u) &\mapsto (x(s, t, u), y(s, t, u), z(s, t, u))
 \end{aligned}
 \tag{1.1}$$

Pour représenter des volumes finis, on limite l'ensemble de départ à un sous ensemble de \mathbb{R}^3 , par exemple le cube unité $[0, 1]^3$. Le volume défini par l'équation 1.1 est donc la déformation

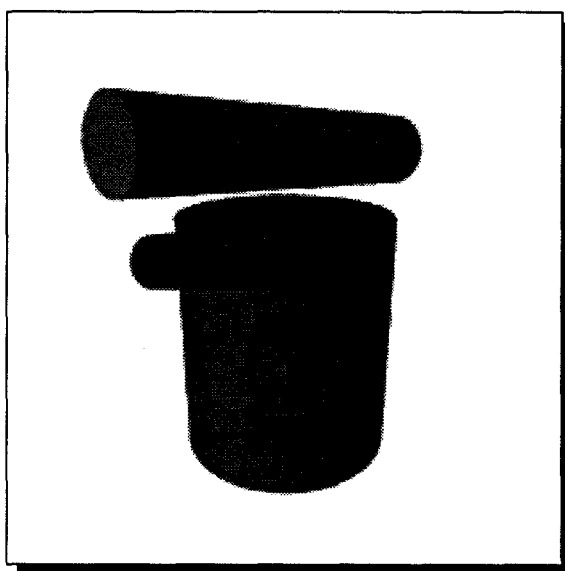


FIG. 1.3 - Trois instanciations de cylindres

de ce cube par les fonctions $x(s, t, u)$, $y(s, t, u)$ et $z(s, t, u)$. La définition paramétrique permet de contrôler la forme de la surface par déformation continue, à l'aide de points de contrôle, en utilisant par exemples les formalismes de Bézier [Lasser85] ou B-spline (issus de la théorie des surfaces, cf. chapitre 2).

Ce type de définition de volume, bien que parfaitement valable du point de vue mathématique, est toutefois très rarement rencontré en pratique, à cause de sa difficulté d'utilisation pour l'affichage.

Primitives volumiques implicites

Définir un volume de façon implicite, c'est donner un critère direct d'appartenance au volume pour tout point de l'espace. Ce critère peut s'exprimer simplement sous la forme d'une inéquation :

$$f(x, y, z) < 0 \quad (1.2)$$

Les points vérifiant cette inéquation sont considérés comme faisant partie du volume ainsi défini. La fonction f définit donc complètement celui-ci. Par exemple, la fonction $f(x, y, z) = x^2 + y^2 + z^2 - 1$ définit la boule de rayon 1 centrée sur l'origine. C'est-à-dire que cette boule est l'ensemble des points :

$$P(x, y, z) \mid x^2 + y^2 + z^2 - 1 < 0 \quad (1.3)$$

Une fonction f non dégénérée (c'est-à-dire qui définit effectivement un volume, par exemple $x^2 + y^2 + z^2 + 42 < 0$ est dégénérée puisqu'aucun point de \mathbb{R}^3 ne vérifie l'inéquation) définit deux demi-espaces : $P_1(x, y, z) \mid x^2 + y^2 + z^2 - 1 < 0$ et $P_2(x, y, z) \mid x^2 + y^2 + z^2 - 1 \geq 0$. On a ainsi directement une distinction intérieur/extérieur.

1.1.2 Primitives surfaciques

Plutôt que de décrire l'ensemble des points faisant partie de l'objet, il est souvent plus simple de donner simplement la frontière séparant l'intérieur de l'extérieur, sous la forme d'une surface. De la même façon que pour les volumes, une surface complexe est en général décrite par un assemblage de primitives surfaciques.

Une condition importante sur la surface est que celle-ci sépare effectivement l'intérieur de l'extérieur de l'objet, autrement dit, celle-ci doit rester fermée. Même si pour la synthèse d'images cette condition est souvent ignorée, puisque l'important est la partie de la scène que l'on voit, elle reste néanmoins essentielle en CAO. Ce test de fermeture des surfaces entraîne un certain nombre de vérifications d'ordre topologique lors de modifications éventuelles de l'objet.

Comme pour les catégories de primitives volumiques, on retrouve les types instanciation, paramétrique et implicite :

Primitives surfaciques instanciées

Ici encore, un certain nombre de surfaces de type fixé (triangle, sphère, etc.) est instanciable en précisant taille, position et orientation. L'exemple le plus répandu est sans doute le modèle polyédrique à facettes triangulaires (figure 1.4).

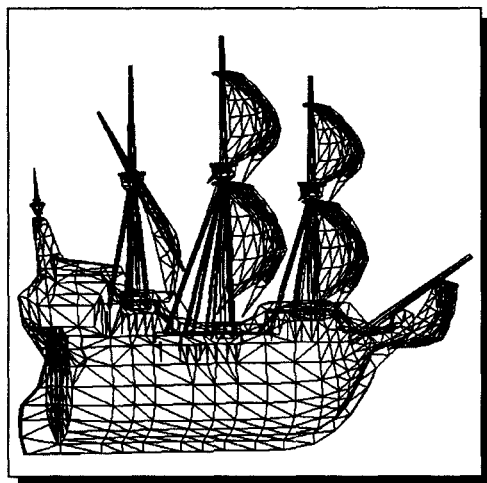


FIG. 1.4 - Polyèdre 3D

Définition paramétrique

Les surfaces paramétriques sont beaucoup plus utilisées que leur équivalent volumique. Une surface est définie par une application de \mathbb{R}^2 (l'espace des paramètres) dans \mathbb{R}^3 (l'espace de modélisation) :

$$\begin{aligned} f : \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (s, t) &\mapsto (x(s, t), y(s, t), z(s, t)) \end{aligned} \quad (1.4)$$

On limite l'espace des paramètres à un domaine de \mathbb{R}^2 , souvent $[0, 1] \times [0, 1]$ (figure 1.5). Cela permet de contrôler (partiellement) l'extension de la surface dans l'espace.

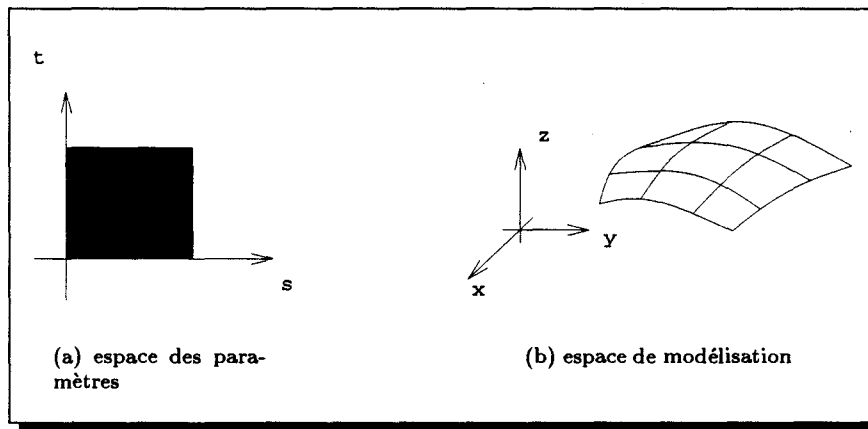


FIG. 1.5 - Surface paramétrique

Les fonctions $x(s, t)$, $y(s, t)$ et $z(s, t)$ sont, comme pour les volumes, des fonctions quelconques de \mathbb{R}^2 dans \mathbb{R} . Ces fonctions sont en général polynomiales, parfois trigonométriques. Les fonctions trigonométriques permettent de définir des surfaces bien connues comme les sphères :

$$\begin{cases} x = \cos \theta \sin \phi & 0 \leq \theta \leq 2\pi \\ y = \sin \theta \sin \phi & 0 \leq \phi \leq \pi \\ z = \cos \theta & \end{cases} \quad (1.5)$$

Les fonctions polynomiales sont à la base de la théorie des surfaces dites à pôles, c'est-à-dire dont la forme est fixée par des points de contrôle de l'espace : surfaces de Bézier, surfaces B-spline, etc. Nous y revenons plus en détail dans le chapitre 2.

Définition implicite

Une surface implicite est définie comme l'ensemble des points dont les coordonnées vérifient l'équation :

$$f(x, y, z) = 0 \quad (1.6)$$

À l'instar des volumes implicites, les surfaces implicites sont donc également définies par la fonction f . En fait, la surface définie par la fonction f est la frontière des volumes également définis par cette fonction ($f(x, y, z) > 0$ et $f(x, y, z) < 0$, figure 1.6).

Classiquement, les fonctions sont polynomiales. Les surfaces sont alors appelées *algébriques*. D'autres définitions implicites existent dans lesquelles la fonction f est modifiable par des points ou un polygone de contrôle, qui permettent une modification intuitive de la forme [Blinn82, Bloomenthal et al.91]. Dans ce cas la fonction f est parfois exponentielle. Certaines surfaces implicites ayant souvent une extension infinie (en particulier les surfaces algébriques), on les limite par un volume frontière, souvent parallélépipédique.

Notons que ces différentes définitions des surfaces ne sont absolument pas exclusives, et que certaines surfaces peuvent avoir une définition implicite et paramétrique. Par exemple, la sphère

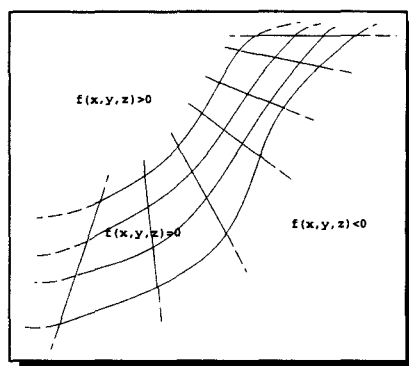


FIG. 1.6 - Équation implicite définissant 2 demi-espaces et une surface

définie par l'équation 1.5 est également définie de façon implicite par :

$$f(x, y, z) = x^2 + y^2 + z^2 - 1 \quad (1.7)$$

1.2 L'affichage temps-réel

Quand un objet ou une scène ont été approchés et représentés par un ensemble de primitives, les algorithmes de rendu exploitent cette représentation pour créer une image synthétique de l'objet. Comme nous l'avons dit précédemment, les performances des algorithmes de rendu dépendent de la complexité des primitives utilisées : actuellement, si des primitives complexes telles les surfaces de Bézier sont utilisées, il n'est pas question d'obtenir une vitesse de rendu direct de ces surfaces suffisante pour des applications interactives (nous voyons pourquoi plus loin).

Les seules primitives pouvant être affichées de façon suffisamment rapide sont les facettes planes, à l'aide d'*accélérateurs graphiques*.

1.2.1 Les accélérateurs graphiques

Un accélérateur graphique est un périphérique ou une carte d'extension qui effectue les calculs d'affichage graphique 3D de façon rapide, grâce à des circuits spécialisés. Actuellement, les accélérateurs sont capables de calculer le rendu par la méthode d'éclairage de Gouraud [Foley et al.90] d'une scène composée de triangles 3D. Les performances atteignent des cadences d'affichage de plusieurs millions de triangles par seconde. Les accélérateurs actuels utilisent la méthode du *rendu projectif* (figure 1.7) : à chaque triangle est appliquée la transformation perspective (en général calculée en amont de l'accélérateur), puis celui-ci est "converti en pixels". Pour cela la couleur de chaque pixel où le triangle est présent est d'abord calculée (indépendamment des autres triangles). Finalement, la couleur finale de chaque pixel est obtenue par l'algorithme d'élimination des parties cachées appelé Z-buffer.

Avec un accélérateur graphique, l'affichage d'un triangle passe donc par le calcul de plusieurs expressions linéaires, lors de la conversion :

1. La détermination de la présence du triangle à l'intérieur du pixel (x, y) peut se faire de deux façons : la première consiste à tester de quel côté est le pixel par rapport aux projections des trois arêtes (figure 1.8 a). La seconde méthode calcule directement les pixels à l'intérieur du

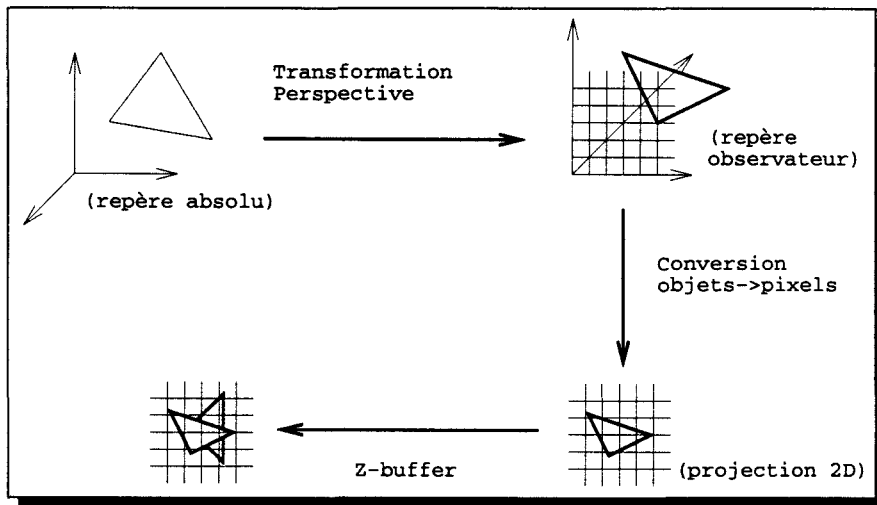


FIG. 1.7 - Rendu projectif

triangle de façon incrémentale. Dans les deux cas, trois calculs d'expressions linéaires sont nécessaires :

$$y = ax + by + c \quad (1.8)$$

- Si, lors de la première étape, on a déterminé que le pixel appartient au triangle, il faut alors déterminer la profondeur du triangle en ce pixel. Celle-ci se calcule à l'aide de l'équation du plan qui supporte la facette, qui est encore une fois une expression linéaire. Ce calcul se fait également de façon incrémentale, à l'aide d'interpolateurs linéaires (figure 1.8 b) ¹.

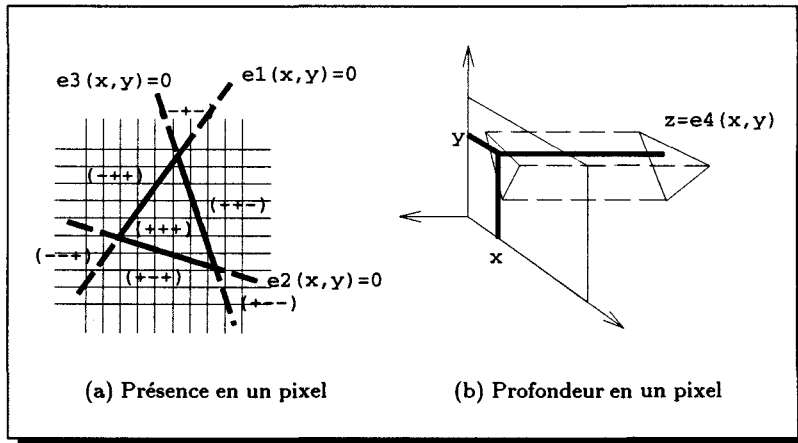


FIG. 1.8 - Expressions linéaires pour l'affichage de facettes

L'algorithme de Gouraud consiste à donner un aspect arrondi à un modèle polyédrique lors de l'affichage. Seules les couleurs aux sommets des triangles sont calculées par la méthode d'éclair-

1. La transformation perspective est supposée avoir déjà été appliquée sur la primitive. C'est pour cette raison que le calcul de la profondeur se fait parallèlement à l'axe z. Pour plus de détails, on se référera à des descriptions plus spécifiques [Karpf93, Nyiri94].

ment. Les couleurs des autres pixels de la facette sont obtenues par interpolation des couleurs aux sommets. Cette interpolation est réalisée lors de la conversion et utilise également des interpolateurs linéaires. Le polyèdre apparaît alors "lissé" (figure 1.9).

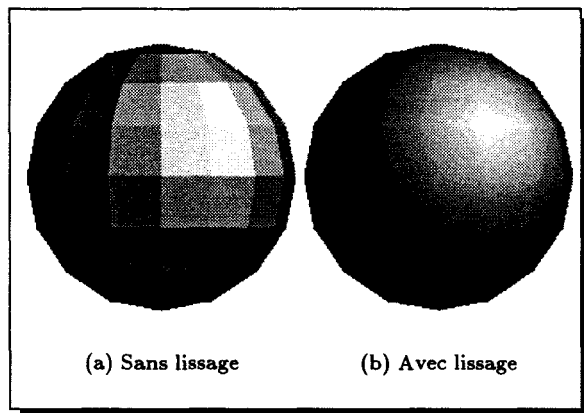


FIG. 1.9 - Méthode de lissage de Gouraud

Cette interpolation améliore donc beaucoup l'affichage de modèles polyédriques qui sont des approximations de surfaces courbes. C'est pour cette raison qu'on la retrouve sur tous les accélérateurs actuels.

1.2.2 Facettisation

Si les facettes sont les seules primitives pouvant être affichées rapidement, tout objet à afficher doit donc être approximé par des facettes. En fait l'objet est d'abord approximé une première fois par des surfaces ou des volumes de complexité élevée (surface de Bézier, cylindres, etc.), puis ces surfaces sont elles mêmes approximées par un polyèdre. Cette seconde approximation (linéaire) est appelée *polygonalisation* ou *facettisation* de la surface. La facettisation des surfaces représente une part importante dans la recherche en infographie. Beaucoup de méthodes existent en fonction du type de primitive (nous y revenons au chapitre 5).

De façon générale, cette double approximation donne donc naissance à deux types de primitives: les *primitives de modélisation* et les *primitives de visualisation*. On appelle primitive de modélisation toute surface ou volume qui représente la première approximation de l'objet, au stade de la modélisation géométrique. Si cette primitive ne dispose pas d'algorithme permettant un rendu rapide, l'objet est approximé par un second type de primitives, plus simples, les primitives de visualisation, affichables directement.

Si l'on s'autorise un rendu très réaliste sans contrainte de temps, et si l'on dispose d'un algorithme de rendu adéquat, la double approximation n'est plus nécessaire puisqu'il suffit de faire l'affichage direct des primitives de modélisation. La méthode de rendu par lancer de rayons permet un rendu direct sur un grand nombre de surfaces complexes [Glassner89]: il suffit de savoir calculer l'intersection de la surface avec une droite. À l'opposé, à partir du moment où le rendu doit être rapide, la complexité des primitives affichable est réduite et la différence modélisation/visualisation apparaît. La facettisation représente le passage des primitives de modélisation vers les primitives de visualisation, dans le cas du rendu temps-réel de scènes facettisées.

1.2.3 Problèmes

Cette séparation des primitives, bien que nécessaire dans la synthèse rapide, n'est pas sans poser de problèmes :

- La simplicité des primitives de visualisation fait qu'il est, dans la plupart des cas, nécessaire d'en utiliser un très grand nombre pour atteindre une approximation satisfaisante. Par exemple, sur la figure 1.9, 128 facettes sont nécessaires pour approcher la sphère. L'objet de la figure 1.4 comprend environ 4700 triangles. La représentation d'une scène complexe nécessite donc des quantités de mémoire parfois très importantes et des vitesses de calcul suffisantes.
- L'aspect visuel de l'objet reste de mauvaise qualité, du moins dans la plupart des cas d'utilisation du lissage de Gouraud : le contour des objets reste polygonal (figure 1.9), les arêtes des facettes restent visibles (phénomène de bandes de Mach), et des effets de distortion perspective apparaissent : l'interpolation des couleurs est calculée dans l'espace image (c'est-à-dire au niveau des pixels), par conséquent les effets de profondeur de la perspective ne sont plus pris en compte.
- Lors d'une modification du modèle (déformation, étirement, etc.), il est nécessaire de facetter au fur et à mesure le nouveau modèle pour l'afficher. Si l'algorithme de facettisation n'est pas suffisamment rapide pour recalculer la facettisation à un rythme proche de 25 fois par seconde, l'aspect temps-réel est perdu. Seules les modifications ne demandant pas un recalcul de la facettisation peuvent donc être faites rapidement : changements de point de vue et déplacements.

En bref, bien que l'affichage des polygones soit la seule méthode rapide actuellement disponible, elle est loin d'être satisfaisante pour toutes les applications.

1.3 Alternatives aux facettes

Une partie de la recherche en infographie vise pourtant à améliorer cette situation. D'une part, de nouveaux algorithmes, de plus en plus efficaces, permettent d'accélérer le processus de rendu direct des primitives de modélisation, sans espoir immédiat d'atteindre des vitesses interactives pour des objets complexes.

D'autre part, au niveau matériel, des tentatives sont menées pour obtenir un rendu rapide de primitives de complexité supérieure aux facettes, sans espoir immédiat, cette fois, d'atteindre un rendu temps-réel de surfaces complexes.

La description que nous faisons de ces dernières reste succincte, sachant qu'elles sont traitées en détail par Chaillou [Chaillou91], Karpf [Karpf93], Nyiri [Nyiri94] et Laporte [Laporte96].

1.3.1 Interpolation quadratique

Ces améliorations des primitives affichées en rendu rapide proviennent évidemment des progrès au niveau des circuits intégrés. La principale amélioration provient de l'utilisation d'interpolateurs quadratiques, calculant des expressions du type :

$$y = ax^2 + by^2 + cxy + dx + ey + f. \quad (1.9)$$

Utilisés à la place des interpolateurs linéaires, il est possible d'afficher une plus grande classe de surfaces, en particulier non planes. L'utilisation d'interpolateurs quadratiques peut se faire à plusieurs niveaux du rendu projectif :

- Un objet peut posséder un contour quadratique (i.e. dont les points vérifient l'équation $ax^2 + by^2 + cxy + dx + ey + f = 0$), c'est-à-dire dont le contour est un ensemble d'arcs de coniques (figure 1.10 a).
- La profondeur peut également être calculée par une expression quadratique (figure 1.10 b).

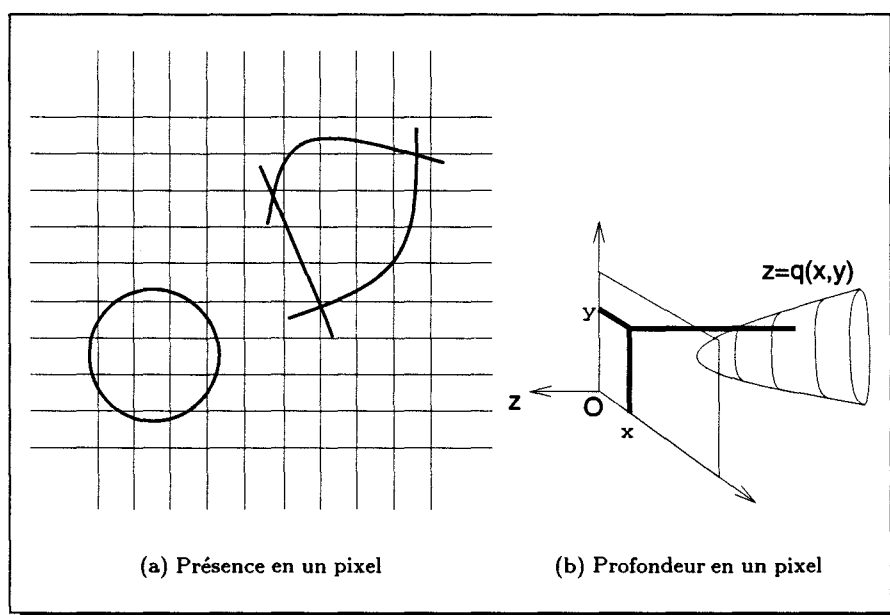


FIG. 1.10 - Utilisation d'expressions quadratiques pour l'affichage

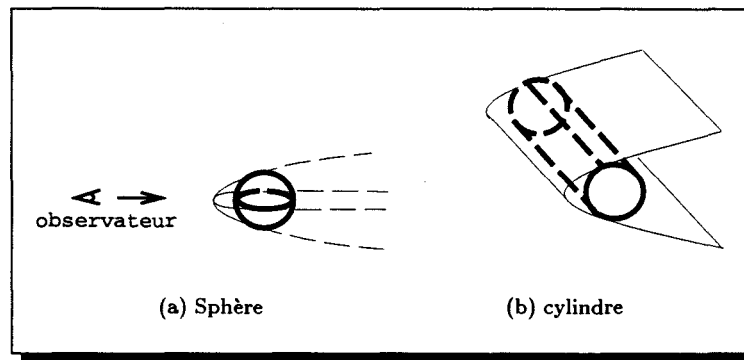
Les surfaces définies par ces deux conditions sont donc les surfaces d'équation :

$$z = ax^2 + by^2 + cxy + dx + ey + f \quad (1.10)$$

définies dans un repère particulier où le plan Oxy est le plan de projection et l'axe Oz est la direction de visée. Les surfaces décrites par cette équation sont soit des paraboloides (figure 3.3 c, page 37) soit des paraboloides cylindriques (figure 3.2 a), soit des paraboloides hyperboliques (figure 3.2 c).

Un problème se pose : la position des primitives est dépendante de l'observateur puisque, selon la définition (équation 1.10), elles sont orientées suivant l'axe Oz , l'axe de direction de vision. C'est un problème majeur puisque les primitives ne sont donc pas *invariantes par rotation* : la rotation (selon un axe différent de Oz) d'une surface définie par la même équation (1.10) ne change pas la forme de la surface mais celle-ci ne pourra plus être représentée par l'équation 1.10. Cela réduit par conséquent considérablement les possibilités de modélisation, qui n'étaient déjà pas énormes du fait du type réduit de surfaces affichables.

L'interpolation quadratique a toutefois été implémentée au sein de la machine *Pixel-Planes 5* [Fuchs et al.89]. Les paraboloides y sont utilisées pour approximer des sphères et les cylindres paraboliques pour approcher des cylindres elliptiques (figure 1.11). Cela permet d'afficher rapidement, bien qu'approximativement, les sphères et cylindres, qui sont des primitives très souvent utilisées en modélisation. Toutefois, cette méthode a été abandonnée par l'équipe de *Pixel-Planes*, les approximations n'étant pas satisfaisantes, en particulier lors de l'affichage d'intersections.

FIG. 1.11 - *Pixel-Planes 5 : utilisation des primitives quadratiques*

1.3.2 Racine carrée

La technologie permet maintenant d'envisager le cablage d'expressions de degré supérieur à 1 à des vitesses raisonnables. Il est ainsi possible d'afficher des surfaces du type $z = f(x, y)$ de degré également supérieurs. Néanmoins, le problème de l'invariance par rotation est toujours présent, ce qui limitera considérablement le nombre de surfaces affichables.

L'amélioration de cette situation, qui a constitué le point de départ du projet IMOGENE [Chaillou91, Karpf93] au LIFL, est l'utilisation d'un extracteur de racine carrée. Intégrée au sein de l'unité de calcul de la profondeur à l'aide d'interpolateurs quadratiques, elle permet de calculer la (ou les) profondeur de n'importe quelle surface algébrique quadratique.

Une surface algébrique quadratique (ou *surface quadrique*) a pour équation :

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + Gx + Hy + Iz + J = 0 \quad (1.11)$$

Par rapport aux surfaces quadratiques précédentes (équation 1.10), une surface quadrique possède des termes en z^2 , xz , yz et z . Il y a deux conséquences à cela :

- Pour un pixel (x, y) , la résolution de l'équation 1.11 donne 0, 1 ou 2 valeurs pour z :

$$\begin{cases} z_1 = -b + \sqrt{b^2 - 2c} \\ z_2 = -b - \sqrt{b^2 - 2c} \end{cases} \quad \text{où} \quad \begin{cases} b = (Ey + Fx + I)/2C \\ c = (Ax^2 + By^2 + Dxy + Gx + Hy + I)/2C \end{cases} \quad (1.12)$$

- Les surfaces quadriques sont invariantes par rotation (on trouvera une démonstration de cette propriété au chapitre 3).

L'affichage des surfaces quadriques s'effectue de la façon suivante : le contour d'une quadrique est toujours une conique, et peut donc être calculé par interpolation quadratique, comme précédemment. Les (éventuelles) profondeurs sont toujours calculées par l'intersection de la surface avec une ligne parallèle à l'axe z passant par le pixel (x, y) (figure 1.12). Les solutions sont données par l'équation 1.12.

Contrairement aux surfaces du type $z = f(x, y)$ pour lesquelles nous avons dit que le degré pouvait être élevé sans complexification énorme du matériel, l'affichage des surfaces algébriques

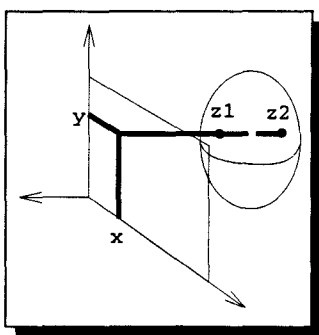


FIG. 1.12 - Profondeurs z_1 et z_2 d'une quadrique en un pixel (x, y)

de degré supérieur à deux représente une difficulté importante qui augmente beaucoup avec le degré, puisque le calcul des profondeurs² nécessite des calculs beaucoup plus complexes : une résolution directe des z nécessite l'extraction de racines cubiques ou quartiques pour les surfaces de degré 3 ou 4 ; pour des degrés supérieurs, il faut utiliser des méthodes numériques (Newton, etc.) très coûteuses en nombre de calculs. Quant aux surfaces paramétriques, elles sont de degré bien supérieur (cf. chapitre 3). Les quadriques représentent par conséquent la solution la plus raisonnable actuellement pour l'affichage matériel d'objets courbes.

1.4 L'affichage matériel des quadriques

Se dire que l'on est capable de résoudre en temps-réel le calcul du contour et des profondeurs pour une quadrique n'est cependant pas suffisant pour construire un système qui sera intéressant pour la synthèse d'images. Trois questions se posent, relatives aux considérations sur la modélisation au début de ce chapitre : comment limiter les surfaces, peut-on afficher les quadriques sous forme volumique, et quelles sont les possibilités d'assemblage ? Nous répondons à ces questions sans entrer dans les détails, l'étude complète de ces questions et du problème de l'affichage matériel en général sont traitées par H. Laporte [Laporte96].

1.4.1 Limiter les quadriques

Les surfaces quadriques (figures 3.1, 3.2 et 3.3, pages 36 et 37) sont toutes, sauf l'ellipsoïde, infinies dans au moins une direction. Si on ne limite pas cette extension, la possibilité de représentation d'objets quelconques, en général finis, est considérablement réduite. La façon la plus simple de limiter la surface est d'utiliser un ensemble de plans, appelés *plans de coupe* et disposés de façon à diviser la quadrique en deux : une partie visible et une partie invisible. Si l'on utilise plusieurs plans de coupe, l'intersection de toutes les parties visibles forme le "morceau" de quadrique à garder.

Ce mécanisme est utilisé par l'algorithme d'affichage. Les plans de coupe sont en fait représentés par un demi-espace, qui se définit par une inéquation linéaire : $Ax + By + Cz + D < 0$. L'intersection de tous les demi-espaces représente un polyèdre de coupe dont l'intersection avec la quadrique donne le morceau visible. Lors du calcul des profondeurs, l'unité vérifie grâce aux inéquations linéaires (projetées) des espaces de coupe si le ou les points obtenus appartiennent ou non au volume de coupe. Si oui, le point est conservé pour l'étape suivante (l'élimination des parties

2. En géométrie algébrique, le théorème de Bezout indique qu'il y a au plus n profondeurs, i.e. n intersections avec une droite, pour une surface algébrique de degré n

cachées), sinon le point est ignoré. Le nombre maximal de plans de coupes par quadrique sera évidemment limité à une valeur constante dépendant du matériel.

1.4.2 Surfaces/volumes quadriques

Comme nous l'avons vu précédemment, les primitives définies sous forme implicite peuvent se présenter sous forme de surface ou de volume, selon que l'on ait à faire à une équation ou une inéquation. Dans le cas des quadriques, cela reste bien entendu vrai. Au niveau de l'affichage tel que nous l'avons présenté, la primitive peut être traitée a priori sous l'une ou l'autre forme. Ce qui changera dans le rendu, c'est l'affichage ou non des plans de coupe (figure 1.13). Il y a donc peu de changements dans l'algorithme et le même matériel peut gérer les deux formes. Encore une fois nous référons le lecteur à Laporte [Laporte96] pour plus de détails.

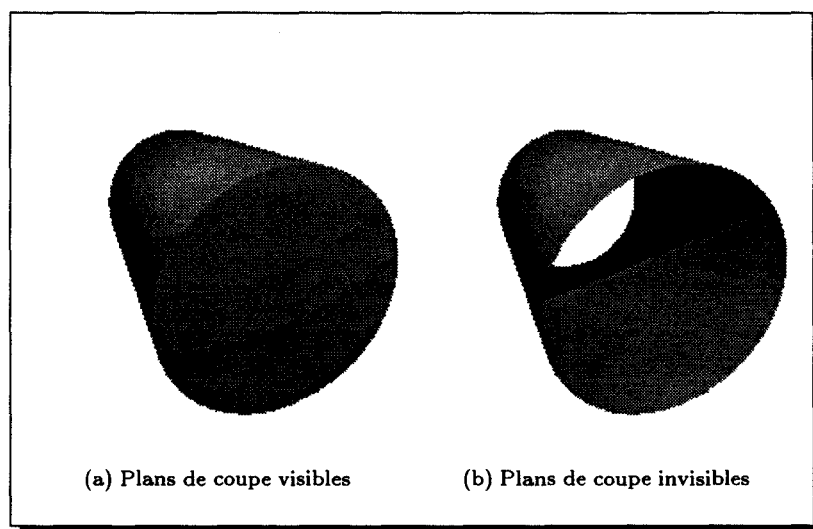


FIG. 1.13 - *Cylindre volumique et surfacique*

1.4.3 Opérations booléennes

À partir du moment où des primitives volumiques sont utilisées, il faut se poser la question des possibilités de construction par opérations booléennes (§ 1.1.1). Avec des primitives quadriques, la modélisation volumique avec CSG a longtemps été et est encore très utilisé pour la CAO/CFAO [Miller86, VK93].

Dans notre cas, l'union est traitée implicitement par l'algorithme du Z-buffer, mais les opérations d'intersection et de différence demandent un traitement particulier : il faut, en chaque pixel, déterminer la profondeur cherchée en parcourant l'arbre CSG pour déterminer quelle primitive est visible. Cette opération prend un temps non négligeable (et surtout non constant puisque l'arbre n'a pas de profondeur fixée), et donc rend impossible la cadence temps-réel [Preux91]. Contrairement à d'autres projets qui ont abandonné la vitesse au profit des opérations CSG (Pixel-Planes [Fuchs et al.89] et la Ray Casting Machine [Ellis et al.91]), le projet d'affichage de quadriques au LIFL a préféré la vitesse. La modélisation volumique s'en trouve considérablement réduite. C'est pour cette raison que toutes les méthodes proposées dans les chapitres suivants se basent sur une modélisation surfacique des quadriques.

1.5 Les problèmes logiciels

Cette section, qui introduit le sujet de cette thèse, concerne les conséquences au niveau logiciel de l'affichage temps-réel des quadriques. À partir du moment où l'on conçoit un accélérateur graphique pour les quadriques, il faut, en amont du processus d'affichage, exploiter au maximum les possibilités de celui-ci.

Le problème au niveau de la modélisation est la *création d'objets complexes à l'aide de quadriques*. Actuellement, tous les objets que l'on désire afficher avec un accélérateur 3D classique sont facettisés (ou polygonalisés). De la même façon, nous cherchons à approcher des modèles complexes à l'aide de quadriques, pour n'importe quelle application, en exploitant les propriétés supplémentaires offertes.

Dans les chapitres suivants, nous voyons ce problème sous deux angles : la création directe de modèles quadriques et l'approximation de modèles plus complexes (figure 1.14).

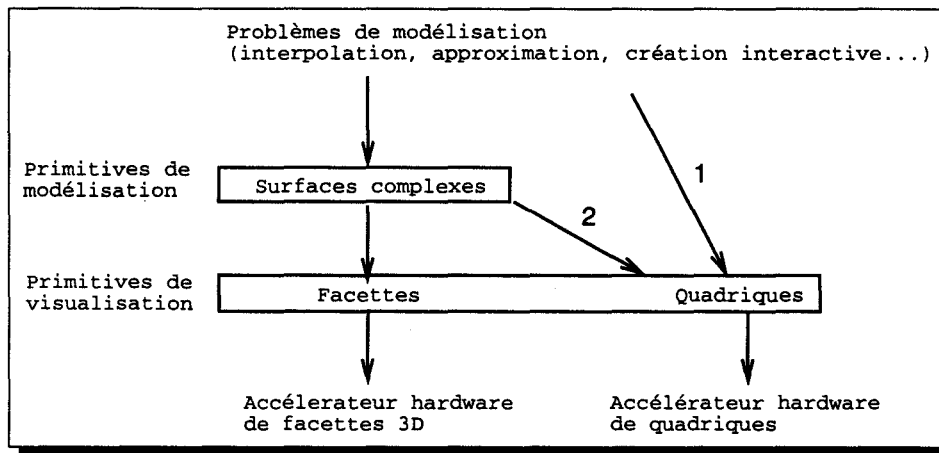


FIG. 1.14 - Pipelines facette et quadrique

- La création directe de modèles quadriques (flèche 1) est la méthode a priori la plus simple : le modèle quadrique est créé en tant qu'approximation d'un objet réel ou pour répondre à des contraintes géométriques ou esthétiques. Les quadriques sont ensuite affichées directement par l'accélérateur. Cette approche sera étudiée au chapitre 4.
- Toutefois, on sait bien que les quadriques ne répondent pas à tous les problèmes de modélisation, et on a souvent besoin de surfaces plus complexes (par exemple paramétriques) pour modéliser correctement. Dans ce cas, il sera nécessaire d'approcher ces surfaces par des quadriques pour l'affichage (flèche 2). C'est l'équivalent de la facettisation. Cette approche sera vue au chapitre 5.

Bilan

À l'heure actuelle, on est encore loin de pouvoir modéliser une scène complète de façon interactive avec visualisation temps-réel. À la complexité des primitives de modélisation s'oppose la simplicité des primitives de visualisation affichables avec accélérateurs. Mais, des efforts sont fait dans les deux sens pour réduire cette différence : d'une part, une partie de la recherche sur les surfaces s'oriente actuellement sur les surfaces de degré faible, aussi bien paramétriques [Sederberg et al.85],

qu'implicites [Bajaj et al.95a, Dahmen et al.93]. D'autre part, comme nous l'avons décrit, il est possible d'afficher rapidement des objets plus complexes que les facettes 3D. Il nous semble qu'actuellement, les primitives quadriques représentent un choix intéressant entre ces deux tendances. Dans les chapitres suivants, nous tentons de valider ce choix, au moins en ce qui concerne la partie logicielle.

Le chapitre 2 présente certaines notions de base à propos des surfaces de Bézier. Bien que n'ayant pas de rapport direct avec les surfaces quadriques, les notions qui y sont présentées sont utilisées dans les chapitres ultérieurs, en particulier lors de l'étude des nouvelles représentations des quadriques.

Chapitre 2

Courbes et surfaces de Bézier

Les courbes et surfaces de Bézier sont parmi les primitives de modélisation les plus répandues en modélisation géométrique. Développées indépendamment par Pierre Bézier chez Renault et par Paul De Casteljaou chez Citroën, elles représentent une grande partie de l'énorme progrès de la CAO accompli à la fin des années 50, grâce à l'industrie automobile.

Définies de façon paramétrique et non implicite, les surfaces de Bézier n'ont pas d'utilisation directe dans les algorithmes étudiés dans les chapitres suivants. Par contre, la représentation de Bernstein-Bézier des surfaces algébriques (chapitre 3), essentielle pour la plupart des méthodes relatives aux quadriques présentées aux chapitres 4 et 5, se base sur l'utilisation des triangles de Bézier fonctionnels. Ce chapitre a donc pour but de présenter (ou rappeler) au lecteur ces derniers.

Ce chapitre est largement inspiré des ouvrages de G. Farin [Farin92], J.-C. Fiorot et P. Jeanin [Fiorot et al.92] et P. Schweizer [Schweizer87], dans lesquels on trouve des descriptions plus complètes et plus détaillées ainsi que toutes les démonstrations (volontairement) omises ici.

2.1 Courbes de Bézier

2.1.1 Définition

Soit $P = (P_0, P_1, \dots, P_n)$ un ensemble de points du plan. La courbe de Bézier C définie par ces points s'écrit sous forme paramétrique :

$$C(t) = \sum_{i=0}^n B_i^n(t) P_i \quad t \in [0, 1] \quad (2.1)$$

où les fonctions B sont les *polynômes de Bernstein*, définis par :

$$B_i^n(t) = \frac{n!}{i!(n-i)!} (1-t)^{n-i} t^i \quad i = 0 \text{ à } n \quad (2.2)$$

Chaque point de la courbe $C(t)$ peut être considéré le barycentre des points P_i , pondéré par les fonctions $B_i(t)$. P est appelé *polygone de contrôle* et les points P_i *points de contrôle* de la courbe. La position de ces points (il y en a $n+1$ pour une courbe de degré n) définit entièrement la forme de la courbe (figure 2.1), ce qui permet de choisir cette forme en ne manipulant que les points. En outre, toutes les grandeurs géométriques d'une courbe (comme par exemple ses dérivées), s'écrivent comme des expressions dépendant exclusivement des coordonnées de ses points de contrôle.

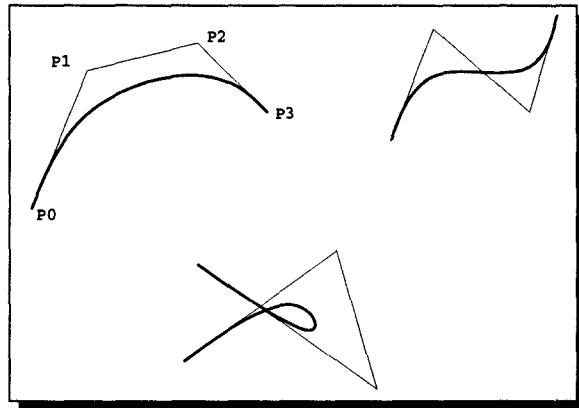


FIG. 2.1 - Courbes de Bézier cubiques ($n = 3$) et leur polygone de contrôle

2.1.2 Les polynômes de Bernstein sur $[0, 1]$

Bien que P. Bézier ait, à l'origine, défini les courbes qui portent son nom de façon géométrique, Forrest, Gordon et Riesenfeld ont montré que cette famille de courbes pouvait être définie à l'aide de base des polynômes de Bernstein [Farin92]. À partir des propriétés de ces polynômes, il est alors possible de démontrer les propriétés géométriques des courbes de Bézier.

Les polynômes de Bernstein de degré n forment une base de l'espace des polynômes de degré n à une variable, de la même façon que la base canonique standard $1, t, t^2, \dots, t^n$. Ceci implique principalement que toute courbe paramétrique exprimée sous la forme canonique standard peut être représentée par une courbe de Bézier. La table 2.1 fait la comparaison entre les deux bases.

Degré	Base canonique	Base de Bernstein
0	1	1
1	1, t	1 - t, t
2	1, t, t ²	(1 - t) ² , 2(1 - t)t, t ²
3	1, t, t ² , t ³	(1 - t) ³ , 3(1 - t) ² t, 3t ² (1 - t), t ³
⋮	⋮	⋮
n	t ⁱ , i = 0..n	B _i ⁿ (t), i = 0..n

TAB. 2.1 - Bases de polynômes

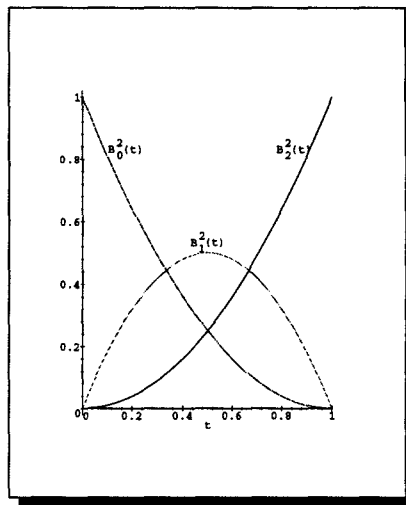


FIG. 2.2 - Graphes des polynômes de Bernstein de degré 2

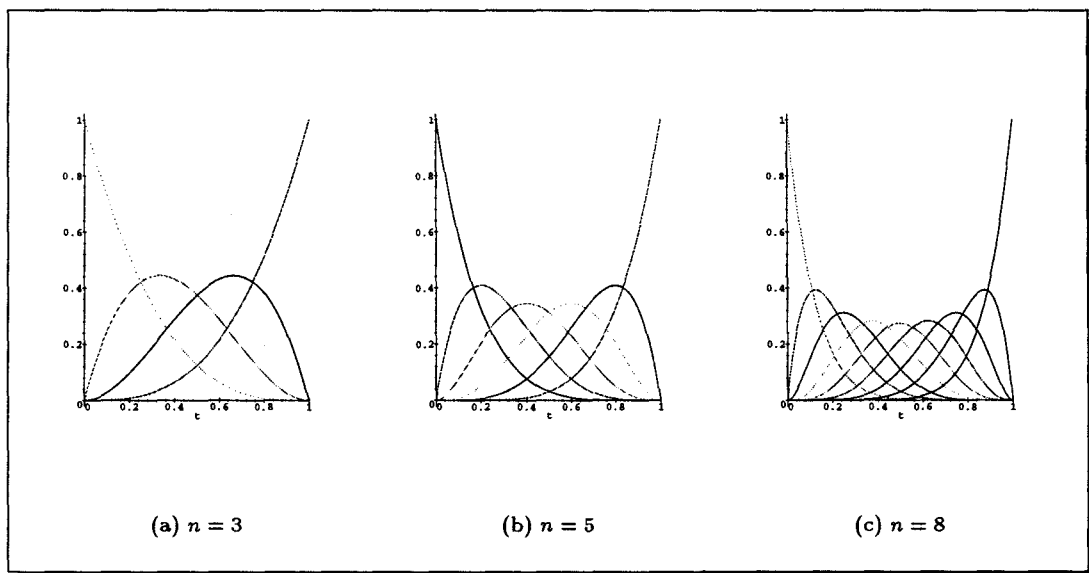


FIG. 2.3 - Graphes des polynômes de Bernstein de degrés 3, 5 et 8

Les polynômes de Bernstein (figures 2.2 et 2.3) ont les propriétés suivantes :

1. Positivité : les polynômes sont positifs sur $[0, 1]$,
 $\forall t \in [0, 1], B_i^n(t) \geq 0$
2. Symétrie : les polynômes sont symétriques par rapport à $t = 1/2$,
 $\forall t \in \mathbb{R}, B_i^n(1-t) = B_i^n(t)$
3. Partition de l'unité : pour tout t , la somme des B_i^n est toujours 1,
 $\forall t \in \mathbb{R}, \sum_i B_i^n(t) = 1$
4. Maxima : chaque polynôme $B_i^n(t)$ admet un maximum dans $[0, 1]$ en i/n . Par exemple, pour les $n=2$ (figure 2.2), les maxima de chaque polynôme sont atteints en $t = 0$ pour B_0^2 , $1/2$ pour B_1^2 et 1 pour B_2^2 .
5. Il existe une relation de récurrence pour le calcul de $B_i^n(t)$:
 $\forall t \in \mathbb{R}, B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$ (lors du calcul récursif, on considère que les termes B_i^n où i est négatif sont nuls. La récursion s'arrête avec $B_1^0(t) = 0$ et $B_0^0(t) = 1$).

2.1.3 Propriétés des courbes de Bézier

A partir des propriétés des polynômes de Bernstein, on démontre les propriétés suivantes des courbes de Bézier :

- La courbe interpole les premier et dernier points de contrôle P_0 et P_n . Pour preuve, on vérifie que $C(0) = P_0$ et $C(1) = P_n$. De plus, la courbe est tangente en P_0 au premier segment du polygone de contrôle ($[P_0P_1]$), et elle est tangente en P_n au dernier segment ($[P_{n-1}P_n]$). On le vérifie en calculant la dérivée en P_0 et P_n : $C'(0) = n(P_1 - P_0)$ et $C'(1) = n(P_n - P_{n-1})$.
- Invariance affine : une transformation affine du plan (translation, rotation, mise à l'échelle) appliquée à une courbe peut s'appliquer soit sur les points de la courbe, soit sur les points de contrôle. Par exemple, une courbe à laquelle on applique une translation peut se calculer soit en calculant les points de la courbe puis en leur appliquant une translation, soit en appliquant la translation aux points de contrôle puis en calculant les points de la courbe.
- Enveloppe convexe : la courbe appartient à l'enveloppe convexe de ses points de contrôle (figure 2.4). La combinaison barycentrique des points de contrôle que représente l'équation 2.1 pour tout t ne contient que des coefficients positifs (les valeurs des polynômes de Bernstein en t , toujours positifs pour $t \in [0, 1]$), donc le barycentre $C(t)$ est bien compris dans l'enveloppe convexe.
- Symétrie : la courbe est symétrique par rapport aux points de contrôle (autrement dit invariante par inversion $t \rightleftharpoons 1-t$). Ceci provient directement de la propriété de symétrie des polynômes de Bernstein.
- Influence des points de contrôle : l'influence maximale du point de contrôle P_i se situe sur la courbe en $t = i/n$ (d'après les maxima des polynômes de Bernstein). Ceci a pour conséquence qu'un point de contrôle influence principalement la partie de la courbe la plus proche de lui.
- Diminution de la variation : le nombre d'intersections de la courbe avec une droite quelconque est inférieur au nombre d'intersections de cette droite avec le polygone de contrôle. Autrement dit, la courbe "adoucit" le polygone de contrôle.

Ces propriétés font que les courbes de Bézier sont des outils très puissants car très intuitifs pour la CAO 2D.

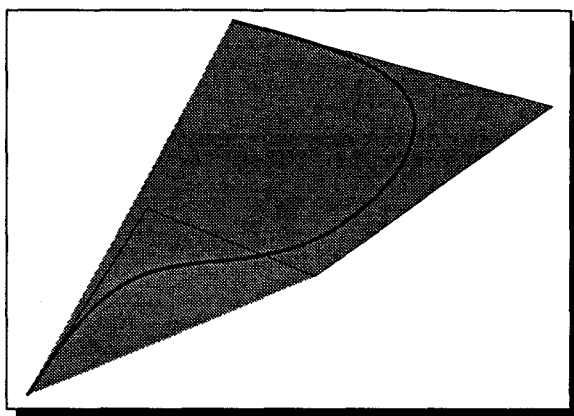


FIG. 2.4 - Courbe de Bézier de degré 4 et son enveloppe convexe (en grisé)

2.1.4 L'algorithme de De Casteljau

L'algorithme de De Casteljau est fondamental pour les courbes de Bézier car il permet non seulement de calculer $P(t)$ pour tout t beaucoup plus rapidement que par l'évaluation directe de l'équation 2.1, mais également d'effectuer facilement certaines opérations courantes sur une courbe : la *subdivision*, qui consiste à "couper" une courbe en un t donné en deux courbes de même degré qui, réunies sont identiques à la courbe de départ; la *polygonalisation* d'une courbe, qui permet l'approximation de celle-ci par un ensemble de segments de droites, dans le but d'afficher la courbe sur un écran ou un traceur; et finalement, l'algorithme de De Casteljau permet de connaître la tangente en un point de la courbe.

L'algorithme prend en entrée les $n + 1$ points de contrôle et une valeur du paramètre t , et calcule n points intermédiaires par interpolations linéaires des points de contrôle. Ce processus est répété en interpolant également les points intermédiaires, et ainsi de suite $n - 1$ fois jusqu'à obtenir le dernier point : $P(t)$. L'algorithme est illustré par le "triangle" (ici pour $n = 3$) :

$$\begin{array}{l} P_0 \\ P_1 \\ P_2 \\ P_3 \end{array} = \begin{array}{|c|} \hline P_0^0 \\ \hline P_1^0 \\ \hline P_2^0 \\ \hline P_3^0 \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline P_0^1 \\ \hline P_1^1 \\ \hline P_2^1 \\ \hline \end{array} \quad \begin{array}{l} P_0^2 \\ P_1^2 \end{array} \quad P_0^3 = P(t)$$

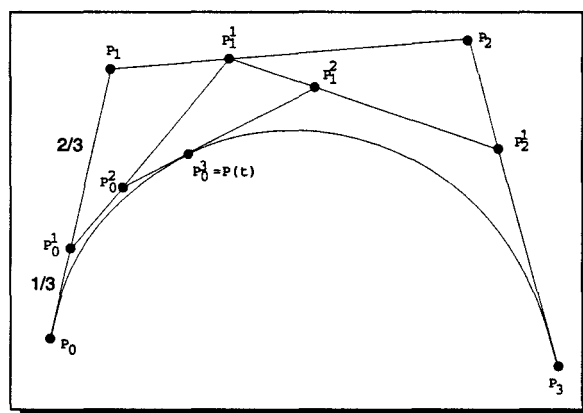
On passe d'une colonne à la suivante par interpolations linéaires. Ici : $P_0^1 = (1 - t)P_0^0 + tP_1^0$. De façon générale :

$$P_i^j = (1 - t)P_i^{j-1} + tP_{i+1}^{j-1} \quad (2.3)$$

Cette formule dérive de l'équation de récurrence des polynômes de Bernstein. L'algorithme possède une interprétation géométrique simple (figure 2.5).

Comme nous l'avons déjà évoqué, l'algorithme de De Casteljau présente quelques propriétés fondamentales pour l'utilisation des courbes de Bézier.

- L'algorithme est numériquement plus stable pour le calcul de $P(t)$ que l'équation 2.1 : les approximations numériques affectent plus la forme canonique que la forme de Bernstein

FIG. 2.5 - Algorithme de De Casteljau pour $t = 1/3$

([Farin92], p. 418). En effet, l'évaluation de $P(t)$ à l'aide de l'algorithme de De Casteljau nécessite l'utilisation de $(n^2 + n)/2$ interpolations linéaires plutôt que l'évaluation de $n + 1$ polynômes de degré n pour la forme canonique.

- La droite passant par les points P_0^{n-1} et P_1^{n-1} (l'avant-dernière colonne du triangle) est tangente à $P(t)$.
- L'algorithme fournit les polygones de contrôle des deux courbes de Bézier qui subdivisent la courbe P au point $P(t)$: $\{P_0^0, P_0^1, \dots, P_0^n\}$ et $\{P_n^0, P_{n-1}^1, \dots, P_0^n\}$ (figure 2.5).

On utilise également les points intermédiaires générés par l'algorithme pour obtenir une approximation linéaire de la courbe, très souvent utilisée pour l'affichage sur écran ou traceur : une première approximation est fournie par le polygone $P_0^0 P_0^1 \dots P_0^n P_1^{n-1} P_2^{n-2} \dots P_{n-1}^1 P_n^0$ (figure 2.5). Cette approximation peut-être améliorée en utilisant récursivement le processus de subdivision décrit précédemment. En effet, on démontre que les approximations successives convergent vers la courbe. En pratique, quatre ou cinq itérations suffisent dans la plupart des cas.

2.2 Surfaces de Bézier

Les premières surfaces mathématiques à offrir un outil puissant pour la conception d'objets complexes furent les surfaces de Coons ([Rogers et al.90], p. 426). Toutefois, l'utilisation de ces surfaces restait plus ou moins complexe à cause de la manipulation de termes peu homogènes ni intuitifs comme les vecteurs tangents et les vecteurs de torsion. La définition de surfaces de Bézier, qui à l'instar des courbes n'utilisent que des points de contrôle, a permis de dépasser ces limitations.

Les principes des courbes de Bézier peuvent facilement être étendus aux surfaces. Cette généralisation peut s'effectuer de deux manières différentes, donnant naissance à deux classes de surfaces de Bézier : les *carreaux* et les *triangles*.

2.2.1 Carreaux

Définition

On considère une courbe de Bézier de degré n dans l'espace. Si l'on déplace chacun des $n + 1$ points de contrôle le long de $n + 1$ autres courbes de Bézier (de degré m), l'ensemble des points parcourus par la courbe forme un morceau de surface ayant la forme d'un carré déformé, appelé carreau (figure 2.6). Ce carreau est entièrement défini par le réseau de $(n + 1) \times (m + 1)$ points de contrôle et paramétré par deux variables (s, t) parcourant $[0, 1] \times [0, 1]$.

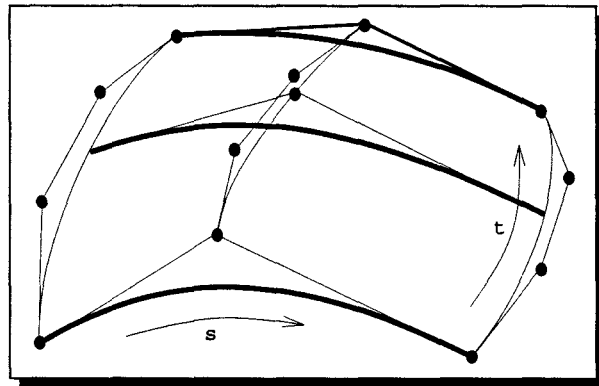


FIG. 2.6 - Construction d'un carreau de Bézier ($n = 2$ et $m = 3$)

L'équation paramétrique d'un carreau de Bézier s'obtient en faisant le produit cartésien de deux courbes (on dit également produit tensoriel).

$$S(s, t) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(s) B_j^n(t) P_{i,j} \quad s, t \in [0, 1] \quad (2.4)$$

Les points de contrôle $P_{i,j}$ forment le réseau de $m \times n$ points de contrôle (figure 2.7).

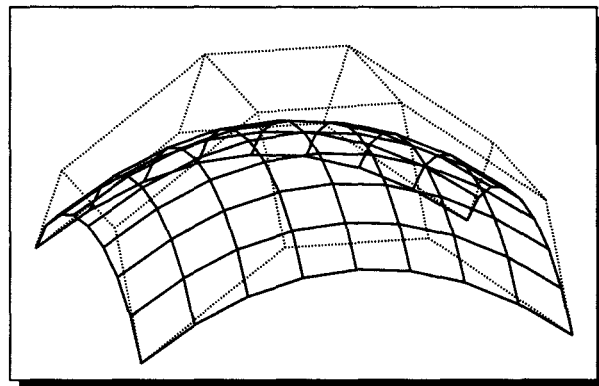


FIG. 2.7 - Carreau de Bézier ($n = m = 3$)

Propriétés

Les carreaux de Bézier conservent la plupart des propriétés des courbes, dérivées des propriétés des polynômes de Bernstein $B_i^m(s)B_j^n(t)$:

- Interpolation : la surface interpole les 4 coins du polygone de contrôle et est tangente en ces points aux quatre plans extrémités du polygone de contrôle.
- Invariance affine : la propriété est équivalente pour toute transformation affine de l'espace.
- Enveloppe convexe : le carreau est entièrement contenu dans le polyèdre qui forme l'enveloppe convexe du polygone de contrôle.
- Symétrie : le carreau est invariant par inversion $s \rightleftharpoons (1 - s)$ et $t \rightleftharpoons (1 - t)$.
- Influence des points de contrôle : le polynôme $B_i^m(s)B_j^n(t)$ admet un maxima dans $[0, 1]^2$ en $(s, t) = (i/m, j/n)$. Par conséquent, le point $P_{i,j}$ a une influence maximale sur la partie du carreau la plus proche de lui.

Seule la diminution de la variation ne se généralise pas aux surfaces. Néanmoins, on peut toujours dire que de façon intuitive la surface "adoucit" son polygone de contrôle. On retrouve donc la même facilité que pour les courbes pour la modélisation "à la main".

L'algorithme de De Casteljau

L'algorithme de De Casteljau pour les courbes se généralise simplement aux carreaux. Le triangle devient une pyramide dont la base est toujours le polygone de contrôle et le sommet est $P(s, t)$, on passe d'un "étage" au suivant en faisant une interpolation bilinéaire de quatre points :

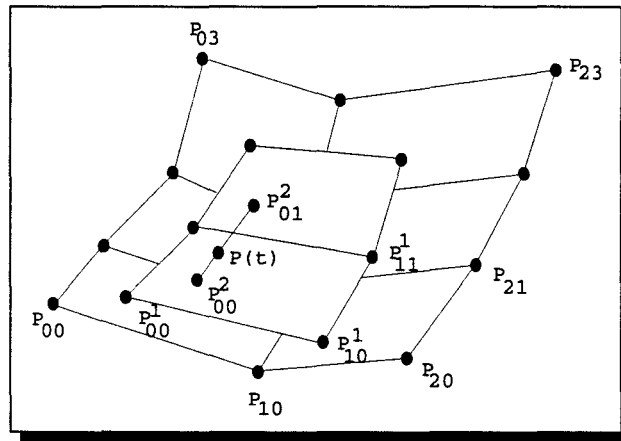
$$P_{i,j}^r = [1 - s \quad s] \begin{bmatrix} P_{i,j}^{r-1} & P_{i,j+1}^{r-1} \\ P_{i+1,j}^{r-1} & P_{i+1,j+1}^{r-1} \end{bmatrix} \begin{bmatrix} 1 - t \\ t \end{bmatrix} \tag{2.5}$$

Par exemple pour $n = 2, m = 3$ (figure 2.8) :

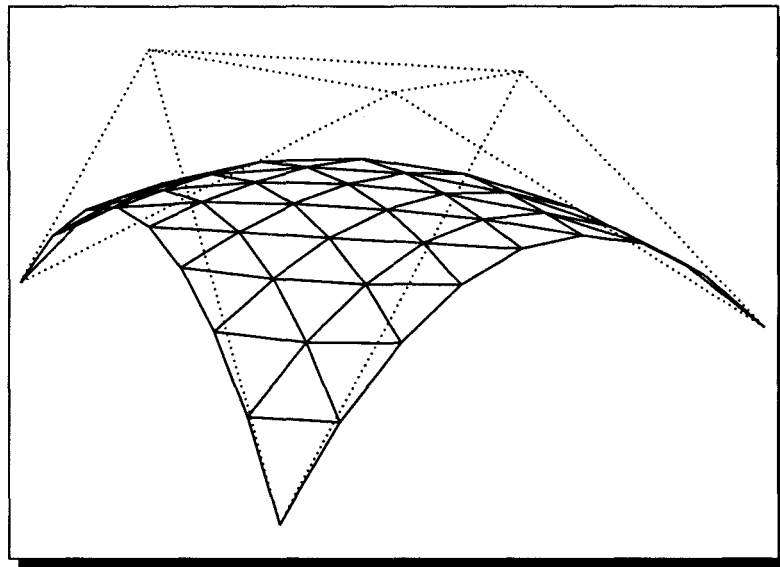
$$\begin{matrix} P_{0,0} & P_{0,1} & P_{0,2} \\ P_{1,0} & P_{1,1} & P_{1,2} \\ P_{2,0} & P_{2,1} & P_{2,2} \\ P_{3,0} & P_{3,1} & P_{3,2} \end{matrix} = \begin{matrix} \boxed{P_{0,0}^0} & \boxed{P_{0,1}^0} & P_{0,2}^0 \\ \boxed{P_{1,0}^0} & \boxed{P_{1,1}^0} & P_{1,2}^0 \\ P_{2,0}^0 & P_{2,1}^0 & P_{2,2}^0 \\ P_{3,0}^0 & P_{3,1}^0 & P_{3,2}^0 \end{matrix} \Rightarrow \begin{matrix} \boxed{P_{0,0}^1} & P_{0,1}^1 \\ \boxed{P_{1,0}^1} & \boxed{P_{1,1}^1} \\ P_{2,0}^1 & P_{2,1}^1 \end{matrix} \Rightarrow \begin{matrix} P_{0,0}^2 \\ \boxed{P_{1,0}^2} \end{matrix} \Rightarrow P_{0,0}^3 = S(s, t)$$

2.2.2 Triangles

Historiquement, les triangles de Bézier (figure 2.9) ont été la première extension des courbes aux surfaces. Les triangles sont à présent moins utilisés que les carreaux, sans doute parce que les premières applications en CAO étaient dans le cadre de l'industrie automobile où les pièces de carrosserie sont souvent de nature rectangulaire.

FIG. 2.8 - *Algorithme de De Casteljau*

A présent, des objets de géométrie quelconque sont conçus à l'aide de surfaces de Bézier et les formes triangulaires s'adaptent mieux dans de nombreux cas¹, en particulier dans les problèmes d'interpolation des points. Avant de définir les triangles de Bézier, il est nécessaire de faire un bref rappel sur les coordonnées barycentriques et de présenter les polynômes de Bernstein à deux variables.

FIG. 2.9 - *Triangle de Bézier quadratique*

Coordonnées barycentriques

Considérons un triangle V_0, V_1, V_2 du plan. Tout point P du plan peut être écrit sous la forme d'une combinaison barycentrique des V_i :

1. Il est toujours possible d'obtenir des triangles à partir de carreaux, en confondant tous les points de contrôle d'un même bord, par exemple. Mais ce genre de méthode engendre des singularités et rend le triangle non symétrique

$$P = \lambda_0 V_0 + \lambda_1 V_1 + \lambda_2 V_2 \quad \text{où} \quad \lambda_0 + \lambda_1 + \lambda_2 = 1 \quad (2.6)$$

les coefficients λ_0, λ_1 et λ_2 sont appelés *coordonnées barycentriques* du point P . Cette notion se généralise à tout espace de dimension d : une base barycentrique y est définie par $d + 1$ points linéairement indépendants V_0, V_1, \dots, V_d . Tout point P de cet espace est défini par ses coordonnées barycentriques $\lambda_0, \lambda_1, \dots, \lambda_d$, telles que :

$$P = \sum_{i=0}^d \lambda_i V_i \quad \text{où} \quad \sum_{i=0}^d \lambda_i = 1 \quad (2.7)$$

Polynômes de Bernstein à deux variables

Les polynômes de Bernstein se généralisent à deux variables (en réalité trois variables barycentriques²) :

$$B_{i,j,k}^n(\lambda_0, \lambda_1, \lambda_2) = \frac{n!}{i!j!k!} \lambda_0^i \lambda_1^j \lambda_2^k \quad \text{où} \quad i + j + k = n \quad (2.8)$$

Ces polynômes conservent toutes les propriétés (généralisées) de la version à une variable: ils sont positifs, symétriques par rapport aux λ_i , partitionnent l'unité et chaque $B_{i,j,k}^n(\lambda_0, \lambda_1, \lambda_2)$ admet un maximum en $(i/n, j/n, k/n)$. La formule de récursion généralisée est :

$$B_{i,j,k}^n(\Lambda) = \lambda_0 B_{i-1,j,k}^n(\Lambda) + \lambda_1 B_{i,j-1,k}^n(\Lambda) + \lambda_2 B_{i,j,k-1}^n(\Lambda) \quad \text{où} \quad \Lambda = (\lambda_0, \lambda_1, \lambda_2) \quad (2.9)$$

Définition

Un triangle de Bézier S est défini par :

$$S(\Lambda) = \sum_{i+j+k=n} B_{i,j,k}^n(\Lambda) P_{i,j,k} \quad (2.10)$$

Les paramètres ne sont plus $(s, t) \in [0, 1]^2$, mais $(\lambda_0, \lambda_1, \lambda_2) \in [0, 1]^3$, tels que $\lambda_0 + \lambda_1 + \lambda_2 = 1$. Cet espace forme un triangle de l'espace des paramètres. La surface est donc la déformation d'un triangle dans l'espace.

Les points de contrôle $P_{i,j,k}$ où $i + j + k = n$ forment un polygone de contrôle triangulaire (figure 2.10).

Propriétés

Encore une fois on retrouve les propriétés généralisées des courbes de Bézier : interpolation (des trois extrémités du polygone de contrôle), invariance affine, enveloppe convexe, symétrie, influence des points de contrôle (chaque polynôme de Bernstein $B_{i,j,k}^n(\Lambda)$ admettant un maximum en $\Lambda = (i/n, j/n, k/n)$).

2. On peut toujours revenir à deux variables en remplaçant par exemple λ_2 par $(1 - \lambda_0 - \lambda_1)$

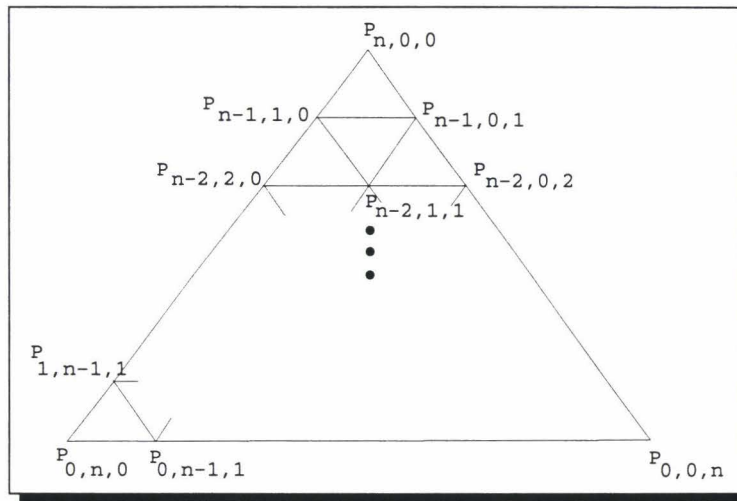


FIG. 2.10 - Polygone de contrôle d'un triangle de Bézier quadratique

Algorithme de De Casteljau

L'algorithme de De Casteljau pour les courbes se généralise également pour les triangles. Ici l'interpolation est linéaire et utilise l'équation 2.9. Pour une surface cubique, on obtient :

$$\begin{array}{ccccccc}
 & & & & P_{3,0,0}^0 & & \\
 & & & & \boxed{P_{2,1,0}^0} & \boxed{P_{2,0,1}^0} & \\
 & & & & P_{1,2,0}^0 & P_{1,1,1}^0 & P_{1,0,2}^0 \\
 P_{3,0,0} & P_{2,1,0} & P_{2,0,1} & = & P_{0,3,0}^0 & P_{0,2,1}^0 & P_{0,1,2}^0 & P_{0,0,2}^0 \\
 P_{1,2,0} & P_{1,1,1} & P_{1,0,2} & & & & & \\
 P_{0,3,0} & P_{0,2,1} & P_{0,1,2} & P_{0,0,2} & & & & \\
 & & & & & & & \\
 & & & & P_{1,0,0}^2 & & & \\
 P_{0,1,0}^2 & P_{0,0,1}^2 & \Rightarrow & P_{0,0,0}^3 = S(\lambda_0, \lambda_1, \lambda_2) & & & &
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 & & \boxed{P_{2,0,0}^1} \\
 & & P_{1,1,0}^1 & P_{1,0,1}^1 \\
 P_{0,2,0}^1 & P_{0,1,1}^1 & P_{0,0,2}^1
 \end{array}
 \Rightarrow$$

Nous utilisons les triangles de Bézier de préférence aux carreaux parce que les triangles de degré n permettent de décrire toutes les surfaces paramétriques de degré n , contrairement aux carreaux qui n'en décrivent qu'un sous ensemble (cf. chapitre 3).

2.3 Courbes et surfaces de Bézier fonctionnelles

Les courbes qui sont les graphes de fonctions de la forme $y = f(x)$ sont dites *fonctionnelles* ou *non paramétriques* ([Farin92], p.70). Elles sont en fait un cas particulier des fonctions paramétriques où l'espace des paramètres est situé sur l'axe des abscisses, i.e. $t = x$:

$$\text{Courbe paramétrique: } \begin{cases} x = f(t) \\ y = g(t) \end{cases} \quad \text{Courbe fonctionnelle: } \begin{cases} x = t \\ y = g(t) \end{cases} \quad (2.11)$$

Si la fonction g est un polynôme de degré n , son graphe peut donc être représenté par une courbe de Bézier de degré n , avec une disposition particulière de $n+1$ points de contrôle : l'équation 2.11 s'écrit sous forme de Bézier

$$C(t) = \begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=0}^n \begin{bmatrix} P_i^x \\ P_i^y \end{bmatrix} B_i^n(t) \quad (2.12)$$

où P_i^x et P_i^y sont les coordonnées du point P_i . On en déduit les abscisses des points de contrôle :

$$t = \sum_{i=0}^n P_i^x B_i^n(t) \Leftrightarrow P_i^x = i/n \quad (2.13)$$

Les ordonnées P_i^y définissent la fonction g et par conséquent la forme de la courbe (figure 2.11).

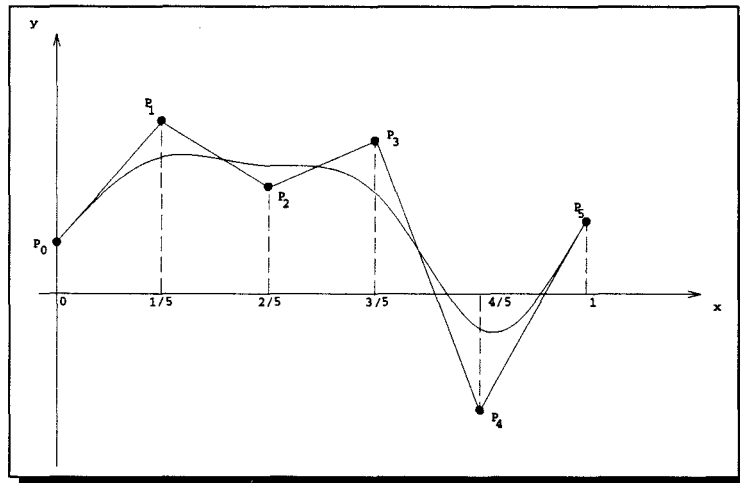


FIG. 2.11 - courbe de Bézier fonctionnelle d'ordre 5

Une nouvelle fois, cette représentation se généralise aux surfaces. Pour les carreaux ([Farin92], p. 286), l'espace des paramètres ($[0, 1]^2$) se situe sur le plan xOy et les coordonnées du point de contrôle $P_{i,j}$ sont $(i/m, j/n, P_{i,j}^z)$ (figure 2.12 a).

Pour les triangles (figure 2.12 b), la représentation fonctionnelle des courbes se généralise de la même façon [Farin92], p. 340) : l'espace des paramètres est un triangle du plan xOy . Les points de la surface et les points de contrôle sont repérés par trois coordonnées barycentriques sur le plan xOy et par une coordonnée z . L'équation de la surface est :

$$S(\lambda_0, \lambda_1, \lambda_2) = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ z \end{bmatrix} = \sum_{i+j+k=n} \begin{bmatrix} i/n \\ j/n \\ k/n \\ P_{i,j,k}^z \end{bmatrix} B_{i,j,k}^n(\lambda_0, \lambda_1, \lambda_2) \quad (2.14)$$

Les points de contrôle ont donc pour coordonnées barycentriques sur le plan $(i/n, j/n, k/n)$ et comme coordonnée z : $P_{i,j,k}^z$.

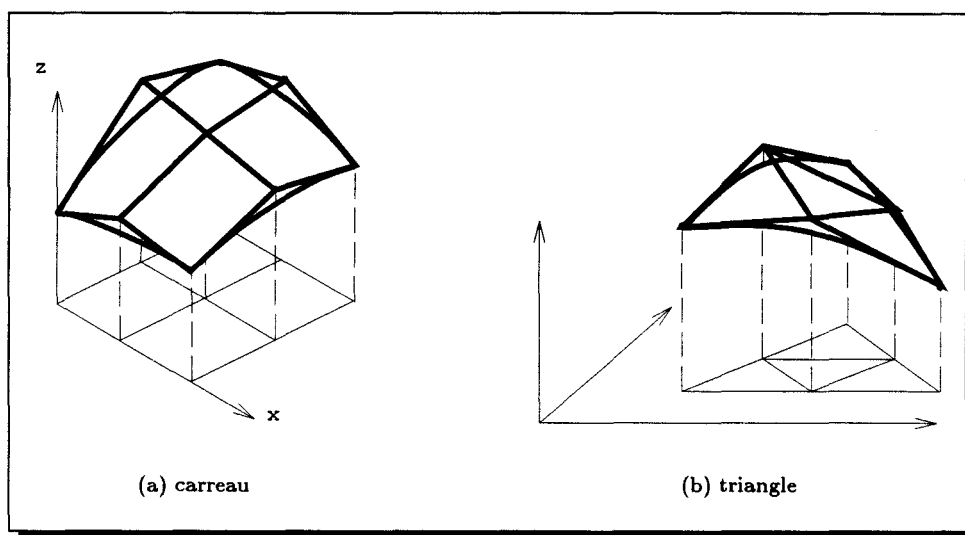


FIG. 2.12 - Surfaces de Bézier fonctionnelles

Bilan

Les courbes de Bézier sont maintenant à la base de toute la théorie des courbes et surfaces paramétriques, et leur formalisme et leurs principes sont utilisés dans beaucoup de domaines de la CAO.

Néanmoins, une limitation importante des courbes de Bézier est l'impossibilité de représenter les courbes algébriques. Par exemple, une courbe de Bézier ne peut être un arc de cercle, quel que soit son degré. Cette limitation existe également au niveau des surfaces : les quadriques ne peuvent être exprimées sous forme de Bézier³. Toutefois, nous voyons au prochain chapitre qu'il existe des représentations des quadriques définies à l'aide des polynômes de Bernstein et qui utilisent les notions de triangles de Bézier fonctionnels.

3. Une généralisation des courbes/surfaces de Bézier, dites *rationnelles*, peuvent représenter des coniques/quadriques. Il s'agit de la projection d'une courbe de \mathbb{R}^4 sur l'hyperplan $w = 1$ [Fiorot et al.92]

Chapitre 3

Les quadriques et leurs représentations

Bien que les quadriques soient des surfaces mathématiques bien connues et beaucoup étudiées, des développements récents sont apparus avec la C.A.O. En effet, l'étude de la conception d'objets composés de primitives quadriques a apporté un certain nombre de résultats concernant des aspects plus fonctionnels de ces surfaces comme le contrôle de la forme ou de la continuité.

Ces résultats récents sur les quadriques sont en fait des cas particuliers de résultats concernant les surfaces algébriques (§3.1.1). Par rapport aux surfaces paramétriques dont on a rapidement compris les propriétés pour la CAO, les surfaces algébriques sont restées jusque récemment moins connues et peu utilisées. Néanmoins elles connaissent actuellement un regain d'attention, puisqu'en offrant des propriétés et des méthodes très différentes par rapport aux surfaces paramétriques, elles sont plus adaptées pour résoudre un certain nombre de problèmes, comme l'interpolation de points, ou l'affichage.

C. Bajaj a été l'un des premiers à proposer les surfaces algébriques comme primitives de modélisation à part entière [Bajaj88]. Sederberg en 1985 [Sederberg85], a introduit la représentation de Bernstein-Bézier, permettant d'obtenir un contrôle se rapprochant des surfaces de Bézier et autorisant donc la réalisation d'algorithmes performants [Menon94, Moore et al.91, Bajaj et al.95a], ou de systèmes de modélisation complets [Bajaj95].

Après avoir rappelé la définition algébrique des quadriques, nous énumérons leurs propriétés intéressantes pour la modélisation géométrique. Nous présentons la représentation de Bernstein-Bézier, ainsi que celles de Lodha-Warren et Sederberg-Zundel. Ces représentations sont à la base des algorithmes de création de surfaces complexes, vus au prochain chapitre.

3.1 Représentation algébrique

La représentation algébrique est la représentation "traditionnelle" des quadriques. C'est de celle-ci que nous déduisons la plupart des propriétés fondamentales. C'est également la représentation utilisée par les accélérateurs graphiques présentés au chapitre 1.

3.1.1 Définition

Les quadriques sont les surfaces de l'espace \mathbb{R}^3 définies comme l'ensemble des points (x, y, z) tels que :

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz + 2Fzx + 2Gx + 2Hy + 2Iz + J = 0 \quad (3.1)$$

A, B, \dots, F sont dix coefficients qui définissent la forme de la surface. Les quadriques sont un sous-ensemble des surfaces *algébriques*, définies par l'équation $P(x, y, z) = 0$ où P est un polynôme, elles-mêmes sous-ensemble des surfaces *implicites*, définies par $f(x, y, z) = 0$ où f est une fonction quelconque.

Les surfaces implicites sont définies par une *condition* que doit remplir un point pour appartenir à la surface plutôt que par une formule constructive, comme une équation paramétrique. On ne retrouve donc pas toutes les propriétés intéressantes des surfaces de Bézier ou des B-splines ni d'algorithmes de construction comme l'algorithme de De Casteljau (chapitre 2), qui font des surfaces paramétriques les primitives les plus utilisées en modélisation.

Néanmoins, ces dernières années ont vu des développements qui ont montré que les surfaces algébriques sont une alternative valable aux surfaces paramétriques : d'une part parce qu'elles constituent une classe de surfaces plus large, et d'autre part parce qu'elles présentent d'autres propriétés intéressantes que l'on ne retrouve pas avec les surfaces paramétriques.

3.1.2 Ecriture matricielle

On écrira également l'équation 3.1 sous la forme :

$$(x \ y \ z \ 1) \begin{pmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & I \\ G & H & I & J \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = 0 \quad (3.2)$$

ou

$$XQX^t = 0 \quad (3.3)$$

Une quadrique est donc entièrement déterminée par la matrice Q de ses coefficients. Cette notation est avantageuse pour effectuer certains calculs comme les transformations : si la matrice T est une matrice de transformation, alors la transformée P' d'un point $P = (x, y, z, 1)^t$ est $P' = PT$, et la transformée d'une quadrique Q est :

$$Q' = T^{-1}QT^{-1t} \quad (3.4)$$

3.1.3 Propriétés

Les surfaces quadriques héritent des propriétés des surfaces algébriques et implicites. Nous énumérons les plus importantes pour la modélisation géométrique et la synthèse d'images.

- Les surfaces algébriques sont un sur-ensemble des surfaces paramétriques. Par exemple, les sphères, comme vu précédemment, ne peuvent être représentées par des surfaces de Bézier (exemple : il n'y a pas de représentation non rationnelle de la sphère).
- Il est facile de déterminer si un point donné (x, y, z) appartient à la surface (il suffit de vérifier si $f(x, y, z) = 0$), et sinon, à quel demi espace il appartient ($f(x, y, z) > 0$ ou $f(x, y, z) < 0$). On peut ainsi développer une notion d'intérieur et d'extérieur, à la base des méthodes par opérations booléennes (chapitre 1).
- Les surfaces algébriques permettent l'utilisation de surfaces de degré quelconque. Le degré d'une surface est considéré ici au sens de Bezout, défini comme le nombre maximal d'intersections de la surface avec une droite quelconque (en comptant correctement les points multiples). Le degré d'une surface algébrique est égal au degré de son équation. Par contre, le degré d'une surface paramétrique par produit tensoriel est $2nm$ où n et m sont les degrés des équations paramétriques en s et t respectivement. Un carreau de Bézier bicubique, par exemple, est de degré 18. Un triangle de Bézier représenté par une équation de degré n est de degré n^2 [Sederberg et al.85]. Alors qu'un carreau de Bézier ne peut pas être de degré impair ou qu'un triangle ne peut être de degré 2, 3, 5, etc., une surface algébrique peut être de degré quelconque¹.
- La classe des surfaces algébriques est fermée pour toutes les opérations classiques en modélisation. On le démontre de façon simple pour les transformations courantes (rotations, translations, changement d'échelle) : Soient une quadrique définie par la matrice Q et la transformation représentée par la matrice M , telle que pour un point P , la transformée de P est PM . Si un point P appartient à Q , on a $PQP^t = 0$ par définition. La transformée par M de P étant PM , les points de Q transformés sont tels que $(PM)Q(PM)^t = 0$ ou $P(MQM^t)P^t = 0$, qui est aussi l'équation d'une quadrique. Donc la transformée d'une quadrique est également une quadrique². D'autres opérations fermées pour les surfaces algébriques sont le calcul d'*offsets* (surface située à une certaine distance d'une surface donnée) et de *blending* (surface effectuant une transition continue entre deux surfaces données). Par exemple, le blending entre deux surfaces algébriques (c'est-à-dire la surface qui adoucit le contact entre deux autres surfaces) est algébrique. Cette propriété, qui n'est pas vraie pour les surfaces paramétriques, est intéressante lors du choix des primitives d'un modèleur : si la classe de surfaces est fermée pour les opérations courantes, un seul type de surfaces est à implémenter.
- Certains algorithmes de rendu réaliste comme le lancer de rayons se font simplement. En effet, calculer l'intersection d'une surface algébrique et d'un rayon s'effectue simplement : si le rayon est exprimé sous forme paramétrique ($x = R_x(t), y = R_y(t), z = R_z(t)$) il suffit de remplacer dans l'équation algébrique x par $R_x(t)$, y par $R_y(t)$ et z par $R_z(t)$ pour obtenir une unique équation de degré n (le degré de la surface) en t . Les n solutions possibles donnent les n intersections possibles de la surface et du rayon. Le même processus avec une surface paramétrique nécessite le calcul de l'équation implicite de la surface, ce qui est en général complexe et produit des équations de degré élevé.

1. En fait, certaines surfaces de Bézier dites *dégénérées* peuvent être de degré inférieur, mais dans certains cas particuliers de position de leurs points de contrôle (trivialement, lorsque tous les points de contrôle d'une surface de degré quelconque sont coplanaires, la surface est un plan et son degré 1).

2. On a démontré ainsi l'invariance par rotation, utilisée au chapitre 1 pour le rendu matériel.

Néanmoins les surfaces algébriques ne sont pas une panacée, puisqu'elles présentent quelques inconvénients pour la modélisation aussi bien que pour le rendu :

- Il n'existe pas de modèle de représentation des surfaces qui permet un contrôle intuitif, du style polygone de contrôle, bien que des progrès récents aient été accomplis dans ce sens (§3.2).
- Les quadriques ne disposent pas de coordonnées de surface, ce qui complique fortement le placage de textures sur la surface, ou la facettisation.
- Lors de la construction de surfaces algébriques par morceaux, les ordres de continuité entre deux surfaces se contrôlent difficilement (des progrès récents ont également été accomplis dans ce domaine, voir §3.1.4).
- Les surfaces algébriques sont sensibles aux problèmes de précision arithmétique [Goldman83]. Par exemple, la détermination du type de quadrique (ellipsoïde, paraboloid...) se fait par la détermination de la nullité de certains invariants. Or, en arithmétique flottante avec précision finie, les valeurs sont rarement exactement nulles. Ce problème touche d'autres opérations courantes comme le test d'égalité de deux quadriques.

Bien que, parmi les surfaces algébriques, le degré des quadriques soit particulièrement faible, celles-ci ont une variété de forme suffisamment large pour beaucoup d'applications de modélisation. Les figures 3.1, 3.2 et 3.3 illustrent les neuf types de quadriques non dégénérées³.

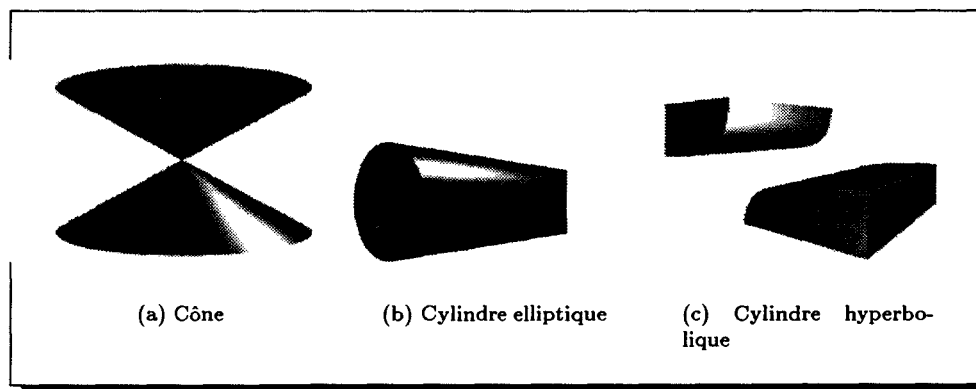


FIG. 3.1 - Quadriques 1

3.1.4 Continuité

Le contrôle de la continuité lors de la construction de surfaces quadriques par morceaux est un problème important, comme c'est le cas de façon générale en modélisation géométrique. Intuitivement, on dit d'un raccord entre deux courbes ou deux surfaces qu'il est G^0 si les surfaces forment une unique courbe ou surface sans trous. Si celles-ci sont G^0 sans former de saillie (les plans tangents à la courbe/surface de part et d'autre sont confondus), on a la continuité G^1 . Ce concept se définit mathématiquement et s'étend aux ordres supérieurs.

Pour l'expression de la continuité, les surfaces algébriques ne disposent pas d'un formalisme aussi simple que les surfaces paramétriques. Néanmoins, certains résultats existent que nous allons

3. Les quadriques ont été coupées par des plans, et sont représentées sous forme volumique (§1.4.2).

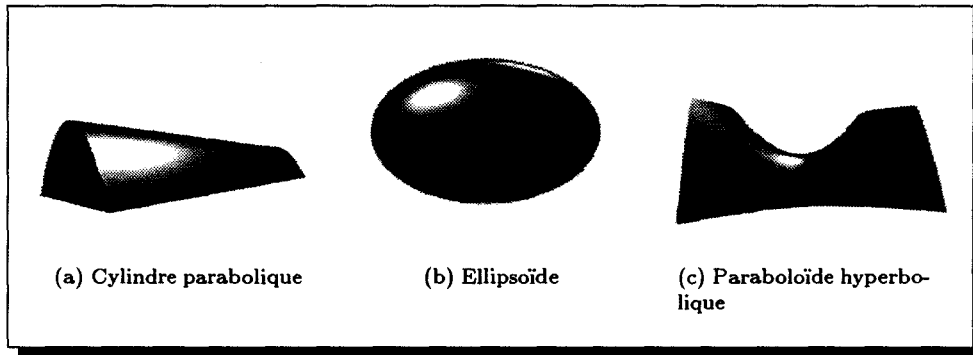


FIG. 3.2 - *Quadriques 2*

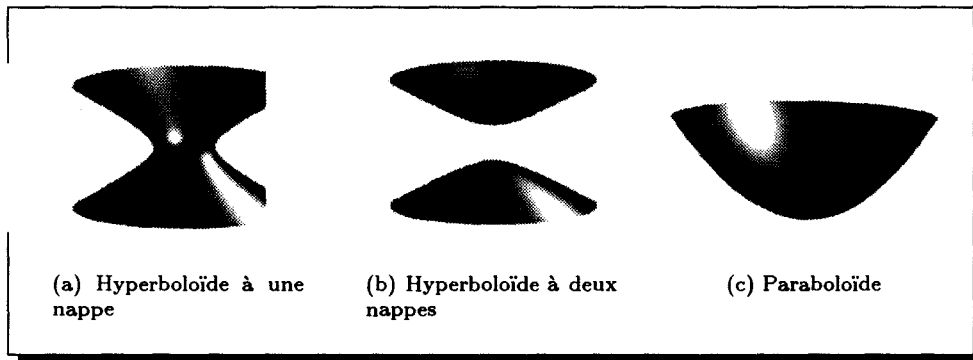


FIG. 3.3 - *Quadriques 3*

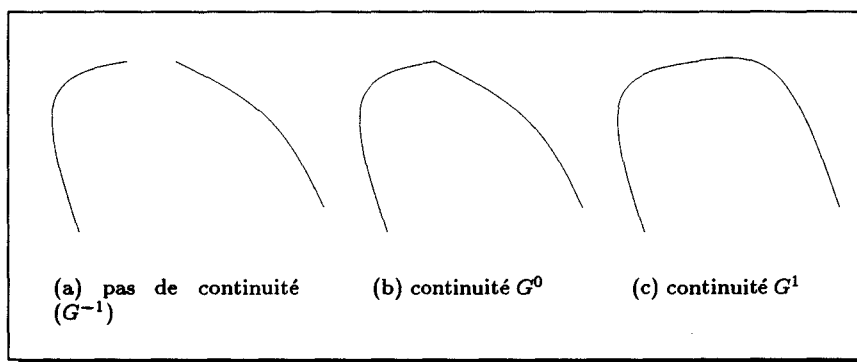


FIG. 3.4 - *Continuité entre deux courbes coniques*

développer dans le cas des quadriques. Le formalisme que nous adoptons est emprunté à Garrity et Warren [Garrity et al.91].

Voici trois définitions qui définissent de façon formelle la continuité géométrique G^n entre deux courbes ou surfaces :

Définition 1 *Un raccord entre deux courbes est G^n au point P si celles-ci peuvent être représentées par deux fonctions qui sont C^n en P (c'est-à-dire dont toutes les dérivées s'accordent en P jusqu'à l'ordre n).*

Cette première définition s'applique aussi bien aux surfaces paramétriques (en dérivant par rapport aux paramètres) qu'aux surfaces implicites (en dérivant par rapport aux coordonnées). Elle s'étend aux surfaces de la façon suivante :

Définition 2 *Un raccord entre deux surfaces F et G est G^n au point P si pour tout plan L passant par P , les courbes $F \cap L$ et $G \cap L$ sont G^n en P .*

Définition 3 *Un raccord entre deux surfaces est G^n le long d'une courbe C si elles se raccordent avec continuité G^n en tous les points de C .*

Ces définitions simples sont cependant difficilement applicables telles quelles, à cause du "peuvent être" de la première définition, qui implique que, par exemple, deux courbes peuvent être G^n en P alors qu'elles sont définies par deux fonctions dont les dérivées ne s'accordent pas jusqu'à la n ème dérivée. On sait simplement qu'elles sont G^n si il existe des fonctions qui s'accordent.

Cette définition est donc en général simplifiée en remplaçant le "peuvent être" par "sont". On parle alors de continuité C^n . Par rapport à la première définition, c'est une condition suffisante mais non nécessaire ($C^n \Rightarrow G^n$). Cependant elle est adoptée dans la plupart des cas pour des raisons de simplicité de calcul.

Pour les surfaces implicites, les quadriques en particulier, Warren [Garrity et al.91] définit la *rescaling continuity*, équivalente à la continuité G^n :

Définition 4 *Deux surfaces algébriques d'équations $f(x, y, z) = 0$ et $g(x, y, z) = 0$ se raccordent avec continuité G^n au point P ou le long d'une courbe algébrique C si et seulement si il existe deux polynômes $a(x, y, z)$ et $b(x, y, z)$ non nuls en P ou sur C tels que toutes les dérivées de $af - bg$ jusqu'à l'ordre n s'annulent en P ou sur C .*

Cette définition, a priori pas vraiment plus simple que la première, entraîne un résultat plus utilisable :

Théorème 1 ([Garrity et al.91]) *Deux surfaces F et G , respectivement d'équations $f(x, y, z) = 0$ et $g(x, y, z) = 0$ se raccordent avec continuité G^n le long d'une courbe algébrique, définie comme l'intersection de F et d'une troisième surface H d'équation $h(x, y, z) = 0$ si et seulement si il existe deux polynômes $a(x, y, z)$ et $b(x, y, z)$ non nuls sur $F \cap H$ tels que :*

$$g(x, y, z) = a(x, y, z)f(x, y, z) + b(x, y, z)h^{n+1}(x, y, z) \quad (3.5)$$

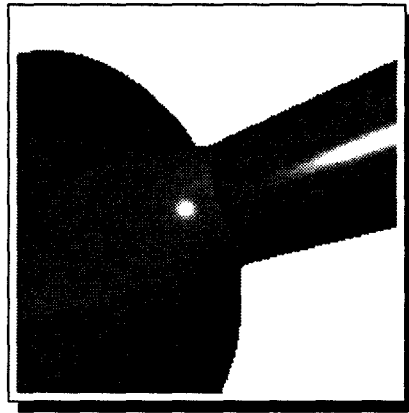


FIG. 3.5 - Raccord G^2 entre une cubique et un cylindre

Ce théorème peut être utilisé de manière constructive : par exemple, toute surface g ayant un raccord G^2 avec le cylindre de rayon 1 et d'axe x d'équation $f(x, y, z) = y^2 + z^2 - 1 = 0$ sur le plan $B(x, y, z) = x - 2$ aura comme équation $g(x, y, z) = a(y^2 + z^2 - 1) + b(x - 2)^3$ où a et b sont deux polynômes quelconques.

Dans le cas des quadriques, le degré étant borné, l'ordre de continuité géométrique qu'il est possible d'atteindre est au plus égal à un :

Proposition: Deux surfaces quadriques se raccordant le long d'une courbe plane avec continuité G^n où $n > 1$ sont identiques.

Preuve:

Soit F une quadrique d'équation $f(x, y, z) = 0$ et H un plan d'équation $h(x, y, z) = 0$ coupant F . Une autre quadrique G d'équation $g(x, y, z) = 0$ est reliée F avec continuité G^n le long de la courbe $F \cap H$ si :

$$g = af + bh^{n+1} \tag{3.6}$$

Comparons les degrés des différents termes. G étant une quadrique, g est de degré 2. L'autre partie de l'égalité doit donc également être de degré 2. f est de degré 2 donc a est de degré inférieur ou égal à 1, et h étant de degré 1 (puisque H est un plan), le terme h^{n+1} est donc de degré $n + 1$. Supposons que $n = 1$, l'équation est de degré 2 à condition que le terme b soit une constante. Mais si $n = 2$, le terme h^{n+1} est alors de degré 3, et par conséquent il faut que b soit le polynôme nul pour que l'égalité reste de degré 2. On a alors :

$$g = af$$

Le terme a est donc a fortiori une constante, et G et F sont identiques puisqu'elles sont définies par deux polynômes proportionnels (les coordonnées d'un point qui vérifient $f(x, y, z) = 0$ vérifient également $kf(x, y, z) = 0$).□

Autre conséquence du Théorème de Warren, deux quadriques différentes ne peuvent se raccorder le long d'une courbe gauche qu'avec continuité G^0 .

La conclusion de ces résultats sur la construction de surfaces quadriques par morceaux est que l'on cherchera donc au mieux la continuité G^1 , i.e. la continuité des plans tangents.

3.2 Représentation de Bernstein-Bézier

Bien que nous ayons introduit les quadriques avec leur définition algébrique, il existe d'autres définitions possibles de la même classe de surfaces. Même si les propriétés géométriques des quadriques (comme la continuité) restent inchangées, les autres représentations peuvent s'avérer au moins aussi intéressantes du point de vue informatique, puisqu'elles permettront de concevoir des méthodes de construction de surfaces complexes, ou de faciliter le calcul du rendu. Par contre elles introduisent le problème de la *conversion* : puisque l'afficheur prend en entrée les coefficients de la forme algébrique, il est toujours nécessaire de convertir le format utilisé. Le coût de cette opération peut s'avérer important pour certains formats.

La représentation de Bernstein-Bézier est actuellement la solution la plus prometteuse pour la construction de surfaces complexes. Introduite par T. Sederberg [Sederberg84], elle s'applique à toute courbe algébrique, surface ou variété de dimension finie. Nous l'introduirons d'abord pour les courbes quadratiques du plan. Et nous verrons que ce cas se généralise facilement aux dimensions et degrés supérieurs.

3.2.1 Représentation des coniques

Soit un repère barycentrique V du plan (cf. chapitre 2). Tout point P est repéré par ses coordonnées barycentriques $(\lambda_0, \lambda_1, \lambda_2)$. On considère l'ensemble des points dont les coordonnées vérifient l'équation :

$$\sum_{i+j+k=2} \frac{2}{i!j!k!} C_{i,j,k} \lambda_0^i \lambda_1^j \lambda_2^k = C_{2,0,0} \lambda_0^2 + 2C_{1,1,0} \lambda_0 \lambda_1 + 2C_{1,0,1} \lambda_0 \lambda_2 + C_{0,2,0} \lambda_1^2 + 2C_{0,1,1} \lambda_1 \lambda_2 + C_{0,0,2} \lambda_2^2 = 0 \quad (3.7)$$

Cette courbe algébrique, dont la forme est définie par les coefficients $C_{i,j}$ est une conique. En effet, l'équation 3.7 peut s'écrire sous forme matricielle :

$$(\lambda_0 \ \lambda_1 \ \lambda_2) \begin{pmatrix} C_{2,0,0} & C_{1,1,0} & C_{1,0,1} \\ C_{1,1,0} & C_{0,2,0} & C_{0,1,1} \\ C_{1,0,1} & C_{0,1,1} & C_{0,0,2} \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = 0 \quad \text{ou} \quad \Lambda C \Lambda^t = 0 \quad (3.8)$$

Or, la définition des coordonnées barycentriques (équation 2.6, page 28) s'écrit de façon matricielle

$$(x \ y \ 1) = (\lambda_0 \ \lambda_1 \ \lambda_2) \begin{pmatrix} V_0^x & V_0^y & 1 \\ V_1^x & V_1^y & 1 \\ V_2^x & V_2^y & 1 \end{pmatrix} \quad \text{ou} \quad X = \Lambda V \quad \text{ou} \quad \Lambda = XV^{-1} \quad (3.9)$$

où V est la matrice des coordonnées cartésiennes des trois points de la base barycentrique. En remplaçant $\Lambda = XV^{-1}$ dans l'équation 3.8, on obtient

$$X(V^{-1}CV^{-1t})X^t = 0 \quad \text{ou} \quad XM X^t = 0 \quad (3.10)$$

M est une matrice constante (indépendante de X), donc l'équation 3.10 est l'équation d'une conique. Conclusion : une conique est définie dans une base barycentrique ou dans une base cartésienne du plan par une matrice de coefficients. L'équation 3.10 donne une formule de conversion barycentrique \rightleftharpoons cartésien. Ce résultat se généralise pour toutes les dimensions et à tous les degrés.

La définition des coniques par coordonnées barycentriques est la représentation de Bernstein-Bézier de celles-ci. Mais pour démontrer les propriétés supplémentaires par rapport à la représentation cartésienne, il est nécessaire d'aller plus loin dans la description et de faire le rapprochement avec les surfaces de Bézier.

En effet, si l'on considère l'équation 3.7 sans le terme $= 0$, on a une fonction de $\lambda_0, \lambda_1, \lambda_2$ dont le graphe est une surface qui peut être représentée sous forme de surface de Bézier fonctionnelle (§ 2.3, page 29). En identifiant les équations 3.7 et 2.14, on obtient :

$$\sum_{i+j+k=2} \frac{2}{i!j!k!} C_{i,j,k} \lambda_0^i \lambda_1^j \lambda_2^k = \sum_{i+j+k=n} B_{i,j,k}^n(\Lambda) P_{i,j,k} \quad (3.11)$$

d'où les coordonnées des points de contrôle (rappelons que les trois premières sont barycentriques sur le plan et que la quatrième est cartésienne, figure 3.6) :

$$P_{i,j,k} = (i/2, j/2, k/2, C_{i,j,k}) \quad (3.12)$$

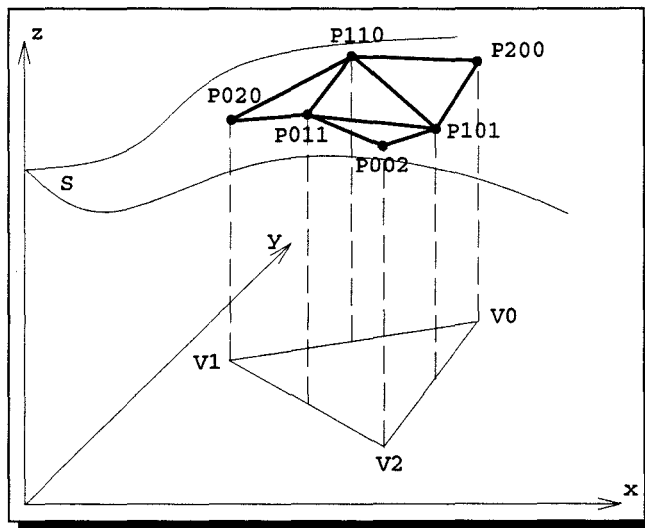


FIG. 3.6 - Surface de Bézier quadratique fonctionnelle

Revenons à la conique. Le terme $= 0$ de l'équation 3.10 signifie simplement que l'on considère les points de la surface fonctionnelle tels que $S(\lambda_0, \lambda_1, \lambda_2) = 0$, c'est-à-dire l'intersection de la surface fonctionnelle S avec le plan $z = 0$ (figure 3.7).

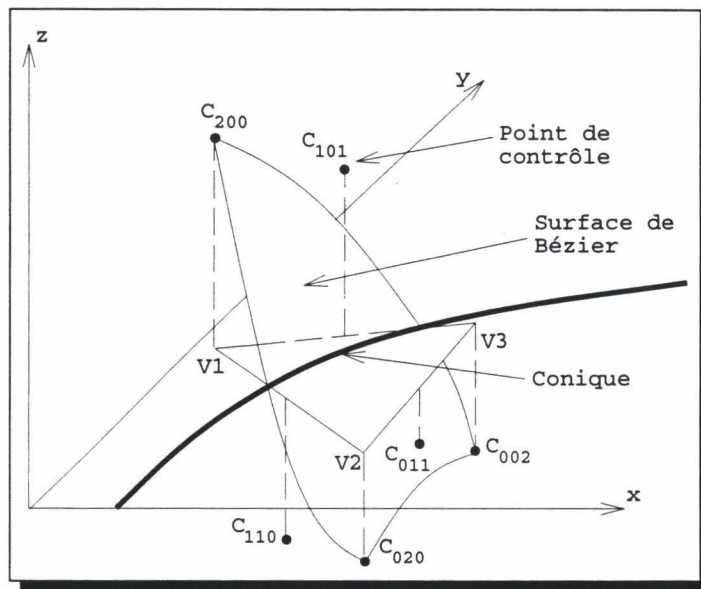


FIG. 3.7 - Conique définie par une surface de Bézier fonctionnelle.

On voit donc qu'un triangle de Bézier fonctionnel de l'espace définit une conique unique du plan. On dira que la surface de Bézier est *au-dessus* de la courbe. Une signification géométrique des coefficients de l'équation apparaît alors : ils représentent l'*élévation* de chaque point de contrôle de S (figure 3.7).

Comme on travaille dans le plan, chaque point de contrôle est projeté et est muni d'un coefficient qui représente son élévation dans l'espace au-dessus. Le plus souvent, la conique est limitée au segment inclus dans le triangle (cela permettra de former des courbes coniques par morceaux, en juxtaposant plusieurs triangles). Nous avons ainsi défini un pseudo-polygone de contrôle pour tout arc de conique (figure 3.8).

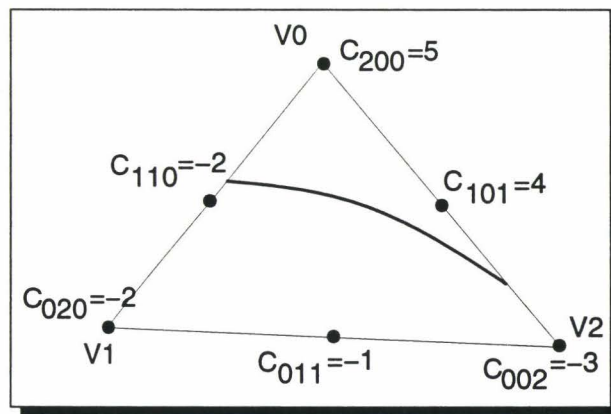


FIG. 3.8 - Segment de conique et son polygone de contrôle

3.2.2 Propriétés

L'analogie avec les triangles de Bézier permet de démontrer les propriétés suivantes :

- Symétrie : la courbe est symétrique par rapport aux coordonnées barycentriques (d'après la propriété de symétrie des triangles de Bézier).
- Interpolation : la courbe interpole le sommet V_i si et seulement si C_{e^i} , le coefficient qui lui est affecté, est nul⁴ (d'après la propriété d'interpolation des courbes de Bézier).
- Tangence : la courbe est tangente au segment $V_i V_j$ en V_i si et seulement si C_{e^i} et $C_{e^i+e^j}$ (le coefficient situé entre V_i et V_j) sont nuls (d'après la propriété d'interpolation des courbes de Bézier).
- Contrôle : chaque coefficient influence principalement la partie de la courbe la plus proche de lui (d'après la propriété de contrôle des surfaces de Bézier).
- Si les coefficients $C_{i,j,k}$ sont tous strictement positifs ou tous strictement négatifs, alors il n'y a pas de segment de quadrique dans le triangle. En effet, l'enveloppe convexe du polygone de contrôle du triangle de Bézier au-dessus est complètement au-dessus ou au-dessous du plan. Donc, d'après la propriété d'enveloppe convexe des triangles de Bézier, le triangle est également soit complètement au dessus ou en dessous du plan. Par conséquent la surface n'intersecte pas le plan à l'intérieur du triangle.

3.2.3 Généralisation

Degré

La représentation de Bernstein-Bézier n'est pas uniquement définie pour les coniques. Les courbes implicites de degré supérieur sont simplement définies par les surfaces de Bézier de degré correspondant. Le polygone de contrôle est toujours un triangle, mais le nombre de points de contrôle augmente avec le degré : il y a $(n+2)(n+1)/2$ coefficients (i.e. points de contrôle) pour une courbe de degré n (figure 3.9).

Dimension

Les polynômes de Bernstein à deux variables s'étendent simplement à d variables :

$$B_{i_0, i_1, \dots, i_d}^n(\lambda_0, \lambda_1, \dots, \lambda_d) = \frac{n!}{i_0! i_1! \dots i_d!} \lambda_0^{i_0} \lambda_1^{i_1} \dots \lambda_d^{i_d} \quad (3.13)$$

Par conséquent, les triangles de Bézier fonctionnels sont généralisables à des hyper-surfaces dans un espace de dimension $d+1$, et à chacune de ces surfaces peut être associée une "hyper-courbe" de dimension d . On appelle d -simplex les triangles de dimension⁵ d . Nous avons vu la définition des courbes algébriques à l'aide de 2-simplex, mais cette définition se généralise aux surfaces, représentées à l'aide de 3-simplex de Bézier [Sederberg85] : la base barycentrique est un

4. e^i représente le vecteur dont l'élément de rang i est 1, les autres étant nuls. Par exemple, $C_{e^1} = C_{0,1,0}$.

5. Définis sur un espace paramétriques de $d+1$ variables barycentriques, 2 pour une courbe, 3 pour une surface, etc.

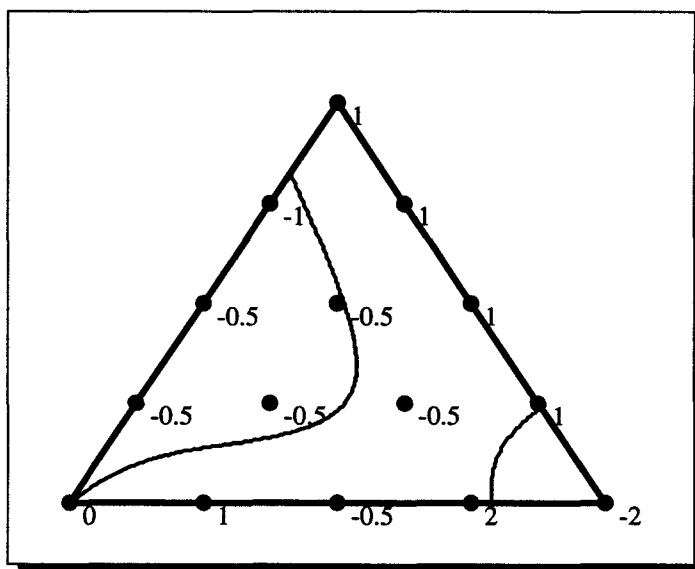


FIG. 3.9 - Courbe quartique et son triangle de contrôle (2 arc de la courbe sont visibles dans le triangle)

tétraèdre, le volume de Bézier au-dessus de la surface algébrique est donc un 3-simplex. L'équation implicite 3.7 se généralise :

$$\sum_{i+j+k+l=2} \frac{2}{i!j!k!l!} C_{i,j,k,l} \lambda_0^i \lambda_1^j \lambda_2^k \lambda_3^l = C_{2,0,0,0} \lambda_0^2 + 2C_{1,1,0,0} \lambda_0 \lambda_1 + 2C_{1,0,1,0} \lambda_0 \lambda_2 + \dots = 0 \quad (3.14)$$

Cette équation décrit toutes les surfaces quadriques. Les coefficients de contrôle se situent sur le tétraèdre de la façon suivante : le point $P_{i,j,k,l}$ se situe aux coordonnées barycentriques $(i/2, j/2, k/2, l/2)$.

Toutes les propriétés de la représentation de Bernstein-Bézier pour les courbes se généralisent directement, puisque les propriétés des triangles de Bézier se généralisent aux n -simplex : symétrie par rapport aux coordonnées, interpolation (la courbe passe par V_i si $C_{e^i} = 0$), tangence (la courbe est tangente à $V_i V_j$ en V_i si $C_{e^i} = C_{e^i + e^j} = 0$).

Pour les quadriques, il faut donc $d = 3$ et $n = 2$. Le triangle de contrôle est un tétraèdre sur lequel on trouve les 10 coefficients de l'équation implicite en coordonnées barycentriques. A l'instar des coniques, on limite l'étendue des surfaces quadriques à la partie se trouvant à l'intérieur du tétraèdre. Cela définit la notion de *patch quadrique*.

La formule de conversion de coniques de la forme de Bernstein-Bézier vers la forme cartésienne et inversement 3.10 est également généralisable à une dimension quelconque et en particulier aux quadriques (attention, la forme matricielle ne s'applique qu'aux courbes ou surfaces de degré 2) :

$$M = V^{-1} C V^{-t} \quad (3.15)$$

où M et C sont les matrices de la quadrique, respectivement sous forme canonique et de Bernstein-Bézier, et V est la matrice des coordonnées cartésiennes des quatre points de la base barycentrique. La conversion dans l'autre sens s'obtient en retournant l'équation :

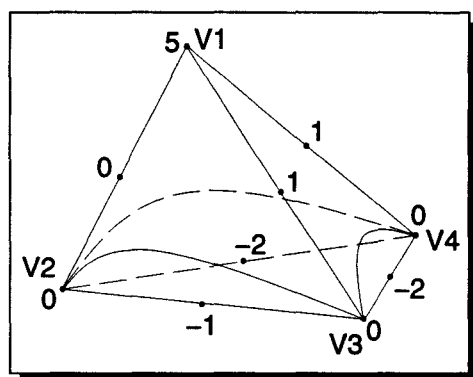


FIG. 3.10 - Patch quadrique et son tétraèdre de contrôle

$$C = VMV^t \quad (3.16)$$

Par exemple, la sphère de rayon 1 centrée à l'origine dans un repère cartésien a pour équation (et matrice):

$$x^2 + y^2 + z^2 - 1 = 0 \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (3.17)$$

Si l'on définit un repère barycentrique aux points $V_1(0, 0, 0)$, $V_2(1, 0, 0)$, $V_3(0, 1, 0)$, $V_4(0, 0, 1)$. La matrice de la sphère sous forme de Bernstein-Bézier est :

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & 2 \end{pmatrix} \quad (3.18)$$

3.2.4 Raccordement

La condition de continuité 3.5 ne s'applique plus quand les quadriques sont sous forme de Bernstein-Bézier. Mais l'utilisation du volume de Bézier au-dessus va nous donner une condition simple. En effet, pour deux patchs quadriques ayant une face de leur tétraèdre de contrôle en commun, la continuité G^1 des deux volumes au-dessus est *suffisante* pour avoir la continuité des deux quadriques [Sederberg84].

Soient donc deux triangles de Bézier ayant un bord de leur polygone de contrôle en commun (figure 3.11).

Théorème 2 (Sederberg [Sederberg et al.85]) Deux triangles de Bézier se raccordent avec continuité G^1 le long d'un bord commun si et seulement si il existe deux réels a et b tels que :

$$\begin{aligned} \overrightarrow{P_5 P_4} &= a \overrightarrow{P_5 P_2} + b \overrightarrow{P_5 P_7} \\ \overrightarrow{P_6 P_5} &= a \overrightarrow{P_6 P_3} + b \overrightarrow{P_6 P_8} \end{aligned} \quad (3.19)$$

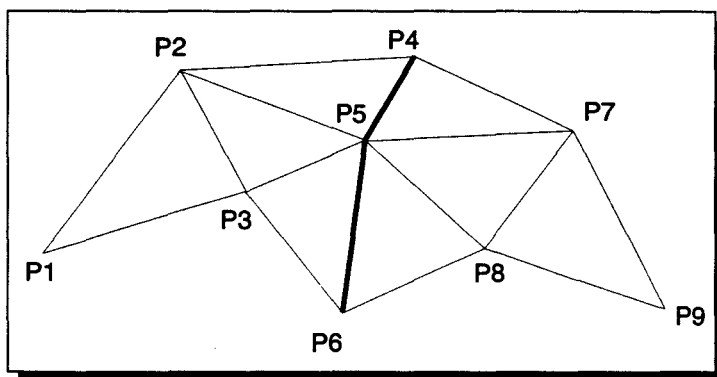


FIG. 3.11 - Triangles de Bézier adjacents

Dans le cas de triangles fonctionnels, cette condition peut être simplifiée:

Proposition Deux triangles de Bézier fonctionnels sont G^1 si et seulement si les points P_2, P_4, P_5 et P_7 et les points P_3, P_5, P_6 et P_8 sont coplanaires.

Preuve:

- équation 3.19 \Rightarrow coplanarité: immédiat puisque la somme de deux vecteurs appartient au plan contenant ces vecteurs.
- coplanarité \Rightarrow équation 3.19: supposons que les points P_2, P_4, P_5 et P_7 soient coplanaires. Alors il existe deux réels a et b tels que: $\overrightarrow{P_5 P_4} = a \overrightarrow{P_5 P_2} + b \overrightarrow{P_5 P_7}$. De même pour les points P_3, P_5, P_6 et P_8 : il existe deux réels a' et b' tels que: $\overrightarrow{P_6 P_5} = a' \overrightarrow{P_6 P_3} + b' \overrightarrow{P_6 P_8}$. Montrons que $a = a'$ et $b = b'$. En projetant les vecteurs sur le plan $z = 0$, on obtient (figure 3.12):

$$\begin{aligned} \overrightarrow{P_5 P_4}^p &= a \overrightarrow{P_5 P_2}^p + b \overrightarrow{P_5 P_7}^p \\ \overrightarrow{P_6 P_5}^p &= a' \overrightarrow{P_6 P_3}^p + b' \overrightarrow{P_6 P_8}^p \end{aligned} \quad (3.20)$$

Or, d'après la position des points de contrôle dans un triangle fonctionnel, on a:

$$\begin{aligned} \overrightarrow{P_6 P_3}^p &= \overrightarrow{P_5 P_2}^p \\ \overrightarrow{P_6 P_8}^p &= \overrightarrow{P_5 P_7}^p \\ \overrightarrow{P_6 P_5}^p &= \overrightarrow{P_5 P_4}^p \end{aligned} \quad (3.21)$$

par conséquent, $a = a'$ et $b = b'$. \square

Pour les quadriques, la condition se généralise: si l'on considère deux patches quadriques ayant une face commune (figure 3.13), la continuité G^1 est obtenue si les points $P_1, P_4, P_5, P_6, P_{10}$ sont appartenent au même hyperplan de \mathbb{R}^4 , ainsi que les points $P_3, P_6, P_7, P_8, P_{11}$ et $P_2, P_5, P_8, P_9, P_{10}$. C'est-à-dire:

$$\begin{cases} a \overrightarrow{P_7 P_6} + b \overrightarrow{P_7 P_3} + c \overrightarrow{P_7 P_8} + d \overrightarrow{P_7 P_{11}} = 0 \\ a' \overrightarrow{P_8 P_5} + b' \overrightarrow{P_8 P_2} + c' \overrightarrow{P_8 P_{12}} + d' \overrightarrow{P_8 P_9} = 0 \\ a'' \overrightarrow{P_4 P_1} + b'' \overrightarrow{P_4 P_5} + c'' \overrightarrow{P_4 P_6} + d'' \overrightarrow{P_4 P_{10}} = 0 \end{cases} \quad (3.22)$$

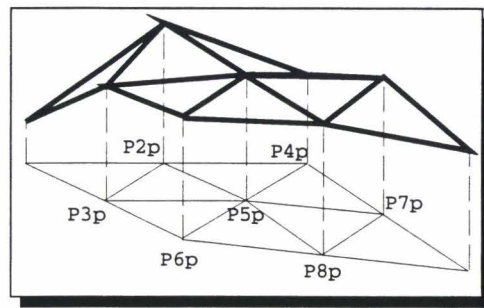


FIG. 3.12 - Points de contrôle projetés

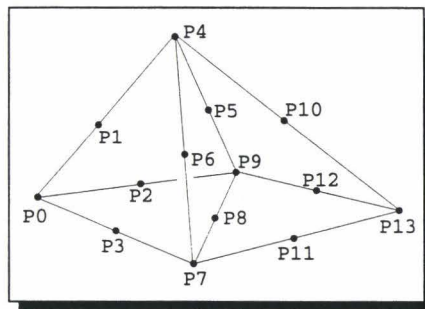
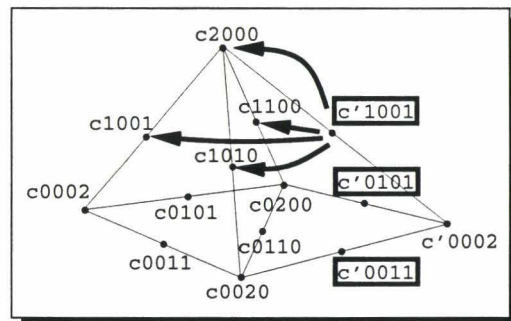


FIG. 3.13 - Points de contrôle de deux patches quadriques adjacents

Cette condition de continuité est exprimable d'une façon parfois plus pratique [Dahmen89] : soit un patch quadrique fixé de tétraèdre $V(V1, V2, V3, V4)$. Un patch adjacent $V'(V1, V2, V3, V5)$ (figure 3.2.4) se raccorde avec continuité G^1 si :

$$\begin{aligned}
 c'_{1001} &= \lambda_0(V5, V)c_{2000} + \lambda_1(V5, V)c_{1100} + \\
 &\quad \lambda_2(V5, V)c_{1010} + \lambda_3(V5, V)c_{1001} \\
 c'_{0101} &= \lambda_0(V5, V)c_{1100} + \lambda_1(V5, V)c_{0200} + \\
 &\quad \lambda_2(V5, V)c_{0110} + \lambda_3(V5, V)c_{0101} \\
 c'_{0011} &= \lambda_0(V5, V)c_{1010} + \lambda_1(V5, V)c_{0110} + \\
 &\quad \lambda_2(V5, V)c_{0020} + \lambda_3(V5, V)c_{0011}
 \end{aligned}
 \tag{3.23}$$



où $(\lambda_0(V5, V), \lambda_1(V5, V), \lambda_2(V5, V), \lambda_3(V5, V))$ sont les coordonnées du point $V5$ dans la base définie par le tétraèdre V .

Conclusion : un patch de Bernstein-Bézier raccordé à un autre patch fixe avec continuité G^0 aura six de ses dix coefficients fixés par les conditions de raccordement. Si le raccordement est G^1 , neuf de ses dix coefficients sont fixés. Cela confirme le fait que la condition de continuité G^1 sur les quadriques est très forte sur leurs degrés de liberté.

3.3 Autres représentations

La représentation de Bernstein-Bézier des quadriques présente de nombreux avantages, en particulier pour l'utilisation que nous faisons des quadriques aux chapitres suivants. Toutefois, d'autres représentations existent, le plus souvent utilisées pour une opération précise comme le rendu ou un type de modélisation particulier comme le CSG.

3.3.1 Le modèle géométrique

Le modèle géométrique a été défini par R. Goldman dans le but d'éviter les problèmes des équations algébriques des quadriques, au sein d'un modèleur CSG ou B-rep [Goldman83]. En particulier pour les problèmes de précision arithmétique, déjà évoqués précédemment : le modèle algébrique ne permet pas de déterminer exactement le type d'une quadrique ou l'égalité de deux quadriques. De plus, certaines grandeurs géométriques sont stockées de façon redondante : supposons qu'un cylindre soit fermé par une demi-sphère. Les rayons du cylindre et de la sphère sont égaux, mais stockés séparément. Il suffit de quelques transformations sur le modèle pour que les deux valeurs commencent à différer, et ainsi de suite.

Pour éviter ces problèmes, le modèle géométrique représente chaque quadrique par :

- Un point de \mathbb{R}^3 qui fixe la position de la surface.
- Deux vecteurs de \mathbb{R}^3 qui fixent l'orientation de la surface.
- Trois coefficients pour la taille de la surface en x, y et z .
- Quatre bits pour le type de la quadrique (9 possibilités, cf. page 36).

Goldman déduit une taille de stockage de 12,5 réels, légèrement supérieure aux 10 réels du modèle algébrique. L'avantage principal de cette représentation est toutefois dans le contenu géométrique des informations stockées : la forme de la surface (i.e. le type) est immédiatement connue puisqu'elle est stockée directement, alors qu'un calcul non simple et très sensible aux erreurs numériques (§ 3.1.3) est nécessaire avec le modèle algébrique. En outre, le fait de stocker la taille de la surface permet de faire du partage de données. Par exemple si deux cylindres ont le même rayon, les deux champs indiquant le rayon (taille) peuvent pointer sur la même valeur. Ainsi, la modification de l'un des deux se répercute directement sur l'autre, sans avoir à recalculer systématiquement l'équation algébrique.

Le problème de cette représentation, c'est d'abord qu'on perd certaines propriétés des quadriques comme le test intérieur/extérieur, qui n'est plus possible sans équation implicite. L'uniformité des algorithmes est également perdue : comme on n'a plus de représentation uniforme pour les quadriques (par exemple, la position ou la taille n'ont pas forcément le même sens selon que l'on parle d'un ellipsoïde ou d'un paraboloid hyperbolique), il est nécessaire de prévoir un cas par type de quadrique pour toutes les opérations comme l'intersection ou l'offset ou encore l'affichage.

Dans le cas qui nous intéresse, la construction de surfaces quadriques par morceaux, on n'aura en général pas besoin de savoir si chaque morceau de quadrique est un morceau de sphère ou

de parabolöide hyperbolique. Ce qui importe, ce sont surtout les propriétés d'interpolation et de raccordement. De même pour la taille ou l'orientation. Nous n'utilisons donc pas ce modèle qui est plus réservé à une modélisation "classique" où les primitives sont manipulées directement par l'utilisateur qui sait exactement la forme dont il a besoin.

3.3.2 Représentation paramétrique

Comme nous l'avons vu auparavant, les quadriques ne peuvent être représentées par des surfaces de Bézier, puisqu'elles ne possèdent pas de paramétrisation polynomiale non rationnelle. Toutefois, toutes les quadriques sont paramétrables avec des fonctions trigonométriques ou des polynômes rationnels.

Les paramétrisations trigonométriques des quadriques s'expriment en fonction du type de la quadrique. Par exemple, un ellipsoïde a pour équation canonique [Rogers et al.90]:

$$\begin{aligned} x &= a \cos \theta \sin \phi \\ y &= b \sin \theta \sin \phi \\ z &= c \cos \theta \end{aligned} \quad 0 \leq \phi \leq \pi, \quad 0 \leq \theta \leq 2\pi \quad (3.24)$$

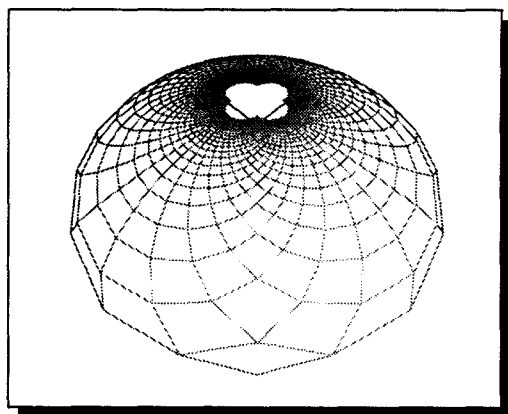
Connaissant le type de quadrique, et les transformations pour l'amener en position canonique (centrée sur l'origine et orientée selon les axes principaux), il est donc possible de trouver les équations trigonométriques de toute quadrique, en appliquant ces transformations à l'équation canonique.

Les formes polynomiales rationnelles des quadriques sont du type $(x, y, z) = \left(\frac{x(s,t)}{w(s,t)}, \frac{y(s,t)}{w(s,t)}, \frac{z(s,t)}{w(s,t)} \right)$. Il y a également une équation canonique par type de quadrique. Par exemple, un ellipsoïde a pour équation :

$$\begin{aligned} x &= 2as/(1 + s^2 + t^2) \\ y &= 2bt/(1 + s^2 + t^2) \\ z &= c(-1 + s^2 + t^2)/((1 + s^2 + t^2)) \end{aligned} \quad (s, t) \in \mathbb{R}^2 \quad (3.25)$$

Les représentations paramétriques offrent quelques propriétés avantageuses : on dispose d'une part d'une formule constructive qui permet de générer simplement un affichage de la surface facettisée et qui donne en particulier la possibilité de plaquer une texture sur la surface. De plus, il est possible de représenter une quadrique exprimée par une équation paramétrique rationnelle comme une surface de Bézier rationnelle. Nous avons déjà évoqué ce problème avec les surfaces de Bézier qui dégénèrent en quadriques (§ 3.1.3). Même si cela semble a priori intéressant de représenter une quadrique sous la forme d'une surface de Bézier, le contrôle de la surface est réduit puisque la position des points de contrôle perd toute liberté, et le contrôle de la continuité est problématique puisqu'on ne mesure plus qu'une continuité de type C^n , dépendante de la paramétrisation.

Les problèmes qui se posent à propos de ces représentations sont de plusieurs ordres. D'une part le contrôle des limites d'une surface doit se faire sur le contrôle des limites de l'espace des paramètres, et si le passage "espace des paramètres \rightarrow espace de modélisation" est immédiat puisqu'il est donné par la définition paramétrique, le passage inverse peut poser beaucoup de problèmes. D'autre part les paramétrisations (en particulier rationnelles) sont non uniformes : il arrive très souvent que les paramètres doivent parcourir une zone infinie pour couvrir une portion

FIG. 3.14 - *Sphère paramétrique*

finie de surface (figure 3.14). Il faut alors prévoir ce genre de phénomènes pour “reparamétriser”. Mais dans tous les cas le paramétrage reste irrégulier.

Le problème le plus important est que dans les deux cas il n’y a pas de représentation unifiée des équations paramétriques : en fonction du type de quadrique, une équation paramétrique est spécifiée. Il n’y a pas d’équation paramétrique globale qui, en faisant varier certains coefficients, décrirait *toutes* les quadriques et seulement elles. Par exemple, toutes les quadriques ont une équation rationnelle biquadratique. Mais toutes les équations rationnelles biquadratiques ne sont pas des quadriques, puisqu’en général elles sont de degré huit.

Beaucoup de recherches en modélisation géométrique visent à étudier les classes de surfaces qui ont à la fois une définition paramétrique et implicite. Pour les quadriques, le cas où une surface paramétrique est toujours une quadrique est réalisé par les surfaces bilinéaires (qui sont toujours des paraboloides hyperboliques). Mais il s’agit d’un cas vraiment trop limitatif. Les recherches se concentrent alors sur les surfaces paramétriques qui *dégénèrent* en quadriques, dans certaines configurations de leurs points de contrôle [Sederberg et al.85, Piegl85, Teller et al.91, Lodha et al.90]. Ces configurations restent encore trop complexes ou restent limitées à certaines quadriques (la sphère le plus souvent). Toutefois certains résultats intéressants existent, comme :

Théorème 3 (Lodha-Warren [Lodha et al.90]) *Une condition nécessaire et suffisante pour qu’un triangle de Bézier quadratique rationnel soit une quadrique est que les plans contenant les trois courbes frontières du triangle s’intersectent en un point de la surface.*

La représentation de Lodha-Warren des quadriques s’appuie sur ce théorème.

3.3.3 La représentation de Lodha-Warren

S. Lodha et J. Warren montrent qu’il existe une transformation perspective qui transforme toute quadrique en triangle de Bézier quadratique fonctionnel [Lodha et al.90]. À partir de ce résultat, ils proposent une procédure de création de surfaces quadriques à l’aide de points de contrôle :

1. Créer un tétraèdre $V(p_{002}, p_{020}, p_{200}, f)$. f est appelé *sommet focal*. Positionner ensuite six points sur V (figure 3.15).

2. Calculer la transformation perspective T qui envoie f à l'infini et transforme p_{002} , p_{020} et p_{200} respectivement en $(0, 0, 0)$, $(0, 1, 0)$ et $(1, 1, 0)$
3. Les points de contrôle transformés par T définissent une surface de Bézier fonctionnelle. La transformée de cette surface par T^{-1} est une quadrique.

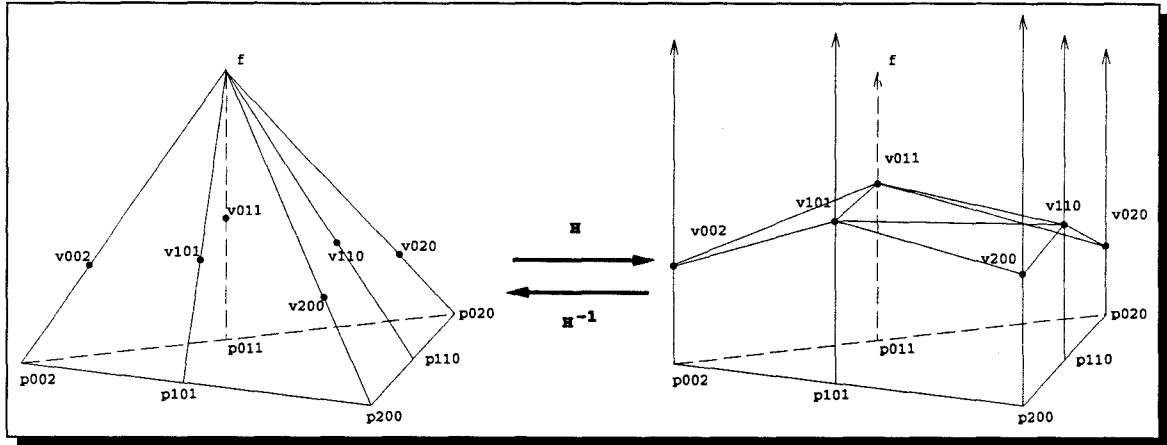


FIG. 3.15 - Méthode de Lodha-Warren de création de quadriques

Les quadriques créés par cette procédure sont donc entièrement définies par le tétraèdre et les six points de contrôle p_{ijk} .

Lodha et Warren prouvent ensuite que la quadrique définie est également un triangle de Bézier rationnel quadratique, dont les points de contrôle sont les points p_{ijk} donnés par l'utilisateur et dont les poids sont déterminés implicitement par la transformation T . En montrant que le sommet f appartient également à la quadrique, on prouve le théorème 3. Remarque importante: même s'il est possible de représenter toutes les quadriques, la configuration particulière du tétraèdre empêche la création de tous les triangles quadriques possibles.

Pour la création de surfaces complexes G^1 , il suffit maintenant d'exploiter les propriétés de continuité des surfaces de Bézier, i.e. la coplanarité des triangles de contrôle adjacents. Malheureusement, c'est là la principale limite de cette représentation : pour que deux triangles quadriques soient liés avec continuité G^1 , il faut qu'ils aient une face de leur tétraèdre et leur sommet focal en commun. Par conséquent tous les triangles d'une surface par morceaux doivent avoir le sommet focal commun. Les tétraèdres sont ainsi réduits à "faire le tour" du sommet. Les surfaces complexes qu'il est possible de modéliser sont alors réduites aux formes convexes, ou en étoile.

3.3.4 La représentation de Sederberg-Zundel

T. Sederberg et A. Zundel ont introduit cette représentation dans le but d'obtenir un rendu correct des surfaces algébriques (prenant en compte les singularités comme les lignes doubles) [Sederberg et al.89]. Ils utilisent dans ce but une variation de la base de Bézier à trois variables (utilisée dans la représentation de Bernstein-Bézier) appelée *Bernstein Pyramid Polynomial* (BPP). La BPP nécessite d'abord la définition d'un système de coordonnées (X, Y, Z) : une base dans ce système est définie par la donnée de 5 points de l'espace formant une pyramide (figure 3.16). La coordonnée Z des points du plan de base est 0, et pour le sommet $Z = 1$. Les coordonnées X et Y vont de $(0, 0)$ pour le coin bas-gauche à $(1, 1)$ pour le coin haut-bas. Tous les points d'une ligne

passant par le sommet ont les mêmes coordonnées X et Y . Le changement de base de ou vers un système de coordonnées cartésien se fait à l'aide de formules linéaires simples.

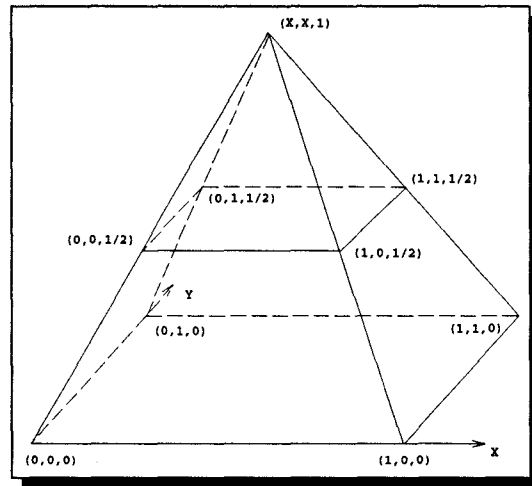


FIG. 3.16 - Base BPP de l'espace

La base BPP des polynômes exprimés avec ces coordonnées s'écrit alors :

$$\text{BPP}_{i,j,k}^n(X, Y, Z) = B_i^{n-k}(X)B_j^{n-k}(Y)B_k^n(Z) \quad (3.26)$$

où les $B_i^n(t)$ sont les polynômes de Bernstein en t . Une surface algébrique est donc définie par :

$$f(x, y, z) = \sum_{k=0}^n \sum_{j=0}^{n-k} \sum_{i=0}^{n-k} \text{BPP}_{i,j,k}^n(X, Y, Z) f_{i,j,k} \quad (3.27)$$

Les points de contrôle $f_{i,j,k}$ sont répartis sur la pyramide (figure 3.17).

Une surface de degré n est représentée sous forme BPP par une équation de degré au plus $3n$. Par exemple, la sphère d'équation $f(x, y, z) = x^2 + y^2 + z^2 - 1$ en coordonnées cartésiennes s'exprime en coordonnées BPP (en plaçant la pyramide autour de l'origine) :

$$f(X, Y, Z) = -50Y^2 + 50Y - 31 + 50X - 50X^2 + 100ZY^2 - 100ZY + 90Z - 100ZX + 100ZX^2 - 50Z^2Y^2 + 50Z^2Y - 75Z^2 + 50Z^2X - 50Z^2X^2 = 0 \quad (3.28)$$

Ce polynôme s'exprime ensuite dans la base BPP (eq. 3.27) pour obtenir les points de contrôle sur la pyramide :

$$f_{i,j,k} = \begin{bmatrix} -31 & -6 & -31 \\ -6 & 19 & -6 \\ -31 & -6 & -31 \end{bmatrix} \begin{bmatrix} 14 & 14 \\ 14 & 14 \end{bmatrix} [-16] \quad (3.29)$$

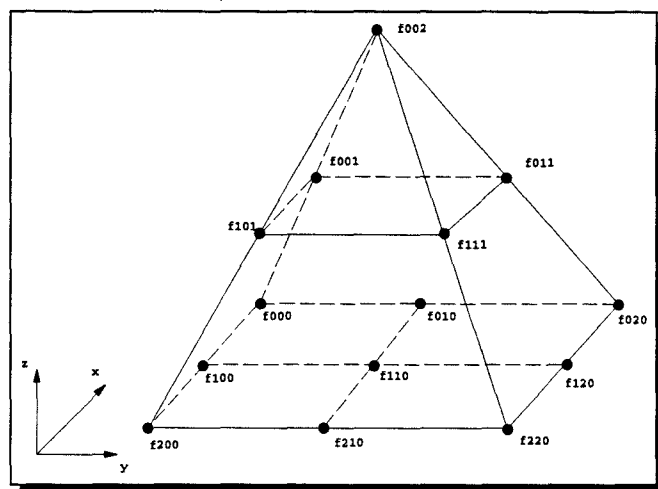


FIG. 3.17 - Polygone de contrôle d'une surface tri-quadratique sous forme BPP

L'algorithme de rendu utilise une base BPP particulière : le sommet de la pyramide est placé à la position de l'observateur et la base est placée à la position de l'écran dans la scène. La surface à afficher, exprimée à l'origine par une équation cartésienne est convertie en BPP. L'algorithme calcule d'abord la silhouette de la surface, c'est-à-dire son "bord", tel qu'il est vu par l'observateur. Exprimée sous la forme d'une courbe implicite $f(x, y) = 0$ dans le plan de l'écran, elle donne un critère rapide de présence de la surface pour un pixel (x, y) donné. La forme BPP permet d'effectuer ce calcul très rapidement. Il suffit ensuite de calculer la couleur de chaque pixel situé à l'intérieur de la silhouette.

Résultat : l'algorithme est très rapide pour une qualité d'image satisfaisante. Il permet d'afficher des surfaces algébriques dont le degré atteint 6 (la silhouette a alors une équation de degré 30) beaucoup plus rapidement qu'avec un algorithme de lancer de rayons, en affichant les singularités (par exemple les lignes multiples), ce que les autres méthodes ne font pas.

Développée spécifiquement pour l'affichage, la représentation BPP ne présente a priori pas de propriétés intéressantes pour la modélisation surfacique : elle utilise un système de coordonnées peu pratique (car non symétrique) et les équations générées sont de degré plus élevé que les équations dans la base standard (ce qui a la conséquence gênante d'utiliser un nombre de points de contrôle supérieur au nombre de degrés de liberté de la surface).

3.4 Bilan

Pour une classe "simple" de surfaces que sont les quadriques, nous avons vu qu'il en existe de nombreuses définitions mathématiques qui expriment chacune plus ou moins simplement les propriétés fondamentales de ces surfaces.

Toutefois, parmi ces représentations, seule celle de Bernstein-Bézier conserve les propriétés de la définition algébrique de façon homogène, tout en apportant des éléments nouveaux pour le contrôle de la forme ou de la continuité. Outre le fait que la conversion de la forme de Bernstein-Bézier vers la forme algébrique et inversement se fait de façon très simple, cette représentation représente donc l'outil le plus prometteur pour la construction de surfaces complexes.

Dans le prochain chapitre, nous entrons dans le vif du sujet. À l'aide des représentations que

3 LES QUADRIQUES ET LEURS REPRÉSENTATIONS

nous venons de voir, nous étudions la construction de surfaces complexes à l'aide de quadriques.

Chapitre 4

Modélisation à l'aide de surfaces quadriques

Les quadriques sont une classe de primitives extrêmement répandue en C.A.O. : les sphères, cônes et cylindres sont des objets géométriques bien connus, si bien que la plupart des systèmes de modélisation antérieurs à 1985 les incorporaient comme primitive principale [Miller86, VK93]. Toutefois les contrôles géométriques que ces systèmes offraient (comme la continuité des raccordements) étaient faibles, voire nulles, et l'on se fiait souvent à des critères uniquement esthétiques pour modéliser.

À présent, il est parfois nécessaire d'obéir à des contraintes beaucoup plus fortes (spécifiées par exemple par un programme d'analyse hydrodynamique ou en propagation de chaleur). Les surfaces paramétriques (cf. chapitre 2) répondent en général bien à ce genre de contraintes, alors que les quadriques sont moins utilisées, sauf dans les domaines où la simplicité de celles-ci permettent que les calculs effectués par le modéleur (intersection, transformations) restent très précis, ou autorisent un contrôle fiable de la topologie du modèle.

Certaines des techniques développées pour les surfaces paramétriques peuvent également être appliquées aux quadriques, en particulier en ce qui concerne le contrôle de la forme : les représentations récentes, vues au chapitre précédent, permettent la conception d'outils de manipulation des quadriques, dans le but d'obtenir de nouvelles possibilités de modélisation réservées à présent aux surfaces de Bézier ou autres B-splines.

Ce chapitre présente dans un premier temps les principales méthodes de création d'objets composés de quadriques. L'ordre choisi dans la présentation va dans le sens d'une augmentation du contrôle de la forme (correspondant à l'ordre chronologique) : du modèle naturel où seul un unique type de quadrique est utilisé (sphère, parabolôïde, etc.), jusqu'aux modèles où le contrôle de la forme est optimal et l'on ne se soucie plus des types de quadrique utilisés, mais plutôt de la forme globale de la surface. Nous présentons ensuite deux modèles spécifiques de surfaces déformables quadriques par morceaux répondant aux problèmes posés par les modèles antérieurs.

4.1 Manipulation directe de la représentation

L'idée la plus simple pour un schéma de modélisation à l'aide de quadriques est probablement de laisser à l'utilisateur la manipulation *directe* des quadriques à travers l'une des représentations du chapitre 3. Par exemple, en choisissant la représentation algébrique, la manipulation directe consiste à laisser l'utilisateur modifier les 10 coefficients de l'équation algébrique (équation 3.1, page 34). Puisque ces coefficients définissent complètement la surface, leur contrôle direct a l'avantage, d'une part, d'exploiter tous les degrés de liberté (on peut obtenir toutes les quadriques). D'autre part, aucune conversion n'est nécessaire pour l'affichage puisque l'on travaille directement dans la représentation finale (cf. chapitre 1).

Toutefois l'utilisateur se rend rapidement compte de la difficulté de cette méthode car tout d'abord la signification géométrique de ces coefficients est très indirecte (§ 3.1.3) : par exemple, la forme de la quadrique n'est déterminée qu'après le calcul du déterminant et des vecteurs propres de la matrice de la quadrique, sans compter les opérations de diagonalisation et d'orthonormalisation ([Leroux et al.90], page 247). De plus, comme il n'y a pas d'expression simple de la continuité, la construction directe de surfaces par morceaux continues est une tâche extrêmement difficile.

La situation s'améliore si on laisse à l'utilisateur la manipulation de patches quadriques sous la forme de Bernstein-Bézier. Le contrôle de la forme est plus simple, mais les autres problèmes subsistent : le nombre élevé de degrés de liberté (10 pour les coefficients et 12 pour le tétraèdre) pour une surface composite qui rend difficile toute manipulation globale de la surface, et le contrôle de la continuité qui n'est pas automatique pour toutes les valeurs des points de contrôle.

Quelle que soit la représentation utilisée, la manipulation directe du modèle n'est pas satisfaisante. Tous les schémas que nous présentons sont donc des modèles de plus haut niveau que les représentations du chapitre 3, censés simplifier la construction de surfaces en exploitant leurs propriétés.

4.2 Modélisation traditionnelle

Ce qui caractérise l'utilisation "classique" des quadriques en modélisation géométrique, c'est que l'on spécifie toujours à la création le type de chaque quadrique que l'on va utiliser : pour modéliser une forme particulière, on va par exemple utiliser un ellipsoïde, ou un morceau de cylindre. Après avoir choisi ce type, les autres paramètres de la surface sont fixés : taille, position, orientation.

Les systèmes de modélisation les plus simples où l'on utilise des quadriques sont en fait à base de sous-classes : seuls certains types de quadriques sont utilisés, soit parce qu'ils correspondent à des formes bien connues soit parce qu'ils répondent à une contrainte spécifique.

4.2.1 Quadriques naturelles

Les quadriques naturelles sont les sphères, les cylindres et les cônes. Ce sont les premières primitives 3D à avoir été utilisées en modélisation géométrique, puisqu'elles correspondent à des formes simples et suffisantes pour modéliser un nombre important d'objets :

- Dans notre environnement, un grand nombre d'objets manufacturés sont composés de morceaux de plans et de quadriques naturelles : arrondis, tubes, etc.

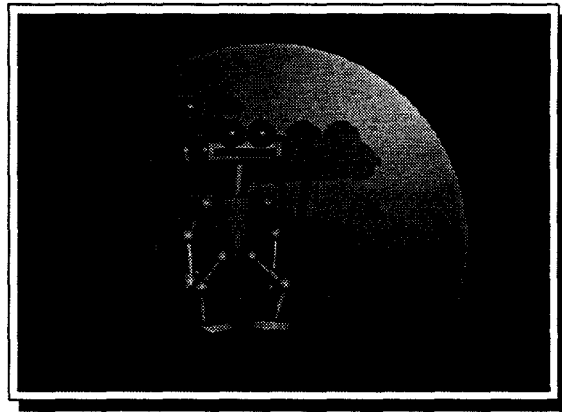


FIG. 4.1 - Personnage créé à l'aide de quadriques naturelles [Nyiri94]

- Certaines applications spécifiques n'utilisent parfois qu'un seul ou deux types : pour la modélisation moléculaire la plupart des représentations 3D utilisent des sphères, avec éventuellement des cylindres pour les représentations éclatées [Max81]. D'autres systèmes sont basés sur des combinaisons cônes/sphères [Max90], ou ellipsoïdes exclusivement [Spencer94].

La modélisation de formes complexes à l'aide de quadriques naturelles s'obtient en combinant des volumes quadriques à l'aide d'opérations booléennes (chapitre 1). C'est un processus assez intuitif, les opérations booléennes se traduisant bien en termes de fabrication (fraisage, chanfrein...).

La représentation des quadriques la plus simple à utiliser pour ce type de modélisation est bien sûr le modèle géométrique (§ 3.3.1), puisque le type de la quadrique y est explicitement codé, ainsi que les valeurs qui peuvent être modifiées par l'utilisateur : position, orientation et taille.

4.2.2 Quadriques spécifiques

Un très grand nombre de problèmes physiques ont des solutions quadratiques. Géométriquement, cette constatation se traduit par l'abondance de structures représentées par des surfaces de degré 2. Ainsi un modèle géométrique issu de contraintes physiques simples, pourra souvent être représenté par des quadriques.

Certains exemples parmi les plus courants sont les tunnels, qui sont modélisés à l'aide de cylindres elliptiques pour rendre optimale la réponse aux contraintes de pression externe, les antennes paraboliques, les membranes de haut-parleurs, etc.

Ici également, la représentation des quadriques la plus intéressante est le modèle géométrique, encore une fois parce que l'on n'utilise qu'un sous-ensemble restreint des surfaces, selon leur type.

4.2.3 Quadriques de révolution

Les objets créés à l'aide de quadriques naturelles ou spécifiques restent néanmoins limités au niveau de la forme. À part certains cas simples (par exemple, une demi-sphère fermant un cylindre), les raccordements ne sont jamais tangents et ne sont pas assurés lors d'une déformation de la surface. De plus, hormis le changement d'échelle, toute opération de déformation continue de la surface (comme l'ajout d'une bosse, d'un creux, ou l'élongation d'une partie de la surface dans

une direction) est non fermée, c'est-à-dire produit des surfaces qui ne sont pas des quadriques. Ces limitations empêchent toute modélisation où certaines contraintes sur la forme existent dans le but de répondre à des propriétés physiques ou éventuellement à des exigences esthétiques.

La première méthode où l'on ne se soucie pas forcément du type de quadrique employé mais plutôt de l'aspect global de la surface et des possibilités de déformation est la création de quadriques de révolution. Malgré que, encore une fois, seule une sous-classe des quadriques est employée, une modélisation par déformation de la surface (qui néanmoins reste de révolution) et un contrôle de la continuité sont possibles. Dans cette optique, nous proposons ici une méthode simple de création et de manipulation de quadriques de révolution.

Caractérisation

Dans le but de proposer une méthode de création par déformation, caractérisons les quadriques de révolution dans la représentation algébrique. Sans perte de généralité, nous supposons que l'axe de révolution est l'axe z . Une quadrique est de révolution autour de z si son intersection avec un plan d'équation $z = k$ est un cercle de centre $(k, 0, 0)$, ce qui s'écrit :

$$\begin{cases} Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz \\ + 2Fzx + 2Gx + 2Hy + 2Iz + J = 0 \\ z = k \end{cases} \equiv \begin{cases} A'(x^2 + y^2) - R^2 = 0 \\ z = k \end{cases} \quad (4.1)$$

On en conclut immédiatement :

$$\begin{cases} A = B = A' \\ D = E = F = G = H = 0 \\ R^2 = -Ck^2 - 2Ik - J \end{cases} \quad (4.2)$$

Une quadrique de révolution autour de z s'écrit donc de manière simplifiée :

$$A(x^2 + y^2) + Cz^2 + Iz + J = 0 \quad (4.3)$$

Suivant les valeurs des coefficients, les quadriques représentées par cette équation peuvent être toutes les quadriques de révolution : cône, cylindre, ellipsoïde, parabolôïde, hyperbolôïde à une ou deux nappes.

Le *profil* de la surface est défini comme étant la courbe formée par l'intersection de la quadrique et de tout plan passant par l'axe Oz , en particulier le plan d'équation $x = 0$. L'équation algébrique du profil dans ce plan est :

$$\begin{cases} A(x^2 + y^2) + Cz^2 + Iz + J = 0 \\ x = 0 \end{cases} \Rightarrow Ay^2 + Cz^2 + Iz + J = 0 \quad (4.4)$$

Modélisation par déformation

A partir de ces définitions, il est possible de concevoir un schéma simple de création de surfaces : la courbe de profil d'une quadrique de révolution peut être directement modélisée par interpolation,

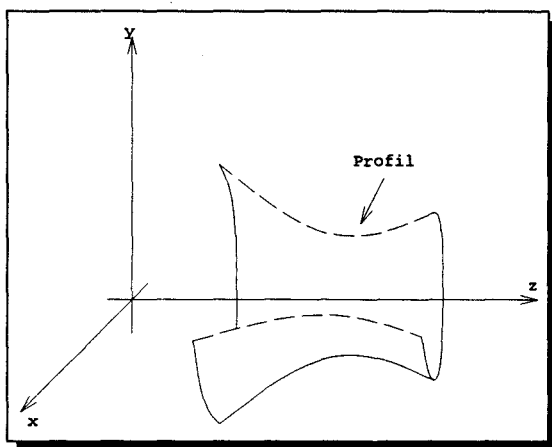


FIG. 4.2 - *Hyperboloïde de révolution*

son équation ayant trois degrés de liberté, trois points d'interpolation la définissent entièrement (figure 4.3) : l'équation de la courbe interpolant les points (y_1, z_1) , (y_2, z_2) , (y_3, z_3) est donnée par le déterminant [Pratt87] :

$$P(y, z) = \begin{vmatrix} y_1^2 & z_1^2 & z_1 & 1 \\ y_2^2 & z_2^2 & z_2 & 1 \\ y_3^2 & z_3^2 & z_3 & 1 \\ y^2 & z^2 & z & 1 \end{vmatrix} \quad (4.5)$$

d'où

$$\begin{cases} A = z_2^2 z_1 - z_1^2 z_2 + z_3^2 z_2 + z_1^2 z_3 - z_2^2 z_3 - z_3^2 z_1 \\ C = y_1^2 z_2 + y_2^2 z_3 + y_3^2 z_1 - y_3^2 z_2 - y_1^2 z_3 - y_2^2 z_1 \\ I = y_1^2 z_3^2 - y_2^2 z_3^2 + y_2^2 z_1^2 - y_3^2 z_1^2 - y_1^2 z_2^2 + y_3^2 z_2^2 \\ J = y_1^2 z_2^2 z_3 - y_3^2 z_2^2 z_1 - y_1^2 z_3^2 z_2 + y_2^2 z_3^2 z_1 - y_2^2 z_1^2 z_3 + y_3^2 z_1^2 z_2 \end{cases} \quad (4.6)$$

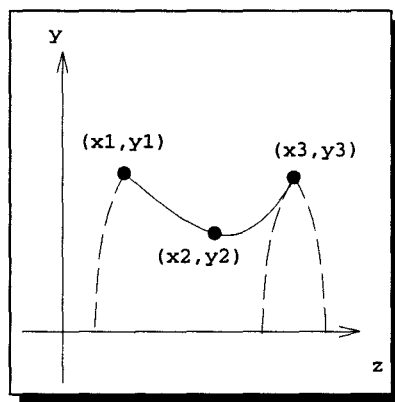


FIG. 4.3 - *Interpolation de 3 points par le profil*

Les coefficients A, C, I et J déterminent complètement le profil, et par conséquent la quadrique de révolution. En principe, les points d'interpolation ne sont pas forcément dans le même plan, mais il est plus facile pour l'utilisateur de saisir le profil dans un plan et de voir la quadrique résultante s'afficher ensuite. On limite la surface le long de l'axe Oz par les coordonnées z minimale et maximale des points d'interpolation.

Ce procédé s'étend facilement aux surfaces par morceaux. Il suffit de définir plusieurs quadriques de révolution selon le même axe, puis de les mettre bout-à-bout. Le raccordement de ces quadriques deux à deux se fait toujours le long d'un cercle, et les ordres de continuité possibles sont comme dans le cas général G^0 et G^1 .

La continuité G^0 est obtenue simplement en confondant les points d'interpolation extrémités des profils adjacents (figure 4.4).

La continuité G^1 des profils des quadriques adjacentes entraîne la continuité G^1 de la surface. Pour cela il suffit de rendre colinéaires les normales à gauche et à droite du point considéré. Pour obtenir les coordonnées vecteur normal au profil au point (y, z) , il suffit de dériver l'équation 4.4 par rapport aux coordonnées :

$$\vec{N} = \nabla(Ay^2 + Cz^2 + Iz + J) = \begin{cases} 2Ay \\ 2Cz + I \end{cases} \quad (4.7)$$

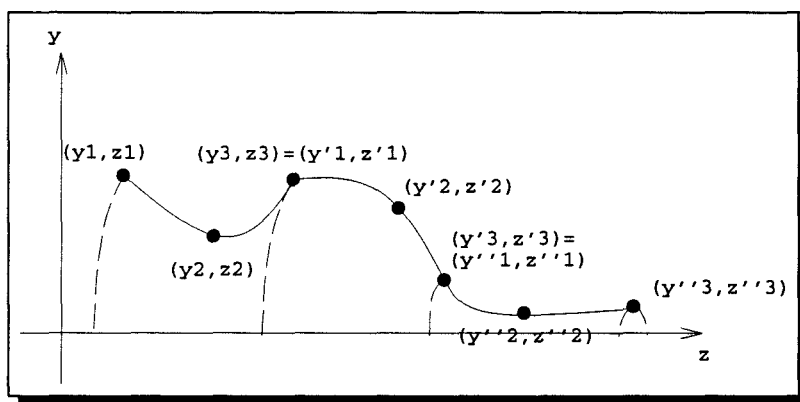
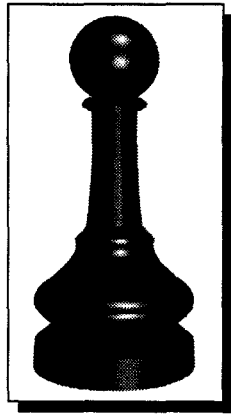


FIG. 4.4 - Surface de révolution par morceaux (continuité G^0 en (y_3, z_3) et G^1 en (y'_3, z'_3)).

Le procédé que nous venons de décrire permet de créer des surfaces complexes composées de quadriques par déformation d'un polygone de contrôle. C'est un premier pas vers des techniques de modélisation de haut niveau. Pour l'instant, les degrés de liberté obtenus sont dans le contrôle de la courbe, en 2D. Obtenir un contrôle maximal des degrés de liberté de la quadrique, dans toutes les dimensions, s'obtient évidemment de façon moins simple, nous le verrons dans les paragraphes suivants.

4.3 Interpolation

La première possibilité pour utiliser les quadriques sans limitations sur la forme est de généraliser l'interpolation de points. L'interpolation par des surfaces algébriques est bien connue et a même été généralisée à l'interpolation de paires point/normale, de courbes (éventuellement avec

FIG. 4.5 - *Objet de révolution (8 quadriques)*

normale en chaque point) et également à l'*approximation* de points : dans le cas où l'interpolation n'a pas de solution, des surfaces passant le plus près possible des points sont calculées.

4.3.1 Méthode de Pratt

V. Pratt étudie les méthodes d'interpolation de points par des familles de surfaces algébriques [Pratt87]. Chaque contrainte (i.e. point d'interpolation) définit une équation. L'ensemble des équations générées par toutes les contraintes constitue un système dont les inconnues sont les coefficients de l'équation algébrique.

Une famille de surfaces algébriques est définie comme un espace vectoriel de fonctions algébriques, dont les bases sont de la forme: $\{b_0(x, y, z), \dots, b_k(x, y, z)\}$. Par exemple, une base de l'espace des sphères (d'équation $A(x^2 + y^2 + z^2) + Bx + Cy + Dz + E = 0$) est $\{x^2 + y^2 + z^2, x, y, z, 1\}$. Il s'agit bien sûr d'un sous-espace de l'espace vectoriel des quadriques.

Chaque point à interpoler (x_i, y_i, z_i) appartient à la surface cherchée, d'équation

$$\sum_{j=0}^k \gamma_j b_j(x, y, z) = 0 \quad (4.8)$$

où les coefficients γ_j sont les inconnues. Le point vérifie donc :

$$\sum_{j=0}^k \gamma_j b_j(x_j, y_j, z_j) = 0 \quad (4.9)$$

Dans l'exemple de la sphère :

$$\gamma_0(x_1^2 + y_1^2 + z_1^2) + \gamma_1 x_1 + \gamma_2 y_1 + \gamma_3 z_1 + \gamma_4 = 0 \quad (4.10)$$

L'ensemble des équations générées par m points à interpoler forme le système :

$$A\Gamma = 0 \tag{4.11}$$

où

$$A = \begin{pmatrix} x_0^2 + y_0^2 + z_0^2 & x_0 & y_0 & z_0 & 1 \\ x_1^2 + y_1^2 + z_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m-1}^2 + y_{m-1}^2 + z_{m-1}^2 & x_{m-1} & y_{m-1} & z_{m-1} & 1 \end{pmatrix} \quad \text{et} \quad \Gamma = (\gamma_0 \quad \gamma_1 \quad \gamma_2 \quad \gamma_3) \tag{4.12}$$

En comparant le nombre k de degrés de liberté de la surface et le nombre m de points à interpoler, Pratt déduit l'existence de solutions :

- $m < k$: il n'y a pas suffisamment de points pour spécifier une surface unique (cas *sous-déterminé*). Pratt ne traite pas ce cas.
- $m = k$: une solution (au minimum) existe, et celle-ci est donnée par le déterminant d'une matrice S dérivée de A . Dans le cas de la sphère :

$$S = \begin{pmatrix} x_1^2 + y_1^2 + z_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 + z_3^2 & x_3 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 + z_4^2 & x_4 & y_4 & z_4 & 1 \\ x^2 + y^2 + z^2 & x & y & z & 1 \end{pmatrix} \tag{4.13}$$

le déterminant de S est un polynome en (x, y, z) qui représente l'équation de la surface solution.

La situation où les points sont tels que plusieurs surfaces différentes les interpolent (caractérisée par le fait que le rang de A est inférieur à m) est un cas sous-déterminé.

- $m > k$: trois cas sont possibles :
 - $\text{rang}(A) < k$: cas sous-déterminé
 - $\text{rang}(A) = k$: tous les points sont sur une unique surface solution. L'équation de celle-ci est obtenue en sélectionnant k lignes de A et en appliquant la méthode précédente.
 - $\text{rang}(A) > k$: cas *sur-déterminé* : il n'existe pas de surface interpolant tous les points. On cherche alors une solution approchée. Pratt donne une méthode simple basée sur une décomposition de la matrice A . Dans un premier temps, il faut calculer la matrice $k \times k$ triangulaire supérieure U telle que : $A^t A = U^t U$ (décomposition de Cholesky). En supprimant la dernière ligne de U , la solution est donnée comme précédemment, en calculant le déterminant associé (équation 4.13).

Dans le cas des quadriques, la base de l'espace vectoriel de l'équation algébrique est

$$\{x^2, y^2, z^2, xy, yz, xz, x, y, z, 1\}.$$

Il s'agit donc un espace de dimension 10, et la méthode d'interpolation montre qu'au moins 9 points sont nécessaires pour spécifier complètement une surface. Dans le cas de surfaces par

morceaux, ce nombre est à multiplier par le nombre de quadriques. C'est beaucoup pour une utilisation directe en modélisation. En outre, deux problèmes importants ne sont pas résolus : le contrôle de la continuité entre quadriques, le problème des nappes (les quadriques interpolant les points en plusieurs surfaces déconnectées, cf. figure 4.6) ou les quadriques dégénérées (deux plans).

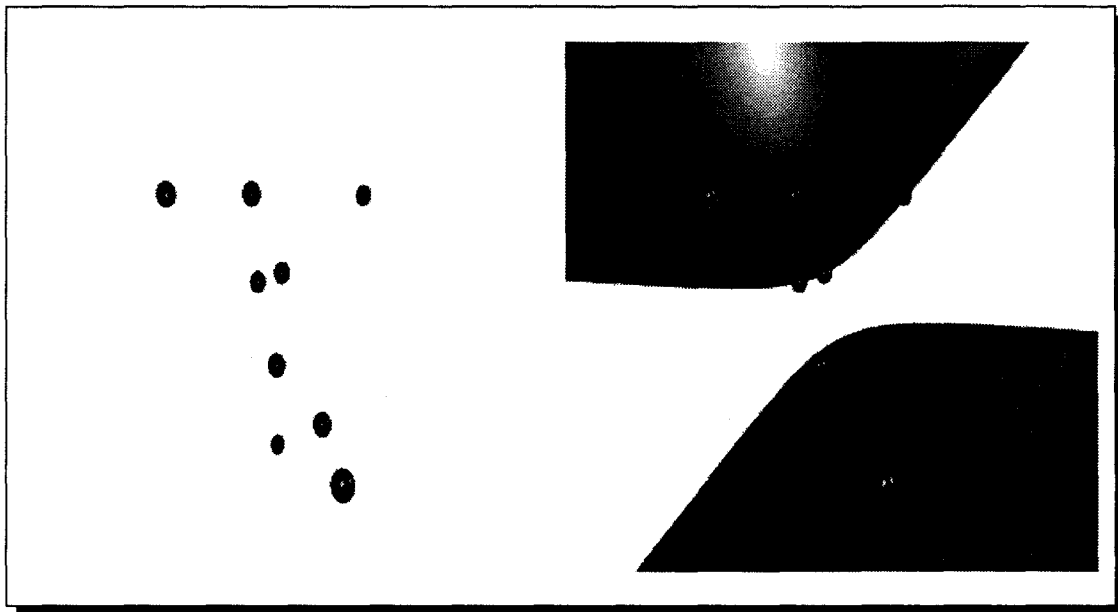


FIG. 4.6 - Interpolation de neuf points par deux nappes d'une même quadrique

4.3.2 Méthode de Bajaj et Ihm

Bajaj et Ihm étendent les résultats de Pratt à l'interpolation de Hermite (c'est-à-dire interpolation de points avec éventuellement normales) de points et de courbes [Bajaj et al.92] :

- points
- points avec vecteurs normaux
- courbes algébriques (définies comme l'intersection de deux surfaces algébriques)
- courbes algébriques avec normales (la normale est exprimée par trois polynômes $n_x(x, y, z)$, $n_y(x, y, z)$, $n_z(x, y, z)$, définis en chaque point de la courbe)
- courbes paramétriques rationnelles, i.e. s'exprimant sous la forme : $x(t) = X(t)/W(t)$, $y(t) = Y(t)/W(t)$, $z(t) = Z(t)/W(t)$.
- courbes paramétriques rationnelles avec normales (la normale est exprimée par trois polynômes $n_x(t)$, $n_y(t)$, $n_z(t)$).

Chaque interpolation à assurer est exprimée par une ou plusieurs équations (en fonction du type de données) qui sont ajoutées au système. Le système $A\Gamma = 0$ (eq. 4.11) est ensuite résolu grâce à une technique classique de résolution des systèmes linéaires homogènes : la *singular value decomposition* qui donne immédiatement le rang du système et toutes les solutions (alors que Pratt n'en sélectionne qu'une).

Le rang r du système détermine le nombre de degrés de liberté de la surface solution. Si $r \geq n^2/2 + 3n/2 + 1$, il existe une surface de degré $n - 1$ solution au problème. Quand le nombre de degrés de liberté restant n'est toutefois pas suffisant pour qu'une surface de degré inférieur satisfasse les contraintes, Bajaj et Ihm proposent de convertir la surface sous forme de Bernstein-Bézier et de laisser l'utilisateur fixer la forme finale de la surface en manipulant les coefficients (en assurant l'invariant qui entraîne l'interpolation).

Si on se limite aux quadriques, cette solution semble plus intéressante que celle de Pratt: trois paires point/normale suffisent pour définir une quadrique, ce qui est plus intéressant du point de vue interactif que la manipulation de 9 points sans normale. Toutefois on n'a d'une part plus de solution exprimée de façon explicite (puisqu'il faut résoudre le système 4.11) et, d'autre part, B. Guo démontre que, dans le cas des l'interpolation point/normale par une quadrique, la comparaison des degrés de liberté et du rang du système est trompeuse puisque le système n'a pas toujours de solution (§ 4.3.3). Finalement, le problème des nappes et des solutions dégénérées existe toujours.

Bajaj, Ihm et Warren étendent la méthode précédente à l'approximation (aux moindres carrés) de points et de courbes, et à l'expression algébrique de l'interpolation d'ordres supérieurs à 1 [Bajaj et al.93]. Les problèmes et inconvénients cités subsistent.

4.3.3 Macropatchs quadriques

Plusieurs auteurs se sont penchés sur certains cas particulier des problèmes précédents : des surfaces de degré fixé qui interpolent des données de type fixé également (par exemple, l'interpolation de points par des cubiques), d'une part dans le but de contrôler l'existence de solutions, et d'autre part d'exprimer celles-ci de façon explicite, sans résolution de systèmes d'équations. Nous présentons deux de ces résultats concernant les quadriques. Le point commun de ces méthodes est qu'elles utilisent la représentation de Bernstein-Bézier (cf. chapitre 3).

Méthode de Dahmen

W. Dahmen s'attaque au problème de l'interpolation de paires points/normales par une surface quadrique par morceaux, avec continuité G^1 [Dahmen89]. Les points sont munis d'une structure de triangulation: ils sont les sommets d'un polyèdre dont les facettes sont des triangles (figure 4.7).

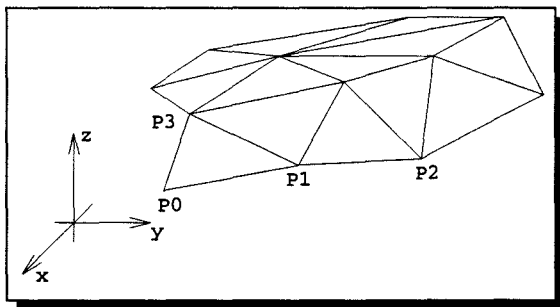


FIG. 4.7 - Triangulation dans l'espace

Le processus commence par la définition d'un *transversal system* : un ensemble de droites telles que

1. Chacune d'entre elles traverse un des triangles.

2. Pour deux triangles adjacents (P_1, P_2, P_3) et (P_2, P_3, P_4) , leurs droites associées, D_1 et D_2 , sont coplanaires et le plan qu'elles définissent intersecte le segment $[P_2P_3]$ (figure 4.8 a).

À partir de cette construction, Dahmen construit la surface interpolante à l'aide de patches quadriques dans la représentation de Bernstein-Bézier : à chaque triangle est associé deux ensembles de 6 carreaux quadriques (appelés *macropatches*, figure 4.8 b). Chaque macropatch forme une surface G^1 interpolant les trois points/normales. Pour rendre l'ensemble des macropatches continu, il reste à "boucher les trous" entre les macropatches des facettes adjacentes à l'aide de 8 nouveaux patches par arête (c'est la structure particulière du transversal system qui rend possible cette étape, cf. figure 4.8 c). Résultat : un réseau de n triangles est interpolé par $24n$ patches quadriques (pour une triangulation fermée).

Le gros inconvénient de cette méthode est que Dahmen ne donne pas d'algorithme de construction du transversal system dans le cas général, même si la construction est simple à réaliser dans certaines configurations particulières (par exemple une triangulation convexe).

W. Dahmen a plus récemment réussi à résoudre le même problème d'interpolation, mais à l'aide de surfaces algébriques de degré 3 [Dahmen et al.93].

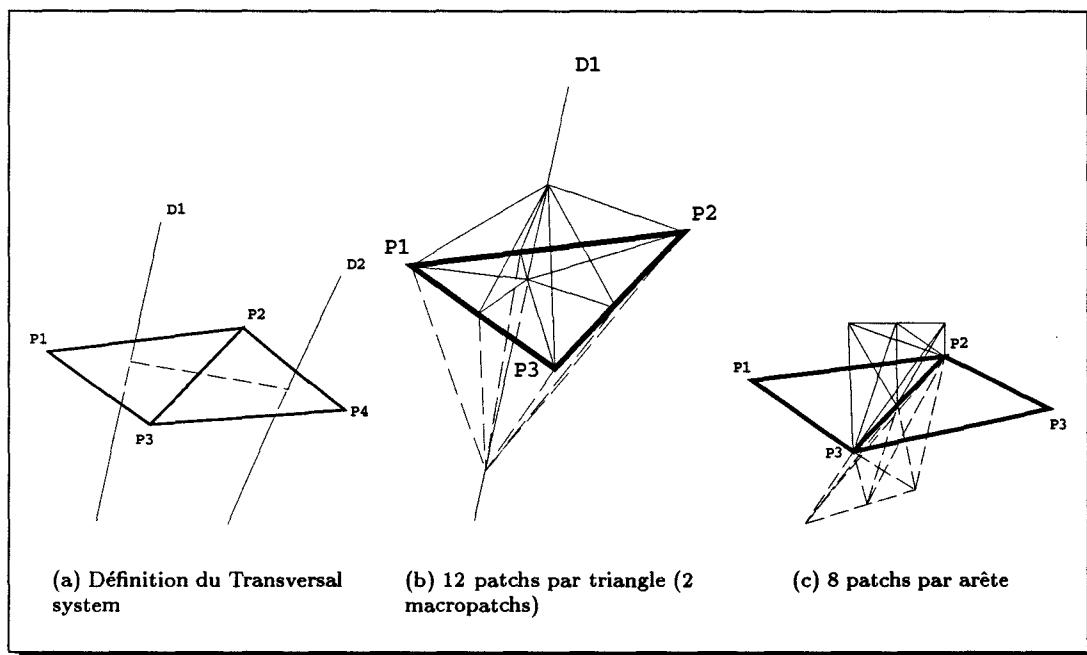


FIG. 4.8 - Méthode de Dahmen

Méthode de Guo

À partir du même énoncé (interpoler un réseau de points/normales par une surface quadrique G^1 par morceaux), B. Guo propose une méthode similaire, sans résoudre complètement le problème [Guo93b].

Par rapport à la méthode de Dahmen qui prenait en compte la longueur du vecteur normal à la surface (i.e. le gradient de la surface quadrique était égal à la normale), Guo utilise la normale dans le but de définir uniquement l'orientation du plan tangent à la surface. Ceci diminue les contraintes

sur la surface et permet donc d'augmenter le nombre de degrés de liberté du système pour réduire le nombre de quadriques. Les 6 patchs quadriques de chaque macropatch sont remplacés par un seul patch (figure 4.9 a). Cela permet également de supprimer le transversal system et de n'utiliser que quatre quadriques par arête, plutôt que 8 (figure 4.9 b).

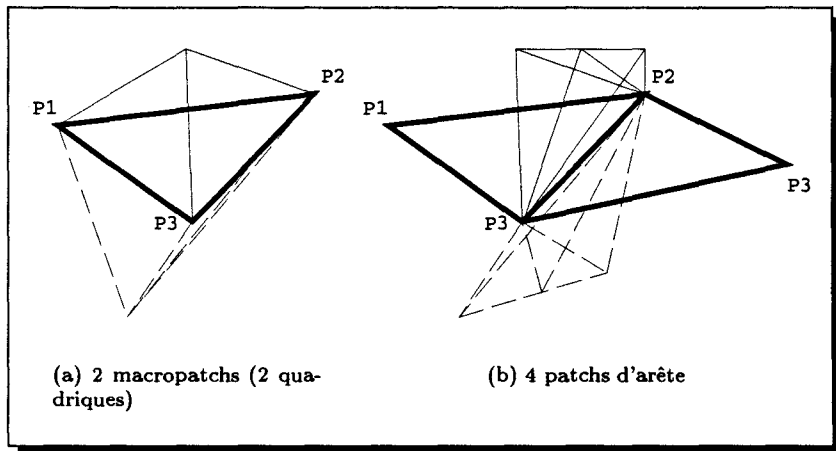


FIG. 4.9 - Méthode de Guo

Toutefois cette amélioration est limitée par une condition sur les normales en chaque point de la triangulation, la *Tangent Planes Condition* (TPC) :

Théorème 4 (d'après [Guo93b]) Soit un triangle (P_1, P_2, P_3) et en chaque point P_i , un vecteur \vec{N}_i . Il existe une quadrique interpolant les trois points et ayant le vecteur \vec{N}_i comme normale en P_i si et seulement si l'une de ces deux propositions est vraie :

1. Deux des trois normales sont orthogonales à l'arête du triangle entre elles (figure 4.9 a) ou,
2. Aucune des normales n'est orthogonale à une arête adjacente et l'équation :

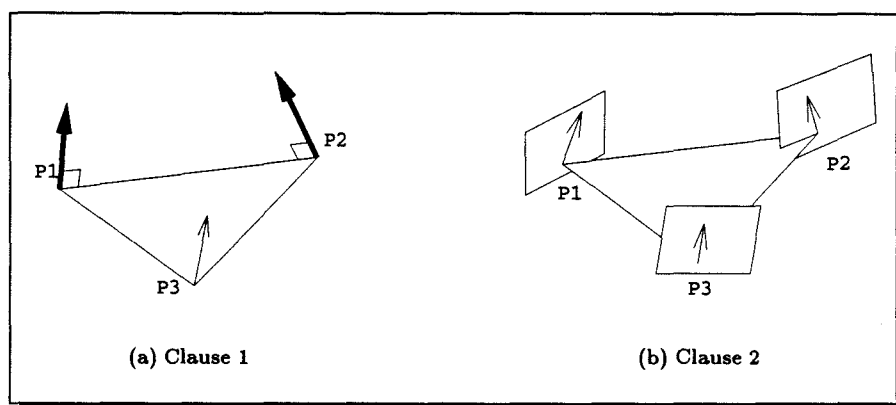
$$\frac{\vec{P}_{12} \cdot \vec{N}_1}{\vec{P}_{13} \cdot \vec{N}_1} \times \frac{\vec{P}_{23} \cdot \vec{N}_2}{\vec{P}_{21} \cdot \vec{N}_2} \times \frac{\vec{P}_{31} \cdot \vec{N}_3}{\vec{P}_{32} \cdot \vec{N}_3} = 1 \quad (4.14)$$

est vérifiée (figure 4.9 b).

Ce théorème montre, comme nous l'avons dit, que même si même si trois points/normales sont a priori suffisant pour définir une quadrique avec les méthodes de Bajaj (§ 4.3.2), ce n'est pas toujours le cas et le système n'admet pas de solutions.

Guo suppose ensuite cette condition vérifiée dans les points/normales qui sont traités par son algorithme. Toutefois, l'interprétation géométrique du cas général (c'est-à-dire l'équation 4.14) de la TPC montre que celle-ci n'est pas vérifiée en général (figure 4.10) : chaque terme $\frac{\vec{P}_{ij} \cdot \vec{N}_i}{\vec{P}_{ik} \cdot \vec{N}_i}$ reste constant si la normale \vec{N}_i est incluse dans un certain plan perpendiculaire à la facette. L'équation 4.14 est donc un invariant sur l'orientation des trois plans pour chaque normale : si deux normales sont fixées, la troisième devra appartenir à un certain plan orthogonal à la facette. Il n'existe pas à notre connaissance d'interprétation géométrique simple de cet invariant sur la position des plans.

Le méthode de Guo n'est donc pas applicable dans le cas général. Elle peut se révéler néanmoins intéressante pour interpoler des points dont la normale n'est pas précisée. En effet, celles-ci peuvent être calculées à partir de la triangulation pour que tous les triangles vérifient la TPC.

FIG. 4.10 - *Interprétation géométrique de la TPC*

Comme Dahmen, Guo a récemment trouvé une solution satisfaisante au problème d'interpolation à l'aide de patches cubiques [Guo93a].

4.4 Nouvelle définition des macropatches quadriques

4.4.1 Objectifs

Nous cherchons à définir une primitive de modélisation quadrique par morceaux, qui offre un nombre maximal de possibilités de modélisation :

1. Interpolation de points
2. Contrôle des normales aux points d'interpolation
3. Possibilité de déformation continue de façon la plus intuitive possible
4. Contrôle simple du raccordement G^0 ou G^1
5. Calcul simple de chaque quadrique permettant une modélisation rapide.

Comme nous l'avons vu, utiliser directement comme primitive un patch quadrique sous la forme de Bernstein-Bézier permet d'atteindre les quatre objectifs, mais pas de façon satisfaisante au niveau de l'interaction : 10 degrés de liberté par quadrique sont beaucoup pour une manipulation directe, et même si ceux-ci sont réduites par les conditions de continuité G^1 , ces conditions sont trop complexes pour être manipulées directement (§ 3.2.4). De plus, le théorème de la *tangent plane condition* de Guo (§ 4.3.3) limite également les possibilités directes de modélisation, puisque celui-ci montre que trois couples point/normales quelconques ne peuvent être interpolés par une seule quadrique.

Nous cherchons plutôt à définir une primitive de haut-niveau, sous la forme d'un macropatch, dont on n'a pas à contrôler tous les degrés de liberté et les conditions de continuité. Le modèle que nous avons choisi sont les surfaces de Bézier : un polygone de contrôle qui fixe les points d'interpolation et les plans de tangence aux limites de la surface et permet un contrôle tout à fait naturel puisque la surface "suit" celui-ci (§ 2.2).

Nous définissons donc dans un premier temps un polygone de contrôle qui définira la position de plusieurs patches quadriques. Celles-ci sont exprimées sous la forme de Bernstein-Bézier : les tétraèdres remplissent l'enveloppe convexe du polygone de contrôle, et les coefficients sont calculés de façon à satisfaire les contraintes d'interpolation de celui-ci et de continuité entre quadriques.

Parce que nous désirons (dans un premier temps du moins) avoir un nombre limité de quadriques par macropatch, il faut que la forme du macropatch reste simple, c'est-à-dire avec un nombre raisonnable de changements de courbure. Par conséquent le polygone de contrôle doit être lui aussi simple. Les surfaces de Bézier les plus simples qu'il est possible d'"imiter" sont les triangles de Bézier quadratiques (figure 2.9, page 27), dont le polygone de contrôle est composé de 6 points (figure 4.11) [Sederberg et al.85] . Trois points (P_1, P_3, P_5) sont interpolés par la surface, et les trois autres (P_2, P_4, P_6) contrôlent les tangentes aux points d'interpolation. Par convenance, nous tentons également d'obtenir les propriétés d'invariance affine et d'enveloppe convexe (cf. chapitre 2) de la surface par rapport à ce polygone de contrôle.

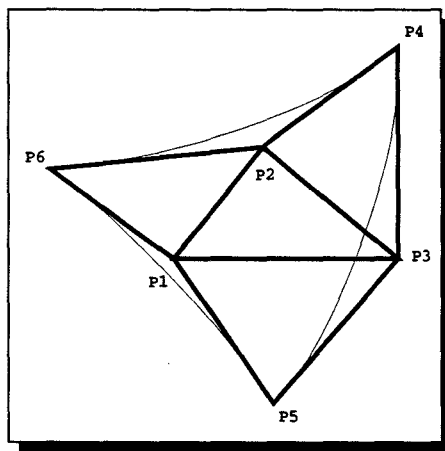


FIG. 4.11 - Polygone de contrôle d'un triangle de Bézier quadratique

La représentation de Bernstein-Bézier est évidemment préférable aux autres, pour les raisons de contrôle de la forme et de la continuité que nous avons évoquées précédemment (§ 3.2). Le critère principal dans toute la discussion qui va suivre est la continuité : nous savons qu'un raccord entre deux quadriques peut être G^0 ou G^1 . Nous essayons donc d'atteindre ces deux possibilités pour tous les raccords : soit entre patches quadriques à l'intérieur d'un macropatch, soit entre macropatches. Selon la continuité des raccords, un nombre différent de quadriques est nécessaire pour construire un macropatch, et les règles de détermination des coefficients de chaque quadrique sont également différentes.

Macropatches avec continuité G^0

La construction du macropatch s'effectue ici similairement à la méthode de Guo, sans prendre en compte la TPC. Quatre quadriques sous forme de Bernstein-Bézier suffisent à créer un macropatch G^0 (figure 4.12).

Les coefficients des quatre quadriques sont fixés de façon à interpoler les points de contrôle et les tangentes (0 aux points d'interpolation et le long des segments tangents), et également pour respecter la continuité G^0 entre quadriques . Les coefficients restant peuvent être fixés arbitrairement ou laissés à l'utilisateur comme coefficients supplémentaires de contrôle de la surface (figure 4.13)

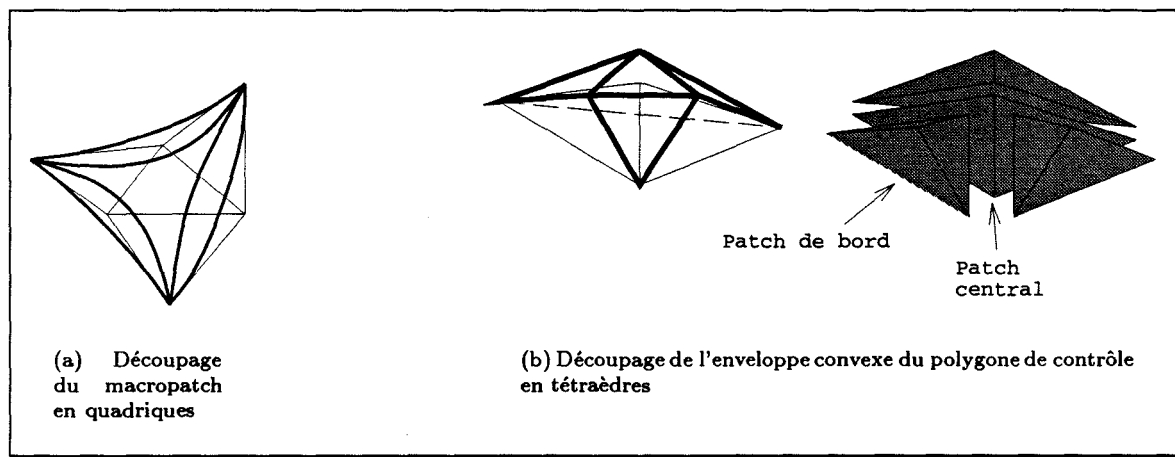


FIG. 4.12 - Macro Patch à 4 quadriques

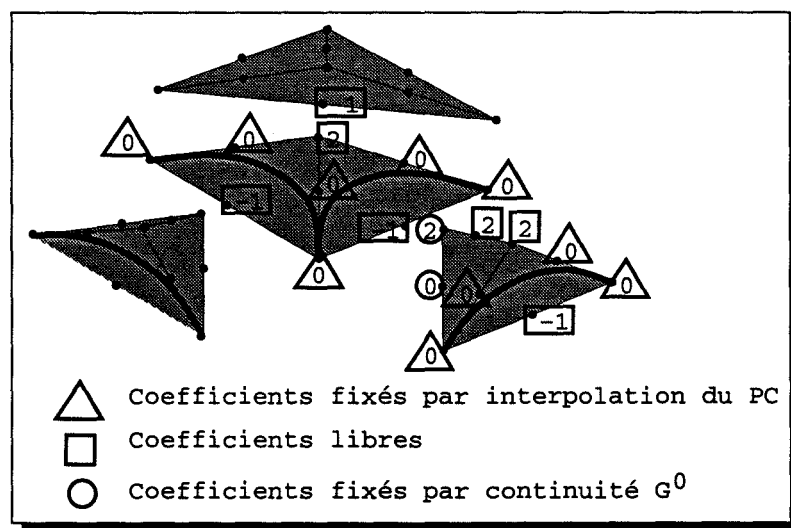
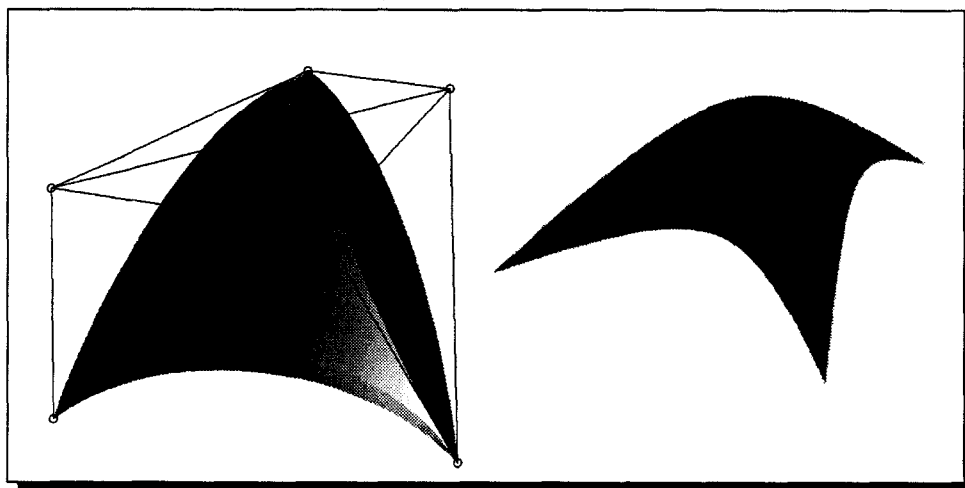
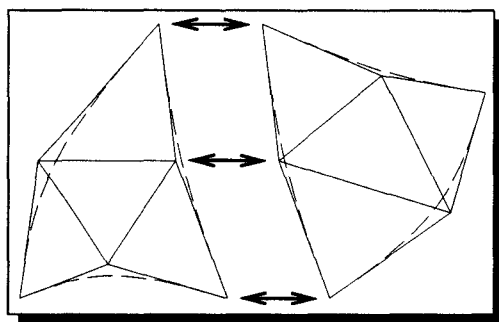


FIG. 4.13 - Coefficients des quadriques pour un macropatch G^0 (les coefficients non indiqués se déduisent par symétrie)

FIG. 4.14 - Exemples de macropatches G^0 à 4 quadriques

Le raccordement G^0 entre deux macropatches, le long d'un bord de chaque surface (figure 4.15), s'obtient en assurant la continuité des patches de part et d'autre de la face commune : les coefficients sur la frontière doivent être égaux (§ 3.2.4). Comme tous les coefficients d'un macropatch et a fortiori ceux appartenant aux faces de contact sont constants, ils sont donc forcément égaux deux à deux lors d'un raccordement. Celui-ci s'obtient donc simplement en raccordant les polygones de contrôle, de la même façon que les triangles de Bézier quadratiques.

FIG. 4.15 - Raccordement G^0 entre macropatches

Macropatches avec continuité G^1

Pour obtenir une surface quadrique par morceaux de continuité G^1 , interpolant le polygone de contrôle, il faut ajouter les contraintes de continuité G^1 entre patches. Pour cette raison, plus de patches sont nécessaires à l'intérieur d'un seul macropatch.

Première approche La première approche que nous ayons adoptée [Froumentin et al.94] a été de fixer arbitrairement le nombre de patches et leur configuration à l'intérieur d'un macropatch

selon plusieurs modèles (figure 4.16) et de calculer les coefficients, selon les contraintes suivantes :

1. Interpolation du polygone de contrôle : les coefficients qui assurent l'interpolation et la tangence sur le polygone de contrôle s'annulent.
2. Continuité G^1 entre les patches : les équations de raccordement (équation 3.23, page 47) sont appliquées à tous les coefficients impliqués dans chaque raccord.

Ces contraintes produisent un système d'équations linéaires, dont les inconnues sont les coefficients des patches du macropatch, dans une configuration particulière du polygone de contrôle. Avec la configuration à quatre quadriques du modèle G^0 , 18 équations sont générées. Dans ce cas le système n'admet pas de solutions pour toutes les positions du polygone de contrôle, ce qui indique donc qu'il n'y a pas moyen de créer un macropatch à 4 quadriques qui reste G^1 quelle que soit la forme du polygone de contrôle. Pour atteindre ce but, il est donc nécessaire d'augmenter le nombre de degrés de liberté du système, c'est-à-dire le nombre de quadriques par macropatch.

Plusieurs modèles de découpage sont possibles, mais on n'obtient de solution totalement G^1 qu'avec un découpage du macropatch en 18 quadriques (dit de Powell-Sabin, figure 4.16 d).

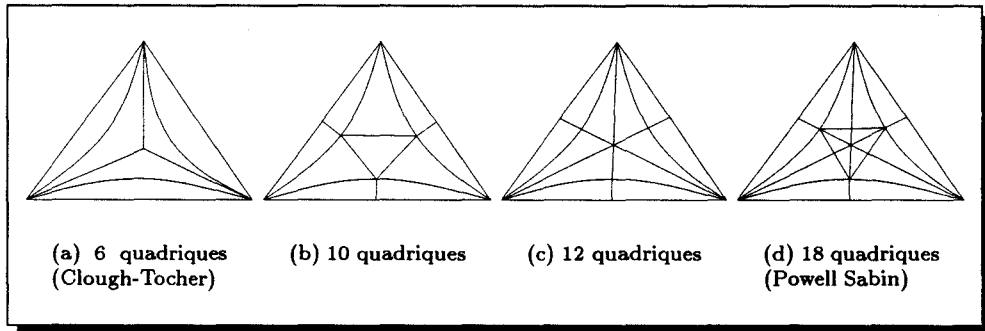


FIG. 4.16 - Différentes configurations de macropatches

Le nombre d'équations à résoudre pour 18 quadriques est 81. On s'en doute, la résolution d'un tel système ne se fait pas de façon immédiate et par conséquent toute possibilité d'utilisation interactive s'en trouve fortement réduite. Nous allons voir qu'il est toutefois possible d'élaborer des modèles de macropatches *explicites*, c'est-à-dire dont les coefficients sont calculables directement.

Solution explicite Nous montrons ici une solution G^1 construite à l'aide de 18 patches, qui évite la résolution du système d'équations, en donnant la valeur de chaque coefficient de façon explicite, à partir du polygone de contrôle. Le découpage, ainsi qu'un exemple sont visibles sur la figure 4.17.

Pour obtenir cette configuration, nous disposons les patches dans l'enveloppe convexe du polygone de contrôle en trois étapes (figure 4.18).

Les coefficients de Bernstein-Bézier des 18 quadriques sont calculés pour obtenir les conditions nécessaires à l'interpolation et à la continuité G^1 :

- Patches centraux (figure 4.19 a) : les trois normales à la surface aux trois points d'interpolation sont calculées à partir du polygone de contrôle. Elles sont ensuite normalisées. Finalement, les coefficients des quadriques sont fixés :
 - Nuls par interpolation : b_4, b_5, b_6 .

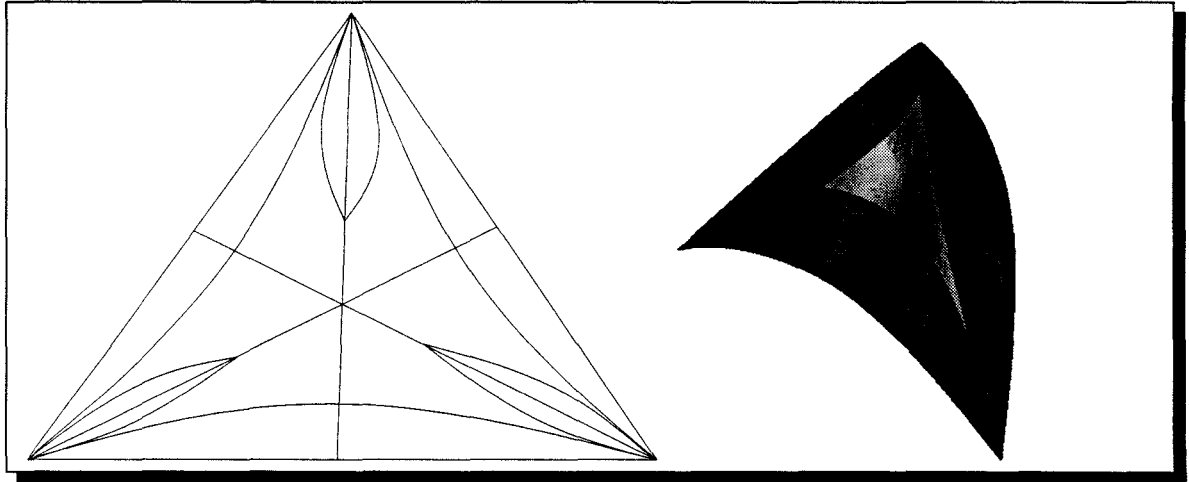


FIG. 4.17 - Macropatch G^1 à 18 quadriques

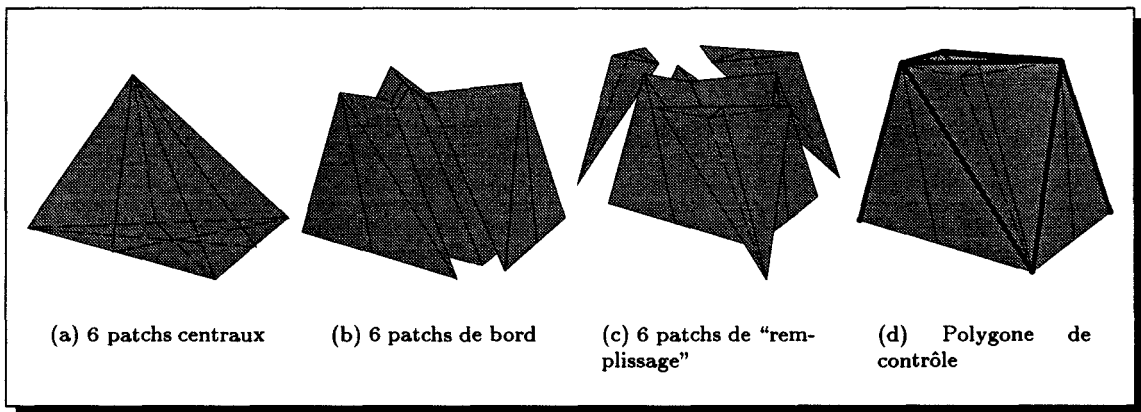


FIG. 4.18 - Construction explicite d'un macropatch G^1

- Calculés à partir de la normale au sommet correspondant, avec l'équation [Dahmen89] :

$$b_{ij} = \vec{n}_i \cdot \vec{v}_i v_j^k \quad (4.15)$$

où \vec{n}_i est le vecteur normal à la surface au point de contrôle P_i : b_{05} , b_{57} , b_{58} .

- Libre (positionné arbitrairement) : b_7
- Fixés par les conditions de raccordement G^1 entre patchs quadriques (équation 3.23) : $b_8 = (b_{58} + b_{48})/2$, $b_{78} = (b_{57} + b_{47})/2$, $b_{07} = (b_{47} + b_{57} + b_{67})/3$, $b_0 = (b_{08} + b_{09} + b_{0A})/3$.

• Patchs de bord : les coefficients sont fixés de trois façons (figure 4.19 b) :

- Nul par interpolation : b_{15} .
- Libres (positionnés arbitrairement) : b_1 .
- Fixés par les conditions de raccordement G^1 : $b_{1A}(= 0)$, b_{17} .

• Patchs de remplissage (figure 4.19 c) :

1. Fixés par les conditions de raccordement G^1 : b_{1D} , b_{3D} , $b_{5D}(= 0)$, b_{7D}

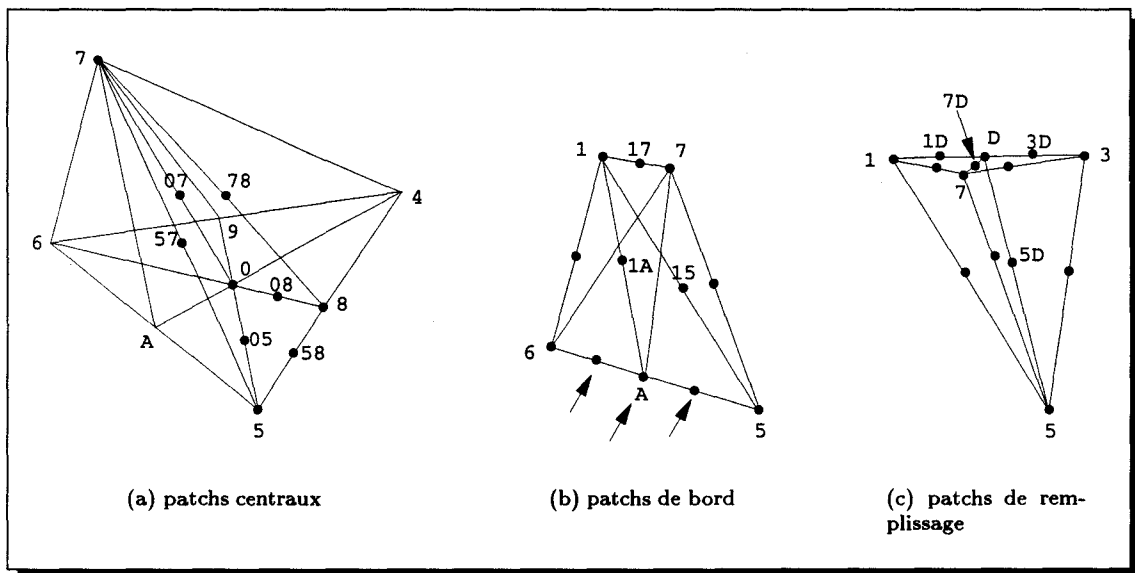


FIG. 4.19 - Coefficients des quadriques pour un macropatch G^1 (les coefficients non indiqués se déduisent soit par symétrie, soit par raccordement G^0)

Le fait que les coefficients puissent être calculés étape par étape (dans l'ordre donné ci-dessus) fait qu'une résolution d'un système d'équations décrivant les contraintes du macropatch est inutile.

Raccordement entre macropatchs

C'est dans le raccordement entre macropatchs qu'apparaissent les limites de notre modèle. Parce que plus de patchs interviennent dans un raccord (il y a deux quadriques par bord du macropatch) et parce qu'il faut toujours assurer la continuité G^1 à l'intérieur du macropatch, moins de degrés de liberté sont "disponibles" pour le raccordement.

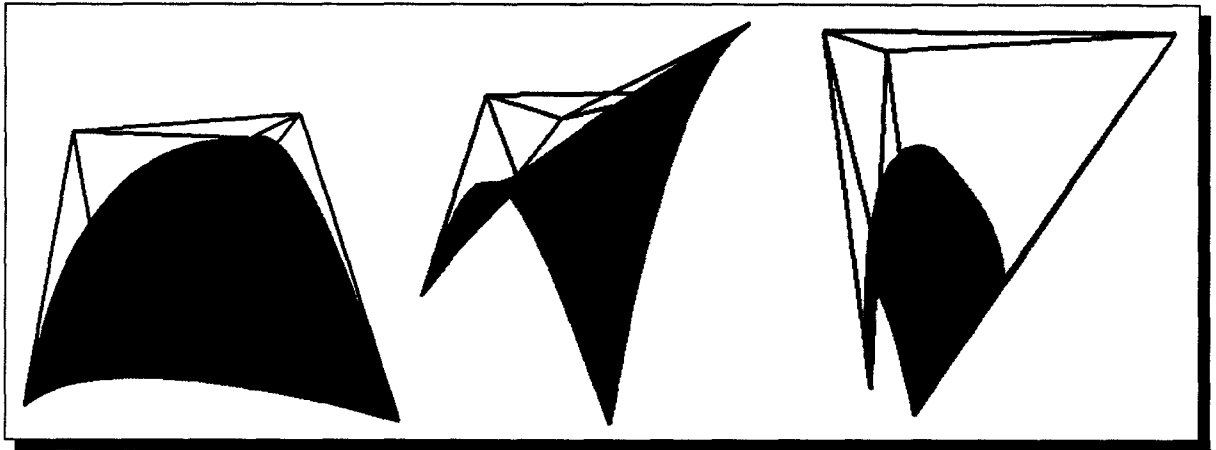
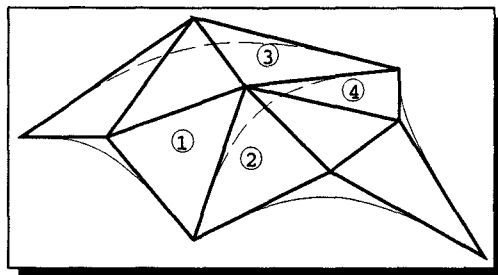


FIG. 4.20 - Déformation du polygone de contrôle

Raccordement G^0 Pour obtenir un raccordement G^0 entre macropatches par raccordement du polygone de contrôle (figure 4.15), il faut que les 4 patches mis en contact (ce sont des patches de bord, figure 4.19 b) soient G^0 deux à deux.

Or, contrairement aux macropatches G^0 où les coefficients étaient constants sur une face, ce n'est plus le cas pour un macropatch G^1 : les trois coefficients du bas (b_{6A} , b_A et b_{5A} , fléchés sur la figure 4.19 a), sont calculés à partir des normales aux points de contrôle du macropatch (b_5 et b_6). Pour obtenir l'égalité de ces coefficients, il faut donc l'égalité des produits scalaires (équation 4.15). Une condition suffisante pour cette égalité est l'égalité des normales aux extrémités des raccords (⑤ et ⑥). Pour le polygone de contrôle du macropatch, cette condition est assurée par la coplanarité des triangles mis en contact (figure 4.21). On le voit, cette condition met en jeu plus de points de contrôle qu'un raccord standard entre macropatches G^0 (figure 4.15), et que par conséquent moins de degrés de liberté sont disponibles pour contrôler la forme de la surface.

FIG. 4.21 - Raccordement G^0 de macropatches G^1 : coplanarité des triangles 1,2 et des triangles 3,4

Raccordement G^1 Pour un raccord G^1 , il faut que les patches quadriques mis en contact (les patches de bord) soient G^1 deux à deux. Cela n'est possible qu'avec une configuration géométrique équivalente au *transversal system* de Dahmen (figure 4.22) : les points $B1$, B , $B2$, $T1$, T et $T2$ doivent être coplanaires. Si l'on veut placer un macropatch $M2$ par rapport à un autre macropatch $M1$ dans cette configuration, il faut donc que le point $B2$ soit placé dans l'alignement du segment $[B1B]$ (figure 4.22 a). Pour le polygone de contrôle du macropatch cette condition s'énonce comme (figure 4.22 b) : le point de contrôle $F2$ doit être sur la droite passant par $F1$ et par B , le milieu

de $[AC]$. Le point $T2$ doit également se situer dans le même plan. Or $T2$ est le centre du triangle $D2, E2$, et T . Cela ajoute donc une condition sur les points de contrôle $D2$ et $E2$.

En résumé, le macropatch $M2$ se raccorde au macropatch $M1$ si ses points de contrôle $D2, E2, F2$ sont tels que :

1. les points $D2$ et $E2$ sont tels que :
 - les triangles ① et ②, et ③ et ④ sont coplanaires (condition G^0).
 - le centre du triangle $[D1, E1, T]$, le point T et le centre du triangle $[D2, E2, T]$ sont alignés.
2. le point $F2$ est sur la droite $(F1, B)$ où B est le milieu de $[A, C]$.

Il reste à fixer les coefficients pour avoir un raccord effectivement G^1 . En fait, la plupart des coefficients ont déjà une valeur correcte puisqu'ils sont fixés par les normales aux points A et C (équation 4.15). Il ne reste que le coefficient affecté au point de contrôle T : la valeur de celui-ci est libre d'après la définition de notre macropatch. Les contraintes G^1 vont supprimer ce degré de liberté : Le point T étant situé sur le segment $[T1, T2]$, il existe deux nombres l_1 et l_2 tels que $T = l_1 T1 + l_2 T2$. Toujours d'après Dahmen [Dahmen89], on a alors : $b_T = l_1 b_{T,T1} + l_2 b_{T,T2}$. Le calcul de b_T par cette équation donne finalement le raccord G^1 .

Il est évident que cette condition est très contraignante et ne permet sûrement pas le même liberté de construction qu'avec des surfaces de Bézier. Encore une fois, il s'agit d'une conséquence de la simplicité des quadriques par rapport aux surfaces de plus haut degré.

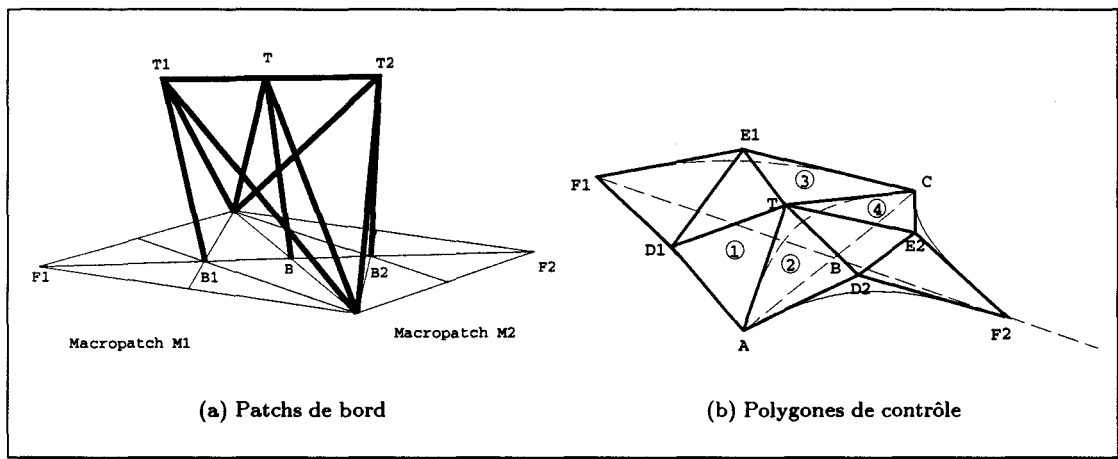


FIG. 4.22 - Raccordement G^1 de deux macropatches G^1

4.4.2 Bilan sur les macropatches

Nous avons construit une primitive à l'aide de quadriques obéissant aux objectifs fixés : la surface interpole correctement le polygone de contrôle et est par conséquent déformable de façon simple. Le macropatch est G^0 ou G^1 selon le nombre de quadriques utilisées.

Toutefois, les conditions de raccordement G^0 et G^1 entre macropatches sont très restrictives et ne laissent que peu de liberté pour les points de contrôle. Comme pour les autres méthodes à base

de macropatches (Dahmen et les *transversal systems*, Guo et la TPC), nous avons une limitation quant à la construction de surfaces quadriques complètement G^1 .

Tous les modèles de macropatches envisagés (figures 4.12, 4.16 et 4.18) sont définis à partir du même polygone de contrôle. Même s'ils n'ont pas les mêmes propriétés de continuité, il est possible, dans le contexte de la visualisation temps-réel, de les considérer comme plusieurs niveaux de description d'une même primitive [Bourdot91]. Le modèle affichable le plus rapidement (c'est-à-dire comportant le moins de quadriques) donnant un aspect de moindre qualité et celui comportant le plus de quadriques offrant le meilleur aspect. Entre les deux, trois modèles intermédiaires (figures 4.16 a, b et c) permettent plusieurs compromis.

Nous allons voir qu'en relâchant une condition a priori non essentielle (le fait d'avoir un nombre constant de quadriques par macropatch) et en adoptant un formalisme nouveau, nous pouvons parvenir aux surfaces G^1 avec une liberté totale des points de contrôle.

4.5 Surfaces quadriques définies par simplex B-splines

Les courbes B-spline, dont la majorité des développements furent introduits par C. De Boor, M. Cox et L. Mansfield [dB78] sont une généralisation des courbes de Bézier [Bezier86]. Alors qu'avec les splines définies par courbes de Bézier, on perd la liberté de position des points de contrôle dès que l'on désire assurer un ordre de continuité en chaque arc de la spline, les B-splines permettent la construction de courbes polynomiales par morceaux qui conservent le même ordre de continuité quelle que soit la position des points de contrôle.

On retrouve ce problème de continuité lors de la construction de quadriques par morceaux à l'aide de la représentation de Bernstein-Bézier : nous l'avons vu, la principale difficulté de la construction des macropatches provient du calcul des coefficients de contrôle pour assurer la continuité G^1 . Une représentation B-spline des quadriques permettrait, à l'instar des courbes, de construire des quadriques par morceaux à l'aide de points de contrôle complètement libres car indépendants des contraintes de continuité.

Les problèmes à résoudre pour atteindre ce but sont, premièrement, de trouver une généralisation des courbes B-splines aux simplex de dimension 3 (de la même façon que les 3-simplex de Bernstein-Bézier généralisent en dimension 4 les courbes de Bézier, cf. chapitre 3). Ensuite il est nécessaire, comme pour la représentation de Bernstein-Bézier, d'adapter ce formalisme aux surfaces algébriques et en particulier aux quadriques.

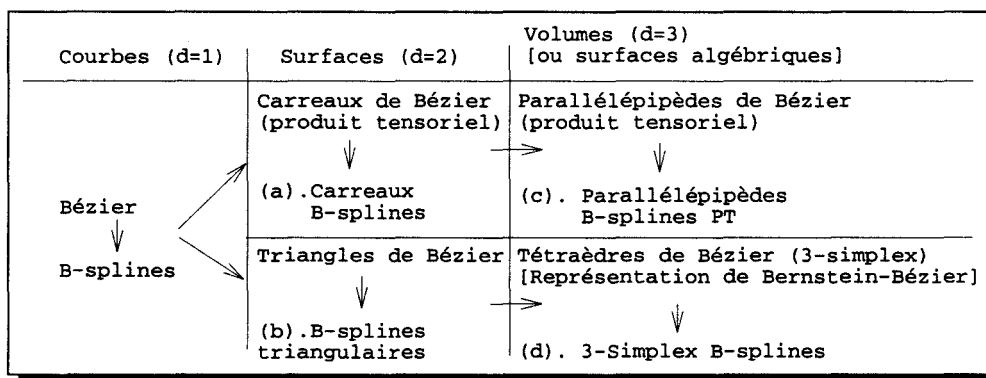


FIG. 4.23 - Classification des courbes, surfaces et volumes de Bézier et B-spline

Les courbes B-spline sont habituellement généralisées aux surfaces par produit tensoriel, comme

les carreaux de Bézier (figure 4.23 a). De récents développements (§ 4.5.2) ont également permis la définition de surfaces B-splines triangulaires (figure 4.23 b).

La définition de quadriques, et de façon plus générale de surfaces algébriques, se fait par la représentation de Bernstein-Bézier qui utilise des tétraèdres (ou 3-simplex) de Bézier (chapitre 3). Nous allons essayer de la même façon de définir les surfaces quadriques par morceaux à l'aide de volumes B-spline. Nous verrons que ceux-ci sont définis, comme les surfaces, soit par produit tensoriel, soit à l'aide de 3-simplex (figure 4.23 c et d).

4.5.1 Volumes B-splines par produit tensoriel

N. Patrikalakis et G. Kriezis ont étudié la définition des surfaces algébriques par volumes de Bézier par produit tensoriel [Patrikalakis et al.89]. L'équation implicite de la surface s'écrit :

$$f(x, y, z) = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^p w_{ijk} B_i^m(x) B_j^n(y) B_k^p(z) = 0 \quad (4.16)$$

La partie gauche de l'égalité est la généralisation de l'équation des carreaux de Bézier fonctionnels par produit tensoriel (équation 2.4, page 25) à une espace paramétrique de dimension 3. L'équivalent B-spline s'écrit :

$$f(x, y, z) = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^p w_{ijk} N_i^M(x) N_j^P(y) N_k^Q(z) = 0 \quad (4.17)$$

où les fonctions $N_i^M(x)$ sont les polynômes B-spline, qui sont une généralisation des polynômes de Bernstein aux polynômes par morceaux.

Une surface algébrique définie par l'équation 4.16 est de degré $m + n + p$. Mais toutes les surfaces de degré $m + n + p$ ne peuvent être représentées par l'équation 4.16. Pour cette raison, la représentation de Bézier par produit tensoriel ne s'applique pas aux quadriques de façon simple. En effet, pour obtenir des surfaces de degré 2 avec l'équation 4.16, deux possibilités existent :

- $m + n + p = 2$, ce qui implique qu'au moins une des valeurs est nulle, ce qui n'est pas valide puisque la surface doit être au minimum de degré 1 dans chaque dimension.
- $m + n + p > 2$, mais tous les termes de degré supérieur à 2 s'annulent. (i.e. $w_{ijk} = 0$ pour $i + j + k > 2$). Certains invariants sur les points de contrôle doivent alors être vérifiés pour conserver la nullité des ces termes. Par exemple, $m = n = p = 1$ donne l'équation algébrique:

$$\begin{aligned} &w_{0,0,0}(1-x)(1-y)(1-z) + w_{0,0,1}(1-x)(1-y)z + w_{0,1,0}(1-x)y(1-z) + \\ &w_{0,1,1}(1-x)yz + w_{1,0,0}x(1-y)(1-z) + w_{1,0,1}x(1-y)z + \\ &w_{1,1,0}xy(1-z) + w_{1,1,1}xyz = 0 \end{aligned} \quad (4.18)$$

ou

$$\begin{aligned}
 & w_{0,0,0} + \\
 & (-w_{0,0,0} + w_{1,0,0})x + \\
 & (w_{0,1,0} - w_{0,0,0})y + \\
 & (w_{0,0,1} - w_{0,0,0})z + \\
 & (w_{0,1,0} - w_{0,0,0} + w_{0,0,1} + w_{1,0,0} - w_{1,1,0} - w_{0,1,1} + w_{1,1,1} - w_{1,0,1})xyz + \\
 & (-w_{1,0,0} + w_{0,0,0} - w_{0,1,0} + w_{1,1,0})xy + \\
 & (w_{0,0,0} + w_{0,1,1} - w_{0,1,0} - w_{0,0,1})yz + \\
 & (-w_{0,0,1} + w_{1,0,1} + w_{0,0,0} - w_{1,0,0})xz = 0
 \end{aligned} \tag{4.19}$$

Cette équation représente une quadrique si et seulement si le terme en xyz est nul, c'est-à-dire :

$$w_{0,1,0} - w_{0,0,0} + w_{0,0,1} + w_{1,0,0} - w_{1,1,0} - w_{0,1,1} + w_{1,1,1} - w_{1,0,1} = 0 \tag{4.20}$$

Les problème de cette représentation, c'est qu'il faudra que cet invariant soit toujours vérifié, quelle que soit la valeur des w_{ijk} . Il n'y a donc pas de liberté totale des points de contrôle, ce qui nous ramène au problème des surfaces qui dégèrent en quadriques (§ 3.1.3). De plus, ce modèle ne permet pas la définition de toutes les quadriques. Pour y arriver, il faut augmenter le degré ($m = n = p = 2$) et par conséquent augmenter le nombre d'invariants (annulation des termes en x^2y , xyz^2 , etc).

La représentation B-spline qui découle des volumes de Bézier par produit tensoriel (équation 4.17) hérite des mêmes problèmes. Nous ne l'utiliserons donc pas pour représenter les quadriques. Par contre l'autre possibilité, les B-splines triangulaires (figure 4.23 d) offrent une solution que nous allons présenter maintenant.

4.5.2 Les simplex B-splines

Plusieurs modèles existent pour la construction de B-splines de dimension quelconque définies sur une triangulation arbitraire. Toutefois le modèle le plus abouti et qui bénéficie d'implémentations fonctionnelles est le modèle de B-splines triangulaires de Dahmen, Micchelli et Seidel [Dahmen et al.92, Seidel94, Greiner et al.94, Fong et al.93], basé sur la théorie des *simplex B-splines* [Dahmen et al.82]. C'est ce modèle que nous utilisons pour définir les surfaces quadriques par morceaux.

Comme pour les quadriques sous forme de Bernstein-Bézier, nous définissons les B-splines triangulaires *fonctionnelles*, en dimension 2 d'abord (i.e. de la forme $z = f(x, y)$).

Espace des paramètres Alors que pour un carreau B-spline, l'espace des paramètres est un rectangle de \mathbb{R}^2 , une 2-simplex B-spline est définie sur une triangulation de \mathbb{R}^2 (figure 4.24 a) :

$$T = \{\Delta(I) = [t_i, t_j, t_k] | I = (i, j, k) \in \mathbf{I} \subseteq \mathbb{N}^3\} \tag{4.21}$$

À chaque sommet t_i est associée une *séquence nodale* $t_{i,0}, t_{i,1}, \dots, t_{i,n}$ (figure 4.24 b), telle que $t_{i_0} = t_0$. n correspond au degré de la surface que l'on va définir.

Définition Si l'on considère l'ensemble de nœuds :

$$V_{\beta}^I = \{t_{i_0,0}, \dots, t_{i_0,\beta_0}, t_{i_1,0}, \dots, t_{i_1,\beta_1}, t_{i_2,0}, \dots, t_{i_2,\beta_2}\} \tag{4.22}$$

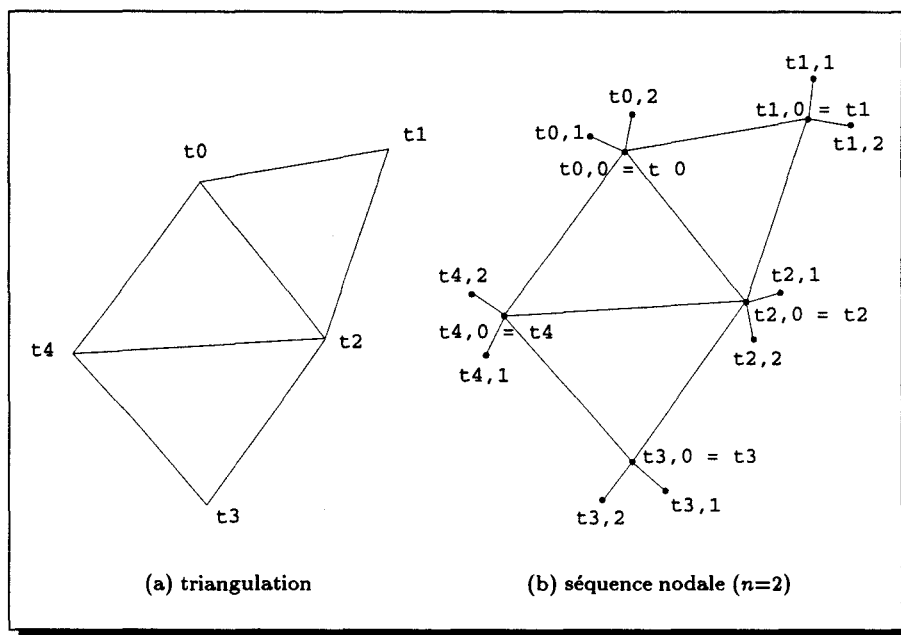


FIG. 4.24 - Exemple de triangulation $I = \{(0, 1, 2), (0, 2, 4), (2, 3, 4)\}$ avec séquence nodale

la fonction B-spline triangulaire $M_\beta^I(u) = M(u|V_\beta^I)$ est définie récursivement au point u sur V_β^I par :

$$M(u|v_0, v_1, v_2) = \frac{X_{[v_0, v_1, v_2]}(u)}{|d(v_0, v_1, v_2)|} \tag{4.23}$$

et

$$M(u|v_0 \cdots v_m) = \sum_{j=0}^2 \lambda_j(u) M(u|v_0 \cdots \hat{v}_j \cdots v_m) \tag{4.24}$$

où

- $X_{[v_0, v_1, v_2]}(u) = \begin{cases} 1 & \text{si } u \in [v_0, v_1, v_2[\\ 0 & \text{sinon} \end{cases}$
est la fonction caractéristique sur l'enveloppe convexe mi-ouverte $[v_0, v_1, v_2[$ (un ensemble mi-ouvert est l'équivalent dans \mathbb{R}^2 d'un intervalle ouvert à droite : les points sur la frontière droite (x maximal) n'appartiennent pas à l'ensemble).
- $d(v_0, v_1, v_2)$ est le double de l'aire du triangle (v_0, v_1, v_2)
- Lors de l'évaluation de l'équation 4.24, on choisit dans l'ensemble de nœuds $v_0 \cdots v_m$ un triplet $W_{i_0, i_1, i_2} = \{v_{i_0}, v_{i_1}, v_{i_2}\}$ de 3 points non alignés¹. $\lambda_j(u)$ représente la j ième coordonnée barycentrique de u par rapport à ces trois points.

1. Quelle que soit la façon de choisir les trois nœuds, on définit une base de l'espace des B-splines sur la triangulation I

- $v_0 \cdots \hat{v}_i \cdots v_m$ représente l'ensemble $v_0 \cdots v_m$ oté de l'élément v_i ,

À partir des équations 4.23 et 4.24 définissant les B-splines triangulaires, on définit les B-splines triangulaires *normalisées* (dans le but d'assurer l'invariance affine) :

$$N_{\beta}^I(u) = d(t_{i_0, \beta_0}, t_{i_1, \beta_1}, t_{i_2, \beta_2}) M_{\beta}^I(u) \quad (4.25)$$

Les fonctions N_{β}^I sont alors les fonctions d'influence (*blending functions*) des B-splines triangulaires. Comme pour les triangles de Bézier, les fonctions d'influence sont combinées avec les points de contrôle pour obtenir l'équation finale de la surface. Ici la sommation est faite d'abord sur toutes les fonctions d'influence, puis sur l'ensemble de la triangulation :

$$F(u) = \sum_I \sum_{|\beta|=n} c_{I, \beta} N_{\beta}^I(u) \quad (4.26)$$

Les points $c_{I, \beta}$ sont les points de contrôle de la surface, pour une facette I donnée, ils se répartissent de la même façon que pour les triangles de Bézier fonctionnels (figure 2.10, page 29).

Exemple

Si $n = 2$ (eq. 4.26), sur une triangulation donnée, pour chaque triangle I , l'équation 4.25 nous donne six fonctions N_{β}^I linéairement indépendantes qui constituent une base des polynômes de degré 2 sur I . On peut donc construire toute fonction polynomiale quadratique sur I comme une combinaison linéaire des six fonctions de base (équation 4.26).

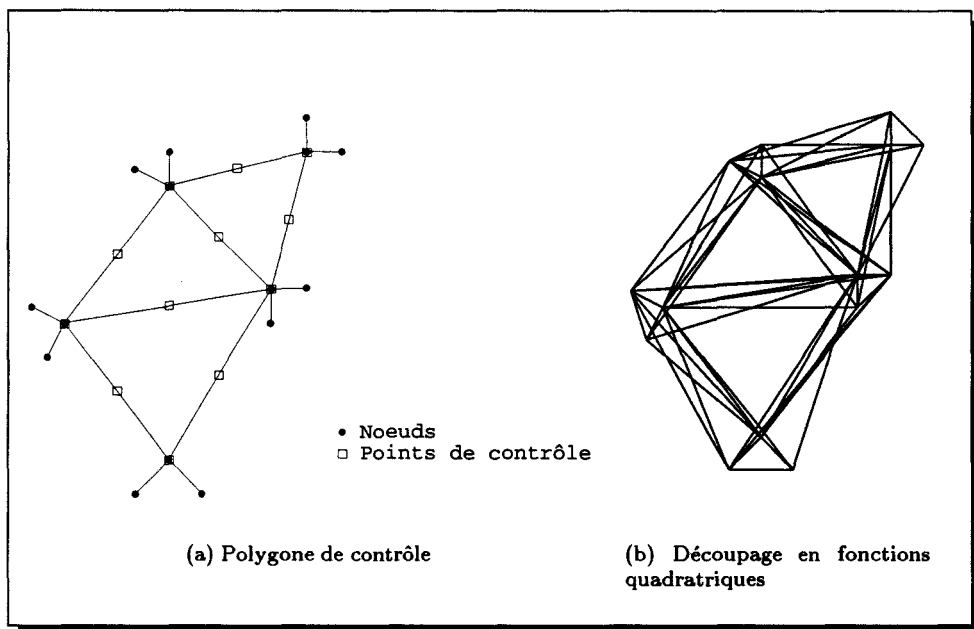


FIG. 4.25 - B-spline quadratique sur \mathbb{R}^2

La figure 4.26 montre un exemple de surface B-spline triangulaire fonctionnelle quadratique, définie sur une triangulation simple, et son polygone de contrôle.

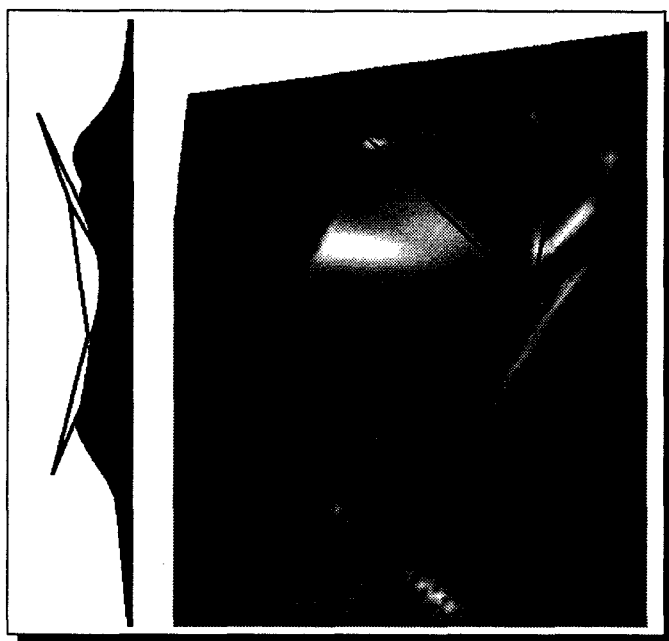


FIG. 4.26 - Exemple de B-spline triangulaire de degré 2 sur une triangulation de dimension 2

Propriétés

1. La surface a une continuité C^{n-1} dans le cas général de position des nœuds, quelle que soit la valeur des coefficients de contrôle $c_{I,\beta}$. La continuité peut être réduite le long d'un côté de la triangulation en alignant les nœuds le long de celui-ci.
2. Il est possible de représenter toute surface polynomiale par morceaux de degré n définie sur une triangulation T à l'aide d'une seule B-spline (équation 4.26).
3. Enveloppe convexe : la surface est entièrement contenue dans l'enveloppe convexe de l'ensemble des points t_{ij} , (les points de contrôle et les nœuds).
4. Contrôle local : la modification d'un point de contrôle n'affecte que la partie de la courbe la plus proche.

4.5.3 Application aux quadriques

Il est possible de créer des surfaces quadriques par morceaux à l'aide de simplex B-splines de deux manières : en dimension 2 (comme défini au paragraphe précédent) ou en dimension 3.

2-simplex

Les simplex B-spline de dimension 2 (figure 4.26) sont des quadriques par morceaux, puisque leur équation est fonctionnelle :

$$z = f(x, y) \quad \text{ou} \quad z - f(x, y) = 0 \quad (4.27)$$

où f est un polynôme par morceaux du second degré, défini par l'équation 4.26. Elles permettent de représenter les surfaces "en nappe" (i.e. un seul z par (x, y)) où la triangulation est dans le plan

xy et les valeurs $c_{I,\beta}$ représentent l'élévation des points de contrôle (figure 4.26). L'utilisation des simplex en dimension 2 est néanmoins limitée par le fait que la surface n'est pas invariante par rotation.

3-simplex

Pour générer des surfaces quadriques les plus générales possibles, il faut utiliser des B-splines triangulaires définies sur une triangulation de \mathbb{R}^3 . Nous avons défini les B-splines sur \mathbb{R}^2 , mais la construction se généralise à n'importe quelle dimension.

Dans \mathbb{R}^3 , la triangulation du plan devient une *tétraédration* : les nœuds sont des points de l'espace 3D et ils forment un ensemble de tétraèdres dans lesquels est défini le volume B-spline $w = f(x, y, z)$, par l'équation 4.26 généralisée. Comme pour la représentation de Bernstein-Bézier, les surfaces algébriques sont définies par l'ensemble des points tels que $f(x, y, z) = 0$. Contrairement à la représentation de Bernstein-Bézier, tous les points à l'extérieur de la tétraédration sont tels que $f(x, y, z) = 0$. Il est donc nécessaire limiter la surface algébrique à l'intérieur de celle-ci.

On obtient ensuite une surface quadrique en fixant simplement le degré à 2. Cette surface est composée de morceaux de quadriques reliés avec continuité G^1 ou G^0 le long d'une face de la tétraédration, selon que les nœuds issus des trois points de celle-ci sont coplanaires ou non.

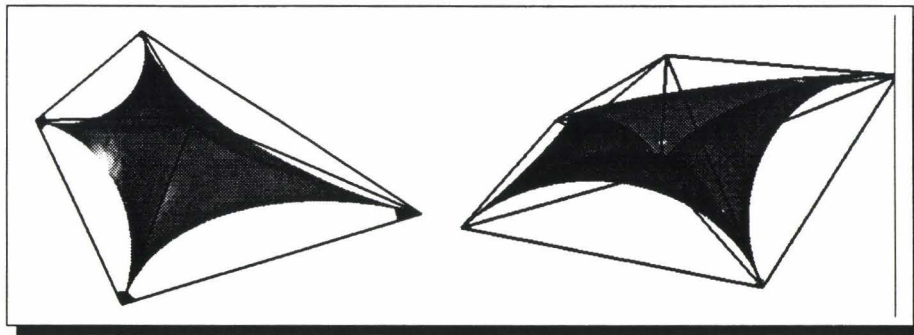


FIG. 4.27 - Simplex B-spline quadrique G^0 défini sur 4 tétraèdres (les nœuds sont confondus avec leur point de contrôle : $t_{i_0} = t_{i_1} = t_i$).

Propriétés

- Les quadriques définies par simplex B-splines héritent des propriétés des surfaces simplex splines : invariance affine, enveloppe convexe, contrôle local.
- Il est possible de représenter *toutes* les surfaces quadriques par morceaux avec une 3-simplex B-spline. La surface reste continue quelle que soit la valeur des points des contrôle.
- la continuité de la surface est contrôlée par la position des nœuds et peut donc être G^1 ou G^0 (figures 4.27 et 4.28).

Inconvénients

Même si ce formalisme décrit des surfaces quadriques par morceaux avec continuité indépendante des points de contrôle, notre objectif primaire, certaines caractéristiques peuvent être plus

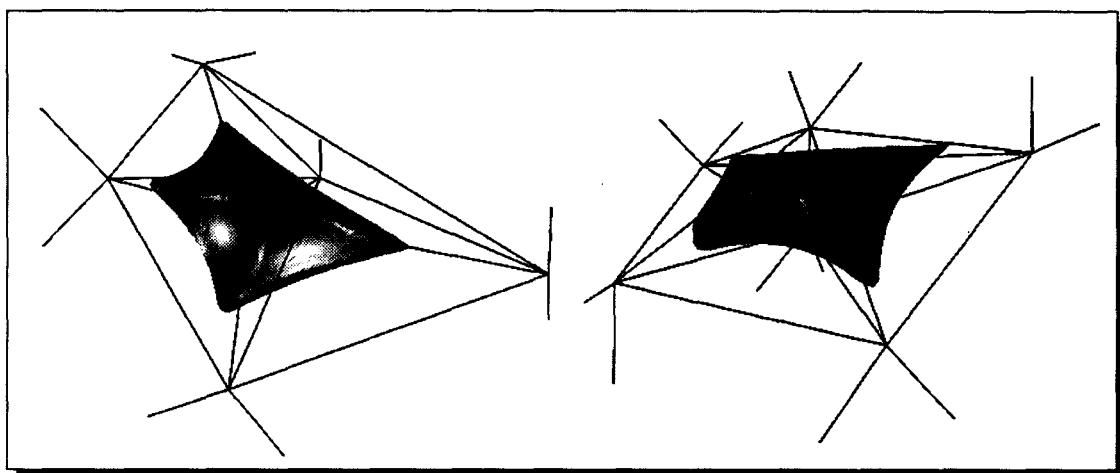


FIG. 4.28 - Simplex B-spline quadrique G^1 , obtenue à partir du simplex G^0 précédent en "tirant les nœuds"

génantes, en particulier dans un contexte de modélisation temps-réel :

- Nombre de quadriques par B-spline : comme on le voit sur l'exemple du 2-simplex (figure 4.25 b) le découpage de l'espace des paramètres produit un nombre beaucoup plus important de quadriques que pour d'autres surfaces simples comme les macropatches (à polygone de contrôle équivalent).
- Degrés de liberté : beaucoup de paramètres définissent la forme d'une surface quadrique définie par 3-simplex B-spline : les points de la tétraédisation, les nœuds et les coefficients de contrôle.
- Le découpage de la surface en quadriques et l'équation algébrique de chacune d'entre elles n'est pas exprimée explicitement dans la représentation (voir le paragraphe "conversion").

Construction de surfaces

Il est possible avec la représentation B-spline des quadriques de reproduire tous les schémas de reconstruction vus auparavant : tout patch ou macropatch, voire une surface complète utilisant la représentation de Bernstein-Bézier peut être représentée à l'aide d'une seule 3-simplex B-spline quadratique. En effet, les simplex G^0 (dont les nœuds sont confondus avec le point dont ils sont issus) peuvent être considérés comme un ensemble de patches de Bernstein-Bézier. Il suffit donc de définir une simplex B-spline G^0 avec comme tétraèdres ceux du macropatch à représenter. La seule différence avec un macropatch est que la fonction implicite est nulle en dehors du tétraèdre (d'après la définition de la fonction $X_{\{v_0, v_1, v_2\}}(u)$, équation 4.23).

Bien entendu, les configurations intéressantes sont obtenues quand les nœuds ne sont pas confondus avec leur point d'origine, ce qui permet d'obtenir une surface G^1 quelle que soit la position des points de contrôle. Dans ce cas, le nombre de quadriques est supérieur aux nombre de tétraèdres du polygone de contrôle (cf. le cas 2D figure 4.25), et dépend de la position des nœuds, puisque ceux-ci déterminent les intervalles 2D de la définition récursive de la B-spline.

Conformément à la définition des simplex B-splines, la continuité peut être réduite le long d'un côté des tétraèdres de contrôle, en alignant les nœuds appartenant à ce côté.

Conversion

La représentation B-spline des quadriques produit une quadrique par morceaux. Chaque "morceau" est un patch quadrique limité par un volume inclus dans la tétraèdrisation. Dans notre cas (conditions du chapitre 1), nous désirons obtenir l'équation algébrique cartésienne de chaque quadrique, ainsi que ses plans de coupe. La valeur de cette équation algébrique en tout point est donnée de façon récursive par la définition du simplex B-spline (équation 4.26), mais on n'a pas directement l'équation de chaque quadrique sous la forme polynomiale, et encore moins les équations de leur volume limite.

Dans un premier temps il faut donc calculer tous les intervalles 3D (volumes) dans lesquels il n'y a qu'un patch quadrique. Ce problème n'est pas simple car la subdivision en patches est implicite dans les équations des B-splines triangulaires. Un exemple sur un 2-simplex est montré figure 4.25. Chaque subdivision du plan (figure 4.25 b) contient une fonction quadratique. Le problème consiste à obtenir la liste de tous ces polygones, à partir des segments générés par les B-splines (ces segments sont issus de l'équation 4.24 dans laquelle on choisit les triplets de points non alignés).

Ce genre de problèmes est bien connu en géométrie algorithmique. Une solution consiste à calculer une triangulation *contrainte* 3D [Hazlewood93] : il s'agit de calculer la triangulation des nœuds de la B-spline qui incorpore les segments générés par l'algorithme B-spline.

Nous présentons néanmoins une méthode plus simple, mieux adaptée à notre problème. Elle consiste, à partir de la triangulation de départ (le polygone de contrôle), à ajouter chaque nouveau triangle (généralisé par l'équation 4.24) en calculant son intersection avec la triangulation courante. Cette méthode nécessite l'utilisation de trois algorithmes "classiques" en algorithmique géométrique :

- calculer l'intersection de deux tétraèdres : l'algorithme de clipping de Sutherland-Hodgman (clipper le tetra T1 successivement par chaque facette de T2) effectue cette opération en temps constant
- calculer la différence booléenne entre deux tétraèdres. Se fait de la même façon, face par face.
- tétraèdriser un polyèdre quelconque, éventuellement concave.

Bien que tous les volumes ne contenant qu'un seul patch quadrique ne soient pas des tétraèdres, cet algorithme les redécoupe en tétraèdres. Cela simplifie le calcul d'intersections (on n'a que des intersections tétraèdre-tétraèdre) malgré une augmentation du nombre de quadriques générées.

La deuxième étape consiste ensuite à calculer l'équation de la quadrique dans chaque tétraèdre élémentaire. Il suffit pour cela de développer l'équation de la B-spline pour un point du tétraèdre. Tous les termes correspondant aux triangles qui ne contiennent pas le point considéré s'annulant (fonction X de l'équation 4.23), il reste la fonction quadratique dont les variables sont les coordonnées barycentriques du point par rapport aux triangles le contenant. Ces coordonnées se convertissent simplement en coordonnées cartésiennes (équation 2.6, page 28). On obtient finalement l'équation algébrique de la quadrique.

Calcul accéléré lors de la modification des coefficients

Deux attributs permettent de modifier la forme d'une B-spline quadrique : la géométrie (la triangulation T), et les coefficients de contrôle ($c_{I,\beta}$, cf. équation 4.26). Or le calcul des fonctions

B-spline ne dépend que de la géométrie et pas de la valeur des coefficients (équations 4.23, 4.24 et 4.25). À condition de précalculer les fonctions $N_{\beta}^I(u)$ et de ne modifier que les coefficients de contrôle, la modification de la surface peut se faire rapidement puisque le calcul n'est modifié qu'au niveau de l'équation 4.26, et l'on évite ainsi tout le processus récursif du calcul de la fonction $M_{\beta}^I(u)$ (équation 4.24).

4.6 Bilan

Nous avons vu que la construction de surfaces complexes à l'aide de quadriques sous forme de Bernstein-Bézier souffrait du défaut "classique" de la construction de courbes ou de surfaces de Bézier par morceaux : les contraintes de continuité limitent la liberté des points de contrôle et vice-versa.

Si l'on se limite à un ordre de continuité G^0 , le problème de la construction de surface reste simple, même si les possibilités de modélisation sont réduites. La construction de surfaces G^1 est simple dans certains cas particuliers comme les surfaces de révolution, mais, dans le cas général, le problème ne possède pas à l'heure actuelle de solution satisfaisante. Toutes les méthodes existantes possèdent chacune un défaut quant au contrôle de la surface.

Dans le but de pallier ce problème, nous avons défini une représentation des quadriques par un schéma B-spline dans le but de rendre indépendants la continuité de la surface et la position des points de contrôle. Ainsi la forme de la surface peut être modifiée sans contrainte de continuité.

L'exploitation complète de cette définition des quadriques pour la synthèse d'image passe par deux voies : soit il est nécessaire de posséder des algorithmes de conversion de la forme B-spline vers la forme algébrique quadratique (ce qui peut être le cas si l'on dispose d'un affichage câblé, cf. chapitre 1). La seconde voie consiste en la réalisation d'algorithmes de rendu spécifiques à la forme B-spline, en particulier des routines d'intersection droite/quadrique pour le lancer de rayons ou le rendu projectif. Dans les deux cas, la complexité des algorithmes proviendra non pas de l'intersection même avec la surface (puisque le degré est limité) mais plutôt des volumes de clipping de chaque patch (par exemple, déterminer quelle quadrique est intersectée).

Au niveau de la modélisation, le nombre important de degrés de liberté de la définition (points de contrôle, coefficients et nœuds) montre qu'une utilisation directe de B-splines est difficile. Nous prévoyons qu'une application utilisant notre modèle de surface passera probablement par un modèle de plus haut-niveau, plus simple pour l'utilisateur.

Un grand nombre de ces modèles de haut-niveau (comme l'interpolation) utilisant la définition standard des simplex B-splines (c'est-à-dire les 2-simplex paramétriques) existe [Greiner et al.94, Fong et al.93, Auerbach et al.91, Pfeiffe et al.94] et il est souvent possible de les extrapoler aux 3-simplex algébriques et donc aux quadriques.

Chapitre 5

Approximation

Les méthodes présentées au chapitre précédent considèrent un cadre dans lequel les surfaces quadriques étaient à la fois primitives de modélisation et primitives de visualisation. Toutefois, ce cadre concerne un nombre limité d'applications puisqu'il existe un grand nombre de primitives de modélisation différentes, chacune répondant à des problèmes géométriques plus ou moins spécifiques. En bref, la plupart des applications n'utilisent pas de quadriques à la modélisation, mais plutôt des surfaces paramétriques ou implicites de degré élevé. Il faut donc prévoir l'affichage à l'aide de quadriques de ces surfaces, quelle que soit leur définition.

La conséquence d'un choix différent pour les primitives de modélisation et de visualisation est la nécessité de disposer d'algorithmes de conversion de l'une vers l'autre. Dans le cas classique de l'affichage des facettes, les primitives de modélisation de toutes sortes sont *facettisées*, c'est-à-dire converties en un ensemble de facettes qui approchent le mieux possible la surface d'origine. Le rendu projectif avec lissage de Gouraud (§ 1.2.1) permet de reproduire à l'affichage l'aspect "arrondi" de la surface d'origine.

À partir du moment où l'on dispose d'un accélérateur qui affiche des surfaces quadriques, on a sans aucun doute beaucoup à gagner en "quadrisant"¹ la surface plutôt qu'en la facettisant : la courbure naturelle des quadriques doit permettre de représenter les surfaces en utilisant moins de primitives, tout en donnant un aspect meilleur au modèle affiché.

Nous précisons dans ce chapitre les difficultés du problème avant de présenter certaines solutions, en fonction du type de surface à quadriser. Nous proposons en particulier un algorithme de réduction de degré permettant d'approximer toute surface algébrique par une quadrique par morceaux.

1. Nous utilisons dans la suite ce néologisme pour désigner l'approximation par une surface quadrique par morceaux

5.1 Problématique

Si nous reprenons le diagramme de comparaison des pipelines facettes et quadriques (figure 5.1), nous nous situons à présent au niveau de la flèche 2, alors que le chapitre précédent concernait la flèche 1 (le pipeline facettes est représenté à gauche).

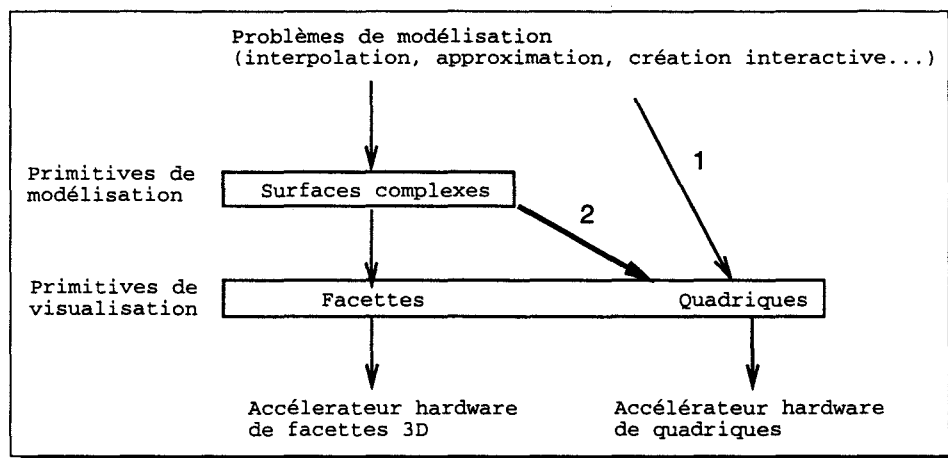


FIG. 5.1 - Pipelines facette et quadrique

L'approche quadrisation, proche du principe de la facettisation, va donc en hériter certaines propriétés :

- Diversité des primitives de modélisation: comme pour la facettisation, la quadrisation a évidemment l'avantage de la versatilité sur la modélisation directe à l'aide de quadriques, puisque le choix de la primitive de modélisation est libre. Il faut bien entendu qu'il existe un algorithme adapté à chaque type de primitive de modélisation utilisé. Or les différences fondamentales de définition de surfaces (implicites, paramétriques, etc., cf. chapitre 1) n'autorisent, à notre connaissance, pas de méthode uniforme de quadrisation (à l'instar des algorithmes de facettisation, qui sont fondamentalement différents selon les primitives de modélisation [Farin92, Bloomenthal88, Velho90, Lorensen et al.87]).
- Contraintes d'approximation: d'un point de vue mathématique, la conversion est essentiellement un problème d'approximation. Toutefois la plupart des solutions antérieures, ou proposées dans un cadre purement mathématique (par exemple [Chuan et al.89]), ne tiennent pas compte d'éléments essentiels pour nous comme l'approximation par quadriques par morceaux, et par conséquent la continuité, le nombre de quadriques générées, etc.

Comme tout processus d'approximation, le problème de la précision et de la mesure d'erreur est ici important et conditionne une variable essentielle pour nous : le nombre de quadriques constituant l'approximation. Celui-ci est également influencé par la continuité désirée du modèle: G^0 ou G^1 .

- Séparation modélisation/visualisation: un troisième aspect important à l'approximation par quadriques est que la distinction entre primitives de modélisation et primitives de visualisation, entraînant l'utilisation d'algorithmes de conversion, supprime partiellement l'aspect temps-réel du problème: la conversion est un processus complexe qui ne peut se faire 25 fois par seconde pour des scènes complexes. Par conséquent, seule la visualisation de scènes rigides et immobiles pourra se faire de façon rapide. Toute modification entraîne un recalcul du modèle de visualisation: on rejoint également sur ce point les modèles à facettes.

5.1.1 État de l'art : la facettisation

Tous les algorithmes de conversion de primitives complexes vers un ensemble de primitives simples utilisent deux processus distincts : l'échantillonnage et la reconstruction. L'échantillonnage consiste dans un premier temps à obtenir des informations sur la surface à approximer, sous la forme de données locales (par exemple, des points). La reconstruction produit le modèle final à partir des échantillons.

Les algorithmes utilisent ces processus de deux manières : ceux-ci sont effectués soit de façon complètement séparée, l'un après l'autre pour toute la surface, soit primitive par primitive. Dans le premier cas, la surface à approcher est totalement échantillonnée, puis la reconstruction se fait. Dans le second, l'approximation est construite par une série d'opérations échantillonnage/reconstruction de proche en proche le long de la surface d'origine ; à chaque opération est produite une primitive de la surface approchante.

La facettisation (ou approximation linéaire) est évidemment la méthode de conversion modélisation-visualisation la plus répandue en synthèse d'images. Nous donnons ici les grandes lignes des algorithmes les plus courants :

- Approximation des surfaces paramétriques : la facettisation s'obtient par subdivision de l'espace des paramètres. Un parcours en deux dimensions² à intervalles réguliers de celui-ci produit un ensemble de points 3D qu'il suffit de relier pour obtenir la facettisation (figure 5.2).

Il existe des algorithmes plus spécifiques par rapport au type de surface. En particulier l'algorithme de De Casteljau pour les surfaces de Bézier (chapitre 2), ou l'algorithme de Cox-de Boor pour les B-splines ([Farin92], page 165) consistent en une subdivision récursive de la surface, jusqu'à ce que les surfaces obtenues soient suffisamment petites pour être approchées par les facettes de leur polygone de contrôle (§ 2.1.4).

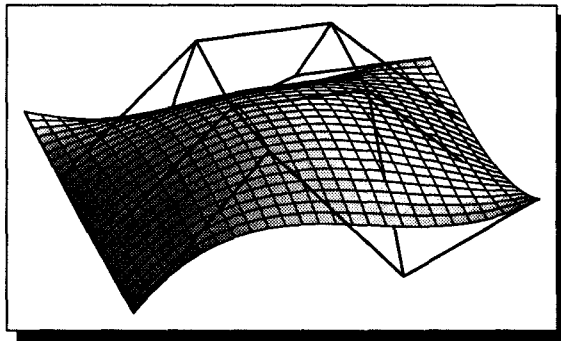


FIG. 5.2 - Carreau de Bézier facettisé par subdivision régulière de l'intervalle $[0, 1]^2$.

- Approximation des surfaces implicites : en l'absence d'espace des paramètres et de coordonnées de surfaces, les algorithmes de facettisation utilisent presque exclusivement des techniques de subdivision spatiale [Bloomenthal88, Ning et al.93]. Inspirés par l'algorithme des "marching-cubes" (utilisé pour représenter des iso-surfaces à partir de données scanner ou IRM [Lorensen et al.87]), le principe de ces algorithmes est la subdivision d'un volume englobant la surface à facettiser en volumes élémentaires (en général des cubes). Pour chacun de ces volumes, l'algorithme vérifie si la surface intersecte celui-ci en calculant la valeur de la fonction implicite en chacun de ses sommets. S'il y a effectivement intersection, celle-ci

2. l'espace des paramètres appartient à un plan (cf. figure 1.5, page 8).

est approximée par une ou deux facettes à l'intérieur du volume. La réunion de toutes les facettes générées par ce processus donne l'approximation de la surface (figure 5.3). Celle-ci est d'autant meilleure que la grille est fine.

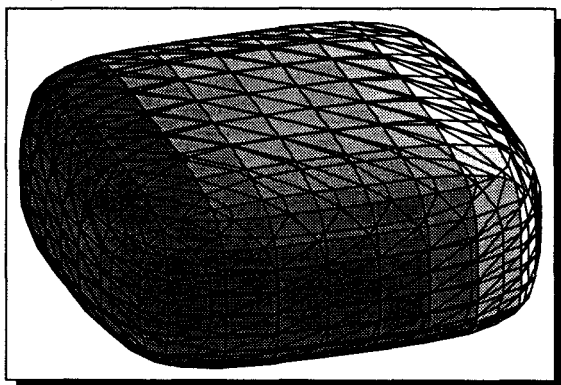


FIG. 5.3 - Surface d'équation $x^4 + y^4 + z^4 - 1 = 0$ facettisée par subdivision spatiale du cube $[0, 1]^3$

Certaines méthodes plus rares effectuent l'échantillonnage et la reconstruction primitive par primitive: à partir d'une facette approchant un morceau de la surface, d'autres facettes voisines sont calculées jusqu'à recouvrir toute la surface [Allgower et al.85]. Certaines approches récentes effectuent ce processus de façon dynamique: les facettes de l'approximation se déplacent sur la surface, dans le but de donner une idée de la forme avec peu de facettes [Witkin et al.94].

Il est raisonnable de penser qu'un algorithme d'approximation à l'aide de quadriques se base sur les mêmes principes que la facettisation: échantillonner la surface de départ et reconstruire à l'aide des algorithmes d'interpolations vus précédemment (chapitre 4) pour la reconstruction.

Nous allons donc tenter d'adapter les schémas des facettes aux quadriques. Nous allons d'abord évoquer les méthodes issues des algorithmes d'interpolation du chapitre précédent, puis nous étudions en détail un algorithme d'approximation par subdivision spatiale.

5.2 Quadratisation par interpolation

La première méthode provient des algorithmes d'interpolation du chapitre précédent. À partir de points de la surface à approximer, les algorithmes de Pratt, Bajaj, Dahmen, Guo, ou les macropatchs que nous avons définis permettent de calculer une surface quadrique par morceaux interpolant ces points.

Pour une interpolation G^0 , le problème est trivial: il suffit de placer un macropatch G^0 pour chaque facette de la triangulation.

Si l'on désire une surface interpolante G^1 , les algorithmes cités plus haut sont utilisables, mais sous certaines réserves quant aux points d'interpolation, que nous précisons ici.

5.2.1 Échantillonnage de surfaces

Le premier problème de la méthode est donc de calculer les points d'interpolation à partir de la surface à approximer. Trois critères importants existent pour calculer ceux-ci: la topologie des

points, leur densité et d'autres facteurs géométriques liés à l'algorithme d'interpolation utilisé.

Topologie

Dans tous les algorithmes de reconstruction, il est préférable de posséder une information topologique sur la surface à reconstruire, plutôt que les points uniquement. En effet, un nuage de points appartenant à une surface ne donne pas la forme de celle-ci, et une reconstruction fautive peut en découler (figure 5.4).

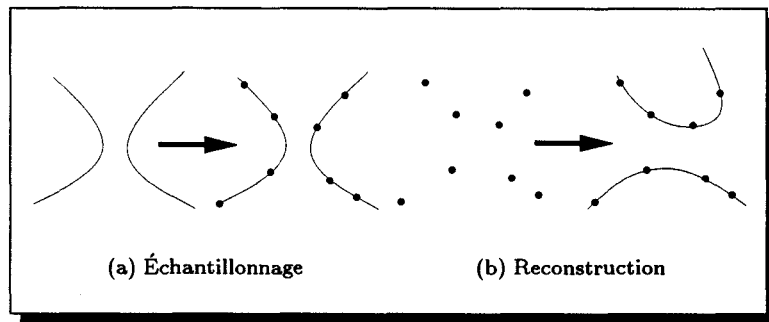


FIG. 5.4 - Échantillonnage équivoque

Il est possible d'éviter ce phénomène en générant une information topologique avec les points d'interpolation. Autrement dit, fournir une triangulation à l'algorithme de reconstruction. Cela revient à faire une facettisation comme première étape à l'interpolation quadratique.

Densité

Par rapport à une facettisation, il n'est pas nécessaire d'échantillonner finement pour avoir une approximation correcte. Un échantillonnage non adaptatif peut s'avérer correct même dans les régions où la courbure de la surface originale change beaucoup. Toutefois, s'il existe des principes d'échantillonnage plus ou moins adaptés à la reconstruction quadratique, ceux-ci sont probablement moins simples que les échantillonnages linéaires ou la densité d'échantillons est directement fonction de la courbure originale.

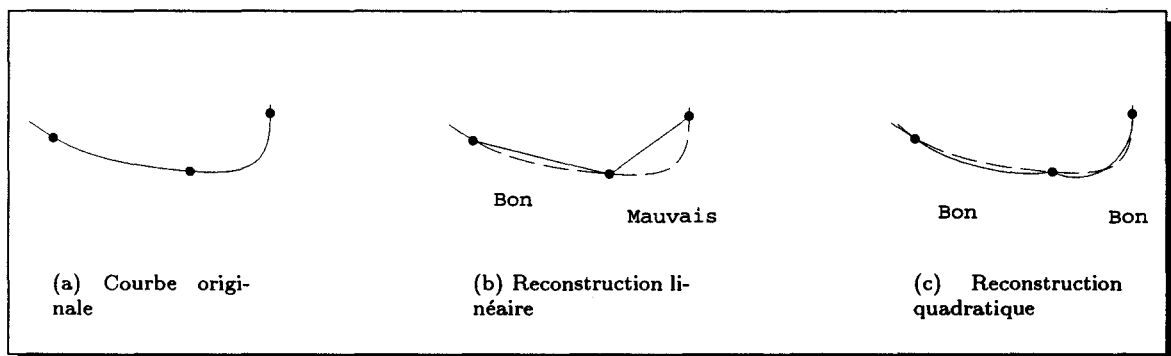


FIG. 5.5 - Densité de l'échantillonnage

D'une façon générale la triangulation de la surface à calculer avant d'interpoler n'est pas

nécessairement aussi fine que pour une facettisation. Par conséquent, un nombre moins important de quadriques est utilisé pour approximer un objet.

Géométrie de la triangulation en fonction de l'algorithme d'interpolation

C'est ici que l'on rencontre le plus de problèmes. Rappelons que les algorithmes d'interpolation du chapitre 4 avaient des exigences fortes quant à la géométrie de la triangulation : l'algorithme de Dahmen nécessite la construction du *transversal system* que l'on ne sait pas faire dans tous les cas de triangulations. L'algorithme de Guo impose, lui, une condition sur les normales aux points d'interpolation. Bien qu'ils soit possible de calculer les normales pour répondre à la condition de Guo, celles-ci risquent d'être assez différentes de celles de la surface d'origine. Le problème revient avec notre modèle de macropatch qui même s'il génère une surface interpolant les points ne garantit pas les normales à cause des conditions de continuité (§ 4.4).

Nous concluons que les algorithmes de reconstruction par interpolation, même s'il permettent à tous les types de surfaces d'être approximés, sont difficilement applicables et complexes à mettre en œuvre pour une surface G^1 . Aussi nous nous tournons vers une seconde possibilité : la subdivision spatiale.

5.3 Subdivision spatiale et réduction de degré

Nous envisageons donc maintenant l'autre possibilité évoquée précédemment, que nous traitons en détail : la subdivision spatiale. Nous proposons une méthode permettant d'approcher une surface quelconque exprimée de façon algébrique par une surface quadrique par morceaux.

L'algorithme que nous présentons est basée sur une méthode de *réduction de degré*, qui permet d'approximer une surface de degré quelconque par une autre surface de degré inférieur, donc pas obligatoirement quadratique.

La méthode consiste dans un premier temps à subdiviser spatialement la surface à approximer. Sur chaque morceau de surface résultant, il suffit d'appliquer un algorithme de réduction de degré sur le 3-simplex de Bézier au-dessus, ce qui donne donc un morceau de surface de degré moindre. La réunion de ces morceaux donne l'approximation.

5.3.1 Les algorithmes de réduction de degré pour les simplex de Bézier

La réduction de degré est quelque chose de bien connu dans le cas des courbes de Bézier [Farin92, Eck93, Petersen84, Lodha et al.94, Watkins et al.88]. La méthode la plus simple consiste à inverser les formules d'*élévation de degré* ([Farin92], p. 68) : l'élévation de degré d'une courbe de degré n produit la même courbe, mais décrite par une équation de degré $n + 1$. Elle possède donc un point de contrôle en plus, ce qui permet donc un contrôle plus fin de la courbe. Le problème de l'élévation de degré est donc de calculer la position des $n + 2$ nouveaux points de contrôle à partir des $n + 1$ points d'origine.

Soit une courbe de Bézier de degré n décrite par les $n + 1$ points de contrôle \hat{b}_i . Cette courbe est également définie comme une courbe de Bézier de degré $n + 1$ par les $n + 2$ points de contrôle b_i , tels que :

$$b_i = \frac{i}{n} \hat{b}_{i-1} + \frac{n-i}{n} \hat{b}_i \quad i = 0, 1, \dots, n \quad (5.1)$$

Inverser cette équation consiste à exprimer les points \hat{b}_i en fonction des points b_i . Le calcul donne deux formules recursives :

$$\hat{b}_i = \frac{nb_i - i\hat{b}_{i-1}}{n-i} \quad i = 0, 1, \dots, n-1 \quad (5.2)$$

et

$$\hat{b}_{i-1} = \frac{nb_i - (n-i)\hat{b}_i}{i} \quad i = n, n-1, \dots, 1 \quad (5.3)$$

Ces deux équations donnent deux courbes de degré $n-1$ qui approximent, chacune à partir d'une extrémité, la courbe originale de degré n (figure 5.6 a et b). Ces approximations sont assez grossières mais le résultat s'améliore sensiblement en les combinant : la courbe finale est obtenue en prenant les $n/2$ premiers points de contrôle de l'approximation de gauche, et les $n/2$ derniers de l'approximation de droite. Si le nombre de points de contrôle est impair, le milieu des deux points centraux est conservé (cf. figure 5.6 c pour le cas quadratique).

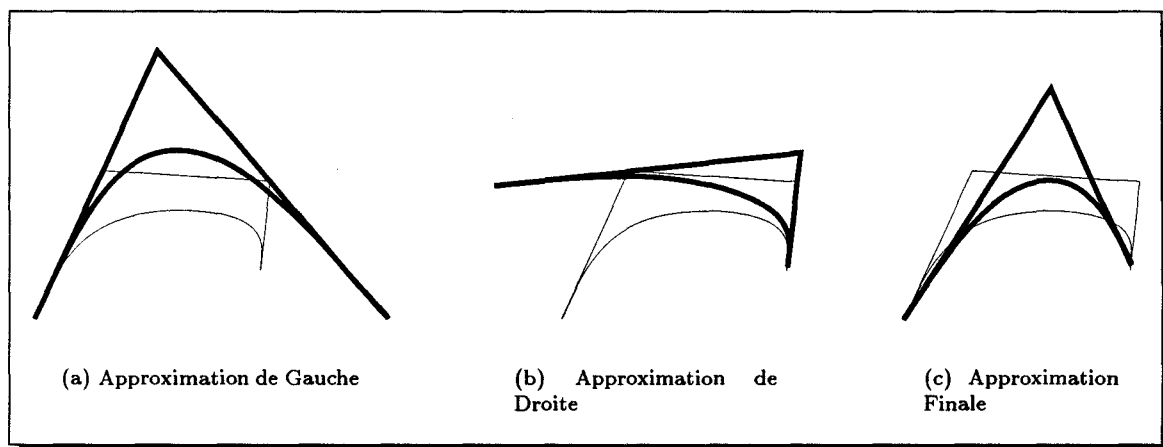


FIG. 5.6 - Réduction de degré simple

Ce résultat se généralise aux dimensions supérieures : C. Petersen étend directement les équations 5.2 et 5.3 aux triangles de Bézier [Petersen84] :

$$\begin{aligned} b_{i-1,j,k}^1 &= \frac{nb_{i,j,k} - j b_{i,j-1,k}^1 - k b_{i,j,k-1}^1}{i} & i \neq 1 \\ b_{i,j-1,k}^2 &= \frac{nb_{i,j,k} - i b_{i-1,j,k}^2 - k b_{i,j,k-1}^2}{j} & j \neq 1 \\ b_{i,j,k-1}^3 &= \frac{nb_{i,j,k} - i b_{i-1,j,k}^3 - j b_{i,j-1,k}^3}{k} & k \neq 1 \end{aligned} \quad (5.4)$$

Ces trois équations décrivent trois approximations de degré $n-1$. Chacune d'entre elles interpole un sommet du triangle original. Il suffit finalement de faire la "moyenne" de ces trois surfaces (chaque point de contrôle est calculé par une combinaison pondérée des points des trois approximations) pour obtenir une approximation correcte. Bien que Petersen ne l'ait pas suggéré, ces équations se généralisent aux simplexes de dimension quelconque.

Lodha et Warren [Lodha et al.90] proposent un autre algorithme pour les simplex de Bézier de dimension quelconque. Celui-ci permet également de réduire le degré directement de n à e directement. Ils démontrent également que les extrémités de la surface de degré n sont interpolées par la surface de degré e jusqu'à l'ordre $\lfloor \frac{e-1}{2} \rfloor$ (où le symbole $\lfloor \rfloor$ désigne la fonction partie entière), et que leur algorithme atteint l'ordre optimal d'approximation: $O(h^{e+1})$, où h est le diamètre du simplex. L'algorithme calcule, à partir du simplex de Bézier de départ, un simplex *multinomial* de degré n , *tronqué* à chaque sommet. Un polynôme multinomial est un polynôme de Bernstein en coordonnées barycentriques auquel il "manque" une coordonnée³ (la base cartésienne est une base multinomiale particulière). *Tronqué* signifie que seules les e premières rangées de coefficients du polynôme à partir du sommet du simplex sont calculées puisque les autres ne sont pas utilisées. Ensuite, un simplex de degré e est déduit de chaque simplex tronqué en interpolant les e premières dérivées de ceux-ci. Finalement, ces simplex sont exprimées dans la base de Bézier, puis assemblés par une combinaison convexe pour former l'approximation finale (figure 5.7).

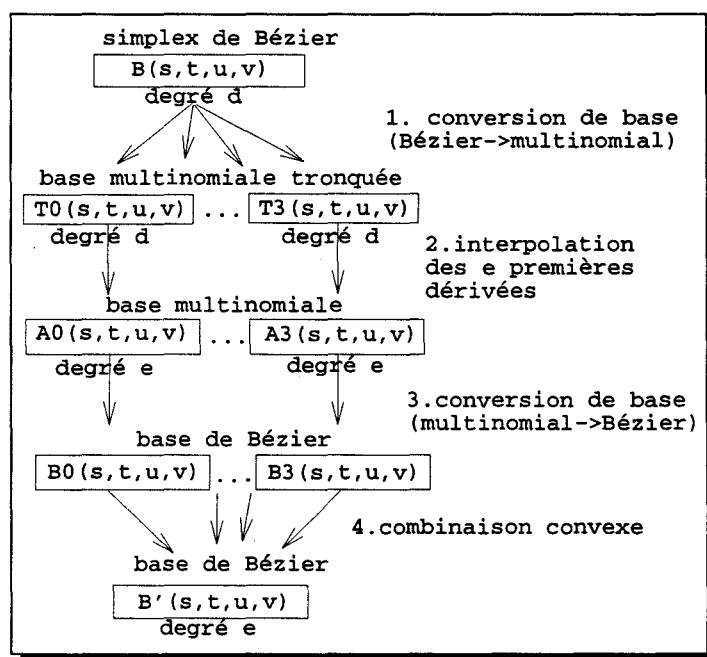


FIG. 5.7 - Description de l'algorithme de Lodha-Warren

Nous avons vérifié que l'algorithme de Lodha-Warren et celui de Petersen itéré donnent les mêmes formules symboliques de réduction de degré quelle que soit la dimension, et pour tout degré jusqu'à 10 (nous conjecturons d'ailleurs l'égalité dans tous les cas). Par exemple, la réduction d'un 3-simplex du degré 4 (de coefficients $B_{i,j,k}^4$) au degré 2 (coefficients $A_{i,j,k}^2$), donne avec les deux algorithmes :

3. Il est toujours possible de substituer une coordonnée d'un polynôme barycentriques par 1 moins la somme des autres coordonnées, puisque la somme de toutes les coordonnées barycentriques est 1. Par exemple :

$$\begin{aligned} a\lambda_1 + b\lambda_2 + c\lambda_3 &= \\ a\lambda_1 + b(1 - \lambda_1 - \lambda_3) + c\lambda_3 &= \\ (a - b)\lambda_1 + (c - b)\lambda_3 + b & \end{aligned}$$

$$\begin{aligned}
A_{0,0,2}^2 &= B_{0,0,4}^4 \\
A_{0,1,1}^2 &= -1/2B_{0,4,0}^4 + B_{0,3,1}^4 - 1/2B_{0,0,4}^4 + B_{0,1,3}^4 \\
A_{0,2,0}^2 &= B_{0,4,0}^4 \\
A_{1,0,1}^2 &= -1/2B_{4,0,0}^4 + B_{3,0,1}^4 - 1/2B_{0,0,4}^4 + B_{1,0,3}^4 \\
A_{1,1,0}^2 &= -1/2B_{4,0,0}^4 + B_{3,1,0}^4 - 1/2B_{0,4,0}^4 + B_{1,3,0}^4 \\
A_{2,0,0}^2 &= B_{4,0,0}^4
\end{aligned} \tag{5.5}$$

Notons que cette expression de la réduction de degré où le degré et la dimension sont fixés (ce qui est vrai dans la plupart des cas) est évidemment plus simple et rapide que l'utilisation de l'algorithme général. On aura donc intérêt, dans ce cas, à calculer les formules symboliques avant tout.

D'autres algorithmes de réduction de degré existent, par exemple de Watkins et Worsey [Watkins et al.88], ou Dannenberg et Nowacki [Dannenberg et al.85], mais ceux-ci sont limités soit aux courbes de Bézier, soit aux surfaces par produit tensoriel.

5.3.2 Réduction de degré appliquée aux surfaces algébriques

La représentation de Bernstein-Bézier des surfaces algébriques exprime ces surfaces à l'aide d'un 4-simplex de Bézier (chapitre 2). Il est donc a priori possible d'utiliser les algorithmes de réduction de degré vus précédemment. Il suffit d'appliquer l'algorithme sur le 3-simplex au-dessus. En général, les surfaces algébriques étant exprimées dans la base cartésienne (c'est-à-dire sous la forme $f(x, y, z) = 0$), il est nécessaire d'ajouter au début et à la fin de l'algorithme une conversion de base pour passer dans la base de Bernstein et vice-versa (figure 5.8).

Les étapes 0a, 0b et 1 (figure 5.8) peuvent se simplifier en un seul changement de base multinomiale.

Propriétés

La plupart des propriétés de la réduction de degré de Lodha-Warren, énoncées dans le cas des 3-simplex, restent vraies :

- Symétrie : l'algorithme est symétrique par rapport aux sommets du tétraèdre.
- Restriction aux frontières : les courbes frontières du patch algébrique réduit sont la réduction des limites du patch algébrique de départ. Cette propriété a pour conséquence la continuité minimale G^0 des surfaces produites par la réduction de degré par morceaux (5.3.3).
- Invariance par transformation affine, les simplex de Bézier étant invariants puisque exprimés en coordonnées barycentriques.
- Précision : réduire le degré d'une surface dont le degré a été élevé auparavant reproduit la surface d'origine.
- Linéarité : la réduction d'une somme de surfaces algébriques est la somme des réductions des surfaces individuelles.

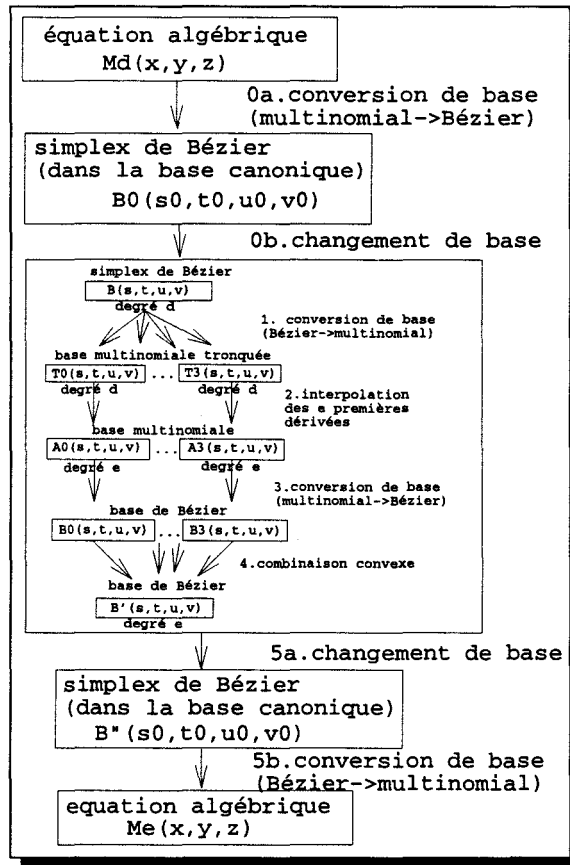


FIG. 5.8 - L'algorithme de Lodha-Warren appliqué aux surfaces algébriques

Par rapport à la réduction de Lodha-Warren sur les 3-simplex de Bézier, on perd les propriétés suivantes :

- Interpolation de Hermite : pour les 3-simplex, le simplex réduit de degré e interpole les $\lfloor \frac{e-1}{2} \rfloor$ premières dérivées aux coins du simplex. Ce n'est évidemment pas applicable aux patches algébriques puisque ceux-ci n'interpolent pas forcément les coins du simplex. La fonction algébrique interpole les points de contrôle aux extrémités, mais pas la surface.
- Ordre d'approximation : l'algorithme de Lodha atteint l'ordre optimal d'approximation $O(h^{e+1})$, où h est le diamètre du simplex. Appliqué aux 3-simplex et aux surfaces algébriques, la propriété reste vraie bien que h soit maintenant une distance dans l'espace quadridimensionnel. Par contre, on ne peut pas appliquer directement ce résultat en dimension 3.

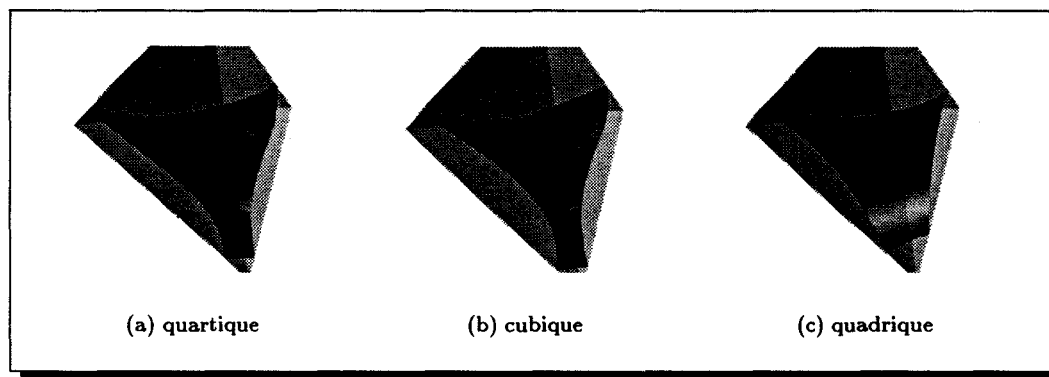


FIG. 5.9 - Une surface quartique et ses approximations cubique et quadrique

5.3.3 Réduction par morceaux

Une extension naturelle des algorithmes d'approximation est l'approximation par morceaux : utiliser plusieurs morceaux de surfaces d'un certain degré pour approximer une surface de degré supérieur. Ce processus améliore notablement l'approximation puisqu'elle permet d'approximer une surface très complexe par plusieurs surfaces qui peuvent être de degré beaucoup plus faible.

L'approximation par réduction de degré peut également se faire par morceaux. Le principe est simple : subdiviser la surface à approximer, et effectuer une réduction de degré sur chacun des morceaux obtenus, en conservant le plus grand ordre de continuité possible entre morceaux adjacents.

Subdivision de l'espace

La première étape est donc de subdiviser l'espace dans lequel est plongée la surface de départ. Une boîte englobante de celle-ci est donc choisie et découpée en tétraèdres dans lesquels sont effectuées les réductions de degré.

Plusieurs méthodes de décomposition de cubes ou de tétraèdres en tétraèdres (récursives et/ou symétriques) sont présentées par D. Moore [Moore92] : un cube est divisé en 6 tétraèdres par le découpage de Kuhn (deux sommets opposés du cube A et B sont choisis, et les 6 chemins directs

possibles de A vers B le long des arêtes du cube forment chacun un tétraèdre) et un tétraèdre se subdivise en 8 sous-tétraèdres (un sous-tétraèdre par sommet laisse un volume central octaédrique découpé en 4 sous tétraèdres).

Une fois que la surface à réduire est subdivisée en cellules tétraédriques, il est possible d'appliquer la réduction de degré sur chaque cellule, et l'ensemble des morceaux de surfaces générés forme l'approximation de la surface de départ.

L'inconvénient majeur reste qu'une surface n'intersecte souvent qu'une faible partie des tétraèdres de la subdivision ; autrement dit, beaucoup de cellules restent vides. Par conséquent, beaucoup de calculs sont faits pour rien. La solution a priori évidente est de détecter les tétraèdres vides pour ne pas y effectuer la réduction de degré. Mais cela pose deux problèmes :

- Détecter si un tétraèdre contient ou non une surface algébrique est une généralisation du problème de la détection de zéros d'un polynôme dans un intervalle donné. Une condition suffisante (mais pas nécessaire) est que le volume de Bézier au-dessus n'intersecte pas le tétraèdre est que les points de contrôle soient tous au-dessus (i.e. positifs), ou tous en-dessous (i.e. négatifs) du tétraèdre. Des tests plus fins existent, mais leur complexité augmente rapidement avec la précision.
- Il arrive que l'élimination de tétraèdres vides proches de la surface de départ entraîne l'apparition de trous dans l'approximation par morceaux. Cela est dû au fait qu'un morceau de surface peut très bien apparaître dans un tétraèdre réduit alors que le simplexe de départ n'intersectait pas la surface.

Finalement, faire la réduction de degré sur tous les tétraèdres s'avère être la seule manière simple de générer l'approximation correctement et dans tous les cas.

Propriétés de la réduction par morceaux

Toutes les propriétés de la réduction simple sont conservées. On a également un résultat sur l'ordre de continuité lors du recollement de morceaux : deux morceaux contigus de degré 2, issus du processus de réduction de degré se joignent avec un ordre de continuité $[\frac{e-1}{2}]$. Cela est dû au fait que d'une part les deux simplexes au-dessus interpolent leurs sommets communs avec continuité $[\frac{e-1}{2}]$ (propriété d'interpolation de Hermite de l'algorithme de Lodha). Comme les deux surfaces algébriques appartiennent aux simplexes respectifs, elles sont elles mêmes C^k le long de la face.

5.3.4 Application aux quadriques

L'algorithme de réduction de degré que nous avons décrit s'applique pour la génération d'approximations de surfaces algébriques de degré quelconque par une ou plusieurs surfaces algébriques de degré inférieur. Les surfaces quadriques sont utilisables dans ce cadre puisque ce sont des surfaces algébriques de degré 2.

La réduction de degré peut se faire de plusieurs manières avec les surfaces quadriques :

- Facettisation : une surface quadrique peut être approximée par un ensemble de patches de degré 1, c'est-à-dire facettes (triangles ou carreaux).
- Une surface de degré quelconque peut être approximée par un ensemble de quadriques.

C'est la seconde possibilité qui nous intéresse ici. L'application est simple : soit une surface de degré quelconque est approximée par une quadrique (figure 5.9) par une réduction de degré jusqu'au degré 2, soit elle est subdivisée pour être réduite par morceaux. Cette seconde possibilité permet évidemment d'obtenir les meilleures approximations.

Une possibilité supplémentaire est la réduction de surfaces elles-mêmes par morceaux. Ceci est bien entendu possible uniquement dans le cas où le découpage utilisé dans la réduction de degré est aligné avec celui du découpage de la surface d'origine.

Propriétés

Les propriétés de la réduction de degré générale (symétrie, restriction, invariance, précision, linéarité) sont bien entendu conservées.

La propriété de continuité nous indique que la surface générée est uniquement G^0 . Ce résultat n'est malheureusement pas optimal pour les modèles quadriques. D'un point de vue pratique, la principale conséquence est qu'il est souvent nécessaire d'avoir une subdivision fine pour "adoucir les angles" et que par conséquent plus de quadriques sont générées.

Deuxième constatation : à cause de la subdivision de l'espace, beaucoup de patches sont générés, mais il apparaît que souvent les patches générés appartiennent à la même quadrique, dans certains cas particuliers comme les surfaces possédant une symétrie ou les surfaces de révolution. Sur l'exemple de la figure 5.10, les 106 patches générés par l'algorithme appartiennent à trois quadriques différentes. De la même façon les modèles de la figure 5.11 peuvent être représentés exactement par, respectivement, 3, 6 et 10 quadriques.

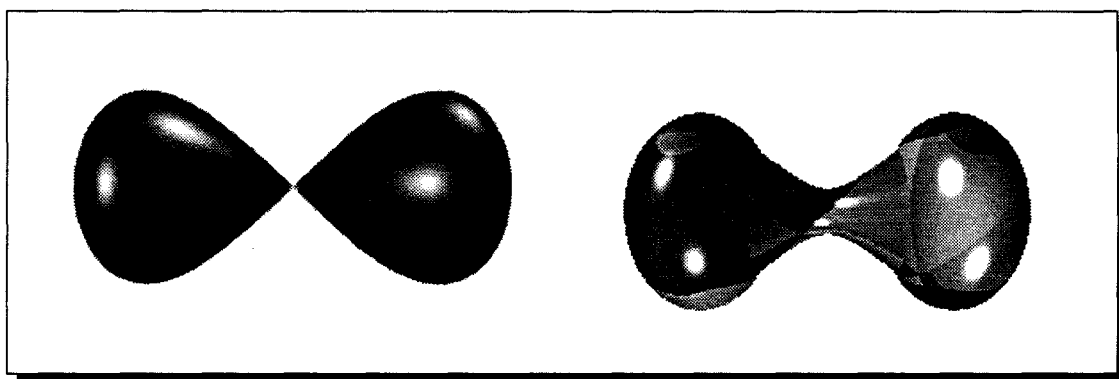


FIG. 5.10 - Surface quartique (gauche) et approximation quadrique (droite)

Améliorations

Nous évoquons dans cette section les possibilités d'amélioration de notre algorithme, dans deux domaines différents : le lissage du modèle quadrique et la mesure d'erreur.

Lissage Même si l'on ne peut obtenir un modèle G^1 par la réduction de degré en quadrique, certaines techniques peuvent être utilisées pour "lisser" un modèle. Même si l'on n'obtient pas la continuité G^1 , l'aspect des modèles est en général amélioré. L'une des plus simples est issue du fait que la subdivision est régulière. En effet, la condition de continuité C^1 sur l'ensemble des 3-simplex de Bézier (qui entraîne la continuité G^1 de la surface quadrique) stipule que l'ensemble

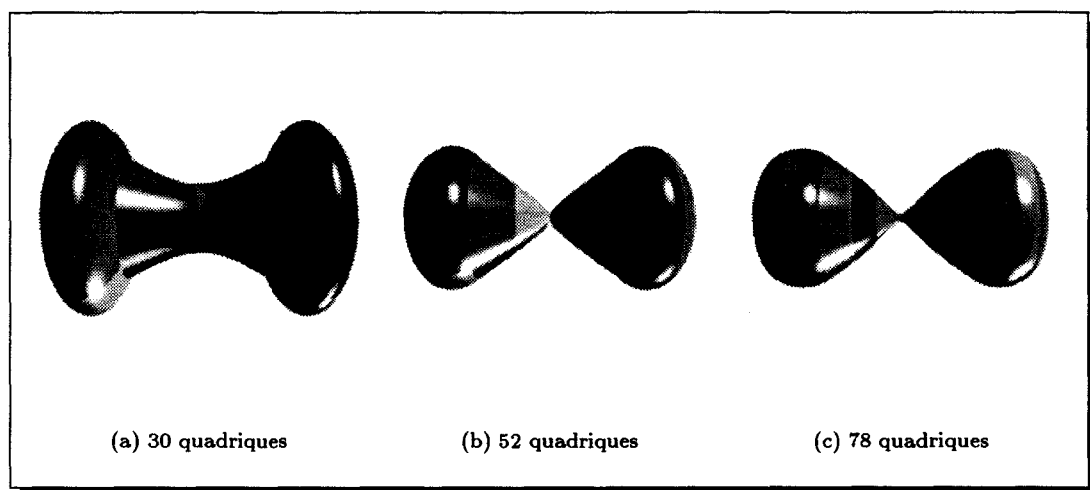


FIG. 5.11 - *Différents découpages pour l'approximation*



FIG. 5.12 - *Approximation d'une surface de degré 5*

des points de contrôle autour d'un sommet doivent être dans le même hyper-plan de \mathbb{R}^4 (§ 3.2.4). Ce n'est le cas ici, mais on peut s'en rapprocher en modifiant le coefficient du point central : en le fixant à la moyenne des coefficients autour (figure 5.13). Même si cette méthode permet d'adoucir le modèle final puisqu'on se "rapproche" des conditions de coplanarité 4D, il n'est pas garanti que celles-ci soient atteintes (puisque'il y a plus de quatre points autour de celui modifié, qui en général ne sont pas coplanaires entre eux).

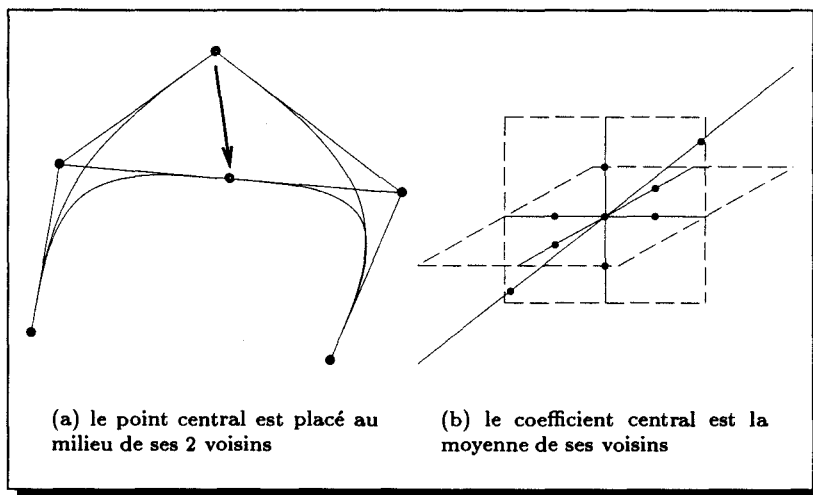


FIG. 5.13 - Lissage d'une spline de Bézier (a) et d'un ensemble de 3-simplex (b)

D'autres méthodes de lissage, extrapolées des méthodes de courbes sont envisageables (par exemple [Ma et al.95]).

Mesure de l'erreur Comme dans tout processus d'approximation, la possibilité de mesurer l'erreur commise est important. En particulier, pour l'algorithme de réduction de degré, la mesure de celle-ci peut être utile pour la détermination du meilleur découpage de la surface. Ce paragraphe introduit les problèmes rencontrés dans notre cas.

La mesure de l'erreur de l'algorithme de réduction de degré appliqué aux surfaces algébriques pose le problème de la *distance* entre deux surfaces algébriques. À notre connaissance, aucune solution satisfaisante, combinant à la fois l'exactitude de la mesure et la simplicité, n'existe à ce jour. Néanmoins l'utilisation de mesures approximatives de la distance donne souvent de bons résultats.

- Mesures d'erreur géométriques : les mesures purement géométriques de distance sont en pratique difficilement calculables de façon exacte. Par exemple, la distance de Hausdorff, définie par :

$$d_H(A, B) = \max\left\{\sup_{x \in B} d(x, A), \sup_{y \in A} d(y, B)\right\} \quad \text{où} \quad d(x, A) = \inf_{y \in A} d(x, y) \quad (5.6)$$

représente la rayon de la plus grande sphère possible dont le centre est sur l'une des surfaces et qui touche la seconde. Calculer cette distance revient à calculer pour chaque point des deux courbes, le rayon du plus petit cercle centré sur lui et touchant l'autre surface. Le rayon du plus grand cercle calculé de cette manière sur tous les points est la distance de Hausdorff. Cette méthode, difficilement mise en œuvre en pratique, l'a été uniquement dans le cas de courbes linéaires par morceaux [Emery86].

- Mesures algébriques : les seules possibilités de mesure de distance utilisées avec des courbes ou surfaces algébriques sont les distances surface/point. Ces mesures sont utilisées par les algorithmes d'approximation de points et de courbes par des surfaces algébriques (par exemple Bajaj [Bajaj et al.92, Bajaj et al.93] ou Pratt [Pratt87]). L'idée est de mesurer la distance du point $P(x_P, y_P)$ à la courbe d'équation $f(x, y) = 0$ non pas en utilisant la distance euclidienne classique, dont le calcul est trop lourd, mais en calculant simplement $|f(x_P, y_P)|$ (figure 5.14).

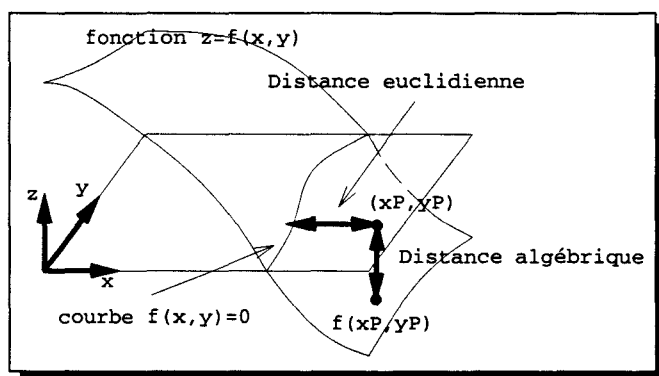


FIG. 5.14 - Distances euclidienne et algébrique d'un point à une courbe

Les problèmes de l'utilisation de cette distance pour une mesure surface/surface sont de deux types :

- Échantillonnage : à nouveau, il est nécessaire de trouver les "bons" points d'une surface qui donnent une distance correcte à l'autre surface. Ajouté au problème d'échantillonner un point sur une surface algébrique, c'est une méthode qui s'avère difficile à réaliser correctement. De la même manière, l'idée de calculer l'intégrale des distances à la première surface le long de la seconde s'avère très difficile pratiquement.
 - Normalisation : comme la fonction αf décrit la même courbe que la fonction f pour tout réel α , la distance algébrique peut être différente pour la même courbe. Il est donc nécessaire de *normaliser* les courbes αf . Plusieurs normalisations existent (la plus simple étant probablement de fixer l'un des coefficients du polynôme de f à 1), mais la plupart d'entre elles posent des problèmes de singularité [Pratt87].
- Mesures paramétriques : la difficulté des mesures algébriques due à l'échantillonnage de points peuvent être levées, au détriment de la qualité de la mesure, en considérant tous les points du domaine de la fonction f , et pas uniquement ceux de la surface $f(x, y, z) = 0$. On peut ainsi envisager l'utilisation des distances *paramétriques*. Ces distances utilisées par les algorithmes d'approximation des courbes et surfaces paramétriques. Elles sont toujours mesurées entre les points de paramètres égaux. Par exemple :

$$d(A, B) = \max_{(x, y) \in \mathbb{R}^2} \{|f_B(x, y) - f_A(x, y)|\} \quad (5.7)$$

Cette distance serait algébrique si l'on avait $(x, y) \in A$ ou $(x, y) \in B$.

Les distances paramétriques entre deux ensembles sont bien connues dans le cas des courbes et surfaces. Lorsque les courbes sont assez semblables, ces distances sont en général satisfaisantes, encore une fois parce que les surfaces générées par l'algorithme de réduction de degré sont relativement proches l'une de l'autre.

Dans le cas des surfaces algébriques, ces distances évitent donc le problème du domaine $((x, y) \in A)$, mais l'utilisation de fonctions comme *max* (équation 5.7) reste complexe, le domaine étant dans notre cas tridimensionnel. Par contre, l'idée d'intégrer devient possible :

$$d(A, B) = \int_{R^3} |f_B(x, y, z) - f_A(x, y, z)| dx dy dz \quad (5.8)$$

Une solution simple et efficace n'existe, à notre connaissance, pas, bien que certaines voies soient à explorer. Une méthode pour le calcul automatique d'un "bon" découpage de l'espace (selon des critères simples) serait néanmoins très intéressante pour beaucoup d'algorithmes basés sur les surfaces algébriques.

Bilan

Calculer l'approximation d'une surface quelconque par des quadriques est jusqu'à présent un problème peu exploré, mais qui risque de devenir fondamental si l'on atteint les possibilités d'affichage présentées au chapitre 1.

Dans cette optique, nous avons introduit un algorithme pour la quadratisation des surfaces algébriques de degré quelconque, qui exploite encore une fois les propriétés de la représentation de Bernstein-Bézier. Cette méthode génère un modèle quadrique par morceaux qui approxime plus ou moins bien la surface d'origine selon le découpage de la boîte englobante choisi.

Bien que notre algorithme ne s'applique qu'aux surfaces algébriques, il est possible d'envisager son application aux surfaces paramétriques simples dont on peut calculer l'équation algébrique (par exemple, les triangles de Bézier quadratiques rationnels sont des surfaces algébriques de degré 4 [Sederberg et al.85]).

Le problème le plus restrictif reste la continuité de l'approximation, uniquement G^0 . Nous pensons que les techniques de lissage proposées peuvent améliorer la situation, voire permettre d'atteindre la continuité G^1 . Bien entendu, des définitions plus complexes des quadriques, dont la représentation par simplex B-spline (chapitre 4) devraient permettre d'améliorer également les résultats obtenus ici.

Chapitre 6

Bilan et Propositions

Dans les deux chapitres précédents, nous nous sommes intéressés à la création de surfaces complexes continues à l'aide de quadriques sous deux aspects : la modélisation directe et l'approximation. Nous avons vu que les méthodes proposées et les résultats obtenus répondaient au problème de départ mais que, même si les nouvelles représentations du chapitre 3 ont permis de construire des schémas complexes de modélisation, la simplicité intrinsèque des quadriques ne permet pas d'atteindre toutes les exigences de la modélisation géométrique actuelle.

Toutefois, les questions fondamentales du chapitre 1 concernant l'aspect logiciel des quadriques en rendu matériel restent posées : à partir du moment où l'on dispose des quadriques comme primitives d'affichage : les quadriques constituent-elles un progrès par rapport aux facettes 3D actuelles ? Et à partir de là, un accélérateur pour l'affichage de quadriques a-t-il un sens ?

Dans ce chapitre final, nous tentons, à la lumière des résultats précédents, d'apporter une réponse à ces questions. Dans une première partie, nous analysons les résultats précédents dans une optique plus générale, puis nous concluons sur les possibilités actuelles et à venir.

6.1 Synthèse

Nous avons présenté au chapitre 1 les possibilités actuelles de l’affichage matériel temps-réel sous la forme de l’évolution des primitives de visualisation : à partir des techniques d’affichage des facettes, il est actuellement possible de passer à l’affichage de surfaces quadratiques, puis aux surfaces quadriques. Celles-ci constituent actuellement la perspective la plus poussée, tout en restant technologiquement raisonnables.

L’étude géométrique des quadriques montre immédiatement les intérêts de l’utilisation de celles-ci pour la synthèse d’images, comparée à l’emploi actuel des facettes planes. Celles-ci sont, d’une part, adaptées pour représenter les objets courbes et, d’autre part, elles se raccordent avec continuité de la tangente. Ainsi il est possible d’éviter les problèmes bien connus de l’affichage de facettes, même avec l’algorithme de lissage de Gouraud : discontinuité de l’éclairage, contours polygonaux, etc. Une seconde conséquence de ces propriétés est que les quadriques permettent de représenter les objets tridimensionnels courbes avec un nombre bien moins élevé de primitives qu’avec les facettes planes traditionnelles. Toutefois, ces propriétés géométriques ne sont exploitables qu’à l’aide de schémas appropriés de construction de surfaces quadriques par morceaux, dont l’étude constitue l’objet principal de cette thèse.

Le chapitre 3 est consacré aux différentes définitions et représentations mathématiques des quadriques. Nous avons vu que les représentations traditionnelles (algébrique et paramétrique) étaient peu adaptées à une modélisation surfacique complexe. Mais de nouvelles représentations des surfaces algébriques, basées sur la théorie des courbes et surfaces de Bézier, permettent actuellement de pallier ce problème en offrant des outils de contrôle pour les surfaces quadriques : polygones de contrôle ou pondérations aux sommets. Parmi ces nouveaux modèles, la représentation de Bernstein-Bézier permet de contrôler les surfaces à l’aide d’un simplexe de Bézier de dimension 3. Cette représentation est à l’origine de nombreux développements récents sur les surfaces algébriques car elle permet, par un contrôle plus intuitif de la forme d’une quadrique et du raccordement entre quadriques, l’élaboration d’outils de haut-niveau pour la création de surfaces par morceaux.

Au chapitre 4, après avoir présenté les méthodes de construction de surfaces quadriques par morceaux, nous avons introduit trois nouveaux schémas de modélisation.

- Premièrement, nous avons présenté un schéma de construction de surfaces quadriques de révolution par interpolation de points, avec contrôle de la continuité aux raccords entre quadriques (§ 4.2.3).
- Deuxièmement, nous avons développé la définition de macro-patchs quadriques, utilisant la représentation de Bernstein-Bézier dans le but de construire des primitives de modélisation simples, imitant les triangles de Bézier quadratiques au niveau du contrôle de la forme (par polygone de contrôle). En combinant un nombre variable de quadriques, nous pouvons atteindre les ordres de continuité G^0 ou G^1 à l’intérieur d’un macropatch ainsi qu’entre macropatchs (§ 4.4).

La définition d’une primitive surfacique déformable par polygone de contrôle nous permet de créer un schéma de modélisation utilisant des processus complexes (comme la déformation continue) nécessitant un nombre réduit de patchs simples. Cela permet d’envisager des solutions de modélisation sans problème de conversion depuis les primitives de modélisation vers les primitives de visualisation (l’exemple le plus courant de ce genre de “conversion” est la facettisation des surfaces complexes).

- Troisièmement, nous avons présenté une solution plus globale résolvant les principaux problèmes des macropatchs. Celle-ci consiste à étendre la représentation de Bernstein-Bézier à la représentation par simplexe B-spline. Cette représentation nous permet de construire

une surface quadrique par morceaux, ayant les propriétés fondamentales des B-splines : la conservation de l'ordre de continuité G^1 de la surface, quelle que soit la position de ses points de contrôle, ainsi que la possibilité de définir directement une surface de topologie quelconque. Par rapport à la représentation de Bernstein-Bézier, cette représentation rend en fait implicites les conditions de raccordement et évite donc les limitations importantes rencontrées lors de la construction des macro-patches (§ 4.5.3).

Même si l'on peut obtenir des outils de modélisation de haut-niveau avec des quadriques, il est néanmoins nécessaire de prévoir des solutions d'affichage dans le cas d'utilisation d'autres types de primitives de modélisation plus courantes : surfaces de Bézier, surfaces implicites, etc. La solution est alors, à l'instar des algorithmes de polygonalisation qui sont utilisées actuellement avec les accélérateurs facettes, de fournir des algorithmes de "quadratisation", c'est-à-dire d'approximation de surfaces quelconques par des quadriques par morceaux. Nous proposons au chapitre 5 une première solution : un algorithme de réduction de degré de surfaces algébriques, qui permet d'approximer toute surface algébrique par une surface algébrique par morceaux de degré inférieur. Cet algorithme permet en particulier d'approximer toute surface algébrique par un ensemble de quadriques.

Toutes les méthodes présentées ont été validées par une implémentation : bien que nous ne disposions pas d'un prototype d'accélérateur quadrique pour les valider en "temps-réel", les schémas que nous avons proposés ont été programmés et testés, et ont servi à produire les images des chapitres précédents.

*

Les résultats obtenus sont encourageants : les nouvelles représentations des quadriques ont ouvert la voie à des possibilités de modélisation beaucoup plus poussées que les traditionnelles quadriques naturelles ou quadriques de révolution. Néanmoins, nous n'avons pas complètement atteint les perspectives envisagées : la construction simple et avec peu de quadriques de surfaces G^1 , facilement contrôlables et affichables en temps-réel. En effet, la complexité des algorithmes ainsi que la difficulté d'obtenir une amélioration réelle au niveau de la qualité des modèles ou du nombre de primitives incite à rester prudent, pour le moment, par rapport aux objectifs visés.

La représentation de Bernstein-Bézier (chapitre 4) n'élimine pas tous les problèmes de définition des surfaces algébriques : les conditions de raccordement ne sont pas géométriques¹ et sont donc difficiles à prendre en compte. Un autre problème non résolu de façon satisfaisante est le contrôle du nombre de nappes à l'intérieur d'un patch. C. Bajaj a énoncé une condition suffisante pour qu'un patch de Bernstein-Bézier de degré quelconque ne comporte qu'une nappe [Bajaj et al.95b], mais celle-ci ajoute encore des contraintes à celles déjà produites par le contrôle de la continuité. Pour cette raison, les algorithmes de construction de quadriques ne vérifient jamais cette condition.

Les solutions que nous avons ensuite proposées, bien que prometteuses, sont encore loin d'offrir les qualités importantes en modélisation surfacique : les macropatches (chapitre 4) se raccordent difficilement avec continuité G^1 , même si un macropatch est lui-même complètement G^1 . Les B-splines quadriques sont, elles, G^1 partout mais sont définies par un nombre important de paramètres (points de contrôle, coefficients, et noeuds) et génèrent un nombre de quadriques important, même dans les cas les plus simples. Ajoutée à cela la difficulté de calculer la forme algébrique (§ 4.5.3) les B-splines quadriques restent une solution encore difficile à adapter au contexte que nous nous sommes donnés.

Les solutions d'approximation par des quadriques (chapitre 5) sont également loin d'offrir quelque chose de satisfaisant. Les algorithmes sont relativement complexes et dépendent de problèmes de métrique non résolus à notre connaissance. La réduction de degré, en particulier, souffre

1. Pour être plus exact, elles sont géométriques mais en dimension 4 (§ 3.23)

du désavantage de ne générer que des modèles G^0 , même si des algorithmes de lissage sont envisageables. En outre, à l'instar des algorithmes du type "marching cubes", la topologie de la surface à approximer n'est pas prise en compte. Finalement, le nombre de primitives quadriques généré reste élevé, comparé à ce à quoi l'on pourrait espérer obtenir.

*

En synthèse, nous pouvons dire que les méthodes que nous avons présentées ou proposées montrent que la construction d'un accélérateur d'affichage de quadriques a un sens, au moins en ce qui concerne l'aspect logiciel.

D'une part, l'utilisation d'un accélérateur quadrique constituera un gain énorme pour toute modélisation utilisant les quadriques naturelles. La sphère, le cylindre ou le cône restent des objets très courants, et il ne faut pas chercher très loin pour voir des paraboloides et autres hyperboloides. En clair, les méthodes présentées au début du chapitre 4 (quadriques naturelles, quadriques spécifiques, quadriques de révolution) trouveront toujours leur utilisation au sein d'un afficheur spécifique. Et comme un afficheur quadrique a toujours la possibilité d'afficher des facettes planes [Laporte96], les quadriques constitueront au minimum une extension aux facettes. Une scène facettisée pourra inclure quelques quadriques pour représenter certaines surfaces (morceaux de sphères, de cylindres...) très coûteuses en nombre de facettes. Le gain en nombre de primitives sera déjà important.

D'autre part, les méthodes présentées ou introduites dans cette thèse sont une réponse crédible au problème de la création ou l'approximation de surfaces complexes à l'aide de quadriques. Bien que certaines des méthodes présentées ne soient pas, nous l'avons vu, optimales pour la conception de surfaces complexes et lisses, elles permettent néanmoins d'exploiter de façon poussée les propriétés géométriques des quadriques et représentent un progrès important par rapport aux méthodes de modélisation classiques.

Nous pensons que ces résultats font des quadriques une solution intéressante en tant que primitives de visualisation, puisqu'elles se révèlent utilisables pour un grand nombre d'applications graphiques temps-réel.

6.2 Perspectives

Vers une méthodologie d'évaluation des performances

L'analyse faite précédemment ne serait complète qu'avec l'addition de résultats quantitatifs concernant les performances globales d'un système graphique complet à base de quadriques. La connaissance exacte de cette information pourrait permettre de trancher pour de bon le problème du gain par rapport aux facettes. Malheureusement il est très difficile de prévoir des performances, que ce soit du côté d'un accélérateur matériel, comme celui présenté par Laporte [Laporte96], ou du côté de méthodes logicielles, comme celles présentées ici :

- *Le Matériel.* Même si l'on annonce couramment les performances de machines graphiques existantes (en nombre de facettes par seconde), ces chiffres ne sont en réalité jamais fixes et dépendent beaucoup de l'image calculée (en particulier de la taille des facettes ou du nombre de facettes par pixel). On peut juste espérer pouvoir calculer une valeur moyenne, même si celle-ci changera beaucoup en fonction de l'utilisation de l'accélérateur.

Deuxièmement, la conception d'un accélérateur quadrique se démarque de la réalisation de celui ci, pour laquelle certains choix sont effectués dans lesquels interviennent des notions

de coût, influant sur le nombre de processeurs mis en parallèle, par exemple. Alors que des données telles que le nombre de transistors par unité sont bien définies à la conception, on peut seulement fournir une évaluation grossière des performances [Laporte96] :

Au niveau des performances, l'architecture de notre système ne permet qu'un parallélisme faible de quelques processeurs. Elle convient donc uniquement pour des accélérateurs bas de gamme. [...] Pour espérer obtenir des performances de rendu comparables aux machines haut de gamme il faut repenser l'architecture complète du système.

- *Le Logiciel.* Pour toutes les méthodes de modélisation, l'évaluation des performances voudrait que l'on mesure également des cadences (en nombre de quadriques par seconde) de production du modèle à envoyer à l'accélérateur, soit pour le calcul d'un modèle soit pour un recalcul (lors d'une déformation par exemple). Encore une fois cela n'a pas de sens puisque ces chiffres varient énormément d'un environnement à l'autre. Encore une fois il est uniquement possible de donner des performances grossières, voire des mesures de complexité (on déduit facilement par exemple que notre algorithme d'approximation de quadriques a une complexité proportionnelle au nombre de tétradres du découpage de l'espace). Dans la pratique, pour la génération de modèles à facettes, cela est très rarement fait.

Donner des informations quantitatives de performances globales d'un système à base de quadriques s'avère par conséquent très hasardeux (au moins autant que pour un système modéleur/accélérateur facettes). Il va de soi qu'une connaissance précise des performances matérielles et logicielles d'un prototype éventuel sera nécessaire pour se prononcer définitivement sur la validité de notre projet.

Amélioration des méthodes de modélisation

Dans quelle direction peuvent évoluer les méthodes de conception de surfaces quadriques ? Nous savons que les quadriques n'atteindront jamais la flexibilité des surfaces de degré supérieur. Néanmoins, il n'existe pas à notre connaissance de limite fondamentale caractérisée aux possibilités de modélisation et de création de surfaces. Donc nous pouvons envisager l'amélioration des algorithmes présentés précédemment. Quatre directions sont a priori possibles :

- Rendre les algorithmes robustes, par exemple supprimer les nappes indésirables ou les cas dégénérés.
- Améliorer la qualité des modèles, en particulier atteindre systématiquement la continuité G^1 .
- Généraliser : adapter un grand nombre de schémas de modélisation (par squelette, par contraintes, etc.), ou approximer par des quadriques tout type de surface.
- Minimiser le nombre de quadriques pour un modèle donné.

La possibilité d'améliorer ces quatre aspects à la fois semble difficile. L'expérience que nous ont apporté les chapitres précédents, ou d'autres tentatives, est que, par exemple, l'amélioration de la qualité des modèles amène une multiplication du nombre de primitives, ou que l'utilisation de la condition de suppression des nappes entraîne une perte de la continuité G^1 . Toutefois certaines possibilités existent et méritent, selon nous, d'être explorées :

- Pour les macropatchs, un modèle complètement G^1 sans contraintes complexes sur la continuité doit pouvoir être atteint. Toutefois, il sera nécessaire d'utiliser encore plus de quadriques.

- Pour les simplex B-splines quadriques, les perspectives sont nombreuses, en particulier l'adaptation des algorithmes récents pour les simplex "classiques" pour des méthodes de construction telles que : l'interpolation, l'approximation ou la modélisation interactive.
- Pour l'approximation par réduction de degré, les directions possibles sont : la génération de surfaces quadriques G^1 , une approximation plus adaptée à la topologie de la surface (pour le découpage de l'espace par exemple) et surtout la généralisation à l'approximation de surfaces autres qu'algébriques (en particulier paramétriques).

Affichage

Rappelons que ce travail a été au départ défini dans le cadre de la synthèse d'image (chapitre 1) et non dans un contexte géométrique pur. Par conséquent les impossibles géométriques peuvent être palliés par des méthodes de rendu, de la même manière que les problèmes géométriques des facettes 3D (pas de continuité G^1) sont palliés (de façon moyenne) par le lissage de Gouraud.

Nous pensons dans une certaine mesure que c'est autant au niveau des algorithmes de rendu que des algorithmes de modélisation que les possibilités d'amélioration des méthodes à base de quadriques existent. En effet, l'important pour nous étant l'aspect visuel du modèle, l'exactitude géométrique est à la limite secondaire². Au niveau de la modélisation, cela laisse une liberté somme toute extrêmement réduite, puisque les discontinuités par exemple se remarquent immédiatement. Par contre certains aspects, voire modifications, du processus de rendu peuvent, eux, améliorer considérablement l'aspect visuel. Citons principalement :

- Les textures : la texture appliquée à un objet peut fortement influencer sa géométrie telle que nous la percevons. Le modèle qui s'applique traditionnellement aux surfaces algébriques est le modèle de *textures 3D* où l'on associe une couleur à tout point de l'espace en fonction de ses coordonnées $\{x, y, z\}$. Ce modèle s'applique lorsqu'il s'agit de représenter des objets sculptés dans du bois, du marbre, etc. Dans le cas où l'on désire un *placage* de texture, les modèles sont moins simples. Il est nécessaire de disposer de *coordonnées de surface*, par exemple une paramétrisation, ce qui n'est pas le cas avec la forme implicite des quadriques. Certains résultats récents montrent que d'autres voies s'ouvrent pour ce problème [Pedersen95, Laporte96].
- La modification couleurs/normales : à l'instar des algorithmes d'interpolation de Gouraud ou de Phong [Foley et al.90] qui permettent d'afficher un modèle facettes géométriquement G^0 , de telle sorte qu'il soit lissé et apparaisse (à peu près) G^1 , il est possible d'envisager une interpolation des couleurs ou des normales : la couleur affichée (ou la normale utilisée pour le calcul de l'éclairage) n'est pas la couleur (ou la normale...) "réelle" de la surface mais un *mélange* (blending) des couleurs/normales de la quadrique sur laquelle on fait le calcul et des quadriques voisines. Une fonction de mélange continue donnerait un aspect G^1 à la surface³. L'avantage est immédiat au niveau de la modélisation : il ne suffit plus que de générer des modèles G^0 , donc à l'aide d'algorithmes bien moins complexes que pour les modèles G^1 (chapitre 4) et produisant un nombre très réduit de primitives.

Bien, entendu ces deux possibilités d'amélioration rendent plus complexes les algorithmes de rendu, et leur implémentation reste à étudier (nous référons encore une fois le lecteur aux travaux de Laporte et de Nyiri pour tout ce qui concerne l'aspect matériel [Laporte96, Nyiri94]).

2. C'est ce qui nous a amené, par exemple, à considérer au chapitre 4 plusieurs modèles macropatchs tous G^0 , mais ayant un aspect visuel meilleur quand le nombre de quadrique par macropatch était plus grand (§ 4.4.2)

3. À part pour les contours qui restent inchangés (figure 1.9).

Dans un contexte géométrique pur, mais dans une perspective plus éloignée, deux possibilités existent.

- L’affichage de surfaces polynomiales de degré supérieur à deux, linéaires dans la direction d’observation, c’est-à-dire de la forme $z = f(x, y)$ (§ 1.3.1). Rappelons que l’affichage est relativement simple puisqu’il ne nécessite pas de calcul de racine en z , mais que les surfaces affichables ne sont alors pas invariantes par rotation puisqu’elles sont forcément alignées dans l’axe de l’observateur. Le fait d’élever le degré permet peut-être d’éviter les problèmes d’affichage que nous avons vu, et autoriser un compromis modélisation/affichage intéressant.
- La meilleure solution du point de vue géométrique, mais bien plus difficile à réaliser au niveau matériel, est bien entendu l’affichage de surfaces algébriques *quelconques* de degré supérieur à deux. Les surfaces algébriques ont beaucoup de propriétés intéressantes pour la modélisation (§ 3.1.3) et beaucoup de méthodes récentes les utilisent, mais la plupart des travaux sur les surfaces algébriques concernent des surfaces de degré supérieur à deux : cubiques [Sederberg90, Lodha et al.92, Dahmen et al.92, Guo93a], de degré 5 [Bajaj et al.92], ou de degré non fixé [Bajaj et al.95a, Moore et al.91]. Rappelons que W. Dahmen et B. Guo ont d’abord publié sur les quadriques (sans résultats complètement satisfaisants, cf. chapitre 4) [Dahmen89, Guo93a], avant de passer aux possibilités plus intéressantes (pour eux) des cubiques [Dahmen et al.93, Guo93a]. Ces travaux montrent que l’augmentation du degré des surfaces affichables, supprimerait la plupart des problèmes rencontrés avec les quadriques pour la construction de surfaces déformables⁴. Malheureusement, l’implémentation matérielle de l’affichage de ces surfaces nécessite le calcul de plusieurs profondeurs en un pixel, qui ne peut se faire de façon exacte que jusqu’au degré 4, et de façon itérative (Newton) ensuite. Alors que cette seconde possibilité paraît impensable à réaliser en temps-réel, la première pourrait, nous l’espérons, être envisagée dans le futur.

4. Les meilleurs exemples sont les triangles de Bézier quadratiques qui sont de degré algébrique 4 [Sederberg et al.85].

Bibliographie

- [Allgower et al.85] Allgower (Eugene L.) et Schmidt (Phillip H.). – An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM Journal of Numerical Analysis*, vol. 22, n° 2, April 1985, pp. 322–346.
- [Auerbach et al.91] Auerbach (S.), Gmelig-Meyling (R.H.J.), Neamtu (M.) et Schaeben (H.). – Approximation and geometric modeling with simplex B-splines associated with irregular triangles. *Computer Aided Geometric Design*, vol. 8, 1991, pp. 67–87.
- [Bajaj et al.92] Bajaj (C.) et Ihm (I.). – Algebraic surface design with Hermite interpolation. *ACM Transactions on Graphics*, vol. 11, n° 1, January 1992, pp. 61–91.
- [Bajaj et al.93] Bajaj (Chanderjit), Ihm (Insung) et Warren (Joe). – Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Transactions on Graphics*, vol. 12, n° 4, october 1993, pp. 327–347.
- [Bajaj et al.95a] Bajaj (Chandrajit L.), Bernardini (Fausto) et Xu (Guoliang). – Automatic reconstruction of surfaces and scalar fields from 3D scans. *Siggraph '95 conference proceedings*. ACM. – ACM Press, 1995.
- [Bajaj et al.95b] Bajaj (Chandrajit L.), Chen (Jindin) et Xu (Guoliang). – Interactive shape control and rapid display of A-patches. *Implicit Surfaces '95*, éd. par Wyvill (Brian) et Gascuel (Marie-Paule). Eurographics, pp. 197–215. – INRIA, 1995.
- [Bajaj88] Bajaj (Chanderjit L.). – Geometric modeling with algebraic surfaces. *The mathematics of surfaces III*, éd. par Handscomb (D.), pp. 3–48. – Oxford University Press, 1988.
- [Bajaj95] Bajaj (Chandrajit L.). – The SHASTRA collaborative modelling and visualization environment. – <http://www.cs.purdue.edu/research/shastra>, 1995.
- [Bezier86] Bézier (Pierre). – *Mathématiques et C.A.O. vol 4 : courbes et surfaces*. – Hermès, 1986.
- [Blinn82] Blinn (James F.). – A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, vol. 1, n° 3, July 1982, pp. 235–256.
- [Bloomenthal et al.91] Bloomenthal (Jules) et Shoemake (Ken). – Convolution surfaces. *Computer Graphics (Siggraph '91 proceedings)*. ACM, pp. 251–256. – July 1991.
- [Bloomenthal88] Bloomenthal (Jules). – Polygonalization of implicit surfaces. *Computer Aided Geometric Design*, vol. 5, n° 4, 1988, pp. 341–355.

BIBLIOGRAPHIE

- [Bohm et al.84] Böhm (Wolfgang), Farin (Gerald) et Kahmann (Jürgen). – A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, vol. 1, 1984, pp. 1-60.
- [Bourdot91] Bourdot (P.). – Génération automatique de niveaux de définition pour la gestion de scènes composées de surfaces sculptées et de polyèdres. *Journées graphiques Gros Plan*. pp. 15-22. – Laboratoire d'informatique fondamentale de Lille, 1991.
- [Chaillou91] Chaillou (C.). – *Étude d'un processeur de visualisation d'images de synthèse en temps réel exploitant un parallélisme massif objet : le projet IMO-GENE*. – Thèse de PhD, Université de Lille I, January 1991.
- [Chuan et al.89] Chuan (Jung Hong) et Hoffmann (Christoph M.). – On local implicit approximation and its applications. *ACM Transactions on Graphics*, vol. 8, n° 4, 1989, pp. 298-323.
- [Dahmen et al.82] Dahmen (W.) et Micchelli (C. A.). – On the linear independance of multivariate B-splines, i. triangulations of simploids. *SIAM Journal of Numerical Analysis*, vol. 19, n° 5, 1982, pp. 993-1012.
- [Dahmen et al.92] Dahmen (W.), Micchelli (C. A.) et Seidel (H. P.). – Blossoming begets B-spline bases built better by B-patches. *Mathematics of Computation*, vol. 59, n° 199, July 1992, pp. 97-115.
- [Dahmen et al.93] Dahmen (Wolfgang) et Thamm-Shaar (Tom-Michael). – Cubicoids: modeling and visualization. *Computer Aided Geometric Design*, vol. 10, n° 2, 1993, pp. 89-108.
- [Dahmen89] Dahmen (Wolfgang). – *Smooth Piecewise quadric patches*, chap. 23, pp. 181-193. – Academic Press, 1989.
- [Dannenberget al.85] Dannenberg (L.) et Nowacki (H.). – Approximate conversion of surface representations with polynomial bases. *Computer Aided Geometric Design*, vol. 2, n° 2, 1985, pp. 123-132.
- [dB78] de Boor (C.). – *A practical guide to splines*. – Springer Verlag, 1978.
- [Eck93] Eck (Matthias). – Degree reduction of Bézier curves. *Computer Aided Geometric Design*, vol. 10, 1993, pp. 237-251.
- [Ellis et al.91] Ellis (J.L.) et Kedem (G.). – The ray-casting engine and ray representations. *Proceedings of ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pp. 255-267. – June 1991.
- [Emery86] Emery (J.D.). – The definition and computation of a metric on planar curves. *CAD*, vol. 18, 1986, pp. 25-28.
- [Farin92] Farin (Gerald). – *Curves and Surfaces for CAGD*. – Academic Press, 1992, 3 édition.
- [Fiorot et al.92] Fiorot (J.-C.) et Jeannin (P.). – *Courbes Splines Rationnelles*. – Masson, 1992.
- [Foley et al.90] Foley (J. D.), van Dam (A.), Feiner (S. K.) et Hughes (J. F.). – *Computer Graphics, principles and practise*. – Addison-Wesley, 1990, second édition.
- [Fong et al.93] Fong (Philip) et Seidel (Hans-Peter). – An implementation of triangular B-spline surfaces over arbitrary triangulations. *Computer Aided Geometric Design*, vol. 10, 1993, pp. 267-275.

-
- [Froumentin et al.94] Froumentin (M.) et Chaillou (C.). – Déformation interactive de carreaux quadriques. *Revue Internationale de CFAO et d'Infographie*, vol. 9, n° 6, 1994, pp. 753–765.
- [Fuchs et al.89] Fuchs (Henry), Poulton (John), Eyles (John), Greer (Trey), Goldfeather (Jack), Ellsworth (David), Molnar (Steve), Turk (Greg), Tebbs (Brice) et Israel (Laura). – Pixel-Planes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories. *Computer Graphics (SIGGRAPH '89 Proceedings)*, éd. par Lane (Jeffrey), pp. 79–88. – juillet 1989.
- [Gardan91] Gardan (Yvon). – *La CFAO, introduction, techniques, et mise en œuvre*. – Hermès, 1991.
- [Garrity et al.91] Garrity (Thomas) et Warren (Joe). – Geometric continuity. *Computer Aided Geometric Design*, vol. 8, 1991, pp. 51–65.
- [Glassner89] Glassner (Andrew). – *An Overview of Ray Tracing*, chap. 1, pp. 1–32. – Academic Press, 1989.
- [Goldman83] Goldman (R. N.). – Two approaches to a computer model for quadric surfaces. *IEEE Computer Graphics & Applications*, vol. 3, 1983, pp. 21–26.
- [Greiner et al.94] Greiner (Günther) et Seidel (Hans-Peter). – Modeling with triangular B-splines. *IEEE Computer Graphics & Applications*, March 1994, pp. 56–60.
- [Guo93a] Guo (B.). – Nonsplitting macro patches for implicit cubic spline surfaces. *Eurographics '93*, éd. par Hubbold (R.J.) et Juan (R.). Eurographics Association, pp. 434–445. – Blackwell Publishers, 1993.
- [Guo93b] Guo (Baining). – Representation of arbitrary shapes using implicit quadrics. *The Visual Computer*, vol. 9, 1993, pp. 267–277.
- [Hazlewood93] Hazlewood (Carol). – Approximating constrained tetrahedrizations. *Computer Aided Geometric Design*, vol. 10, 1993, pp. 67–87.
- [Karpf93] Karpf (S.). – *Architectures massivement parallèles pour la synthèse d'images en temps réel*. – Thèse de PhD, Université de Lille I, January 1993.
- [Laporte96] Laporte (Hervé). – *Étude logicielle et matérielle d'un système de visualisation temps réel basé sur la quadrique*. – Thèse de PhD, Université des Sciences et Technologies de Lille, 1996.
- [Lasser85] Lasser (Dieter). – Bernstein-Bézier representation of volumes. *Computer Aided Geometric Design*, vol. 2, 1985, pp. 145–149.
- [Leroux et al.90] Leroux (Alain) et Pomes (Roland). – Quadriques. *Toutes les applications de Maple en physique, en chimie et en mathématiques*, pp. 243–268. – Vuibert Supérieur, 1990.
- [Lodha et al.90] Lodha (Suresh) et Warren (Joe). – Bézier representation for quadric surface patches. *Computer Aided Design*, vol. 22, n° 9, November 1990, pp. 574–579.
- [Lodha et al.92] Lodha (Suresh) et Warren (Joe). – Bézier representation for cubic surface patches. *Computer Aided Design*, vol. 24, n° 12, December 1992, pp. 643–650.
-

BIBLIOGRAPHIE

- [Lodha et al.94] Lodha (Suresh) et Warren (Joe). – Degree reduction of Bézier simplexes. *Computer Aided Design*, vol. 26, n° 6, October 1994, pp. 735–746.
- [Lorensen et al.87] Lorensen (William E.) et Cline (Harvey E.). – Marching cubes: a high resolution 3D surface construction algorithm. *SIGGRAPH '87 Conference Proceedings*. ACM, pp. 163–169. – 1987.
- [Ma et al.95] Ma (Lizhuang) et Peng (Qunsheng). – Smoothing of free-form surfaces with Bézier patches. *Computer Aided Geometric Design*, vol. 12, 1995, pp. 231–249.
- [Max81] Max (Nelson L.). – Computer representations of molecular surfaces. *IEEE Computer Graphics & Applications*, August 1981, pp. 21–29.
- [Max90] Max (Nelson L.). – Cone-spheres. *Siggraph '90 conference proceedings*, pp. 59–62. – August 1990.
- [Menon94] Menon (Jai P.). – Constructive shell representations for freeform surfaces and solids. *IEEE Computer Graphics & Applications*, march 1994, pp. 24–36.
- [Miller86] Miller (James R.). – Sculptured surfaces in solid models: issues and alternative approaches. *IEEE Computer Graphics & Applications*, vol. 6, n° 12, December 1986, pp. 37–48.
- [Moore et al.91] Moore (Doug) et Warren (Joe). – Approximation of dense scattered data using algebraic surfaces. *Proc. of the 24th Hawaii International Conference on System Sciences*, pp. 681–690. – Kauai, Hawaii, 1991.
- [Moore92] Moore (Douglas W.). – *simplicial mesh generation with applications*. – Thèse de PhD, Cornell University, 1992.
- [Ning et al.93] Ning (Paul) et Bloomenthal (Jules). – An evaluation of implicit surface tilers. *IEEE Computer Graphics & Applications*, November 1993, pp. 33–41.
- [Nyiri94] Nyiri (Eric). – *Étude de la quadrique comme primitive d'affichage en synthèse d'images*. – Thèse de PhD, Université de Lille I, 1994.
- [Patrikalakis et al.89] Patrikalakis (Nicholas M.) et Kriezis (George A.). – Representation of piecewise continuous algebraic surfaces in terms of B-splines. *The Visual Computer*, vol. 5, 1989, pp. 360–374.
- [Pedersen95] Pedersen (Hans Køhling). – Decorating implicit surfaces. *Siggraph '95 Conference Proceeding*. ACM. – 1995.
- [Petersen84] Petersen (Carl S.). – Adaptive contouring of three-dimensional surfaces. *Computer Aided Geometric Design*, vol. 1, 1984, pp. 61–74.
- [Pfeifle et al.94] Pfeifle (Ron) et Seidel (Hans-Peter). – *Fitting Triangular B-splines to Functionnal scattered data*. – Rapport technique n° 3, Universität Erlangen, 1994.
- [Piegl85] Piegl (Les). – Representation of quadric primitives by rational polynomials. *Computer Aided Geometric Design*, vol. 2, 1985, pp. 151–155.
- [Pratt87] Pratt (Vaughan). – Direct least-squares fitting of algebraic surfaces. *Computer Graphics (SIGGRAPH '87 Proceedings)*, éd. par Stone (Maureen C.), pp. 145–152. – July 1987.

- [Preux91] Preux (A.). – *Le CSG et le temps réel*. – Thèse, Université de Lille I, September 1991.
- [Rogers et al.90] Rogers (David F.) et Adams (J. Alan). – *Mathematical elements for computer graphics*. – McGraw-Hill, 1990.
- [Schweizer87] Schweizer (Philippe). – *Infographie II*. – Presses Polytechniques Romandes, 1987.
- [Sederberg et al.85] Sederberg (T.) et Anderson (D.). – Steiner surface patches. *IEEE Computer Graphics and Applications*, vol. 5, n° 5, 1985, pp. 23–36.
- [Sederberg et al.89] Sederberg (T. W.) et Zundel (A. K.). – Scan line display of algebraic surfaces. *Siggraph 1989 Conference Proceedings*. ACM, pp. 147–156. – 1989.
- [Sederberg84] Sederberg (Thomas W.). – Planar piecewise algebraic curves. *Computer Aided Geometric Design*, vol. 1, 1984, pp. 241–255.
- [Sederberg85] Sederberg (Thomas W.). – Piecewise algebraic surface patches. *Computer Aided Geometric Design*, vol. 2, 1985, pp. 53–59.
- [Sederberg90] Sederberg (Thomas W.). – Techniques for cubic algebraic surfaces. *IEEE Computer Graphics & Applications*, vol. 10, n° 5, 1990, pp. 12–21.
- [Seidel94] Seidel (Hans-Peter). – *Polar forms and triangular B-spline surfaces*. – Rapport technique, Erlangen University, Germany, 1994.
- [Spencer94] Spencer (Andrew). – *Ecstatica*. – Jeu vidéo, 1994.
- [Teller et al.91] Teller (S. J.) et Séquin (C. H.). – Constructing easily invertible Bézier surfaces that parameterize general quadrics. *Proceedings of the symposium on solid modelling foundations and CAD/CAM applications*. ACM, pp. 61–69. – ACM Press, 1991.
- [Velho90] Velho (Luiz). – Adaptive polygonalization of implicit surfaces using simplicial decomposition and boundary constraints. *Eurographics '90*, éd. par Vandoni (C.E.) et Duce (D.A.). Eurographics Association, pp. 125–136. – 1990.
- [VK93] Van Kleij (Reinier). – *Display of solid models with quadratic surfaces*. – Thèse de PhD, Faculty of Technical Mathematics and Informatics, Delft University of Technology, The Netherlands, 1993.
- [Watkins et al.88] Watkins (M. A.) et Worsey (A. J.). – Degree reduction of Bézier curves. *Computer Aided Design*, vol. 20, n° 7, September 1988, pp. 398–405.
- [Witkin et al.94] Witkin (Andrew P.) et Heckbert (Paul S.). – Using particles to sample and control implicit surfaces. *Siggraph 1994 Conference Proceedings*. ACM, pp. 269–277. – 1994.

