



**HAL**  
open science

# Contribution à l'intégration des surfaces gauches dans les modèles de solides

Myriam Sahnoune

► **To cite this version:**

Myriam Sahnoune. Contribution à l'intégration des surfaces gauches dans les modèles de solides. Informatique [cs]. Université Paul Verlaine - Metz, 1990. Français. NNT : 1990METZ017S . tel-01775882

**HAL Id: tel-01775882**

**<https://hal.univ-lorraine.fr/tel-01775882v1>**

Submitted on 24 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Laboratoire de Recherche en Informatique de Metz

# THESE

présentée à

L'UNIVERSITE DE METZ

en vue de l'obtention du grade de

DOCTEUR DE L'UNIVERSITE DE METZ

(mention Sciences)

SPECIALITE INFORMATIQUE

par

Myriam SAHNOUNE

**"CONTRIBUTION A L'INTEGRATION DES SURFACES  
GAUCHES DANS LES MODELES DE SOLIDES"**

Soutenue le 20 décembre 1990 devant la commission d'examen

Messieurs

J.M . BRUN (rapporteur), Pdg de Coretech International

Y. GARDAN (Directeur de thèse), Professeur à l'Université de Metz

A . LE MEHAUTE (rapporteur), Professeur à l'Université de Lille

A . ROUX (examineur), Professeur à l'Université de Metz

M . VERON (examineur), Professeur à l'Université de Nancy

Laboratoire de Recherche en Informatique de Metz

# THESE

présentée à

L'UNIVERSITE DE METZ

en vue de l'obtention du grade de

DOCTEUR DE L'UNIVERSITE DE METZ

(mention Sciences)

SPECIALITE INFORMATIQUE

par

Myriam SAHNOUNE



**"CONTRIBUTION A L'INTEGRATION DES SURFACES  
GAUCHES DANS LES MODELES DE SOLIDES"**

Soutenue le 20 décembre 1990 devant la commission d'examen

Messieurs

J.M . BRUN (rapporteur), Pdg de Coretech International  
Y. GARDAN (Directeur de thèse), Professeur à l'Université de Metz  
A . LE MEHAUTE (rapporteur), Professeur à l'Université de Lille  
A . ROUX (examineur), Professeur à l'Université de Metz  
M . VERON (examineur), Professeur à l'Université de Nancy

BIBLIOTHEQUE UNIVERSITAIRE - METZ	
N° inv	19900385
Cote	SI13 90/17
Loc	Magasin

*<< ... Mais je dois rester sur le chemin que j'ai emprunté, si je ne fais rien, si je n'étudie pas, si je ne cherche pas, alors je suis perdu ... >>*

*Vincent Van Gogh*

*A maman ,*

## Remerciements

Ce travail a été réalisé au Laboratoire de Recherche en Informatique de Metz, sous la direction de Monsieur le Professeur Yvon Gardan. Je tiens à lui exprimer toute ma reconnaissance et toute ma gratitude pour la confiance qu'il m'a témoignée, pour son aide, son dynamisme et sa compétence dont il m'a fait profiter tout au long de ce travail.

Qu'il me soit permis de remercier Messieurs J.M Brun, Pdg de Coretech International et A Le Méhauté, Professeur à l'Université de Lille, pour avoir accepté de rapporter ce travail, pour les remarques pertinentes qu'ils m'ont adressées et leur participation à la commission d'examen.

Mes remerciements vont également à Messieurs A Roux, Professeur à l'Université de Metz et M. Véron, Professeur à l'Université de Nancy, pour avoir accepté de juger mon travail.

Je suis très reconnaissante à Jean-Pierre, Maître de conférence à l'Université de Metz, pour l'intérêt qu'il a porté à mon travail et pour avoir assuré une lecture attentive du manuscrit.

Je tiens également à remercier Jean-Pierre et Robin pour avoir réalisé les photographies qui illustrent mon travail.

Je ne saurais oublier les autres membres du laboratoire, qui m'ont gentiment et si patiemment supportée tout au long de la réalisation de ce manuscrit.

## TABLE DES MATIERES

INTRODUCTION .....	11
CHAPITRE I Apport d'une approche mathématique dans la modélisation des solides .....	14
I . 1 Introduction .....	14
I . 2 Principes généraux .....	14
I . 2 . 1 Domaines d'application .....	15
I . 2 . 2 Caractérisation mathématique des solides .....	15
I . 2 . 3 Passage au modèle informatique .....	16
I . 2 . 4 Autres contraintes sur le modèle informatique .....	17
I . 3 Principaux modèles de solides utilisés en C.A.O .....	18
I . 4 Propriétés des modèles informatiques .....	23
I . 4 . 1 La représentation par les limites (B-Rep) .....	23
I . 4 . 2 La représentation par arbre de construction (CSG : Constructive Solid Geometry) .....	25
I . 4 . 3 Boîtes et arbres octaux .....	26
I . 4 . 4 La représentation par extrusion .....	27
I . 4 . 5 La représentation par paramétrage .....	28
I . 4 . 6 Conversion entre modèles .....	28



I . 5	Les difficultés liées au passage modèle mathématique / modèle informatique .	29
I . 5 . 1	Validité .....	29
I . 5 . 1 . 1	<i>Les différentes approches</i> .....	30
I . 5 . 1 . 2	<i>Validité par construction</i> .....	31
I . 5 . 1 . 3	<i>Contrôle de validité</i> .....	32
I . 5 . 2	Intérêt de la non unicité .....	35
I . 5 . 3	Mise en oeuvre des opérations booléennes sur les modèles par les limites	36
I . 5 . 4	Evaluation de propriétés sur un arbre de construction .....	37
I . 6	Conclusion .....	41
CHAPITRE II	Approche critique des modèles de surfaces ...	43
II.1	Définition d'un carreau de Bézier .....	43
II. 1. 1	Définition des polynômes de Bernstein .....	43
II. 1. 2	Définition d'un carreau de Bézier .....	44
II. 2	Définition d'une surface B-spline .....	45
II. 2. 1	Définition des fonctions de base .....	45
II. 2. 2	Définition d'une surface B-spline .....	47
II. 3	Définition d'une surface B-spline rationnelle .....	50
II. 4	Capacité de modélisation .....	52
II. 4. 1	Facilités de création de formes libres .....	52
II. 4. 2	Approximation et interpolation .....	54
II. 4. 3	Modélisation de quadriques .....	55
II. 4. 4	Raccordement .....	56
II .5	Manipulation de surfaces .....	56
II. 5. 1	Algorithmes fondamentaux .....	56
II. 5. 1. 1	<i>Algorithmes pour les courbes de Bézier</i> .....	57
II. 5. 1. 2	<i>Algorithmes pour les courbes B-splines</i> .....	59

II. 5. 2 Applications usuelles .....	63
II.5. 2. 1 <i>Echantillonnage d'une surface</i> .....	63
II.5. 2. 2 <i>Echantillonnage des normales</i> .....	66
II.5. 2. 3 <i>Subdivision et Approximation</i> .....	67
II. 5. 3 Conclusion .....	73
II.6 Conclusion .....	74
II. 6. 1 Bilan .....	74
II. 6. 2 Limitations communes .....	74
<b>CHAPITRE III Intégration des surfaces gauches</b>	
<b>dans un modèle de solides</b> .....	<b>77</b>
III .1 Introduction .....	77
III .2 Rappel des propriétés importantes des NURBS pour la modélisation des solides	77
III.2 .1 Définition paramétrique .....	77
III.2. 2 Produit tensoriel .....	78
III.2. 3 Réseau de contrôle .....	78
III.2. 4 Les quadriques .....	78
III .3 Différentes approches d'intégration des surfaces gauches dans les modèles de solides	79
III.3. 1 Approche B-Rep .....	79
III.3.1. 1 <i>Mise en oeuvre d'approximations</i> .....	79
III.3.1. 2 <i>Assemblage de faces ou d'objets</i> .....	81
III.3.1. 3 <i>Mise en oeuvre de carreaux restreints</i> .....	82
III.3. 2 Approche CSG .....	85
III.3.2. 1 <i>Objets de base définis par des surfaces gauches</i> .....	86
III.3.2. 2 <i>Formes implicites</i> .....	87
III.3. 3 Conclusion .....	88
III .4 Le modèle solide choisi .....	88
III.4. 1 Définition de l'arbre CSG .....	88
III.4. 2 Implantation de l'arbre CSG .....	90

III .5	Modèle des solides de base .....	91
III.5. 1	Définition .....	91
III.5. 2	Construction .....	94
III.5. 3	Déformation .....	100
III .6	Evaluation de l'arbre .....	101
III.6. 1	Définition du Ray-Casting .....	102
III.6. 2	Intersection rayon/surface .....	103
III.6. 3	Intersection rayon/solide de base .....	108
III.6. 4	Intersection rayon/solide quelconque .....	113
III .7	Mise en oeuvre .....	117
III.7. 1	Création de l'arbre CSG .....	117
III.7.1. 1	<i>Définition du volume englobant d'une face</i> .....	117
III.7.1. 2	<i>Définition du volume englobant d'un solide de base</i> .....	118
III.7. 2	Définition des paramètres de vision .....	118
III.7. 3	Evaluation de l'arbre .....	119
III.7.3. 1	<i>Procédure rayon_solide</i> .....	119
III.7.3. 2	<i>Procédure rayon_surface</i> .....	120
III.7.3. 3	<i>Tableau de résultats d'intersection</i> .....	121
III.7. 4	Définition complète d'un solide quelconque .....	126
III.7. 5	Affichage du résultat d'une évaluation .....	126
III . 8	Améliorations .....	127
 <b>CONCLUSION</b> .....		 130
 <b>ANNEXE 1</b> <b>Représentation exacte d'une sphère à l'aide des Nurbs</b>		 134
 <b>ANNEXE 2</b> <b>Résolvante de deux équations polynomiales sous la forme d'un déterminant de Bezout</b> .....		 137

<b>ANNEXE 3</b>	<b>Recherche des racines d'un polynome par la méthode de Bairstow .....</b>	<b>139</b>
<b>ANNEXE 4</b>	<b>Génération des points de Bézier d'une surface B-Spline donnée .....</b>	<b>142</b>
<b>ANNEXE 5</b>	<b>Exemples de réalisation .....</b>	<b>146</b>
<b>BIBLIOGRAPHIE</b>	<b>.....</b>	<b>150</b>

# INTRODUCTION

La modélisation géométrique, même si elle ne permet pas de représenter toutes les caractéristiques d'un objet, reste l'un des outils fondamentaux des systèmes de C.F.A.O. Actuellement, le monde de la C.A.O s'intéresse plus particulièrement aux modèles solides. Dans ces types de modèles, les objets sont en théorie parfaitement connus.

Parallèlement aux recherches consacrées à la modélisation solide, se sont développés des systèmes fondés sur la modélisation des surfaces. Il faut constater que les travaux sur les surfaces se sont longtemps développés de manière totalement indépendante de ceux sur les solides.

On a ainsi abouti à deux grandes familles de modèles géométriques : les modèles solides et les modèles de surfaces, dont les fondements sont différents. En effet, les premiers sont essentiellement basés sur des structures adaptées, sans que les tentatives de formulation mathématique n'aient véritablement abouti, alors que les modèles de surfaces sont basés sur des représentations mathématiques. Les morceaux de surfaces ne sont pas reliés entre eux pour former des solides (aucune information topologique entre les carreaux n'est conservée). Mais de plus en plus, on aimerait avoir un modèle volumique avec des surfaces gauches. Ce qui permettrait de représenter des solides à forme libre.

Nous nous proposons de contribuer à l'intégration des surfaces dans les modèles de solides. Cette contribution se situe à deux niveaux :

- au niveau conceptuel et méthodologique, en essayant de considérer les problèmes d'intégration comme des questions fondamentales, pouvant trouver des familles de solutions aux propriétés démontrées et remettre en cause les approches mathématiques actuelles.
- au niveau de la réalisation en montrant sur une maquette combien les difficultés théoriques se compliquent de difficultés pratiques lors de la mise en oeuvre.

Après avoir rappelé les principes des différents modèles solides et leurs propriétés, nous montrons donc dans un premier chapitre les difficultés liées au passage d'un modèle mathématique à un modèle informatique. Malgré ces obstacles, nous donnons quelques éléments de réflexion, qui soulignent la nécessité d'une approche formelle de la modélisation des solides. Nous détaillons et comparons dans le deuxième chapitre les différents modèles de surfaces. L'orientation vers des modèles mathématiques (essentiellement paramétriques) permet de faire ressortir des propriétés, inconvénients et avantages, qui pourtant, n'apparaissent que rarement dans la littérature. Nous nous sommes en particulier attachés, à évaluer de manière précise l'apport et le rôle des formes rationnelles (Bézier rationnelles et NURBS). Cette étude nous a paru indispensable, tant des notions peu précises, voire des inexactitudes étaient diffusées. Puis, dans la dernière partie, nous nous attachons à montrer, à partir de l'existant, par quels moyens on peut tenter d'intégrer les modèles de surfaces dans les modèles solides. Les différentes approches, sont tout d'abord comparées, en essayant de donner une méthodologie pour aborder les grandes familles de solutions. Nous proposons ensuite une solution à ce problème et terminons par la présentation d'une maquette, que nous avons réalisée pour valider notre approche.

# CHAPITRE I





### I . 2 . 1 Domaine d'application

Nous ne considérons que la représentation des solides. Un solide respecte les conditions suivantes:

- rigidité : un solide a une forme invariante, quelles que soient sa position et son orientation (les distances et les angles algébriques sont conservés)
- finitude : un solide occupe une portion finie de l'espace;
- homogénéité : un solide a un intérieur, considéré comme homogène.
- un point de l'espace ne peut appartenir qu'à un solide au plus.

Un système de modélisation des solides doit également respecter des conditions sur les classes représentables et les opérations applicables ; en particulier :

- tout modèle constructible doit correspondre à un solide (validité);
- tout objet doit être représentable (puissance).

L'étude de la représentation des solides implique d'analyser les deux aspects fondamentaux de la modélisation:

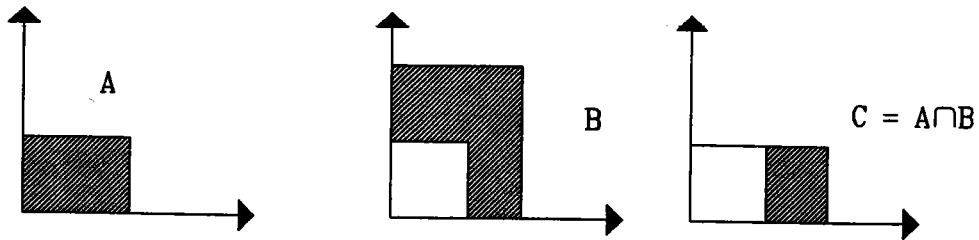
- la représentation des caractéristiques géométriques des objets;
- la mise en oeuvre d'opérations, en particulier des opérations de composition ou de modification de solides.

L'objectif de la modélisation est de constituer une représentation des objets qui soit utilisable pour la simulation (au sens large)...

### I . 2 . 2 Caractérisation mathématique des solides

D'après [REQ 77], les modèles de solides sont des classes de congruence de sous-ensembles de  $E^3$ , bornés, fermés, réguliers et semi-analytiques. Une classe de congruence est la collection de sous-ensembles de  $E^3$  qui peuvent être obtenus l'un de l'autre par des séquences de rotations et de translations.

Les solides étant caractérisés, un modèle doit respecter les conditions de validité exprimées en 1.1. Toute opération sur des solides doit être une loi de composition interne dans l'ensemble des solides mathématiques. Il est clair, par exemple, que les opérations booléennes ne respectent pas cette condition. En effet, elles ne sont pas régulières au sens où elles ne préservent pas la dimension et l'homogénéité des solides. Il est facile de montrer [GAR 83] que la propriété "régulier" n'est pas respectée.



L'intersection de A et de B a créé une arête isolée.

Il est donc nécessaire de définir des opérations booléennes dites régularisées, définies par:  
A et B étant deux solides,

$$A \cup^* B = r(A \cup B)$$

$$A \cap^* B = r(A \cap B)$$

$$A -^* B = r(A - B)$$

où  $r$  désigne l'application qui à un sous-ensemble de  $E^3$  associe l'adhérence de son intérieur.

### I . 2 . 3 Passage au modèle informatique

Soit une représentation informatique E définie, composée d'un ensemble de représentations syntaxiquement correctes. On définit une relation  $s$  de M (espace mathématique) dans E.

Le domaine de  $s$  est noté D et l'image de D par  $s$  est notée I.

On a:

$D \subset M$  : tous les objets ne sont pas modélisables  
dans la représentation mathématique définie

$I \subset E$  : toutes les représentations syntaxiquement  
correctes dans E ne sont pas forcément valides

On définit les qualités du modèle informatique:

Une représentation  $r$  de I est non ambiguë, si elle correspond à un seul objet.

Une représentation  $r$  de  $I$  est unique si l'objet correspondant n'admet pas d'autre représentation que  $r$ . C'est à dire  $s(s^{-1}(r)) = r$ . Le modèle  $I$  est unique si tous ses éléments sont uniques.

Une représentation est valide si elle appartient à  $I$ . La solution la plus simple pour assurer la validité serait que toutes les représentations syntaxiquement correctes soient valides ( $I=E$ ).

De nombreux modèles informatiques sont possibles. Se pose alors le problème de la cohérence entre ces modèles. On dira que deux relations de représentations  $s$  et  $s'$  sont équivalentes, si chaque image par  $s$  possède un équivalent  $r'$  par  $s'$ , et  $r$  et  $r'$  sont équivalents s'ils représentent le même ensemble d'objets dans  $M$ .

Deux relations de représentation non ambiguës sont équivalentes si et seulement si elles ont le même domaine.

Deux représentations  $r$  et  $r'$  sont cohérentes s'il existe au moins un objet  $m$  de  $M$  tel que  $s(m)=r$  et  $s'(m)=r'$ . Si les modèles ne sont pas uniques,  $m$  peut avoir plusieurs images dans  $E$  et  $E'$ . Si les modèles sont ambiguës, d'autres éléments de  $M$  peuvent avoir pour images  $r$  et  $r'$ .

Considérons, par exemple, un modèle ambigu, qui représenterait un solide par des vues de face et de dessus. Il est fondamental d'assurer la cohérence entre les vues de face et de dessus; ces deux représentations ne doivent cependant pas être équivalentes, ce qui tendrait à affirmer que deux objets ayant la même vue de dessus sont identiques.

## I . 2 . 4 Autres contraintes sur le modèle informatique

En admettant que l'on soit capable de définir les relations définies en I.2.3, il faut tenir compte, en plus, d'un certain nombre de qualités que l'on peut exiger d'un modèle informatique.

- la concision: cette notion, difficile à définir de manière précise, représente la quantité d'informations nécessaires pour représenter un objet. Elle n'est cependant pas totalement indépendante des traitements que l'on appliquera sur les modèles: une certaine redondance des informations sera ainsi souvent utile pour l'efficacité, voire l'applicabilité, de certains algorithmes.

- les facilités de mise en oeuvre d'algorithmes: cette qualité est liée aux algorithmes que l'on souhaitera mettre en oeuvre.

- l'ouverture: cette qualité n'a pas d'influence réelle sur le passage du modèle mathématique au modèle informatique.

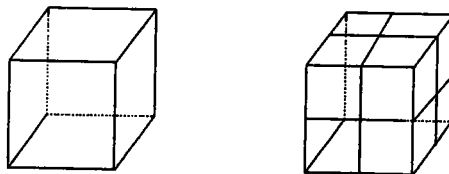
### I. 3 Les principaux modèles de solides utilisés en C.A.O

On peut considérer que les modèles géométriques utilisés couramment en CAO sont les suivants:

- le modèle par les limites ( ou B-Rep)
- le modèle par historique (ou arbre de construction ou CSG: Constructive Solid Geometry): plus récent, il est encore souvent limité aux opérations booléennes.

Ces différents modèles sont basés sur les notions suivantes :

. Le modèle par les limites (B-Rep: Boundary representation) [MIL 86]: C'est un modèle qui conserve les frontières de l'objet . Les solides sont décrits par un ensemble de faces, d'arêtes et points . Suivant le modèle par les limites choisi , les faces qui représentent un même objet peuvent être très différentes par leurs formes et par leurs nombres .Par exemple , de façon intuitive, on représente un cube par la réunion de six faces planes mais on peut également le définir de façon tout à fait valide par 24 faces.



Les faces des différents modèles B-Rep respectent toutes un même ensemble de règles énoncées par A.A.Requicha [MOR 85]

- 1 - Un nombre fini de faces définissent la frontière d'un solide.
- 2 - Une face est un sous-ensemble de la frontière du solide.
- 3 - L'union de toutes les faces d'un objet définit sa frontière.
- 4 - Une face est elle-même un sous-ensemble ou une région limitée d'une autre surface.
- 5 - Une face doit avoir une aire finie non nulle.

On pourrait résumer ce modèle en disant que le système conserve la "peau" de l'objet et sait, dans la modélisation solide, de quel côté est la matière. Un tel modèle comprend en général des informations géométriques (coordonnées des sommets, équations de faces...), des informations topologiques (façon dont sont reliées les informations géométriques) et des informations annexes (couleurs des faces...). Il suffit par artifice d'orienter le sens de parcours des contours limitant les faces pour distinguer intérieur et extérieur (suivant la règle de MOEBIUS : la matière est à gauche quand on parcourt le contour dans le sens positif). On emploie également les termes de modèles par les frontières ou par les bords.

Pour représenter des polyèdres, par exemple, on pourra, ainsi, se contenter de conserver:

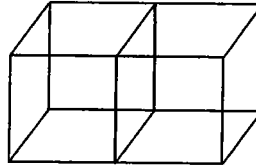
- pour la géométrie : des coordonnées (X, Y, Z) des sommets,
- pour la topologie : les arêtes en fonction des sommets et les faces en fonction des arêtes constituant leur contour, avec un sens d'orientation.
- en annexe : la couleur de chaque face.

En réalité, d'autres relations topologiques sont conservées dans le modèle, pour faciliter les traitements : par exemple, le fait que deux faces aient été déduites d'une même surface ("par facettisation"), peut être connu par le degré de continuité de l'arête et de la relation "faces adjacentes à cette face". Des outils puissants de manipulation de ces modèles, comme les opérateurs d'Euler [WIL 85], permettent d'assurer la cohérence topologique.

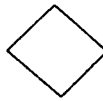
De nombreux modèles sont basés sur des faces planes. Quelques modèles manipulent des faces de type quadriques [BRAI 75] : l'introduction de quadriques ne modifie pas fondamentalement le modèle, dans la mesure où l'équation des faces peut être conservée. Cependant les algorithmes d'opérations booléennes se révèlent plus complexes à mettre en oeuvre. En effet, une opération booléenne sur un modèle B-Rep implique une nouvelle évaluation des frontières du solide résultat. Cette évaluation triviale dans le cas d'un modèle polyédrique est déjà beaucoup plus difficile avec des quadriques, elle nécessite le calcul de l'intersection entre 2 quadriques [LEV 79].

- . Le modèle par arbre de construction (CSG : Constructive Solid Geometry) [REQ 83]: est une méthode de construction de solide qui définit un solide complexe comme étant le résultat de la composition de solides de bases (primitives) ou de demi-espaces. Les opérations booléennes sont utilisées pour définir la composition [BOY 79]. La méthode par arbre de construction inclut les concepts suivants :

- Opérations booléennes régularisées : c'est à dire que l'on utilise des opérations booléennes qui préservent la dimension et l'homogénéité du solide [TIL 80]. On veut pouvoir affirmer que la composition de solides reste un solide. Par exemple l'intersection de 2 cubes se touchant par une face est égale à cette face. Dans le cas d'une intersection régularisée, l'intersection est égale à l'ensemble vide.



intersection  
non régularisée =



intersection régularisée =



- Classification d'un point par rapport à un objet (point -membership classification) : il s'agit pour tout point de l'espace de savoir s'il est HORS , SUR , ou DANS l'objet considéré . Le principe de classification est une méthode couramment employée pour évaluer un modèle CSG .
- Evaluation des frontières ( boundary evaluation ) : L'évaluation des frontières d'un arbre de construction est la conversion d'une représentation CSG en une représentation par les frontières (B-Rep) [PER 88]. On distinguera trois types de modèles B-Rep :

**\*B-Rep polygonal** : Toutes les faces sont planes; dans ce cas les primitives de l'arbre de construction sont approchées par des polyèdres et la conversion revient à effectuer des opérations booléennes sur des polyèdres. Quoique simple dans le principe, ce problème présente plusieurs difficultés: les très nombreux cas particuliers, les incohérences topologiques et les imprécisions numériques.

**\*B-Rep avec des quadriques [LEV 79]** : Levin montre comment déterminer la courbe intersection entre 2 quadriques et comment trouver une représentation paramétrique de cette courbe. Il devient alors possible de représenter une portion de quadrique et les objets enveloppés par ce type de surface.

\*B-Rep avec des " patches " : les faces sont des surfaces en général bicubiques. Deux méthodes ont été proposées pour effectuer les opérations booléennes sur de telles représentations [CAR 82], [CAS 87]. Ces méthodes testent l'intersection de tous les couples de patches. Elles utilisent toutes les deux des approximations des courbes intersections. Elles diffèrent dans la représentations des intersections. Carlson utilise des quadtree tandis que Casale définit la notion de carreaux restreints. Cette approche sera détaillée dans le chapitre trois.

Ce modèle est appelé ainsi parce qu'il peut être représenté par un arbre, bien qu'un modèle plus général soit de type hiérarchique. En général, on trouve aux feuilles de l'arbre des objets primitifs paramétrables et aux noeuds des opérations. Les noeuds sont en général des opérations booléennes (union, intersection, différence) ou des transformations solides (translation, homothétie, rotation). A chaque noeud correspond un objet, même si celui-ci n'est pas réellement "calculé". En fait, plutôt que de parler d'arbre de construction, il vaudrait mieux parler de "conservation de l'historique". Il y a une volonté de conserver une information "générique". Ces modèles ont été introduits depuis relativement peu de temps dans les systèmes de CAO. Ils sont assez souvent limités aux opérations booléennes et ne prennent pas forcément en compte tous les types d'objets.

Un arbre de construction peut avoir pour feuilles des demi-espaces : un demi-espace est défini par une relation qui partitionne l'espace en deux demi-espaces possédant une frontière commune :

$$D = \{x,y,z \in E^3 / f(x,y,z) \geq 0\}$$

L'opération fondamentale, qui consiste à établir l'appartenance d'un point à un solide, est alors extrêmement simple au niveau des feuilles, dans la mesure où il suffit de reporter les coordonnées du point dans l'équation  $f(x,y,z)$  : si la valeur obtenue est positive, le point appartient au demi-espace considéré, sinon, le point appartient au complémentaire du demi-espace considéré [GAR 87].

On peut également facilement décrire les objets de base tels que parallélépipèdes, cylindres, cônes... à l'aide d'intersections booléennes de demi-espaces.



PRIMITIVES	EQUATIONS DES SURFACES
Parallélépipède	$X=0$ et $0 \leq Y, Z \leq 1$ $X=1$ et $0 \leq Y, Z \leq 1$ $Y=0$ et $0 \leq X, Z \leq 1$ $Y=1$ et $0 \leq X, Z \leq 1$ $Z=0$ et $0 \leq X, Y \leq 1$ $Z=1$ et $0 \leq X, Y \leq 1$
Cylindre	$Z=0$ et $X^2 + Y^2 \leq 1$ $Z=1$ et $X^2 + Y^2 \leq 1$ $X^2 + Y^2 = 1$ et $0 \leq Z \leq 1$
Cône	$Z=0$ et $X^2 + Y^2 \leq 1$ $X^2 + Y^2 - Z^2 = 0$ et $0 \leq Z \leq 1$
Sphère	$X^2 + Y^2 + Z^2 = 1$
Tore	$(X^2 + Y^2 + Z^2 + 1 - r^2) - 4(X^2 + Y^2) = 0$

Parmi les autres types de modèles dont l'importance réelle est relativement faible dans les systèmes industriels, on peut citer:

- l'approche spatiale: il s'agit de diviser l'espace en "petits" éléments et de spécifier pour un solide donné, pour chaque petit élément s'il est rempli ou non. Pour atteindre une bonne précision, il est nécessaire de diviser l'espace en éléments "suffisamment" petits. La modélisation par arbres octaux (Octree) permet de minimiser le nombre d'informations à conserver. Elle consiste à considérer des divisions de l'espace global en huit, puis chacun des sous-espaces en huit, jusqu'à ce que l'une des conditions suivantes soit remplie [AYA 85]:

- . le sous-espace considéré ne contient aucune partie de l'objet à modéliser: il est vide et il est donc inutile de continuer à le subdiviser;
- . le sous-espace considéré est complètement contenu dans l'objet modélisé : il est plein et il est inutile de continuer à le subdiviser;

. le sous-espace considéré est de la taille minimale admise (il correspond au "petit" élément): il est soit complètement dans l'objet (il est plein), soit complètement en dehors de l'objet (il est vide), soit partiellement dans l'objet. Dans ce dernier cas, on peut appliquer plusieurs stratégies, par exemple, considérer que si le volume du sous-espace considéré compris dans l'objet est supérieur à la moitié du volume total du sous-espace, il est considéré comme plein (sinon il est vide).

La modélisation spatiale est souvent utilisée pour faciliter et accélérer des algorithmes (arbres octaux, volumes englobants ...) [CLE 84].

- l'extrusion généralisée: on peut considérer que tout solide peut être décrit par une face qui se déforme en suivant une certaine trajectoire. Cette approche, facile à appréhender pour certaines catégories d'objets (tuyauteries, par exemple), reste difficile à mettre en oeuvre [MIN 91].

#### I . 4 Propriétés des modèles informatiques

Nous examinons dans cette partie les modèles évoqués précédemment, en montrant les fondements mathématiques utilisés et leurs implications réelles. Nous ne détaillons les propriétés que sur les modèles B-Rep et CSG.

##### I . 4 . 1 Représentation par les limites (B-rep)

Une telle représentation comprend des informations géométriques, topologiques et des informations annexes, n'ayant pas une influence directe sur la forme de l'objet, mais sur sa représentation (couleur, transparence...). Elle est non ambiguë et non unique.

Cependant, assurer la validité d'une représentation peut être un problème très complexe. La validité au sens topologique est actuellement bien prise en compte. En effet, la formule d'Euler définit les conditions entre les différents éléments:

$$S - A + F = 2(C - H) + R$$

avec:

- S : nombre de sommets
- A : nombre d'arêtes
- F : nombre de faces
- C : nombre de composantes connexes
- H : nombre d'anses
- R : nombre de trous (anneaux) dans les faces

On peut alors considérer la relation d'Euler comme l'équation d'un hyperplan  $h$  dans un espace de dimension 6, dont les axes sont  $\langle S, A, F, C, H, R \rangle$ . Par définition, un opérateur d'Euler est un vecteur non nul dans cet hyperplan.

Un tel six-uplet ne caractérise pas de manière unique un solide.

Cinq opérateurs suffisent pour réaliser toutes les opérations.

- *mvsf* : crée un solide initial composé d'un sommet et d'une face ("créer sommet, composante connexe, face", soit "make vertex, shell, face");
- *mev* : crée une arête à partir d'un sommet existant et d'un nouveau sommet ("make edge, vertex");
- *mef* : éclate une face existante avec une nouvelle arête entre deux sommets existants, et crée une nouvelle face ("make edge, face");
- *kemr* : éclate une face existante en deux composants en supprimant une arête ("kill edge, make ring");
- *kfmrh* : compose deux faces, de telle sorte que la frontière de l'une devienne un trou (anneau) dans l'autre ("kill face, make ring, hole").

Pour une utilisation plus facile, on implante un plus grand nombre d'opérateurs.

Les principaux intérêts de la construction d'un modèle par des opérateurs d'Euler sont:

- toute représentation d'un objet par les frontières (valide) peut être construite par des opérateurs d'Euler;
- toute représentation construite par les opérateurs d'Euler satisfait toutes les conditions de validité topologique;

L'implantation informatique est faite sous la forme d'un graphe FAS (Faces, Arêtes, Sommets), quelle que soit la structure réelle employée.

La validité géométrique n'est pas du tout prise en compte par les opérateurs d'Euler.

La validité topologique et/ou géométrique peut être assurée soit :

- à la construction : les algorithmes mis en oeuvre lors de la constitution du modèle doivent donc assurer cette validité.
- a posteriori : il s'agit d'un contrôle de validité.

On verra en I.5 quelles méthodes peuvent être mises en oeuvre pour assurer la validité.

On conçoit que cette représentation, intéressante car elle permet de représenter de nombreux objets et parce qu'elle est assez proche d'opérations de visualisation est très éloignée des considérations mathématiques qui ont pu être faites auparavant.

#### I . 4 . 2 La représentation par arbre de construction (CSG:Constructive Solid Geometry)

Cette représentation conserve, pour un objet donné (solide), les opérations qui ont été mises en oeuvre pour le définir. En général limitée aux opérations booléennes, elle pourrait cependant inclure d'autres opérations (extrusions...).

La structure est un arbre binaire, dont les noeuds sont des opérations booléennes ou des transformations affines et les feuilles sont soit des demi-espaces, soit des objets primitifs.

Considérons tout d'abord un arbre dont les noeuds sont des objets primitifs. Les bases mathématiques qui régissent une telle modélisation sont faciles à décrire:

$$\langle \text{arbre} \rangle := \langle \text{objet primitif} \rangle / \langle \text{arbre} \rangle \langle \text{op. booléenne} \rangle \langle \text{arbre} \rangle / \\ \langle \text{arbre} \rangle \langle \text{op. de transformation} \rangle \langle \text{arg. de déplacement} \rangle$$

La famille de solides représentée par cet arbre peut être modélisée par la grammaire :

<solide> := <primitive>/<solide composé>

<solide composé> := <solide composé>/<opérateur> <solide>

<opérateur> := / U\*/-\*/translation/rotation/...

<primitive> := <cube>/<cylindre>/<sphère>/<cône>/...

Un arbre de construction est non ambigu, mais non unique. Le domaine dépend des objets primitifs et des opérateurs. Dans la mesure où les objets primitifs sont valides et que les opérateurs sont des lois de composition internes, la représentation est toujours valide. On peut cependant remarquer qu'une telle représentation est purement théorique. Les problèmes réels vont se poser lorsqu'elle devra être évaluée.

Lorsqu'il faut évaluer une telle représentation, on utilise deux grandes classes de méthodes:

- soit on évalue l'objet dans un autre modèle, en général par les limites: il "suffit", en partant des objets primitifs, dont on peut connaître les limites, d'appliquer les opérateurs jusqu'à l'évaluation de l'ensemble de l'arbre (ou de sous-arbres). Les difficultés sont identiques à celles rencontrées dans la représentation par les limites en I.4.1 . Lorsque l'objet est connu par ses limites, on peut lui appliquer les algorithmes de visualisation (élimination des parties cachées...), de calculs (volume...), etc...

- soit on met en oeuvre des algorithmes spécifiques à l'évaluation d'une opération donnée. Par exemple, si l'on souhaite visualiser l'objet, on pourra effectuer un lancer de rayons, chaque rayon correspondant à un pixel de l'écran.

Lorsque les feuilles sont des demi-espaces, la validité devient pratiquement impossible à démontrer. A priori très intéressant, ce type de modélisation se heurte à des difficultés insurmontables, aujourd'hui. On pourra toujours mettre en oeuvre une telle représentation, mais sans pouvoir, dans de nombreux cas prouver sa validité.

### I . 4 . 3 Boîtes et arbres octaux

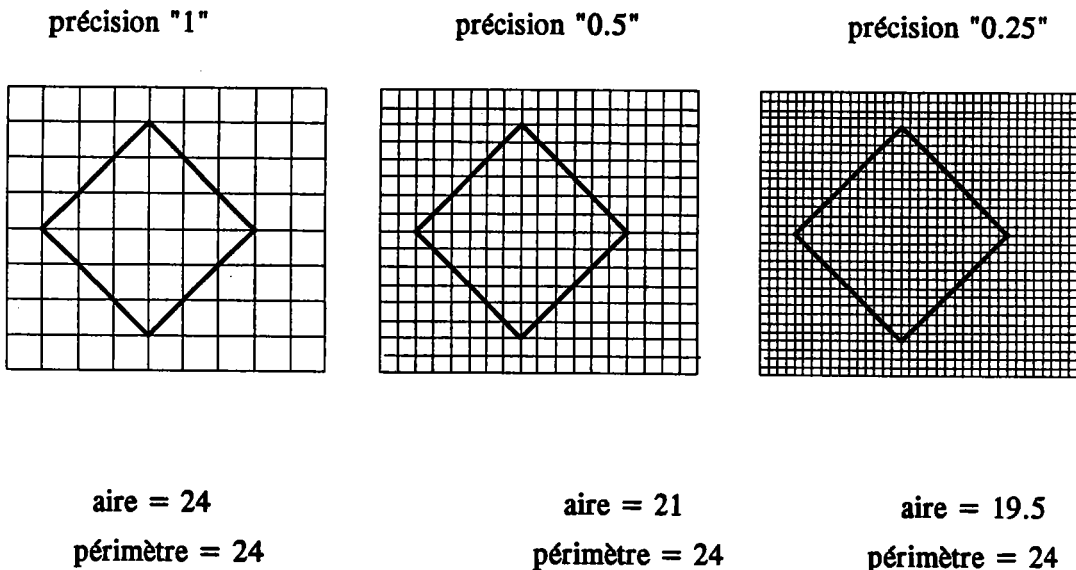
Ce type de représentation considère une partition de l'espace en petites boîtes. La représentation par arbre octal n'est qu'une amélioration, au sens place occupée, de la représentation par boîtes. Elle peut être considérée comme un cas particulier de l'arbre de construction, dans lequel les feuilles seraient des boîtes et les noeuds l'opérateur de collage.

Une telle représentation est unique à un niveau de précision donné, non ambiguë et la validité est facile à prouver. Malheureusement, la concision n'est pas respectée et l'est d'autant moins que les boites sont petites (l'arbre octal est une méthode d'amélioration de la concision).

La mise en oeuvre d'opérations booléennes est immédiate.

D'autres calculs seront extrêmement faciles à décrire (le volume, par exemple). En revanche, la surface en 3D (respectivement le périmètre en 2D) n'est pas calculable directement (il faut appliquer des algorithmes plus complexes), ce qui apparaît dans l'étude mathématique de cette modélisation.

EXEMPLE: en 2D, la représentation par arbre octal ne permet pas de faire converger le calcul du périmètre, en prenant, sans précautions, le nombre de cotés sur le bord:



Le calcul de l'aire converge, pas celui du périmètre.

#### I . 4 . 4 Représentation par extrusion

Les extrusions les plus couramment utilisées sont la translation et la révolution. Le domaine est donc limité aux classes d'objets qui peuvent être construits par ces méthodes. La validité d'une représentation peut être assurée, par validation des paramètres d'extrusion et du contour ou des contours utilisés.

Dans le cas des extrusions généralisées (à partir d'une face se déformant et décrivant une trajectoire), la théorie des espaces fibrés peut s'appliquer. Ces représentations sont difficiles à manipuler et la validité d'une représentation peut difficilement être prouvée.

#### I . 4 . 5 Représentation par paramétrage

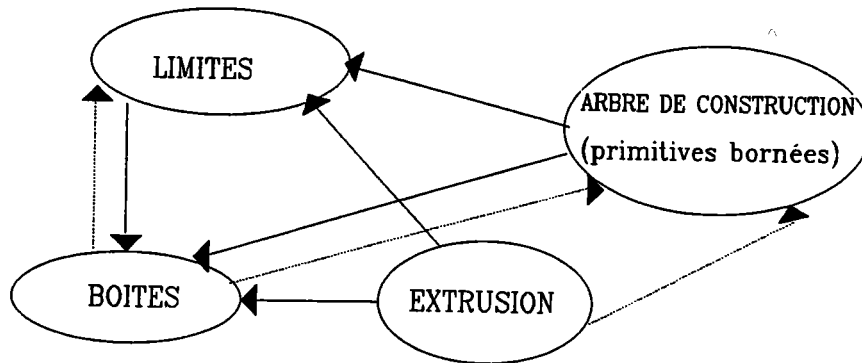
Un objet est représenté par un n-uplet, représentant la famille à laquelle il appartient et la liste de paramètres le décrivant. Une telle représentation peut se comparer à l'arbre de construction, en considérant que l'on se contente de modéliser les caractéristiques de l'objet. Dans la mesure où les familles sont connues et les domaines de validité des paramètres précisés, elle est valide et non ambiguë. Cependant l'évaluation doit être mise en oeuvre pour effectuer des opérations telles qu'opérations booléennes (sauf si on les intègre dans un arbre de construction dont les feuilles peuvent être des objets paramétrés quelconques), visualisation, calculs...

Pour évaluer ces objets, on pourra, comme dans l'arbre de construction, soit déterminer une représentation par les limites et lui appliquer tout algorithme valide, soit mettre en oeuvre un algorithme par classe d'opération.

#### I . 4 . 6 Conversions entre modèles

Aucun modèle ne permet actuellement, de prendre en compte l'ensemble des objets que l'on rencontre en CAO. Il est donc nécessaire d'implanter des algorithmes de passage d'un modèle à un autre. Il est clair, que s'il est possible (pas forcément très facilement) de passer d'un modèle par arbre de construction à un modèle par les limites, l'inverse est généralement impossible. Même si certaines méthodes d'inversion peuvent, dans certains cas, être mises en oeuvre, elles ne respecteront pas obligatoirement la logique de l'opérateur qui a construit le modèle. Or, très souvent, c'est cette logique qui est importante.

## Passages entre modèles



FACILE —————

POSSIBLE - - - - -

### I.5 Les difficultés liées au passage MODELE MATHEMATIQUE/MODELE INFORMATIQUE

Dans cette partie, nous étudierons les problèmes liés au passage d'un modèle mathématique (théorique) à un modèle informatique (pratique). Des études théoriques ont été menées pour tenter de résoudre les problèmes posés (opérations booléennes, volume, constitution d'ensemble ...). Par exemple, les opérateurs d'Euler permettent de mieux appréhender les modèles et les opérations.

Nous allons nous intéresser à la modélisation des solides avec des surfaces gauches. Avant même d'avoir analysé les différents modèles de surfaces gauches possibles, il est évident que certaines notions induites par la définition des solides doivent être précisées .

Considérons quelques problèmes difficiles dans le cadre de la modélisation des solides, à titre d'illustration [GAR 90].

#### I . 5 . 1 Validité



### *I . 5 . 1 . 1 Les différentes approches*

La validité d'un modèle de solides est assurée par les conditions suivantes :

- rigidité : le solide décrit dans le modèle a une forme invariante, quelles que soient sa position et son orientation ;
- finitude : le volume du solide décrit dans le modèle doit être fini ;
- homogénéité : tout solide a un intérieur considéré comme homogène ; de cette condition découle qu'un point de l'espace ne peut être affecté qu'à un seul objet. Le fait que l'intérieur soit considéré comme homogène représente l'état actuel de la modélisation. Il est indéniable que cette condition devrait être affinée pour prendre en compte des matériaux non homogènes.

Ces conditions très générales sont indépendantes de la famille de modèle utilisée. Elles doivent être converties en conditions vérifiables du point de vue informatique, qui dépendront du modèle utilisé.

Ces conditions peuvent :

- soit être assurées pour un modèle donné : cela signifie que quelle que soit la fonction appliquée au modèle, on peut affirmer que si cette opération est correcte dans un domaine donné, le résultat sera correct avec certaines conditions de précision ; par exemple, si l'on utilise un modèle par les frontières, une fonction déterminant le volume, donnera le volume (et ne donnera jamais une valeur infinie, ce qui pourrait se produire si le solide était ouvert) ; la fonction "appartenance d'un point au solide" pourra être mise en oeuvre en utilisant la méthode de la demi-droite et en comptant le nombre d'intersections significatives (si le modèle n'assurait pas la validité, cette méthode ne pourrait pas être employée, dans la mesure où elle est basée sur le fait que l'objet sur lequel elle s'applique est par définition un solide).
- soit être assurées pour chaque fonction : dans ce cas, la validité du modèle n'est pas vérifiée, mais chaque fonction qui peut s'appliquer au modèle doit assurer la validité du résultat . Par exemple, un modèle CSG n'assure pas la validité de l'objet modélisé. Mais, si l'on évalue l'objet modélisé par les frontières, les opérations booléennes appliquées seront régularisées pour assurer la validité du modèle par les frontières. De même la fonction "appartenance d'un point au solide" devra être mise en oeuvre avec précaution pour éviter les erreurs (voir I.5.4)

On peut tenter d'assurer la *validité* d'un modèle de deux manières :

- *à la construction* : toute opération construisant ou modifiant un solide se charge de la validité. Par exemple, l'utilisation d'opérateurs d'Euler assure la validité topologique lorsque l'on met en oeuvre un modèle par les bords. On peut également assurer la validité d'un modèle par les bords d'un objet par révolution. Les opérations booléennes régularisées, qui permettent de composer des objets, peuvent assurer la validité du résultat.
- *a posteriori* : on considère le modèle d'un objet et on vérifie que ce modèle représente bien un solide. Cette vérification nécessite de déterminer des règles applicables aux modèles (de manière spécifique) correspondant aux règles générales de validité. L'équivalence entre règles appliquées et règles générales n'est, dans la plupart des cas, pas démontrée.

Nous ne considérons dans la suite que l'assurance de la validité du modèle, par construction, ou a posteriori, pour un modèle par les frontières. Certains problèmes algorithmiques de mise en oeuvre seront repris dans les parties suivantes.

#### I. 5 . 1 . 2 *Validité par construction*

La validité topologique peut être assurée en utilisant des opérateurs d'Euler. On peut toutefois constater que des problèmes de performance, voire de mise en oeuvre (par exemple la modélisation d'un tore, [MIN88]) existent. On peut donc utiliser des primitives qui ne vérifient pas la validité topologique. Cela implique que c'est la fonction de construction qui les utilise qui vérifie la validité (sinon, on est obligé d'effectuer une vérification a posteriori). Cette approche peut être défendable dans plusieurs cas de figures, par exemple :

- le concepteur maîtrise parfaitement les contraintes de construction des objets : la modélisation par des facettes planes d'une sphère peut ainsi être réalisée en utilisant des primitives ne vérifiant pas la validité. Cette modélisation peut d'ailleurs assurer non seulement la validité topologique, mais également très facilement les conditions géométriques.

- certaines méthodes peuvent impliquer des passages temporaires par des représentations non valides, sachant qu'à la fin de l'opération, le résultat sera valide. Ainsi certaines méthodes de mise en oeuvre d'opérations booléennes effectuent une construction progressive de l'objet, sans assurer que chaque représentation intermédiaire soit valide. Dans ce cas, non seulement les opérateurs d'Euler sont inutiles, mais ils seraient extrêmement contraignants.

### *1.5.1.3 Contrôle de la validité*

Certaines méthodes de contrôle de validité opérant sur des facettes planes ont été publiées. Elles donnent un ensemble de contraintes à vérifier sur le modèle. Par exemple, dans [EAS 84], les objets sont représentés dans un modèle par les frontières et les faces sont triangulaires, avec, en particulier, les conditions de validité suivantes :

- une face ne peut avoir d'intersection avec une autre face qu'en une arête commune ou un sommet commun (cette condition doit assurer la continuité homogène) ;
- il existe au moins un chemin d'arêtes reliant deux arêtes quelconques (cette condition vérifie la connexité, le solide étant considéré ici comme constitué d' un seul ensemble de faces) ;
- le lien de chaque sommet est un simple polygone convexe ; le lien d'un sommet S est défini comme l'ensemble des arêtes opposées au sommet S dans chaque triangle ayant S comme sommet (cette condition évite les objets "ouverts" et assure donc la finitude);
- les faces doivent être correctement orientées, ce qui peut être assuré en vérifiant que chaque arête est parcourue une fois dans un sens, une fois dans l'autre pour les deux faces qu'elle sépare (cette condition assure qu'un point est soit à l'intérieur, soit à l'extérieur du solide) ;

Pour aborder la vérification de la validité, il est sans doute préférable de distinguer la validité topologique et la validité géométrique. Cette approche permettra ainsi d'appliquer soit les deux familles de conditions (vérification intégrale), soit uniquement l'une des familles (par exemple, un modèle construit avec des opérateurs d'Euler sera vérifié du point de vue géométrique uniquement).

- *conditions topologiques* :

- a) une arête est partagée par deux faces et deux faces seulement ;
- b) une arête est parcourue une fois dans un sens, et une fois dans l'autre pour les deux faces la partageant ;
- c) les seuls lieux d'intersection des arêtes sont les sommets. C'est-à-dire qu'il y a un sommet à chaque extrémité d'arête.

- *conditions géométriques* :

- a) deux faces ne peuvent se couper qu'en une arête partagée ;
- b) une face ne peut "s'auto-intersecter" ;

Les conditions topologiques sont relativement faciles à vérifier, dans la mesure où cela se réduit à un problème de graphe.

Les conditions géométriques peuvent nécessiter des traitements complexes. Pour vérifier les conditions géométriques, il faut mettre en oeuvre des algorithmes de calcul d'intersection. Même si dans ce cas, il n'est pas nécessaire de déterminer l'intersection, mais uniquement le fait qu'il y ait intersection ou pas, cela peut être difficile et coûteux, lorsqu'il ne s'agit pas de faces planes. Le cas de "l'auto intersection" semble pouvoir être résolu de manière naturelle par une approximation lorsque les faces sont gauches : on approche les faces par des facettes planes (par triangularisation), puis on vérifie l'auto-intersection comme un problème d'intersection sur un ensemble de facettes planes.

```

fonction AUTOINTERSECTION (F : face gauche) : booléen ;
DEBUT
TRIANGULARISER(S,facteur_de_triangularisation) ;
    résultat : n triangles
    validité := vrai ;
    i:= 1 ;
    répéter
        j:=i
        répéter
            j := j + 1 ;
            si INTER(trianglei,trianglej) alors validité := faux ;
        jusqu'à (validité = faux )ou (j=n)
        i := i + 1 ;
    jusqu'a validité = (faux) ou( i=n)
    autointersection:=non validité
FIN

```

On peut noter que la vérification d'intersection de deux triangles T1 et T2 est très simple. En effet, il suffit de :

```

succès := faux ;
choisir et marquer un segment S de T1;
répéter
    si S coupe le plan P portant T2 alors
        calculer le point d'intersection I de S avec P ;
        si I appartient à T2 alors succès := vrai
        la vérification d'appartenance est très simple, s'agissant d'un convexe
        sinon choisir segment suivant non marqué de S ;
jusqua succès ou plus de segment de T1 non marqué ;
si non succès idem sur T2 ;

```

Toute la difficulté réside dans le choix du facteur de triangularisation. En effet, un mauvais choix peut introduire des erreurs, d'autant que l'on cherchera, pour des raisons évidentes de performance, à limiter la valeur de ce facteur. Des méthodes pour associer ce facteur à la courbure devraient être introduites.

Par ailleurs, il est important d'effectuer une vérification de validité de manière aussi locale que possible (en général, en fonction des modifications effectuées, on cherchera à localiser tous les éléments qui peuvent influencer sur la validité).

## I . 5 . 2 Intérêt de la non unicité

La non unicité d'un modèle n'est pas obligatoirement un inconvénient. Son intérêt peut être montré sur un modèle de type CSG.

L'arbre est généralement construit en suivant les opérations choisies par l'opérateur. Considérons un tel arbre :

$$S := T((AU*B) -*C)$$

où A,B,C sont des primitives et T une translation.

On pourra "redescendre" la translation à chaque primitive pour aboutir à l'arbre :

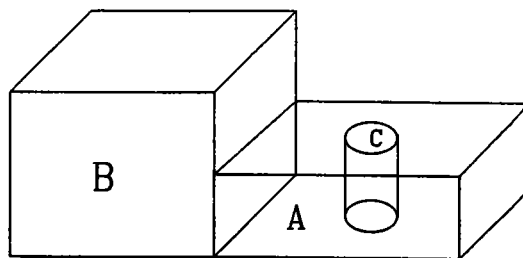
$$S := (T(A)U*T(B)) -*T(C)$$

Cette nouvelle formulation, équivalente à la précédente pourra être intéressante, car ainsi on peut appliquer une multiplication par une seule matrice sur chaque objet de base lors d'une évaluation par les limites.

On pourra également chercher à déterminer un arbre DSG (Destructive Solid Geometry [GAM90]) :

$$S := (((V-*comp_{V-B}(A))-*C$$

où V est le parallélépipède englobant A et B (ici il est facile à construire) et l'opération comp détermine le complémentaire d'une primitive par rapport à un volume parallélépipédique, dans l'exemple il est égal (V-B). L'arbre mis sous cette forme peut représenter (dans ce cas de manière volontairement simpliste) des opérations d'usinage à partir d'un brut V.



Enfin, on cherchera à déterminer une structure d'arbre qui soit bien adaptée à d'autres types de traitements. Par exemple, de manière générale, la détection d'objets nuls ou n'ayant aucun intérêt pour tel ou tel traitement, impliquera une remise en forme de l'arbre. Bien entendu, ces modifications posent le problème de la cohérence entre les différentes représentations, chacune ayant un objectif bien défini. La structure générique restera assez souvent celle qui correspond à la méthode de construction.

### I . 5 . 3 Mise en oeuvre des opérations booléennes sur des modèles par les limites

Les opérations ensemblistes sont l'un des moyens très utilisés pour construire des solides. Leur mise en oeuvre sur des modèles par des facettes planes est relativement difficile. Pourtant, leur définition théorique (mathématique) est très simple. On peut utiliser deux grandes classes de méthodes (qui ont en réalité de nombreux points communs), soit en utilisant les opérateurs d'Euler, soit en utilisant la méthode de Braid. Les méthodes basées sur les opérateurs d'Euler ne seront pas détaillées ici. Nous nous contentons de montrer les bases mathématiques de la méthode de Braid et les limites de cette méthode lors de son application informatique.

On considère deux classes de solides: les positifs et les négatifs:

un objet X positif est une application Q de  $\mathbb{R}^3$  dans  $\{0;1/2;1\}$ .

avec:

$$Q^{-1}(1) = \text{int}(X)$$

$$Q^{-1}(1/2) = \text{front}(X)$$

$$Q^{-1}(0) = \text{ext}(X)$$

Pour tout point q de  $\text{front}(X)$ , on note  $N(q,X)$ , le vecteur normal sortant.

Un objet X négatif est une application Q de  $\mathbb{R}^3$  dans  $\{0;-1/2;-1\}$ .

Les opérations sur les solides sont la négation et la somme. La négation d'un objet Q est définie par:

$$Q' = -Q$$

$$N(q,Q') = -N(q,Q)$$

En simplifiant, la somme est définie par la table:

+	1	1/2	0	-1/2	-1
1	1	1	1	1/2	0
1/2	1	c	1/2	b	0
0	1	1/2	0	0	0
-1/2	1/2	b	0	0	0
-1	0	0	0	0	0

Les valeurs b et c, dépendent de la position respective des deux objets. Si en un point de la frontière considérée (et en un voisinage de ce point), les normales sont de signes opposés, on a  $c=1$  et  $b=0$ , sinon  $b=c=1/2$ .

Une fois que l'on a exprimé ces notions mathématiques sur les objets, il ne reste plus ... qu'à étudier tous les cas possibles (et ils sont nombreux) que l'on peut rencontrer dans une opération d'union, intersection ou différence entre deux objets (la description algorithmique de ces cas pouvant remplir à elle seule un ouvrage dense): intersections arête/face...

#### I . 5 . 4 Evaluation d'une propriété dans un arbre de construction

L'arbre de construction a deux avantages:

- il représente un historique de construction;
- sa structure est facile à manipuler.

L'évaluation d'un tel arbre peut se faire pour visualiser l'objet, ou d'une manière générale pour déterminer une "vue" (au sens Bases de Données) de l'objet représenté par l'arbre de construction.

Etudier une propriété d'un objet représenté par un arbre de construction peut se faire de deux manières:

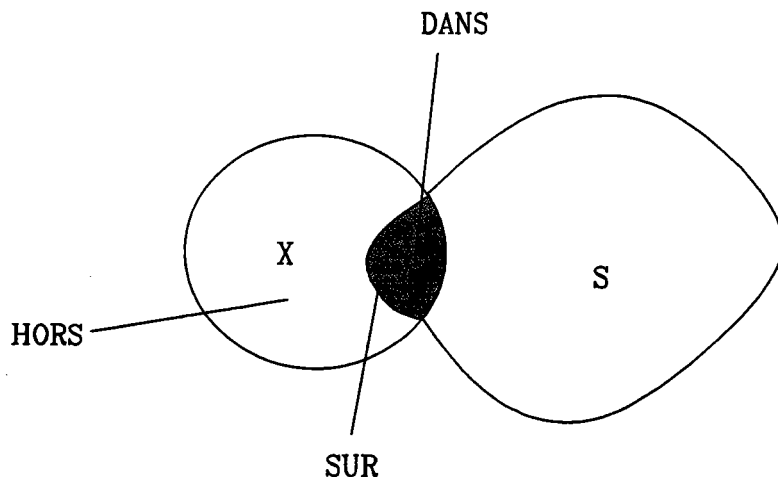


- on évalue l'objet dans une représentation connue (limites, par exemple), et on effectue le calcul de la propriété sur cette représentation. En général, on passe par une approximation, donc une perte d'informations, mais on peut appliquer les algorithmes qui ont été définis sur cette approximation, sans devoir réévaluer l'arbre. L'évaluation dans une représentation connue peut se faire, soit uniquement pour la racine de l'arbre (évaluation globale de l'arbre), soit pour chaque noeud (chaque noeud correspondant à un sous-arbre, donc à un objet). Cette seconde solution, plus coûteuse, peut permettre une plus grande souplesse dans les modifications.
- on applique le calcul de la propriété directement sur l'arbre de construction. c'est le cas du Ray-casting, pour une évaluation de type visualisation.

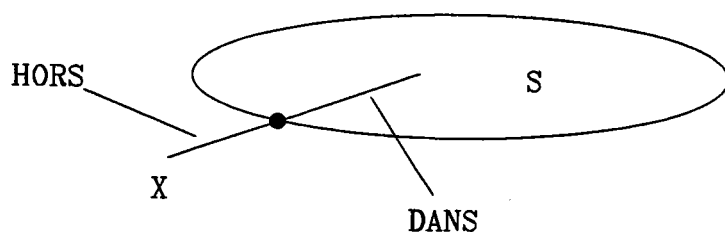
Pour effectuer une évaluation d'un arbre, on peut utiliser différentes méthodes. Nous présentons l'approche par classification par rapport à un ensemble, principalement développée à Rochester [REQ 85]. On peut étendre ce type de méthode à des algorithmes d'évaluation spécifique, comme le calcul d'une section, avec détermination des frontières analytiques.

Classer un ensemble  $X$ , par rapport à un autre ensemble  $S$ , consiste à déterminer quelles sont les parties de  $X$  dans  $S$ , sur la frontière de  $S$  et à l'extérieur de  $S$ .  $S$  est supposé régulier dans un sous-espace  $W$  de  $E$ ,  $X$  est régulier dans un sous-ensemble  $W'$  de  $W$ . On procède à une segmentation (partition) de  $X$  en sous-ensembles  $X$  dans  $S$ ,  $X$  sur  $S$  et  $X$  hors  $S$ .

#### EXEMPLE 1



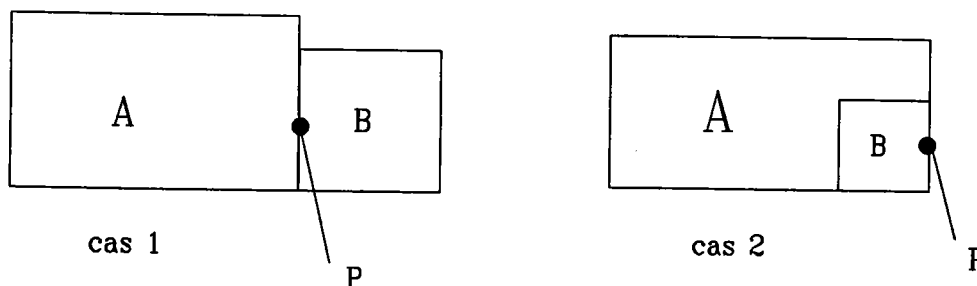
## EXEMPLE 2



remarque:  $X$  est de dimension 1.  $X$  sur  $S$  est réduit à un point, On régularise les sous-ensembles, en effectuant la fermeture de l'intérieur, ce qui donne  $X$  sur  $S$  nul (aucun sous-ensemble de dimension 0 sur  $S$ ).

Etant donnés deux ensembles  $A$  et  $B$ , dont la combinaison par une opération  $op$  est  $S$  ( $S = A op B$ ) et un sous-ensemble  $X$ , le partitionnement de  $X$  relativement à  $S$  se ramène à: connaissant ( $X$  dans  $A$ ,  $X$  sur  $A$ ,  $X$  hors  $A$ ) et ( $X$  dans  $B$ ,  $X$  sur  $B$ ,  $X$  hors  $B$ ), trouver ( $X$  dans  $S$ ,  $X$  sur  $S$ ,  $X$  hors  $S$ ). On retrouvera ici des notions utilisées dans le ray-casting. Cependant, ces dernières ne suffisent pas dans le cas d'ambiguïtés:

## EXEMPLE



Dans le cas 1,  $P$  est hors de  $A \cap^* B$ , dans le cas 2,  $P$  est sur  $A \cap^* B$

Pour traiter ces ambiguïtés, on introduit la notion de voisinage:

Soit  $S$  un sous-ensemble régulier de  $W$  et  $P$  un point de  $W$ , le voisinage régulier de rayon  $R$  supérieur à 0 de  $P$  par rapport à  $S$  est l'intersection régularisée avec  $S$  d'une boule ouverte de rayon  $R$  centrée en  $P$  :

$$N(P,S;R) = B(P;R) \text{ inter } S$$

Un point est dans  $S$  s'il existe un voisinage plein, sur  $S$  s'il existe un voisinage ni vide, ni plein et hors  $S$  s'il a un voisinage vide.

Pour trouver le voisinage de P dans l'exemple précédent, on effectue l'opération par rapport aux voisinages de P par rapport à A et B:

dans le cas 1 :

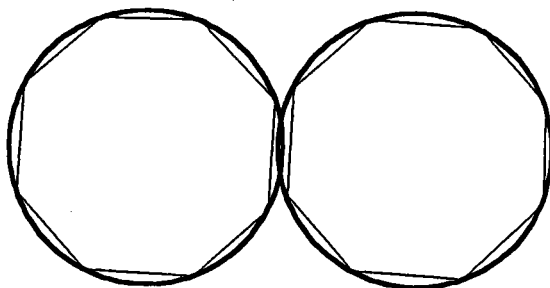
$$\left( \blacktriangleleft \cap \blacktriangleright \right)^* = \emptyset \quad , \text{ le point est hors de } A \cap B$$

dans le cas 2 :

$$\left( \blacktriangleleft \cap \blacktriangleleft \right)^* = \blacktriangleleft \quad , \text{ le point est sur } A \cap B$$

On pourra appliquer ces notions de voisinage à d'autres sous-ensembles, pour les classer: par exemple, une ligne peut être classée par rapport à un polygone, ou une face par rapport à un solide.

Ces notions de voisinages sont en réalité relativement difficiles à gérer en CAO. En effet elles rejoignent un problème fondamental lié à l'utilisation de l'ordinateur: celui-ci travaille avec une certaine précision et les erreurs qui peuvent se produire dans certains calculs peuvent conduire à des résultats faux (par exemple sur les cas particuliers, qui sont justement ceux traités par des notions de voisinages) [GAR 88]. Les approximations qui sont faites peuvent également entacher l'ensemble des résultats d'erreurs grossières:



Les deux cercles se coupent. Pourtant, s'ils sont approchés par des polygones (grossiers pour l'explication), on risque de trouver qu'ils sont disjoints.

## **I. 6 Conclusion**

Dans cette présentation, n'ayant aucune ambition d'exhaustivité, nous nous sommes contentés de faire ressortir un certain nombre de problèmes liés au passage mathématique/informatique. La résolution de ces problèmes est indispensable pour mettre en place une formalisation de la modélisation géométrique des solides.

On constate cependant une stagnation évidente dans les évolutions des modèles. Aucun modèle actuel ne paraît répondre à tous les besoins de la CAO. Il est donc nécessaire d'effectuer des conversions entre modèles (cf I.4.6). La prise en compte des surfaces gauches, comme frontières de solides se heurte à des difficultés très importantes. Cela tient sans doute au fait que les surfaces gauches correspondent à un modèle mathématique assez bien maîtrisé, contrairement aux modèles de solides. Les problèmes liés aux cas particuliers et à la précision des calculs (pour ne pas dire aux erreurs de calcul) nécessitent toujours des traitements au cas par cas, ce qui rend impossible la preuve des algorithmes implantés.

Ces quelques exemples montrent, s'il en était besoin, la nécessité de définir des modèles mathématiques et de mettre en oeuvre des modèles informatiques issus de ces modèles mathématiques.

# CHAPITRE II

## II.1 Définition d'un carreau de Bézier

### II.1.1 Définition des polynômes de Bernstein

Soit  $n$  un entier fixé  $m \geq 1$ , soit  $i$  dans  $[0, m]$

On appelle polynôme de Bernstein la quantité:

$$B_i^m(u) = C_m^i u^i (1-u)^{m-i}$$

où  $u$  varie dans l'intervalle  $[0, 1]$

La famille  $\{ B_0^m, \dots, B_m^m \}$  forme une base de l'espace vectoriel des polynômes de degré inférieur ou égal à  $m$

#### Propriétés des polynômes de Bernstein

$$1) \quad \sum_{i=0}^m B_i^m(u) = 1 \quad \forall u$$

$$2) \quad B_i^m(u) \geq 0 \quad \forall u, \forall i$$

$$3) \quad B_i^m(0) = 0 \quad \text{si } i > 0 \quad \text{et} \quad B_0^m(0) = 1$$

$$B_i^m(1) = 0 \quad \text{si } i < m \quad \text{et} \quad B_m^m(1) = 1$$

#### 4) Dérivation

$$\left[ B_i^m \right]'(u) = m \left[ B_{i-1}^{m-1}(u) - B_i^{m-1}(u) \right]$$

## 5) Récurrence

$$B_i^m(u) = u B_{i-1}^{m-1}(u) + (1-u) B_i^{m-1}(u)$$

### II.1.2 Définition d'un carreau de Bézier

On appelle carreau de Bézier la surface définie par

$$\begin{aligned} S : [0,1] \times [0,1] &\longrightarrow \mathbb{R}^3 \\ (u,v) &\longmapsto \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^m(u) B_j^n(v) \end{aligned}$$

où les  $P_{ij}$  sont des points de  $\mathbb{R}^3$ . L'ensemble des  $P_{ij}$  est appelé réseau ou polyèdre des points de contrôle.

La surface est de degré  $m$  en  $u$  et  $n$  en  $v$ . Par la suite nous dirons qu'elle est de degré  $m \times n$ . [BEZ 86]

#### Remarque : Courbe de Bézier

On appelle courbe de Bézier la courbe définie par

$$\begin{aligned} R : [0,1] &\longrightarrow \mathbb{R}^3 \\ t &\longmapsto R(t) = \sum_{i=0}^m P_i B_i^m(t) \end{aligned}$$

où les  $P_i$  sont des points de  $\mathbb{R}^3$ . Ils sont appelés points de contrôle de la courbe.

#### Propriétés des carreaux de Bézier

Les propriétés (1) et (2) des polynômes de Bernstein entraînent deux propriétés importantes des carreaux de Bézier

##### 1) Propriété de l'enveloppe convexe

pour tout couple  $(u,v)$ ,  $S(u,v)$  appartient toujours à l'enveloppe convexe du réseau de contrôle  $\{P_{ij}\}$

## 2) Invariance par changement affine

Soit  $f$  : une application affine de  $\mathbb{R}^3$  dans  $\mathbb{R}^3$  alors

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

la propriété 3) des polynômes de Bernstein entraîne

$$3) S(0,0) = P_{00}, S(1,0) = P_{m0}$$

$$S(0,1) = P_{0n} \text{ et } S(1,1) = P_{mn}$$

Le réseau et le carreau se touchent aux 4 coins.

4) Grâce à la relation (4), on montre facilement que:

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

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

## II.2 Définition d'une surface B-spline

### II.2.1 Définition des fonctions de base

Soit  $m \geq 1$  un entier fixé, soit  $k$  un entier compris entre 1 et  $(m+1)$ .

Soit une suite finie de nombres réels  $\{u_i\}_{i=0..m+k}$  telle que:

- $u_i \leq u_{i+1}$  pour tout  $i$
- $u_{k-1} < u_{m+1}$
- $u_i < u_{i+k}$  pour tout  $i$

Les  $u_i$  sont appelés les noeuds (ou valeurs nodales),  $\{u_i\}_{i=0..m+k}$  est appelé vecteur nodal (ou table des noeuds) et est noté  $tndu$ .

Si  $u_{i-1} < u_i = \dots = u_{i+r} < u_{i+r+1}$ , chacun des noeuds  $u_i \dots u_{i+r}$  est de multiplicité  $r$ .

D'après la définition de  $tndu$ , la multiplicité maximale d'un noeud est  $k$ .

Lorsque  $u_{i+1} - u_i = \text{constante}$  pour tout  $i$ , le vecteur nodal est dit uniforme.



- Lorsque
- \*  $u_0 = \dots = u_{k-1} = 0$
  - \*  $u_{i+1} - u_i = \text{constante}$  pour  $(k-1) < i < m$
  - \*  $u_{m+1} = \dots = u_{m+k}$

le vecteur nodal est dit quasi uniforme.

Les fonctions de base ou B-spline sont notés  $N_i^r$  où  $r$  est l'ordre de la B-spline et  $i$  varie de  $[0, m+k-r]$ .

On peut les définir suivant une relation de récurrence. [BOEH 84]

Soit  $u$  un réel quelconque

$$N_i^1(u) = 1 \text{ si } u \text{ dans } [u_i, u_{i+1}[$$

$$N_i^1(u) = 0 \text{ si } u \text{ n'appartient pas à } [u_i, u_{i+1}[$$

pour  $1 < r < k$

$$N_i^r(u) = \frac{u - u_i}{u_{i+r-1} - u_i} N_i^{r-1}(u) + \frac{u_{i+r} - u}{u_{i+r} - u_{i+1}} N_{i+1}^{r-1}(u)$$

**convention** : lorsqu'un des dénominateurs intervenant dans la formule est nul, le quotient correspondant est nul.

### Propriétés des fonctions de base

- 1) support:  $N_i^r(u) = 0$  si  $u$  n'appartient pas à  $[u_i, u_{i+r}[$
- 2) Positivité:  $N_i^r(u) > 0$  si  $u$  appartient à  $[u_i, u_{i+r}[$   
si  $u_i = u_{i+r}$   $N_i^r$  est identiquement nulle
- 3) Partition de l'unité: Si  $u$  appartient  $]u_{r-1}, u_{m+k-r+1}[$

$$\sum_{i=0}^{m+k-r} N_i^r(u) \equiv 1$$

4) Dérivation [BAR 88]:

$$\left( N_i^r \right)'(u) = (r-1) \cdot \left( \frac{1}{u_{i+r-1} - u_i} N_i^{r-1}(u) - \frac{1}{u_{i+r} - u_{i+1}} N_{i+1}^{r-1}(u) \right)$$

avec la même convention en cas de nullité du dénominateur

5) Ordre de continuité: une fonction de base  $N_i^k$  est  $(k-1-d_j)$  dérivable en un noeud  $u_j$  de multiplicité  $d_j$  ( $i < j < i+k$ ) ( $c$ -à- $d$  appartenant à  $]u_j, u_{i+k}[$ ).

6) Comportement des fonctions de base au voisinage d'un noeud de multiplicité maximale

Soit  $u_j$  un noeud de multiplicité  $k$ . On suppose que

$$u_{i-1} < u_i = u_{i+1} = \dots = u_{i+k-1} < u_{i+k}$$

$N_{i-1}^k$  vaut 1 en  $u_i^-$  et 0 en  $u_i^+$ ,  $N_i^k$  vaut 0 en  $u_i^-$  et 1 en  $u_i^+$ . Les autres fonctions de base s'annulent en ce noeud.

Si  $u_j$  n'est que de multiplicité  $k-1$ , par exemple si

$$u_{i-1} < u_i = u_{i+1} = \dots = u_{i+k-2} < u_{i+k-1}, \text{ alors } N_{i-1}^k \text{ vaut 1 en } u_i.$$

Les autres fonctions de base s'annulent en  $u_i$ .

## II.2.2 Définition d'une surface B-spline

Soient  $m, n$  deux entiers fixés  $m \geq 1$ ,  $n \geq 1$  et  $0 < k < m+2$  et  $0 < l < n+2$ .

Soient  $tnu = \{u_0, \dots, u_{m+k}\}$ ,  $tnv = \{v_0, \dots, v_{n+l}\}$  deux vecteurs nodaux fixés.

Soient  $(m+1) \times (n+1)$  points de  $\mathbb{R}^3$  notés  $P_{ij}$  pour  $i=0..m, j=0..n$ .

On appelle surface B-spline la surface définie par:

$$S : [u_{k-1}, u_{m+1}] \times [v_{l-1}, v_{n+1}] \longrightarrow \mathbb{R}^3$$

$$(u, v) \longrightarrow \sum_{i=0}^m \sum_{j=0}^n P_{ij} N_i^k(u) N_j^l(v)$$

$\{P_{ij}\}$  est le polyèdre (ou réseau) de contrôle de la surface.

Les  $N_i^k$  sont associés au vecteur nodal  $\{u_i\}$ , les  $N_j^l$  sont associés au vecteur nodal  $\{v_j\}$ .

La surface est dite d'ordre  $k \times l$ , d'ordre  $k$  dans la direction  $u$ , d'ordre  $l$  dans la direction  $v$ .

La surface est entièrement définie par  $m, n, k, l, t_{ndu}, t_{ndv}$  et le réseau  $\{P_{ij}\}$ .

**Remarque: courbe B-spline**

Soient un entier  $m > 0$  fixé et  $k$  un entier tel que  $0 < k < m+2$ .

Soient  $t_{ndu} = \{u_0, \dots, u_{m+k}\}$  un vecteur nodal fixé et  $\{P_i\}_{i=0 \dots m+1}$  une suite de points de  $\mathbb{R}^3$ . On appelle courbe B-spline la courbe définie par

$$R : [u_{k-1}, u_{m+1}] \xrightarrow{\quad} \mathbb{R}^3$$

$$t \xrightarrow{\quad} R(t) = \sum_{i=0}^m P_i N_i^k(t)$$

Les  $P_i$  sont appelés points de contrôle.

**Propriétés des surfaces B-spline.**

1) Polynômiale par morceaux.

Sur chaque pavé non vide  $[u_g, u_{g+1}] \times [v_d, v_{d+1}]$   $S(u, v)$  est un polynôme en  $(u, v)$  de degré  $(k-1) \times (l-1)$ .

2) Comportement local.

La propriété (1) des fonctions de base entraîne les résultats suivants:

Si  $(u, v)$  appartient à  $[u_g, u_{g+1}] \times [v_d, v_{d+1}]$ ,

$$S(u, v) = \sum_{i=g-k+1}^g \sum_{j=d-l+1}^d P_{i,j} N_i^k(u) N_j^l(v)$$

donc  $S(u, v)$  ne dépend que des points représentés par la matrice  $A_{gd}$ :

$$\begin{bmatrix} P_{g-k+1, d-l+1} & \dots & P_{g-k+1, d} \\ \vdots & & \vdots \\ P_{g, d-l+1} & \dots & P_{g, d} \end{bmatrix}$$

Inversement le point de contrôle  $P_{ij}$  n'a d'influence que sur le pavé  $[u_i, u_{i+k}] \times [v_j, v_{j+l}]$ .

### 3) Enveloppe convexe

Les propriétés (1) (2) et (3) des fonctions de base entraînent que la restriction de la surface à un pavé paramétrique  $[u_g, u_{g+1}[ \times [v_d, v_{d+1}[$  est entièrement incluse dans l'enveloppe convexe des points de la matrice  $A_{gd}$ .

### 4) Invariance par transformation affine.

Soit  $F$  une transformation affine de  $\mathbb{R}^3$  vers  $\mathbb{R}^3$ . La propriété (3) des fonctions de base entraîne que:

$$\forall (u,v) \quad F(\mathbf{S}(u,v)) = \sum_{i=0}^m \sum_{j=0}^n F(\mathbf{P}_{i,j}) N_i^k(u) N_j^l(v)$$

### 5) Dérivation

La forme des dérivées des fonctions de base nous permet de démontrer que:

$$\frac{\partial \mathbf{S}}{\partial u}(u,v) = (k-1) \sum_{i=1}^m \sum_{j=0}^n \frac{\mathbf{P}_{i,j} - \mathbf{P}_{i-1,j}}{u_{i+k-1} - u_i} N_i^{k-1}(u) N_j^l(v)$$

$$\frac{\partial \mathbf{S}}{\partial v}(u,v) = (l-1) \sum_{i=0}^m \sum_{j=1}^n \frac{\mathbf{P}_{i,j} - \mathbf{P}_{i,j-1}}{v_{j+1} - v_j} N_i^k(u) N_j^{l-1}(v)$$

Les dérivées partielles sont donc encore des surfaces B-spline dont les caractéristiques dépendent de celles de  $S$ .

### 6) Influence de l'ordre $k \times l$

Nous exprimons brièvement à l'aide d'un tableau, la manière dont évoluent les propriétés de la surface lorsque l'ordre croît.

ordre	caractère local	capacité de la surface à bien approcher son réseau de définition	ordre de continuité	régularité de la surface
↗	↘	↘	↗	↗

De manière générale l'élévation de l'ordre "tend" la surface.

## Remarques importantes

- 1) Supposons que les noeuds terminaux dans tndu et tndv soient de multiplicité maximale.  
C'est à dire que

$$u_0 = u_1 = \dots = u_{k-1} < u_k$$

$$v_0 = v_1 = \dots = v_{l-1} < v_l$$

$$u_m < u_{m+1} = u_{m+2} = \dots = u_{m+k}$$

$$v_n < v_{n+1} = v_{n+2} = \dots = v_{n+l}$$

alors la propriété (6) des fonctions de base entraîne que la surface passe par les quatre "coins" du réseau de définition, plus précisément nous avons:

$$S(u_{k-1}, v_{l-1}) = P_{00}$$

$$S(u_{m+1}, v_{l-1}) = P_{m0}$$

$$S(u_{k-1}, v_{n+1}) = P_{0n}$$

$$S(u_{m+1}, v_{n+1}) = P_{mn}$$

- 2) Lien avec les carreaux de Bézier

Si les conditions suivantes sont vérifiées:

$$* k = m+1$$

$$* l = n+1$$

$$* u_0 = u_1 = \dots = u_{k-1} < u_{m+1} = u_{m+2} = \dots = u_{m+k}$$

$$* v_0 = v_1 = \dots = v_{l-1} < v_{n+1} = v_{n+2} = \dots = v_{n+l}$$

Alors la surface B-spline est un carreau de Bézier de degré  $(k-1) \times (l-1)$ , ayant  $\{P_{ij}\}$   
 $i=0\dots m ; j=0\dots n$  pour réseau de définition.

### II.3 Définition d'une surface B-spline rationnelle

La définition de ce dernier type de surface nécessite l'introduction de deux fonctions mathématiques.

On appelle H la fonction :

$$\mathbb{R}^4 \longrightarrow \mathbb{R}^3$$
$$(x, y, z, t) \longmapsto H(x, y, z, t) = \left( \frac{x}{t}, \frac{y}{t}, \frac{z}{t} \right)$$

et T l'application :

$$\mathbb{R}^3 \longrightarrow \mathbb{R}^4$$
$$(x, y, z) \longmapsto T(x, y, z) = (x, y, z, 1)$$

Soit  $\{P_{ij}\}_{i=0\dots m; j=0\dots n}$  un réseau de points de contrôle en 3D. Soient deux tables de noeuds  $tnu = \{u_0, \dots, u_{m+k}\}$  et  $tnv = \{v_0, \dots, v_{n+l}\}$ .

A chaque point de contrôle  $P_{ij}$  est associé un poids  $\beta_{ij} > 0$

Soit  $S$  l'application :

$$\begin{aligned} [u_{k-1}, u_{m+1}] \times [v_{l-1}, v_{n+1}] &\longrightarrow \mathbb{R}^4 \\ (u, v) &\longmapsto \sum_{i=0}^m \sum_{j=0}^n \beta_{ij} T(P_{ij}) N_i^k(u) N_j^l(v) \end{aligned}$$

$S$  représente une surface B-spline polynômiale d'ordre  $k \times l$  dans l'espace homogène.

Une surface B-spline rationnelle est alors définie de la manière suivante :

$$\begin{aligned} \sigma : [u_{k-1}, u_{m+1}] \times [v_{l-1}, v_{n+1}] &\longrightarrow \mathbb{R}^3 \\ (u, v) &\longmapsto \sigma(u, v) = H(S(u, v)) \end{aligned}$$

En d'autres termes

$$\sigma(u, v) = \frac{1}{\sum_{i=0}^m \sum_{j=0}^n \beta_{ij} N_i^k(u) N_j^l(v)} \sum_{i=0}^m \sum_{j=0}^n \beta_{ij} P_{ij} N_i^k(u) N_j^l(v)$$

**Remarques :**

Un carreau de Bézier rationnel est défini suivant la même démarche : pondération des points de contrôle, puis définition d'un carreau de Bézier polynômial dans l'espace homogène et enfin projection de celui-ci dans l'espace euclidien.

Lorsque les vecteurs nodaux associés à une surface B-spline rationnelle sont non uniformes, celle-ci est appelée surface B-spline rationnelle non uniforme ou NURBS (pour Non Uniform Rational B-Spline).

## Propriétés des surfaces B-spline rationnelles

- 1) **Rationnelle par morceaux**  
Sur chaque pavé paramétrique non vide  $[u_i, u_{i+1}[ \times [v_j, v_{j+1}[$ , la surface est une fraction rationnelle en ses paramètres.
- 2) 3) 4) Les propriétés de comportement local, d'enveloppe convexe et d'invariance par transformation affine des B-spline polynomiales sont intégralement conservées par les B-spline rationnelles.

Les deux remarques importantes concernant les B-spline polynomiales sont encore valables pour les rationnelles car elles sont liées aux vecteurs nodaux.

## II .4 Capacité de modélisation

### II.4.1 Facilités de création de formes libres

Les méthodes que nous comparons sont toutes des moyens de création très interactifs, elles permettent assez facilement la création et la modification de surfaces quelconques.

#### \*Surfaces de Bézier

La plus simple des méthodes que nous décrivons est celle des carreaux de Bézier, elle permet d'obtenir des surfaces très régulières. La normale et les courbures calculées en chaque point ne varient jamais brusquement sauf dans le cas où le réseau de points de contrôle se recoupe. Si on veut réaliser une forme plus irrégulière ou plus travaillée, on est obligé d'augmenter le nombre de points de définition et donc d'élever le degré de la surface car une surface de  $(n+1) \times (m+1)$  points de contrôle est un polynôme de degré  $n+m$ .

L'association de plusieurs carreaux de Bézier permet cependant la création de formes complexes sans élever le degré du carreau. Cette solution soulève le problème du raccordement qui sera évoqué ultérieurement.

Il est également possible de modifier interactivement un ou plusieurs points de contrôle de la surface. La modification d'un point de contrôle entraîne une modification globale du carreau.

### \* Surfaces B-spline

Les difficultés de création de formes complexes peuvent être en partie résolues grâce à l'utilisation des fonctions B-splines uniformes. Une surface B-spline est en fait constituée d'un assemblage de carreaux polynomiaux dont l'indépendance croît avec l'éloignement. Grâce au degré, on contrôle l'ordre de continuité à la frontière de deux carreaux. Pour un ordre fixé de la surface, les points de définition peuvent être aussi nombreux que l'on veut. La création de surfaces très variées en est grandement facilitée. On peut remarquer que c'est le principal ajout des surfaces B-splines uniformes par rapport aux carreaux de Bézier.

Un point de définition n'influence qu'une partie de la surface. Son déplacement n'entraîne qu'une déformation locale.

Les fonctions B-spline non uniformes constituent un outil de modélisation plus riche. En effet, l'influence respective des différents points de contrôle peut varier et ainsi aider à modifier la surface résultante. Il est alors facile de créer des arêtes vives. On augmente l'influence d'une ligne de points de contrôle en répétant une valeur nodale dans une des tables de noeuds. Pour conserver au moins une continuité  $C^0$ , on ne peut répéter plus de (ordre-1) fois une même valeur nodale. La variation de l'influence d'une ligne de points de définition est quantifiée. Il est impossible de créer un point anguleux qui ne soit pas intersection de deux arêtes vives.

### \* Surfaces B-spline rationnelles

La capacité de modélisation des surfaces B-splines rationnelles est plus étendue encore. La pondération des points de contrôle, rendue explicite par définition, varie de 0 à plus l'infini. L'influence de chacun d'eux peut donc être fixée finement, permettant entre autres la création de points anguleux.

### **REMARQUE:** Multiplicité spatiale:

Il est toujours possible, quelque soit la modélisation choisie, de donner à plusieurs points du réseau la même position spatiale. Cela permet d'attirer la surface vers cette position. C'est un moyen de créer des carreaux dégénérés.



## II.4.2 Approximation et interpolation

L'interpolation et l'approximation d'un nuage de points par une surface sont des problèmes souvent rencontrés en C.A.O.

L'interpolation consiste à faire passer une surface par un ensemble donné de points de l'espace.

Le but de l'approximation est de définir une surface qui soit en moyenne la plus proche possible de l'ensemble de points. Les points donnés n'appartiennent pas nécessairement à la surface obtenue.

Les différents types de surfaces présentés offrent une solution naturelle au problème de l'approximation, ces surfaces constituent des approximations de leurs réseaux de contrôle.

### **\* Approximations et interpolation à l'aide de surfaces de Bézier**

Un carreau de Bézier est une approximation relativement grossière de son réseau de points.

On réalise une interpolation de Bézier en définissant autant de points de contrôle que de points à interpoler. Les points de définition sont obtenus en résolvant un système linéaire. Si les points à interpoler sont irrégulièrement espacés, la surface résultante pourra présenter des oscillations parasites. Le degré de la surface est directement lié au nombre de points à interpoler, alors pour éviter des degrés trop importants, il est possible d'utiliser plusieurs carreaux de Bézier raccordés entre eux. C'est cette solution qui est la plus couramment utilisée.

### **\* Approximation et interpolation à l'aide de surfaces B-spline**

Une surface B-spline constitue également une approximation de son réseau de définition. On module la finesse de l'approximation grâce à l'ordre. En diminuant celui-ci, on rapproche la surface de son filet de points. D'une manière générale, l'augmentation de l'ordre "tend" la surface.

Pour réaliser une interpolation, le nombre de points de contrôle doit être égal au nombre de points à interpoler mais le degré peut être bien inférieur à ce nombre. Les surfaces que l'on obtient ainsi restent numériquement manipulables.

Une paramétrisation uniforme convient mal lorsque les points d'interpolation sont très irrégulièrement espacés. Si, dans de telles conditions, on réussit à obtenir une solution, elle peut être très différente du résultat escompté [DAN 89].

Les B-splines non uniformes offrent des possibilités accrues tant pour l'interpolation que pour l'approximation. La possibilité d'augmenter la multiplicité d'un noeud permet d'attirer la surface vers un point du réseau. Ceci constitue une méthode d'affinement local de l'approximation. Une répartition non uniforme des noeuds permet l'adaptation de la méthode d'interpolation au cas d'un ensemble de points mal répartis. Les résultats obtenus sont de bien meilleure qualité qu'avec des B-spline uniformes.

#### **\* Approximation et interpolation par des surfaces B-spline rationnelles**

Les B-spline rationnelles n'apportent pas d'avantages décisifs dans ces domaines. Toutefois dans le cadre d'une approximation, la faculté de pondération des points offre un moyen supplémentaire d'attirer la surface vers certains points.

### **II . 4 .3 Modélisation de quadriques**

Les quadriques sont des objets courants en C.A.O. Leurs propriétés sont connues depuis très longtemps et ils existent déjà dans de nombreux systèmes [SAR ]. Pour conserver une unité dans la représentation, il est souhaitable de pouvoir modéliser à l'aide des mêmes outils des surfaces de forme libre et les quadriques. Les surfaces décrites à l'aide de polynômes ne permettent qu'une approximation des quadriques. Par exemple, il est impossible de trouver une paramétrisation polynomiale d'une portion de sphère ou de cylindre. Toutefois, les B-splines non uniformes peuvent donner une bonne approximation. La méthode B-spline rationnelle propose une paramétrisation exacte de n'importe quelle quadrique, elle permet même la paramétrisation d'un tore. Mais il faut éviter de conclure trop vite que toutes les surfaces dont la définition contient une fonction trigonométrique peuvent être décrites de cette façon [MIL 86]. Par exemple, les surfaces sinusoidales (tôles ondulées, etc ) ne peuvent pas l'être. Il est en effet impossible d'écrire simultanément  $x$  et  $\cos x$  comme des fractions rationnelles d'une seule variable. La description du processus de définition d'une sphère est détaillée en annexe.

## II.4.4 Raccordement

On entend par raccordement toutes les techniques qui permettent de relier deux surfaces existantes. Les raccords doivent être le plus lisse possible. Les surfaces doivent vérifier des conditions aux bords. Ces conditions sont exprimées par des systèmes d'équations où interviennent des dérivées partielles jusqu'à l'ordre 2. Du point de vue mathématique, ces conditions expriment la continuité des normales et des courbures au passage d'un carreau à l'autre. A partir de ces équations, on obtient des systèmes linéaires en les points de contrôle des surfaces concernées. Dans la pratique, on utilise, soit les équations de continuité complètes [VER 73], soit un cas particulier de ces équations [BOH 84]. Dans la plupart des cas, les équations simplifiées fournissent une solution satisfaisante. Cependant, dans certains cas, il est nécessaire d'envisager les équations complètes (par exemple raccorder un cylindre par sa base à un plan déformé).

### - Résolution à l'aide des équations complètes

Le problème se traite très bien dans le cas des Bézier polynomiales [VER 73], mais il n'admet pas de solutions avec les B-spline polynomiales. Pour contourner cette difficulté, il suffit de décomposer une surface B-spline en plusieurs carreaux de Bézier. Dans le cas des surfaces rationnelles, on ne peut pas envisager de traiter le problème dans l'espace homogène car les équations complètes, qui expriment la continuité des normales et des courbures, n'ont pas de sens dans celui-ci. Il est donc nécessaire de projeter les surfaces rationnelles dans l'espace euclidien. Les équations ont alors une forme beaucoup plus complexes due à la présence des dénominateurs.

### - Résolution à l'aide des équations simplifiées

Les équations simplifiées expriment simplement la continuité des dérivées partielles jusqu'à l'ordre 2 à travers la frontière. Elles se résolvent facilement dans le cas des surfaces polynomiales. N'étant plus liées à la continuité des normales et des courbures, elles conservent tout leur sens dans l'espace homogène. Il est donc possible de raccorder deux surfaces rationnelles dans cet espace. L'utilisation des surfaces rationnelles n'induit pas de difficultés supplémentaires dans la mise en oeuvre de cette méthode.

## II . 5 Manipulation de surfaces

### II . 5 .1 Algorithmes fondamentaux

## II . 5 . 1 . 1 Algorithmes de calculs pour les courbes de Bézier

### \* Algorithme de De Casteljaou

L'algorithme de De Casteljaou est l'outil de base de manipulation d'une courbe de Bézier. Il sert à calculer pour une valeur  $u^*$  fixée du paramètre, le point correspondant sur la courbe.

Grâce aux propriétés de récurrence des polynômes de Bernstein, Paul De Casteljaou a établi cet algorithme simple et efficace.

### Description de l'algorithme

Soit T une courbe de Bézier :  $T: u \rightarrow R(u)$

#### Initialisation

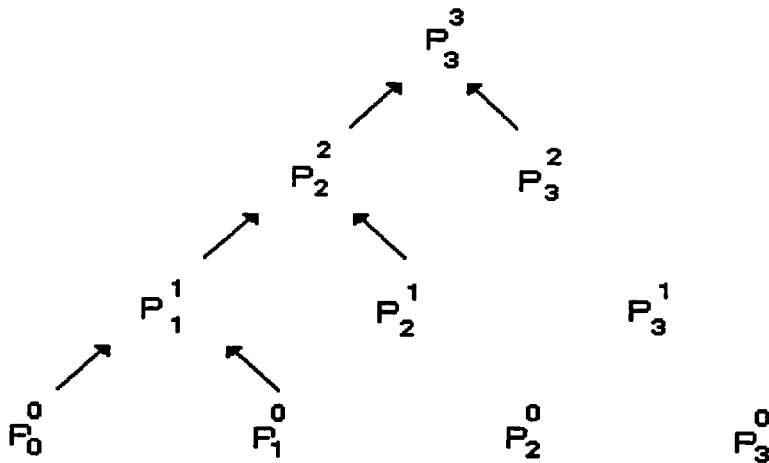
$$P_i^0 = P_i \text{ pour } i = 0..m$$

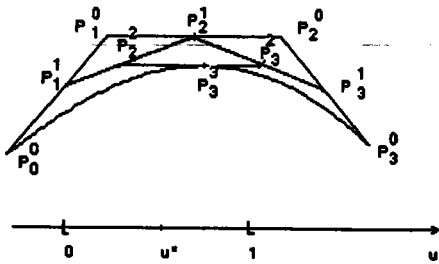
#### récurrence

$$P_i^r(u) = (1-u^*) P_{i-1}^{r-1} + u^* P_i^{r-1} \text{ pour } r \text{ variant de } 1 \text{ à } m$$

et  $i$  de  $r$  à  $m$

Ce processus est parfaitement représenté sous la forme d'un schéma triangulaire.





- Propriétés fondamentales de l'algorithme

Cet algorithme est également un très bon outil de subdivision. La courbe  $\{R(u), u:0..1\}$  est la réunion des arcs  $\{R(u), u:0..u^*\}$  et  $\{R(u), u:u^*..1\}$ . Ces 2 arcs sont des courbes de Bézier de degré  $m$  dont les polygones de contrôle sont les 2 pentes du schéma triangulaire. Plus précisément:

$$\{R(u), u:0..u^*\} = \{P_0^0 B_0^m(t) + P_1^1 B_1^m(t) + \dots + P_m^m B_m^m(t), t:0..1\}$$

$$\{R(u), u:u^*..1\} = \{P_m^m B_0^m(t) + P_{m-1}^{m-1} B_1^m(t) + \dots + P_0^0 B_m^m(t), t:0..1\}$$

\* Elévation du degré

Soit  $T$  une courbe de Bézier de degré  $m$ , définie comme dans l'introduction. Il est possible de définir  $T'$  une courbe de Bézier de degré  $(m+1)$  telle que pour tout  $u$  dans  $[0..1]$ ,  $T(u) = T'(u)$ .

Description de l'algorithme

Initialisation

$$P'_0 = P_0$$

Réurrence

$$\text{pour } i=1..m+1 \quad P'_i = a_i P_{i-1} + (1-a_i) P_i$$

$$\text{où } a_i = i/(m+1)$$

Remarque: cet algorithme permet de faire coïncider les degrés de deux courbes.

\* Passage à la base canonique

On désire écrire une courbe de Bézier  $R(u)$  sous la forme

$$R(u) = A_0 + A_1u + A_2u^2 + \dots + A_mu^m$$

Description de l'algorithme

Initialisation

$$\Delta_i^0 = P_i \quad \text{pour } i=0..m$$

Récurrence

$$\Delta_i^r = \Delta_{i+1}^{r-1} - \Delta_i^{r-1} \quad \text{pour } r=1..m \text{ et } i=0..m-r$$

Finalement

$$A_i = C_m^i \Delta_0^i$$

Exemple pour  $m = 3$  :

$$\begin{array}{cccc} & \Delta_0^3 & & & \\ & \Delta_0^2 & \Delta_1^2 & & \\ & \Delta_0^1 & \Delta_1^1 & \Delta_2^1 & \\ \Delta_0^0 & \Delta_1^0 & \Delta_2^0 & \Delta_3^0 & \end{array}$$

II.5.1.2 *Algorithmes de calculs pour les courbes B-spline*

\* Algorithme de Cox-De Boor

L'algorithme de Cox-De Boor est aux B-spline ce que l'algorithme de De Casteljaou est aux Bézier. Son rôle est le calcul de points sur une courbe.

Soit  $T$  une courbe B-spline d'ordre  $k$ , définie par  $(m+1)$  points  $\{P_0 \dots P_m\}$ , de table nodale  $\{u_0 \dots u_{m+k}\}$ .  $T$  est décrite par

$$u \mapsto R(u) = \sum_{i=0}^m P_i N_i^k(u) \quad u \text{ varie entre } u_{k-1} \text{ et } u_{m+1}$$

### Description de l'algorithme

#### Initialisation

- recherche de l'unique indice  $d$  tel que

$$u_d \leq u^* < u_{d+1}$$

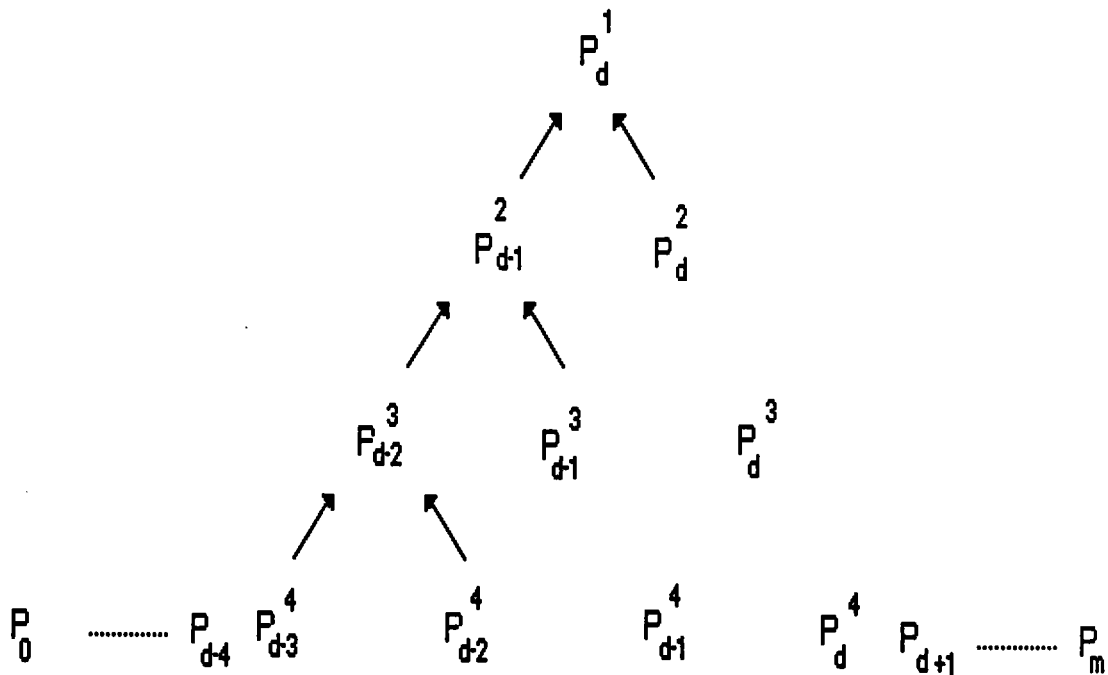
$$P_i^k = P_i \text{ pour } i = d-k+1 \dots d$$

#### Récurrence

$$P_i^r = a_{ir} P_i^{r+1} + (1-a_{ir}) P_{i-1}^{r+1} \text{ pour } r \text{ variant de } (k-1) \text{ à } 1$$

$$\text{où } a_{ir} = (u^* - u_i) / (u_{i+r} - u_i) \text{ pour } i \text{ variant de } (d-r+1) \text{ à } d$$

Exemple pour  $k = 4$



Cet algorithme trouve sa justification dans les propriétés de récurrence et de support borné des fonctions B-spline.

### Redéfinition de la courbe

L'application de Cox-De Boor correspond à l'insertion de  $u^*$  ( $k-1$ ) fois dans la table des noeuds. Par conséquent, la courbe peut être redéfinie.

Les  $(m+k)$  points  $P_0, \dots, P_{d-k}, P_{d-k+1}^k, \dots, P_d^1, \dots, P_d^k, P_{d+1}, \dots, P_m$  et les  $(m+2k)$  valeurs nodales

$$v_0 = u_0, \dots, v_d = u_d$$

$$v_{d+1} = u^*, \dots, v_{d+k-1} = u^*$$

$$v_{d+k} = u_{d+1}, \dots, v_{m+2k-1} = u_{m+k}$$

constituent les nouveaux éléments de définition de la courbe  $T$ . On constate que l'ordre de  $T$  n'a pas été modifié. Le nouveau polygone de contrôle a été obtenu en substituant la base du schéma triangulaire par ses 2 pentes. Cette modification rapproche le polygone de la courbe.

Remarque : Cet algorithme permet de diviser la courbe  $T = \{u \rightarrow R(u), u: u_{k-1} \dots u_{m+1}\}$  en 2 nouvelles courbes  $T_g = \{u \rightarrow R(u), u: u_{k-1} \dots u^*\}$  et  $T_d = \{u \rightarrow R(u), u: u^* \dots u_{m+1}\}$ .

$T_g$  est une courbe B-spline d'ordre  $k$  définie par les  $(d+1)$  points

$P_0, \dots, P_{d-k}, P_{d-k+1}^k, \dots, P_d^1$  et par le vecteur nodal

$$tndw = \{w_0 = v_0, \dots, w_{d+k-1} = v_{d+k-1}, w_{d+k} = u^*\}.$$

$T_d$  est une courbe B-spline d'ordre  $k$  définie par les  $(m+k-d)$  points

$P_d^2, \dots, P_1^k, P_{d+1}, \dots, P_m$  et par le vecteur nodal

$$tnds = \{s_0 = u^*, s_1 = v_{d+1}, \dots, s_{k-1} = v_{d+k-1}, s_k = v_{d+k}, \dots, s_{m-d+2k-1} = v_{m+2k-1}\}$$

#### \* Algorithme d'Oslo

C'est un algorithme de raffinement [BAR 88]. Soit  $T$  une courbe B-spline, il permet l'insertion simultanée de valeurs nodales différentes dans la table des noeuds de  $T$ . L'algorithme d'Oslo fournit un nouveau polygone plus proche de la courbe  $T$ .

#### \* Algorithme de Boehm

C'est un cas particulier de l'algorithme d'Oslo. Il insère une unique valeur nodale.



**Remarque:** On peut montrer que l'utilisation de l'algorithme de Cox - De Boor revient à répéter (k-1) fois l'algorithme de Boehm avec la même valeur nodale. On peut trouver une comparaison de ces différents algorithmes dans [DAN 89].

**\* Algorithme de calcul des dérivées successives**

Cet algorithme dû à Boehm[BOEH 84] permet le calcul simultané des dérivées jusqu'à l'ordre k-1 d'une courbe B-spline d'ordre k.

Soit T une courbe B-spline d'ordre k définie par les points  $P_0 \dots P_m$ . et le vecteur nodal  $\{u_0 \dots u_{m+k}\}$ . T est décrite par:

$$R : [u_{k-1}, u_{m+1}[ \longrightarrow R^3$$

$$t \longmapsto R(t) = \sum_{i=0}^m P_i N_i^k(t)$$

Soit  $u^*$  fixé appartenant à  $[u_{k-1}, u_{m+1}[$ , On désire calculer les dérivées de R en  $u$ .

**Description de l'algorithme**

**Initialisation**

- Recherche de l'unique indice d tel que u appartient à  $[u_d, u_{d+1}[$ .  
 puis initialisation "base triangle inférieur droit":

$$P_i^0 = P_i \text{ pour } i = d-k+1 \dots d$$

**Récurrance triangle inférieur droit:**

$$P_i^r = (P_i^{r-1} - P_{i-1}^{r-1}) / (u_{i+k-r} - u_i) \text{ pour } r = 1 \dots k-1 \text{ et}$$

$$i = d - (k-r) + 1 \dots d$$

**Récurrance triangle supérieur gauche:**

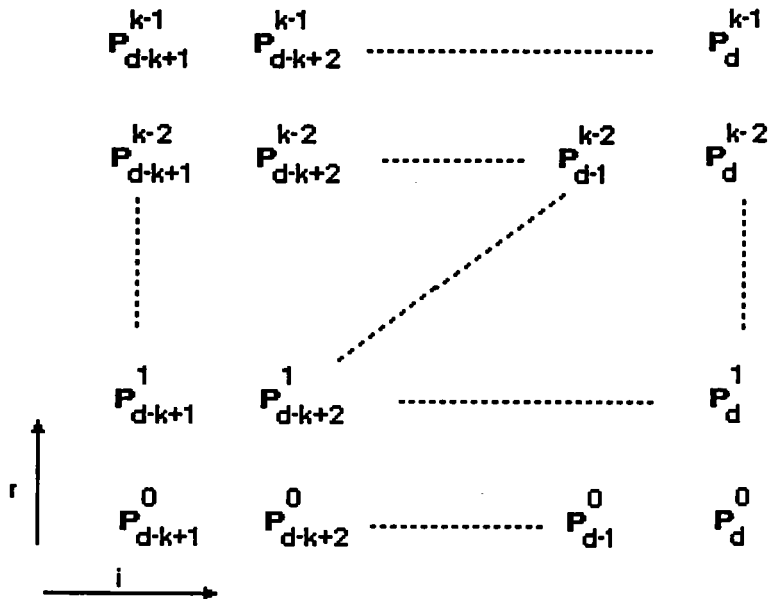
$$P_i^r = P_i^{r-1} + (u^* - u_{d+r-(k-1)}) P_{i+1}^{r-1} \text{ pour } r = 1 \dots k-1 \text{ et}$$

$$i = d-k+1 \dots d-k+r$$

La dérivée de rang r est donnée par:

$$R^{(r)}(u^*) = (k-1)(k-2) \dots (k-r) P_{d+r-(k-1)}^{k-1} \text{ pour } r = 0 \dots k-1$$

Il y a donc 2 schémas triangulaires superposés qui forment un schéma carré. La plus haute ligne du carré fournit les dérivées successives.



## II.5.2 Applications usuelles

### II.5.2.1 Echantillonnage d'une surface

Il s'agit de calculer un grand nombre de points de la surface. Echantillonner une surface aide à la visualisation ou à la modélisation par facettes.

#### \* Echantillonnage d'un carreau de Bézier

Pour le calcul d'un grand nombre de points, la méthode la plus rapide est de réécrire notre carreau dans la base canonique des polynômes à 2 variables. Ensuite, on utilise le schéma de Horner pour le calcul de chaque point.

Soit  $C$  un carreau de Bézier défini comme dans II.1.2.

Passage à la forme canonique:

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

On veut réécrire  $S(u,v)$  sous la forme:

$$\sum_{i=0}^m \sum_{j=0}^n A_{ij} u^i v^j$$

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

$$\text{On pose } Q_i(v) = \sum_{j=0}^n P_{ij} B_j^n(v)$$

Pour  $i$  fixé,  $Q_i(v)$  est une courbe de Bézier qui s'exprime à l'aide de l'algorithme de passage à la base canonique sous la forme

$$\sum_{j=0}^n D_{ij} v^j$$

$$S(u,v) = \sum_{i=0}^m B_i^m(u) \left( \sum_{j=0}^n D_{ij} v^j \right)$$

$$S(u,v) = \sum_{j=0}^n v^j \left( \sum_{i=0}^m D_{ij} B_i^m(u) \right)$$

$$\text{On pose } R_j(u) = \sum_{i=0}^m D_{ij} B_i^m(u)$$

Pour  $j$  fixé,  $R_j(u)$  est également une courbe de Bézier qui s'écrit sous la forme

$$R_j(u) = \sum_{i=0}^m A_{ij} u^i v^j$$

$$\text{d'où } S(u,v) = \sum_{i=0}^m \sum_{j=0}^n A_{ij} u^i v^j$$

L'algorithme de passage à la forme canonique d'un carreau de Bézier consiste à calculer dans un premier temps les  $D_{ij}$  et ensuite les  $A_{ij}$ .

Pour le calcul de points d'un carreau de Bézier, on peut également se servir de l'algorithme de De Casteljau, néanmoins quand le nombre de points est important, la méthode que l'on vient de décrire est plus efficace.

\* **Echantillonnage d'une surface B-spline**

Passage à la base canonique pour une courbe B-spline

Soit une courbe B-spline d'ordre de vecteur nodal  $\{u_0 \dots u_{m+k}\}$  et de points de contrôle  $P_0 \dots P_m$ . Pour chaque noeud  $u_d$  tel que  $k-1 \leq d \leq m$  et tel que  $[u_d, u_{d+1}[$  non vide, on peut calculer toutes les dérivées et donc grâce à la formule de Taylor-Young obtenir le développement polynomial de la courbe sur l'intervalle  $[u_d, u_{d+1}[$ .

Echantillonnage d'une surface B-spline

Pour échantillonner une B-spline, nous allons également utiliser une méthode qui va réécrire notre surface dans la base canonique des polynômes. Soit  $C$  une surface B-spline, les notations sont les mêmes que celles utilisées dans l'introduction. Sur chaque pavé  $[u_i, u_{i+1}[ \times [v_j, v_{j+1}[$  ( $i=0 \dots m, j=0 \dots n$ ), notre surface est une polynomiale en  $(u, v)$  de degré  $(k-1) \times (l-1)$ .

Soit  $(u, v)$  dans le pavé  $[u_d, u_{d+1}[ \times [v_g, v_{g+1}[$ .  $d$  et  $g$  sont tels que  $u_d < u_{d+1}$  et  $v_g < v_{g+1}$ . Sur ce pavé, la surface s'écrit

$$\begin{aligned}
 \mathbf{s}(u, v) &= \sum_{i=g-k+1}^d \sum_{j=d+1}^g P_{ij} N_i^k(u) N_j^l(v) \\
 \mathbf{s}(u, v) &= \sum_{i=g-k+1}^d N_i^k(u) \left( \sum_{j=d+1}^g P_{ij} N_j^l(v) \right)
 \end{aligned}$$

Le terme entre parenthèses noté  $Q_i(v)$  est une courbe B-spline. En utilisant la méthode de passage à la base canonique pour une courbe B-spline, on obtient

$$\begin{aligned}
 Q_i(v) &= \sum_{j=0}^{l-1} C_{ij}(g) (v-v_g)^j \\
 \mathbf{s}(u, v) &= \sum_{j=0}^{l-1} (v-v_g)^j \left( \sum_{i=g-k+1}^d C_{ij}(g) N_i^k(u) \right)
 \end{aligned}$$

De même, le terme entre parenthèses peut être réécrit sous la forme

$$\sum_{i=0}^{k-1} A_{ij}(g,d) (u-u_d)^i$$

D'où  $S(u,v) = \sum_{i=0}^{k-1} \sum_{j=0}^{l-1} A_{ij}(g,d) (u-u_d)^i (v-v_g)^j$

Sur chaque pavé non vide  $[u_d, u_{d+1}[ \times [v_g, v_{g+1}[$ , on obtient par ce procédé une matrice  $[A_{ij}(d,g)]_{i=0..k-1, j=0..l-1}$  des coefficients de la forme canonique.

### Cas particulier des B-splines uniformes

Pour des ordres  $k$  et  $l$  fixés, les fonctions de base peuvent être déduites les unes des autres par translation. Les  $k$  premières fonctions de bases en  $u$  et les  $l$  premières fonctions de base en  $v$  sont tabulées, ensuite tous les calculs sont remplacés par des références à ces tabulations. Les algorithmes deviennent alors beaucoup moins coûteux en temps et en place mémoire.

#### \* Echantillonnage de surfaces B-spline rationnelles

Dans l'espace homogène, les B-spline rationnelles sont des polynomiales. La méthodologie reste la même, les B-splines rationnelles sont traités comme des B-spline classiques. L'ajout de la quatrième coordonnée augmente d'un tiers les calculs.

#### II.5.2.2 Echantillonnage des normales

Beaucoup d'algorithmes de visualisation nécessitent la connaissance des normales à la surface traitée. Souvent pour chaque point calculé, on désire connaître simultanément la normale associée. Nous venons de montrer que chaque type de surface peut être écrit dans la base canonique des polynômes. La normale est fonction des dérivées partielles en  $u$  et  $v$  et une dérivée partielle d'un polynôme est encore un polynôme. On peut donc échantillonner les dérivées de la même façon que la surface.

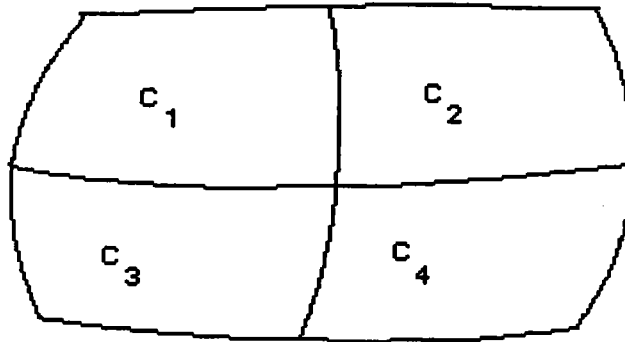
Dans le cas des B-spline rationnelles, les dérivées partielles sont plus compliquées car on est alors obligé de dériver des fonctions rationnelles.

## II.5.2.3 Subdivisions -Approximations

### II.5 .2.3.a Techniques

#### \* Subdivision d'un carreau de Bézier

La subdivision d'un carreau de Bézier consiste à le découper en 4 carreaux de Bézier adjacents. Les lignes de coupe sont des lignes isoparamétriques. Les 4 carreaux obtenus forment une partition du carreau initial.



Les carreaux obtenus sont de même degré que le carreau initial.

- Calculs des nouveaux réseaux de contrôle

On choisit  $(u^*, v^*)$  dans le pavé  $[0,1] \times [0,1]$ . Le couple  $(u^*, v^*)$  détermine les 2 lignes de coupe:  $S(u^*, v)$  et  $S(u, v^*)$ . IL est commode d'écrire le réseau de contrôle de C sous la forme matricielle suivante.

$$P = \begin{bmatrix} P_{00} & \dots & P_{0n} \\ \vdots & & \vdots \\ P_{i0} & & P_{in} \\ \vdots & & \vdots \\ P_{m0} & \dots & P_{mn} \end{bmatrix}$$

Dans un premier temps, nous allons découper C en 2 carreaux C'<sub>1</sub> et C'<sub>2</sub> selon la courbe  $S(u, v^*)$ .

$$C'_1 = C_1 \cup C_3 \quad C'_2 = C_2 \cup C_4$$

Chaque ligne de la matrice P détermine une courbe de Bézier en v que l'on découpe à l'aide de l'algorithme de De Casteljau en la valeur v\*.

Chaque ligne [P<sub>ij</sub>]<sub>j=0..n</sub> est donc prise à son tour comme la base du schéma triangulaire de Casteljau.

Chaque [P<sub>ij</sub>]<sub>j=0..n</sub> est remplacé par [Q<sub>ij</sub>]<sub>j=0..2n</sub>

Nous obtenons 2 matrices de points qui sont les réseaux de C<sub>1</sub> et C<sub>2</sub>.

$$\begin{array}{c} \begin{bmatrix} \bar{Q}_{00} & \dots & \bar{Q}_{0n} \\ \vdots & & \vdots \\ \bar{Q}_{m0} & \dots & \bar{Q}_{mn} \end{bmatrix} \quad \begin{bmatrix} \bar{Q}_{0n} & \dots & \bar{Q}_{02n} \\ \vdots & & \vdots \\ \bar{Q}_{mn} & \dots & \bar{Q}_{m2n} \end{bmatrix} \\ C_1 \qquad\qquad\qquad C_2 \end{array}$$

C<sub>1</sub> et C<sub>2</sub> sont découpés suivant la courbe S(u\*,v) : pour j fixé  
 Chaque colonne [Q<sub>ij</sub>]<sub>i=0..m</sub> est remplacé à l'aide de l'algorithme de De Casteljau par [R<sub>ij</sub>]<sub>i=0..2m</sub>

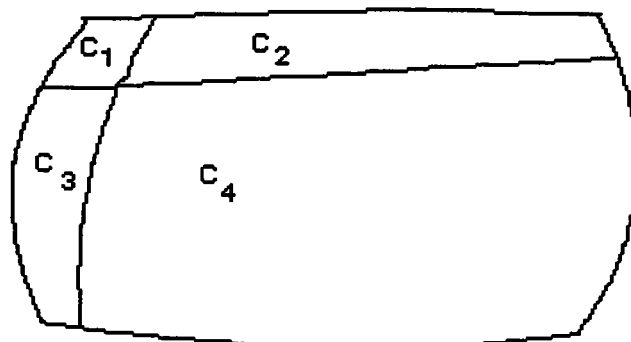
Finalement on obtient 4 nouvelles matrices

$$\begin{array}{c} \begin{bmatrix} \bar{R}_{00} & \dots & \bar{R}_{0n} \\ \vdots & & \vdots \\ \bar{R}_{m0} & \dots & \bar{R}_{mn} \end{bmatrix} \quad \begin{bmatrix} \bar{R}_{0n} & \dots & \bar{R}_{02n} \\ \vdots & & \vdots \\ \bar{R}_{mn} & \dots & \bar{R}_{m2n} \end{bmatrix} \\ \begin{bmatrix} \bar{R}_{m0} & \dots & \bar{R}_{mn} \\ \vdots & & \vdots \\ \bar{R}_{2m0} & \dots & \bar{R}_{2mn} \end{bmatrix} \quad \begin{bmatrix} \bar{R}_{mn} & \dots & \bar{R}_{m2n} \\ \vdots & & \vdots \\ \bar{R}_{2mn} & \dots & \bar{R}_{2m2n} \end{bmatrix} \end{array}$$

qui représentent les points de contrôle associés à C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>.

**Remarque:**

- On peut s'arrêter à  $C_1$  et  $C_2$  et ainsi découper uniquement en deux parties le carreau  $C$ .
- Le choix de  $u^*$  et  $v^*$  est libre. Il est possible d'obtenir des carreaux de taille très différentes.



- Le point  $R_{mn}$ , commun aux 4 matrices, est le point  $S(u^*, v^*)$ . [BAR 88].

**\* Subdivision d'une surface B-spline**

Le processus de subdivision d'une surface B-spline est analogue à celui d'un carreau de Bézier. Soit  $C$  une surface B-spline notée comme dans l'introduction, nous allons construire une partition de  $C$ ,  $C = C_1 \cup C_2 \cup C_3 \cup C_4$ . Les lignes de coupe sont  $S(u^*, v)$  et  $S(u, v^*)$ ,  $(u^*, v^*)$  choisi dans  $[u_{k-1}, u_{m+1}] \times [v_{l-1}, v_{n+1}]$ .

Les 4 surfaces obtenues sont encore des surfaces B-spline de même ordres que  $C$ .

- Calcul des nouveaux réseaux de définition:



Le réseau de contrôle C peut s'écrire sous la forme matricielle

$$\begin{bmatrix} P_{00} & \dots & P_{0n} \\ \vdots & & \vdots \\ P_{i0} & & P_{in} \\ \vdots & & \vdots \\ P_{m0} & \dots & P_{mn} \end{bmatrix}$$

Supposons  $(u^*, v^*)$  choisi dans  $[u_g, u_{g+1}[ \times [v_d, v_{d+1}[$ , on choisit de découper d'abord selon la courbe  $S(u, v^*)$ , on obtient les surfaces  $C'_1$  et  $C'_2$ .  $C'_1 = C_1 \cup C_3$ ,  $C'_2 = C_2 \cup C_4$ . Chaque ligne  $[P_{ij}]_{j=0..n}$  est traité comme suit:

L'algorithme de Cox-De Boor ayant pour données les  $l$  points  $P_{i,d-1+1}, \dots, P_{i,d}$ , la table nodale  $tndv$  et la valeur  $v^*$ , rend les  $(2l-1)$  points  $D_{d-1+1}, \dots, D_d, \dots, D_{d+1-1}$ .

La ligne  $[P_{ij}]_{j=0..n}$  est remplacée par la ligne  $[Q_{ij}]_{j=0..n+1-1}$

$$Q_{i0} = P_{i0} \quad \dots \quad Q_{i,d-1} = P_{i,d-1}$$

$$Q_{i,d-1+1} = D_{d-1+1} \quad \dots \quad Q_{i,d+1-1} = D_{d+1-1}$$

$$Q_{i,d+1} = P_{i,d+1} \quad \dots \quad Q_{i,n+1-1} = P_{i,n}$$

Les matrices

$$\begin{bmatrix} \bar{Q}_{00} & \dots & Q_{0d} \\ \vdots & & \vdots \\ Q_{m0} & \dots & Q_{md} \end{bmatrix} \quad \begin{bmatrix} \bar{Q}_{0d} & \dots & Q_{0n+1-1} \\ \vdots & & \vdots \\ Q_{md} & \dots & Q_{m n+1-1} \end{bmatrix}$$

déterminent les réseaux de contrôle de  $C'_1$  et  $C'_2$

Les colonnes des matrices vont être traitées comme le furent les lignes. L'algorithme de Cox - De Boor remplace les  $k$  points  $Q_{g-k+1,j}, \dots, Q_{g,j}$  par les  $(2k-1)$  points  $D_{g-k+1}, \dots, D_{g+k-1}$ . Cox-De Boor est utilisé avec  $u^*$  et la table des noeuds  $tndu$ .

On obtient alors les colonnes  $[R_{ij}]_{i=0..m+k-1}$

$$\begin{aligned} R_{0,j} &= Q_{0,j} \quad \dots \quad R_{g-k,j} = Q_{g-k,j} \\ R_{g-k+1,j} &= D_{g-k+1} \quad \dots \quad R_{g+k-1,j} = D_{g+k-1} \\ R_{g+k,j} &= Q_{g+1,j} \quad \dots \quad R_{m+k-1,j} = Q_{m,j} \end{aligned}$$

Finalement, on obtient 4 nouvelles matrices

$$\begin{array}{cc} \left[ \begin{array}{ccc} \bar{R}_{00} & \dots & R_{0d} \\ & & \\ R_{g0} & \dots & R_{gd} \end{array} \right] & \left[ \begin{array}{ccc} \bar{R}_{0d} & \dots & R_{0n+l-1} \\ & & \\ R_{gd} & \dots & R_{gn+l-1} \end{array} \right] \\ \\ \left[ \begin{array}{ccc} \bar{R}_{g0} & \dots & R_{gd} \\ & & \\ R_{m+k-10} & \dots & R_{m+k-1d} \end{array} \right] & \left[ \begin{array}{ccc} \bar{R}_{gd} & \dots & R_{gn+l-1} \\ & & \\ R_{m+k-1d} & \dots & R_{m+k-1n+l-1} \end{array} \right] \end{array}$$

qui sont les réseaux de contrôle des surfaces  $C_1, C_2, C_3, C_4$ .

Les tables nodales associées à  $C_1$  sont:

$$\{u_0, \dots, u_g, u^*, \dots, u^*\} \quad u^* \text{ est répété } k \text{ fois}$$

et

$$\{v_0, \dots, v_d, v^*, \dots, v^*\} \quad v^* \text{ est répété } l \text{ fois}$$

Les tables nodales associées à  $C_2$  sont:

$$\{u^*, \dots, u^*, u_{g+1}, \dots, u_{m+k}\} \quad u^* \text{ est répété } k \text{ fois}$$

et

$$\{v_0, \dots, v_d, v^*, \dots, v^*\} \quad v^* \text{ est répété } l \text{ fois}$$

Les tables nodales associés à  $C_3$  sont:

$\{u_0, \dots, u_g, u^*, \dots, u^*\}$   $u^*$  est répété  $k$  fois

et

$\{v^*, \dots, v^*, v_{d+1}, \dots, v_{n+1}\}$   $v^*$  est répété  $l$  fois

Les tables nodales associés à  $C_4$  sont:

$\{u^*, \dots, u^*, u_{g+1}, \dots, u_{m+k}\}$   $u^*$  est répété  $k$  fois

et

$\{v^*, \dots, v^*, v_{d+1}, \dots, v_{n+1}\}$   $v^*$  est répété  $l$  fois

Remarque:  $R_{g,d} = S(u^*, v^*)$

\* **Subdivision d'une surface B-spline rationnelle**

Les surfaces B-spline rationnelles sont étudiées dans l'espace homogène où elles sont polynomiales comme les B-spline classiques. Le processus de subdivision est alors strictement le même.

II.5.2.3.b Objectifs

Les procédés que nous venons de décrire peuvent être réitérés, la surface initiale est alors décomposée en un arbre quaternaire. Les réseaux de contrôle successifs approchent de mieux en mieux leurs surfaces associées. Cela constitue un moyen d'approximation polyédrique de la surface. La connaissance de la surface est préservée à chaque étape de la subdivision récursive. Grâce à la propriété de l'enveloppe convexe commune à toutes les surfaces étudiées, il est possible d'associer à chaque noeud une boîte englobante. Toutes ces informations sont très utiles pour l'intersection d'une surface avec toute autre entité géométrique.

## II, 5.3 Conclusion

- \* Les techniques de subdivision récursive et d'échantillonnage permettent toutes les deux de définir une approximation polyédrique d'une surface. L'échantillonnage d'une surface est beaucoup moins coûteux en temps et en espace qu'une subdivision récursive mais elle perd la notion de surface.
- \* Les calculs mis en oeuvre, tant pour échantillonner que pour une subdivision récursive, sont moins importants pour les surfaces de Bézier que pour les surfaces B-spline. Car les algorithmes de base manipulant les courbes B-spline sont plus compliqués que ceux manipulant les courbes de Bézier.
- \* Décomposition d'une surface B-spline en carreaux de Bézier  
Une surface B-spline est une polynomiale par morceaux, ses paramètres de définition  $(u,v)$  évoluent dans des pavés délicats à manipuler. Par contre, les paramètres d'un carreau de Bézier évoluent toujours dans le pavé  $[0,1] \times [0,1]$ . De plus, les opérations que nous venons de détailler, sont plus simples à réaliser avec des Bézier qu'avec des B-spline. Il est donc souvent avantageux de décomposer une surface B-spline en une partition de carreaux de Bézier.

Pour décomposer une surface B-spline en une partition de carreaux de Bézier, il suffit de subdiviser la surface selon tous les couples  $(u_j, v_j)$   $i=k..m, j=1..n$ .

Chaque feuille de l'arbre quaternaire est un carreau de Bézier [BAR 88]

**REMARQUE:** De la même façon, une B-spline rationnelle peut être partitionnée en carreaux de Bézier rationnels.

## II.6 Conclusion

### II.6.1 Bilan

La forme B-spline est un outil de modélisation plus souple que la forme Bézier. Cette souplesse est due d'une part au caractère "polynomial par morceaux" et d'autre part à la relative indépendance entre le degré et le nombre de points. La puissance de modélisation des B-spline est supérieure à celle des Bézier. Les B-spline rationnelles étendent encore les possibilités.

La création de surfaces de raccordement est délicate à réaliser quelque soit le type de représentation choisi.

La nature, plus complexe des B-spline, rend en général les traitements plus coûteux qu'avec les Bézier.

Une stratégie, conservant la puissance des B-spline et la simplicité des Bézier est de concevoir les objets à l'aide des B-spline puis lors de traitements délicats ( par exemple: calculs d'intersection) de transformer celles-ci en carreaux de Bézier.

Pour conclure, nous pouvons affirmer que l'outil de modélisation le plus général, parmi tous ceux que nous venons d'étudier est la forme B-spline rationnelle. Elle n'apporte pas de difficultés théoriques nouvelles tout en augmentant considérablement l'éventail des surfaces modélisables. Son seul inconvénient est d'engendrer un tiers de calculs supplémentaires dûs à l'existence d'une quatrième coordonnée. Il faut tout de même noter que la pondération des fonctions splines (par la coordonnée homogène) n'est pas simple à utiliser dans la pratique, il vaut mieux envisager les Nurbs comme un modèle interne. Toutes ces considérations sont des considérations théoriques, il faut ne pas oublier les aspects pratiques tels que la facilité de mise en oeuvre effective.

### II.6.2 Limitations communes

Toutes les surfaces que nous avons étudiées, sont connues à travers une paramétrisation. C'est une définition "descriptive": elle permet un échantillonnage facile mais elle ne donne pas une équation cartésienne de la surface. Une équation cartésienne est indispensable à la résolution exacte des problèmes d'intersection. Il existe des techniques qui permettent d'obtenir une équation implicite d'une surface polynomiale de degré  $n$  selon  $u$  et de degré  $m$  selon  $v$  mais le résultat est en général de degré  $2n \times m$ . Ces équations sont rarement utilisables d'un point de vue informatique car les degrés sont trop élevés.

La connaissance d'une équation implicite du carreau ne permet pas de délimiter ce carreau sur la surface algébrique définie par l'équation.

Tous les types de modélisation présentés font appel à la notion de produit tensoriel. Cette définition entraîne l'existence de deux directions principales en  $u$  et  $v$ . Plus exactement, l'existence de deux champs de courbes privilégiés: les lignes isoparamétriques.

Il n'est pas possible de représenter une courbe arbitraire inscrite sur la surface comme une courbe de Bézier ou B-spline alors que c'est toujours le cas pour une ligne isoparamétrique. Cette lacune prend toute son importance lors du raccordement de deux surfaces le long d'une courbe quelconque. En pratique, il est nécessaire de se ramener à un raccordement le long de lignes isoparamétriques (au prix de subdivisions).

- La définition par produit tensoriel implique que les surfaces réalisées soient toutes des rectangles déformés. Elles se prêtent mal à la modélisation de surfaces n'ayant pas quatre sommets. [GIA 88]

# CHAPITRE III

### **III . 1 Introduction**

Dans le premier chapitre, nous nous sommes intéressés à la modélisation des solides en C.A.O. Nous avons étudié les fondements mathématiques des différents modèles existants. Dans le deuxième chapitre, nous nous sommes intéressés aux différents types de modèles de surface. Nous avons mené une étude comparative de ces différents modèles en essayant de mettre en évidence leurs avantages et leurs inconvénients respectifs.

Dans cette partie, après un bref rappel des propriétés des Nurbs, nous allons détailler les différentes possibilités d'intégration des surfaces dans les modèles solides. Nous terminerons ce chapitre par les essais que nous avons programmés pour valider notre approche.

### **III . 2 Rappel des propriétés importantes des NURBS pour la modélisation des solides**

#### **III. 2.1 Définition paramétrique**

Les surfaces rationnelles , comme tous les autres modèles surfaciques de la C.A.O sont connues par l'intermédiaire d'une paramétrisation . C'est une définition "descriptive" : elle permet un échantillonnage facile mais elle ne donne pas une équation cartésienne de la surface . Une équation cartésienne serait indispensable à la résolution exacte des problèmes d'intersection et permettrait de séparer l'espace en deux demi-espaces.



### III . 2 . 2 Produit tensoriel

Les surfaces rationnelles font , comme les surfaces polynomiales , appel à la notion de produit tensoriel . Cette caractéristique entraîne l'existence de 2 directions principales en  $u$  et  $v$  . Plus exactement , cela induit l'existence sur la surface de 2 champs de courbes privilégiés : les lignes isoparamétriques . Il n'est pas possible de représenter une courbe arbitraire inscrite sur la surface comme une courbe de Bézier ou B-spline alors que c'est toujours le cas pour une ligne isoparamétrique . Cette lacune prend toute son importance lors du raccordement de deux surfaces le long d'une ligne quelconque . La grande facilité de manipulation des lignes isoparamétriques a donc un gros revers [GIA 88]. La définition par produit tensoriel implique que les surfaces réalisées soient toutes des rectangles déformés. Elles se prêtent mal à la modélisation des surfaces n'ayant pas 4 sommets ; si l'on considère les surfaces définies par les coordonnées barycentriques, des carreaux triangulaires peuvent être modélisés, mais la généralité de la modélisation est perdue [BÖH 84]( les carreaux triangulaires doivent être traités comme un cas particulier) .

### III . 2 . 3 Réseau de contrôle

Chaque carreau est construit à partir d'un réseau de points de contrôle . Tout point du carreau est une combinaison linéaire convexe (barycentre) du réseau . Cela a plusieurs conséquences immédiates:

- Le réseau de définition constitue une approximation de la surface.
- La surface est toujours contenue dans l'enveloppe convexe de son réseau de contrôle. Cette propriété permet de définir facilement un volume englobant à une surface donnée, ce qui est très intéressant dans de nombreux algorithmes.
- Pour effectuer une transformation affine sur une surface , il suffit de l'effectuer sur chacun des points de contrôle . En revanche une transformation perspective ne peut être directement appliquée sur les points de contrôle.

### III . 2 . 4 Les quadriques

Les quadriques sont des objets courants en C.A.O . Leurs propriétés sont connues depuis très longtemps et elles existent déjà dans de nombreux systèmes. Pour conserver une unité dans la représentation , il est souhaitable de pouvoir modéliser à l'aide des mêmes outils des surfaces de forme libre et les quadriques. Les surfaces décrites par des polynômes ne permettent qu'une approximation des quadriques. Par exemple , il est impossible de trouver une paramétrisation polynomiale d'une portion de sphère ou de

cylindre. Toutefois , les B-spline non uniformes peuvent donner de bonnes approximations. La méthode B-spline rationnelle propose une paramétrisation exacte de n'importe quelle quadrique; elle permet même la paramétrisation exacte d'un tore. Les Bézier et les B-spline rationnelles, par l'intermédiaire de ce procédé, permettent de décrire un grand nombre de surfaces définies habituellement à l'aide des fonctions trigonométriques ( cf annexe ). Mais il faut éviter de conclure trop vite que toutes les surfaces dont la définition contient une fonction trigonométrique peuvent être décrites de cette façon [MIL86]. Par exemple , les surfaces sinusoidales (tôle ondulée , ondes concentriques sur l'eau ) ne peuvent pas l'être . Il est , en effet , impossible d'écrire simultanément  $x$  et  $\cos x$  comme des fractions rationnelles d'une seule variable [ROC 87].

### III . 3 Différentes approches d'intégration des surfaces gauches dans les modèles de solides

On peut naturellement considérer l'intégration des surfaces gauches dans les modèles de solides selon l'approche B-Rep ou l'approche CSG.

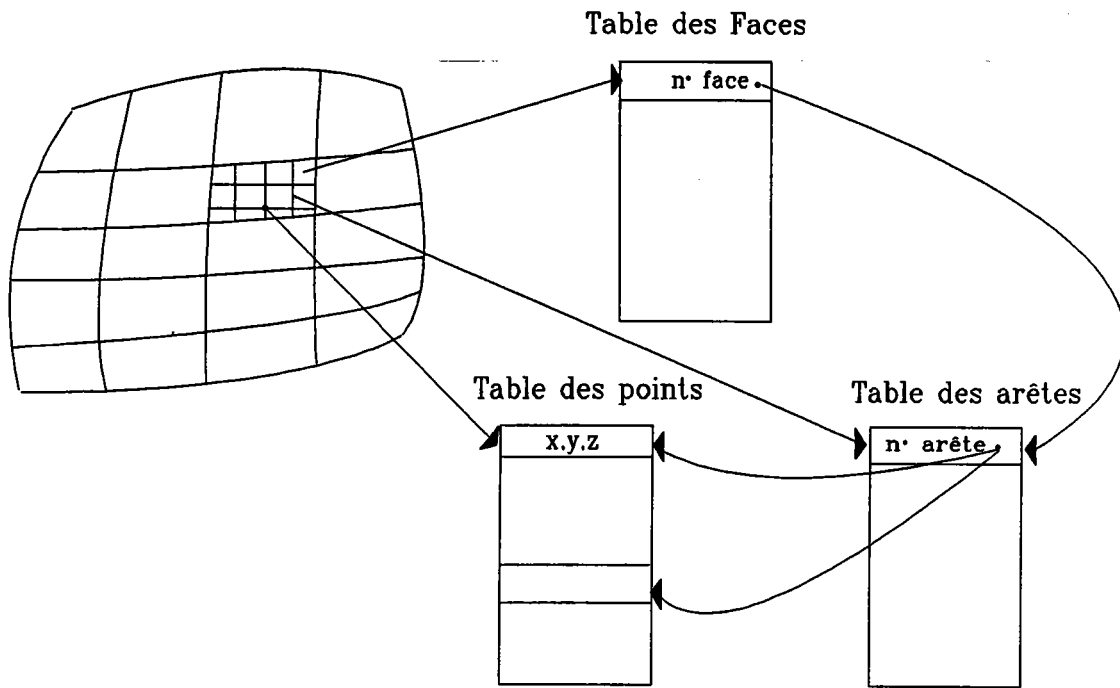
#### III. 3 . 1 Approche B-Rep

Trois grandes familles de modèles sont envisageables [MIL 89]:

- les modèles utilisant des approximations ;
- les modèles par assemblage de faces ou d'objets ;
- les modèles basés sur des carreaux restreints ;

##### III . 3 . 1 . 1 *Mise en oeuvre d'approximations*

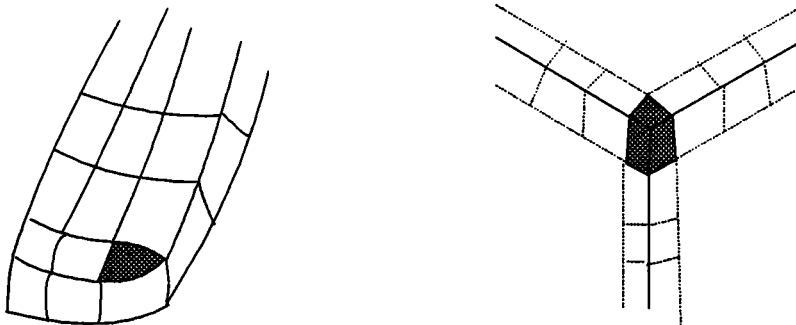
L'approximation la plus évidente est d'abandonner la notion de surface gauche, pour l'approcher par un ensemble de facettes planes ou presque planes , en particulier dès qu'une opération booléenne doit être effectuée. Pour tous les modèles de surfaces que nous avons étudiées , la méthode utilisée "pour facettiser" une surface est la méthode d'échantillonnage et ensuite de construire le modèle B-Rep associé. Le volume de données à conserver est important si on veut obtenir une approximation précise par contre l'évaluation des nouvelles frontières est simplifiée car elle se réduit à des intersections entre objets plans . Le modèle est donc fonction du facteur de polygonalisation.



On peut tenter de conserver des traces de la définition des surfaces gauches, ce qui revient en quelque sorte, à mettre en place une notion d'historique. Pour conserver la définition mathématique de la surface, on peut, au lieu de l'échantillonner, la subdiviser récursivement en quatre carreaux jusqu'à ce que les carreaux soient suffisamment "plans". On conservera la subdivision dans un quadtree.

Si on conserve en parallèle la définition mathématique de la surface associée à son approximation polyédrique, il faut après chaque modification de la surface (par déplacement d'un point de contrôle) redéfinir entièrement le modèle associé.

On peut toutefois mettre en place une approximation, dont l'objectif est de maintenir une structure topologique cohérente [FOR 78] : l'objectif est de créer une structure du type modélisation B-Rep, qui approche le modèle à surfaces gauches.



*La partie gauche du schéma représente la surface que l'on veut modéliser, la partie droite, le modèle "grossier" associé.*

La stratégie devient alors de construire cette structure "grossière", puis de la lisser pour aboutir au modèle final. Le modèle grossier est utilisé comme base de la topologie et comme outil d'évaluation. Cette approche n'est cependant pas généralisable à toutes les formes et implique une discipline à la construction. De plus ce type de modélisation a du mal à prendre en compte les modifications des solides après lissage .

### III . 3 . 1 . 2 *Assemblage de faces ou d'objets*

Il s'agit de décrire un solide comme un assemblage cohérent de faces et/ou de définir des solides complexes par des assemblages cohérents de solides. Les solides sont définis par collage de faces le long du arête commune ou par collage de solide le long de faces jointives. Le collage permet d'éviter les problèmes liés à la prise en compte des opérations booléennes dans leur généralité.

Les faces de type surfaces gauches peuvent être prises en compte dans la structure B-Rep, en considérant des solides constitués par des faces gauches assemblées. La topologie Face/Arête/Sommet, pourra être maintenue, en gérant les correspondances :

- face = surface gauche
- arête = courbe gauche
- sommet = point

Une face gauche est considérée de la même façon qu'une face analytique. Elle est limitée par des courbes (avec éventuellement des courbes internes). La mise en oeuvre d'un tel modèle, en se contentant d'assembler des faces jointives ne pose pas de problème insurmontable. Cependant, dès que l'on souhaite mettre en oeuvre des opérations complexes (booléennes, par exemple), il est nécessaire de définir formellement la notion de carreau restreint.

Dans ASM [CAS 85], un solide est défini comme un assemblage sans recouvrement de composants solides définis comme des hyper carreaux. Un hyper carreau est défini sur un espace solide comme un carreau paramétrique (cf figure 4). Un hyper carreau est la généralisation d'un carreau de surface. Le domaine paramétrique du solide a pour axes  $u, v, w$ .

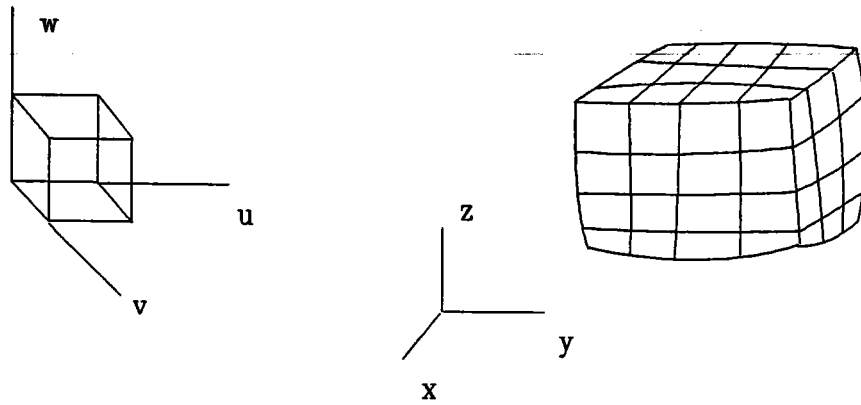


FIGURE 4

On définit un hyper carreau de degré 3 selon les directions paramétriques par :

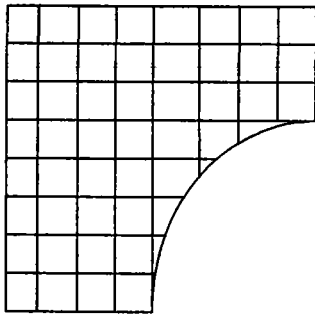
$$Z(u,v,w) = \sum_{k=1}^4 \sum_{j=1}^4 \sum_{i=1}^4 P_{ijk} u^{4-i} v^{4-j} w^{4-k}$$

$(P_{ijk})$  représente le réseau de contrôle du solide .

Dans ce cas encore, le modèle utilisé impose une discipline dans la construction des solides, en mettant en oeuvre, soit des opérations bien adaptées au modèle utilisé (solide réglé entre deux surfaces gauches, solide correspondant à un solide de base déformé, extrusion), soit en décomposant le solide à construire en solides qui s'assembleront sans se recouvrir. Ce modèle ne prend pas en compte la totalité des opérations booléennes. Il gère uniquement le collage de 2 objets . Cette opération n'est qu'une restriction de l'opération d'union.

### III . 3 . 1 . 3 Mise en oeuvre de carreaux restreints

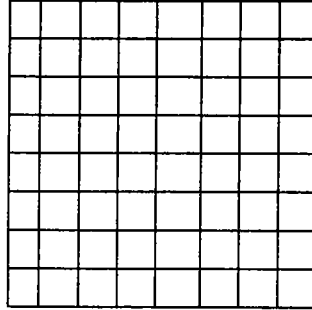
Un solide est défini par une collection de faces. Ces faces sont définies paramétriquement comme des carreaux restreints. Rappelons brièvement qu'un carreau restreint est défini par la donnée d'un carreau de surface paramétrique et la donnée d'un sous ensemble R de l'intervalle paramétrique. Pour définir de façon formelle un carreau restreint , on peut utiliser une structure hiérarchique de type CSG. L'arbre ainsi définie est un arbre 2D.



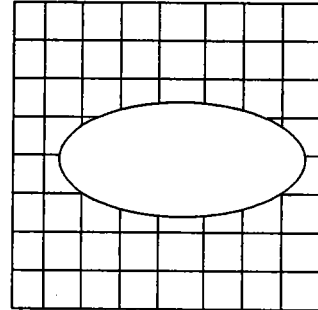
Espace paramétrique

R

$D = P1 \cap P2 \cap P3 \cap P4$



E = extérieur d'une ellipse



Considérons l'exemple des opérations booléennes [CAS 87].

Etant donnés deux solides S1 et S2, définis par des faces de type carreaux restreints, une opération booléenne se déroulera de la manière suivante :

Préparer les données ;

*on peut dans cette phase mettre en oeuvre les méthodes habituelles pour faciliter et accélérer les calculs, telles que tests sur les boîtes englobantes ...*

Pour chaque face Fi de S1 faire

Pour chaque face Fj de S2 faire

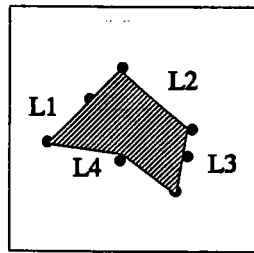
calculer l'intersection de Fi et Fj ;

*le résultat est une suite ordonnée de points (plusieurs méthodes peuvent être employées), pour lesquels on a déterminé les paramètres (u,v) pour Fi et (u',v') pour Fj. On obtient donc à la fin de la double boucle, pour chaque face, un ensemble de segments définis dans le domaine paramétrique de chaque face.*

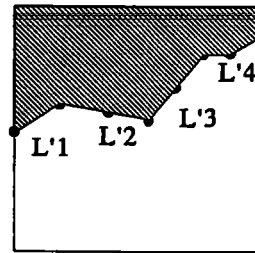
Pour chaque face faire

constituer des courbes fermées

*cette opération consiste à relier les segments définis dans chaque espace paramétrique en courbes fermées, qui définiront un intérieur et un extérieur (figure 5 ).*



espace paramétrique  
associé à  $F_i$



espace paramétrique  
associé à  $F_j$

FIGURE 5

- rechercher les portions à conserver

*cette opération consiste à rechercher quelles sont les portions (demi-plans) à conserver, c'est-à-dire, pour chaque courbe fermée dans l'espace paramétrique, l'intérieur ou l'extérieur. Cette opération est explicitée ci-dessous. Le résultat est un carreau restreint, qui correspond à la définition du solide utilisée.*

Explicitons l'opération "recherche des portions à conserver".

Pour effectuer cette recherche, on peut se contenter de déterminer la caractéristique d'un point quelconque, puis de reconstituer par adjacence les parties à conserver. On effectue donc les opérations suivantes :

un point quelconque est choisi dans l'espace paramétrique ;

*On s'arrange toutefois pour que ce point ne soit pas trop proche d'une courbe paramétrique*

on détermine le point correspondant en 3D ;

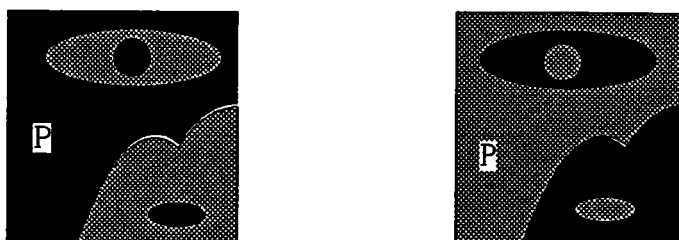
on détermine si ce point appartient ou non à l'autre solide ;

on détermine en fonction de l'opération booléenne considérée et de la classification du point par rapport à l'autre solide s'il appartient au résultat ou non ; (cf figure 6)

*cette opération est très employée dans tous les algorithmes réalisant les opérations booléennes. Si B est l'autre solide, elle est mise en oeuvre par :*

si (union) et  $(P \in B)$  alors négatif  
 si union et  $(P \notin B)$  alors positif  
 si (intersection) et  $(P \in B)$  alors positif  
 si (intersection) et  $(P \notin B)$  négatif  
 si (différence) et  $(P \in B)$  alors positif  
 si (différence) et  $(P \notin B)$  alors négatif

on détermine par adjacence les caractéristiques de chaque îlot ;



**FIGURE 6**

*la partie hachurée de la figure de gauche représente la surface dans l'espace paramétrique qui est conservée suivant la classification du point P (ici le point P n'est pas conservé). La figure de droite représente la surface conservée si le point P appartient au domaine .  
 le passage d'un îlot à un autre implique un changement d'état (quelques cas particuliers peuvent toutefois poser des problèmes)*

### III . 3 . 2 Approche CSG

Il pourrait paraître assez facile sur un plan théorique d'intégrer les surfaces gauches à un modèle CSG [MAN 88]: en effet, il "suffirait" de considérer un modèle dont les feuilles seraient des demi-espaces définis par une frontière de type surface gauche. Malheureusement, un obstacle incontournable ne permet pas cette approche : une surface paramétrique (que ce soit une Bézier, une B-Spline, une Nurbs ...) ne définit pas des demi-espaces ! En effet, on ne dispose pas d'une formulation du type  $f(x,y,z)$ , qui permettrait de définir un côté "positif" et un côté "négatif".



Une telle approche peut donc s'effectuer selon deux méthodes :

- une première méthode consiste à conserver la formulation paramétrique des surfaces, mais de considérer des objets de base qui sont, non pas des demi-espaces, mais des objets primitifs solides, dont la topologie s'appuie sur des surfaces gauches déformables ;
- une seconde méthode consiste à abandonner la formulation paramétrique des surfaces et à définir des demi-espaces à partir de formes implicites ( $f(x,y,z)$ ).

### III . 3 . 2 . 1 *Objets de bases définis par des surfaces gauches*

On considère un arbre CSG, dont les feuilles sont des objets primitifs, parfaitement définis du point de vue de leur topologie. Les faces de ces objets sont définies dans une représentation par des réseaux de points de contrôle et une définition paramétrique. Le choix de la définition paramétrique est, à priori, quelconque. Cependant l'une des représentations (les Nurbs) paraît la mieux adaptée, dans la mesure où elle permet de représenter exactement les quadriques : ainsi les cylindres, sphères ... seront représentés dans la formulation Nurbs adéquate [TIL 83].

L'opérateur pourra donc :

- appeler et placer un objet primitif (par exemple, un cube) ;
- modifier sa forme, puisque les faces seront représentées par des équations paramétriques
- composer ces objets par opérations booléennes.

On pourra étendre cette classe d'objets à des objets dont on maîtrise les méthodes de construction, comme des objets par extrusion.

Les problèmes à résoudre ne sont cependant pas évidents. En effet :

- étant donné un arbre CSG composé de la manière décrite ci-dessus, il faut évaluer cet arbre pour différentes opérations. On pourra mettre en oeuvre les deux évaluation classiques :

\*évaluation directe de l'arbre (méthode du tracé de rayons) : cette évaluation sera utile lors d'affichages, mais impliquera des temps de traitement assez importants [KAJ 82]. Nous développerons cette méthode dans le chapitre suivant.

\*évaluation d'un modèle B-Rep : cette méthode pourra déboucher, soit sur un B-rep approché (par exemple, par facettes planes), ce qui sera relativement facile à effectuer en utilisant des algorithmes déjà développés pour les modèles B-Rep existant, soit un B-Rep "exact", ce qui implique la mise en oeuvre d'algorithmes complexes . Une méthode pour résoudre ce problème est d'utiliser une subdivision récursive des surfaces dont on veut connaître les intersections. Quand les morceaux de carreaux sont suffisamment "plans", on calcule les intersections entre les différentes parties. Les points obtenus serviront de point de départ à une méthode de Newton. Il faut ensuite recoller les différents fragments de courbes obtenus [HOU 85].

- Assurer la cohérence d'un objet primitif, après des modifications de sa forme est un problème très complexe. Nous reparlerons de ce problème en III.5.3. On peut cependant considérer, que dans la plupart des cas, l'opérateur ne fait que des manipulations cohérentes.

### III. 3 . 2 . 2 *Formes implicites*

Il est bien entendu possible de manipuler des objets de base sous leur forme canonique, sans les "facettiser". Ainsi, dans GMSolid [BOY 79], les surfaces sont des quadriques et les intersections sont calculées sur ce modèle. Pour chaque quadrique , on utilise un traitement spécifique pour calculer l'intersection rayon/surface. Là encore l'évaluation du modèle doit être faite pour un certain nombre d'opérations et pourra se faire, soit par traversée de l'arbre, soit par évaluation d'un modèle B-Rep.

Dans le cas des surfaces définies par produit tensoriel, il semble qu'il soit possible de déterminer une forme implicite de toute forme rationnelle [SED 84] en utilisant la théorie de l'élimination. Il s'agit d'exprimer une condition nécessaire et suffisante pour assurer qu'un système d'équations algébriques a les mêmes racines que les racines à l'infini d'un polynôme, appelé le résultant (par exemple, les résultants de Sylvester ou Bezout et Caylay). Le principal handicap de cette méthode est le degré élevé de l'équation implicite que l'on détermine. Par exemple, pour une surface paramétrique de degré  $m$  en  $u$  et de degré  $n$  en  $v$  , on obtient une équation implicite de degré  $2nm$ .

### III . 3 . 3 Conclusion

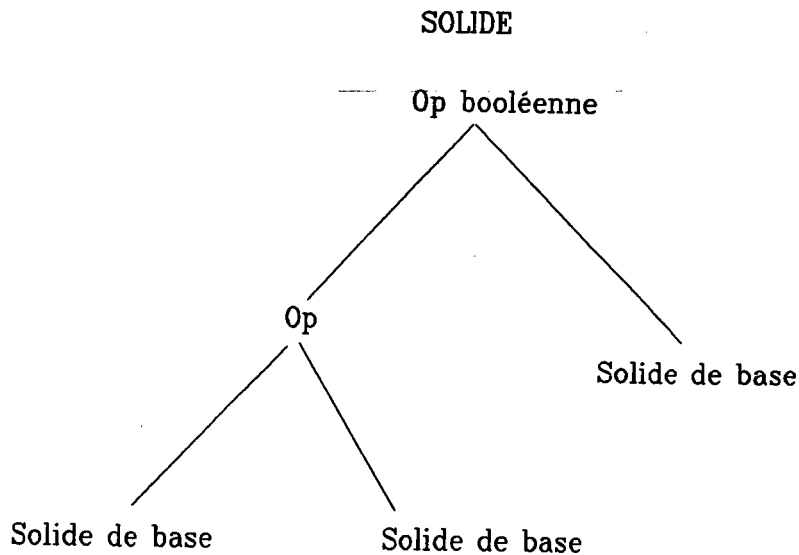
L'étude des différents types de modélisation solide nous a conduit à penser , qu'une solution à l'intégration des surfaces gauches dans un modèle de solides passerait soit par un modèle de type B-Rep soit un modèle de type CSG . En effet , l'approche spatiale ne permet pas facilement de conserver le modèle mathématique ; l'extrusion généralisée est limitée dans le type d'objets modélisables facilement. Une modélisation solide de type B-Rep avec une approximation entraîne la perte de la notion de surface, un volume de données important. Cette structure permet difficilement la modification des solides ainsi modélisés. Il faut tout de même noter que les traitements sont très simples dans la mise en oeuvre des opérations booléennes. Les techniques de "modèle grossier", d'assemblage et de définition "d'hyper patches" ne permettent pas aisément la mise en oeuvre des opérations booléennes bien que cette voie soit très prometteuse , notre choix s'est porté sur une modélisation de type CSG avec des feuilles définies comme des solides primitifs. Ce qui nous a séduits c'est la concision de la structure et la "prise en compte naturelle" des opérations booléennes. Les feuilles sont des solides de base plutôt que des demi-espaces car les méthodes pour définir les équations implicites des surfaces paramétriques induisent des équations algébriques difficiles à manipuler et à gérer.

### III . 4 Le modèle solide choisi

En fonction de l'étude que nous avons menée au chapitre précédent , le choix entre un modèle B-Rep et un modèle CSG s'est orienté vers le second , d'une part parce que sa définition se rapproche plus des considérations mathématiques pour la modélisation des solides que nous avons évoqués au chapitre 1 et d'autre part afin d'envisager à travers des manipulations de l'arbre l'intérêt de la conservation de l'historique et de la non unicité de la modélisation (modification de partie, prise en compte de la conception par caractéristique de forme). Il est clair qu'il s'agit d'un choix délibéré, non fondé sur un critère de qualité difficile à définir entre les deux modes de représentation.

#### III . 4 . 1 Définition de l'arbre CSG

Les arbres CSG que nous allons utiliser sont des arbres définis à partir de solide de base ( ils définiront les feuilles) . Un solide sera défini par composition d'opérations booléennes régularisées (intersection , réunion , différence ) sur les feuilles .



Les feuilles sont la réunion de surfaces gauches modélisés par des NURBS . Nous expliciterons la définition des feuilles dans le prochain paragraphe. A tout moment , nous considérons les feuilles comme des solides cohérents. Un solide cohérent est défini comme un sous espace de  $R^3$  fermé et orientable. C'est à dire qu'un solide occupe une portion finie de l'espace et qu'en tout point du solide, on définit l'intérieur et l'extérieur. Pour réaliser un solide cohérent, les différentes faces du solide doivent vérifier un ensemble de règles:

- 1 - Toutes les faces doivent être fermées
- 2 - Toutes les faces doivent être bornées
- 3- Toutes les faces doivent être d'un seul morceau
- 4 - Chaque face doit être orientable
- 5 - Les faces du solide ne doivent pas se couper entre elles sauf le long des courbes frontières.

La condition (4) est une condition supplémentaire par rapport au modèle polyédrique: dans le cas de faces planes, une face ne peut se couper elle même.

\*Pour assurer la fermeture d'une face , il suffit d'imposer à toutes les lignes ou toutes les colonnes de points de contrôle d'avoir un premier point égal au dernier.

\*Par définition, les faces décrites par des NURBS sont bornées.

\*Pour être sûre qu'une NURBS soit d'un seul morceau , il suffit d'imposer aux valeurs nodales une multiplicité au plus égale à l'ordre de la spline.

\*La vérification de la propriété 4 est plus difficile à mettre en oeuvre. En effet, une face sera orientable si elle ne "s'auto-intersecte pas". D'un point de vue mathématique, il n'existe pas de démarche heuristique qui permettent de prouver l'existence d'une "auto-intersection". Nous sommes obligé d'utiliser une méthode algorithmique.

Dans un premier temps, nous décomposons la face en un ensemble de carreaux de Bézier (le détail de cette décomposition sera donné au prochain paragraphe). Puis nous échantillonons les carreaux jusqu'à obtenir des éléments quasi plans et nous vérifions l'existence d'une intersection. Notre but n'étant pas de déterminer toutes les intersections, on s'arrêtera dès que l'on en aura détectée une. De plus, nous savons qu'il suffit de tester l'intersection uniquement sur les carreaux dont les réseaux de contrôle se coupent.

\* les faces d'un même solide ne doivent se couper que le long des courbes frontières; pour vérifier cette condition, nous utiliserons la même méthode que précédemment sur les carreaux des 2 faces.

En utilisant les opérations pré-citées , nous pouvons garantir la cohérence du modèle .

### III . 4 . 2 Implantation de l'arbre CSG

L'arbre CSG est défini par une structure d'arbre binaire. Aux noeuds de l'arbre, on trouve soit une opération booléenne soit un solide de base. Une opération booléenne sera décrite par son nom. Une feuille pointera sur un solide primitif stocké dans un modèle.

```

nature_noeud_CSG = (union , intersection , différence , feuille )
arbre_CSG       = pointeur sur noeud_CSG
noeud_CSG       = enregistrement
                   pere : arbre_CSG
                   si nature_noeud_CSG =
                     union , intersection , différence : fg fd : arbre_CSG
                     feuille : pointeur sur un solide de base

```

Dans la littérature , beaucoup d'arbre CSG considèrent un troisième type de noeud à savoir les transformations solides. On peut facilement montrer qu'il est équivalent ,dans un arbre CSG de tenir compte des transformations solides à un niveau quelconque de l'arbre ou redescendre cette information au niveau des feuilles. Puisque les solides de base sont définis à partir d'un modèle mathématiques (les NURBS) qui prend très facilement en compte toutes les transformations solides, nous considérerons que les transformations sont prises en compte au niveau des feuilles.

### III . 5 Modèle des solides de base

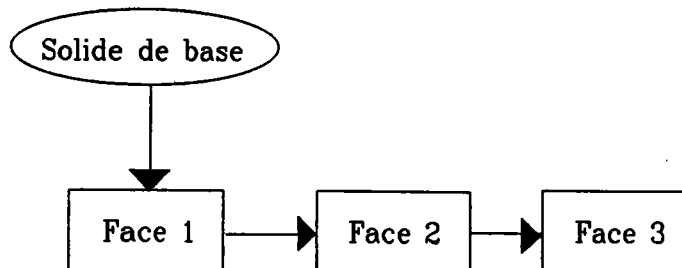
Les feuilles de l'arbre sont définies dans un modèle de type "Assemblage de faces". Nous avons créé un lien entre le modèle "solide CSG" et le modèle "Solide de Base". Les noeuds terminaux de l'arbre CSG pointent sur les structures du modèle "Solide de Base".

#### III . 5 . 1 Définition

Actuellement , les primitives de base du modèle solide sont des sphères , des cônes , des cylindre, des tores, des boîtes et des pyramides. Cette définition n'est pas restrictive et d'autres solides de base peuvent être envisagées. Par exemple, des primitives définies par révolution ou par extrusion généralisée. Toutes ces primitives sont définies à partir du modèle mathématiques des NURBS décrit au premier chapitre (on trouvera en annexe le détail de la définition). Chaque primitive est constituée d'1 ou 2 ou 3 faces.

Ce nombre de faces suffit à la modélisation de n'importe quel solide. Nous pouvons de cette façon modéliser tous les solides qui sont intégrés dans les modèles classiques par arbre de construction. De plus, trois faces maximum seront suffisantes à l'intégration de nouvelles primitives (définies par extrusion généralisée , par révolution ou "à la main "). La première face servira à décrire l'enveloppe du solide ,les deuxième et troisième (si nécessaire) assureront la compacité du solide. Ces nouvelles primitives devront également vérifier les règles de Définition d'un solide cohérent. Une primitive sera constituée :

- 1 face si la surface est compact (fermée et bornée) par exemple dans la cas d'une sphère ou d'un tore
- 2 faces si la surface est "semi fermée "( cône ou pyramide )
- 3 faces si la surface est ouverte (cylindre ou boîte )



Pour chaque face , nous conservons les informations suivantes :

**\* Définition de la surface B-Spline décrivant la face**

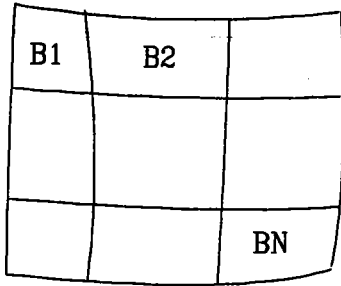
- table des points de contrôle
- table des noeuds selon u et selon v
- ordre de la surface suivant les 2 paramètres

**\* Décomposition de la surface de la surface B-spline en une mosaïque de carreaux de Bézier de même degré que la surface B-spline initiale .**

**\* Définition des carreaux de Bézier dans la base canonique des polynômes .**

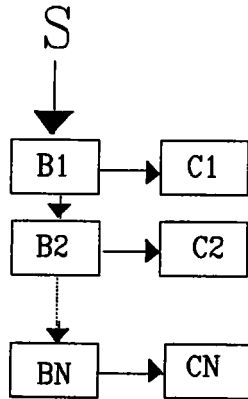
La définition de la surface sous la forme B-spline rationnelle est utile , au moment de la création de la face . C'est également cette formulation qui est utilisée si l'on déforme les solides car d'après les conclusions du premier chapitre nous savons que c'est cette formulation qui est la plus souple. La représentation dans la base canonique est utilisée lors de la mise en oeuvre du Ray-Casting . Une surface NURBS est une polynomiale par morceaux , soit S une telle surface. De ce fait, il existe pour chaque intervalle paramétrique  $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$  un système d'équations définissant la surface sur ce domaine. Pour l'algorithme d'intersection Droite/Surface, il est plus intéressant de considérer S comme un ensemble de carreaux définis dans la base canonique des polynômes. Pour définir cet ensemble de carreaux , on utilise une base intermédiaire : la base des polynômes de Bernstein. Ce changement de base correspond à exprimer la surface B-Spline en un ensemble de carreaux de Bézier. Ce changement de base sera détaillé dans le prochain paragraphe.

S une surface B Spline



$$S = B1 \cup B2 \dots \cup BN$$

$$S = C1 \cup C2 \dots \cup CN$$



La formulation de la surface S dans la base canonique permet de déterminer les points d'intersection entre elle et une droite. Cette détermination est le noyau de l'algorithme du Ray-Casting.

Nous utiliserons un système (A) d'équations pour définir une droite de l'espace( il existe toujours deux plans P1 et P2 tels qu' une droite soit égal à P1 inter P2).

$$(A) \begin{cases} AX + BY + CZ + D = 0 \\ ApX + BpY + CpZ + Dp = 0 \end{cases}$$

La méthode pour déterminer les points d'intersection entre une surface et une droite consiste à injecter les différentes équations paramétriques ( écrites dans la base canonique) de la surface dans la système (A). Nous décrivons , un peu plus plus tard, le détail de ce processus (III.6.2.2).



Nous conservons également la définition de la surface B-Spline sous la forme de Bézier pour pouvoir définir des volumes englobants aux différents carreaux. Le volume englobant d'un carreau ou d'une surface est un parallélépipède rectangle (parallèle aux axes) qui contient toute la surface ou tout le carreau. Le volume englobant du carreau sera défini à partir de ses points de Bézier. Un carreau de Bézier est toujours contenu dans l'enveloppe convexe de son réseau de contrôle. Il est intéressant de conserver ces différents volumes englobants car l'intersection Droite/Surface prend beaucoup de temps, si le volume englobant n'est pas coupé par la droite il est inutile d'envoyer le processus d'intersection. Nous connaissons un englobant de S (définis par des points de contrôle B-Spline) mais les englobants des différents carreaux sont tous contenus dans englobant de S et donc ils s'assurent un meilleur filtre à notre algorithme.

### III . 5 . 2 Construction d'un solide de base

#### III . 5 . 2 . 1 Construction d'un solide constitué d'une seule face

**1ère ETAPE :** A partir des données intrinsèque du solide ( par exemple centre rayon ) le logiciel génère automatiquement la surface B-spline correspondante . Pour ce faire ,le logiciel utilise la méthode décrite dans le chapitre un . C'est à dire que le logiciel génère dans un premier temps les points de contrôle et les tables des noeuds correspondant à la quadrique primitive puis il applique à ce solide la transformation affine adéquate qui le placera dans l'espace conformément aux souhaits de l'utilisateur. On trouvera dans les annexes le détail de ces différentes constructions.

**2ème ETAPE :** consiste à définir les points de Bézier associées à la surface B-Spline. Pour toutes les surfaces B-Spline, il est toujours possible de déterminer un nouveau réseau de contrôle qui , sur chaque intervalle paramétrique  $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$ , équivaut à la définition de ce carreau dans la base des Bernstein. Soit S une surface B-spline d'ordre k dans la direction u et d'ordre l dans la direction v. Pour déterminer ce nouveau réseau de contrôle, il faut augmenter la multiplicité des noeuds selon u jusqu'à (k-1) et augmenter la multiplicité des noeuds selon v jusqu'à l(l-1). Notre méthode est fondée sur l'algorithme de Böhm [ BÖH 84]. Cet algorithme permet d'insérer une valeur nodale  $u^*$  et de calculer les nouveaux points de contrôle. Cet algorithme a été décrit pour une courbe B-spline mais se généralise aisément aux surfaces car elles sont définies par produit tensoriel (cf annexe).

description de l'algorithme : Passage B-spline/ Bézier

PARAMETRES : T : matrice des points de contrôle  
ordre\_u : ordre de la surface selon u  
ordrev : ordre de la surface selon v  
nbu : nombre de lignes de T  
nbv : nombre de colonnes de T  
Tndu : table des noeuds selon u  
Tndv : table des noeuds selon v

DEBUT

POUR tous les éléments de Tndu FAIRE

TANT QUE multiplicité de u < (ordre\_u-1) FAIRE

ajouter u à Tndu

calcul de la nouvelle ligne de points de contrôle

nbu = nbu+1

FINTANTQUE

FINPOUR

POUR tous les éléments de Tndv FAIRE

TANT QUE multiplicité de v < (ordrev-1) FAIRE

ajouter v à Tndv

calcul de la nouvelle colonne de T

nbv = nbv+1

FINTANTQUE

FINPOUR

FIN

**3ème étape :** A l'aide des points de Bézier , nous définissons 2 listes de carreaux de même degré que la surface S. Une première liste dans la base des polynômes de Bernstein et la seconde dans la base canonique des polynômes. Dans la première étape, nous avons défini un nouveau réseau de contrôle à la surface B-Spline construite par l'utilisateur. Sur chaque intervalle paramétrique non vide  $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$  les points de contrôle agissant sur cette portion de surface sont :  $[P_{m,n}]$  pour m variant de  $(i-k+1)$  à i et n variant de  $(j-l+1)$  à j. Les points  $[P_{m,n}]$  sont également les points de Bézier de la définition de ce carreau dans la base des Bernstein. Pour la surface NURBS S , nous initialisons une suite de carreaux qui forment une partition de S dans la base des Bernstein. Puis, à chacun des carreaux, nous appliquons le changement de base Bernstein / Canonique. On obtient pour chaque carreau une formulation du type :

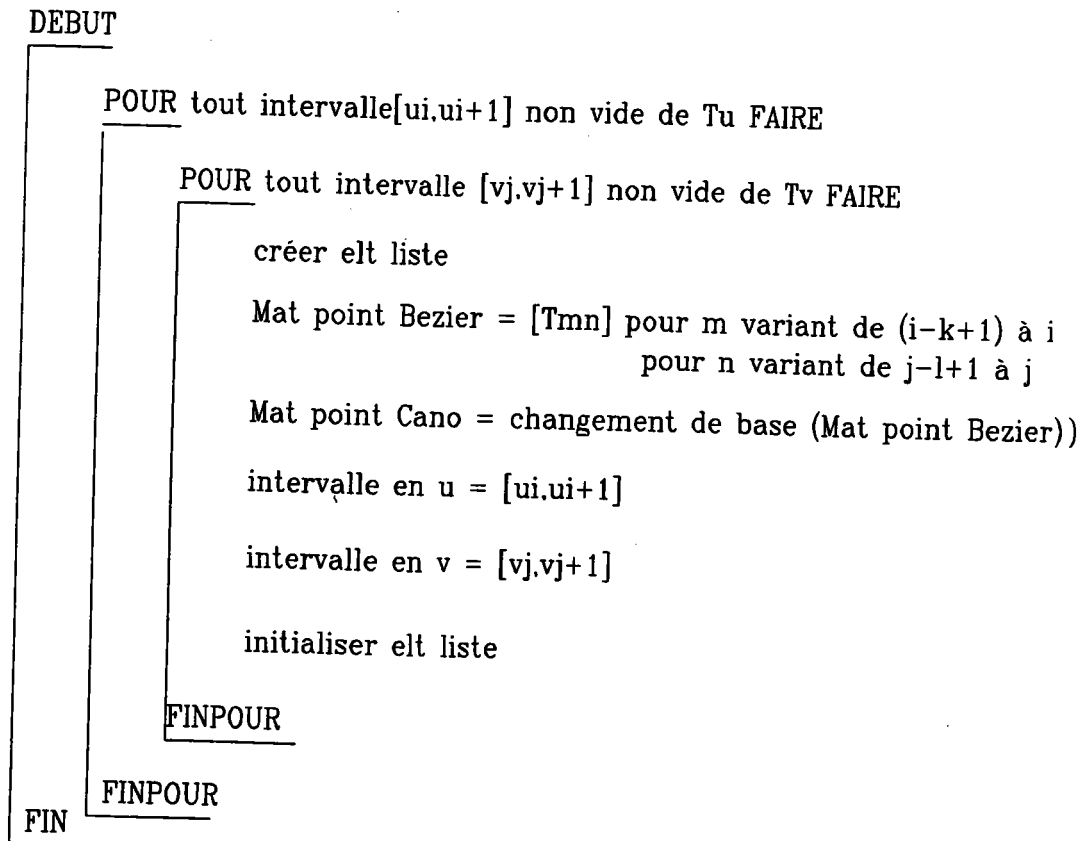
$$C(u,v) = \sum \sum A_{i j} u^i v^j$$

Cette formulation est mieux adaptée à la résolution de "intersection Rayon / Surface "

## Description de l'algorithme

Paramètres d'entrée : T : table des points établie dans l'étape 1  
Tu , Tv : tables des noeuds associées à T  
ordreu , ordrev : ordres de la surface BSpline

Paramètres de sortie : 2 listes de carreaux  
dans la base de Bernstein  
dans la base canonique

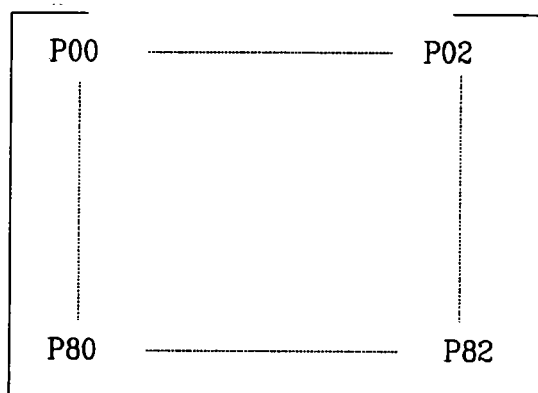


En résumé, la méthode de création d'un solide primitif à une face se décompose de la façon suivante:

- 1) Création de la surface NURBS
- 2) Passage (B-Spline -Bézier) (étape 1)
- 3) Décomposition en carreaux de Bézier (étape 2)
- 4) Changement de base (étape 3)

### III . 5 . 2 . 2 Construction d'un solide constitué de plusieurs faces :

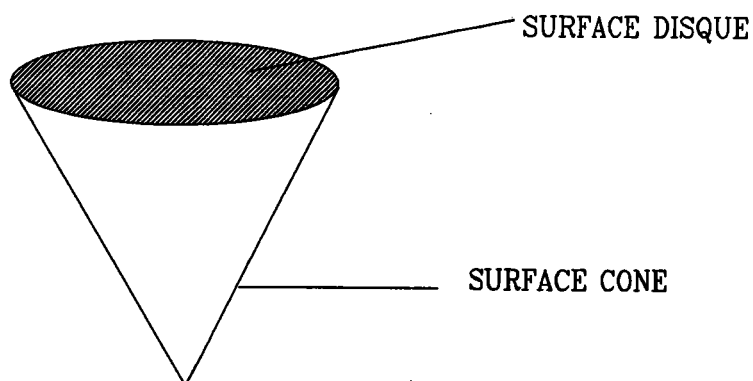
Les solides ayant plusieurs faces ont une face privilégiée : la première face les autres sont des faces de fermeture. Par exemple , prenons le cas du cône . Nous définissons comme précédemment une surface cône par la donnée du centre de la base, de sa hauteur et du vecteur de la direction principale . Ce qui définit une matrice  $[P_i j]$  des points de contrôle et 2 tables nodales.



$$tndu = [ u_0 , \dots , u_{11} ]$$

$$tndv = [ v_0 , \dots , v_5 ]$$

Pour définir le volume cône , il reste à déterminer les points de contrôle de la surface Disque qui fermera le cône.



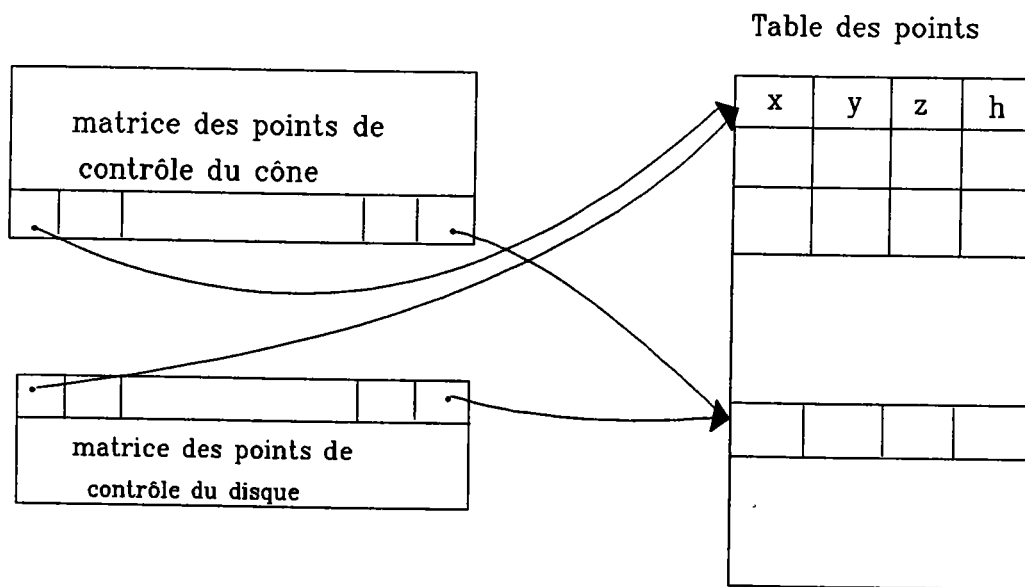
La courbe frontière circulaire de la base du cône est une courbe B-spline définie par la ligne des points  $[P_i]_{i=0..8}$  de la matrice MAT\_CONE et de la table des noeuds Tndu .

Le Disque sera donc défini comme une surface B-Spline avec les contraintes suivantes:

**1ère ligne de la matrice disque = dernière ligne de la matrice cône**

**Tndu\_disque = Tndu\_cone .**

Pratiquement , dans le modèle, les matrices des points de contrôle sont définies comme des tableaux d'entier. Chaque entier est une entrée dans la table des points. Pour faire coïncider géométriquement 2 points et les lier topologiquement, il suffit de leur donner la même entrée dans la table des points.



De même, nous avons créé dans le modèle, une table des scalaires qui contient toutes les valeurs nodales que le système utilise. De la même façon, les tables des noeuds des surfaces sont modélisées dans un tableau d'entier, pour lier 2 valeurs nodales, il suffit de leur donner la même entrée dans la table des scalaires.

De cette façon, on garantit la fermeture du solide cône pendant la phase de déformation. Ensuite, les étapes de 2 et 3 sont effectuées pour toute les faces du solide.

### III . 5 . 3 Déformation des solides primitifs

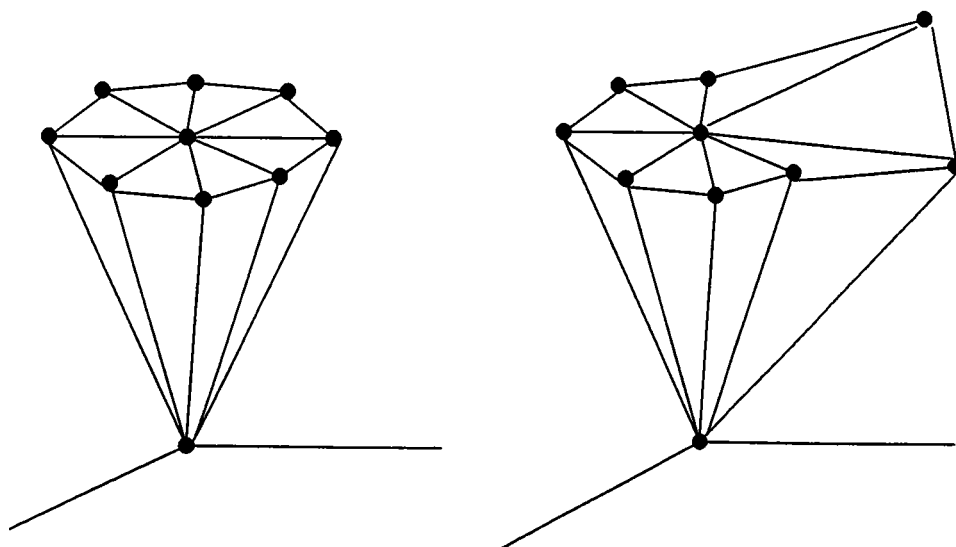
Le modèle, ainsi défini, ne se différencie pas du point de vue de l'utilisateur d' un modèle classique intégrant les quadriques. Dans ce paragraphe , nous mettrons en évidence l' apport de la définition mathématique des faces.

Notre objectif est de pouvoir disposer de solides à " formes libres" pour la conception de nouveaux objets . Les surfaces B\_spline offrent une grande liberté de création ou de déformation .

A partir d'un solide de base , l'utilisateur aura toute latitude pour concevoir de nouvelles formes à partir de celles déjà acquises. La déformation se fera de manière interactive par déplacement ou par définition de nouveaux points de contrôle. L'existence d'un lien topologique entre les différentes faces d'un même solide permet la répercussion immédiate de la modification d'une face sur toutes les autres faces du solide. Par exemple, si on modifie un point de contrôle de la surface cône et que ce point appartienne également à la définition du disque. Comme ce n'apparaît qu'une seule fois dans la table des points, la modification de ses coordonnées est immédiatement prise en compte également dans la définition du disque.

Ensuite, pour toutes les faces modifiées, il faudra reprendre les 2ème et 3ème étapes de la construction d'un solide. En effet, la liste des carreaux de Bézier et des carreaux "canoniques" devra être recalculée en fonction des nouvelles tables des points et des nouvelles valeurs nodales des faces déformées. Le processus sera évidemment invisible pour l'utilisateur mais il est essentiel pour la cohérence de l'application en cours.

Réseau de contrôle du disque et du cône



### III . 6 Principes de l'évaluation de l'arbre

Il existe deux façons d'évaluer un arbre CSG;

- \* une méthode directe , qui consiste à évaluer directement la propriété cherchée sans évaluer au préalable l'objet par ses frontières. L' analyse se fait directement sur les branches de l'arbre.

#### *Exemple*

- appartenance d'un point à un solide peut se faire directement par appartenance du point par rapport à chacun des solides de base puis classification des résultats par rapport au sommet de l'arbre.
- visualisation réaliste : par un méthode de Ray-Casting

- \* une méthode indirecte, qui consiste à déterminer les frontières du solide puis à évaluer la propriété cherchée sur le modèle par les frontières.

#### *Exemple*

- Appartenance d'un point à un solide : On utilise la méthode de la demi-droite. On compte le nombre d'intersection entre une demi-droite quelconque qui part du point et le solide. Nous reparlerons ultérieurement (III.6.2.1) de cette méthode.

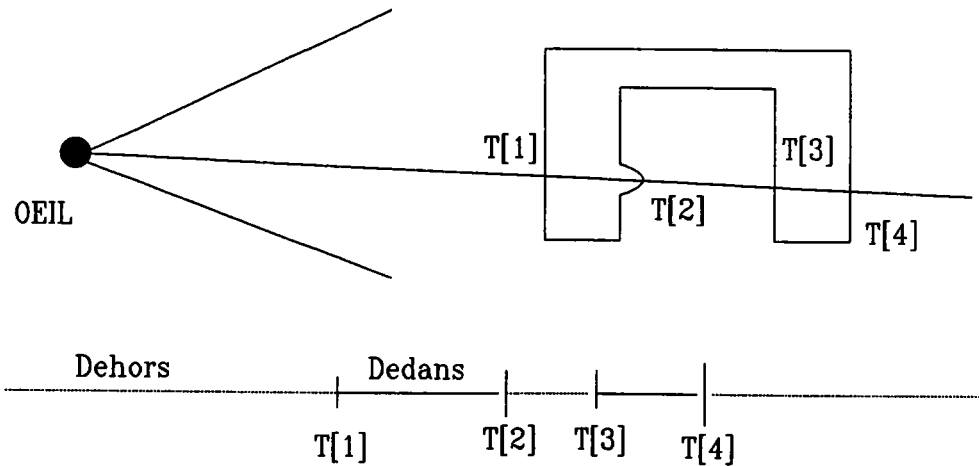
Pour évaluer les frontières d'un solide, il faut déterminer les courbes frontières entre les différentes faces des solides de base qui prennent part à la définition du solide et également, dans le cas présent, quelles parties de l'intervalle paramétrique de la surface NURBS associée à la face est utile. Pour modéliser formellement ce concept, il faut introduire la notion de carreau restreint(cf II ). Cette mise en oeuvre n'est pas simple parce que les NURBS n'ont pas de définition implicite simple. De plus, il est "très délicat" de définir une courbe frontière qui appartienne aux 2 surfaces qui se coupent.



Nous avons privilégiée une méthode directe pour évaluer l'arbre CSG. Nous avons implanté un Ray-Casting pour la visualisation. C'est une méthode "naturelle" pour évaluer un arbre de construction. Elle est fondée sur le principe "Diviser pour Régner". Elle est implicite au sens que l'on ne connaît pas explicitement les frontières de l'objet. Nous obtenons, tout de même, suffisamment de renseignements pour toutes les applications que nous souhaitons mettre en oeuvre. Pour réaliser cette évaluation, il fallait dans un premier temps déterminer les points d'intersection entre un rayon (droite) et une surface, puis déterminer les portions de la droite qui sont HORS et DANS le solide de base, et enfin, combiner les différents résultats obtenus en respectant les opérations booléennes mises en oeuvre dans l'arbre CSG. La méthode du Ray-Casting nous a permis d'éviter de prendre en compte le problème délicat de l'intersection Surface/Surface et de tenir compte uniquement de l'intersection Rayon/Surface.

### III . 6 . 1 Principe du Ray-Casting [ROT 82]

Le principe du Ray-Casting est très simple, il simule l'inverse du processus de la photographie. Pour chaque élément de l'image (pixel), on envoie un rayon jusqu'à l'oeil de l'observateur, on calcule toutes les intersections du rayon avec la scène et on les classe suivant leur distance à l'oeil.



Il faut ensuite combiner les résultats en tenant compte des opérations booléennes mises en oeuvre pour la définition du solide. La technique du Ray-Casting permet entre autres la visualisation réaliste (avec ou sans ombrages), la visualisation au trait, analyse (calcul de volumes, de centre d'inertie, etc) de solides complexes.

Pour mettre en oeuvre un Ray-Casting dans notre logiciel, nous devons déterminer toutes les intersections entre une droite et une surface NURBS.

### III.6.2 Intersection Rayon /Surface

Le problème à résoudre est de calculer les différentes intersections entre une droite et une surface NURBS. Deux méthodes sont possibles : une méthode qui utilise directement les équations de la droite et de la surface et une autre qui utilise une approximation de la surface.

#### III . 6 . 2 . 1 Méthode par approximation

La méthode consiste à échantillonner la surface en éléments plans puis à déterminer si le point d'intersection entre la droite et le plan appartient à la restriction considéré. L'approximation de la surface se fait par une méthode d'échantillonnage (cf I). On obtient un ensemble de carreaux "quasi-plan" si l'approximation n'est pas trop grossière puis on triangularise chacun des carreaux pour obtenir des éléments nécessairement plans. On calcule facilement l'intersection entre la droite et le plan formé par les sommets du triangle. Si l'on détecte une intersection , il faut vérifier si le point P obtenu appartient ou non au triangle. Pour déterminer cette appartenance , il existe 3 techniques classiques et relativement faciles à mettre en oeuvre [GAR 88].

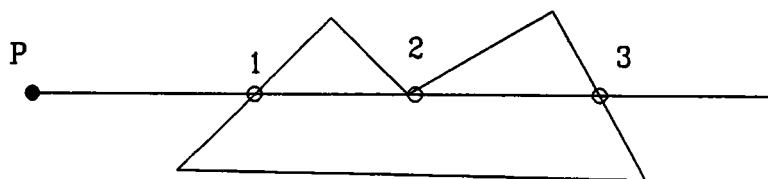
- Technique de la demi-droite : On trace une droite quelconque à partir du point P. Le nombre d'intersection avec le contour détermine l'appartenance ou non du point au contour.

nombre d'intersections pair -----> P n'appartient au contour

nombre d'intersections impair -----> P appartient au contour

La mise en oeuvre de cet algorithme présente tout de même quelques difficultés. En effet, on parle de compter le nombre des intersections, il faut uniquement tenir des intersections utiles.

#### Exemple



D'après la règle citée précédemment, le point P doit appartenir au contour ce qui manifestement faux, l'intersection 2 doit compter soit 0 fois soit 2 fois.

- Technique des angles capables: On calcule pour tous les points du polygone les angles  $(P_i, P, P_{i+1})$ . On calcule la somme des angles

Si la somme est égal à 0 —————> P est à l'extérieur du polygone

Si la somme est égal  $2n\pi$  —————> P est à l'intérieur

L'intérêt de cette méthode réside dans sa simplicité et dans le fait que le test sur la somme n'implique pas une grande précision (le résultat est, soit proche de 0 soit proche de  $2n\pi$ ). Son inconvénient majeur est l'utilisation d'une fonction trigonométrique.

- Technique appartenance d'un point à un convexe (dans ce cas c'est la plus intéressante)

On teste pour chacune des droites définies par deux points consécutifs du convexe si P est du même coté qu'un autre sommet quelconque du convexe par rapport à la droite. Ce test est très rapide dans le cas d'un triangle.

La difficulté de mise en oeuvre de cette méthode provient essentiellement du choix du facteur de triangularisation. En effet, il faut déterminer le nombre de points nécessaires pour échantillonner la surface pour que l'approximation ne soit pas trop grossière. Ce nombre va dépendre et de la position de la surface par rapport à l'observateur (ce nombre va diminuer si l'observateur n'est pas en face de la surface) et également de la forme de la surface (l'échantillonnage doit être important si la surface est très "courbe"). Il faut donc adapter l'échantillonnage aux propriétés locales de la surface. Nous avons choisi d'utiliser pour résoudre le problème d'intersection Droite/Surface directement les équations des surfaces: d'une part pour se servir pleinement de la définition mathématiques des faces des solides et d'autre part parce que cette méthode est totalement indépendante du point de vue et des propriétés locales des surfaces.

### III . 6 . 2 . 2 Méthodes fondées sur l'équation de la surface

Nous avons choisi d'utiliser pour résoudre le problème d'intersection Droite/Surface directement les équations des surfaces. D'une part pour se servir pleinement de la définition mathématiques des faces des solides et d'autre part parce que cette méthode est totalement indépendante du point de vue et des propriétés locales des surfaces.

### 1ère Méthode [SWE 86]

Soit  $S$  la surface dont on veut connaître les intersections avec la droite  $D$ .

Soient  $\text{Plan1}$  et  $\text{Plan2}$  deux plans tels que  $\text{Plan1} \cap \text{Plan2} = D$

$$A_1 x + B_1 y + C_1 z + D_1 = 0$$

(E)

$$A_2 x + B_2 y + C_2 z + D_2 = 0$$

Les points d'intersection entre la surface  $S$  et la droite  $D$  sont déterminés en injectant l'équation  $S(u,v)=(x(u,v);y(u,v);z(u,v))$  dans le système (E).

Pour résoudre ce système de 2 équations non linéaire à 2 inconnues, on utilise la Méthode de Newton. Le gros inconvénient de cette méthode réside dans le choix des valeurs de l'itération.

### 2ème Méthode [KAJ 82] :

C'est une méthode qui utilise la technique des Résolvantes. Elle est fondée sur un algorithme décrit par Kayjia .

*Soit  $S$  une surface NURBS*

*[P] son réseau de points de contrôle*

*$T_u, T_v$  : ses tables des noeuds en  $u$  et  $v$  respectivement*

*$k, l$  ses ordres en  $u$  et respectivement.*

*Et soit  $D$  une droite de l'espace*

Pour résoudre le problème, nous considérons la définition de la surface  $S$  dans la base canonique des polynômes (cf III.5.2) et rechercher les intersections de la droite avec les différents carreaux.

$$S = \bigcup_m C_m$$

$$S \cap D = \bigcup_m (C_m \cap D)$$

Quel que soit  $m$ ,  $C_m = \sum_{ij} A_{ij} u^i v^j$  (les points  $A$  sont des points de l'espace homogène)

quel que soit  $D$ , il existe  $PL1$  et  $PL2$  2 plans de l'espace tels que

$$D = PL1 \cap PL2$$

$$D = \begin{cases} A_1 X + B_1 Y + C_1 Z + D_1 H = 0 \\ A_2 X + B_2 Y + C_2 Z + D_2 H = 0 \end{cases}$$

Nous considérons les équations des plans dans l'espace homogène.

Le système à résoudre est le suivant :

$$(E) \quad \begin{cases} \sum \sum a_{ij} u^i v^j = 0 \\ \sum \sum b_{ij} u^i v^j = 0 \end{cases}$$

$$\text{avec } a_{ij} = A_1 X A_{i,j} + B_1 Y A_{i,j} + C_1 Z A_{i,j} + D_1 H A_{i,j}$$

$$b_{ij} = A_2 X A_{i,j} + B_2 Y A_{i,j} + C_2 Z A_{i,j} + D_2 H A_{i,j}$$

On peut considérer les 2 équations bivariées du système (E) comme des équations monovariées en  $u$  (par exemple) et dont les coefficients dépendent d'un paramètre  $v$ .

$$(E') \quad \begin{cases} \sum \sum a_{ij}(v) u^i = 0 \\ \sum \sum b_{ij}(v) u^i = 0 \end{cases}$$

Grâce à cette approche, le problème se réduit à la recherche des solutions communes de deux équations polynomiales à une variable. Pour ce faire, nous allons utiliser la méthode des résultantes. On trouvera en annexe la méthode détaillée pour calculer la résultante de ce système. Nous allons écrire le déterminant de Bezout de la Résolvante (E') et calculer ses racines. Nous pouvons résumer la procédure d'intersection comme suit :

**ETAPE 1** Calcul du déterminant de Bezout  $R(v)$  de la résultante de  $(E')$  que l'on vient de décrire

**ETAPE 2** Résolution de  $R(v)$

**ETAPE 3** Détermination des couples  $(u_i, v_j)$  pour toutes les solutions en  $v$  trouvées.

**ETAPE 2 : Résolution de  $R(v)$**

Le degré de l'équation dépend de l'ordre de la surface dans les 2 directions paramétriques, il est égal à  $2x(k-1)x(l-1)$ . Nous nous intéressons principalement à des NURBS de degré  $3 \times 3$  (ordre  $4 \times 4$ ) ou de degré  $2 \times 2$  (ordre  $3 \times 3$ ). Ce qui nous amène à déterminer les racines d'un polynôme de degré au plus égal à 18.

Il existe de nombreuses méthodes pour rechercher les racines d'un polynôme (Newton, Bairstow, Laguerre)[RAL 65]. Nous avons choisi la méthode de Bairstow car des trois c'est la plus robuste et la plus rapide [MICAD 88].

Cette méthode consiste à rechercher des polynômes de degré 2 qui divisent le polynôme  $R$  dont on cherche les racines. On divise  $R$  par le diviseur trouvé et on réitère le procédé jusqu'à ce que le degré de  $R$  soit inférieur ou égal à 2. Cette méthode permet de déterminer toutes les racines de  $R$ . Le détail de ces calculs est donné en annexe.

**ETAPE 3** Détermination des couples  $(u, v)$  qui déterminent les points d'intersection entre la surface et la droite.

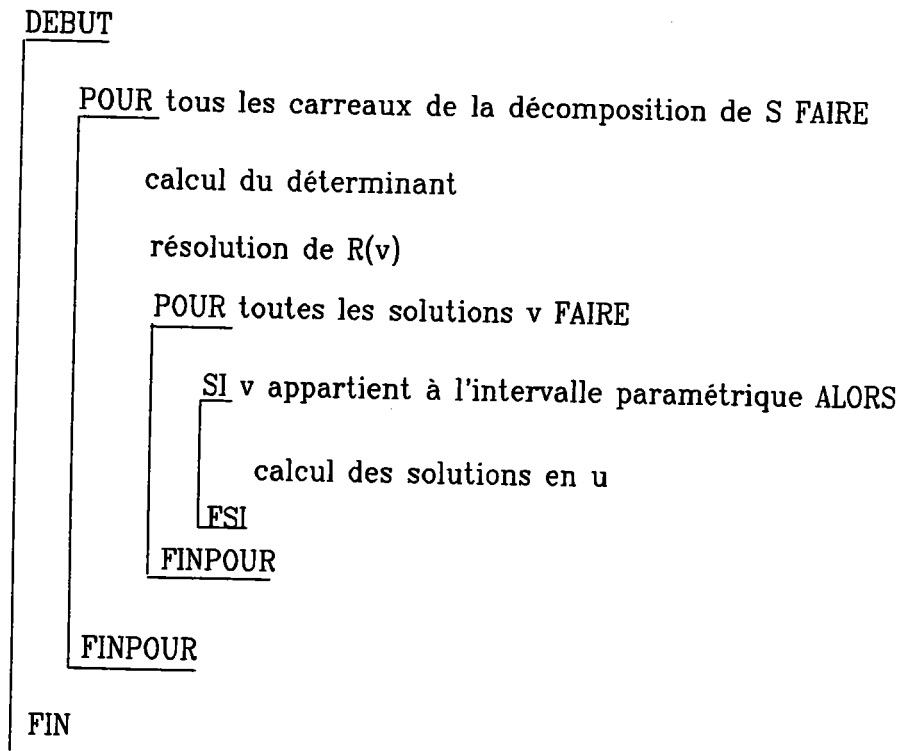
On introduit dans le système  $(E')$  les différentes valeurs  $v$  déterminées puis on résout ce nouveau système qui admet  $u$  pour inconnue.

### ALGORITHME Intersection Droite / Surface

Paramètres d'entrée : A,B,C,D,Ap,Bp,Cp,Dp : coefficients des plans qui définissent la droite

: Définition de la surface

Paramètres de sortie : liste des points d'intersection



### III . 6. 3 Intersection Rayon / Solide de base

#### III . 6 . 3 .1 Principe de l'algorithme

Pour déterminer les différentes portions de la droite qui sont HORS ou DANS dans le solide , il suffit d'appliquer à chacune des faces du solide l'algorithme précédent.

### ALGORITHME Rayon / Solide

Paramètres d'entrée : un solide

: un rayon

Paramètres de sortie : une liste de segments

DEBUT

POUR toutes les faces du solide FAIRE

intersection Rayon/Surface

FINPOUR

création de la liste de segments

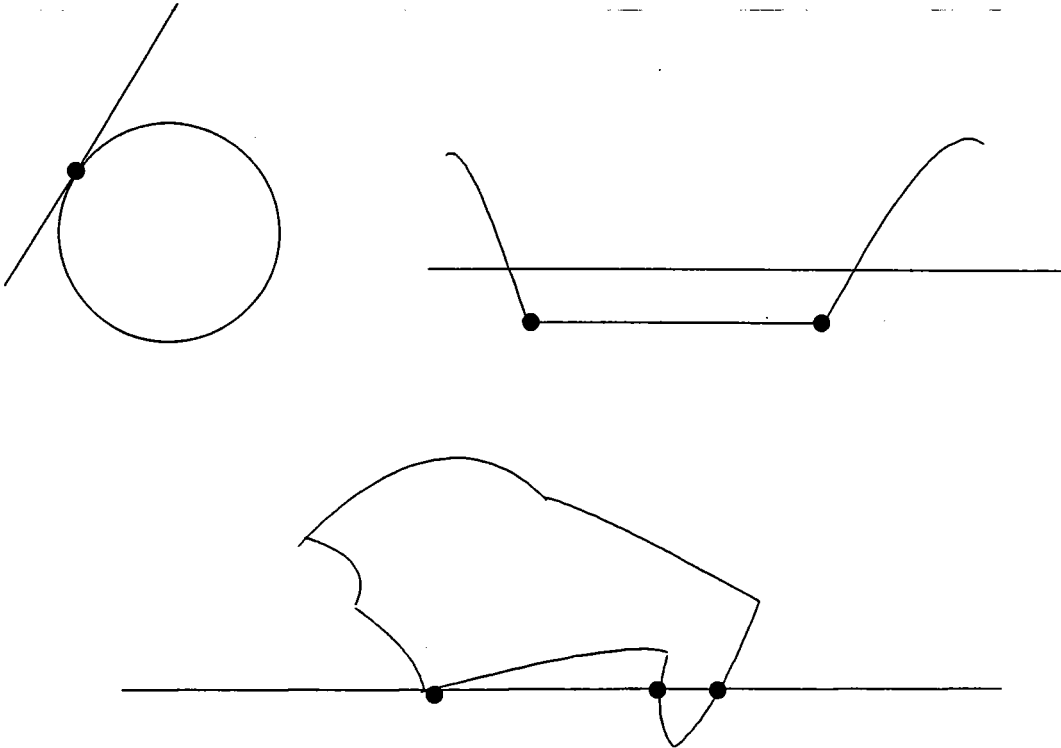
FIN

#### III . 6 . 3 . 2 *Création de la liste de segments*

A l'aide de l'algorithme précédent, nous avons déterminé tous les points d'intersection entre un solide et un rayon. Mais il est clair que certaines intersections ne sont pas à considérer car elles ne correspondent pas à une entrée dans la matière ou à une sortie. Il faut tenir compte des portions du rayon qui se trouve SUR le solide.



## EXEMPLES :



Avant de pouvoir créer une partition sur le rayon, il faut déterminer les conditions pour lesquelles une intersection est significative.

### Critère de définition d'une intersection significative :

Pour un point  $P$  d'intersection donné  $P=S(u,v)$ , on calcule la normale  $N$  à la surface en ce point ( $N=$  produit vectoriel(  $dS/du$ ,  $dS/dv$ ).

Si le rayon passant par  $P$  n'appartient pas au plan tangent(le produit scalaire du rayon avec  $N$  différent de 0) alors on considère qu'il y a traversée et l'intersection est significative.

Il est plus délicat de décider si l'intersection est significative quand la droite appartient au plan tangent. Ce problème s'apparente au problème de classification d'un point par rapport à un contour. Il existe deux méthodes classiques pour résoudre ce type de problème :

- Sur un voisinage d'un point, la surface se comporte comme son plan tangent en ce point . L'intersection sera considérée comme significative si la surface traverse le plan tangent au voisinage de l'intersection. Les surfaces que nous utilisons sont définies par produit tensoriel. Ce mode de définition privilégie deux directions : les isoparamétriques. Pour déterminer si la surface  $S$  traverse le plan tangent, nous vérifierons si l'une ou l'autre des courbes isoparamétriques  $C_1=S(u_i,v)$  ou  $C_2=S(u,v_i)$  coupe le plan tangent. Pour ce faire, considérons les points  $P_1 = S(u_i-du,v_i)$ ,  $P_2=S(u_i+du,v_i)$ ,  $P_3=S(u_i,v_i-dv)$  et  $P_4=S(u_i,v_i+dv)$ .  $P_1$  et  $P_2$  appartiennent à  $C_1$ ,  $P_3$  et  $P_4$  appartiennent à  $C_2$ . Si les couples  $(P_1,P_2)$  et  $(P_3,P_4)$  n'appartiennent plus au plan tangent on peut conclure sinon, le voisinage que nous avons considéré est trop petit pour être significatif et il faut l'élargir pour pouvoir conclure.

Description de l'algorithme : Classification d'une intersection par rapport à une surface

Paramètres d'entrée : S : une surface  
: PL : un plan  
:  $u_i, v_i$  : coord paramétriques de l'intersection

Paramètres de sortie : valide : Vrai si significative  
Faux sinon

DEBUT

PU1=S( $u_i - d_u, v_j$ ) PU2=S( $u_i + d_u, v_j$ )

PV1=S( $u_i, v_j - d_v$ ) PV2=S( $u_i, v_j + d_v$ )

i=1

TANT QUE (PU1 et PU2 dans PL) et (PV1 et PV2 dans PL) FAIRE

i=i+1

PU1=S( $u_i - i d_u, v_j$ ) PU2=S( $u_i + i d_u, v_j$ )

PV1=S( $u_i, v_j - i d_v$ ) PV2=S( $u_i, v_j + i d_v$ )

FINTANTQUE

TANT QUE (PU1 ET PU2 dans PL) FAIRE

i=i+1

PU1=S( $u_i - i d_u, v_j$ ) PU2=S( $u_i + i d_u, v_j$ )

FINTANTQUE

TANT QUE (PV1 et PV2 dans PL) FAIRE

i=i+1

PV1=S( $u_i, v_j - i d_v$ ) PV2=S( $u_i, v_j + i d_v$ ) FAIRE

FINTANTQUE

SI (PU1 et PU2 du même coté de PL) et (PV1 et PV2 du même cote de PL) ALORS

valide = FAUX

SINON

valide=VRAI

FINSI

FIN

*Remarque* : La liste des segments est créée à partir de tous les points d'intersection trouvés lors de Intersection Rayon /Surface pour toutes les faces du solide. La liste de points est triée par rapport à sa distance à l'oeil. Le plus le plus proche est considéré comme une entrée dans le solide si l'intersection est significative sinon le point est rejeté. Le point suivant de la liste est un point de sortie si , si l'intersection est significative. On réitère le processus jusqu'à ce que la liste des points intersection soit vide.

C'est une méthode excellente car elle se sert des caractéristiques locales de la surface au voisinage de l'intersection. Son inconvénient majeur réside dans le choix du "du " et du "dv". En effet, le choix du voisinage est crucial, trop grand il n'est pas significatif, trop petit il oblige à réitérer le processus.

- L'autre méthode pour éviter de décider si une intersection est significative ou non est de déplacer légèrement la droite pour éviter de traiter le problème. C'est l'équivalent de la méthode de décalage de la droite dans la classification d'un point par rapport à un contour. Cette méthode oblige à refaire tous les calculs d'intersection. C'est une méthode très brutale.

### III. 6 . 4 Intersection Rayon / Solide\_Composite

Par la méthode précédente, pour chacun des solides de base qui compose le solide , nous obtenons un ensemble de segments qui définissent les ENTREES et LES SORTIES du rayon. Pour déterminer la partie de la droite dans le solide résultat, il faut combiner les différents segments en tenant compte des opérations booléennes mises en jeu.

Par définition , un point appartient à l'union de 2 objets s'il appartient soit à l'un soit à l'autre. En considérant les opérations booléennes régularisées, un point appartient à l'union de 2 objets s'il appartient à l'adhérence de l'intérieur de  $(O1 \cup O2)$

Un point appartient à l'intersection de deux objets s'il appartient aux deux en même temps. En considérant les opérations booléennes régularisées, un point appartient à l'intersection de deux objets s'il appartient à l'adhérence de l'intérieur de  $(O1 \cap O2)$

Un point appartient à la différence de deux objets s'il appartient au premier sans appartenir au 2ème. En considérant les opérations booléennes régularisées, un point appartient à la différence de deux objets s'il appartient à l'adhérence de l'intérieur de  $(O1 - O2)$

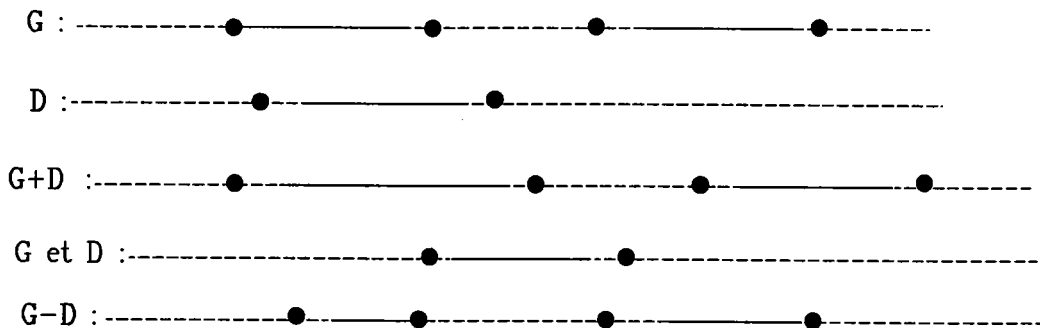
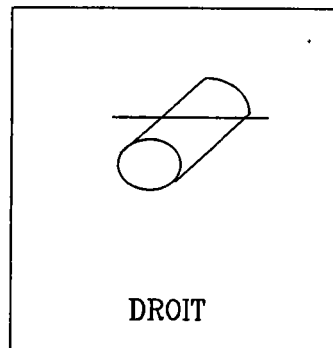
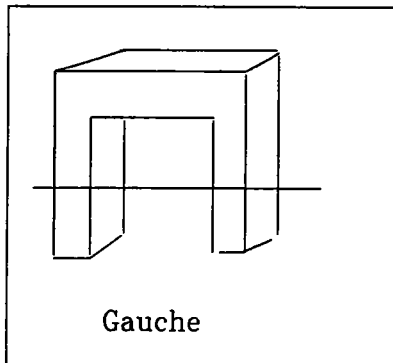
Nous pouvons résumer dans la table suivante les différentes règles qui régissent la combinaison des solides par rapport aux opérations booléennes.

Table de composition des opérations booléennes

Opérateur	Gauche	Droit	Composé
UNION	DANS	DANS	DANS
	DANS	DANS	DANS
	DANS	DEHORS	DANS
	DEHORS	DANS	DANS
	DEHORS	DEHORS	DEHORS
INTERSECTION	DANS	DANS	DANS
	DANS	DEHORS	DEHORS
	DEHORS	DANS	DEHORS
	DEHORS	DEHORS	DEHORS
DIFFERENCE	DANS	DANS	DEHORS
	DANS	DEHORS	DANS
	DEHORS	DANS	DEHORS
	DEHORS	DEHORS	DEHORS

**EXEMPLE :**

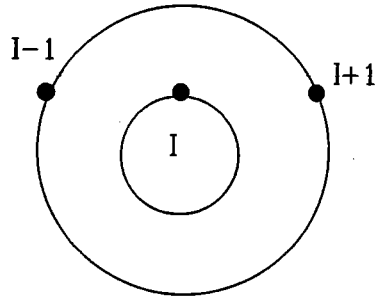
Considérons un solide composé des éléments suivants :



L'intersection du 1er solide avec le rayon définit une partition du rayon de même pour le 2ème solide

Les opérations booléennes mises en oeuvre sont des opérations booléennes régularisées. En effet, pour éviter que le solide construit par opération booléenne n'ait un intérieur vide ou une face ou une arête pendante, nous ne considérons dans la liste des intersections que les points qui traduisent un changement d'état.

Par exemple, avant l'intersection I, le rayon était considéré comme étant dans le solide, l'intersection I ne présente pas de changement pour le rayon donc on ne la retiendra pas. Par contre, l'intersection (I+1) sera conservée et indiquera la sortie du solide pour le rayon R.



Pour qu'il y ait changement d'état, il "faut" que le rayon traverse la surface. Localement, dans un voisinage du point d'intersection, la surface peut être assimilée à son plan tangent, donc il suffit de vérifier que le rayon traverse le plan Tangent.

La mise en oeuvre du Ray-Casting est maintenant très simple. Il suffit de parcourir l'arbre de construction et à chaque noeud

Si le noeud est terminal ( une feuille) lancer la procédure Rayon / Solide

Si le noeud est une opération booléenne combiner les segments obtenus par le fils gauche et le fils droit.

#### ALGORITHME RAY - CASTING

Paramètres d'entrée : un arbre CSG

: un rayon

Paramètres de sortie : partition du rayon en segments

DEBUT

SI le noeud est une primitive ALORS

intersection Rayon/Primitive

SINON

Combine(RayCasting(fils gauche),RayCasting(fils droit))

FINSI

FIN

### III . 7 . Mise en oeuvre

la maquette permet actuellement :

- La création de l'arbre CSG
- La définition des paramètres de visualisation
- L'évaluation de l'arbre
- L' affichage du résultat d'une évaluation:  
nous avons implanté cet algorithme pour tester la validité de l'approche.

Nous allons décomposer et détailler ces différentes actions.

#### III . 7 . 1 Création de l'arbre CSG

La création se fait de manière interactive. L'utilisateur définit les paramètres caractéristiques des différentes primitives qu'il veut utiliser et indique les opérations booléennes qu'il veut faire intervenir.

Pour chaque solide de base, le logiciel crée et stocke les différentes surfaces NURBS nécessaires à la définition du solide et retourne le numéro du solide qui apparaîtra dans l'arbre CSG.

A la fin de cette étape, les différents solides de base seront dans le modèle décrit dans III. 5 et l'arbre CSG est entièrement construit.

Au cours de cette étape, nous avons aussi calculé le volume englobant de chacun des solides de base.

##### III . 7 . 1 . 1 *Définition du volume englobant d'une face*

Rappelons qu' un solide est défini comme un ensemble de (1,2 ou 3) faces. Les faces sont définis par des NURBS.

Une surface NURBS est toujours contenue dans l'enveloppe convexe de son polyèdre de contrôle. Nous définissons son volume englobant par un parallélépipède rectangle dont les faces sont parallèles aux axes du système de référence.



On cherche les maxima et les minima en des coordonnées des points de contrôle et le volume englobant est défini par 2 points :  $P1 = (Xmin, Ymin, Zmin)$  et  $P2 = (Xmax, Ymax, Zmax)$ . Ce parallélépipède est convexe et donc contient l'enveloppe convexe du réseau de contrôle : C'est bien un englobant de la surface. Nous avons choisi cet englobant pour sa facilité de définition et le peu de calcul nécessaire à vérifier qu'un rayon le coupe ou non.

### III . 7 . 1 . 2 Définition du volume englobant d'un solide de base

Pour chacune des faces du solide, nous définissons un telle boîte englobante; le volume englobant du solide sera alors la boîte englobante contenant chacune de ses boîtes.

### III . 7 . 2 Définition des paramètres de la vision [PUC 87]

Les définitions et les primitives du logiciel graphique de base que nous utilisons sont celles décrites dans [FOL 82].

Pour une projection, il faut spécifier :

\*le plan de projection :

par un point VRP et une normale VPN

\*Une fenêtre de projection dans le plan de projection:

L'orientation de la fenêtre est donnée par un vecteur VUP qui indique la direction de l'axe vertical. La fenêtre est définie par son coin BAS Gauche et son coin HAUT Droit.

\* 2 plans de clipping : Plan de clipping avant et arrière:

Ceux ci sont parallèles au plan de projection; leur position est donnée sur l'axe qui a VRP pour origine et VNP comme vecteur directeur.

\* Un point de vue (l'oeil) :

sa position est donnée relativement à VRP

\* Une clôture écran :

il s'agit de coordonnées écran , la clôture est donnée par le point bas gauche et le point haut droit. On associe également, aux coordonnées écran, une troisième coordonnée Z. Cette coordonnée dépend de la profondeur du point dans la fenêtre. En effet, le plan de clipping avant et arrière définissent le maximum et le minimum en Z du cône de vision. la coordonnée en Z de la clôture peut être affectée à la couleur du pixel. Donc pour définir totalement une clôture, on donnera également un Zmin et un Zmax.

A partir de ces données, on initialise les différents paramètres de la projection. On conserve la transformation perspective et son application inverse.

### III . 7 . 3 Evaluation de l'arbre

La procédure d'évaluation de l'arbre se déroule comme suit :

Elle admet comme paramètre d'entrée :

- \* un arbre CSG qui contient le solide à analyser
- \* un numéro de fenêtre et de clôture
- \* la position de l'observateur

Paramètres de sortie :

Affichage du solide  
soit au trait soit en vue réaliste

Pour tous les pixels de la clôture, on cherche leurs coordonnées réelles ( $x_r, y_r, z_r$ ) dans l'espace utilisateur. Elles sont obtenues grâce à un passage Clôture-Fenêtre mémorisé à l'initialisation des paramètres de vision.

On parcourt l'arbre récursivement si le noeud est une feuille( ie un solide de base), on applique la procédure Rayon\_Solide.

#### III . 7 . 3.1 Procédure Rayon\_Solide

Avant tout, on vérifie que la droite qui part de l'oeil vers le point ( $x_r, y_r, z_r$ ) coupe le volume englobant du solide.

SI il n'y a pas d'intersection le traitement s'arrete là

SINON POUR chaque face on détermine les points d'intersection avec le rayon.

### III . 7 . 3 . 2 Procédure Rayon\_Surface

Paramètres d'entrée :

un rayon

définition complète de la surface qui contient les 2 décompositions décrites en III.3.2

Paramètres de sortie :

la liste des points intersection

Si le rayon ne coupe pas le volume englobant B-spline de la surface le traitement s'arrete là

SINON POUR tous les carreaux de la décomposition :

SI le rayon ne coupe pas le volume englobant du carreau de Bézier le traitement s'arrete

SINON

on calcule les paramètres de deux plans qui définissent le rayon

calcule le déterminant de Bezout associé  $R(v)$ (cf III.3)

POUR déterminer les racines de  $R(v)$ ,on lance dans un premier temps une procédure qui recherche  $v_1$  et  $v_2$  tels que  $R(v_1) < 0$  et  $R(v_1)$  proche de 0 et  $R(v_2) > 0$  et  $R(v_2)$  proche de 0

Ceux sont ces valeurs qui vont servir de point de départ à la méthode de

Bairstow (cf annexe) qui détermine les racines de l'équation. On réitère le procédé jusqu'à ce que le degré de  $R(v)$  soit inférieur au égal à 2.

Pour toutes les racines de  $R(v)$

SI  $v$  n'appartient pas à l'intervalle paramétrique on abandonne la valeur

SINON par la méthode décrite en III . 3 on calcule les valeurs de  $u$  possibles

POUR toutes les valeurs de  $u$  dans l'intervalle paramétrique du carreau

on calcule la distance à l'oeil de  $C(u,v)$  ( $\lambda$ )

on calcule les coordonnées cartésienne  $(x,y,z)$

on ajoute un nouveau point à liste de point courante.

Après avoir déterminé tous les points d'intersection,on trie selon la profondeur et on élimine les points doubles (si le point appartient à la frontière entre 2 carreaux).

### III . 7 . 3 . 3 Tableau de Resultats d'intersection

Nous avons choisi pour tester notre méthode pour déterminer les intersections entre une droite et une NURBS, d'utiliser une sphère. Avec une sphère, nous connaissons les points d'intersection théorique. On les calcule à partir de l'équation cartésienne de la sphère

$$(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 = R^2 \text{ et de l'équation paramétrique de la droite}$$

$$P_1P = A P_1P_2$$

Nous avons comparé nos résultats avec les points théoriques et également avec les résultats obtenus à partir d'une triangulation de la sphère.

**Sphère de centre (0,0,0) et de rayon 1**

**Solution théorique :**

$$x=0, y=0, z=1$$

DROITE	SPHERE coef de poi= 6	SPHERE coef de poi=12	SPHERE coef de poi= 32	SPHERE NURBS
P1   0 0 1	x1=0.00000000	x1=0.00000000	x1=0.00000000	x1=0.00000000
	y1=0.00000000	y1=0.00000000	y1=0.00000000	y1=0.00000000
	z1=1.00000000	z1=1.00000000	z1=1.00000000	z1=1.00000000
P2   10 0 1	E=0.00000000	E=0.00000000	E=0.00000000	E=0.00000000

**Sphère de centre (0,0,0) et de rayon 1**

**Solution théoriques**

$$x = \sqrt{3}/3, y = \sqrt{3}/3, z = \sqrt{3}/3$$

$$x = -\sqrt{3}/3, y = -\sqrt{3}/3, z = -\sqrt{3}/3$$

DROITE	SPHERE coef de pol= 6	SPHERE coef de pol= 12	SPHERE coef de pol= 32	SPHERE NURBS
P1   1 1 1	x1=0.52337290	x1=0.55437050	x1=0.57605200	x1=0.57736290
	y1=0.52337290	y1=0.55437050	y1=0.57605200	y1=0.57722500
	z1=0.52337290	z1=0.55437050	z1=0.57605200	z1=0.57736280
	E=0.093449155	E=0.03980207	E=0.00224841	E=0.00000006
P2   -1 -1 -1	x2=-0.52337290	x2=-0.55437050	x2=-0.57605210	x2=-0.57734640
	y2=-0.52337290	y2=-0.55437060	y2=-0.57605210	y2=-0.57735130
	z2=-0.52337290	z2=-0.55437060	z2=-0.57605200	z2=-0.57735650
	E=0.09349144	E=0.03980201	E=0.00224841	E=0.00000203

**Sphère de centre (0,0,0) et de rayon 1**

**Solutions théoriques :**

$$x = 1/2, y = 0, z = \sqrt{3}/2$$

$$x = -1, y = 0, z = 0$$

DROITE	SPHERE coef de pol= 6	SPHERE coef de pol= 12	SPHERE coef de pol= 32	SPHERE NURBS
P1   1/2 0 $\sqrt{3}/2$	x1=0.50000000	x1=0.50000000	x1=0.50000000	x1=0.4999896
	y1=-0.23205080	y1=0.00000000	y1=-0.00512251	y1=0.00000431
	z1=0.63397460	z1=0.86602540	z1=0.86090290	z1=0.86603140
	E=0.15989800	E=0.00000000	E=0.00441974	E=0.00000000
P2   1/2 3/2 $\sqrt{3}$	x2=0.50000000	x2=0.50000000	x2=-0.50000000	x2=0.50000001
	y2=-0.63397460	y2=-0.86602530	y2=-0.86056580	y2=-0.86602590
	z2=-0.23205080	z2=-0.00000005	z2=0.00545959	z2=-0.0001072
	E=0.15989800	E=0.00000006	E=0.00470948	E=0.00000048

**Sphère de centre (0,0,0) et de rayon 1**

**Solutions théoriques :**

$$x=-1, y=0, z=0$$

$$x=1/2, y=0, z=\sqrt{3}/3$$

DROITE	SPHERE coef de pol= 6	SPHERE coef de pol=12	SPHERE coef de pol= 32	SPHERE NURBS
P1 $\left  \begin{array}{l} 1/2 \\ 0 \\ \sqrt{3}/2 \end{array} \right.$	x1=0.36602540	x1=0.50000000	x1=0.49563530	x1=0.49999680
	y1=0.00000000	y1=0.00000000	y1=0.00000000	y1=0.00000262
	z1=0.78867510	z1=0.8660254	z1=0.86350540	z1=0.86602720
	E=0.13052710	E=0.00000000	E=0.00436151	E=0.00000006
P2 $\left  \begin{array}{l} \sqrt{3}+1/2 \\ 0 \\ \sqrt{3}+1/2 \end{array} \right.$	x2=-0.86602540	x2=-0.99683790	x2=-0.99999900	x2=-0.99999990
	y2=0.00000000	y2=0.00000000	y2=0.00000000	y2=0.00000660
	z2=0.7735022	z2=0.01180080	z2=0.00000004	z2=-0.00000923
	E=0.13052710	E=0.00309823	E=0.00000006	E=0.00000012

**Sphère de centre (0,0,0) et de rayon 10.**

**Solution théorique :**

$$x=0, y=0, z=10$$

DROITE	SPHERE coef de pol= 6	SPHERE coef de pol= 12	SPHERE coef de pol= 32	SPHERE NURBS
P1 $\left  \begin{array}{l} 0 \\ 0 \\ 1 \end{array} \right.$			x1=-0.000000119	x1=0.00000000
			y1=0.00000000	y1=0.00000000
			z1=10.00000000	z1=10.00000000
P2 $\left  \begin{array}{l} 10 \\ 0 \\ 10 \end{array} \right.$			E=0.00000000	E=0.00000000

**Sphère de centre (0,0,0) et de rayon 10**

**Solutions théoriques :**

$$x=10.\sqrt{3}/3, y=10.\sqrt{3}/3, z=10.\sqrt{3}/3$$

$$x=-10.\sqrt{3}/3, y=-10.\sqrt{3}/3, z=-10.\sqrt{3}/3$$

DROITE	SPHERE coef de poi= 6	SPHERE coef de poi=12	SPHERE coef de poi= 32	SPHERE NURBS
P1   1 1 1	x1=5.23372800	x1=5.543705000	x1=5.76052000	x1=5.77362900
	y1=5.23372800	y1=5.54370500	y1=5.76052000	y1=5.577325000
	z1=5.23372800	z1=5.54370500	z1=5.76052000	z1=5.577362900
	E=0.93491550	E=0.39801980	E=0.02248573	E=0.00000000
P2   -1 -1 -1	x2=-5.23372900	x2=-5.54370600		x2=-5.577344800
	y2=-5.23372900	y2=-5.54370600		y2=-5.577339500
	z2=-5.23372900	z2=-5.54370600		z2=-5.57735900
	E=0.93491550	E=0.39801980		E=0.00001526

**Sphère de centre (0,0,0) et de rayon 10**

**Solutions théoriques :**

$$x=5, y=-5.\sqrt{3}, z=0$$

$$x=5, y=0, z=5.\sqrt{3}$$

DROITE	SPHERE coef de poi= 6	SPHERE coef de poi=12	SPHERE coef de poi= 32	SPHERE NURBS
P1   5 0 √3/2	x1=5.00000000	x1=5.00000000	x1=5.00000000	x1=4.99989700
	y1=-2.32050800	y1=0.000000000	y1=-0.05122493	y1=0.00004300
	z1=6.33974600	z1=8.66025400	z1=8.60902900	z1=8.66031500
	E=1.5989800	E=0.0000000	E=0.04419708	E=0.00000095
P2   5 3/2 √3 .10	x2=5.00000000	x2=5.00000000	x2=5.00000000	x2=5.00000001
	y2=-6.33974600	y2=-8.66025300	y2=-8.60565800	y2=-8.66026700
	z2=-2.32050800	z2=-0.00000005	z2=0.05459618	z2=-0.00001974
	E=1.59898000	E=0.00000006	E=0.04709435	E=0.00001144

**Sphère de centre (0,0,0) et de rayon 10**

**Solutions théoriques :**

$$x=5, y=0, z=5.\text{sqrt}(3)$$

$$x=-10, y=0, z=0$$

DROITE	SPHERE coef de pol= 6	SPHERE coef de pol=12	SPHERE coef de pol= 32	SPHERE NURBS
P1 $\left  \begin{array}{l} 5 \\ 0 \\ \sqrt{3}/2 \end{array} \right. 10$	x1=3.66025400	x1=5.00000000	x1=4.95635400	x1=4.99995700
	y1=0.00000000	y1=0.00000000	y1=0.00000000	y1=0.000009294
	z1=7.88675100	z1=8.66025400	z1=8.63505500	z1=8.66026800
	E=1.30527200	E=0.00000000	E=0.00436151	E=0.00000000
P2 $\left  \begin{array}{l} \sqrt{3}+1/2 \\ 0 \\ \sqrt{3}+1/2 \end{array} \right.$	x2=-8.66025400	x2=-9.99999900	x2=-9.99999900	x2=-1.00000000
	y2=0.00000000	y2=0.00000000	y2=0.00000000	y2=-0.00000061
	z2=7.7350220	z2=0.00000095	z2=0.00000095	z2=-0.00000923
	E=1.30527200	E=0.00000095	E=0.00000095	E=0.00000095

**Sphère modélisée par une surface NURBS**

On constate avec la méthode qui utilise la sphère comme une surface NURBS que l'erreur commise est faible et de plus l'erreur ne varie pas beaucoup quelque soit la position de la droite et quel que soit le rayon de la sphère.

**Sphère triangularisée**

Si la sphère est assimilée à un ensemble de triangles, L'erreur varie en fonction du rayon de la sphère et de la position de la droite. Evidemment, l'erreur dépend du coefficient de polygonalisation.

Tous les points de la sphère ont une courbure constante. Ceci facilite les problèmes liés à la triangularisation. En effet, pour une surface quelconque, les régions à forte courbure( qu'il faudrait détecter) nécessitent un facteur de triangularisation beaucoup plus important pour conserver la même précision.



### III . 7 . 3 . 4 *Création de la liste de segments*

**Paramètres d'entrée :**

la liste des points d'intersection entre un rayon et un solide de base

**Paramètres de sortie :**

la liste des segments

**Pour tous les points de la liste**

on calcule la normale du carreau en ce point

SI le rayon appartient au plan tangent de la surface

On vérifie si le point d'intersection est significatif ou non si il n'est pas significatif il est rejeté

SINON le point détermine l'origine ou l'extrémité d'un segment. (Si le précédent était origine (respectivement extrémité) le point courant est extrémité (respectivement origine ))

Cette liste de segments définit tout le solide de base le long du trajet du rayon.

### III . 7 . 5 *Définition complète du solide*

Pour chaque solide primitif, nous avons défini sa liste de segments caractéristique le long du trajet du rayon. Récursivement, nous combinons ces différents segments en respectant les opérations booléennes mises en jeu suivant les règles établies en III . 3

Actuellement, nous nous servons de ces résultats uniquement pour l'affichage du solide. Mais nous pourrions assez facilement étendre notre application aux calculs d'autres propriétés du solide (calculs de volume, calculs de centre d'inertie).

### III . 7 . 6 *Affichage du résultat d'une évaluation*

Actuellement deux modes d'affichage sont implantés :

Affichage réaliste :

Pour chaque pixel de la cloture, nous avons une partition du rayon, donc afficher un solide, il suffit de considérer l'origine du premier segment de la partition. C'est le point du solide le plus proche de l'observateur par rapport à ce rayon. L'intensité lumineuse du pixel est fonction de la distance à l'observateur et de l'angle de la normale avec le rayon.

### Affichage au trait :

Pour réaliser un affichage au trait avec élimination des parties cachées, il suffit de garder en mémoire le buffer de trame de la ligne précédente. Au lieu de mémoriser la couleur des pixels, on mémorise dans le buffer de trame le numéro de la face du solide. Ainsi, lorsque l'on a les buffers de deux lignes consécutives,  $bufy$  et  $bufy+1$  on fait :

Pour  $x$  de  $xmin$  à  $xmax-1$  Faire

Si  $(bufy[x] < > bufy+1[x])$  ou  $(bufy[x] < > bufy[x+1])$  Alors  
afficher le pixel  $(x,y)$

fsi

Fpour

### III . 8 Améliorations

Actuellement, la maquette, développée en MS.Pascal fonctionne sur PC. Les performances sont catastrophiques d'un point de vue d'une utilisation interactive du logiciel. Par exemple, la visualisation d'un solide prend environ une douzaine d'heures sur un PC AT avec coprocesseur de calcul. Pour améliorer les résultats, nous allons définir de nouveaux filtres sur les volumes englobant pour utiliser moins souvent l'algorithme d'intersection droite/ Surface. C'est cet algorithme qui est très coûteux en temps.

Il existe encore beaucoup de choses à améliorer dans le processus de calcul des racines du déterminant de la résolvante. La méthode que nous utilisons est la méthode de Bairstow. Dans cette méthode, les valeurs de départ de l'itération peuvent être quelconques, mais si elles sont "bien choisies", le nombre d'itérations nécessaires pour parvenir au polynôme diviseur décroît. Nos travaux futurs porteront sur la mise en oeuvre d'un algorithme qui choisira judicieusement les valeurs initiales des itérations. Pour l'affichage des solides, on peut limiter la précision des calculs et utiliser une méthode d'incrémentation des polynômes pour déterminer les racines qui nous intéressent [CAU 88].

Par définition, un Ray-Tracing est une méthode longue, son surnom de "Brutal Force Method" est justifié. Elle est longue et coûteuse mais parvient au résultat demandé. Nous essayons d'améliorer nos performances en utilisant des techniques de subdivision de l'espace qui ont pour but de minimiser le nombre d'intersection Rayon SOLIDE à calculer

[CLE 88].

Pour améliorer, les performances du Ray\_Tracing, on peut tenter d'utiliser la propriété de non unicité de l'arbre CSG. Définir un nouvel arbre équivalent qui minimiserait les calculs. Par exemple, dans un modèle B-REP, la détection d'objet nul (réduit à l'ensemble vide) est trivial tandis que dans un arbre de construction, cette détection n'est pas simple. Pourtant lancer un grand nombre de rayons sur un solide qui est globalement égal à l'ensemble vide est très coûteux. Tilove [TIL 84] propose une méthode pour détecter de tels objets et redéfinir un nouvel arbre qui n'en tient plus compte. La prise en compte de cet algorithme peut améliorer grandement les temps de calculs.

Un autre type d'amélioration est encore nécessaire, il s'agit de la gestion des "EPSILONS". Tous les calculs sont effectués sur des nombres réels et nous utilisons des méthodes de résolution qui sont itératives, leurs conditions d'arrêt portent sur la proximité de 2 valeurs. Une étude mathématique est nécessaire pour déterminer les relations qui existent entre les différents EPSILON.

CONCLUSION

Notre exposé voulait faire ressortir un certain nombre de problèmes liés à la représentation informatique des solides. Pour cela nous avons essayé de souligner la nécessité d'une approche formelle de leur modélisation. Nous espérons avoir montré la nécessité de définir des modèles mathématiques en préalable à la mise en oeuvre des modèles informatiques. Nous pensons avoir également souligné que le passage d'un modèle mathématique à un modèle informatique n'est pas toujours chose aisée.

Nous voulions également prouver qu'il était possible d'intégrer à un modèle solide des surfaces gauches sans passer par une "facettisation" de la surface celle-ci ne permettant pas de conserver les propriétés mathématiques de la surface. Comme pour la modélisation des solides, nous pensons que la modélisation des surfaces doit s'appuyer sur une représentation mathématique.

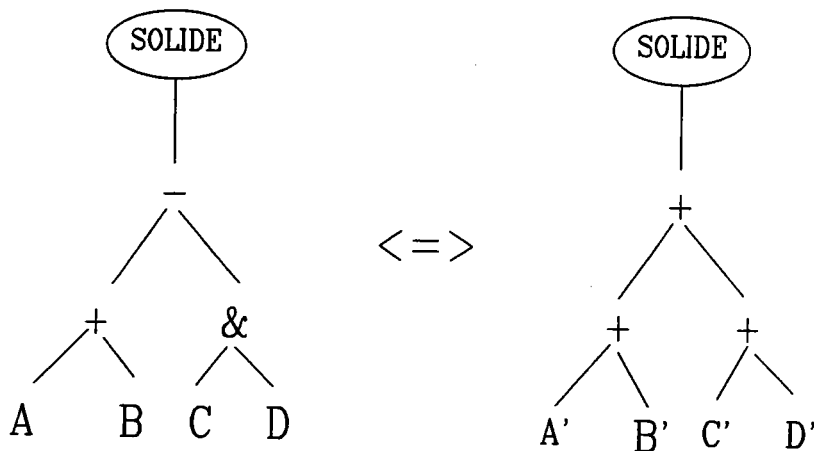
Les surfaces couramment employées en C.F.A.O sont des surfaces paramétriques. Elles sont particulièrement bien adaptées à l'interaction ce qui est un atout non négligeable car les bons systèmes de C.F.A.O se veulent interactifs.

Nous pensons avoir souligné les nombreux problèmes (algorithmiques ou liés à la validité des modèles) qu'occasionne la prise en compte des surfaces dans les modèles de solides. La solution que nous avons proposée, ne se veut en aucun cas parfaite, elle a néanmoins pour mérite de conserver intégralement toutes les propriétés mathématiques des surfaces qu'elle intègre au modèle solide. Les surfaces que nous utilisons sont des Nurbs, elles ne constituent pas une représentation idéale mais restent suffisamment générales pour contenir les autres types de surfaces couramment employés. Nous le considérons comme un modèle interne complètement transparent à l'utilisateur en particulier parce que la manipulation de la coordonnée homogène n'est pas facile pour une personne non avertie.

Les principes énoncés ci-dessus servent à définir les volumes primitifs d'un arbre CSG. Ce type de représentation repose sur des concepts mathématiques. Il permet la création d'une grande variété de solides, cette faculté étant encore accrue par l'utilisation de surfaces gauches pour la définition des volumes de base. On peut ainsi dire que l'on sait représenter des solides à formes libres.

Cette étude permet d'envisager dans de bonnes conditions les extensions suivantes:

- l'implantation d'un algorithme de passage du modèle CSG à un modèle par les limites (B-Rep). A cet effet, nous aimerions intégrer à l'actuelle maquette, les travaux sur les carreaux restreints développés dans notre laboratoire [ZID 90]; cette approche est un passage obligé si on veut conserver, lors des opérations booléennes, les définitions mathématiques des volumes primitifs.
- l'exploitation de la propriété de non unicité d'un arbre CSG: les performances de la maquette doivent être améliorées. Nous pensons que dans cet esprit l'approche par arbre équivalent est sans doute prometteuse. L'idée est de définir un arbre et un algorithme par opération. Par exemple, si on veut calculer le volume d'un solide défini par un arbre de construction, on peut retranscrire ce dernier sous la forme d'un arbre (équivalent) qui contient uniquement des opérations d'union (ce qui n'est pas toujours trivial). Le volume du solide est alors égal à la somme des volumes des solides primitifs.



- un autre avantage du schéma CSG est de bien se prêter par définition à l'utilisation des techniques de parallélisme.
- la modélisation par arbre CSG tend également à intégrer de nouvelles primitives comme les objets définis par des extrusions généralisées: ce sont par exemple les prismes à base quelconque, les objets de révolution ou ce qu'on convient d'appeler les cylindres généralisés, définis par le déplacement d'un contour plan selon une trajectoire quelconque.

- nous avons également l'intention de nous intéresser à d'autres formes de représentation de surfaces. En effet, comme nous l'avons montré, les surfaces paramétriques classiques ont certes beaucoup d'avantages, mais également des inconvénients qu'il ne convient pas de négliger. Les surfaces paramétriques n'offrent pas la possibilité de déterminer facilement des équations implicites simples et faciles à utiliser. De plus, on n'inscrit que très difficilement une courbe quelconque sur des surfaces définies par produit tensoriel. Pour ces raisons, la modélisation des surfaces par des formes algébriques devrait être étudiée. En effet, ce mode de représentation a l'avantage d'être fermé pour un certain nombre d'opérations à savoir: on peut définir l'intersection de deux surfaces algébriques sous la forme d'une équation algébrique, une surface algébrique sépare l'espace en deux demi-espaces. Il ne faudrait tout de même pas trop hâtivement conclure que ce mode de représentation deviendra un modèle idéal de surfaces car nous pensons qu'il n'a pas la propriété de s'adapter facilement à l'interactivité et à la programmation des commandes numériques.

ANNEXE I



## ANNEXE I

### Représentation exacte d'une sphère à l'aide des NURBS

Nous allons traiter le cas particulier de la sphère unité, c'est-à-dire la sphère de centre (0,0,0) et de rayon 1. Grâce aux propriétés d'invariance par transformation affine, la connaissance des points de contrôle définissant la sphère unité est suffisante pour une sphère quelconque.

- Représentation paramétrique de la sphère unité :

$$R(t, w) = \begin{cases} x(t, w) = \sin t \cos w \\ y(t, w) = \sin t \sin w \\ z(t, w) = \cos t \end{cases}$$

$$(t, w) \in ]0, 2\pi[ \times ]0, \pi[$$

Nous transformons la paramétrisation du 1er octant.  $0 \leq t \leq \pi/2$  et  $0 \leq w \leq \pi/2$

$$\text{Posons } \cos t = \frac{1 - u^2}{1 + u^2} \qquad \sin t = \frac{2u}{1 + u^2}$$

$$\cos w = \frac{1 - v^2}{1 + v^2} \qquad \sin w = \frac{2v}{1 + v^2}$$

La représentation paramétrique R s'écrit maintenant dans l'espace homogène :

$$R(u, v) = \begin{cases} 2v(1 - u^2) \\ 4uv \\ (1 - v^2)(1 + u^2) \\ (1 + u^2)(1 + v^2) \end{cases} \quad \begin{matrix} 0 \leq u \leq 1 \\ 0 \leq v \leq 1 \end{matrix}$$

Pour obtenir la forme :

$$S(u, v) = \sum_{i=0}^2 \sum_{j=0}^2 \beta_{ij} T(P_{ij}) N_i^3(u) N_j^3(v)$$

Il suffit d'effectuer le changement de base : { base canonique }  $\rightarrow$  { base des Bernstein }.  
Pour ce faire, il suffit d'égaliser les coefficients des équations de  $R(u,v)$  et  $S(u,v)$ .

On obtient alors neuf points de contrôle qui déterminent le premier octant de la sphère unité. Puis par symétrie, on détermine entièrement la surface B-spline rationnelle qui décrit exactement la sphère unité. Pour définir une sphère quelconque, on fait subir aux points de contrôle, la translation (pour définir un centre différent de  $(0,0,0)$ ) et une homothétie correspondant au rayon de la sphère.

On détermine facilement des équations paramétriques, comparables à celles de la sphère, pour les autres quadriques. Ensuite, en utilisant la méthode décrite précédemment, on détermine aisément les points de contrôle de la nurbs équivalente.

ANNEXE II

## ANNEXE II

### Résolvante de deux équations polynomiales sous la forme d'un déterminant de Bezout

Soient  $a(x)$ ,  $b(x)$  deux polynômes à coefficients réels :

$$a(x) = a_0 x^n + \dots + a_n$$

$$b(x) = b_0 x^m + \dots + b_m$$

On peut supposer, sans nuire à la généralité de l'exposé, que  $n \geq m$ . C'est-à-dire qu'il existe  $p$  tel que  $n = m + p$ .

Si  $a$  et  $b$  ont une racine commune  $r$ , on a  $a(r) = b(r) = 0$

Considérons le système d'équations suivants :

$$\left[ \begin{array}{l} b_0 a(r) - a_0 r^p b(r) = 0 \\ (b_0 + b_1 r) a(r) - (a_0 + a_1 r) r^p b(r) = 0 \\ \vdots \\ (b_0 r^{m-1} + b_1 r^{m-2} + \dots + b_m) a(r) - (a_0 r^{m-1} + \dots) r^p b(r) = 0 \\ r^{p-1} b(r) = 0 \\ \vdots \\ b(r) = 0 \end{array} \right.$$

Si on développe ces différentes équations, nous obtenons un système de  $n$  équations à  $n$  inconnues (à savoir  $1, r, r^2, \dots, r^{n-1}$ ). Pour que ce système admette des solutions, il faut que le déterminant du système soit nul. Le déterminant du système est exactement la résolvante de  $a$  et  $b$ .

ANNEXE III

### ANNEXE III

## Recherche des racines d'un polynôme par la méthode de Bairstow

Soit à chercher les racines du polynôme P de degré strictement supérieur à 2:

$$P_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_n \quad (\text{équation 1})$$

Etant donné le trinôme  $x^2 + px + q$  où p et q sont des constantes quelconques,  $P_n(x)$  peut s'écrire :

$$P_n(x) = (x^2 + px + q)(b_0x^{n-2} + b_1x^{n-3} + \dots + b_{n-3}x + b_{n-2}) + Rx + S \quad (\text{équation 2})$$

A chaque couple (p,q) correspond un ensemble de valeurs  $b_0, b_1, \dots, b_{n-2}, R$  et S. La méthode consiste à trouver p et q tels que  $R=0$  et  $S=0$ : l'équation (2) montre alors que les racines de  $x^2 + px + q$  sont également les racines de  $P_n(x)$ . Il reste alors à chercher les racines du polynôme de degré (n-2) :

$$P_{n-2}(x) = b_0x^{n-2} + b_1x^{n-3} + \dots + b_{n-3}x + b_{n-2}$$

On opère comme précédemment, d'où deux nouvelles racines (réelles distinctes ou confondues, ou complexes) et ainsi de suite jusqu'à obtention d'un polynôme de degré 1 ou 2.

La première étape consiste donc à déterminer  $b_0, \dots, b_{n-2}$  en fonction de p et q, ce qui se fait en développant l'équation 2 et en l'identifiant à l'équation 1. Auparavant, il est commode de poser :

$$R = b_{n-1} \text{ et } S = p b_{n-1} + b_n \quad (\text{équation 3})$$

D'où on obtient pour  $k$  variant de 0 à  $n$

$$\begin{aligned} b_k &= a_k - pb_{k-1} - qb_{k-2} \\ b_{-2} &= b_{-1} = 0 \end{aligned} \quad (\text{équation 4})$$

Les relations de récurrence (équation 4) permettent de déterminer  $R$  et  $S$  en fonction de  $p$  et  $q$ . Pour trouver  $p$  et  $q$  tels que  $R = 0$  et  $S = 0$ , on utilise la méthode de Newton Raphson. A  $p_0$  et  $q_0$  arbitraires correspondent  $R(p_0, q_0) = R_0$  et

$S(p_0, q_0) = 0$ . On cherche alors  $dp$  et  $dq$  tels que  $R(p_0 + dp, q_0 + dq) = 0$  et  $S(p_0 + dp, q_0 + dq) = 0$ .

La résolution de système nous amène à poser de nouvelles relations de récurrence.

$$\begin{aligned} c_k &= b_k - pc_{k-1} - qc_{k-2} \\ c_{-2} &= c_{-1} = 0 \end{aligned} \quad (\text{équation 5})$$

et on peut prouver que :

$$\begin{aligned} dp &= P/D \text{ et } dq = Q/D \\ D &= c_{n-2}^2 - c_{n-1}c_{n-3} + b_{n-1}c_{n-3} \\ P &= b_{n-1}c_{n-2} - b_n c_{n-3} \\ Q &= b_n c_{n-2} - b_{n-1}c_{n-3-1} + b_{n-1}^2 \end{aligned}$$

Pour résumer la procédure est la suivante :

- 1/ Partant de  $P_n(x)$ , on se donne  $p_0$  et  $q_0$  arbitraires
- 2/ On en déduit  $b_0, \dots, b_n$
- 3/ On calcule  $c_0, \dots, c_{n-1}$
- 4/ On pose  $p_1 = p_0 + dp$  et  $q_1 = q_0 + dq$ , et on recommence les opérations précédentes jusqu'à obtention d'un couple  $(p_j, q_j)$  sensiblement égal au couple  $(p_{j-1}, q_{j-1})$ .
- 5/ La résolution de  $x^2 + p_j x + q_j$  donne deux racines de  $P_n$
- 6/ On recommence le processus ci-dessus à partir du polynôme  $P_{n-2}$  dont les coefficients sont les  $b_0, \dots, b_{n-2}$

ANNEXE IV



## ANNEXE IV

### Génération des points de Bézier d'une surface B-Spline donnée

1. Génération des points de Bézier d'une courbe B-Spline. Soit

$$C(u) = \sum_{i=0}^m P_i N_i^k(u)$$

une courbe B-Spline d'ordre  $k$  et de table des noeuds  $tndu = \{u_i\}_{i=0..m+k}$

Il est facile de montrer que si deux noeuds consécutifs  $u_n$  et  $u_{n+1}$ , tels que  $u_n$  soit strictement inférieur à  $u_{n+1}$ , sont tous deux de multiplicité  $(k-1)$ , la seule fonction B-Spline ne s'annulant sur l'intervalle  $[u_n, u_{n+1}[$  est précisément la polynomiale de Bernstein définie sur  $[u_n, u_{n+1}]$ . Cette constatation permet de calculer les points de Bézier correspondant à une courbe B-Spline donnée, il suffit d'insérer chaque valeur nodale jusqu'à ce qu'elle soit de multiplicité  $(k-1)$ .

2. Insertion d'une valeur nodale

Soit  $u$  une valeur nodale associée à la courbe  $C$ . Il existe un unique intervalle paramétrique  $[u_{\text{delta}}, u_{\text{delta}+1}[$ , avec,  $u_{\text{delta}} < u_{\text{delta}+1}$  tel que  $u$  appartienne à cet intervalle. Les seuls points de contrôle qui influencent cet arc de courbe sont :

$$P_{\text{delta}-k+1}, P_{\text{delta}-k+2}, \dots, P_{\text{delta}}$$

On calcule une nouvelle série de points  $P_i^u$ , pour  $i$  variant de  $\text{delta}-k+1$  à  $\text{delta}$  grâce à la première étape de l'algorithme de Cox-DeBoor.

$$P_i^u = (1-a_i) P_{i-1} + a_i P_i$$

avec  $a_i = (u - tndu[i]) / (tndu[i-k+1] - tndu[i])$

La courbe C peut alors être définie par la table tndu à laquelle on a rajouté le noeud u et par les points :

$$P_0, \dots, P_{\delta-k}, P_{\delta-k+1}, P_{\delta-k+2}, \dots, P_{\delta}, P_{\delta+1}, \dots, P_m$$

### 3. Détermination des points de Bézier d'une surface B-Spline

Les surfaces B-Spline sont définies par produit tensoriel; pour  $u^*$  fixé,  $S(u^*,v)$  est une courbe B-Spline. Pour déterminer les points de Bézier d'une telle surface, il suffit d'augmenter la multiplicité de toutes les valeurs nodales de la table des noeuds selon u, respectivement selon v (voir algorithme ci-dessous).

### Algorithme Transformation B-Spline/Bézier

Paramètres d'entrée/sortie :     $T_p$             : matrice des points de contrôle  
  
    $T_{ndu}$         : table des noeuds selon u  
  
    $T_{ndv}$         : table des noeuds selon v  
  
    $ordreu$      : ordre de la surface selon u  
  
    $ordrev$      : ordre de la surface selon v  
  
    $nbptu$       : nombre de points selon u  
  
    $nbptv$       : nombre de points selon v

```
DEBUT
  i = ordreu
  TANTQUE i <= nbptu FAIRE
    TANTQUE multiplicité de  $T_{ndu}[i] < ordreu - 1$  FAIRE
      calculer les nouveaux points de la courbe
      insérer ces points dans  $T_p$ 
      insérer la nouvelle valeur dans  $T_{ndu}$ 
    FINTANTQUE
    i=i+1
  FINTANQUE

  j = ordrev
  TANTQUE j <= nbptv FAIRE
    TANTQUE multiplicité de  $T_{ndv}[j] < ordrev - 1$  FAIRE
      calculer les nouveaux points de la courbe
      insérer ces points dans  $T_p$ 
      insérer la nouvelle valeur dans  $T_{ndv}$ 
    FINTANTQUE
    j=j+1
  FINTANQUE
FIN
```

ANNEXE V

## ANNEXE V

### Exemples de réalisation

Les illustrations jointes sont des photographies prises directement sur un écran.

L'écran en question est un écran Secapa géré par un PC muni d'une carte Galaxy 16 couleurs. C'est ce qui justifie la faiblesse du dégradé. On aurait pu utiliser une technique de "Dithering" [PUC 89] pour améliorer le rendu, mais la qualité de l'image n'était pas notre objectif.

Une observation de ces images permet une discussion rapide des avantages et inconvénients de la méthode :

- Soumission aux imprécisions :

Les points parasites violemment illuminés( PlancheI.2) correspondent à une non détection de la première intersection avec l'objet : la première face coupée est donc celle du fond ce qui justifie l'affichage d'une arête réduite à un point. Ces points reflètent l'instabilité des epsilons dans le calcul des intersections. Ils sont rejetés car ils sont considérés comme étant "trop loin" de la surface. Cet inconvénient peut être éliminé par une gestion plus fine des epsilons.

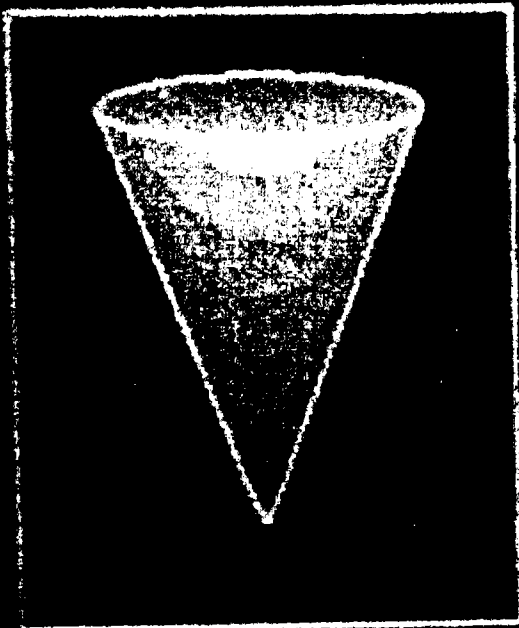
- Absence d'arêtes(PlancheII.2) :

Les facettes bordant le cube sont décrites par une même surface et donc les arêtes n'appartiennent pas à la silhouette ne sont pas mises en évidence.

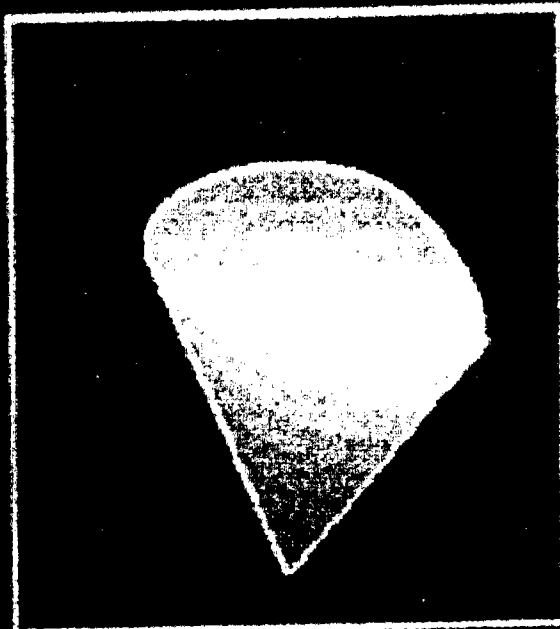
- L'aptitude de la méthode à la déformation est illustrée par les images plancheI.2 et plancheI.3. Sur la plancheI.3, on a translaté dans l'axe du cylindre, le centre de la première base et déplacé un point de la deuxième base perpendiculairement à cet axe.

- Les images plancheI.4 et plancheII montrent des exemples d'opérations booléennes.

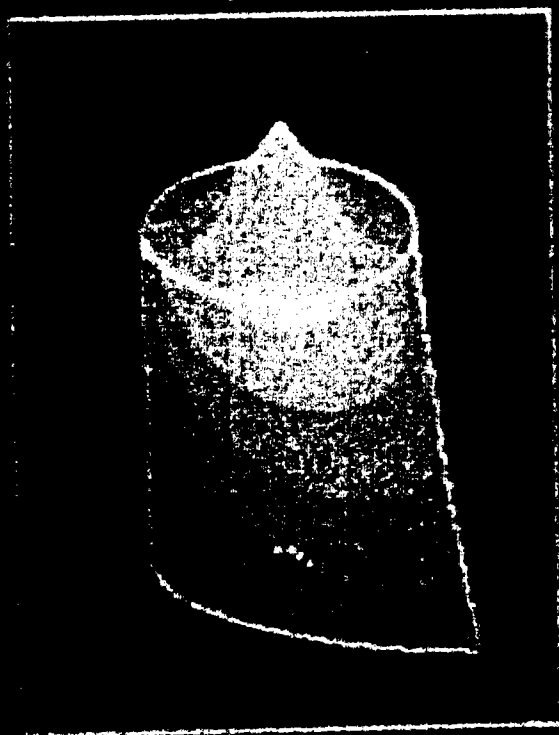
1 1



1 2



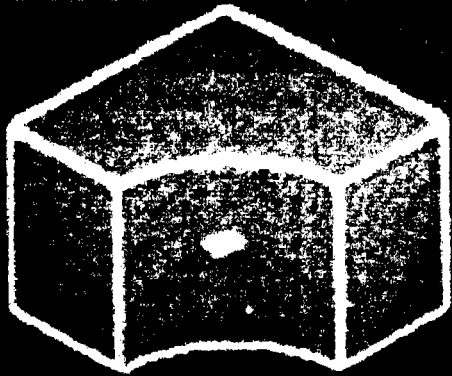
1 3



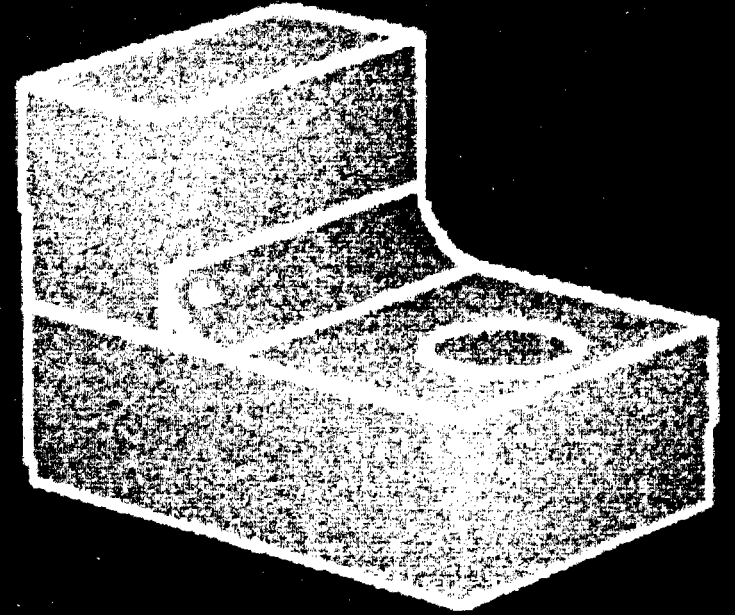
1 4



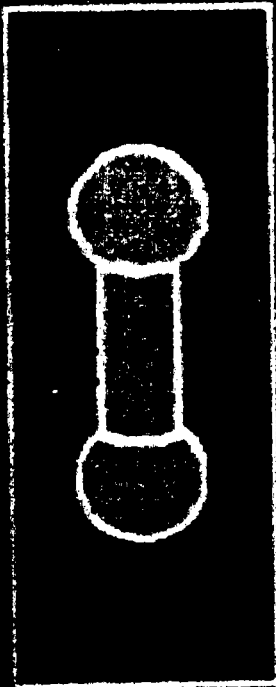
111



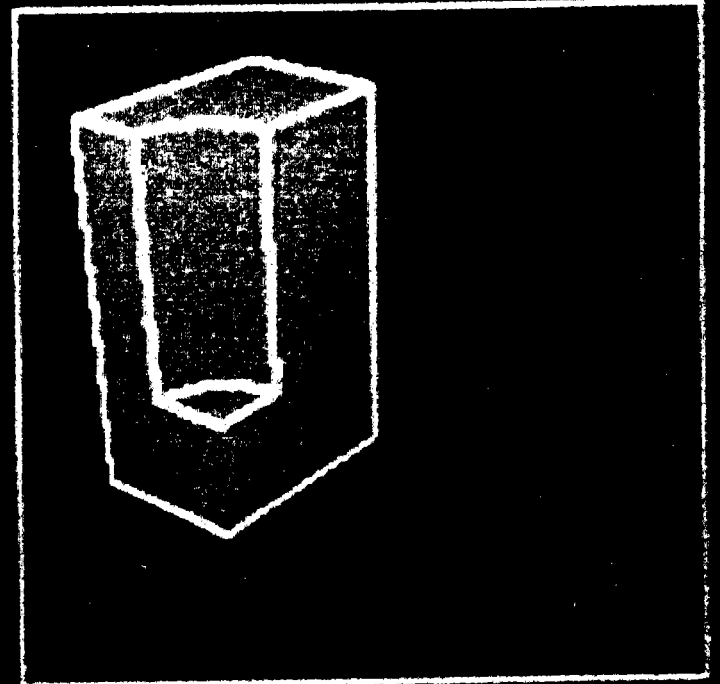
112



113



114



# BIBLIOGRAPHIE



## BIBLIOGRAPHIE

[AGO 76]: M.K AGOSTON

"Algebraic topology A first course "

Marcel Dekker , INC New-York and Basel

[AOM 90]: S. AOMURA, T. UEHARA

"Self-intersection of an offset surface"

Computer Aided Design volume 22 n 7 1990

[AYA 85]: D. AYALA - P. BRUNET- R.JUAN- I.NAVAZO

"Object representation by means of nonminimaldivision quadrees and octrees"

ACM transactions on graphics vol 4 numéro 1 1985

[BAU 74]: BAUMGART

"Geometric modelling for computer vision"

STAN -CS-74-463 Stansdford artificial Intelligence laboratory

[BAR 87]: Brian A .BARKSKY-Richard H.BARTELS-John C.BEATY

"B-SPLINES" Mathématiques et C.A.O volume 6

éditions HERMES PARIS 1987

[BEZ 86]: P. BEZIER

"Courbes et Surfaces" Mathématiques et C.A.O volume 4

éditions HERMES PARIS 1986

- [BOEH 84]: BOEHM-G.FARIN-J.KAHMANN  
"A survey of curve and surface methods in CAGD"  
Computer Aided Geometric Design vol 1 numéro 1 1984
- [BOO 78]: C . de BOOR  
"A practical guide to splines"  
éditions Springer Verlag New-York 1978
- [BOU 87]: K.BOUATOUCH, M.O.MADONI, T.PRIOL, B.ARNALDI  
"A new algorithm of space tracing using CSG model"  
EUROGRAPHICS 87
- [BOY 79]: J.W. BOYSE  
"Data structure for a solid modeller"  
Research Publication GMR-2933
- [BRA 75]: I.C. BRAID  
"The synthesis of solid bounded by many faces"  
Communication of the A.C.M avril 1975 vol 18 n 4
- [BRA 75]: I.C BRAID  
"Six systems for shape design and representation : a review"  
Report P-75MM-O1 Bournemouth England, CAM-I Inc 1975
- [BRO 82]: BROWN  
"PADL2 : a technical summary"  
IEEE 1982
- [CAR 82]: W.CARLSON  
"An algorithm and data structure for 3D object synthesis"  
Computer Graphics vol 16 n 3 1982

- [CAS 85]: M. S.CASALE, E. L.STANTON  
 "An overview of analytic solid modeling"  
 IEEE CGA février 1985
- [CAS 87]: M. S.CASALE, E. L.STANTON  
 "Free-form solid modeling with trimmed surface patches"  
 IEEE CGA janvier 1987
- [CAS 85]: P. de CASTELJAU  
 "Formes à pôles" Mathématiques et C.A.O volume 2  
 éditions HERMES Paris 1985
- [CAT 74]: E.E CATMULL  
 "A subdivision algorithm for computer display of curved surfaces"  
 Tech. rep. UTEC-CSc-74-133, Det of Computer Science, University of  
 UTAH, Salt Lake City 1974
- [CAU 88]: R.CAUBET, Y.DUTHEN, R.PUJADO, M.Z. SANDOUK  
 "Lancer de rayons optimisés pour les surfaces gauches de formes libres"  
 MICAD 88 éditions HERMES PARIS 1988
- [CHE 86]: P. CHENIN, M. COSNARD, Y. GARDAN, F. ROBERT, Y. ROBERT, P.  
 WITOMSKI  
 "Méthodes de base" Mathématiques et C.A.O volume 1  
 éditions HERMES Paris 1986
- [CHO 90]: B.K. CHOI, W.S. YOO, C.S. LEE  
 "Matrix representation for NURB curves and surfaces"  
 Computer Aided Design volume 22 n 4 1990

- [CLE 88]: J G.CLEARY , G. WYVILL  
"Analysis of an algorithm for fast ray tracing using uniform space subdivision"  
The visual computer 1988
- [COO 87]: S A. COONS  
"Méthode matricielle" Mathématique et C.A.O volume 5  
éditions HERMES Paris 1987
- [DAN 89]: M.DANIEL  
"Modélisation de courbes et surfaces par des B-Splines. Application à la C.A.O et à la visualisation des formes. "  
Thèse à l'université de Nantes 1989
- [DOD 84]: J.R DODSWORTH  
"Current issues in solid modelling"  
CAD modelling éditions, Computational mechanics center Southampton  
1984
- [DOK 90]: TOR DOKKEN  
"Surface representation"  
Revue internationale de CFAO et d'infographie vol 5 n 2 éditions HERMES  
Paris
- [EAS 74]: C.M.EASTMAN,K. PREISS  
"A review of solid shape modeling based on integrity verification"  
C.A.D vol 16 n 2 1974
- [FAU 79]: I.D. FAUX, M.J. PRATT  
"Computational Geometry for Design and Manufacture"  
éditions John Wiley New-York 1979

- [FIT 90]: P.A. FITZHORN  
"Language of topologically valid bounding manifolds"  
Computer Aided Design volume 22 n 7 1990
- [FOL 82]: J.D. FOLEY, A. VAN DAM  
"Fundamental of interactive computer graphics"  
éditions ADDISON, WESLEY 1982
- [FOR 78 ]: A.R.FORREST  
" An unified approach to geometric modeling"  
ACM siggraph computer graphic 1978
- [GAR 82]: Y. GARDAN  
" Systèmes de C.F.A.O"  
éditions HERMES Paris 1982
- [GAR 83]: Y. GARDAN, M. LUCAS  
" Techniques graphiques interactives et C.A.O"  
éditions HERMES Paris 1983
- [GAR 87]: Y. GARDAN  
" La C.F.A.O introduction, techniques et mise en oeuvre"  
éditions HERMES Paris 1987 2ème édition
- [GAR 88]: Y. GARDAN  
" Elements de C.A.O volume 1 matériel et logiciel"  
éditions HERMES Paris 1987

- [GAR 89]: Y.GARDAN, D. MICHEL, M. SAHNOUNE  
" Etude comparée des formes paramétriques pour la modélisation des surfaces"  
Rapport de recherche LRI METZ 1989
- [GAR 90]: Y. GARDAN, D. MICHEL , M. SAHNOUNE  
" Comparaison des propriétés des formes rationnelles et non rationnelles pour la modélisation des surfaces"  
Actes de la 9ème conférence internationale de MICAD éditions HERMES Paris 1990
- [GAR 90]: Y. GARDAN  
" Sur quelques réflexions concernant le passage du modèle mathématique au modèle informatique en modélisation des solides"  
Rapport de recherche LRI Metz 1990
- [GAS 90]: Y.GARDAN, M.SAHNOUNE  
"Intégration des surfaces gauches dans les modèles de solides"  
Revue internationale de CFAO et d'infographie Vol 5 n 2 éditions HERMES Paris 1990
- [GIA 88]: H. GIAUME  
"Séminaire 'La modélisation des surfaces'"  
Journée du 21 juin 1988 éditions HERMES Paris 1988
- [GIA 90]: H . GIAUME  
"Applications des pôles généralisés"  
Revue internationale de CFAO et d'infographie éditions HERMES Paris 1990

- [GOL 85]: R . N GOLDMAN  
"The method of resolvents: A technique for the implicitization, inversion,  
and intersection of non planar parametric rational cubics curves."  
Computer Aided Geometric Design vol 2 1985
- [HAN 83]: P . HANRAHAN  
"Ray tracing algebraic surfaces"  
Computer graphics volume 17 n 3 1983
- [HOU 85]: E. G.HOUGHTON, J. D.FACTOR, C. L.SABHARWAL  
" Implementation of a divide and conquer method for intersection of  
parametric surfaces"  
Computer Aided Geometric Design 2 1985
- [JAC 80]: JACKINS, TANIMOTO  
"Octrees and their use in representing 3D objects"  
éditions ACADEMIC PRESS 1980
- [KAJ 82]: J. T. KAJIYA  
" Ray tracing parametric patches "  
ACM computer graphics volume 16 numéro 3 1982
- [KAJ 83]: J.T. KAJIYA  
"New techniques for ray tracing procedullary defined objects"  
Computer graphics volume 17 n 3 1983
- [KAY 86]: T. KAY, J.T KAJIYA  
"Ray tracing complex scenes"  
ACM vol 20 n 4 1986

- [KOR 61]: J . KORGANOFF  
"Méthodes de calculs numériques"  
Tome 1 algèbre non linéaire éditions DUNOD Paris 1961
- [LAN 79]: J. LANE  
"A generalised scan-line algorithm for the computer display of  
parametrically defined surfaces"  
Computer graphics and Image Processing, 11, 1979
- [LEE 82]: Y.T. LEE, A.A.A.G. REQUICHA  
"Algorithms for computing volumes and other integral properties" II  
A.C.M vol 25 n 9 1982
- [LEV 76]: J. LEVIN  
"A parametric algorithm for drawing picture of solid object composed of  
quadric surfaces"  
Communication of A.C.M vol 19 n 10 1976
- [MAN 88]: M. MANTYLA  
"An introduction to Solid modelling"  
éditions Computer Science Press Rockville, M.d 1988
- [MAN 82]: M . MANTYLA  
"An inversion algorithm for geometric models"  
Computer graphics vol 3 1982
- [MAR 87]: MARTIN, MARTIN  
"Les algorithmes de calculs d'intersection de solides définis par leurs  
bords"  
Revue internationale de CFAO et d'infographie éditions HERMES vol 2 n4  
Paris



- [NEW 79]: W.M. NEWMAN, R.F SPROULL  
"Principles of interactive computer graphics"  
éditions Mac-Graw-Hill New-York 1979
- [MIL 86]: J. R.MILLER  
" Sculptured Surfaces in solid modeling: Issues and alternative approaches"  
IEEE CGA decembre 1986
- [MIL 89 ]: J. R.MILLER  
" Architectural Issues in solid modelers"  
IEEE CGA decembre 1989
- [MIN 88]: C . MINICH  
"Operateurs d'extrusion et de révolution. Rôle des opérateurs d'EULER"  
Note interne LRI METZ
- [MIN 91]: C . MINICH  
Thèse de l'université de Metz (à paraître)
- [MOR 85]: M . E . MORTENSON  
"Geometric modeling"  
éditions JOHN WILEY and sons New-York 1985
- [PER 88]: B. PEROCHE, J. ARGENCE, D. GHAZANFARPOUR, D. MICHELUCCI  
"La synthèse d'images"  
éditions HERMES Paris 1988

- [PUC 89]: S. PUCCI  
"Utilisation des Octrees pour la conversion d'un modèle CSG à un modèle par les frontières"  
Rapport de recherche (89-2) LRI METZ 1989
- [RAL 65]: A . RALSTON  
"A first course in numerical analysis"  
éditions Mac Graw Hill 1965
- [REQ 83]: A.A.G. REQUICHA - H.B.VOELCKER  
" Solid modeling : Current status and research directions"  
IEEE CGA octobre 1983
- [REQ 85]: A.A.G REQUICHA, H.B VOELCKER  
"Boolean operations in solid modelling. Boundary evaluation and merging algorithms"  
Proceeding of IEEE vol 73 n 1 1985
- [REQ 87]: A.A. REQUICHA  
"Mathematical models of rigid solid objects"  
Tech memo 28, Production automation project, University of Rochester
- [RIE83]: R.F. RIESENFELD  
"A view of spline-Based Solid Modelling"  
Proc Autofact 5, SME Detroit 1983
- [ROG 76]: D.F. ROGERS, J.A. ADAMS  
"Mathematical elements for computer graphics"  
éditions Mac-Graw-Hill New-York 1976

- [ROG 85]: D.F. ROGERS  
"Procedural elements for computer graphics"  
éditions Mac-Graw-Hill New-York 1985
- [ROT 82]: S. D. ROTH  
" Ray casting for modeling solids"  
Computer graphics and Image processing 18 1982
- [ROC 87]: A. ROCWOOD  
"A generalised scanning technique for display of parametrically defined surfaces"  
IEEE 1987
- [SAH 88]: M . SAHNOUNE  
"Les B-Splines rationnelles"  
note interne LRI METZ 1988
- [SAR 83]: R. F. SARRAGA , W. C. Waters  
" Free-form surfaces in GMSolid: Goals ans issues"  
Proc General Motors Symp solid modeling Plenum New -York 1983
- [SED 84]: T.W. SEDERBERG , D.C. ANDERSON, R.N. GOLDMAN  
" Implicit representation of curves and surfaces"  
Computer Vision and Image processing 28 1984
- [SED 84]: T.W SEDERBERG, D.C ANDERSON  
"Ray tracing on steiner patches"  
Computer graphics vol 18 n 3 1984

- [SED 85]: T.W. SEDERBERG  
"Piecewise algebraic surface patches"  
Computer Aided Geometric Design 2 1985
- [SWE 86]: M . A . J SWEENEY, R.H BARTELS  
"Ray tracing free form B-Splines surfaces"  
IEEE 1984
- [TIL 80]: R. TILOVE  
"Set membership classification: An unified approach to geometric intersection problem"  
IEEE 1980
- [TIL 80]: R. TILOVE, A. REQUICHA  
"Closure of boolean operations on geometric entities"  
C.AD vol 12 n 5 1980
- [TIL 83]: W. TILLER  
" Rational B-Splines for curve and surface representation"  
IEEE CGA septembre 1983
- [TIL 84]: R . TILOVE  
"A null object detection algorithm for constructive solid geometry"  
Communications of A.C.M vol 27 n 7 1984
- [VER 73]: M . VERON  
"Contribution à l'étude des surfaces numériques Unisurf. Conditions de Raccordement"  
Thèse de Doctorat d'Etat, Université Nancy I 1973

- [WEI 85]: K . WEILER  
"Edge based data structures for solid modelling in curved-surface  
environnements"  
IEEE CG et A 1985
- [WIL 85]: P . R WILSON  
"Euler formulas and geometric modelling"  
IEEE CGA 1985
- [ZID 90]: A. ZIDNA  
" Contribution à la modélisation des carreaux troués"  
Thèse de l'université de Metz décembre 1990

## RESUME

La modélisation géométrique est un outil fondamental de la C.A.O. Actuellement, le monde de la C.A.O s'intéresse plus particulièrement aux modèles solides. Indépendamment des recherches consacrées aux modèles solides, se sont développés des systèmes fondés sur des modèles de surfaces. Mais de plus en plus, on aimerait avoir un modèle volumique avec des surfaces gauches pour pouvoir représenter des solides à formes libres. Nous nous proposons de contribuer à l'intégration des surfaces gauches dans les modèles solides. Après avoir rappelé les principes des différents modèles solides et leurs propriétés, nous essayons d'explicitier les difficultés liées au passage d'un modèle mathématique à un modèle informatique. Puis nous détaillons et comparons les différents modèles de surfaces. Enfin nous nous attachons à montrer, à partir de l'existant, par quels moyens on peut tenter d'intégrer les modèles de surface dans les modèles solides.

## MOTS CLES

B-Rep, B-Spline, C.A.O, CSG, Modélisation Géométrique, Modèle de solides, Modèle surfacique, NURBS, Surface Gauche, Surface de Bézier,

## RESUME

La modélisation géométrique est un outil fondamental de la C.A.O. Actuellement, le monde de la C.A.O s'intéresse plus particulièrement aux modèles solides. Indépendamment des recherches consacrées aux modèles solides, se sont développés des systèmes fondés sur des modèles de surfaces. Mais de plus en plus, on aimerait avoir un modèle volumique avec des surfaces gauches pour pouvoir représenter des solides à formes libres. Nous nous proposons de contribuer à l'intégration des surfaces gauches dans les modèles solides. Après avoir rappelé les principes des différents modèles solides et leurs propriétés, nous essayons d'explicitier les difficultés liées au passage d'un modèle mathématique à un modèle informatique. Puis nous détaillons et comparons les différents modèles de surfaces. Enfin nous nous attachons à montrer, à partir de l'existant, par quels moyens on peut tenter d'intégrer les modèles de surface dans les modèles solides.

## MOTS CLES

B-Rep, B-Spline, C.A.O, CSG, Modélisation Géométrique, Modèle de solides, Modèle surfacique, NURBS, Surface Gauche, Surface de Bézier,