

Conception et visualisation d'objets. Notes du cours de  
Christine Kéribin, rédigées par Pierre Pansu

February 27, 2003



# Chapter 1

## Synopsis du cours

On a choisi de présenter les logiciels de CAO en partant de la surface (attentes de l'utilisateur) pour plonger vers le coeur (algorithmes, outils mathématiques).

La XAO (CAX) (X doit être remplacé par conception, modification, exploration, fabrication...) est vieille d'une trentaine d'années tout au plus. Le terme CAO désignait initialement les outils logiciels qui permettent à l'ordinateur de remplacer la planche à dessin. Le champ d'application de la CAO s'est élargi (d'où le terme XAO). Les logiciels d'aujourd'hui peuvent, en incorporant des règles-métiers, capturer efficacement l'intention de l'utilisateur, et allègent ainsi sa tâche bien au-delà de ce que pouvait faire la planche à dessin.

Un logiciel de CAO se compose

- d'un modelleur géométrique ;
- d'un outil de visualisation ;
- d'un certain nombre d'applications ;
- d'un contrôleur qui gère les interactions entre les 3 outils précédents.

Le modelleur géométrique, c'est la planche à dessin. Son vocabulaire est celui de la géométrie (points, droites, plans, courbes B-splines, surfaces NURBS...) et de la topologie (sommets, arêtes, faces, intérieur/extérieur, union, intersection...). A ce langage de bas niveau, on superpose au chapitre 3 un langage de haut niveau, celui des entités technologiques (**features**). Chaque feature, un congé ou un chanfrein par exemple, résume une longue suite de commandes et d'opérations à effectuer par le modelleur et la visualisation, et contient des règles en usage dans un métier. Le meilleur moyen de saisir la notion de feature est de pratiquer l'atelier de conception de pièces et d'assemblage mécanique d'un logiciel comme CATIA.

Rentrons dans le détail du fonctionnement d'un modelleur géométrique. Abstraitemment, un objet 3D est défini par son bord, formé d'une collection de faces orientées. La partie mathématique du chapitre 4 traite le problème de savoir quand une telle donnée (appelée BRep pour **B**oundary **R**epresentation) est valide, i.e. borde effectivement un solide. La formule d'Euler fournit seulement une condition nécessaire, et on donne une condition nécessaire et suffisante. Sur le plan algorithmique, on détaille la structure de données qui représente un objet virtuel dans ACIS. En plus de la description BRep de l'objet, le modelleur de CATIA maintient au cours de la réalisation de l'objet un arbre des spécifications qui retrace la suite des features employés : il s'agit d'un modèle mixte BRep/CSG. L'un des avantages est la possibilité de revenir en arrière (**undo**), au prix d'une gestion délicate des mises à jour.

L'opération de base dont est capable un modelleur géométrique est le calcul de l'intersection de deux solides. Au chapitre 5, on décrit l'algorithme dans un cas simple : l'intersection d'un polyèdre avec un plan.

Encore plus profond, on trouve la méthode pour calculer une discrétisation approchée de la courbe d'intersection de deux surfaces. C'est un cas particulier de la méthode de cheminement. Au chapitre 6, on décrit l'algorithme ainsi que des outils pour ajuster les paramètres : initialisation, pas, critère de fin.

Une fois trouvée une suite de points proches de la courbe d'intersection, on peut construire une courbe B-spline qui passe par ces points par interpolation. Cet algorithme, décrit au chapitre 7, se ramène à l'inversion d'une matrice carrée. Il est utile de comprendre la différence entre interpolation et approximation. Une courbe construite par interpolation passe par les points prescrits. Une approximation ne passe pas par les points prescrits, mais minimise seulement l'écart quadratique à ces points. Elle peut avoir moins de points de contrôle qu'il n'y a de points prescrits.

La deuxième fonction essentielle d'un logiciel de CAO est la production de vues de l'objet. Le chapitre 8 contient une brève introduction à la synthèse d'images : principes physiologiques et physiques, modèles d'éclairage, lancer de rayon, perspective.

## Chapter 2

# Fondements de la XAO

### 2.1 Qu'est-ce que la XAO ?

La XAO (CAX) (X doit être remplacé par conception, modification, exploration, fabrication...) est vieille d'une trentaine d'années tout au plus. Le terme CAO désignait initialement les outils logiciels qui permettent à l'ordinateur de remplacer la planche à dessin. Le champ d'application de la CAO s'est élargi (d'où le terme XAO).

#### 2.1.1 Attentes de l'utilisateur

Un logiciel de XAO est un outil pour créer des objets virtuels,

- les visualiser,
- en prédire des propriétés (dimensions, encombrement, faisabilité du montage/démontage, simulation aérodynamique, thermique, acoustique...).

De nos jours, le logiciel doit aussi

- être utilisable par des non spécialistes ;
- capturer efficacement l'intention de design de l'utilisateur ;
- incorporer des règles métier, le savoir-faire de l'entreprise (**knowledgeware**) ;
- permettre à tous les acteurs autorisés, dans et parfois hors de l'entreprise, d'agir sur les objets (**concurrent engineering**).

#### 2.1.2 Un champ d'application élargi

La demande des utilisateurs s'est élargie : le logiciel de XAO doit être (au minimum) intégré au dispositif qui assure les fonctions suivantes.

- Processus de fabrication : pilotage de machines outils ;
- Suivi de l'évolution des produits dans le temps : gérer les différentes versions ;
- Gestion des données techniques : les mettre à la disposition de tous les acteurs, accélérer la prise de décision ;
- Gestion de stocks, de flux : cela nécessite une modélisation de l'entreprise elle-même ;
- Gestion de la paie, du personnel.
- Intégration des sous-traitants.

### 2.1.3 A chacun sa XAO

Le marché propose des produits dont le degré de sophistication (et le coût) varie, depuis

- les outils de dessin de pièces (e.g. Autocad, SolidConcept, Solid Modeling) ;
- les outils de conception de produits composites (e.g. Solidworks (Dassault-Systèmes), Solid-edge (Unigraphics), Desktop (Parametric Technologies), Topsolid (Topcad)) ;  
jusqu'aux
- outils de gestion de processus (e.g. Enovia/VPM (Dassault-Systèmes), Windchill (Parametric Technologies), Metaphase (SDRC), Iman (Unigraphics)).

Certains produits (e.g. CATIA) ont une vocation généraliste. Ils s'efforcent de traiter les problèmes fondamentaux dans la plus grande généralité possible, ils proposent des applications adaptées à un grand nombre de métiers différents.

D'autres sont ciblés pour un métier, ce sont des *produits-niches* : aérodynamique, architecture, architecture navale, assemblage, chaudronnerie, électricité, électronique, emboutissage, fonderie, forge, outillage, plasturgie, prototypage, robotique, routage, style, thermique, tôlerie, topographie...

### 2.1.4 Tendances

- Reproduire les métiers.
- Digitaliser toute l'entreprise, jusqu'aux sous-traitants.
- Permettre le partage de l'expérience à l'échelle de l'entreprise.
- Abolir les distances : rôle d'internet pour le partage de données au sein de l'entreprise, développement de services CAO à distance.

## 2.2 Un peu d'histoire

Avant : la planche à dessins, suivi de la maquette. Multiples inconvénients.

- Modifications pénibles.
- Précision insuffisante, difficulté à représenter des surfaces gauches.
- La réalisation et la lecture des plans nécessite une expertise.
- Coût prohibitif des maquettes.
- Facteur temps.

Préhistoire : Bézier chez Renault, De Casteljaou chez Citroën, à la fin des années 60.

Les premiers produits 2D apparaissent au début des années 70 (l'aéronautique est en pointe). Ils rencontrent des résistances : changement radical de métier pour les dessinateurs. Malgré la lenteur des ordinateurs, le gain de productivité croissant va convaincre.

La 3D, d'abord polyédrique et filaire, puis incorporant courbes et surfaces gauches, va se répandre en même temps que les stations graphiques. Epoque de standardisation des menus. La 4D aujourd'hui signifie

- le rendu réaliste en temps réel,
- la simulation de mouvements de caméra,

- la simulation des cinématiques,
- la simulation des évolutions.

Le temps de création d'un modèle de voiture est passé de 10 à 2 ou 3 ans. Ce gain de productivité résulte d'une évolution des méthodes et des processus au cours de laquelle les outils CAO ont joué un rôle essentiel.

## 2.3 Les composantes d'un produit de CAO

Un logiciel de CAO doit gérer plusieurs modes de représentation de l'objet.

- Le mode géométrique : la structure de données interne.
- Le mode graphique : ce qui est affiché à l'écran.
- Le mode fonctionnel : la liste des spécifications fournies par l'utilisateur.

La composition du logiciel reflète ces différents modes.

1. Le modèle, sur lequel agit le modéleur géométrique. L'essentiel du contenu du cours y est consacré.
2. Les vues : la vue 3D réaliste, l'arbre des spécifications, éventuellement des représentations simplifiées (maillage). Ces vues doivent autoriser une lecture inverse : possibilité de sélectionner un élément.
3. Un certain nombre d'applications. Elles réalisent les fonctionnalités offertes à l'utilisateur. Depuis les entrées/sorties (visualisation, impression), le calcul de grandeurs géométriques (distances, inerties), jusqu'aux fonctions métiers : assemblage de pièces, production de plans, simulation d'usinage, moulage, fraisage.
4. Un contrôleur gère les interactions entre modèle, vues et applications.

## 2.4 Contraintes et difficultés

### 2.4.1 Contraintes

Assurer la cohérence entre affichage, détection et représentation interne : gérer des versions de travail et des versions archivées. Distinguer les données stables (références absolues, ordre des éléments, nommage) des données instables (positions géométriques, résultats de calculs, numéros d'ordre assignés par l'ordinateur). Décider quel est le modèle maître, en cas d'ingénierie simultanée.

Pérennité du stockage : la durée de vie d'un modèle d'avion est supérieure à 20 ans.

Portabilité : bien isoler ce qui dépend de la plateforme.

Optimisation des calculs et de l'utilisation de la mémoire : il faut souvent choisir un compromis entre les deux.

Sémantique : l'interface est destinée à un utilisateur non informaticien, mais expert d'un métier.

Prévoir l'évolution ultérieure du logiciel. Par exemple, de nouvelles primitives ont été ajoutées dans la version 5 de CATIA.

### 2.4.2 Difficultés

Il y a des problèmes fondamentaux en modélisation géométrique pour lesquels il n'existe pas actuellement de solution universelle.

- Gérer les incohérences inhérentes à la modélisation et à l'algorithmique. Le calcul de l'intersection de deux solides n'est pas rigoureusement commutatif. Une arête commune à deux faces possède une représentation dans le plan des paramètres de chaque surface : cela donne deux courbes voisines mais distinctes dans l'espace, laquelle choisir ?
- Gérer les tolérances (**tolerant modelling**). Un modéleur doit calculer avec une haute précision ( $10^{-16}$ ) mais accepter des données externes de faible précision (données de scanner, données fournies par un autre logiciel). Or certaines décisions (collision ou non) dépendent de la tolérance choisie.



# Chapter 3

## Features

Le modelleur géométrique, c'est la planche à dessin. Son vocabulaire est celui de la géométrie (points, droites, plans, courbes B-splines, surfaces NURBS,...) et de la topologie (sommets, arêtes, faces, intérieur/extérieur, union, intersection,...). A ce langage de bas niveau, on superpose un langage de haut niveau, celui des entités technologiques (**features**). Chaque feature, un congé ou un chanfrein par exemple, résume une longue suite de commandes et d'opérations à effectuer par le modelleur et la visualisation, et contient des règles en usage dans un métier.

### 3.1 Vue méthodologique de conception

L'objet visualisé sur la figure possède la description géométrique suivante.

- Un pavé,
- on fait la différence du pavé avec 4 demi-espaces,
- on fait la différence du solide obtenu avec 2 cylindre,
- on fait la réunion du solide obtenu avec un petit pavé,
- on fait la réunion du solide obtenu avec un petit pavé oblique en veillant à ce qu'il ne dépasse pas.

Voici une deuxième description géométrique de l'objet.

- 30 sommets
- 61 arêtes (dont 53 rectilignes et 8 demi-cercles)
- 22 faces (dont 18 planes et 4 demi-cylindres)
- un solide.

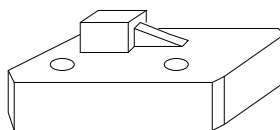


Figure 3.1:

Pour compléter la description, il faut donner les coordonnées de chaque sommet, pour chaque arête, indiquer ses sommets, pour chaque face, indiquer les arêtes qui la délimitent et la normale sortant du solide.

Or le concepteur le décrira plutôt comme suit.

- Un socle peu épais à base rectangulaire,
- des chanfreins aux quatre angles,
- deux trous cylindriques débouchants,
- un petit pavé,
- un raidisseur s'appuyant sur le côté droit du petit pavé.

Pour compléter la description, il suffit de donner les dimensions du socle, les coordonnées des axes des trous, les diamètres des trous, les dimensions du pad. Les dimensions exactes des chanfreins et du raidisseur sont d'importance secondaire.

La description métier est plus économique, et accessible à un non-mathématicien. Les mots-clés *chanfrein*, *trou débouchant*, *raidisseur*, qui viennent des métiers (menuiserie, fonderie) suffisent pour résumer une série d'opérations, comme on va le voir au paragraphe suivant sur un exemple. Ils désignent ce qu'on appelle des *features* (faute d'un terme français suffisamment répandu). Des règles en usage dans les métiers (par exemple : ne pas percer un trou trop près du bord) peuvent être incorporées.

## 3.2 Exemple : le congé d'arête

Le *congé* de raccordement (**fillet**) consiste à remplacer une arête vive rentrante par une face arrondie de section circulaire qui connecte les deux faces de l'arête vive (voir figure). La construction d'un congé comporte les étapes suivantes.

Phase de dialogue

1. Sélection à la souris par l'utilisateur d'une arête. Question à l'utilisateur : souhaite-t-il spécifier le rayon de la face cylindrique à insérer ?

Phase interne

2. Recherche des faces d'appui (i.e. navigation dans la topologie de l'objet).
3. Calcul de la surface arrondie et des lignes de soudure (algorithme de cheminement, voir chapitre 6).
4. Calcul de l'intersection de la surface arrondie avec deux faces verticales.
5. Relimitation des 4 faces concernées (suppression d'une arête, création de 2 arêtes, modification de 2 arêtes et de 2 faces, création d'une face).

Phase de Mise à jour

6. Ajout du feature congé dans l'arbre des spécifications.
7. Stockage des nouvelles données.
8. Affichage de l'objet modifié (création de facettes, lancer de rayon...).
9. Mise à jour de la base de données techniques.

Le feature congé apparaît comme la combinaison de trois éléments.

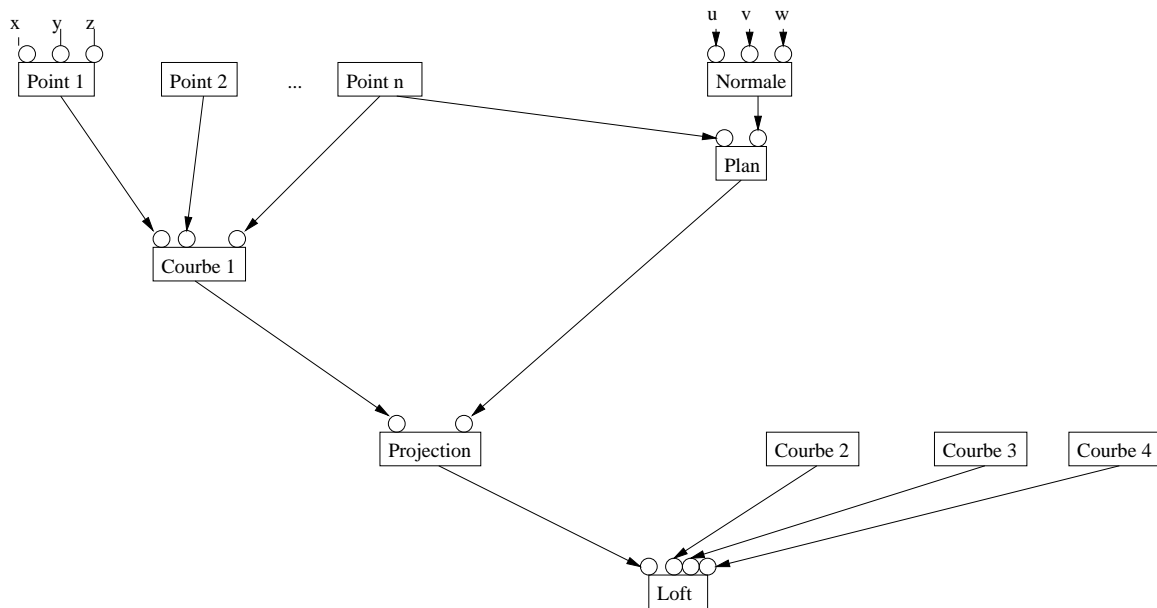


Figure 3.2:

- Une spécification (fabriquer un congé), comportant un attribut (arête).
- Un opérateur *build* (composé de plusieurs opérateurs).
- Un résultat.

Il peut donc être représenté par une boîte où arrive une flèche (l'attribut arête) et d'où part une flèche (build) vers le résultat. En empilant de tels schémas, on obtient un graphe orienté sans boucles, l'*arbre des spécifications*.

Exemple : construction d'une surface à partir de 3 courbes et d'une courbe obtenue en projetant une courbe NURBS sur un plan.

Attention. Le build est lancé par la mise à jour. Avant la phase de mise à jour, il peut y avoir incohérence entre la spécification et le résultat.

### 3.3 Features utilisés en mécanique

Voici un catalogue non exhaustif.

#### 3.3.1 Motifs

Création de primitives (point, droite, cercle, ellipse, spline interpolée, plan,...).

#### 3.3.2 Transformations

Translation, rotation, symétrie, homothétie. Déformation d'une courbe ou d'une surface définie par des points de contrôle. Découpe.

#### 3.3.3 Features à contour comportant un ajout de matière

Extrusion. L'utilisateur sélectionne un contour fermé plan (qui borde un domaine du plan) et spécifie une longueur et l'une des deux directions normales au plan. Le solide obtenu est la réunion

des translatsés du domaine plan, par un vecteur situé sur la normale choisie, de longueur inférieure à la longueur spécifiée.

Raidisseur. Pavé soudé entre deux faces adjacentes destiné à renforcer l'objet. L'utilisateur spécifie l'arête commune aux deux faces.

Révolution. L'utilisateur sélectionne un plan, un axe de ce plan, un contour qui ne coupe pas l'axe, un angle entre  $0^0$  et  $360^0$ . La surface obtenue est la réunion des images du contour choisi par des rotations d'axe choisi et d'angle compris entre  $0^0$  et l'angle choisi.

Bossage. Ajout d'une extrusion.

Nervure. Pavé soudé le long d'une plaque pour éviter qu'elle ne plie. L'utilisateur spécifie un contour le long de la plaque. L'ordinateur ajoute un solide obtenu en balayant un carré le long du contour.

### 3.3.4 Features à contour comportant un enlèvement de matière

Poche. Soustraction à un solide d'une extrusion. L'utilisateur sélectionne un contour fermé dans une face du solide.

Trou. Soustraction à un solide d'un cylindre. L'utilisateur sélectionne un cercle dans une face du solide. Un trou peut être borgne ou débouchant, voir paragraphe suivant.

Gorge. Poche parallèle (et non perpendiculaire) à une face.

Rainure. Soustraction d'une nervure à un solide. L'utilisateur spécifie un contour le long d'une face du solide.

### 3.3.5 Features contextuels

Ce sont les features pour lesquels il n'existe pas de définition absolue. Par exemple, il n'est pas évident comment on doit terminer le congé le long d'une arête rentrante d'un solide dont les arêtes vives sortantes ont été préalablement arrondies.

Dans ce cas, l'algorithme doit donner un bon résultat dans les situations les plus courantes (seul un expert du métier peut en juger) et renonce à traiter les exceptions.

Arrondi (**round** ou **blend**). Remplace une arête vive sortante par une surface de section circulaire qui connecte les deux faces adjacentes à l'arête vive.

Chanfrein (**chamfer**). Casse une arête vive en la remplaçant par une face inclinée.

Coque (**shell**). Procédé pour obtenir un solide mince à partir d'un solide massif. L'utilisateur spécifie le solide massif, une épaisseur, éventuellement des faces à retirer.

Dépouille (**draft**). Modification apportée à un moule pour faciliter le démoulage. Les faces tangentes à la direction de démoulage sont inclinées d'un angle minimum.

Surépaisseur (**offset** ou **thickness**) : surplus de matière prévu dans la forme du moule, de sorte que la pièce moulée puisse être usinée avant d'atteindre l'état de surface désiré.

## 3.4 Associativité

Un feature bien conçu doit faire en sorte que si des paramètres d'une construction complexe sont modifiés, l'effet du feature corresponde toujours à l'intention de l'utilisateur. Autrement dit, il doit obtenir de l'utilisateur une information plus riche.

Exemple : différents types de trous. Pour un trou borgne (on ne peut pas passer une tige au travers), l'utilisateur doit donner une profondeur. Mais pour un trou débouchant, il vaut mieux faire préciser à l'utilisateur

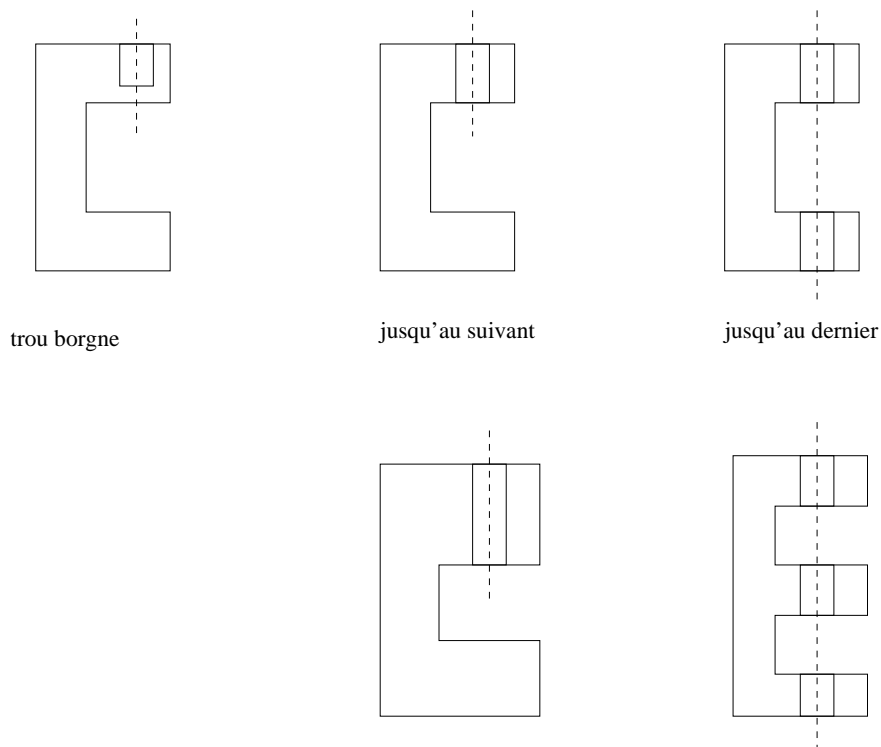


Figure 3.3:

- s'il souhaite que le trou débouche sur la première face rencontrée et s'arrête là,
- ou s'il préfère que le trou traverse le solide de part en part (jusqu'à la dernière face rencontrée).

### 3.5 Problème de la reconnaissance de features

La figure ci-contre peut-être interprétée

- ou bien comme une coque munie d'un raidisseur,
- ou bien comme 2 poches creusées dans un pavé.

Du point de vue de la fabrication, c'est la 2ème interprétation qui est la bonne. Il est en effet malcommode d'ajouter un raidisseur au fond d'une poche profonde.

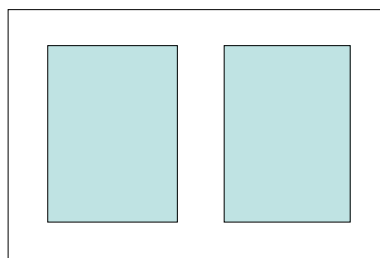


Figure 3.4:

C'est dire que le problème, à partir d'une description géométrique d'un objet, d'en déduire automatiquement un processus de fabrication, n'est pas simple. Les logiciels traitant ce problème ont fait leur apparition récemment. Néanmoins, il s'agit d'un problème largement ouvert.

## Chapter 4

# Modélisation du solide

### 4.1 Distinguer géométrie et topologie

Une boîte de conserve cylindrique peut être décrite par 4 faces, 2 faces planes paramétrables respectivement par

$$(u, v) \mapsto (u, v, 0) \quad \text{et} \quad (u, v) \mapsto (u, v, h)$$

pour  $(u, v)$  dans le disque de centre 0 et de rayon  $r$ , et deux faces tracées sur un cylindre

$$(u, v) \mapsto (r \cos(u), r \sin(u), v)$$

pour  $v \in [0, h]$  et  $u \in [0, \pi]$  (resp.  $u \in [-\pi, 0]$ ).

Une petite erreur sur l'intervalle de variation de  $v$  dans la paramétrisation des faces cylindriques transforme une boîte de conserve fermée en une boîte de conserve ouverte. Autrement dit, une perturbation infime de la *géométrie* provoque un changement radical de *topologie*.

La topologie se préoccupe de questions comme

- l'objet est-il ouvert ou fermé ?
- quel est l'intérieur, l'extérieur ?
- ces deux faces sont-elles oui ou non collées l'une à l'autre le long d'une arête ?
- ces deux courbes se raccordent-elles en ce sommet ?
- quelles sont leurs orientations respectives ?

Une fois qu'on a bien spécifié ce genre de contraintes sur l'objet, de petites perturbation des éléments (positions des sommets, formes des arêtes et des faces) affecteront peu l'objet et les opérations (intersections...) qu'on va faire dessus.

Les mots-clé de la topologie dans le contexte de la CAO sont

- courbe plane, courbe gauche ;
- surface ;
- face ;
- arête ;
- ordre circulaire ;
- orientation ;
- coïncidence ;

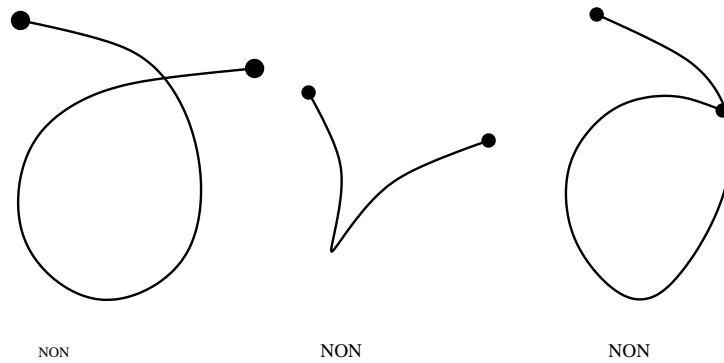


Figure 4.1:

- adjacence ;
- variété.

La géométrie se préoccupe de la forme exacte des objets. Ses mots-clés sont

- droite, cercle, ellipse ;
- courbe de Bézier, B-spline, NURBS ;
- plan, sphère, cylindre, cône ;
- carreau de Bézier, surface NURBS,
- tube, surface de révolution, surface de balayage, loft ;
- longueur, volume, masse, moment d'inertie ;
- courbure, torsion.

## 4.2 Modèle BRep

Pour un logiciel de CAO, la représentation interne d'un solide consiste à lister des objets composant le bord du solide et leurs relations. Par exemple, le bord est formé de faces, i.e. de morceaux découpés dans des surfaces, chaque morceau étant bordé par des arêtes, i.e. des courbes tracées sur des surfaces. Dans cette section, on donne des définitions mathématiques des termes employés en CAO.

### 4.2.1 Courbes

Pour un modéleur géométrique, une courbe dans le plan ou l'espace, c'est une application  $c$  d'un intervalle fermé de  $\mathbf{R}$  dans  $\mathbf{R}^3$ . Le logiciel ne sera capable d'effectuer des opérations sur la courbe (comme la construction d'une surface par balayage, par révolution...) que si les conditions suivantes sont satisfaites.

- Le vecteur vitesse  $c'(t)$  ne s'annule jamais.
- La courbe ne possède pas de point double : à deux valeurs distinctes de  $t$  correspondent deux points distincts dans l'espace.



Autrement dit, les cas de figure suivants sont interdits.

Ces conditions garantissent que l'aspect de la courbe au voisinage de chaque point est celui attendu : à un changement de coordonnées (curvilignes) près, la courbe coïncide localement avec une droite, ou avec une demi-droite dans le cas des extrémités. Le théorème suivant précise ceci.

**Théorème 1** Soit  $c : [a, b] \rightarrow \mathbf{R}^n$  une courbe paramétrée de classe  $C^k, k \geq 1$ . On suppose que

- Le vecteur vitesse  $c'(t)$  ne s'annule jamais.
- La courbe ne possède pas de point double : à deux valeurs distinctes de  $t$  correspondent deux points distincts dans l'espace.

Notons  $X = \{c(t) ; t \in [a, b]\}$  et  $\dot{X} = \{c(t) ; t \in ]a, b[ \}$ . Alors

1. Pour tout  $p \in \dot{X}$ , il existe un voisinage  $U$  de  $p$  dans  $\mathbf{R}^n$ , un difféomorphisme de classe  $C^k$   $\phi : U \rightarrow V$  sur un voisinage  $V$  de 0 dans  $\mathbf{R}^n$  et une droite  $D$  passant par l'origine telle que  $\phi(X \cap U) = D \cap V$ .
2. Pour  $p = c(a)$  ou  $p = c(b)$ , il existe un voisinage  $U$  de  $p$  dans  $\mathbf{R}^n$ , un difféomorphisme de classe  $C^k$   $\phi : U \rightarrow V$  sur un voisinage  $V$  de 0 dans  $\mathbf{R}^n$  et une demi-droite  $D_+$  passant par l'origine telle que  $\phi(X \cap U) = D_+ \cap V$ .

**Preuve.** Théorème des fonctions implicites + argument de compacité. ■

Dans la terminologie de la topologie différentielle, un sous-ensemble qui satisfait à la condition 1 de la conclusion du théorème est une *sous-variété de dimension 1* (**1-dimensional submanifold**) de  $\mathbf{R}^n$ . Un sous-ensemble dont chaque point satisfait à l'une des conditions 1 et 2 de la conclusion du théorème est une *sous-variété à bord de dimension 1* (**1-dimensional submanifold with boundary**) de  $\mathbf{R}^n$ , les points concernés par le modèle 2 sont les points du bord. Ils forment un ensemble localement fini. Pour abrégier, on emploiera le terme *courbe régulière*. On désignera (de façon peu satisfaisante) l'ensemble des points concernés par le modèle 1 par *l'intérieur* de la courbe.

### 4.2.2 Surfaces

Pour un modéleur géométrique, une surface dans l'espace, c'est une application  $s$  d'un domaine plan fermé et borné dans  $\mathbf{R}^3$ . Le logiciel ne sera capable d'effectuer des opérations sur la courbe (comme la construction d'une surface équidistante par offset...) que si les conditions suivantes sont satisfaites.

- Les vecteurs dérivées partielles  $\frac{\partial X}{\partial u}$  et  $\frac{\partial X}{\partial v}$  sont partout linéairement indépendants.
- La surface ne possède pas de point double : à deux valeurs distinctes de  $(u, v)$  prises dans  $D$  correspondent deux points distincts dans l'espace.

Autrement dit, les cas de figure suivants sont interdits.

Ces conditions garantissent que l'aspect de la surface au voisinage de chaque point est celui attendu : à un changement de coordonnées (curvilignes) près, la surface coïncide localement avec un plan, ou avec un demi-plan le long de son bord. Le théorème suivant précise ceci.

**Théorème 2** Soit  $D$  un domaine plan fermé et borné, bordé par une courbe régulière. Soit  $X : D \rightarrow \mathbf{R}^3$  une surface paramétrée de classe  $C^k, k \geq 1$ . On suppose que

- Les vecteurs dérivées partielles  $\frac{\partial X}{\partial u}$  et  $\frac{\partial X}{\partial v}$  sont partout linéairement indépendants.
- La surface ne possède pas de point double : à deux valeurs distinctes de  $(u, v)$  prises dans  $D$  correspondent deux points distincts dans l'espace.

Notons  $X = \{X(u, v) ; (u, v) \in D\}$  et  $\dot{X} = \{X(u, v) ; (u, v) \in \dot{D}\}$ . Alors

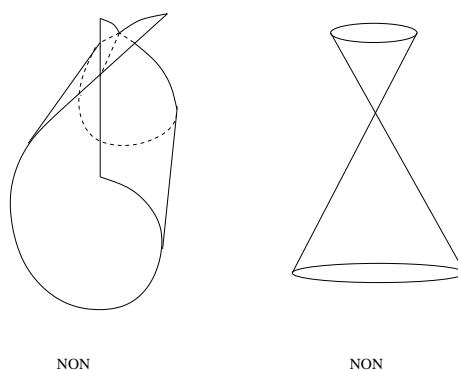


Figure 4.2:

1. Pour tout  $p \in \overset{\circ}{X}$ , il existe un voisinage  $U$  de  $p$  dans  $\mathbf{R}^3$ , un difféomorphisme de classe  $C^k$   $\phi : U \rightarrow V$  sur un voisinage  $V$  de 0 dans  $\mathbf{R}^3$  et un plan  $\Pi$  passant par l'origine telle que  $\phi(X \cap U) = \Pi \cap V$ .
2. Pour tout  $p \in X_s$  et  $\text{minus} \overset{\circ}{X}$ , il existe un voisinage  $U$  de  $p$  dans  $\mathbf{R}^3$ , un difféomorphisme de classe  $C^k$   $\phi : U \rightarrow V$  sur un voisinage  $V$  de 0 dans  $\mathbf{R}^n$  et un demi-plan  $\Pi_+$  passant par l'origine telle que  $\phi(X \cap U) = \Pi_+ \cap V$ .

**Preuve.** Théorème des fonctions implicites + argument de compacité. ■

Dans la terminologie de la topologie différentielle, un sous-ensemble qui satisfait à la condition 1 de la conclusion du théorème est une *sous-variété de dimension 2* (**2-dimensional submanifold**) de  $\mathbf{R}^3$ . Un sous-ensemble dont chaque point satisfait à l'une des conditions 1 et 2 de la conclusion du théorème est une *sous-variété à bord de dimension 2* (**2-dimensional manifold with boundary**) de  $\mathbf{R}^3$ , les points concernés par le modèle 2 sont les points du bord. Ils forment une sous-variété sans bord de dimension 1 de  $\mathbf{R}^3$ . Pour abrégier, on emploiera le terme *surface régulière* pour désigner une sous-variété de dimension 2. On désignera (de façon peu satisfaisante) l'ensemble des points concernés par le modèle 1 par l'*intérieur* de la surface.

**Définition 1** Soit  $X$  une surface régulière, paramétrée par  $X : D \rightarrow \mathbf{R}^3$ . Le plan tangent à  $X$  en  $p$  est le plan vectoriel engendré par les vecteurs  $\frac{\partial X}{\partial u}$  et  $\frac{\partial X}{\partial v}$ .

### 4.2.3 Faces planes

Les faces d'un cube sont des carrés. Un carré est un domaine plan mais il n'est pas bordé par une courbe régulière. D'où la nécessité d'une notion plus vaste.

**Définition 2** Un sous-ensemble fermé  $X \subset \mathbf{R}^n$  est dit propre (**proper**) s'il est égal à l'adhérence de son intérieur.

Pour un domaine plan, cela signifie qu'il n'y a pas de filaments.

**Définition 3** Un graphe planaire est une réunion finie de courbes régulières dont les intérieurs sont deux à deux disjoints. On appelle arêtes (**edges**) les courbes qui constituent le graphe et sommets (**vertices**) leurs extrémités.

**Définition 4** Une face plane (**planar face**) est un fermé propre du plan dont le bord est un graphe planaire

Autrement dit, les cas de figure suivants sont exclus.

Alors que les cas de figure suivants sont admis.

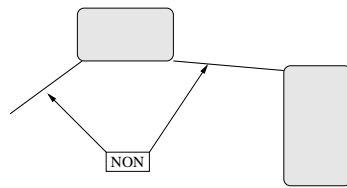


Figure 4.3:

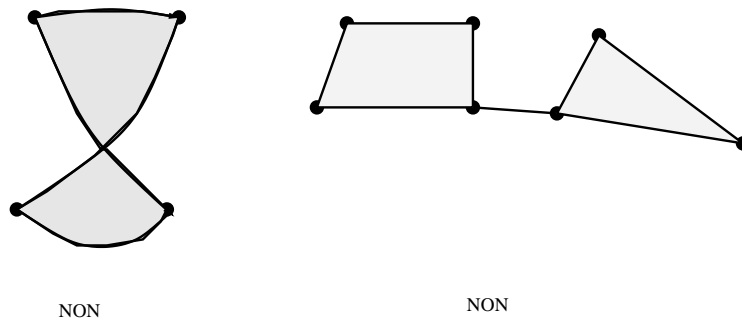


Figure 4.4:

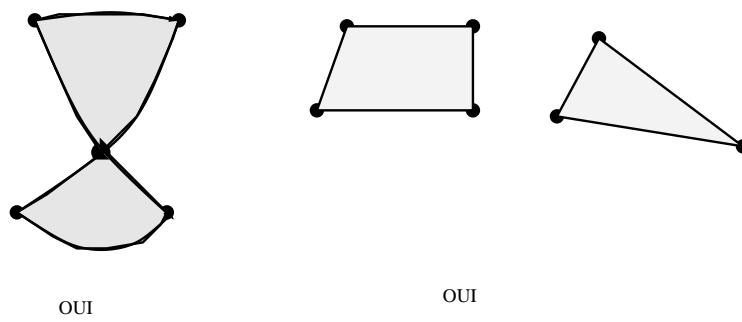


Figure 4.5:

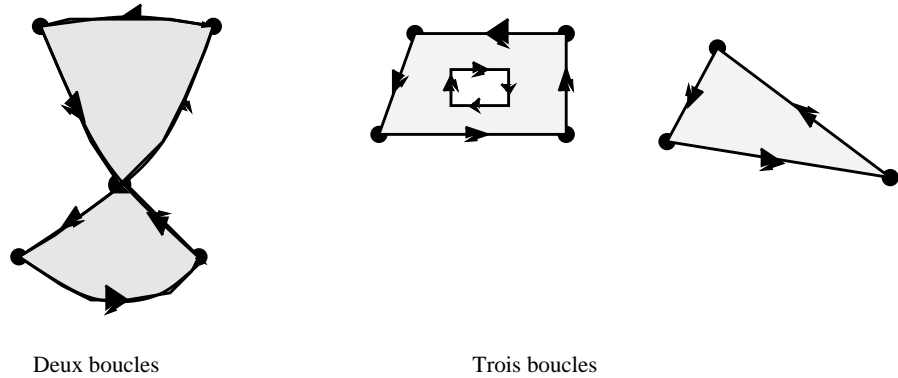


Figure 4.6:

#### 4.2.4 Boucles

Soit  $F$  une face plane. On oriente son bord suivant la convention de *la matière à gauche* : lorsqu'on parcourt une arête du bord, on trouve la face  $F$  sur sa gauche.

Soit  $p$  un sommet du bord de  $F$ . Traçons un petit cercle  $C$  de centre  $p$ . Orientons ce cercle dans le sens trigonométrique. Alors  $F \cap C$  est une réunion d'intervalles de la forme  $[a, b]$ , où  $a$  et  $b$  sont les intersections de  $C$  avec deux arêtes,  $a = C \cap A$  et  $b = C \cap B$ . Nécessairement, l'arête  $B$  arrive en  $p$  et  $A$  en part. On appelle  $A$  le *successeur* de  $B$ . Chaque arête possède un unique successeur et un unique prédécesseur.

On appelle *boucle* (loop) une suite d'arêtes correspondant à un cycle de la permutation successeur. Le bord de  $F$  est ainsi décomposé en un nombre fini de boucles.

#### 4.2.5 Caractéristique d'Euler-Poincaré

**Théorème 3** *Il existe une application  $\chi$ , appelée caractéristique d'Euler-Poincaré (Euler characteristic) qui associe un entier relatif à chaque espace topologique  $X$ , et qui possède les propriétés suivantes.*

- Si  $X$  et  $Y$  sont homéomorphes, alors  $\chi(X) = \chi(Y)$ .
- Si  $X$  est un point, un segment ou un polygone convexe fermé, alors  $\chi(X) = 1$ .
- Si  $Y$  et  $Z$  sont deux sous-espaces d'un même espace topologique, alors

$$\chi(Y \cup Z) = \chi(Y) + \chi(Z) - \chi(Y \cap Z).$$

**Exemple.** Si  $G$  est un graphe planaire possédant  $A$  arêtes et  $S$  sommets, alors  $\chi(G) = S - A$  (preuve par récurrence sur le nombre d'arêtes).

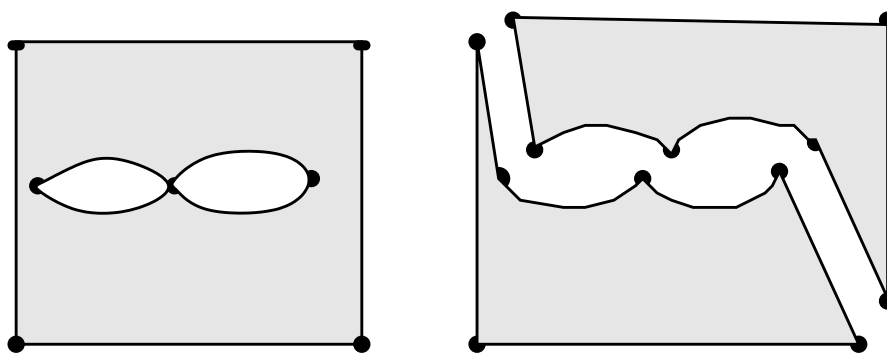
**Exemple.** En utilisant les axiomes, calculer la caractéristique d'Euler de la face plane décrite sur la figure.

**Théorème 4** *Soit  $F$  une face plane connexe, bordée par  $b$  boucles. Alors l'intérieur de  $F$ ,  $\text{int}(F)$ , a pour caractéristique d'Euler-Poincaré*

$$\chi(\text{int}(F)) = 2 - b.$$

*Si de plus les boucles du bord de  $F$  sont disjointes et sans points doubles (i.e. un même sommet est utilisé au plus une fois dans une boucle), alors*

$$\chi(F) = 2 - b.$$



Calcul de la caractéristique d'Euler par la formule d'addition

Figure 4.7:

**Preuve.** Voir [?]. ■

**Remarque.** La caractéristique d'Euler est non seulement invariante par homéomorphisme, c'est de plus un invariant d'*homotopie*.

**Définition 5** On dit que deux espaces topologiques  $X$  et  $Y$  sont homotopes (**homotopic**) s'il existe deux applications  $f : X \rightarrow Y$  et  $g : Y \rightarrow X$  et des familles continues d'applications  $h_t : X \rightarrow X$  et  $k_t : Y \rightarrow Y$  telles que

- $h_0 = f \circ g$  et  $h_1 = id_X$  ;
- $k_0 = g \circ f$  et  $k_1 = id_Y$ .

**Exemple.** Soit  $X$  l'anneau  $X = \{(x, y) ; 1/2 < x^2 + y^2 < 2\}$  et  $Y$  le cercle unité. Alors  $X$  et  $Y$  sont homotopes.

En effet, soit  $f : X \rightarrow Y$  la projection radiale  $(x, y) \mapsto (x, y)/\sqrt{x^2 + y^2}$  et  $g : Y \rightarrow X$  l'injection. Pour déformer  $f \circ g$ , il suffit de poser

$$h_t(x, y) = \frac{1}{t + (1-t)\sqrt{x^2 + y^2}}(x, y).$$

Comme  $g \circ f = id_Y$ ,  $k_t = id_Y$  convient.

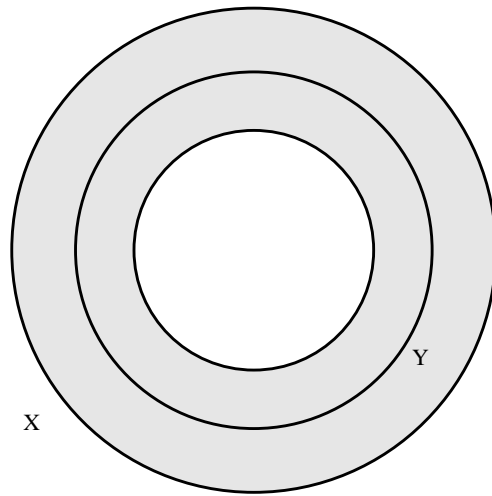
On montre que toute face plane  $F$  est homotope à un graphe planaire à un seul sommet. Le nombre d'arêtes de ce graphe s'appelle la *connectivité* (**connectivity**)  $\kappa(F)$  de la face. L'invariance par homotopie de la caractéristique d'Euler donne

$$\chi(F) = 1 - \kappa(F).$$

**Exemple.** La connectivité de la face de la figure 9 vaut 2.

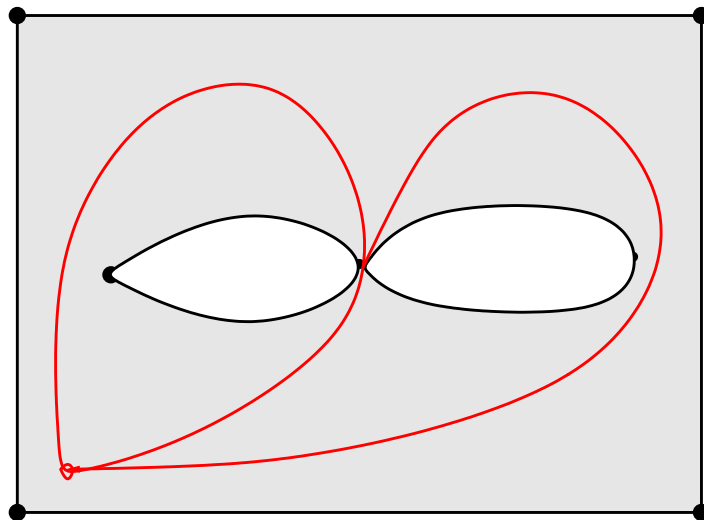
#### 4.2.6 Orientation

Pour un modéleur géométrique, un solide est donné par son bord. La donnée du bord en tant qu'ensemble de points est insuffisante. En effet, une sphère borde deux solides, une boule, et le complémentaire d'une boule. Certes, on peut éliminer le complémentaire de la boule parce qu'il n'est pas borné. C'est un critère global, inutilisable lors de modifications locales. Il est plus simple de spécifier, pour chaque morceau du bord, de quel côté se trouve le solide considéré. Cela consiste à se donner une orientation normale.



Un anneau est homotope à un cercle

Figure 4.8:



Connectivité = 2

Figure 4.9:

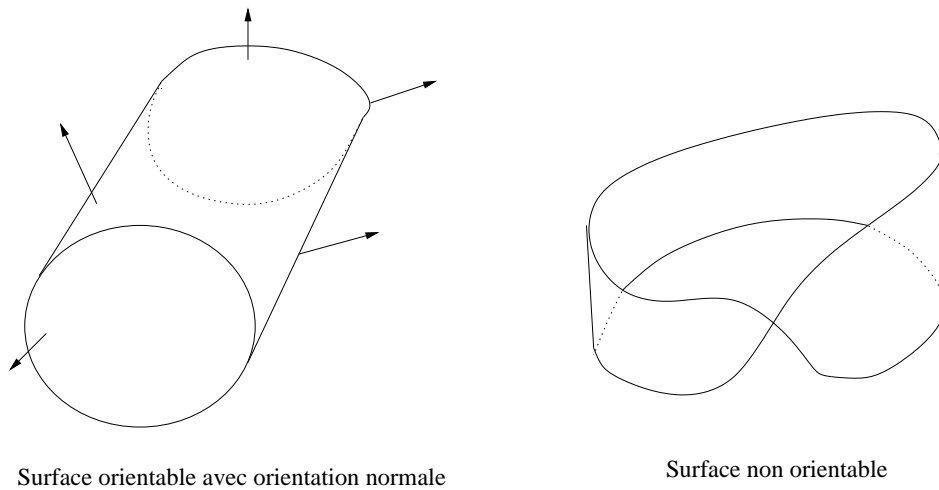


Figure 4.10:

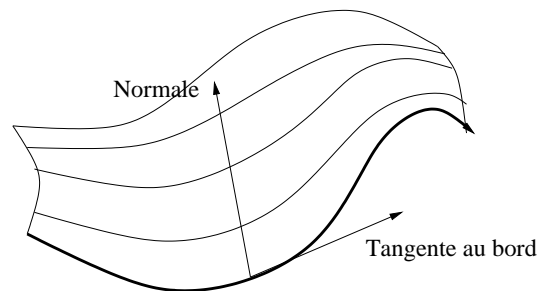


Figure 4.11:

**Définition 6** Soit  $X$  une surface régulière dans  $\mathbf{R}^3$ . Une orientation normale de  $X$  en  $p$  consiste à choisir l'un des deux vecteurs unitaire orthogonaux au plan tangent  $T_p X$ . Une orientation normale de  $X$  est un choix continu d'orientation en chaque point de  $X$ .

**Remarque.** Chaque paramétrisation  $X : D \rightarrow \mathbf{R}^3$  de  $X$  détermine une orientation normale de la surface. Le vecteur unitaire normal correspondant est

$$\frac{\frac{\partial X}{\partial u} \wedge \frac{\partial X}{\partial v}}{\left\| \frac{\partial X}{\partial u} \wedge \frac{\partial X}{\partial v} \right\|}.$$

Comme les surfaces viennent la plupart du temps avec une paramétrisation, l'orientation normale est sous-entendue.

**Remarque.** Néanmoins, certaines surfaces ne sont pas orientables (voir figure). Il faut au moins deux paramétrisations locales pour couvrir l'ensemble de la surface.

**Définition 7** Une face induit une orientation sur chacune de ses arêtes : en marchant sur la surface le long de l'arête, avec la normale qui va des pieds à la tête, on doit garder la face sur sa gauche.

### 4.2.7 Faces gauches et BRep

**Définition 8** Une face (face)  $F$  est la donnée d'une surface régulière  $X : D \rightarrow \mathbf{R}^3$  normalement orientée et d'une face plane  $f \subset D$ . Une arête (resp. un sommet) de  $F$  est la donnée d'une arête (resp. d'un sommet) de  $f$  et de l'application  $X : D \rightarrow \mathbf{R}^3$ .

**Définition 9** Une BRep est la donnée

- d'un nombre fini de faces ;
- pour chaque sommet, de la liste des arêtes auxquelles il appartient ;
- pour chaque arête, de la liste des faces auxquelles elle appartient.

On fait les hypothèses suivantes

- l'intersection de deux faces distinctes est une réunion d'arêtes ;
- l'intersection de deux arêtes distinctes consiste en un ou deux sommets.

Un solide est un compact propre bordé par une BRep.

Une BRep est dite valide si elle borde un solide.

**Remarque.** Les listes peuvent sembler redondantes : mathématiquement, une arête est ou n'est pas contenue dans le bord d'une face. En fait, dire si deux arêtes appartenant à des faces distinctes coïncident pose un problème pratique. En effet, une arête se calcule en composant une paramétrisation de courbe et une paramétrisation de surface. C'est donc une grandeur numérique (suite de points à coordonnées flottantes), entachée d'erreur. Décider si deux telles grandeurs coïncident dépend d'un choix de tolérance. Chaque logiciel de CAO ayant sa tolérance, il est plus prudent de spécifier qui est égal à qui.

CATIA fait des hypothèses moins restrictives sur les BRep : deux faces peuvent s'intersecter suivant des portions d'arêtes. En revanche, il exige des faces *connexes* délimitées par des boucles *sans points doubles*.

**Théorème 5** Soit  $B$  une BRep dont les faces  $F_1, \dots, F_k$  sont connexes. Notons  $A$  le nombre d'arêtes et  $S$  le nombre de sommets de  $G$ . Alors

$$\chi(B) = S - A + \sum_{i=1}^k \chi(\text{int}(F_i)).$$

**Preuve.** D'après le paragraphe précédent, le bord de chaque face  $F_i$  est une réunion de boucles. On dessine, à l'intérieur de  $F_i$ , des boucles proches du bord. On obtient ainsi une face plus petite  $F'_i$ , voir figure. Ces petites faces sont deux à deux disjointes.

Alors  $B$  s'écrit comme réunion des  $F'_i$  et d'un épaississement  $G'$  du graphe  $G$ . Comme  $G' \cap F'_i$  est une réunion de boucles, sa caractéristique d'Euler est nulle. Par conséquent

$$\chi(B) = \chi(G') + \sum_{i=1}^k \chi(F'_i).$$

Si  $F_i$  est bordé par  $b$  boucles,  $F'_i$  est bordée par  $b$  boucles disjointes et sans points doubles, donc  $\chi(F'_i) = \chi(\text{int}(F_i))$  d'après le théorème 4. Pour voir que  $\chi(G') = \chi(G) = S - A$ , on peut vérifier que  $G'$  est homotope à  $G$ . Voici un argument plus direct. On découpe  $G'$  en pièces simples : épaississements de sommets (homéomorphes à des polygones convexes) et épaississements d'arêtes (homéomorphes à des rectangles), de caractéristique d'Euler égale à 1. L'intersection d'un épaississement d'arête et d'un épaississement de sommet est un arc homéomorphe à un intervalle, donc sa caractéristique d'Euler vaut 1. Autrement dit, le calcul de la caractéristique d'Euler de  $G'$



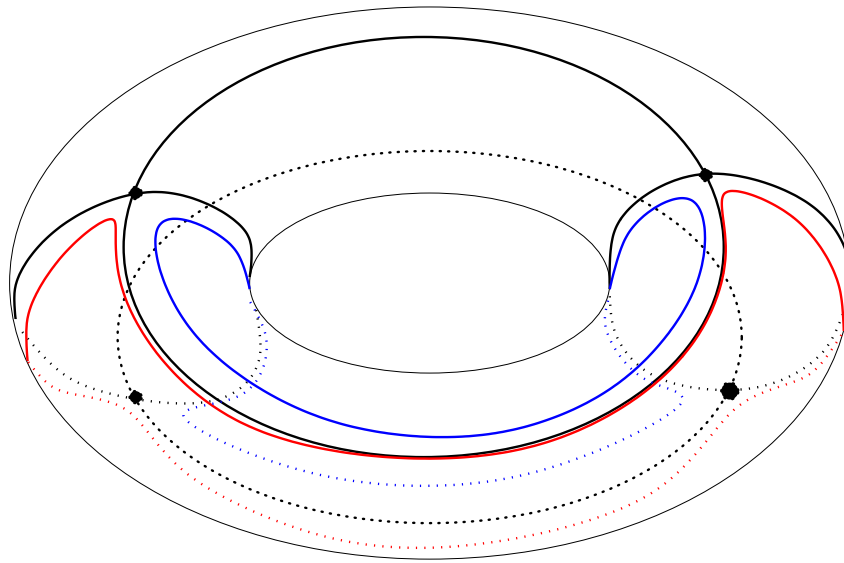


Figure 4.12:

(et celle de  $G$ ) au moyen de la formule d'addition font intervenir les mêmes ingrédients et la même combinatoire, donc donne le même résultat. ■

**Remarque.** Soit  $P$  un polyèdre convexe de  $\mathbf{R}^3$ . Alors chaque face du bord de  $P$  est un polygone convexe, donc la caractéristique d'Euler de son intérieur vaut 1. Par conséquent

$$S - A + F = \chi(\partial P).$$

où  $S$  est le nombre de sommets,  $A$  le nombre d'arêtes et  $F$  le nombre de faces de  $P$ . Enfin,  $\chi(\partial P)$  ne dépend pas de  $P$ , car les bords de deux polyèdres convexes quelconques sont homéomorphes (utiliser la projection radiale d'une surface sur l'autre). On peut donc faire le calcul pour un tétraèdre. On trouve que  $\chi(P) = 2$ . L'identité

$$S - A + F = 2$$

est due à L. Euler. Le théorème ci-dessus est une généralisation qui englobe le cas où les faces ne sont pas homéomorphes à des polyèdres convexes.

**Exercice 1** Soit  $K$  le solide obtenu en perçant un trou de section carrée à travers un parallélépipède. Calculer la caractéristique d'Euler de  $\partial K$ .

#### 4.2.8 Condition de validité d'une BRep plane

La question de savoir si une BRep est *valide*, i.e. borde effectivement un solide, sera résolue au paragraphe suivant. On commence par examiner le problème correspondant en 2 dimensions.

On remarque qu'une orientation normale pour une courbe plane équivaut à une orientation de la courbe. En effet, la tangente s'obtient en tournant la normale de  $90^0$  dans le sens trigonométrique. Orienter la normale, c'est orienter la tangente, voir figure.

Ce choix est cohérent avec la convention de la matière à gauche. Pour définir une BRep plane, il est commode d'orienter les tangentes plutôt que les normales.

**Définition 10** Une BRep plane est un graphe orienté.

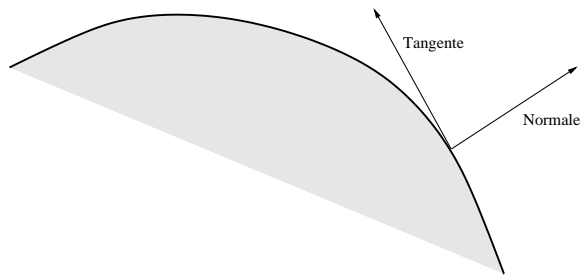


Figure 4.13:

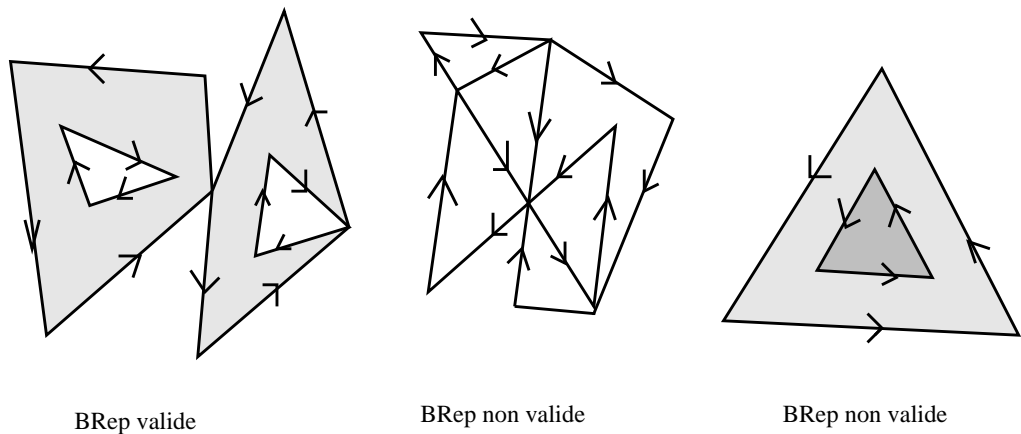


Figure 4.14:

**Théorème 6** Une BRep plane connexe est valide (autrement dit, un graphe orienté borde un fermé propre du plan conformément à la règle de la matière à gauche) si et seulement si, en chaque sommet, il arrive un nombre pair d'arêtes, alternativement rentrantes et sortantes.

**Preuve.** Voir le cours de topologie différentielle. ■

**Remarque.** Attention, dans le théorème, la fermé propre peut être non borné. Par exemple, si le graphe se réduit à un cercle (partagé en deux demi-cercles) parcouru dans le sens des aiguilles d'une montre, la face qu'il borde est le complémentaire du disque.

**Remarque.** Dans le cas des graphes non connexes, il faut une condition supplémentaire, voir figure 4.14.

On trouvera un énoncé dans le cours de topologie différentielle.

#### 4.2.9 Lien

Le lien d'une BRep  $B$  en un sommet  $S$  est un graphe orienté tracé sur une sphère, i.e. une BRep en une dimension de moins. Il sert à donner un critère de validité *locale* : la BRep  $B$  est valide *au voisinage de  $S$*  si et seulement si le lien est une BRep valide, i.e. borde une face sphérique.

**Définition 11** Soit  $B$  une BRep dans  $\mathbf{R}^3$ , et  $S$  un sommet de  $B$ . Le lien (link) de  $B$  en  $S$  est l'intersection de  $B$  avec une petite sphère centrée en  $S$ . On le dilate et le translate pour le voir comme un sous-ensemble de la sphère unité.

Cette définition est ambiguë : quel rayon de sphère prendre ?

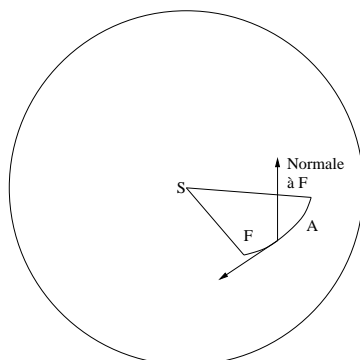


Figure 4.15:

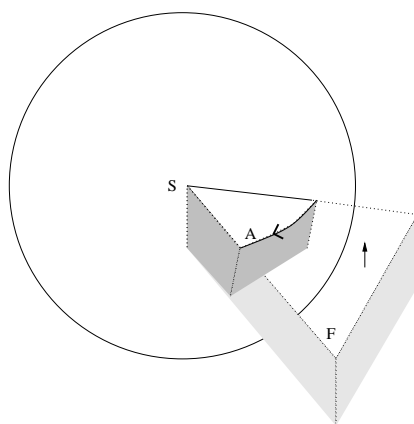


Figure 4.16:

**Proposition 12** *Le lien de  $B$  en  $S$  est un graphe. Il est bien défini à un homéomorphisme de la sphère près. I.e. il existe  $\epsilon > 0$  tel que si  $r < \epsilon$  et  $r' < \epsilon$ , il existe un homéomorphisme  $h$  de  $\partial B(S, r)$  sur  $\partial B(S, r')$  tel que  $h(B \cap \partial B(S, r)) = B \cap \partial B(S, r')$ .*

**Preuve.** Cela résulte d'arguments de transversalité, voir le cours de topologie différentielle. Si  $r$  est assez petit,  $\partial B(S, r)$  est transverse à chacune des faces et arêtes contenant  $S$ , donc l'intersection est un graphe qui dépend continûment de  $r$  et converge vers un graphe formé d'arcs de grands cercles. ■

**Définition 13** *Soit  $B$  une BRep dans  $\mathbf{R}^3$ , et  $S$  un sommet de  $B$ . Le lien  $L$  de  $B$  en  $S$  hérite d'une orientation comme suit. Chaque arête  $A$  est l'intersection d'une face  $F$  avec une petite sphère  $\partial B(S, r)$ . En marchant sur la boule  $B(S, r)$ , on doit parcourir  $A$  en ayant la normale à  $F$  sur sa droite.*

**Remarque.** L'arête  $A$  fait partie du bord de la face  $F \cap B(S, r)$ . Si  $B$  borde un solide  $X$ , l'orientation induite par  $B$  est *opposée* à celle induite par  $F \cap B(S, r)$ . En effet,  $A$  borde la face sphérique  $X \cap \partial B(S, r)$ . L'orientation à mettre sur  $A$  doit être celle induite par  $X \cap \partial B(S, r)$ . Comme  $F \cap B(x, r)$  fait aussi partie du bord de  $X \cap B(S, r)$ ,  $F \cap B(x, r)$  et  $X \cap \partial B(S, r)$  induisent sur  $A$  des orientations opposées.

**Définition 14** *Soit  $B$  une BRep qui borde un fermé propre  $X$ . Soit  $A$  une arête de  $B$ . Soient  $F$  et  $F'$  deux arêtes contenant  $B$ . Soit  $S$  un sommet de  $A$ . On dit que  $F$  et  $F'$  sont adjacentes si les arêtes correspondantes dans le lien de  $B$  en  $S$  sont consécutives.*

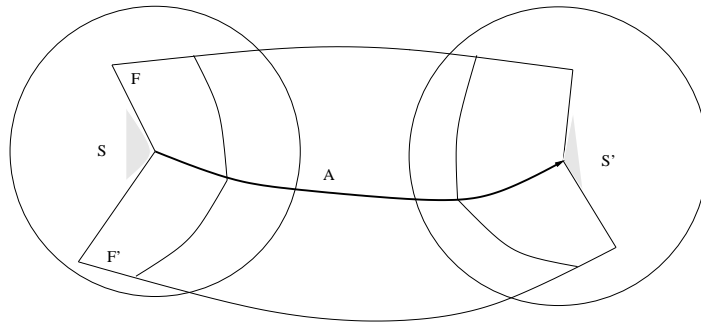


Figure 4.17:

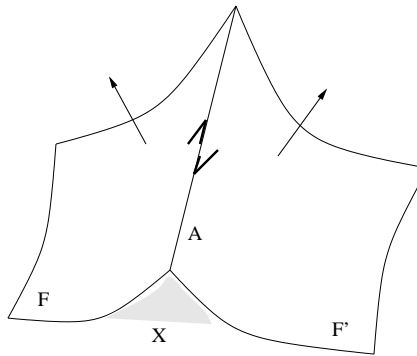


Figure 4.18:

**Remarque.** Ca ne dépend pas du choix de sommet.

**Proposition 15** Soit  $B$  une BRep qui borde un fermé propre  $X$ . Soit  $A$  une arête de  $B$ . Soient  $F$  et  $F'$  deux arêtes contenant  $B$ . Si  $F$  et  $F'$  sont adjacentes, elles induisent sur  $A$  des orientations opposées.

#### 4.2.10 Condition de validité d'une BRep

**Théorème 7** Soit  $B$  une BRep connexe dans  $\mathbf{R}^3$ . On suppose que, pour tout sommet  $S$  de  $B$ , le lien  $\text{lien}(B, S)$  est une BRep valide dans la sphère unité. Alors  $B$  est valide. Inversement, si  $B$  est valide (i.e.  $B$  est le bord d'un fermé propre), alors chaque lien est une BRep valide dans la sphère unité.

**Preuve.** Voir le cours de topologie différentielle. ■

**Corollaire 16** Soit  $B$  une BRep connexe dans  $\mathbf{R}^3$ . On suppose que chaque lien de  $B$  en un sommet est à nouveau connexe. Alors  $B$  est valide si et seulement si, pour chaque arête  $A$  de  $B$ , les faces contenant  $A$  sont en nombre pair et induisent sur  $A$  des orientations alternées lorsqu'on tourne autour de  $A$ .

**Preuve.** Par définition, chaque arête  $A$  contenant un sommet  $S$  coupe une petite sphère de centre  $S$  en un sommet  $a$  du lien. Quand on coupe les faces contenant  $A$  avec la petite sphère, on obtient les arêtes du lien contenant  $a$ . Une face induisant sur  $A$  l'orientation rentrant en  $S$  correspond à une arête du lien rentrant en  $a$ . Le lien est valide au voisinage de  $a$  si et seulement si,

quand on tourne autour de  $a$ , on rencontre alternativement des arêtes rentrantes et sortantes. Par conséquent, en tournant autour de  $A$ , on rencontre des faces qui induisent sur  $A$  des orientations alternativement rentrante et sortante de  $S$ . ■

**Remarque.** Soit  $X$  un solide. Le lien du bord de  $S$  en un sommet  $S$  est non connexe si et seulement si, au voisinage de  $S$ ,  $X$  est formé de deux solides qui se touchent en  $S$  seulement. Ce n'est pas une situation courante.

**Remarque.** Le fermé propre fourni par le théorème peut être non borné (e.g. complémentaire d'une sphère). Au plus une des composantes connexes de son intérieur est non bornée. Les adhérences des autres sont des solides.

**Définition 17** Soit  $B$  une BRep. Les composantes connexes de  $B$  s'appellent les coquilles (**shells**).

**Remarque.** Soit  $B$  une BRep *localement valide*, i.e. satisfaisant l'hypothèse du théorème. Alors chacune des coquilles  $C_i$  de  $B$  borde un fermé propre  $X_i$ . La question de savoir si la BRep  $B$  est valide se ramène à une compatibilité entre les orientations des  $C_i$ . Il suffit de trouver une ligne polygonale qui coupe chacune des coquilles, telle que lorsqu'on la parcourt, on entre et sorte alternativement.

### 4.2.11 Variétés

Les logiciels de CAO ne manipulent pas des solides dans la généralité décrite aux paragraphes précédents. Par exemple, un objet réel n'est pratiquement jamais formé de deux solides partageant exactement une arête ou un sommet. Toutefois, de tels objets virtuels peuvent s'avérer nécessaires lors d'étapes intermédiaires, c'est pourquoi une certaine généralité est utile.

Dans un objet réel, l'intérieur de l'objet est *connexe* au voisinage de chaque point. Cela se traduit par les propriétés suivantes

1. connexité au voisinage d'un sommet  $S$  : le lien  $\text{lien}(B, S)$  partage la sphère en exactement deux composantes connexes ;
2. connexité au voisinage d'une arête  $A$  ayant pour sommet  $S$  : le lien en  $a$  de  $\text{lien}(B, S)$  partage le cercle en exactement deux composantes connexes.

La seconde condition signifie qu'en chaque sommet  $a$  de  $\text{lien}(B, S)$ , il arrive une unique arête et il repart une unique arête. La première entraîne que  $\text{lien}(B, S)$  est connexe, et donc que  $\text{lien}(B, S)$  est une boucle.

**Définition 18** Soit  $B$  une BRep. Un sommet  $S$  est un point 2-séparant (**manifold point**) si  $\text{lien}(B, S)$  est une boucle. On dit qu'une BRep est eulérienne, ou bien une variété (**manifold**) si chaque sommet est 2-séparant.

**Théorème 8** Une coquille eulérienne (i.e. qui est une variété) est valide. Elle sépare l'espace en exactement deux composantes connexes, dont une est bornée.

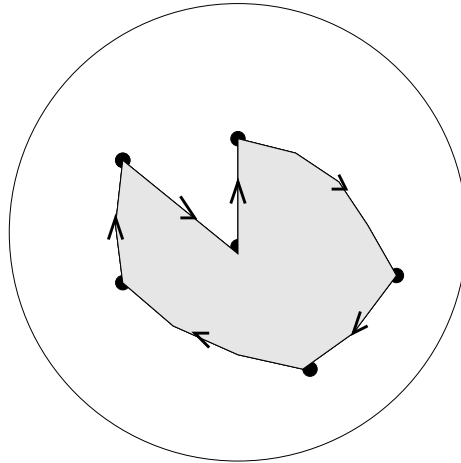
**Théorème 9** Classification des surfaces orientables compactes. Soit  $B$  une BRep de faces  $F_1, \dots, F_k$ . On suppose que

1.  $B$  est une variété (i.e. eulérienne) ;
2.  $B$  est connexe.

Alors sa caractéristique d'Euler-Poincaré

$$\chi(B) = S - A + \sum_{i=1}^k \chi(F_i)$$

est paire, de la forme  $\chi(B) = 2 - 2g$ . L'entier  $g$  s'appelle le genre (**genus**) de  $B$ . C'est le nombre maximum de boucles deux à deux disjointes qu'on peut tracer sur  $B$  sans la déconnecter.



Lien d'un solide en un point 2-séparant

Figure 4.19:

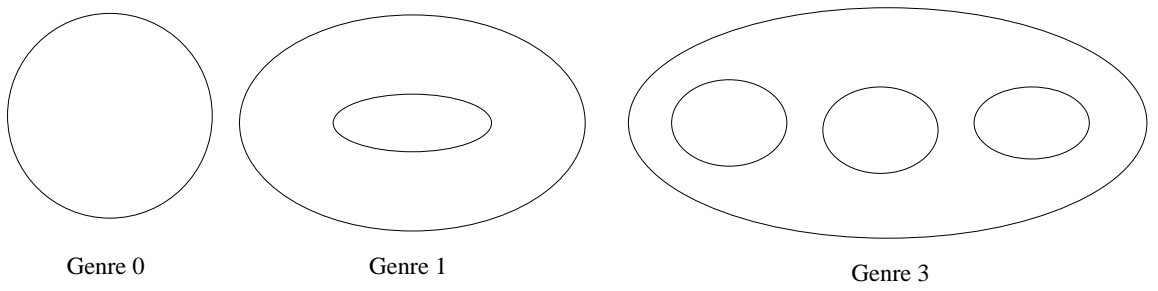


Figure 4.20:

**Remarque.** Ce théorème donne une condition nécessaire de validité pour une BRep dont on pense raisonnablement que c'est une variété.

## 4.3 Les classes C++ d'ACIS

Toutes les classes décrivant des morceaux constitutifs d'objets virtuels sont dérivées de ENTITY.

### 4.3.1 Classes mathématiques

Comme les classes mathématiques ne figurent pas par défaut dans C++, ACIS comporte les classes suivantes.

- `position`, `vector`, `unit_vector`, `matrix`, `transf`, `interval`, `box`, `par_pos`, `par_dir`, `parameter`.
- `angle_between`, `are_parallel`, `are_perpendicular`, `distance_to_plane`, ...

### 4.3.2 Classes géométriques

ACIS manipule divers types de courbes et surfaces.

- Des courbes et surfaces analytiques : droite (STRAIGHT), CIRCLE, ELLIPSE, PLANE, CONE, TORUS, SPHERE.
- Des courbes et surfaces *interpolées* (INTCURVE) : ce sont d'une part les courbes splines, d'autre part les courbes qui sont le résultat de calculs (intersection de surfaces, silhouettes et autres solutions de problèmes géométriques).
- Des courbes et surfaces composites (COMPCURVE) : ce sont des listes ordonnées de courbes analytiques et/ou interpolées. Cela comprend les surfaces maillées disponibles dans le husk optionnel *Mesh*.
- Des courbes dans l'espace des paramètres : les PCURVE.

Mathématiquement, une *pcurve* est la donnée d'une surface paramétrée par un domaine plan  $D$  et d'une courbe dans  $D$ .

D'ENTITY dérivent les classes géométriques abstraites APOINT, CURVE, SURFACE et PCURVE. De la classe CURVE (resp. SURFACE) dérivent les classes géométriques spécifiques comme ELLIPSE, STRAIGHT ou SPHERE, PLANE, SPLINE.

### 4.3.3 Classes topologiques

- BODY est l'entité topologique de plus haut niveau. Il est composé de lumps qui sont ses composantes connexes. Il référence l'un de ses lumps.
- LUMP désigne une région connexe et bornée de l'espace, bordée par des shells. Il référence l'un de ses shells et pointe vers son successeur dans la liste des lump constituant le body qui le contient.
- SHELL est un ensemble de faces (et/ou de wires) connexe, non nécessairement eulérien. Il référence une de ses faces, et pointe vers son successeur dans la liste des shells constituant le lump qui le contient.
- SUBSHELL est une partie d'un shell. Ce n'est pas obligatoire.
- FACE est l'analogue 2D d'un body. C'est une portion d'une surface géométrique bordée par des loops, pourvue des attributs suivants :

- orientation par rapport à la géométrie support : `FORWARD` ou `REVERSED`,
- multiplicité (`sidedness`) : la face borde un solide (par défaut) : `SINGLE-SIDED`, ou représente une coque infiniment fine : `DOUBLE-SIDED`, `BOTH_OUTSIDE`, ou représente une fissure à l'intérieur d'un solide : `DOUBLE-SIDED`, `BOTH_INSIDE`.

Une face pointe sur son support qui est une surface, référence un de ses loops et pointe vers son successeur dans la liste des faces constituant le shell qui la contient.

- `LOOP` désigne une composante connexe de la frontière d'une face. Il est composée de coedges liés circulairement. Le loop référence un de ses coedge, ainsi que son successeur dans la liste des loop de la face.
- `COEDGE` enregistre l'occurrence d'un edge dans une boucle (loop) d'une face. Chaque coedge pointe vers son successeur et son prédécesseur dans le loop. Noter que l'ordre circulaire des coedges d'un loop n'est pas arbitraire.
- `EDGE` décrit la topologie associée à une courbe. Il possède deux extrémités qui sont des vertex, et un support qui est une curve ou une pcurve. Il est pourvu des attributs suivants :
  - orientation par rapport à la géométrie support : `FORWARD` ou `REVERSED`,
  - agencement des coedges qui le contiennent : s'il y en plus de 2, ils sont liés circulairement.
- `VERTEX` désigne un sommet, qui appartient à des edges. Il pointe vers un apoint.
- `WIRE` est un ensemble connexe d'edges qui ne sont pas reliés à des faces et qui n'enferment aucun volume. Un shell (et par conséquent un lump et un body) peut n'être constitué que de wires.

#### 4.3.4 Tolérances

- `resabs` : Résolution absolue ; c'est la tolérance de confusion entre deux points :  $10^{-6}m$ .
- `resnor` : Résolution normalisée=`resabs`/plus grande valeur=  $10^{-10}$ .
- `resfit` : C'est la tolérance de confusion entre deux courbes ou surfaces :  $10^{-3}m$ .
- Pas d'unité absolue, mais pour les fichiers de transfert, le `mm` est choisi.

#### 4.3.5 Bibliographie sur ACIS

Voir l'aide en ligne. Elle comporte pour chaque classe un `.h` accompagné de commentaires, et des articles généraux (`discussion topics`).

Deux livres.

### 4.4 Modèle procédural (CSG)

L'utilisateur qui construit pas à pas un solide part de primitives simples (plans, sphères, cylindres), de solides obtenus par des constructions simples à partir de courbes (extrusions, surfaces de révolution, de balayage), et les combine par intersection, réunion, différences ensemblistes. Son point de vue, celui de la **Constructive Solid Geometry**) est donc totalement différent de la représentation interne du solide.

#### 4.4.1 Primitives

- Primitives simples : parallélépipède (`cuboid`), sphère, cylindre, cône.
- Primitives à contour : extrusion, pyramide, solide de révolution, tube.
- Primitives complexes : solides délimités par un ensemble de faces.



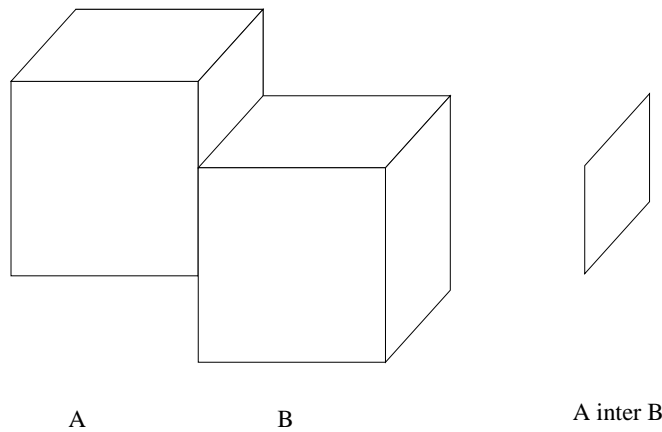


Figure 4.21:

#### 4.4.2 Opérations booléennes régularisées

Un solide est en particulier un fermé propre. Or le complémentaire d'un fermé propre est ouvert, l'intersection de fermés propres n'est pas un fermé propre en général (voir figure).

**Définition 19** Soient  $X, Y$  des fermés propres de  $\mathbf{R}^n$ . On note  $X^*$  l'adhérence du complémentaire de  $X$ . On note  $X \cap^* Y$  l'adhérence de l'intérieur de  $X \cap Y$ . Ces deux opérations engendrent les opérations booléennes régularisées.

**Exemple.** La *différence régularisée* de  $X$  et de  $Y$  est

$$X -^* Y = X \cap^* Y^*.$$

#### 4.4.3 Arbre CSG

La suite d'opérations

- Créer la primitive  $A$ .
- Créer la primitive  $B$ .
- Créer la primitive  $C$ .
- Union des solides  $B$  et  $C$ .
- Saisie du rayon  $r_1$ .
- Arrondi de rayon  $r_1$  sur l'arête d'intersection de  $B$  et  $C$ .
- Saisie du rayon  $r_2$ .
- Arrondi de rayon  $r_2$  sur l'arête d'intersection de  $B$  et  $A$ .

se traduit par le graphe ci-dessous appelé *arbre CSG*.

Ce graphe peut-être résumé dans la *formule postfixée*

$$A B C \cup^* r_1 \text{ Arrondi} -^* r_2 \text{ Arrondi},$$

qui se lit de droite à gauche comme suit : les arguments de chaque opérateur binaire ( $\cup^*$ ,  $-^*$ , Arrondi) sont les solides décrits par les deux premières formules rencontrées à gauche de l'opérateur.

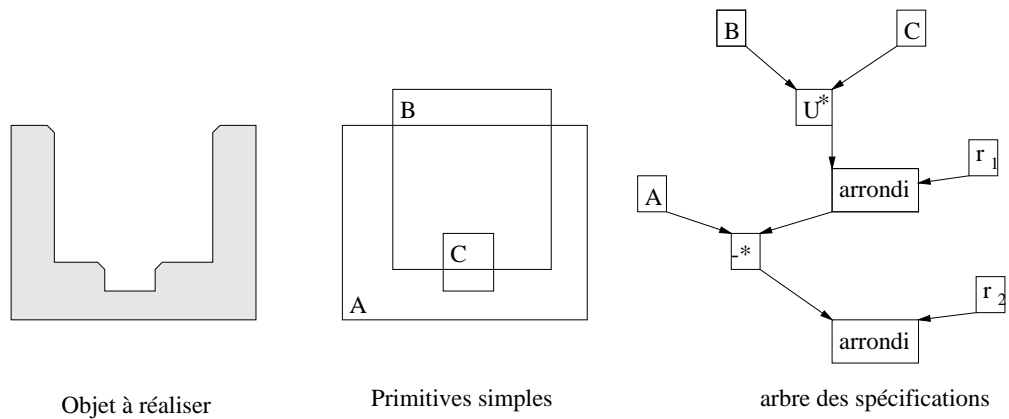


Figure 4.22:

#### 4.4.4 Mise à jour

A chaque modification du solide, il faut modifier à la fois la procédure (arbre CSG) et le modèle BRep (résultat de la procédure) : dans l'ordre, on modifie d'abord la procédure, ensuite le modèle BRep, puis on effectue la mise à jour.

Comme les modifications sont très fréquentes, la mise à jour doit être optimisée.

Les solutions adoptées se regroupent en trois familles.

- Mise à jour par re-jeu (**replay**) de toute la procédure. Cela permet de recréer le solide à partir de sa seule description procédurale, d'où économie de mémoire. Mais le coût de la modification, quelle qu'elle soit, est proportionnelle à la complexité globale du solide.
- Mise à jour par rejeu de toutes opérations postérieures à la modification. Cela nécessite de stocker les solides intermédiaires. Le coût dépend de l'équilibre de l'arbre CSG : si l'arbre est de profondeur  $n$ , avec  $n$  petit, une modification n'affectera qu'un chemin de  $n$  noeuds, bien plus petit que l'arbre entier. La conception de l'arbre dépend des performances que l'on en attend : les opérations susceptibles d'être modifiées doivent si possible figurer près de la racine. Voir les exercices.
- Mise à jour par rejeu des opérations voisines (dans l'espace) de la modification. Du bon usage des boîtes d'encrembrement.

#### Exemple.

Le solide  $S_0$  est décrit par la formule postfixée

$$A E - * F - * B C \cup * D \cup * - *$$

et par l'arbre

On doit modifier  $F$  en  $F'$ . On connaît une boîte d'encrembrement, i.e. une boîte  $V$  qui contient  $F$  et  $F'$ .

#### Algorithme

- Séparation du solide. On décompose

$$S_0 = S' \cup * S'' \text{ où } S' = S_0 \cap * V \text{ et } S'' = S_0 - * V.$$

L'arbre CSG de  $S'$  est

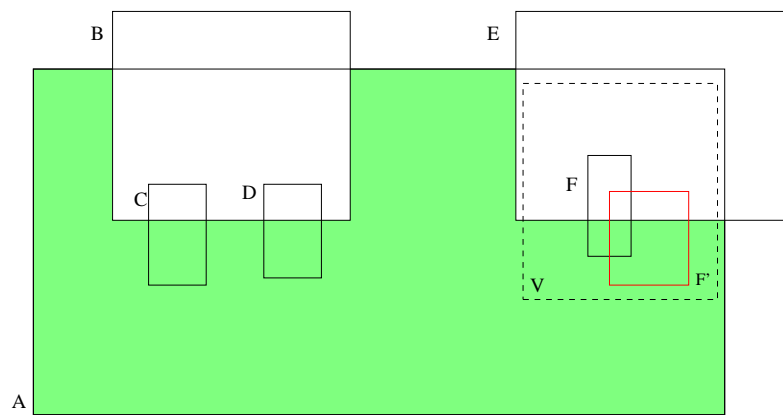


Figure 4.23:

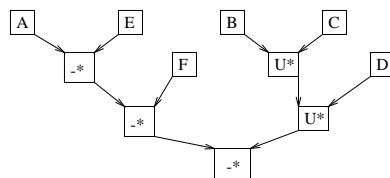


Figure 4.24:

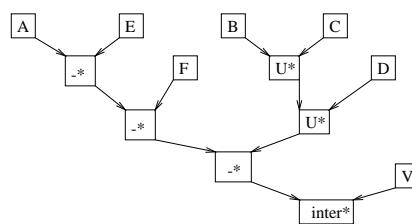


Figure 4.25:

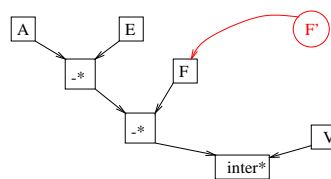


Figure 4.26:

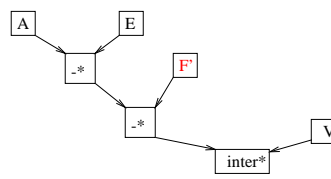


Figure 4.27:

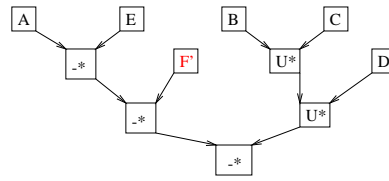


Figure 4.28:

- Réduction de l'arbre CSG de  $S'$ . On élimine les noeuds triviaux (opérations dont le résultat est déjà connu : arguments vides).
- Évaluation. On évalue le solide  $S'_{mod}$  obtenu en remplaçant  $F$  par  $F'$ .
- Synthèse. On produit la BRep bord du solide  $S_{mod} = S'_{mod} \cup S''$ .

Le gain est dû au fait que le re-jeu ne s'applique que dans le (petit) arbre CSG de  $S'$ .

## Chapter 5

# Opérations booléennes

Dans ce chapitre, on décrit, dans un cas simple, l'algorithme qui calcule la BRep de l'intersection régularisée de deux solides donnés par leur BRep. L'algorithme n'est pas conceptuellement compliqué, c'est son implémentation qui l'est, en raison

- de la multitude de cas particuliers : dans tous les systèmes existants, il reste des cas particuliers non traités ;
- des difficultés liées à la prise de décisions fondée sur des données numériques imprécises ;
- de la difficulté mathématique de la détermination de l'intersection de deux surfaces gauches (voir chapitre 6).

Deux références : [PGMR], [H].

### 5.1 Description de l'algorithme : version 2D

Commençons par une version 2D : l'intersection régularisée d'un polygone plan  $X$  avec un demi-plan  $H$  bordé par une droite  $D$ .

La méthode consiste à construire les listes de composants par dimension croissante : sommets, arêtes, faces.

- **Mise à jour de la liste des sommets.**
  - Positionner les sommets de  $X$  par rapport au demi-plan (i.e. les marquer d'un +, - ou 0 suivant qu'ils sont extérieurs, intérieurs ou sur le bord de  $H$ ).
  - Qualifier les arêtes (i.e. les marquer d'un ++, +-, +0, -0, --, 00) suivant les attributs de leurs sommets.
  - Pour chaque arête marquée +-, calculer son intersection avec  $D$  et l'ajouter, marquée d'un \*, à la liste des sommets (figure 5.1).
  - Supprimer les sommets marqués +.
- **Mise à jour de la liste des arêtes.**
  - **Détermination des arêtes coupantes.** Soit  $P_1, \dots, P_k$  la liste ordonnée des sommets se trouvant sur la droite  $D$ . Il s'agit de déterminer lesquels parmi les segments  $[P_i, P_{i+1}]$  sont des arêtes de l'intersection nettoyée  $X \cap^* H$ .
    - \* **Première méthode (celle de CATIA).** Garder l'arête  $[P_i, P_{i+1}]$  si son milieu est un point intérieur de  $X$ . **Inconvénient** : il faut savoir détecter l'intérieur de  $X$ .

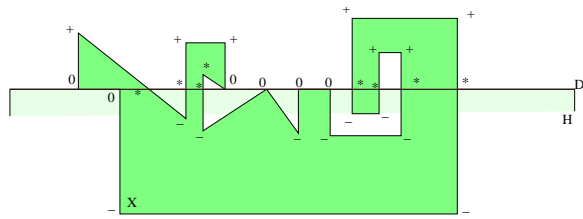
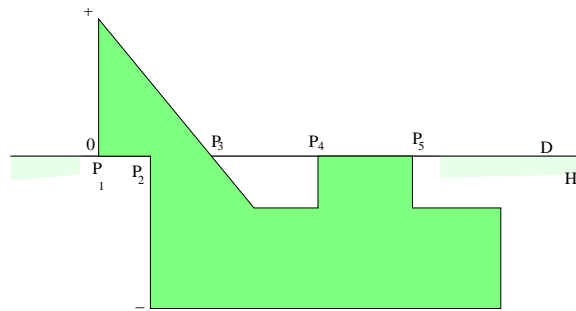


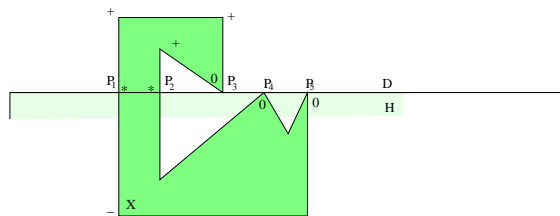
Figure 5.1: Mise à jour de la liste des sommets



L'arête  $P_2P_3$  est coupante, l'arête  $P_4P_5$  est retenue, l'arête  $P_1P_2$  rejetée

Figure 5.2:

- \* **Deuxième méthode (plus économique).** Garder l'arête  $[P_i, P_{i+1}]$  si elle entre dans un secteur délimité par une arête entrant en  $P_i$  et son successeur.
- **Cas particulier des arêtes préexistantes de  $X$  contenues dans  $D$ .** Si  $[P_i, P_{i+1}]$  est une arête de  $X$  contenue dans  $D$ , la retenir si la normale sortante de  $X$  y coïncide avec celle de  $H$  (figure 5.2).
- **Nettoyage.** Il se peut que des sommets se retrouvent isolés. Les supprimer (figure 5.3).
- **Relimitation des arêtes préexistantes.** Chaque arête  $+ -$  de  $X$  est remplacée par une arête  $* -$ .
- **Mise à jour du successeur.** Une arête relimitée en  $-*$  (resp. en  $* -$ ) a pour successeur une arête contenue dans  $D$  (resp. ne change pas de successeur).
- **Mise à jour de la liste des boucles.** Les boucles sont les cycles de la permutation des arêtes ainsi obtenue. Parcourir la liste des boucles de successeur en successeur jusqu'à ce



Le sommet isolé  $P_3$  est éliminé,  $P_2$  est retenu

Figure 5.3:

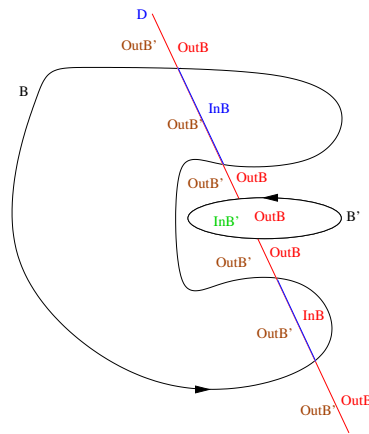


Figure 5.4:

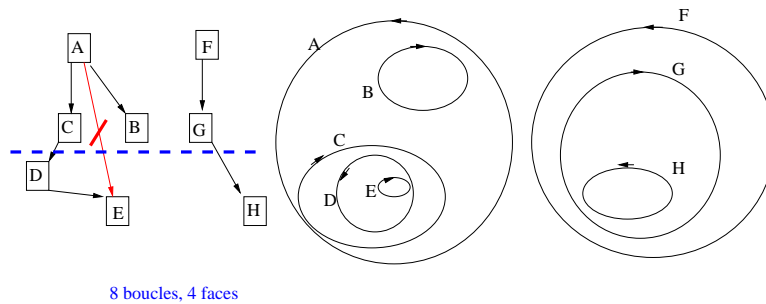


Figure 5.5:

qu'on retrouve l'arête initiale. Recommencer avec la première arête qui n'a pas été déjà rencontrée.

- **Mise à jour de la liste des faces.** Il s'agit de regrouper les boucles par faces connexes.
  - **Test d'inclusion de boucles.** Il s'agit de décider si deux boucles  $B$  et  $B'$  peuvent border une face, et laquelle contient l'autre. Pour cela, tracer une droite orientée  $\Delta$  qui coupe  $B$  et  $B'$ . Affecter à chacun des intervalles délimités sur  $\Delta$  par  $\Delta \cap B$  un type  $InB$  ou  $OutB$ , suivant qu'aux extrémités, l'intervalle est rentrant ou sortant de  $B$ . De même, affecter un type  $InB'$  ou  $OutB'$ . On obtient ainsi un double type pour chacun des intervalles délimités sur  $\Delta$  par  $\Delta \cap (B \cup B')$ . Si l'un des intervalles porte le double type  $(InB, InB')$ , c'est que  $B \cup B'$  borde une face connexe. Si le premier point d'intersection de  $\Delta$  avec  $B \cup B'$  est sur  $B$ , alors  $B$  contient  $B'$ . Sinon, c'est  $B'$  qui contient  $B$ .
  - **Construction de la forêt des boucles.**
    - \* Construire un graphe orienté dont les sommets sont les boucles et une arête orientée relie  $B$  à  $B'$  si  $B$  contient  $B'$ .
    - \* Eliminer les raccourcis (voir figure 5.5).
    - \* Couper aux profondeurs 2, 4,...
  - **Construction de la liste des faces, et pour chaque face, de la liste de ses boucles.** Chaque arbre constitutif de la forêt obtenue décrit une face, sa boucle extérieure et ses boucles intérieures (figure 5.5).

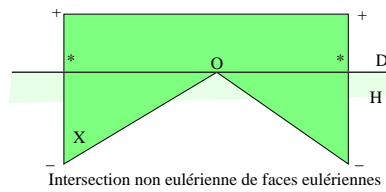


Figure 5.6:

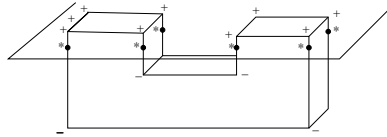


Figure 5.7: Mise à jour de la liste des sommets

- **Cas particulier.** Si la droite  $D$  ne coupe pas le bord de  $X$ , il faut décider si  $X \subset H$  ou non. Pour cela, appliquer le test d'inclusion de boucles.

**Remarque.** Même si  $X$  est eulérien, il se peut que  $X \cap^* H$  ne le soit pas, voir figure 5.6.

## 5.2 Description de l'algorithme en 3D

Cette fois-ci,  $X$  est un polyèdre dans l'espace et  $H$  un demi-espace bordé par un plan  $P$ .

Comme en 2D, on construit les listes d'éléments par dimension croissante

- **Mise à jour de la liste des sommets.** Marquage des sommets, élimination des sommets marqués  $+$ , calcul des intersections  $*$  des arêtes marquées  $+-$  avec  $P$  (figure 5.7).
- **Mise à jour de la liste des arêtes.** Relimitation des arêtes  $+-$  en arêtes  $*-$ .
- **Mise à jour de la liste des faces.**
  - **Relimitation des faces préexistantes.** A chaque face coupant  $P$  et non contenue dans  $P$ , appliquer l'algorithme 2D.
  - **Détermination des faces de coupe.** L'étape précédente a fourni un lot d'arêtes contenues dans  $P$ , les arêtes coupantes des faces relimitées, qui s'ajoutent aux arêtes préexistantes de  $X$  contenues dans  $P$ . Ne retenir que les arêtes  $a$  telles que quand on se déplace à partir de  $a$  dans la direction de  $P$ , on peut rentrer dans l'intérieur de  $X$ . Cela se lit sur les directions des normales aux faces de  $X$  contenant  $a$  (autrement dit, sur le lien de  $X$  aux sommets de  $a$ ). Voir figure 5.8.
  - **Cas des faces préexistantes contenues dans  $P$ .** Retenir une telle face  $F$  si le long de  $F$ , la normale sortante de  $X$  coïncide avec celle de  $H$ .

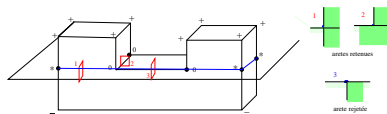


Figure 5.8:



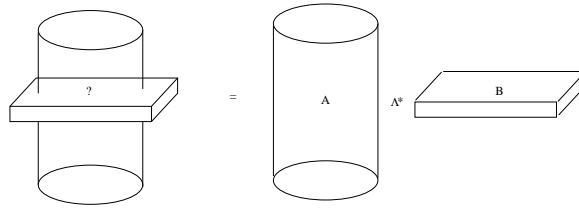


Figure 5.9:

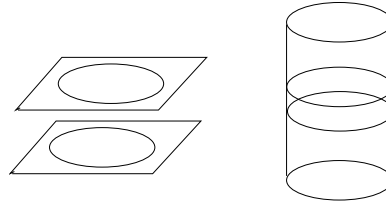


Figure 5.10: Calcul des intersections des faces

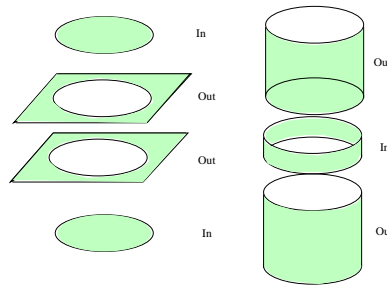
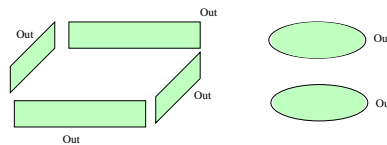
- **Nettoyage.** Supprimer les sommets et arêtes contenus dans  $P$  et isolés, i.e. adjacents à aucune arête (resp. face).
- **Mise à jour de la liste des coquilles, et pour chaque coquille, de la liste de ses faces.**
  - Construire le graphe des faces : il possède un sommet par face, et une arête du graphe correspond à deux faces qui partagent au moins une arête.
  - Les coquilles (composantes connexes du bord du solide) correspondent aux composantes connexes du graphe des faces.
- **Mise à jour de la liste des lumps.**
  - **Test d'inclusion de coquilles.** Identique au test d'inclusion des boucles en 2D.
  - **Construction de la forêt des coquilles.** Idem.
  - **Mise à jour de la liste des lumps, et pour chaque lump, de la liste de ses coquilles.** Idem.
- **Cas particulier.** Si le plan  $P$  ne coupe pas le bord de  $X$ , il faut décider si  $X \subset H$  ou non. Pour cela, appliquer le test d'inclusion de coquilles.

## 5.3 Extension de l'algorithme

### 5.3.1 Cas de deux solides

Soient  $A$  et  $B$  deux solides (e.g. un cylindre et un pavé). Pour simplifier, on suppose que chacun ne comporte qu'une coquille.

1. Déterminer les paires de faces  $f$  de  $A$  et  $g$  de  $B$  qui s'intersectent. S'il n'y en a pas, aller directement en 3.
2. Pour chaque paire  $\{f, g\}$  de faces qui se coupent, construire la coupe de  $B$  avec la surface contenant  $f$ . On obtient les faces de  $A \cap^* B$  contenues dans  $f$ . Transférer les sommets et les arêtes construites sur les faces de  $B$  correspondantes et redélimiter ces faces (figure 5.11).

Figure 5.11: Classification *In/Out* pour l'intersection (faces issues de faces intersectantes)Figure 5.12: Classification *In/Out* pour l'intersection (faces non intersectantes)

3. Pour chaque face de  $A$  qui ne coupe aucune face de  $B$ , faire un test d'inclusion et la classer *In/Out* suivant qu'elle est à l'intérieur ou à l'extérieur de  $B$ . Idem pour les faces de  $B$  qui ne coupent aucune face de  $A$  (figure 5.12).
4. Assembler les faces marquées *In*, supprimer les faces marquées *Out* (figure 5.13).

### 5.3.2 Réunion, différence régularisée

Une modification minimale de l'algorithme permet d'obtenir la réunion ou la différence au lieu de l'intersection.

Pour la réunion, lors de la dernière phase, assembler les faces marquées *Out* et supprimer les faces marquées *In* (figure 5.14).

Pour la différence régularisée  $A -^* B$ , lors de la phase 3, changer la dénomination de toutes les faces sur le bord de  $B$  (figure 5.15), et assembler les faces marquées *In* (figure 5.16).

## 5.4 Problèmes de robustesse

### 5.4.1 Incohérences

Comme on considère chaque face de  $A$  indépendamment dans l'étape 2, des incohérences peuvent arriver pour des faces  $f_1$  et  $f_2$  de  $A$  adjacentes le long d'une arête  $a$  : en coupant ces faces avec une face  $g$  de  $B$ , on crée deux fois le sommet  $S = a \cap g$ , avec des coordonnées légèrement différentes (figure 5.17). Il faut dans le résultat identifier les deux sommets obtenus. On le fera mécaniquement si la distance entre les deux sommets est inférieure à la tolérance fixée. Cette

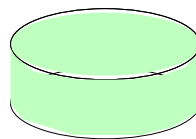


Figure 5.13: Intersection : résultat

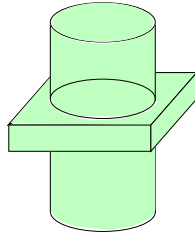


Figure 5.14: Réunion : résultat

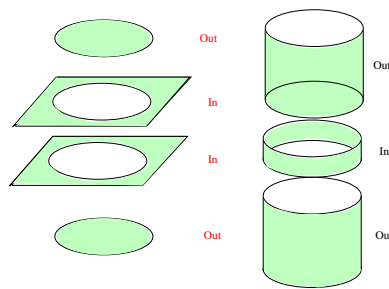


Figure 5.15: Différence : changement de dénomination des faces de  $B$

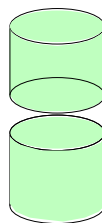


Figure 5.16: Différence régularisée : résultat

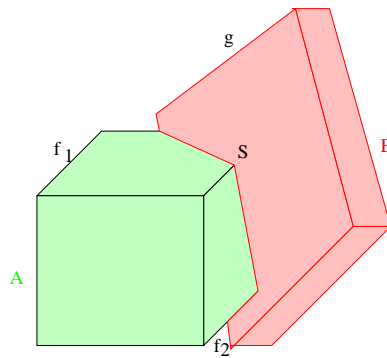


Figure 5.17: Double calcul d'un nouveau sommet

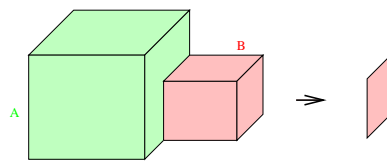


Figure 5.18: Lame parasite

distance peut s'avérer supérieure si les solides  $A$  et  $B$  ont été produits par un logiciel de tolérance plus élevée puis importés, par exemple. Le résultat est un solide présentant une arête ultra-courte, ce qui sera source de difficultés lors d'opérations ultérieures.

### 5.4.2 Zones de confusion

Cela se produit par exemple si la face  $f$  de  $A$  et la face  $g$  de  $B$  sont dans le même plan. On doit déterminer l'intersection  $f \cap g$  en tant que polygones plans, puis décider si  $f \cap g$  est ou non sur le bord de  $A \cap^* B$ . C'est le cas si et seulement si les faces ont la même orientation normale. Sinon, la face, qui fait partie de l'intersection ensembliste  $A \cap B$ , n'est pas dans l'intersection régularisée. On est en présence d'un cas limite, et donc d'instabilité : en cas d'imprécision sur la géométrie de  $A$  et de  $B$ , on risque de créer une lame ultra-fine parasite (figure 5.18).

### 5.4.3 Autres difficultés

Coût CPU : proportionnel à  $NN'$  si  $A$  a  $N$  faces et  $B$  en a  $N'$ . Pour diminuer ce coût, on utilise les boîtes d'encombrement, voir le cours d'algorithmique.

L'importation de modèles d'un système vers un autre nécessite toujours une interface, car chaque système a son choix de restrictions topologiques. Par exemple, ACIS admet la modélisation d'une boîte de conserve cylindriques avec seulement 3 faces et deux arêtes, ce que CATIA n'admet pas.

## 5.5 Terminologie de ACIS

- `api_slice` crée le graphe des arêtes coupantes.
- `api_imprint` crée les faces relimitées sur les body de départ.
- `api_imprint_stitch` crée le body non eulérien par couture des frontières.

- `api_remove_face` supprime une face du body qui l'incluait.
- `api_intersect`, `api_subtract`, `api_unit` effectuent l'ensemble de l'opération.



# Chapter 6

## Cheminement

### 6.1 Champ d'application

Il s'agit d'une méthode numérique pour résoudre toute équation dont l'ensemble des solutions est une courbe. Elle s'applique notamment aux problèmes suivants.

- Intersection de deux surfaces.
- Lignes de reflet.
- Appuis d'une surface de raccordement.
- Silhouette d'un objet.
- Projection d'une courbe sur une surface.
- Trajet outil pour l'usinage.

La mise en équation de chacun de ces problèmes fait l'objet d'un exercice (voir feuille). On détaille les différents cas d'intersections de surfaces.

### 6.2 Intersection de surfaces

#### 6.2.1 $X_1$ paramétrée, $X_2$ donnée implicitement

Soit  $X_2$  une surface donnée par une équation, i.e.

$$X_2 = \{(x, y, z) \in \mathbf{R}^3 \mid f(x, y, z) = 0\}.$$

Par exemple, un plan  $\{ax + by + cz - d = 0\}$ , une sphère  $\{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - R^2 = 0\}$ , un cylindre d'axe vertical  $\{(x - x_0)^2 + (y - y_0)^2 - R^2 = 0\}$ , un cône d'axe vertical  $\{(x - x_0)^2 + (y - y_0)^2 - (z - z_0)R^2 = 0\}$ .

Soit  $X_1$  une surface donnée par une paramétrisation

$$(u, v) \in D \quad \mapsto s(u, v) \in \mathbf{R}^3.$$

Calculer l'intersection  $X_1 \cap X_2$ , c'est d'abord déterminer l'ensemble  $C$  des  $(u, v) \in D$  tels que  $s(u, v) \in X_2$ , et ensuite calculer  $s(C)$ . Autrement dit, on obtient naturellement une **arête** ou **pcurve** dans  $X_1$  (voir les classes d'ACIS, 4.3).

L'équation à résoudre dans le domaine plan  $D$  est donc

$$F(u, v) = 0,$$

où  $F = f \circ s : D \rightarrow \mathbf{R}$ .

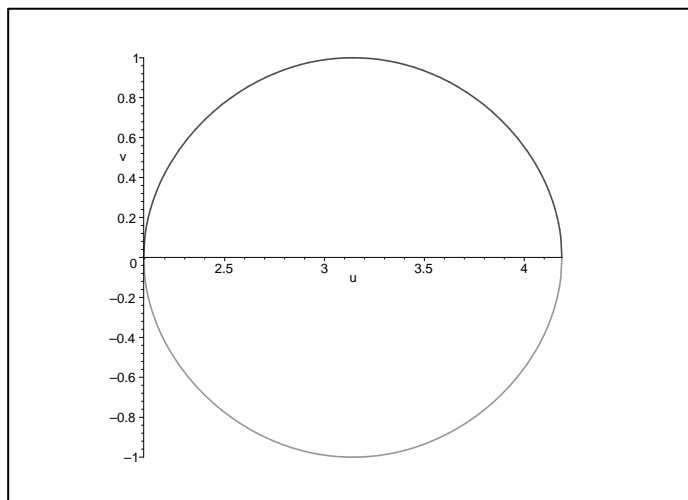


Figure 6.1: La ligne d'intersection dans le plan des paramètres du cylindre

**Exercice 2** Déterminer analytiquement l'intersection du cylindre d'axe vertical passant par  $(1, 0, 0)$  et de rayon 1 avec la sphère unité, en utilisant une paramétrisation du cylindre.

**Solution de l'exercice 2.** Intersection d'un cylindre et d'une sphère.

On paramètre le cylindre par

$$(u, v) \mapsto s(u, v) = (1 + \cos u, \sin u, v)$$

pour  $(u, v) \in [0, 2\pi] \times \mathbf{R}$ .

Alors  $s(u, v)$  est sur la sphère unité si et seulement si

$$(1 + \cos u)^2 + \sin^2 u + v^2 - 1 = 0.$$

Cette équation admet une solution analytique

$$v = \pm \sqrt{1 - (1 + \cos u)^2 - \sin^2 u}$$

Autrement dit, l'ensemble des solutions dans le plan  $(u, v)$  est la réunion des deux courbes paramétrées

$$c_1 : u(t) = t, \quad v(t) = \sqrt{1 - (1 + \cos t)^2 - \sin^2 t}$$

et

$$c_2 : u(t) = t, \quad v(t) = -\sqrt{1 - (1 + \cos t)^2 - \sin^2 t},$$

où  $t \in [\frac{2\pi}{3}, \frac{4\pi}{3}]$  dans les deux cas. ■

Dans cet exemple, on trouve une expression explicite pour l'arête d'intersection. C'est en général impossible (voir exercice 3). Il faut recourir à une résolution approchée, par une méthode numérique.



**Exercice 3** Soit  $c$  la courbe de Bézier de polygone de contrôle

$$P_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} \frac{1}{4} \\ 0 \\ 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{6} \\ 0 \end{pmatrix}, \quad P_3 = \begin{pmatrix} \frac{3}{4} \\ \frac{1}{2} \\ 0 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Calculer  $c(t)$  pour  $t \in [0, 1]$ . Soit  $X$  le cône sur  $c$  de sommet l'origine. Paramétrer la surface  $X$ . La courbe d'intersection de  $X$  avec un cône quadratique a-t-elle toujours une expression analytique ?

**Solution de l'exercice 3.** Intersection d'un carreau de Bézier et d'un cône quadratique.

Le calcul donne  $c(t) = (t, t^2, t^4)$ , d'où la paramétrisation

$$s(u, v) = (uv, uv^2, uv^4), \quad u \in \mathbf{R}, \quad v \in [0, 1]$$

pour le cône. Considérons la famille de quadriques  $Q_{a,b,c,d,e,f}$  d'équations

$$ax^2 + by^2 + cz^2 + dxy + eyz + fzx = 0.$$

Il s'agit de toutes les quadriques qui sont des cônes de sommet l'origine. Alors  $s(u, v) \in Q_{a,b,c,d,e,f}$  si et seulement si

$$u^2(av^2 + bv^4 + cv^8 + dv^3 + ev^6 + fv^5) = 0.$$

L'ensemble des solutions est la réunion du segment  $\{u = 0\}$ , de la droite  $\{v = 0\}$  et des droites  $\{v = v_i\}$ , où les  $v_i$  sont les racines de l'équation  $a + dv + bv^2 + fv^3 + ev^4 + cv^6 = 0$ . Donner une expression explicite de l'ensemble des solutions suppose d'être capable de résoudre l'équation du 6ème degré la plus générale au moyen de quadratures, ce qui est impossible. ■

### 6.2.2 $X_1$ et $X_2$ données par des équations

On suppose les surfaces  $X_1$  et  $X_2$  données implicitement, i.e.

$$X_1 = \{(x, y, z) \in \mathbf{R}^3 \mid f_1(x, y, z) = 0\}, \quad X_2 = \{(x, y, z) \in \mathbf{R}^3 \mid f_2(x, y, z) = 0\}.$$

Pour déterminer l'intersection  $X_1 \cap X_2$ , il faut résoudre le système de 2 équations à 3 inconnues

$$\begin{aligned} f_1(x, y, z) &= 0 \\ f_2(x, y, z) &= 0, \end{aligned}$$

i.e. résoudre  $F(x, y, z) = 0$  où  $F : \mathbf{R}^3 \rightarrow \mathbf{R}^2$  est donnée par

$$F(x, y, z) = (f_1(x, y, z), f_2(x, y, z)).$$

L'ensemble des solutions est directement la courbe d'intersection dans  $\mathbf{R}^3$ .

### 6.2.3 $X_1$ et $X_2$ données par des paramétrisations

On suppose les surfaces  $X_1$  et  $X_2$  données par des paramétrisations

$$(u_1, v_1) \in D_1 \mapsto s_1(u_1, v_1), \quad (u_2, v_2) \in D_2 \mapsto s_2(u_2, v_2).$$

Alors déterminer l'intersection  $X_1 \cap X_2$  revient à trouver les couples  $(u_1, v_1)$  et  $(u_2, v_2)$  tels que

$$s_1(u_1, v_1) = s_2(u_2, v_2),$$

i.e. résoudre

$$F(u_1, v_1, u_2, v_2) = 0$$

où  $F : D_1 \times D_2 \rightarrow \mathbf{R}^3$  est définie par

$$F(u_1, v_1, u_2, v_2) = s_1(u_1, v_1) - s_2(u_2, v_2).$$

Supposons l'ensemble des solutions paramétré par  $t \mapsto (u_1(t), v_1(t), u_2(t), v_2(t))$ . Alors la courbe  $t \mapsto (u_1(t), v_1(t)) \in D_1$  décrit l'arête dans  $X_1$ , la courbe  $t \mapsto (u_2(t), v_2(t)) \in D_2$  décrit l'arête dans  $X_2$ , et les deux applications  $t \mapsto s_1(u_1(t), v_1(t))$  (resp.  $t \mapsto s_2(u_2(t), v_2(t))$ ) sont des paramétrisations (approchées en général) de l'intersection. Noter que les deux arêtes sont *couplées*, i.e. le même paramètre sert à les paramétrer.

## 6.3 Théorie générale

### 6.3.1 Position du problème

L'exemple des intersections de surfaces montre qu'il faut savoir traiter en général les systèmes non linéaires de la forme  $F = 0$  où  $F : \mathbf{R}^{n-1}$ . Toutefois, on explique le principe de la méthode principalement dans le cas où  $n = 2$ .

### 6.3.2 Tangente à la courbe des solutions

À quelle condition l'ensemble des solutions est-il une courbe ? Tout fermé du plan est le lieu des zéros d'une fonction  $F : \mathbf{R}^2 \rightarrow \mathbf{R}$  de classe  $C^\infty$ . Il faut donc une hypothèse sur  $F$ . Le théorème des fonctions implicites fournit une réponse. Supposons  $F$  de classe  $C^k$ ,  $k \geq 1$ . Soit  $(u_0, v_0)$  une solution. Si la dérivée partielle

$$\frac{\partial F}{\partial u}(u_0, v_0) \neq 0$$

alors, pour  $u$  voisin de  $u_0$ , l'équation  $F(u, v) = 0$  possède une unique solution  $v$  voisine de  $v_0$ , et celle-ci s'écrit  $v = \phi(u)$  où la fonction  $\phi$  est de classe  $C^k$ . Au voisinage de  $(u_0, v_0)$ , l'ensemble des solutions coïncide donc avec une courbe de classe  $C^k$ , le graphe de  $\phi$ .

**Exemple.** Dans l'exemple de l'exercice 2, l'hypothèse du théorème des fonctions implicites est satisfaite, sauf aux deux points  $(\frac{2\pi}{3}, 0)$  et  $(\frac{4\pi}{3}, 0)$ .

Pour que l'ensemble des solutions soit une courbe (graphe d'une fonction de  $u$  ou bien d'une fonction de  $v$ ) au voisinage de  $(u_0, v_0)$ , il suffit donc que l'une des dérivées partielles

$$\frac{\partial F}{\partial u}(u_0, v_0) \quad \text{ou} \quad \frac{\partial F}{\partial v}(u_0, v_0)$$

ne s'annule pas. Autrement dit, il suffit que le gradient

$$\nabla_{(u_0, v_0)} F = \begin{pmatrix} \frac{\partial F}{\partial u}(u_0, v_0) \\ \frac{\partial F}{\partial v}(u_0, v_0) \end{pmatrix}$$

soit non nul. Dans ce cas, on peut deviner quelle est la tangente en  $(u_0, v_0)$  à la courbe d'équation  $F = 0$ .

**Proposition 20** Soit  $D$  un domaine de  $\mathbf{R}^2$ ,  $f : D \rightarrow \mathbf{R}$  une fonction de classe  $C^1$ ,  $p_0 = (u_0, v_0)$  un point de  $D$  tel que  $f(p_0) = 0$  et  $\nabla_{p_0} F \neq 0$ . Alors l'ensemble des solutions de l'équation  $F = 0$  est une courbe au voisinage de  $p_0$  dont la tangente est

- le noyau de  $D_{p_0} F$  ;
- la droite orthogonale au gradient  $\nabla_{p_0} F \neq 0$  ;
- la droite vectorielle d'équation

$$u \frac{\partial F}{\partial u}(u_0, v_0) + v \frac{\partial F}{\partial v}(u_0, v_0) = 0.$$

**Preuve.** Soit  $t \mapsto c(t) = (u(t), v(t))$  une paramétrisation locale de la courbe d'équation  $F = 0$  au voisinage de  $p_0 = c(t_0)$ . Comme la fonction  $F$  est constante le long de  $c$ ,

$$0 = \frac{dF(c(t))}{dt} \Big|_{t=t_0} = D_{p_0}F(c'(t_0)) = \nabla_{p_0}F \cdot c'(t_0),$$

donc le vecteur directeur de la tangente  $c'(t_0)$  est dans le noyau de  $D_{p_0}F$  (resp. orthogonal au gradient). ■

**Remarque.** L'énoncé ci-dessus est local : au voisinage de  $p_0$ , on voit une seule courbe. Vu globalement, l'ensemble des solutions est en général formé de plusieurs courbes.

### 6.3.3 Cas de l'intersection de deux surfaces

**Proposition 21** *Pour que l'intersection de deux surfaces  $X_1$  et  $X_2$  soit une courbe au voisinage de  $p$ , il suffit que les plans tangents à  $X_1$  et  $X_2$  en  $p$  soient distincts. Dans ce cas, la tangente en  $p$  à  $X_1 \cap X_2$  est l'intersection des plans tangents*

$$T_p(X_1 \cap X_2) = T_pX_1 \cap T_pX_2.$$

**Preuve.** Si  $X_1$  est donnée par une paramétrisation  $s$  telle que  $p = s(u_0, v_0)$  et  $X_2$  par une équation  $f$ , alors  $T_pX_1$  est engendré par

$$\frac{\partial s}{\partial u}(u_0, v_0) \quad \text{et} \quad \frac{\partial s}{\partial v}(u_0, v_0)$$

et  $T_pX_2 = \ker D_p f$

L'équation à résoudre est  $F = 0$  où  $F = f \circ s$ . Alors

$$\frac{\partial F}{\partial u}(u_0, v_0) = D_p f \left( \frac{\partial s}{\partial u}(u_0, v_0) \right) \quad \text{et} \quad \frac{\partial F}{\partial v}(u_0, v_0) = D_p f \left( \frac{\partial s}{\partial v}(u_0, v_0) \right).$$

Si  $T_pX_1 \neq T_pX_2$ ,  $D_p f$  n'est pas identiquement nulle sur  $T_pX_1$ , donc l'un de ces deux nombres est non nul, et le théorème des fonctions implicites s'applique. L'ensemble des solutions dans  $D$  est d'intersection  $X_1 \cap X_2$  est donc une courbe qu'on peut paramétrer par  $t \mapsto (u(t), v(t))$ . L'intersection  $X_1 \cap X_2$  est l'image de la courbe  $t \mapsto c(t) = s(u(t), v(t))$ . Comme  $c$  est contenu dans  $X_1$  (resp.  $X_2$ ), son vecteur vitesse est contenu dans  $T_pX_1$  (resp.  $T_pX_2$ ), donc  $T_p(X_1 \cap X_2) = T_pX_1 \cap T_pX_2$ . ■

### 6.3.4 Coeur de l'algorithme

Supposons qu'on sache calculer, pour chaque point  $p$  de  $\mathbf{R}^2$ , la valeur  $F(p) \in \mathbf{R}$  et les dérivées partielles de  $F$ . Supposons connue une solution  $p_0$  de l'équation  $F = 0$ . Alors on peut imaginer un algorithme appelé *cheminement* (**continuation method**) pour placer une suite de points sur la courbe des solutions.

- **Prédiction.** Etant donnée une solution approchée  $p_0$ , on calcule un vecteur unitaire directeur  $\tau_0$  de  $\ker D_{p_0}F$ , et on pose  $p_{1,e} = p_0 + sh\tau_0$ , où  $h > 0$  est le *pas*, un paramètre à fixer en fonction de la géométrie de la fonction  $F$ , et  $s$  un signe  $\pm$ , déterminant le sens de la progression. C'est la *prédiction* pour le point  $p_1$ .
- **Correction.** Le point  $p_{1,e}$  situé sur une tangente à la courbe, n'est pas sur la courbe en général. Typiquement, il se trouve à une distance de la courbe de l'ordre de  $h^2$  (voir figure). On aimerait pouvoir projeter  $p_{1,e}$  sur la courbe. On effectue une *correction* légèrement différente : on cherche l'intersection de la courbe avec la droite passant par  $p_{1,e}$ , orthogonale à  $\tau_0$ . Pour cela, on utilise la méthode de Newton-Raphson.

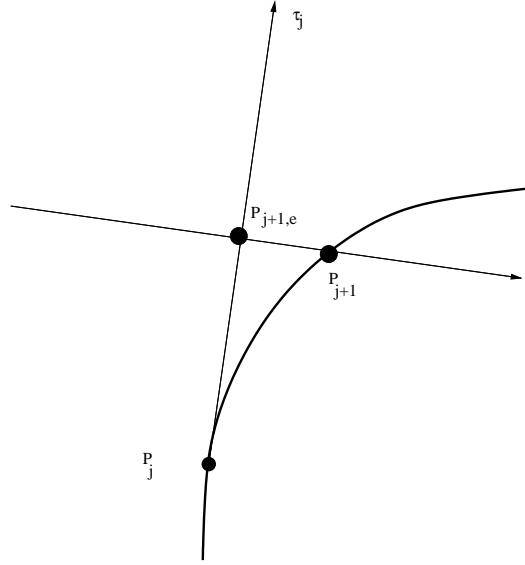


Figure 6.2:

### 6.3.5 Rappel sur la méthode de Newton-Raphson en dimension 1

C'est une méthode numérique adaptée à la résolution d'équations de la forme  $F = 0$  où  $F : \mathbf{R}^{n-1} \rightarrow \mathbf{R}^{n-1}$ . Rappelons son principe lorsque  $n = 2$ .

Soit  $I$  un intervalle de  $\mathbf{R}$  et  $f : I \rightarrow \mathbf{R}$  une fonction de classe  $C^1$ . On cherche une solution de l'équation  $f = 0$ . On suppose connue une *solution approchée*, i.e. un point  $x_0 \in I$  tel que  $f(x_0)$  est petit.

Pour construire une meilleure approximation  $x_1$ , on remplace la courbe représentative de  $f$  par sa tangente (voir figure), i.e. on remplace la fonction  $f$  par la fonction affine

$$g : x \mapsto f(x_0) + f'(x_0)(x - x_0).$$

On choisit pour  $x_1$  la solution de l'équation  $g(x_1) = 0$ , soit

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

On montre aisément que si  $z$  est une solution *non dégénérée*, i.e. telle que  $f(z) = 0$  et  $f'(z) \neq 0$ , alors il existe un voisinage  $J$  de  $z$  tel que pour tout choix de  $x_0 \in J$ , la suite définie par récurrence par

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)}$$

est définie pour tout  $j$  et converge à vitesse surexponentielle vers  $z$ . De plus, on peut estimer la largeur de  $J$  en fonction d'une borne sur la dérivée seconde de  $f$  et d'une borne inférieure sur  $|f'(z)|$ , et donner une estimation *a priori* de la forme

$$|x_j - z| \leq \frac{1}{1 - k} |x_j - x_{j+1}|$$

où

$$k = \sup_{x \in J} \left| \frac{f(x)f''(x)}{f'(x)^2} \right| < 1.$$

Cette estimation fournit un test d'arrêt : on s'arrête lorsque  $|x_j - x_{j+1}| = |f(x_j)/f'(x_j)|$  est inférieur à la précision souhaitée.

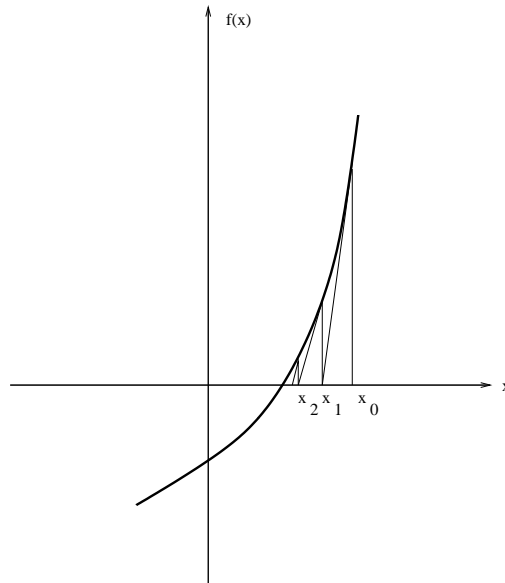


Figure 6.3:

Attention, l'énoncé ci-dessus ne promet qu'un petit bassin d'attraction pour chaque solution non dégénérée. Si  $x_0$  n'appartient pas à l'un de ces bassins, le comportement de la suite est en général chaotique.

### 6.3.6 Méthode quasi-Newton

C'est une simplification de la méthode de Newton-Raphson : au lieu de calculer à chaque pas d'itération la valeur  $f'(x_j)$ , on prend une valeur constante  $L$ , considérée comme une bonne approximation de  $f'(z)$ . Autrement dit, on pose

$$x_{j+1} = x_j - \frac{f(x_j)}{L}.$$

Si  $x_0$  est dans un voisinage  $J$  assez petit d'une solution non dégénérée  $z$ , la suite converge toujours vers  $z$ , mais simplement exponentiellement cette fois. La vitesse de convergence est contrôlée par  $\sup_{x \in J} |1 - L^{-1}f'(x)|$ .

### 6.3.7 Mise en oeuvre de la méthode quasi-Newton

Revenons au cheminement. Il s'agit de corriger la position estimée  $p_{1,e} = p_0 + hs\tau_0$  où  $\tau_0$  est unitaire, orthogonal à  $\nabla_{p_0}F$ . On cherche  $p_1$  sur la droite passant par  $p_{1,e}$  et orthogonale à  $\tau_0$ , i.e. sous la forme  $p_1 = p_{1,e} + t\nabla_{p_0}F$  où  $t \in \mathbf{R}$ . L'équation à résoudre est  $f(t) = 0$  où

$$f(t) = F(p_{1,e} + t\nabla_{p_0}F).$$

La dérivée de  $f$  vaut

$$f'(t) = \nabla_{p_0}F \cdot \nabla_{p_{1,e} + t\nabla_{p_0}F}F.$$

Comme approximation de la dérivée, on prend

$$L = \nabla_{p_0}F \cdot \nabla_{p_0}F = |\nabla_{p_0}F|^2.$$

On itère donc

$$t_{j+1} = t_j - F(p_{1,e} + t_j\nabla_{p_0}F)/|\nabla_{p_0}F|^2.$$

Le test d'arrêt porte sur la taille de l'incrément  $F(p_{1,\epsilon} + t_j \nabla_{p_0} F) / |\nabla_{p_0} F|^2$  : dès qu'il est inférieur à la précision voulue, on s'arrête. A l'usage, on voit que quelques pas d'itération suffisent en général, lorsque  $F(p_{1,\epsilon}) / |\nabla_{p_0} F|^2$  est suffisamment petit. Le développement de Taylor montre que  $F(p_{1,\epsilon}) / |\nabla_{p_0} F|^2$  est de l'ordre de  $h^2 / |\nabla_{p_0} F|$ . Le choix d'un pas  $h$  suffisamment petit est donc nécessaire pour la convergence de la méthode quasi-Newton.

### 6.3.8 Généralisation aux dimensions supérieures

La méthode de Newton-Raphson s'étend aux dimensions supérieures.

Soit  $U$  un ouvert de  $\mathbf{R}^n$  et  $f : U \rightarrow \mathbf{R}^n$  une application de classe  $C^1$ . On cherche une solution de l'équation  $f = 0$ . On suppose connue une solution approchée  $q_0 \in U$  tel que  $f(q_0)$  est petit. Pour construire une meilleure approximation  $p_1$ , on remplace l'application  $f$  par l'application affine tangente

$$g : q \mapsto f(q_0) + D_{q_0} f(q - q_0).$$

Ici,  $D_{q_0} f$  est représenté par la *matrice jacobienne*  $J_f$ , une matrice carrée  $n \times n$ . Ses coefficients sont les dérivées partielles des composantes de  $f$  : si  $f = (f_1, \dots, f_n)$ , alors

$$(J_f)_{ij} = \frac{\partial f_i}{\partial u_j}(q_0).$$

On choisit pour  $q_1$  la solution de l'équation  $g(q_1) = 0$ . C'est un système linéaire de  $n$  équations à  $n$  inconnues de rang  $n$ . Symboliquement,

$$q_1 = q_0 - (D_{q_0} f)^{-1}(f(q_0)).$$

Une méthode quasi-Newton consiste à remplacer la matrice variable  $D_{q_0} f$  par une matrice constante  $L$  qui en constitue une bonne approximation. Autrement dit,  $q_1$  est la solution du système linéaire

$$f(q_0) + L(q_1 - q_0) = 0.$$

### 6.3.9 Le Théorème des submersions

Ce théorème garantit que l'ensemble des solutions est une réunion de courbes.

**Théorème 10** (Version locale). *Soit  $F : \mathbf{R}^n \rightarrow \mathbf{R}^{n-1}$  une application de classe  $C^k$ ,  $k \geq 1$ . Soit  $p_0$  un point de  $\mathbf{R}^n$  tel que  $F(p_0) = 0$ . On suppose que la différentielle  $D_{p_0} F$  est de rang  $n - 1$ . Alors, au voisinage de  $p_0$ , l'ensemble des solutions de l'équation  $F = 0$  est une courbe régulière de classe  $C^k$ , dont la tangente en  $p_0$  est la droite vectorielle  $\ker D_{p_0} F$ .*

**Preuve.** Soit  $v$  un vecteur directeur de  $\ker D_{p_0} F$ . On suppose que la  $i$ -ème composante de  $v$  est non nulle. On applique le théorème d'inversion locale à  $G : \mathbf{R}^n \rightarrow \mathbf{R}^n$  définie par

$$G(p) = (F(p), x_i)$$

où  $x_i$  désigne la  $i$ -ème fonction coordonnée. Au voisinage de  $p_0$ , l'ensemble des solutions est l'image du dernier axe de coordonnées par le difféomorphisme réciproque de  $G$ , c'est donc une courbe régulière.

Si  $t \mapsto c(t)$  est une paramétrisation locale de cette courbe, alors

$$D_{p_0} F(c'(t_0)) = \frac{dF(c(t))}{dt} \Big|_{t=t_0} = 0,$$

donc  $c'(t_0) \in \ker D_{p_0} F$ . ■

**Corollaire 22** (Version globale). *Soit  $F : \mathbf{R}^n \rightarrow \mathbf{R}^{n-1}$  une application de classe  $C^k$ ,  $k \geq 1$ . On suppose qu'en tout point  $p$  de  $\mathbf{R}^n$  tel que  $F(p) = 0$ , la différentielle  $D_p F$  est de rang  $n - 1$ . Alors, l'ensemble des solutions de l'équation  $F = 0$  est une sous-variété de dimension 1, i.e. une réunion de courbes régulières de classe  $C^k$ .*

### 6.3.10 Cheminement en dimension $n$

Le coeur de l'algorithme de cheminement s'adapte en toute dimension.

- Calcul d'un vecteur unitaire directeur de la tangente à la courbe des solutions au point  $p_j$ . Il s'agit de résoudre un système linéaire de  $n - 1$  équations à  $n$  inconnues, de rang  $n - 1$ .
- Choix du pas et du sens de progression, calcul de la position estimée  $p_{j+1,e}$ .
- Calcul d'une base  $(\nu_1, \dots, \nu_{n-1})$  de l'hyperplan orthogonal à la tangente. Il s'agit à nouveau de résoudre un système linéaire de  $n - 1$  équations à  $n$  inconnues, de rang  $n - 1$ .
- Résolution approchée, par une méthode quasi-Newton, de l'équation

$$F(p_{j+1,e} + t_1\nu_1 + \dots + t_{n-1}\nu_{n-1}) = 0$$

dans  $\mathbf{R}^{n-1}$ .

- Calcul de  $p_{j+1} = p_{j+1,e} + t_1\nu_1 + \dots + t_{n-1}\nu_{n-1}$ .

## 6.4 Description de l'algorithme complet

Plusieurs aspects du problème ont été éludés lors de la discussion théorique.

- Comment trouver une solution initiale  $p_0$  ?
- Comment choisir le pas et le sens de progression pour la prédiction ?
- Sur quel critère arrêter la progression ?
- Que fait on lorsqu'on arrive en un point où l'hypothèse  $DF$  de rang  $n - 1$  n'est pas satisfaite ?
- Comment passer d'un fichiers de points à une vraie courbe paramétrée ?

On examine ces points un par un.

### 6.4.1 Détermination des inits

- Solutions sur la frontière. On cherche d'abord une solution sur le bord du domaine  $D$ . Si  $\partial D$  est un BRep, le problème sur chaque face se ramène à résoudre une équation  $F' : D' \rightarrow \mathbf{R}^{n-1}$  où  $D' \subset \mathbf{R}^{n-1}$ . La méthode de Newton en dimension  $n - 1$  s'applique. On prend comme solutions approchées (inits dans la méthode de Newton) les points d'une grille.
- Cas où les courbes de solutions ne coupent pas la frontière. Si on ne trouve pas de solution sur le bord, on cherche des points où la courbe des solutions se retourne, i.e. est tangente à l'un des hyperplans de coordonnées. Cela se traduit par une equation supplémentaire. Par exemple, la tangente à la courbe des solutions est parallèle à l'hyperplan  $\{x_n = 0\}$  si et seulement si le premier mineur  $M_F$  de la matrice jacobienne  $J_f$  est nul. On doit donc résoudre le système  $G = 0$  où  $G = (F, M_F) : \mathbf{R}^n \rightarrow \mathbf{R}^n$ . La méthode de Newton en dimension  $n$  s'applique. On prend comme solutions approchées (inits dans la méthode de Newton) les points d'une grille.
- Elimination des points d'init redondants. S'il y a des singularités (points où les hypothèses du théorème des submersions ne sont pas satisfaites), elles figurent probablement dans la liste des points d'init : il faut les supprimer. Chaque composante connexe de l'ensemble des solutions est ou bien une courbe fermée ou bien un arc reliant deux points du bord. Il peut contenir plusieurs des points trouvés lors de la recherche des points d'init : il faut retirer ces points de la liste des points d'init lorsqu'on les rencontre au cours de la progression.

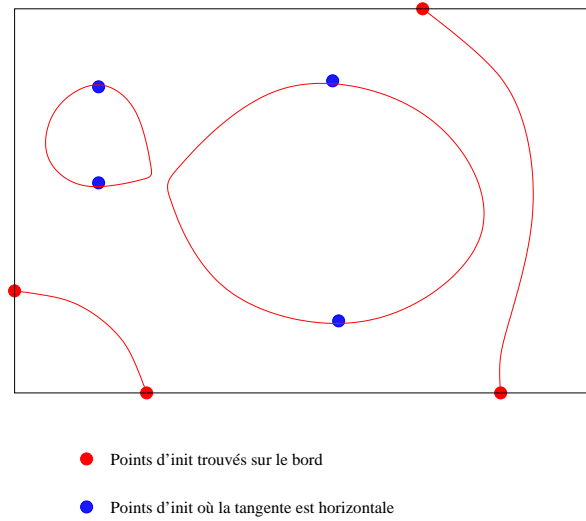


Figure 6.4:

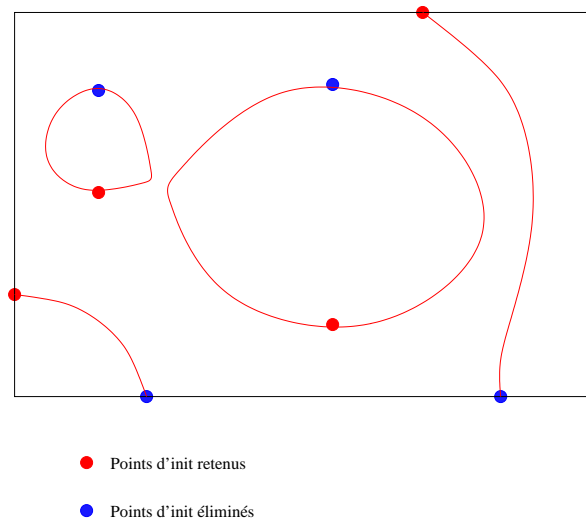


Figure 6.5:



### 6.4.2 Choix du pas

- Pas trop petit, sinon ça n'avance pas.
- Pas trop grand, sinon le quasi-Newton ne converge pas ou saute vers une autre composante de l'ensemble des solutions.
- Pas de recette universelle.

Toutefois, voici une approche géométrique du choix du pas : on choisit le pas de sorte que la *flèche*, i.e. la distance  $|p_{j+1} - p_{j,e}|$  soit égal à une constante  $d$ .

**Proposition 23** *Soit  $t \mapsto c(t)$  une courbe plane. Notons  $s \mapsto T(s)$  sa tangente en  $p = c(0)$ , parcourue à vitesse 1. On suppose que  $c$  paramétrée par sa projection sur  $T$ , de sorte que  $c(t)$  soit le point d'intersection de  $c$  et de la droite orthogonale à  $T$  passant par  $T(t)$ . Alors la distance du point  $c(t)$  à la droite  $T$  satisfait*

$$|c(t) - T(t)| = \frac{1}{2}|\kappa|t^2 + o(t^2)$$

lorsque  $t$  tend vers 0, où  $\kappa$  est la courbure de  $c$  en  $p$ . Si  $c$  est l'ensemble des solutions de l'équation  $F = 0$ , alors

$$|\kappa| = \frac{|D_p^2 F(\tau)|}{|\nabla P_F|}$$

où  $\tau$  est un vecteur unitaire tangent à  $c$  en  $p$ .

**Preuve.** Dans le repère orthonormé  $(p, \tau, \nu)$ , dont le premier axe est  $T$  et dont le second est orienté en direction de  $c$ , les coordonnées de  $c(t)$  sont  $x(t) = t$  et  $y(t) = |c(t) - T(t)|$ . Par définition de la courbure,  $|\kappa| = y''$  d'où  $y = \frac{1}{2}|\kappa|t^2 + o(t^2)$ .

En dérivant deux fois l'identité  $F(t, y(t)) = 0$  en  $t = 0$ , on obtient

$$\frac{\partial^2 F}{\partial x^2}(p) + y''(0) \frac{\partial F}{\partial y}(p) = 0.$$

Or

$$\frac{\partial^2 F}{\partial x^2}(p) = D_p^2 F(1, 0) = D_p^2 F(\tau)$$

et

$$\frac{\partial F}{\partial y}(p) = \nabla_p F \cdot \nu = \nabla_p F \cdot \frac{\nabla_p F}{|\nabla_p F|} = |\nabla_p F|. \blacksquare$$

Cette proposition entraîne que pour  $h$  petit,

$$|p_{j+1} - p_{j,e}| \sim \frac{1}{2}|\kappa(p_j)|h^2$$

et suggère donc de prendre  $h_j = \sqrt{2dR_j}$  où  $R_j$  est le rayon de courbure de  $c$  en  $p_j$ .

**Remarque.** Les classes NURBS de CATIA comportent une méthode qui retourne les dérivées secondes. Néanmoins, le calcul du rayon de courbure est cher, et on peut l'approcher par

$$\tilde{R}_j = \frac{|p_j - p_{j-1}|}{|\tau_j - \tau_{j-1}|}.$$

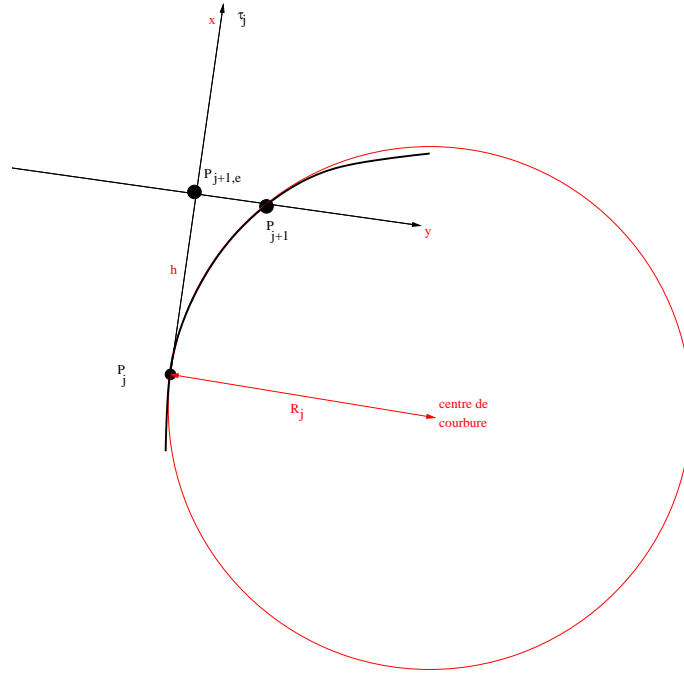


Figure 6.6:

### 6.4.3 Choix du sens de progression et test d'arrêt

Si le point d'init  $p_0$  est sur le bord, on choisit le signe  $s$  de sorte que le vecteur  $s\tau_0$  pointe vers l'intérieur du domaine, et on poursuit par continuité.

**Remarque.** Noter que le cas où la courbe solution est tangente au bord en un point  $p$  est doublement difficile à traiter. D'abord, l'hypothèse  $D_p f$  inversible de la méthode de Newton n'est pas satisfaite, ce qui rend difficile la détection de la solution  $p$ . D'autre part,  $\tau_0$  est tangent au bord, et il est possible que  $p$  soit une solution isolée.

Si le point d'init  $p_0$  est à l'intérieur du domaine, le choix initial du signe  $s$  est indifférent.

La progression doit s'arrêter

- Lorsqu'on arrive au bord du domaine.
- Lorsqu'on revient à proximité du point d'init (cas d'une composante fermée de l'ensemble des solutions).

### 6.4.4 Détection des singularités

Une *singularité*, pour la résolution de l'équation  $F = 0$ , c'est une solution  $p$  où le gradient de  $F$  s'annule. *Génériquement*,  $F$  est de Morse en  $p$ , i.e.  $F$  est une forme quadratique à changement de coordonnées curvilignes près.

Plaçons nous en dimension 2. Dans ce cas, au voisinage de  $p$ , l'ensemble des solutions est

- ou bien réduit à un point ;
- ou bien formé de deux courbes lisses ayant des tangentes distinctes.

Dans le second cas, la tangente est continue. Par conséquent la droite engendrée par le vecteur

$$\tau = \frac{1}{|\nabla F|} \begin{pmatrix} -\frac{\partial F}{\partial v} \\ \frac{\partial F}{\partial u} \end{pmatrix}$$

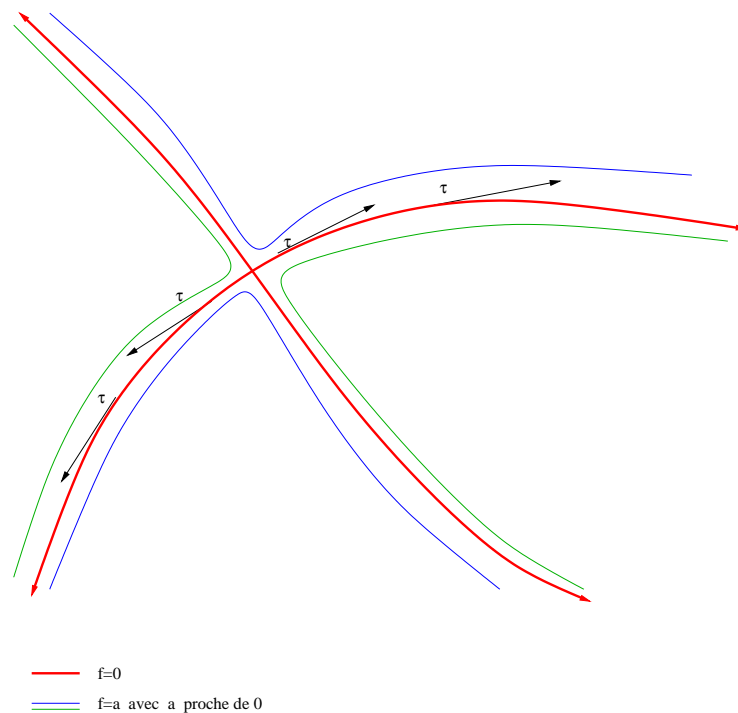


Figure 6.7:

est continue, mais le vecteur  $\tau$  lui-même change de signe sur cette droite. Il est donc important de détecter ce changement brutal de signe (en comparant  $\tau_j$  à  $\tau_{j-1}$ ) et de changer le signe  $s$  en conséquence.

Le traitement correct de singularités plus rares nécessite les méthodes du calcul formel.

#### 6.4.5 Du discret au continu

La méthode du cheminement produit un certain nombre de tableaux de points. Il est d'usage d'interpoler les points de chaque tableau par une courbe B-spline cubique par exemple.

Le cas des composantes fermées nécessite un soin particulier : pour éviter que la B-spline ne fasse une épingle à cheveux, il faut arrêter la progression avant d'avoir dépassé le point d'init et choisir le point d'init comme premier et dernier point de contrôle (voir figure).

### 6.5 Alternative au cheminement

- Méthodes du type grille. On décompose le domaine en petits carrés, on compare les signes pris par la fonction  $F$  aux sommets de chaque carré pour prolonger le tracé à travers un carré. C'est simple mais coûteux en mémoire. La précision est fixée par celle de la grille. Il est difficile d'adapter le pas à la courbure.
- Méthode champ de vecteurs. Il s'agit de voir l'ensemble des solutions comme une réunion de trajectoires du champ de vecteurs  $\tau$ . C'est une simplification du cheminement (suppression de l'étape de correction) qui bénéficie des méthodes d'itération développées pour les équations différentielles. La méthode devient instable au voisinage d'une singularité.

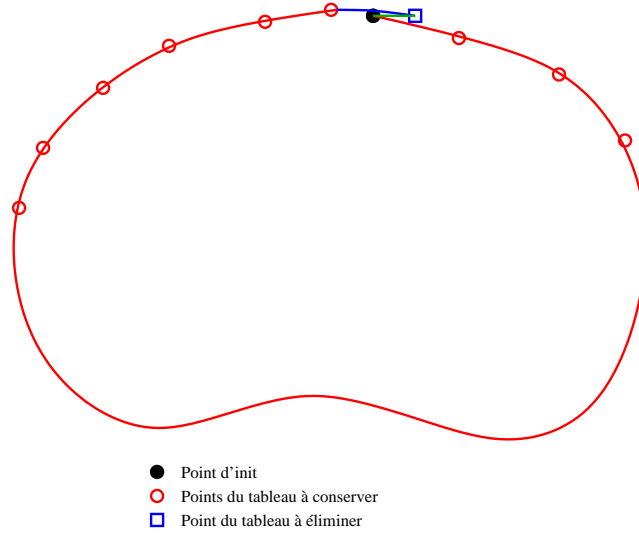


Figure 6.8: Modification du tableau de points avant interpolation

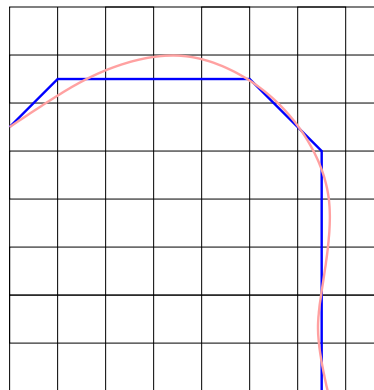


Figure 6.9: Méthode de grille

# Chapter 7

## Interpolation et approximation

### 7.1 Interpolation

*Interpoler* une famille de  $N + 1$  points  $Q_j$  de  $\mathbf{R}^n$ , c'est trouver une courbe qui passe par tous ces points. En plus de prescrire des positions, on peut aussi vouloir prescrire des tangentes.

Le prototype du problème d'interpolation consiste à se donner  $a = t_0 < t_1 < \dots < t_N = b$  et  $Q_0, \dots, Q_N \in \mathbf{R}$ , et chercher une fonction  $x : [a, b] \rightarrow \mathbf{R}$  telle que  $f(t_i) = Q_i$  pour tout  $i = 0, \dots, N$ , voir figure 7.1.

#### 7.1.1 Polynôme d'interpolation de Lagrange

Une réponse classique à la question consiste à chercher  $f$  parmi les polynômes de degré au plus  $N$ .

**Théorème 11** *Etant donnés  $a = t_0 < t_1 < \dots < t_N = b$  et  $Q_0, \dots, Q_N \in \mathbf{R}$ , il existe un unique polynôme  $x : [a, b] \rightarrow \mathbf{R}$  de degré  $\leq N$  tel que  $x(t_i) = Q_i$  pour tout  $i = 0, \dots, N$ . On l'appelle le polynôme d'interpolation de Lagrange.*

**Preuve.** Soit  $E$  l'espace vectoriel des polynômes à coefficients réels, de degré  $\leq N$ . Alors  $\dim E = N + 1$ . On considère l'application linéaire

$$E \rightarrow \mathbf{R}^{N+1}, \quad x \mapsto (x(t_0), \dots, x(t_N)).$$

Pour montrer qu'elle est bijective, il suffit de vérifier qu'elle est injective. Or si  $x(t_0) = \dots = x(t_N) = 0$ , le polynôme  $x$  a  $N + 1 > \deg(f)$  racines distinctes, donc il est nul. ■

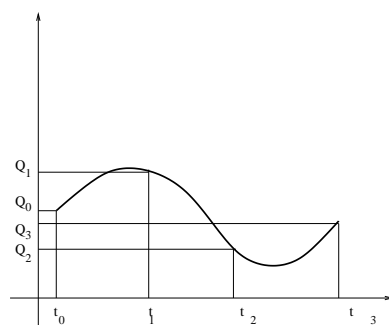


Figure 7.1: Interpolation en dimension 1

### 7.1.2 Erreur d'interpolation

La qualité d'un procédé d'interpolation se mesure par une *estimation d'erreur*. Il s'agit de vérifier si l'interpolante approche bien une fonction donnée suffisamment régulière. Autrement dit, étant donnée une fonction  $f$  suffisamment différentiable, de majorer la différence entre  $f$  et la fonction  $x$  qui *interpole*  $f$  aux points  $t_i$ , i.e. telle que  $x(t_i) = f(t_i)$  pour  $i = 0, \dots, N$ .

Pour le polynôme d'interpolation de Lagrange, l'estimation d'erreur classique est la suivante.

**Théorème 12** *Soit  $f$  une fonction de classe  $C^{N+1}$  sur  $[a, b]$  et  $x_{Lag, N+1}$  le polynôme d'interpolation de Lagrange de  $f$  aux points  $a = t_0 < \dots < t_N = b$ . Alors*

$$\|f - x_{Lag, N+1}\|_{\infty} \leq N! h^{N+1} \|f^{(N+1)}\|_{\infty}$$

où  $h = \max\{t_{i+1} - t_i\}$ .

**Preuve.** Posons  $g = f - x$ . Alors  $g(t_i) = 0$  et  $g^{(N+1)} = f^{(N+1)}$ . Le théorème de Rolle donne un zéro de  $g'$  dans chaque intervalle, i.e.  $N$  points  $t_{1,i}$ ,  $i = 0, \dots, N-1$  tels que  $g'(t_{1,i}) = 0$  et  $t_{1,i} \in [t_i, t_{i+1}]$ . Une nouvelle application du théorème de Rolle donne un zéro  $t_{2,i}$  de  $g''$  dans chaque intervalle  $[t_{1,i}, t_{1,i+1}] \subset [t_i, t_{i+2}]$ , pour  $i = 0, \dots, N-2$ . On continue jusqu'à trouver un zéro  $t_{N,0}$  de  $g^{(N)}$  dans l'intervalle  $[t_{N-1,0}, t_{N-1,1}] \subset [t_0, t_N] = [a, b]$ .

Etant donné  $t \in [t_i, t_{i+1}]$ , on majore

$$\begin{aligned} |g^{(N)}(t)| &= |g^{(N)}(t) - g^{(N)}(t_{N,0})| \\ &= \left| \int_{t_{N,0}}^t g^{(N+1)}(s) ds \right| \\ &= \left| \int_{t_{N,0}}^t f^{(N+1)}(s) ds \right| \\ &\leq |t - t_{N,0}| \|f^{(N+1)}\|_{\infty} \\ &\leq Nh \|f^{(N+1)}\|_{\infty}, \end{aligned}$$

d'où

$$\|g^{(N)}\|_{\infty} \leq Nh \|f^{(N+1)}\|_{\infty}.$$

Ensuite on montre par récurrence sur  $k$  que

$$\|g^{(N-k)}\|_{\infty} \leq N(N-1) \dots (N-k+1) h^{k+1} \|f^{(N+1)}\|_{\infty}.$$

En effet, notant  $i_k = \min\{i, N-k\}$ ,

$$\begin{aligned} |g^{(N-k)}(t)| &= |g^{(N-k)}(t) - g^{(N-k)}(t_{N-k, i_k})| \\ &\leq |t - t_{N-k, i_k}| \|g^{(N-k+1)}\|_{\infty} \\ &\leq (N-k+1)h N(N-1) \dots (N-k+2) h^k \|f^{(N+1)}\|_{\infty} \\ &= N(N-1) \dots (N-k+1) h^{k+1} \|f^{(N+1)}\|_{\infty}. \end{aligned}$$

Lorsque  $k = N$ , on obtient l'inégalité de l'énoncé. ■

### 7.1.3 Des polynômes aux splines

La faiblesse de l'interpolation de Lagrange, c'est que l'erreur d'interpolation croît avec  $N$ . Cela se traduit expérimentalement par de grandes oscillations du polynôme d'interpolation, même si  $f$  est très simple. D'où l'idée d'interpoler par des fonctions polynômiales par morceaux, dont le degré n'augmente pas avec le nombre de points d'interpolation.

Par exemple, la fonction  $X_1$  qui est affine sur chaque intervalle  $[t_i, t_{i+1}]$  et interpole une fonction  $f$  de classe  $C^2$  satisfait, d'après le théorème 12 appliqué sur chaque intervalle,

$$\|f - X_1\|_{\infty} \leq 2h^2 \|f^{(2)}\|_{\infty}.$$

Comme on souhaite interpoler par des fonctions plus différentiables que  $X_1$ , on cherche l'interpolante dans l'espace des fonctions de classe  $C^{k-1}$  polynômiales de degré  $k$  sur chaque intervalle  $[t_i, t_{i+1}]$ , appelées *fonctions splines* (la terminologie a été introduite par Schoenberg en 1946 : en anglais, *spline* désigne une bande de métal souple utilisée par les dessinateurs pour tracer une jolie courbe entre deux points). C'est ainsi que sont nées les fonctions et les courbes B-splines.

### 7.1.4 Interpolation par des courbes B-splines

On se limite aux courbes de degré 3, pour simplifier. On cherche à faire passer une courbe B-spline de degré  $k$  de positions et vitesses aux extrémités prescrites par  $N - 1$  points  $Q_i$ . Le problème se divise en deux phases.

Première phase : On se fixe un vecteur de noeuds  $\mathbf{t}$  et on cherche un polygone de contrôle  $\mathbf{P}$  tel que la courbe B-spline  $X_k$  correspondante passe par les  $Q_i$  aux noeuds. L'interpolation se traduit alors par la résolution d'un système linéaire.

Deuxième phase : on cherche à optimiser le choix du vecteur de noeuds. C'est typiquement non linéaire.

### 7.1.5 Le problème linéaire

A nouveau, on se contente d'énoncer le théorème pour des B-splines de degré 3. Une généralisation à tout degré impair se trouve dans [K], Theorem 8.30 page 172.

**Théorème 13** Soient  $Q_0, \dots, Q_N$  des points de  $\mathbf{R}^n$ . Soient  $v_a, v_b$  deux vecteurs de  $\mathbf{R}^n$ . Soit  $\mathbf{t}$  un vecteur de noeuds vissé aux extrémités, de la forme

$$t_0 = t_1 = t_2 = t_3 = a < t_4 < \dots < t_{N+2} < b = t_{N+3} = t_{N+4} = t_{N+5} = t_{N+6}.$$

Il existe un unique polygone de contrôle  $\mathbf{P} = (P_0, \dots, P_{N+2})$  tel que la courbe B-spline de degré 3 associée satisfasse

$$\forall j = 0, \dots, N, X_3(t_{j+3}) = Q_j, \quad X_3'(a) = v_a, \quad \text{et } X_3'(b) = v_b.$$

**Preuve.** Comme chaque coordonnée se traite indépendamment, on peut supposer que  $n = 1$ . Dans ce cas, on considère l'application linéaire

$$\mathbf{R}^{N+3} \rightarrow \mathbf{R}^{N+3}, \quad (P_0, \dots, P_{N+2}) \mapsto (X_3'(a), X_3(t_3), \dots, X_3(t_{N+3}), X_3'(b)).$$

Pour prouver qu'elle est bijective, il suffit de prouver qu'elle est injective. Cela résulte immédiatement du lemme suivant appliqué à la fonction  $f = 0$ . En effet, si les points et vecteurs interpolés sont tous nuls, le lemme donne  $X_3'' \equiv 0$ . Avec la condition initiale  $X_3(a) = X_3'(a) = 0$ , cela entraîne que  $X_3 \equiv 0$ , et donc que les  $P_i$  sont tous nuls. ■

**Lemme 24** Soient  $f, x : [a, b] \rightarrow \mathbf{R}$  deux fonctions de classe  $C^2$ . On suppose que

- $x$  est polynômiale de degré 3 sur chaque intervalle  $[t_i, t_{i+1}]$ ,  $i = 3, \dots, N + 2$  ;
- $f(t_i) = x(t_i)$  pour  $i = 3, \dots, N + 3$  et  $f'(a) = x'(a)$ ,  $f'(b) = x'(b)$ .

Alors

$$\int_a^b (f''(t) - x''(t))^2 dt = \int_a^b f''(t)^2 dt - \int_a^b x''(t)^2 dt.$$

**Remarque.** Ce lemme signifie que la spline  $x$  qui interpole une fonction  $f$  est la projection orthogonale de  $f$  sur le sous-espace des splines, pour le produit scalaire  $f \cdot g = \int_a^b f''(t)g''(t) dt$ , voir figure 14.

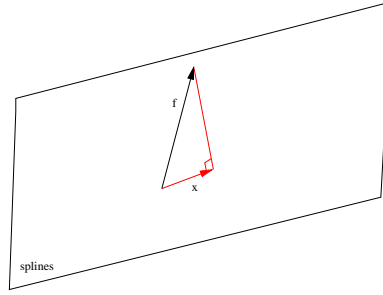


Figure 7.2: L'interpolante est une projection orthogonale

**Preuve.** On vérifie immédiatement que

$$\int_a^b (f''(t) - x''(t))^2 dt - \int_a^b f''(t)^2 dt + \int_a^b x''(t)^2 dt = -2R$$

où

$$R = \int_a^b (f''(t) - x''(t))x''(t) dt.$$

On intègre par parties sur chaque intervalle

$$\begin{aligned} \int_{t_i}^{t_{i+1}} (f''(t) - x''(t))x''(t) dt &= (f'(t_{i+1}) - x'(t_{i+1}))x''(t_{i+1}) - (f'(t_i) - x'(t_i))x''(t_i) \\ &\quad - \int_{t_i}^{t_{i+1}} (f'(t) - x'(t))x'''(t) dt \\ &= (f'(t_{i+1}) - x'(t_{i+1}))x''(t_{i+1}) - (f'(t_i) - x'(t_i))x''(t_i) \\ &\quad - (f(t_{i+1}) - x(t_{i+1}))x'''(t_{i+1}-) - (f(t_i) - x(t_i))x'''(t_i+) \\ &\quad + \int_{t_i}^{t_{i+1}} (f(t) - x(t))x^{(4)}(t) dt \\ &= (f'(t_{i+1}) - x'(t_{i+1}))x''(t_{i+1}) - (f'(t_i) - x'(t_i))x''(t_i), \end{aligned}$$

car  $x^{(4)} \equiv 0$  et  $f(t_{i+1}) - x(t_{i+1}) = f(t_i) - x(t_i) = 0$ . En additionnant, il vient

$$\begin{aligned} R &= (f'(b) - x'(b))x''(b) - (f'(a) - x'(a))x''(a) \\ &= 0, \end{aligned}$$

car  $f'(a) - x'(a) = f'(b) - x'(b) = 0$ . ■

### 7.1.6 Estimation de l'erreur d'interpolation

**Théorème 14** Soit  $f : [a, b] \rightarrow \mathbf{R}$  une fonction de classe  $C^2$ . Soit  $X_3$  la fonction B-spline de degré 3 qui l'interpole en  $N + 1$  point plus les dérivées aux bornes, selon le théorème 13. Alors

$$\|f - X_3\|_\infty \leq \frac{h^{3/2}}{2} \|f''\|_2 \quad \text{et} \quad \|f' - X_3'\|_\infty \leq h^{1/2} \|f''\|_2$$

où  $h = \max\{t_{i+1} - t_i\}$ .



**Preuve.** Posons  $g = f - X_3$ . Le lemme 24 donne

$$\begin{aligned}\|g''\|_2^2 &= \int_a^b g''(t)^2 dt \\ &= \int_a^b f''(t)^2 dt - \int_a^b X_3''(t)^2 dt \\ &\leq \|f''\|_2^2.\end{aligned}$$

Par construction,  $g$  s'annule aux  $t_i$ , donc, d'après le théorème de Rolle,  $g'$  s'annule au moins une fois dans chaque intervalle  $[t_i, t_{i+1}]$ . Tout point  $t \in [a, b]$  est donc à distance au plus  $h$  d'un point  $t'$  tel que  $g'(t') = 0$ . On écrit

$$\begin{aligned}|g'(t)| &= |g'(t) - g'(t')| \\ &= \left| \int_{t'}^t g''(s) ds \right| \\ &\leq \left( \int_{t'}^t ds \right)^{1/2} \left( \int_{t'}^t g''(s)^2 ds \right)^{1/2} \\ &\leq h^{1/2} \|g''\|_2 \\ &\leq h^{1/2} \|f''\|_2,\end{aligned}$$

d'après l'inégalité de Cauchy-Schwarz. Ceci montre que

$$\|g'\|_\infty \leq h^{1/2} \|f''\|_2$$

Tout point  $t \in [a, b]$  est à distance au plus  $h/2$  d'un point  $t''$  tel que  $g(t'') = 0$ . Par conséquent

$$\begin{aligned}|g(t)| &= |g(t) - g(t'')| \\ &= \left| \int_{t''}^t g'(s) ds \right| \\ &\leq |t - t''| \|g'\|_\infty \\ &\leq \frac{h^{1/2}}{2} \|f''\|_2. \blacksquare\end{aligned}$$

**Remarque.** Dans [K], Theorem 8.34 page 178, on trouve l'amélioration suivante. *Sous les hypothèses du Théorème 14, et si de plus  $f$  est de classe  $C^4$ , alors*

$$\|f - X_3\|_\infty \leq \frac{h^4}{16} \|f^{(4)}\|_\infty.$$

### 7.1.7 Choix du vecteur de noeuds

Etant donnés les points  $Q_i$  à interpoler, quel est le meilleur choix des  $t_i$  ? Pour éviter que la dérivée de la courbe interpolante soit grande, il vaut mieux que des points  $Q_i$  et  $Q_{i+1}$  éloignés soient interpolés en des valeurs  $t_i$  et  $t_{i+1}$  éloignées. Autrement dit, il faut corrélérer les espacements  $t_{i+1} - t_i$  avec les distances  $\|Q_{i+1} - Q_i\|$ .

Un choix simple consiste à poser

$$t_{i+1} - t_i = \|Q_{i+1} - Q_i\|.$$

Ce choix s'appelle la *paramétrisation cordale* (**chordal parametrization**). Elle a pour effet de produire une courbe interpolante paramétrée à vitesse à peu près constante, voir figure 7.3.

Lorsque les distances  $\|Q_{i+1} - Q_i\|$  varient brutalement, il y a de meilleurs choix, voir [HL], figures 4.36a et 4.36b pages 205 et 206.

On gagne de la souplesse à fixer des valeurs  $s_j$  où interpoler indépendantes des noeuds et à fixer ensuite les noeuds de façon à optimiser le conditionnement de la matrice du système linéaire, voir [HL].

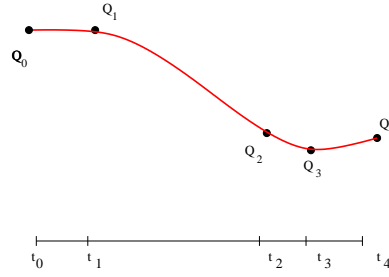


Figure 7.3: Paramétrisation chordale

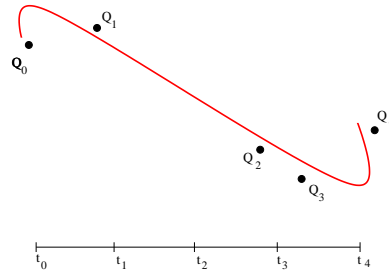


Figure 7.4: Approximation d'un nuage de points par une courbe

## 7.2 Approximation

### 7.2.1 Position du problème

Le problème d'*approximation* d'une famille de  $N + 1$  points  $Q_j$  par une courbe consiste, au lieu d'exiger que la courbe passe exactement par les  $Q_j$ , i.e.  $x(s_j) = Q_j$  (ce qui nécessite autant de degré de liberté que d'équations, à savoir  $n(N + 1)$ ), à chercher à minimiser l'écart quadratique

$$\sum_j \|x(s_j) - Q_j\|^2$$

où  $x$  varie dans un espace possédant typiquement bien moins de degrés de liberté, voir figure 7.4.

**Exemple.** Le problème d'approximation de  $N + 1$  points  $Q_j$  par une application constante, est bien posé.

En effet, l'intervalle  $[0, 1]$  est subdivisé par exemple par  $s_j = j/N$ ,  $j = 0, \dots, N$ . L'inconnue est une application constante de  $[0, 1]$  dans  $\mathbf{R}^n$ , i.e. un point  $x \in \mathbf{R}^n$ . Il doit minimiser

$$E(P, v) = \sum_{j=0}^N \|x - Q_j\|^2.$$

La solution est le centre de gravité des points  $Q_j$ . Dans ce problème, l'inconnue  $x$  vit dans un espace de dimension  $n < n(N + 1)$ .

### 7.2.2 Approximation par une courbe B-spline

Montrons que le problème se ramène à nouveau à la résolution d'un système linéaire.

Soit

$$a = t_0 = \dots = t_k < t_{k+1} < \dots < t_{m-k} = \dots = t_m = b$$

un vecteur de noeuds vissé aux extrémités. Soit  $\mathbf{P} = (P_0, \dots, P_{m-k-2}) \in \mathbf{R}^{n(m-k-1)}$  un polygone de contrôle. La courbe B-spline de degré  $k$  correspondante est donnée sur  $[a, b]$  par  $X_k(t) =$

$\sum_{i=0}^{m-k-2} B_{i,k}(t)P_i$ . Soient  $a \leq s_0 < \dots < s_N \leq b$  des valeurs distinctes dans  $[a, b]$  et  $Q_0, \dots, Q_N$  des points dans  $\mathbf{R}^n$ .

Le problème d'approximation consiste à trouver  $\mathbf{P}$  qui minimise

$$E(\mathbf{P}) = \sum_{j=0}^N \|X_k(s_j) - Q_j\|^2.$$

On remarque que  $E$  est un polynôme du second degré sur  $\mathbf{R}^{n(m-k-1)}$ . Si on représente les coordonnées de tous les  $P_i$  simultanément par une colonne  $V$  à  $n(m-k-1)$  coefficients, alors  $E$  s'écrit

$$E(\mathbf{P}) = {}^T V A V + 2 {}^T B V + c$$

où  $A$  est une matrice symétrique positive,  $B$  est la colonne des coordonnées des vecteurs

$$\sum_j B_{i,k}(s_j) Q_j,$$

et

$$c = \sum_j \|Q_j\|^2.$$

La fonction  $E$  atteint donc son minimum aux vecteurs  $V$  solutions de l'équation vectorielle

$$A V + B = 0.$$

Il s'agit d'un système linéaire à  $n(m-k-1)$  équations et  $n(m-k-1)$  inconnues. Noter que la taille ne dépend pas du nombre de points à approximer.

Typiquement, l'approximation est utilisée pour alléger des données : remplacer une courbe composite ou comportant un grand nombre de points de contrôle par une seule B-spline ayant peu de points de contrôle.

### 7.3 Résolution numérique d'un problème d'interpolation

On se donne le vecteur de noeuds

$$t_0 = t_1 = t_2 = t_3 = 0 < t_4 = 1 < \dots < t_{N+2} = N - 1 < N = t_{N+3} = t_{N+4} = t_{N+5} = t_{N+6}$$

dans l'intervalle  $[0, N]$ . Il s'agit de trouver le polygone de contrôle (à  $N+3$  sommets) de la B-spline qui passe par le point  $Q_i$  en  $t_{i+3}$  et a pour dérivées  $v_0$  (resp.  $v_N$ ) aux extrémités. On peut supposer que  $n = 1$ .

On utilise les valeurs des fonctions B-splines relatives au vecteur de noeuds ci-dessus (en exercice du cours de géométrie différentielle).

$$X_3(0) = P_0, \tag{7.1}$$

$$X_3'(0) = 3(P_1 - P_0), \tag{7.2}$$

$$X_3(t_4) = \frac{1}{4}P_1 + \frac{7}{12}P_2 + \frac{1}{6}P_3, \tag{7.3}$$

$$X_3(t_{i+3}) = \frac{1}{6}P_i + \frac{2}{3}P_{i+1} + \frac{1}{6}P_{i+2} \tag{7.4}$$

pour  $i \geq 2$ .

Il s'agit de résoudre le système  $A\mathbf{P} = \mathbf{Q}$  pour  $\mathbf{Q} = (Q_0, v_0, Q_1, \dots, Q_{N-1}, v_N, Q_N)$  et  $\mathbf{P} \in \mathbf{R}^{N+3}$  et

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ -3 & 3 & 0 & 0 & 0 & \dots \\ 0 & \frac{1}{4} & \frac{7}{12} & \frac{1}{6} & 0 & \dots \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

La matrice  $A$  n'est pas tout à fait symétrique. Comme on préfère les matrices symétriques, on résoud le système équivalent

$${}^T A A P = {}^T A Q.$$

La matrice  ${}^T A A$  est *coercive*,

$$A p \cdot A p \geq \frac{1}{3} \|P\|^2.$$

Cela indique que la résolution devrait être efficace. On applique la méthode du *gradient conjugué*.

### 7.3.1 Rappels sur la méthode du gradient conjugué

Soit  $E$  un espace de Hilbert et  $q$  une forme quadratique définie positive sur  $E$ ,  $q(x) = Ax \cdot x$ . Alors  $x$  est solution de l'équation  $Ax = y$  si et seulement si  $x$  est le minimum de la fonction  $z \mapsto f(z) = \frac{1}{2}Az \cdot z - y \cdot z$ . La méthode consiste à chercher les minima  $x_r$  de la fonction  $f$  sur les sous-espaces vectoriels croissants

$$E_r = \text{vect}\{y, Ay, \dots, A^r y\}.$$

Alors  $x_r - x_{r-1}$  est donné par une formule simple au moyen de produits scalaires. La suite  $x_r$  converge vers la solution véritable  $x_\infty$ . En particulier, si  $E$  est de dimension  $N$ , alors  $x_N = x_\infty$ . L'expérience prouve que  $x_{N/5}$  est souvent déjà très proche de  $x_\infty$ .

**Proposition 25** *On définit trois suites  $x_r$ ,  $v_r$  et  $e_r$  par  $x_0 = 0$ ,  $v_0 = y$ , les relations de récurrence*

$$\begin{aligned} x_{r+1} &= x_r - \frac{e_r \cdot v_r}{Av_r \cdot v_r} v_r, \\ v_{r+1} &= e_{r+1} - \frac{Av_r \cdot e_{r+1}}{Av_r \cdot v_r} v_r \end{aligned}$$

et

$$e_r = Ax_r - y.$$

Alors la fonction  $f : x \mapsto \frac{1}{2}Ax \cdot x - y \cdot x$  atteint son minimum sur  $E_r$  en  $x_r$ . La suite  $x_r$  converge vers l'unique solution  $x_\infty$  de l'équation  $Ax = y$ . De plus, on a l'estimation a posteriori

$$A(x_\infty - x_r) \cdot (x_\infty - x_r) = e_r \cdot (x_{r+1} - x_r).$$

Si  $A$  est *coercive*, i.e.  $\|x\|^2 \leq C Ax \cdot x$ , on a l'estimation a posteriori

$$\|x_\infty - x_r\| \leq C \|Ax_r - y\|.$$

**Preuve.** Définissons  $x_r$  comme le point de  $E_r$  où la restriction de  $f$  à  $E_r$  atteint son minimum. Il existe et est unique par stricte convexité. Pour la même raison, si  $E_\infty$  est l'adhérence de la réunion des  $E_r$ , la fonction uniformément convexe  $f$  atteint son minimum sur  $E_\infty$  en un unique point  $x_\infty$ .

On utilise le produit scalaire associé à la forme quadratique  $q$ ,

$$a \cdot_A b = Aa \cdot b.$$

Relativement à cette structure euclidienne, les ensembles de niveau de  $f$  sont des sphères centrées en  $x_\infty$ . Les ensembles de niveau de la restriction de  $f$  à  $E_r$  sont des sphères centrées en  $x_r$ . Soit  $s \leq r - 1$ . Au point  $x_s$  où  $f$  atteint son minimum sur  $E_s$  cette sphère est tangente à  $E_s$ , donc  $x_r - x_s \perp_A E_s$ . En particulier  $x_\infty - x_s \perp_A E_s$ .

Posons  $e_r = Ax_r - y$ . Alors  $e_r \in E_{r+1}$ . Or  $e_r = A(x_r - x_\infty) \perp E_r$ , donc en particulier  $e_r \cdot e_{r-1} = 0$ .

Considérons la suite  $v_r$  définie par  $v_0 = y$  et par la relation de récurrence

$$v_{r+1} = e_{r+1} - \frac{v_r \cdot_A e_{r+1}}{v_r \cdot_A v_r} v_r.$$

Montrons par récurrence sur  $r$  que  $v_r \in E_{r+1}$ ,  $v_r \perp_A E_r$  et

$$x_{r+1} = x_r - \frac{e_r \cdot v_r}{Av_r \cdot v_r} v_r.$$

Lorsque  $r = 0$ ,  $v_0 = y \in E_1$  par définition et  $E_0 = \{0\}$ .

Supposons connu que  $v_{r-1} \in E_r$ ,  $v_{r-1} \perp_A E_{r-1}$  et

$$x_r = x_{r-1} - \frac{e_{r-1} \cdot v_{r-1}}{Av_{r-1} \cdot v_{r-1}} v_{r-1}.$$

On remarque que  $e_r \in AE_r \subset E_{r+1}$  donc  $v_r \in E_{r+1}$ .

D'autre part,  $x_\infty - x_r \perp_A E_r$ , donc  $e_r = A(x_\infty - x_r) \perp_A A^{-1}E_r$ . Or  $AE_{r-1} \subset E_r$  donc  $e_r \perp_A E_{r-1}$ . Comme  $v_r$  est un vecteur du plan engendré par  $e_r$  et  $v_{r-1}$ , il est orthogonal à  $E_{r-1}$ . Par construction,  $v_r \perp_A v_{r-1}$ .

Si  $v_{r-1} = 0$ , alors d'après l'hypothèse de récurrence,  $x_r = x_{r-1}$ , donc  $e_r = e_{r-1}$ . Comme  $e_r \cdot e_{r-1} = 0$ , nécessairement  $e_{r-1} = 0$ , donc  $x_\infty = x_{r-1}$ . On conclut que  $x_{r+1} = x_r$ ,  $v_r = 0$  et la relation de récurrence pour  $x_{r+1}$  est satisfaite.

On peut donc supposer que  $v_{r-1} \neq 0$ . Alors  $E_r$  est engendré par  $E_{r-1}$  et  $v_{r-1}$ , donc  $v_r \perp_A E_r$ .

Si  $v_r = 0$ , alors  $e_r$  est colinéaire à  $v_{r-1}$ . Or  $e_r \perp_A E_{r-1}$ . Nécessairement  $e_r = 0$ , donc  $x_{r+1} = x_r$  et la relation de récurrence pour  $x_{r+1}$  est à nouveau satisfaite.

On peut donc supposer que  $v_r \neq 0$ . Alors  $v_r$  est une base de l'orthogonal de  $E_{r-1}$  dans  $E_r$ , donc il existe un réel  $\lambda$  tel que  $x_{r+1} - x_r = \lambda v_r$ . Comme cette valeur de  $\lambda$  minimise  $f(x_r + \lambda v_r)$ , nécessairement

$$\lambda = -\frac{x_r \cdot_A v_r - y \cdot v_r}{v_r \cdot_A v_r},$$

comme annoncé.

Par construction, la distance (mesurée au moyen de  $x \mapsto AX \cdot x$ ) de  $x_\infty$  au sous-espace  $E_r$  est atteinte en  $x_r$ . Comme la réunion des  $E_r$  est dense dans  $E_\infty$ , cette distance, égale à  $A(x_\infty - x_r) \cdot (x_\infty - x_r)$ , tend vers 0. Remarquer que les vecteurs  $x_{r+1} - x_r$  sont deux à deux orthogonaux (ils sont colinéaires aux vecteurs obtenus en orthonormalisant la base  $y, Ay, \dots, A^r y \dots$ ). On a donc

$$\sum_{r=0}^{\infty} A(x_{r+1} - x_r) \cdot (x_{r+1} - x_r) = Ax_\infty \cdot x_\infty.$$

Enfin,

$$\begin{aligned} \|x_r - x_\infty\|^2 &\leq C A(x_r - x_\infty) \cdot (x_r - x_\infty) \\ &= (Ax_r - y) \cdot (x_r - x_\infty) \\ &\leq \|Ax_r - y\| \|x_r - x_\infty\| \end{aligned}$$

d'où

$$\|x_\infty - x_r\| \leq C \|Ax_r - y\|. \blacksquare$$

### 7.3.2 Méthode de Jacobi

On peut aussi utiliser la méthode de résolution de Jacobi (voir [?], Theorem 4.2 page 55). Soit  $D$  est la diagonale de  $A$ . Alors

$$Ax = y \Leftrightarrow x = -D^{-1}(A - D)x + D^{-1}y$$

et on itère la transformation affine  $x \mapsto -D^{-1}(A - D)x + D^{-1}y$ . Celle-ci est contractante lorsque  $\mathbf{R}^{n+1}$  est muni de la norme  $\ell^1$ . En effet,

$$q = \|D^{-1}(A - D)\|_{\ell^1 \rightarrow \ell^1} = \max\{|A_{jj}|^{-1} \sum_{i \neq j} |A_{ji}|; j = 0, \dots, m\}$$

qui dans le cas présent vaut  $q = 1/2$ . Posant  $x_0 = O$  et

$$x_{\nu+1} = -D^{-1}(A - D)x_{\nu} + D^{-1}y,$$

on peut affirmer que la solution  $x$  satisfait

$$\|x - x_{\nu}\|_{\ell^1} \leq \frac{q^{\nu}}{1 - q} \|D^{-1}y\|_{\ell^1} .$$

et dans le cas présent  $\frac{q^{\nu}}{1 - q} = 2^{1-\nu}$ .

# Chapter 8

## Synthèse d'images

### 8.1 Géométrie affine

Dans ce chapitre seulement, on met des flèches sur les vecteurs, pour bien distinguer les points des vecteurs.

#### 8.1.1 Points et vecteurs

Dans  $\mathbf{R}^n$  rapporté au repère  $(O, \vec{e}_1, \dots, \vec{e}_n)$ , un point  $p$  est représenté par la colonne de ses coordonnées

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

qui sont les composantes du vecteur  $\vec{Op}$ . Si  $p$  et  $q$  sont des points,  $\vec{v}$  un vecteur de composantes  $V$ ,  $a$  un réel, on note  $q - p = \vec{pq}$ ,  $r = p + v$  le point tel que  $\vec{pr} = \vec{v}$ ,  $ap + (1 - a)q$  le barycentre de  $(p, a)$  et  $(q, (1 - a))$ . La notation est justifiée par les calculs en coordonnées : si  $X$  (resp.  $Y$ , resp.  $V$ ) est la colonne des coordonnées de  $p$  (resp.  $q$ , resp. des composantes de  $\vec{v}$ ) dans un repère, alors la colonne des coordonnées de  $\vec{pq}$  est  $Y - X$ , celle de  $r$  est  $X + V$ , celle de  $ap + (1 - a)q$  est  $aX + (1 - a)Y$ .

Autrement dit, la différence de deux points est un vecteur, le barycentre de deux points est un point, on peut ajouter un vecteur et un point. Attention, on n'ajoute pas deux points.

#### 8.1.2 Notation matricielle des applications affines

**Définition 26** Une application  $f : \mathbf{R}^n \rightarrow \mathbf{R}^n$  est affine s'il existe une application linéaire  $L : \mathbf{R}^n \rightarrow \mathbf{R}^n$  telle que pour tous points  $p, q$  de  $\mathbf{R}^n$ ,

$$f(q) = f(p) + L(\vec{pq}).$$

L'endomorphisme  $L$  s'appelle l'application linéaire tangente à  $f$ .

Soit  $(O, \vec{e}_1, \dots, \vec{e}_n)$  un repère de  $\mathbf{R}^n$ . La matrice de  $f$  dans ce repère est

$$M_f = \begin{pmatrix} M_L & Of(\vec{O}) \\ 0 & 1 \end{pmatrix}$$

où  $M_L$  désigne la matrice de  $L$  dans la base  $(\vec{e}_1, \dots, \vec{e}_n)$ .

**Exemple.** Soit  $f$  la translation de vecteur  $\vec{v} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$  dans  $\mathbf{R}^3$ . Alors pour tous  $p, q \in \mathbf{R}^3$ ,  $f(q) - f(p) = q - p$ , donc  $f$  est affine d'application linéaire tangente l'identité. De plus,  $f(O) = O + \vec{v}$ .

Par conséquent

$$M_f = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

**Exemple.** Soit  $f$  l'homothétie de rapport  $\lambda$  et de centre  $p_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$  dans  $\mathbf{R}^3$ . Alors pour tous  $p, q \in \mathbf{R}^3$ ,  $f(q) - f(p) = \lambda(q - p)$ , donc  $f$  est affine d'application linéaire tangente  $\lambda id$ . De plus,  $f(p_0) = p_0$ , d'où  $f(O) = p_0 + \lambda(O - p_0)$ . Par conséquent

$$M_f = \begin{pmatrix} \lambda & 0 & 0 & (1 - \lambda)x_0 \\ 0 & \lambda & 0 & (1 - \lambda)y_0 \\ 0 & 0 & \lambda & (1 - \lambda)z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

**Proposition 27** Soit  $f : \mathbf{R}^3 \rightarrow \mathbf{R}^3$  une application affine,  $p$  un point de  $\mathbf{R}^3$  de coordonnées  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ ,  $p' = f(p)$  de coordonnées  $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$ . Alors

$$M_f \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}.$$

Par conséquent, si  $f$  et  $g$  sont des applications affines, alors

$$M_{f \circ g} = M_f M_g.$$

L'écriture matricielle des applications affines est évidemment commode pour manipuler informatiquement des points et des transformations.

**Exercice 4** Ecrire la matrice  $4 \times 4$  de la symétrie par rapport au plan passant par  $p_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$  et

orthogonal au vecteur unitaire  $\vec{v} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ .

**Solution de l'exercice 4.** Matrice  $4 \times 4$  d'une symétrie.

Soit  $\vec{u}$  un vecteur de  $\mathbf{R}^3$ . La projection de  $\vec{u}$  sur la droite vectorielle engendrée par  $\vec{n}$  est

$$(\vec{u} \cdot \vec{n})\vec{n}$$

donc la projection de  $\vec{u}$  sur le plan vectoriel orthogonal à  $\vec{n}$  est

$$u - (\vec{u} \cdot \vec{n})\vec{n}.$$

Par conséquent, la symétrie vectorielle par rapport au plan orthogonal à  $\vec{n}$  est

$$L(\vec{u}) = u - 2(\vec{u} \cdot \vec{n})\vec{n}.$$



Il vient

$$M_f = \begin{pmatrix} 1 - 2a^2 & -2ab & -2ac \\ -2ab & 1 - 2b^2 & -2bc \\ -2ac & -2bc & 1 - 2c^2 \end{pmatrix}$$

Par définition,  $f(p_0) = p_0$  donc  $f(O) = p_0 + L(O - p_0)$  d'où

$$M_f = \begin{pmatrix} 1 - 2a^2 & -2ab & -2ac & 2a^2x_0 + 2aby_0 + 2acz_0 \\ -2ab & 1 - 2b^2 & -2bc & 2abx_0 + 2b^2y_0 + 2bcz_0 \\ -2ac & -2bc & 1 - 2c^2 & 2acx_0 + 2bcy_0 + 2c^2z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \blacksquare$$

**Exercice 5** Ecrire la matrice  $4 \times 4$  de la rotation d'angle  $\theta$  et d'axe la droite  $D$  passant par

$$p_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \text{ et de vecteur directeur unitaire } \vec{v} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}.$$

**Solution de l'exercice 5.** Matrice  $4 \times 4$  d'une rotation.

Tourner un vecteur  $\vec{u}$  du plan orthogonal à  $\vec{n}$  de  $90^\circ$  consiste à remplacer  $\vec{u}$  par  $\vec{n} \wedge \vec{u}$ . Le tourner d'un angle  $\theta$  à remplacer  $\vec{u}$  par

$$\cos \theta \vec{u} + \sin \theta \vec{n} \wedge \vec{u}.$$

Par conséquent, la rotation vectorielle  $R$  d'angle  $\theta$  et d'axe  $\vec{n}$  est donnée par

$$\begin{aligned} R(\vec{u}) &= (\vec{u} \cdot \vec{n})\vec{n} + \cos \theta (u - (\vec{u} \cdot \vec{n})\vec{n}) + \sin \theta \vec{n} \wedge (u - (\vec{u} \cdot \vec{n})\vec{n}) \\ &= (1 - \cos \theta)(\vec{u} \cdot \vec{n})\vec{n} + \cos \theta u + \sin \theta \vec{n} \wedge u. \end{aligned}$$

La matrice de  $R$  est donc

$$\begin{pmatrix} a^2 + (1 - a^2) \cos \theta & ab(1 - \cos \theta) - c \sin \theta & ac(1 - \cos \theta) + b \sin \theta \\ ab(1 - \cos \theta) + c \sin \theta & b^2 + (1 - b^2) \cos \theta & bc(1 - \cos \theta) - a \sin \theta \\ ac(1 - \cos \theta) - b \sin \theta & bc(1 - \cos \theta) + a \sin \theta & c^2 + (1 - c^2) \cos \theta \end{pmatrix}.$$

Par définition,  $f(p_0) = p_0$  donc  $f(O) = p_0 + R(O - p_0)$  d'où

$$M_f = \begin{pmatrix} a^2 + (1 - a^2) \cos \theta & ab(1 - \cos \theta) - c \sin \theta & ac(1 - \cos \theta) + b \sin \theta & X \\ ab(1 - \cos \theta) + c \sin \theta & b^2 + (1 - b^2) \cos \theta & bc(1 - \cos \theta) - a \sin \theta & Y \\ ac(1 - \cos \theta) - b \sin \theta & bc(1 - \cos \theta) + a \sin \theta & c^2 + (1 - c^2) \cos \theta & Z \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

où

$$X = -(1 - a^2)(1 - \cos \theta)x_0 - (ab(1 - \cos \theta) - c \sin \theta)y_0 - (ac(1 - \cos \theta) + b \sin \theta)z_0,$$

$$Y = -(ab(1 - \cos \theta) + c \sin \theta)x_0 - (1 - b^2)(1 - \cos \theta)y_0 - (bc(1 - \cos \theta) - a \sin \theta)z_0,$$

$$Z = -(ac(1 - \cos \theta) - b \sin \theta)x_0 - (bc(1 - \cos \theta) + a \sin \theta)y_0 - (1 - c^2)(1 - \cos \theta)z_0. \blacksquare$$

## 8.2 Perspective

### 8.2.1 Vues

Une *vue* d'une scène 3D est déterminée par les données suivantes.

- La position de la caméra : un point  $C$  de coordonnées  $\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$ .

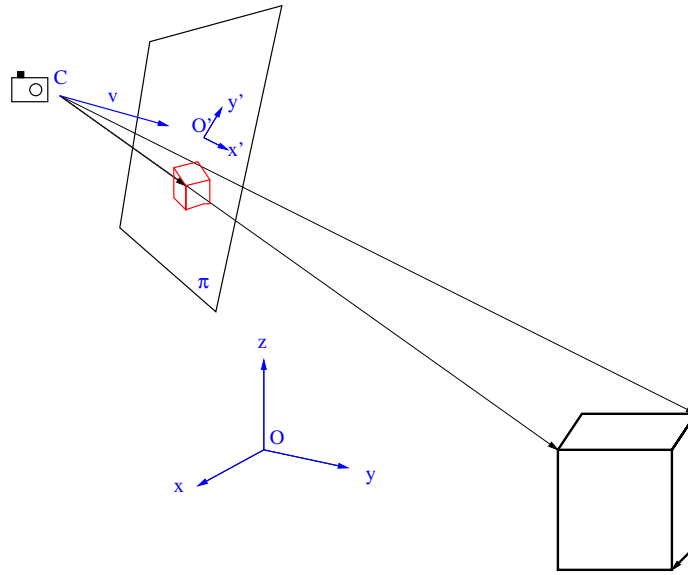


Figure 8.1:

- Une direction de visée : un vecteur unitaire  $\vec{v}$  de composantes  $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ .
- Un *écran*, i.e. un plan  $\Pi$  perpendiculaire à la direction de visée : il est déterminé par sa distance à  $C$ , un réel positif  $d$ .
- Un repère orthonormé  $(O', \vec{e}'_1, \vec{e}'_2)$  du plan  $\Pi$ .

Voici un moyen systématique de construire le repère  $(O', \vec{e}'_1, \vec{e}'_2)$ . Il suffit de choisir une fois pour toute un vecteur unitaire  $\vec{v}$  non colinéaire à  $\vec{v}$ . On prend pour  $O'$  la projection orthogonale de  $C$  sur  $\Pi$ , soit  $O' = C + d\vec{v}$ . On prend

$$\vec{e}'_1 = \frac{\vec{v} \wedge \vec{v}}{|\vec{v} \wedge \vec{v}|}$$

et

$$\vec{e}'_2 = \vec{v} \wedge \vec{e}'_1.$$

Une *vue* de la scène est une application  $W : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ , qui aux coordonnées d'un point dans le *repère du monde*  $(O, \vec{e}_1, \vec{e}_2, \vec{e}_3)$  associe les coordonnées de sa projection sur l'écran  $\Pi$ , dans le *repère de l'image*  $(O', \vec{e}'_1, \vec{e}'_2)$ .

**Définition 28** La vue en perspective depuis  $C$  sur l'écran  $\Pi$  consiste à projeter un point  $p$  sur  $p'$ , point d'intersection de la droite  $Cp$  avec  $\Pi$ .

**Remarque.** La projection n'est pas définie si  $p$  est dans le plan passant par  $C$  et parallèle à  $\Pi$ . C'est normal : on n'arrive pas à voir dans les directions situées à  $90^\circ$  de sa direction de vision.

**Proposition 29** On choisit  $\nu = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ . Les coordonnées de  $p'$  sont données, en fonction des

coordonnées  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  de  $p$ , par les formules

$$x' = \frac{-cd(x - x_0) + ad(z - z_0)}{\sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0))},$$

$$y' = \frac{abd(x - x_0) - d(a^2 + c^2)(y - y_0) + bcd(z - z_0)}{\sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0))}.$$

**Preuve.** Voir exercice 6. ■

**Exercice 6** On choisit  $\nu = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ . Calculer les coordonnées  $\begin{pmatrix} x' \\ y' \end{pmatrix}$  de  $p'$  en fonction des coordonnées  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  de  $p$ .

**Solution de l'exercice 6.** *La projection perspective.*

On travaille d'abord dans le repère du monde, supposé orthonormé.

Un point  $P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$  est dans  $\Pi$  si et seulement si  $O\vec{P} \cdot \vec{\nu} = 0$ , i.e. si  $a(X - x_0 - da) + b(Y - y_0 - db) + c(Z - z_0 - dc) = 0$ , autrement dit,  $a(X - x_0) + b(Y - y_0) + c(Z - z_0) = d$ .

On paramètre la droite  $Cp$  par

$$t \mapsto c(t) = (1 - t)C + tp = \begin{pmatrix} (1 - t)x_0 + tx \\ (1 - t)y_0 + ty \\ (1 - t)z_0 + tz \end{pmatrix}.$$

Le point  $c(t)$  est dans  $\Pi$  si et seulement si

$$t = \frac{d}{a(x - x_0) + b(y - y_0) + c(z - z_0)}.$$

Par conséquent, les coordonnées de la projection  $p'$  dans le repère du monde sont

$$\begin{pmatrix} x_0 + \frac{d(x - x_0)}{a(x - x_0) + b(y - y_0) + c(z - z_0)} \\ y_0 + \frac{d(y - y_0)}{a(x - x_0) + b(y - y_0) + c(z - z_0)} \\ z_0 + \frac{d(z - z_0)}{a(x - x_0) + b(y - y_0) + c(z - z_0)} \end{pmatrix}.$$

On calcule les composantes des vecteurs du repère de l'image.

$$\frac{\vec{\nu} \wedge \vec{\nu}}{=} \begin{pmatrix} -c \\ 0 \\ a \end{pmatrix},$$

d'où

$$\vec{e}_1 = \begin{pmatrix} -\frac{c}{\sqrt{a^2 + c^2}} \\ 0 \\ \frac{a}{\sqrt{a^2 + c^2}} \end{pmatrix}$$

et

$$\vec{e}_2 = \frac{\vec{\nu} \wedge \vec{e}_1}{=} \begin{pmatrix} \frac{ab}{\sqrt{a^2 + c^2}} \\ -\sqrt{a^2 + c^2} \\ \frac{bc}{\sqrt{a^2 + c^2}} \end{pmatrix}$$

Les coordonnées de  $p'$  dans le repère de l'image s'obtiennent par

$$x' = O'\vec{p}' \cdot \vec{e}_1' = \frac{-cd(x - x_0) + ad(z - z_0)}{\sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0))},$$

$$y' = O'\vec{p}' \cdot \vec{e}_2' = \frac{abd(x - x_0) - d(a^2 + c^2)(y - y_0) + bcd(z - z_0)}{\sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0))}. \blacksquare$$

**Remarque.** Lorsque  $C$ ,  $d$  et  $v$  varient, la méthode ci-dessus ne fournit tous les angles de prise de vue de la scène possibles. Pour les avoir tous, il faut encore faire tourner la caméra autour de la direction de visée, i.e. remplacer  $(\vec{e}_1', \vec{e}_2')$  par

$$(\cos \psi \vec{e}_1' + \sin \psi \vec{e}_2', -\sin \psi \vec{e}_1' + \cos \psi \vec{e}_2')$$

pour  $\psi \in [0, 2\pi]$ .

## 8.2.2 Coordonnées homogènes

On constate (proposition 29), que la vue en perspective s'exprime par des fractions rationnelles

$$x' = \frac{-cd(x - x_0) + ad(z - z_0)}{\sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0))}, \quad y' = \frac{abd(x - x_0) - d(a^2 + c^2)(y - y_0) + bcd(z - z_0)}{\sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0))}.$$

Complétons le repère de l'image en un repère orthonormé de  $\mathbf{R}^3$  en posant  $\vec{e}_3' = \vec{v}$ . La troisième coordonnée de  $p'$  est évidemment  $z' = 0$ .

On ramène ces expressions rationnelles à des expressions linéaires en adoptant la convention suivante : Un point  $p$  de  $\mathbf{R}^3$  peut être représenté, non seulement par  $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ , mais par n'importe quel vecteur de  $\mathbf{R}^4$  qui lui est proportionnel. En particulier, un point  $p'$  du plan  $\Pi$  peut être représenté par n'importe quel vecteur de  $\mathbf{R}^3$  proportionnel à  $\begin{pmatrix} x' \\ y' \\ 0 \\ 1 \end{pmatrix}$ .

Avec cette convention, on peut choisir pour représentant de la vue en perspective  $p'$  d'un point  $p$  le vecteur

$$\begin{pmatrix} -cd(x - x_0) + ad(z - z_0) \\ abd(x - x_0) - d(a^2 + c^2)(y - y_0) + bcd(z - z_0) \\ 0 \\ \sqrt{a^2 + c^2}(a(x - x_0) + b(y - y_0) + c(z - z_0)) \end{pmatrix}.$$

Ce vecteur est l'image du vecteur  $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$  par l'endomorphisme de  $\mathbf{R}^4$  de matrice

$$\begin{pmatrix} -cd & 0 & ad & cdx_0 - adz_0 \\ abd & -d(a^2 + c^2) & bcd & -abd x_0 + d(a^2 + c^2)y_0 - bcd z_0 \\ 0 & 0 & 0 & 0 \\ a\sqrt{a^2 + c^2} & b\sqrt{a^2 + c^2} & c\sqrt{a^2 + c^2} & \sqrt{a^2 + c^2}(-ax_0 - by_0 - cz_0) \end{pmatrix}$$

C'est quand même plus sympathique (et commode à implémenter informatiquement) de passer par une matrice pour représenter la vue en perspective.

**Définition 30** Soit  $p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$  un point de  $\mathbf{R}^3$ . On appelle coordonnées homogènes de  $p$  tout quadruplet  $(u, v, w, t)$  proportionnel à  $(x, y, z, 1)$ . On note  $p = [x; y; z; t]$ .

Une application  $f : \mathbf{R}^3 \rightarrow \mathbf{R}^3$  (éventuellement définie seulement en dehors d'un hyperplan affine) est dite projective s'il existe une application linéaire  $L : \mathbf{R}^4 \rightarrow \mathbf{R}^4$  telle que, pour tout

$$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbf{R}^3, \text{ d'image } f(p) = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix},$$

$$L \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \text{ est proportionnel à } \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Autrement dit, en coordonnées homogènes,  $f$  est donnée par une matrice  $4 \times 4$ .

**Remarque.** La définition précédente s'étend à toutes les dimensions.

**Exemple.** Toute application affine est en particulier projective.

**Exemple.** La vue en perspective est une application projective, définie en dehors de l'hyperplan parallèle à l'écran passant par la caméra, non surjective (son image est contenue dans l'écran).

**Exemple.** Soient  $\Pi_1$  et  $\Pi_2$  deux plans affines dans  $\mathbf{R}^3$ ,  $C$  un point qui n'est pas dans  $\Pi_2$ . Considérons la restriction  $f$  à  $\Pi_1$  de la vue en perspective sur  $\Pi_2$ . Après choix de coordonnées cartésiennes sur  $\Pi_2$  et  $\Pi_1$ , cette application devient une application projective de  $\mathbf{R}^2$  dans  $\mathbf{R}^2$ . Elle est définie en dehors d'une droite  $D_1$ , l'intersection de  $\Pi_1$  avec le plan parallèle à  $\Pi_2$  passant par  $C$ . Son image est le complémentaire d'une droite  $D_2$ , l'intersection de  $\Pi_2$  avec le plan passant par  $C$  parallèle à  $\Pi_1$ . Si  $C$  n'est pas sur  $\Pi_1$ ,  $f$  est une bijection de  $\Pi_1 \setminus D_1$  sur  $\Pi_2 \setminus D_2$ . Dans ce cas, la bijection réciproque est la vue en perspective de  $\Pi_2$  sur l'écran  $\Pi_1$  depuis  $C$ .  $f$  est affine si et seulement si  $\Pi_1$  et  $\Pi_2$  sont parallèles.

**Exercice 7** Soit  $\Pi_1$  le plan passant par  $O'_1 = O$  et de vecteurs directeurs  $\vec{u}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$  et  $\vec{v}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ .

Soit  $\Pi_2$  le plan passant par  $O'_2 = O$  et de vecteurs directeurs  $\vec{u}_2 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$  et  $\vec{v}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ . Soit

$C = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$ . Calculer la matrice  $3 \times 3$  qui représente la vue en perspective de  $\Pi_1$  sur  $\Pi_2$  depuis  $C$ .

Vérifier qu'elle est inversible.

**Solution de l'exercice 7.** Vue en perspective d'un plan.

Soit  $p_1 \in \Pi_1$ , de coordonnées  $(x_1, y_1)$  dans le repère  $(O_1, \vec{u}_1, \vec{v}_1)$ . Alors  $p_1 = O + x_1 \vec{v}_1 + y_1 \vec{u}_1$  a pour coordonnées  $\begin{pmatrix} x_1 \\ y_1 \\ x_1 \end{pmatrix}$  dans le repère du monde. De même, un point  $p_2 \in \Pi_2$ , de coordonnées

$(x_2, y_2)$  dans le repère  $(O_2, \vec{u}_2, \vec{v}_2)$  a pour coordonnées  $\begin{pmatrix} x_2 \\ y_2 \\ -x_2 \end{pmatrix}$  dans le repère du monde. L'équation

de  $\Pi_2$  est  $X + Z = 0$ . Un point courant

$$(1-t)C + tp_1 = \begin{pmatrix} tx_1 \\ ty_1 \\ tx_1 + 2 - 2t \end{pmatrix}$$

sur la droite  $Cp_1$  est dans  $\Pi_2$  si et seulement si  $2tx_1 + 2 - 2t = 0$ , d'où  $t = 1/1 - x_1$  et les coordonnées de  $p_2) = f(p_1)$  sont

$$x_2 = \frac{x_1}{1 - x_1} \quad \text{et} \quad y_2 = \frac{y_1}{1 - x_1}.$$

On constate que  $\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}$  est proportionnel à  $\begin{pmatrix} x_1 \\ y_1 \\ 1 - x_1 \end{pmatrix}$  qui est l'image de  $\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$  par la matrice  $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$ . On conclut que  $f$  est projective. Sa matrice a pour déterminant 1 donc elle est inversible. ■

### 8.2.3 Transformations projectives

Une *transformation projective* de  $\mathbf{R}^n$ , c'est une application projective dont la matrice est inversible. On montre aisément que deux matrices inversibles de taille  $n+1$  définissent la même transformation de  $\mathbf{R}^n$  si et seulement si elles sont proportionnelles. D'autre part, si  $L$  et  $L'$  sont des matrices associées à des transformations  $f$  et  $f'$ , l'application projective associée à  $LL'$  est  $f \circ f'$ . Par conséquent, les transformations projectives de  $\mathbf{R}^n$  forment un groupe isomorphe à  $Gl(n+1, \mathbf{R}) / \sim$  où  $L \sim L'$  s'il existe un réel non nul  $\lambda$  tel que  $L' = \lambda L$ . Ce groupe quotient, noté  $PGL(n+1, \mathbf{R})$ , a pour dimension  $(n+1)^2 - 1$ . Par exemple, si  $n = 3$ , une transformation projective dépend de 15 paramètres, alors qu'une transformation affine dépend de seulement 12 paramètres.

### 8.2.4 Vues en perspective d'objets

**Exercice 8** Vérifier que, vue en perspective, une droite reste en général une droite. Quelles sont les exceptions ? Montrer que, vues en perspectives, les droites parallèles à une droite  $D$  sont en général concourantes en un point appelé point de fuite de  $D$ . Quelles sont les exceptions ? Où se trouvent les points de fuites des droites contenues dans un même plan ?

**Solution de l'exercice 8.** *Vue en perspective de droites.*

Si  $D$  est une droite qui ne passe pas par la caméra  $C$ , sa vue en perspective est l'intersection du plan contenant  $D$  et  $C$  avec l'écran  $\Pi$ . C'est donc une droite. Si  $D$  passe par  $C$ , elle est vue comme un point. Si  $D$  passe par  $C$  et est parallèle à  $\Pi$ , elle n'est pas vue du tout.

Si  $D$  et  $D'$  sont des droites parallèles ne passant pas par  $C$  et non parallèles à  $\Pi$ , les plans  $Q$  et  $Q'$  qu'elles déterminent avec  $C$  se coupent suivant une troisième droite  $D''$  parallèle à  $D$  et à  $D'$ , qui passe par  $C$ . Comme  $D''$  n'est pas parallèle à  $\Pi$ , la vue en perspective de  $D''$  est un point  $p$  par lequel passent les vues en perspective de  $D$  et  $D'$ . Vues en perspective, toutes les droites parallèles à  $D$  passent donc par  $p$ . Si  $D$  est parallèle à  $\Pi$ ,  $D''$  ne coupe pas  $\Pi$ , donc les vues en perspective de  $D$  et  $D'$  sont parallèles.

Si  $D$  est contenue dans un plan  $\Pi'$  non parallèle à  $\Pi$ , la droite parallèle à  $D$  passant par  $C$  est contenue dans le plan  $\Pi''$  parallèle à  $\Pi'$  passant par  $C$ , donc le point de fuite  $p$  de  $D$  est contenu dans la droite intersection de  $\Pi''$  et de  $\Pi$ . Les points de fuites des droites de  $\Pi'$  sont donc tous alignés. ■

**Remarque.** Dans une ville, les immeubles sont souvent des parallélépipèdes aux arêtes parallèles à trois directions fixées. Vues en perspective, ces trois familles d'arêtes sont portées par des droites concourantes en trois points. Si la direction de visée est horizontale, les arêtes verticales restent parallèles, et il ne reste plus que deux points de concours nettement visibles sur l'image : on parle de *perspective à deux points de fuite*.

**Exercice 9** Soit  $D_+$  une demi-droite,  $S$  un segment de droite. Qu'est ce qu'ils donnent, vus en perspective ? Attention, il y a de multiples cas de figure.

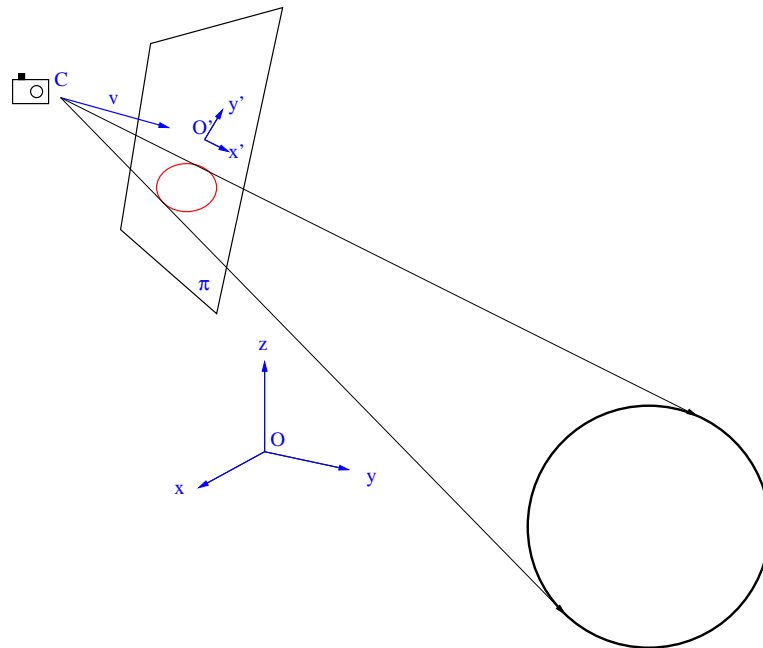


Figure 8.2:

**Solution de l'exercice 9.** *Vue en perspective d'une demi-droite ou d'un segment de droite.*

Soit  $D$  la droite. Si  $D$  est parallèle à  $\Pi$ , la projection sur  $\Pi$  est affine, donc envoie une demi-droite (resp. un segment) porté par  $D$  sur une demi-droite (resp. un segment).

Supposons que  $D$  n'est pas parallèle à  $\Pi$ . Vue en perspective, c'est une droite, mais un point qui tend à l'infini sur  $D$  tend en perspective vers le point de fuite de  $D$ . D'autre part, il y a un point de  $D$  invisible en perspective, c'est le point d'intersection  $q$  de  $D$  avec le plan parallèle à  $\Pi$  passant par  $C$ . Quand on s'approche de  $q$  sur  $D$ , on part à l'infini en perspective. L'image d'un segment de  $D$  qui ne contient pas  $q$  est donc un segment. L'image d'un segment de  $D$  d'extrémité  $q$  est une demi-droite. L'image d'un segment qui admet  $q$  comme point intérieur est la réunion de deux demi-droites. L'image d'une demi-droite qui ne contient pas  $q$  est un segment d'extrémité le point de fuite. L'image d'une demi-droite d'origine  $q$  est une demi-droite d'origine le point de fuite. L'image d'une demi-droite dont l'intérieur contient  $q$  est la réunion de deux demi-droites dont l'une a pour extrémité le point de fuite. ■

**Remarque.** On a décrit une vue comme la projection sur l'écran de tout l'espace. En réalité, une vue ne montre que ce qui se trouve devant l'écran. La vue en perspective réelle d'une droite est donc la projection d'une demi-droite ne coupant pas le plan parallèle à l'écran passant par la caméra : on voit un segment dont les extrémités sont le point de fuite et l'intersection de la droite avec l'écran.

**Exercice 10** *A quoi ressemble une sphère vue en perspective ?*

**Solution de l'exercice 10.** *Vue en perspective d'une sphère.*

La réunion des droites passant par  $C$  et coupant la sphère  $S$  est un cône de révolution  $K$  dont l'axe passe par  $C$  et me centre de  $S$ . Si  $\Pi$  est orthogonal à l'axe (i.e. si le centre est sur l'axe de visée),  $K \cap \Pi$  est un disque. Sinon, c'est une ellipse dont le grand axe est dans le plan contenant l'axe de visée et le centre de  $S$ . L'ellipse est d'autant plus allongée que  $S$  est loin de l'axe de visée, cette distance étant rapportée à la distance de  $S$  à la caméra. ■

**Remarque.**

Lorsqu'on programme une vue en perspective d'une scène, attention aux unités de longueur. La caméra doit être placée loin de la scène, sinon les distorsions (sphères qui deviennent des ellipses) seront trop visibles.

## 8.3 Couleur

### 8.3.1 Nature physique de la lumière

La lumière est une onde électromagnétique. On peut la décomposer comme combinaison d'ondes monochromatiques, chacune caractérisée par une *longueur d'onde*, un réel positif. Autrement dit, l'espace des ondes lumineuses est un espace vectoriel de dimension infinie. Le comportement de la lumière lorsqu'elle interagit avec la matière dépend de la longueur d'onde. Par exemple, la vitesse de propagation d'une onde dans un milieu transparent dépend de la longueur d'onde, c'est l'origine du phénomène de *dispersion* : au passage d'un milieu transparent à un autre, différentes longueurs d'ondes sont réfractées selon des angles différents, et une lumière blanche incidente se sépare en un faisceau comportant toutes les couleurs de l'arc-en-ciel.

### 8.3.2 Perception de la couleur

L'appareil visuel humain est limité : il ne sait pas distinguer les différentes longueurs d'ondes. Des expériences ont montré que pour construire presque toutes les couleurs que l'appareil visuel humain sait distinguer, il suffit de combiner trois lumières monochromatiques. Toutefois, il n'y a pas de choix de ces lumières monochromatiques de base qui soit plus naturel qu'un autre, du point de vue physiologique (lois de Grassmann).

Mathématiquement, on traduit cela par la modélisation suivante :

- L'espace des couleurs perceptibles est une variété de dimension 3.
- Cet espace ne possède pas de système de coordonnées meilleur qu'un autre.

Pour manipuler des couleurs, on choisit des coordonnées sur l'espace des couleurs perceptibles.

- **Synthèse additive.** Les teintes primaires additives sont le rouge, le vert et le bleu. Dans ce système, on décrit donc une teinte par trois réels  $(r, v, b)$  variant chacun entre 0 et 1. Ce système est bien adapté au pilotage des moniteurs d'ordinateurs, formés de la juxtaposition de trois types de diodes électroluminescentes émettant respectivement une lumière à dominante rouge, verte et bleue respectivement. La lumière blanche est modélisée par une valeur égale et maximale de rouge, de vert et de bleu, i.e.  $r = v = b = 1$ .
- **Synthèse soustractive.** Les teintes primaires soustractives sont le cyan, le magenta et le jaune. Dans ce système, on décrit donc une teinte par trois réels positifs  $(c, m, j)$ . Ce système est bien adapté au pilotage des imprimantes couleur. En effet celles-ci appliquent sur la papier trois encres qui absorbent respectivement le rouge, le vert et le bleu. La lumière réfléchi par le papier imprimé d'une couleur primaire prend donc la teinte complémentaire cyan, magenta ou jaune. L'absence de lumière (noir) est modélisé par une valeur égale et maximale de cyan, de magenta et de jaune, i.e.  $c = m = j = 1$ .
- Pour une normalisation plus précise des couleurs, d'autres systèmes de coordonnées sont utilisés, voir [PGMR], p. 220.

Le changement de coordonnées RVB  $\rightarrow$  CMJ est affine. Si une même teinte est représentée respectivement par  $(r, v, b)$  et  $(c, m, j)$ , alors

$$c = 1 - r, \quad m = 1 - v, \quad j = 1 - b.$$

**Remarque.** Les modélisations RVB et CMJ sont imparfaites. Deux moniteurs de fabrications différentes ne rendront pas les couleurs RVB de façon identique. Les encres étant ce qu'elles sont,



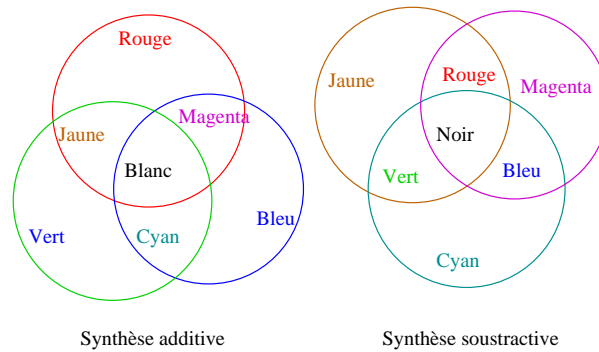


Figure 8.3:

un mélange égal de cyan, magenta et jaune donne en général du marron et non du noir (on ajoute donc une quatrième coordonnée N pour compléter). Les encres n'étant pas parfaites, un tel mélange donne

Deux vecteurs  $(r, v, b)$  proportionnels correspondent en gros à la même teinte mais à une clarté différente : l'énergie lumineuse émise par un moniteur réglé sur  $(tr, tv, tb)$ ,  $t > 1$ , est plus grande que celle émise sous  $(r, v, b)$ . Par exemple, les différents niveaux de gris sont modélisés par  $(t, t, t)$ ,  $t$  variant de 0 (noir) à 1 (blanc). Cela conduit à la définition simpliste mais commode suivante.

**Définition 31** La clarté d'une couleur  $(r, v, b)$  est  $\sqrt{r^2 + v^2 + b^2}$ .

**Exercice 11** Paramétrer (i.e. donner les fonctions  $r(x, y)$ ,  $v(x, y)$ ,  $b(x, y)$ ) un dégradé sur  $[0, 1]^2$ , allant du vert à gauche au magenta à droite, et de clarté croissant de 0 en bas à 1 en haut.

**Solution de l'exercice 11.** Dégradé.

Le vert, c'est  $(r, v, b) = (0, 1, 0)$ , le magenta  $(r, v, b) = (1, 0, 1)$ . Aller du vert à gauche au magenta à droite, c'est poser (par exemple)

$$(r, v, b) = a(x, y)(x, 1 - x, x)$$

où la fonction  $a$  permet d'ajuster la clarté. En  $y = 0$ , la clarté est nulle, donc  $a(x, 0) = 0$ . En  $y = 1$ , on veut une clarté de 1, soit  $a(x, 1)\sqrt{x^2 + (1 - x)^2 + x^2} = 1$ . On pose donc

$$a(x, y) = \frac{y}{\sqrt{3x^2 + -2x + 1}}$$

d'où

$$r(x, y) = \frac{xy}{\sqrt{3x^2 + -2x + 1}}, \quad v(x, y) = \frac{(1 - x)y}{\sqrt{3x^2 + -2x + 1}}, \quad b(x, y) = \frac{xy}{\sqrt{3x^2 + -2x + 1}}. \quad \blacksquare$$

## 8.4 Elimination des parties cachées

Supposons donnée une scène 3D dont les objets ont une couleur et une teinte connue. Autrement dit, à chaque point de la scène est associé un vecteur  $(r, v, b)$ . On rencontre ce cas de figure par exemple dans la représentation d'une surface mathématique par un logiciel de calcul (e.g. maple, matlab).

On prend une vue en perspective de la scène, depuis une caméra placée en  $C$ , sur un écran  $\Pi$ . Construire l'image consiste à affecter un vecteur  $(r, v, b)$  à chaque pixel du plan  $\Pi$ . L'élimination des parties cachées consiste à appliquer la règle suivante.

**Définition 32** Eliminer les parties cachées consiste à déterminer, pour chaque point  $p'$  de l'écran  $\Pi$ , parmi les bords d'objets de la scène qui coupent la demi-droite  $Cp'$ , celui qui se trouve le plus près de  $p'$ , et à affecter à  $p'$  la couleur du point d'intersection  $p$  de ce bord d'objet avec  $Cp'$ , soit

$$r(p') = r(p), \quad v(p') = v(p), \quad b(p') = b(p).$$

### 8.4.1 Algorithme : tampon de profondeur

On paramètre la demi-droite  $Cp'$  par  $Z \mapsto C + Z\vec{Cp}'$ . Au cours de l'algorithme, on maintient deux variables

- une valeur courante de  $Z$ , notée  $Z_{cour}$  ;
- une valeur courante du vecteur couleur, noté  $I_{cour}$ .

Il faut se donner une profondeur maximum d'exploration,  $Z_{max}$ , et la couleur du fond,  $I_{fond}$ . La couleur affectée à un pixel  $p'$  est la valeur de  $I_{cour}$  en fin de boucles.

Pour chaque pixel  $p' \in \Pi$ .

- Initialiser  $Z_{cour} = Z_{max}$  et  $I_{cour} = I_{fond}$ .
- Pour chaque bord d'objet  $\beta$  de la scène.
  - Trouver  $Z_\beta$  tel que  $C + Z_\beta\vec{Cp}' \in \beta$
  - Test :
    - \* si  $Z_\beta < Z_{cour}$ , mettre à jour  $Z_{cour} = Z_\beta$ ,  $I_{cour} = I_\beta = I_{C+Z_\beta\vec{Cp}'}$
    - \* si  $Z_\beta \geq Z_{cour}$ , ne rien faire.

### 8.4.2 Avantages et limitations

**Avantages.**

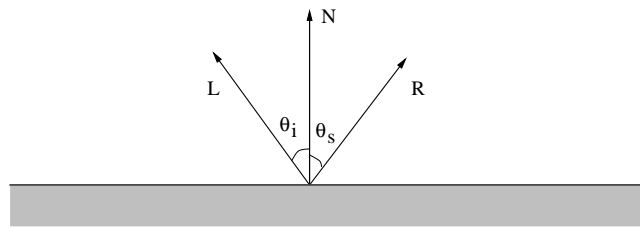
- Simplicité.
- Pas de limite aux nombre d'objets traités.
- Insensible à l'ordre dans lequel les objets sont traités.
- Pas de cas particuliers (intersections d'objets...).

**Limitations**

- Pas de gestion native des ombres.
- Traitement difficile des transparences, réflexions.
- Antialiasage difficile.

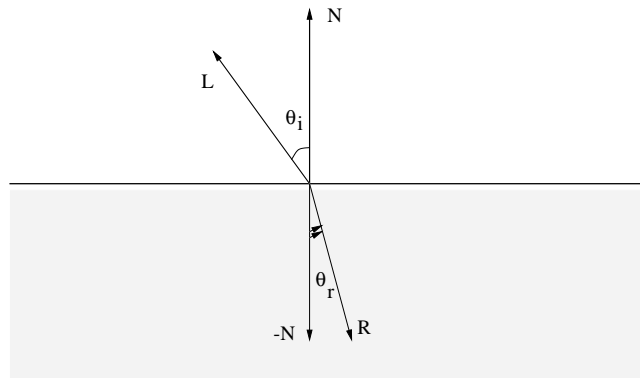
## 8.5 Tracé de rayons

C'est une élaboration de la méthode d'élimination des parties cachées qui effectue le calcul de la couleur en tenant compte des lois de la réflexion et de la réfraction.



Loi de Descartes

Figure 8.4:



Loi de Snell

Figure 8.5:

### 8.5.1 Réflexion et réfraction

- **Réflexion spéculaire : loi de Descartes.** Un rayon dit *incident* arrive d'une source ponctuelle  $S$  en un point  $p$  d'une surface parfaitement réfléchissante, de normale sortante  $\vec{N}$ . On note  $\vec{L}$  le vecteur unitaire de  $p$  vers la source,  $\vec{R}$  le vecteur unitaire parallèle au rayon réfléchi. Alors

$$\vec{R} \text{ est le symétrique de } \vec{L} \text{ par rapport à } \vec{N}.$$

Autrement dit, l'angle de réflexion spéculaire  $\theta_s = \cos^{-1}(\vec{R} \cdot \vec{N})$  est égal à l'angle d'incidence  $\theta_i = \cos^{-1}(\vec{L} \cdot \vec{N})$ .

- **Réfraction : loi de Snell.** Le rayon *incident* arrive d'une source ponctuelle  $S$  en un point  $p$  de l'interface entre deux milieux transparents, de normale sortante  $\vec{N}$ . On note  $\vec{L}$  le vecteur unitaire de  $p$  vers la source,  $\vec{R}$  le vecteur unitaire parallèle au rayon réfracté. On note  $\theta_i = \cos^{-1}(\vec{L} \cdot \vec{N})$  l'angle d'incidence et  $\theta_r = \cos^{-1}(\vec{R} \cdot -\vec{N})$  l'angle de réfraction. Enfin, on note  $n_i$  et  $n_r$  les indices de réfraction des deux milieux. Alors

$$n_i \sin \theta_i = n_r \sin \theta_r.$$

Si  $n_i \sin(\theta_i)/n_r > 1$ , il n'y a pas de rayon réfracté : c'est la réflexion totale, conformément à la loi de Descartes.

### 8.5.2 Principe de la méthode

De nouveau, on prend une vue en perspective de la scène, depuis une caméra placée en  $C$ , sur un écran  $\Pi$ . Il s'agit de calculer le vecteur couleur affecté à chaque pixel  $p' \in \Pi$ .

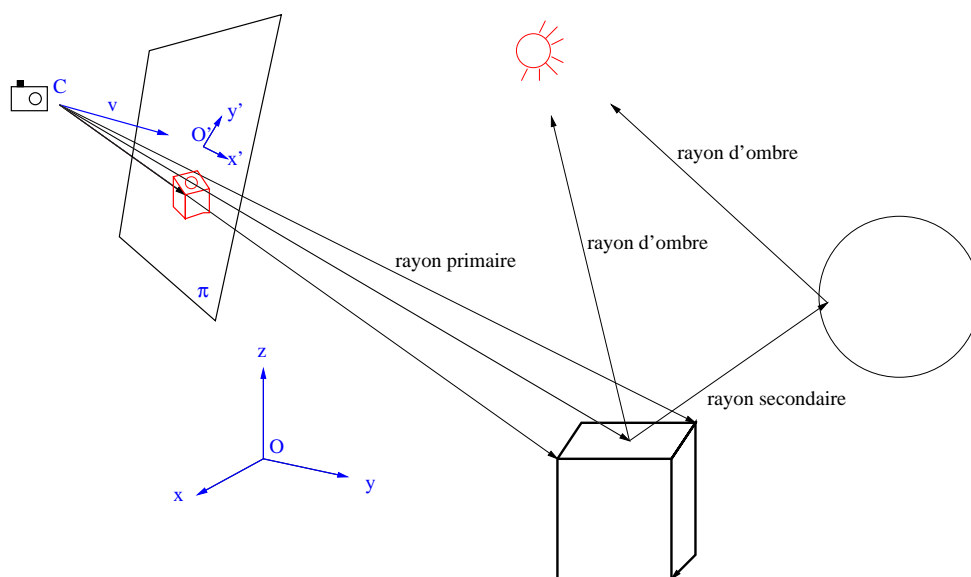


Figure 8.6:

**Définition 33** Le tracé de rayons (*ray-tracing* ou *ray-casting*) consiste à lancer, pour chaque point  $p'$  de l'écran  $\Pi$ , un rayon issu de la caméra passant par  $p'$  (appelé rayon primaire), à trouver le premier parmi les bords d'objets de la scène qu'il rencontre, et à affecter à  $p'$  une couleur qui dépend

- de l'éclairage ambiant ;
- de l'éclairage en provenance d'une source ;
- de l'éclairage en provenance d'un autre objet situé dans la direction du rayon réfléchi ou du rayon réfracté (appelés rayons secondaires) ;
- des caractéristiques de l'objet

selon le modèle d'éclairage choisi.

**Définition 34** Un modèle d'éclairage est une fonction  $ME$  qui prend comme arguments un point situé à la surface d'un objet, une direction d'observation, des couleurs, l'éclairage ambiant et les éclairages provenant des objets et des sources, les directions et distances des objets et des sources, et rend une couleur, l'éclairage émis par l'élément de surface dans la direction considérée.

On détaillera un exemple de modèle d'éclairage dans la section 8.6.

**Remarque.** Chaque objet en reflétant d'autres, on est amené à construire un *arbre de rayons*, dont la racine est le rayon primaire, qui engendre des rayons secondaires, qui engendrent des rayons tertiaires, etc.. et dont les feuilles sont les rayons d'ombre.

### 8.5.3 Algorithme

On se donne

- un flottant, la profondeur d'exploration maximale  $Z_{max}$  ;
- un entier, le nombre maximal de réflexions à prendre en compte  $N_{max}$  ;

- une couleur, l'éclairement ambiant  $I_{ambient}$  ;
- un modèle d'éclairement  $ME$ .

L'algorithme utilisé pour le traitement de chaque pixel présente deux étapes.

La première construit l'arbre des rayons, de la racine (rayon primaire) aux feuilles (rayons d'ombre). La seconde calcule la couleur du pixel courant en remontant l'arbre des feuilles à la racine.

Lors de la première étape, on lance d'abord le rayon primaire de la caméra à travers le pixel courant, il coupe le premier objet qu'il rencontre en un point  $p$  de la scène. De  $p$ , on lance des rayons dits *rayons d'ombre* vers les sources. D'autre part, on applique rigoureusement les lois de Descartes et de Snell pour lancer de  $p$  des rayons secondaires vers des objets. Autrement dit, chaque noeud (rayon incident) engendre  $n_p + 1$  (ou  $n_p + 2$ ) rayons fils suivant que l'objet rencontré est transparent ou non, où  $n_p$  est le nombre de sources visibles de  $p$ . On s'arrête à la  $N_{max}$ -ème génération.

Lors de la seconde étape, on calcule un éclairement pour chaque noeud de profondeur  $k$ , par valeurs décroissantes de  $k$ , i.e. en commençant par les feuilles. Pour cela on évalue le modèle  $ME$  sur les arguments suivants.

- La direction d'émission est la direction du rayon incident.
- Pour une feuille (source), le modèle  $ME$  fournit une intensité, dépendant éventuellement de la direction et de la distance.
- Pour un noeud interne de profondeur  $k$ , le modèle prend en compte les éclaircements calculés sur les noeuds fils, leur direction et éventuellement leur distance.
- L'éclairement ambiant  $I_{ambient}$ .

La couleur affectée au pixel courant est l'éclairement obtenu à la racine.

#### 8.5.4 Avantages et limitations

**Avantages.**

- Simplicité.
- Pas de limite au nombre d'objets traités : la complexité croît linéairement en fonction du nombre d'objets.
- Insensible à l'ordre dans lequel les objets sont traités.
- Réalisme des images obtenues : de nombreux phénomènes optiques peuvent être pris en compte.

**Limitations**

- Coût en CPU proportionnel au nombre de réflexions prises en compte.
- Antialiasage difficile.

#### 8.5.5 Lissage

Quand le bord de l'objet est polygonal, la fonction éclairement calculée par le lancer de rayons est discontinue, ce qui rend les facettes visibles. La raison est que la normale à l'objet est discontinue.

Il convient donc, préalablement au tracé de rayon, de *lisser* la normale. Voici comment Phong procède.

- Calculer les normales aux facettes.

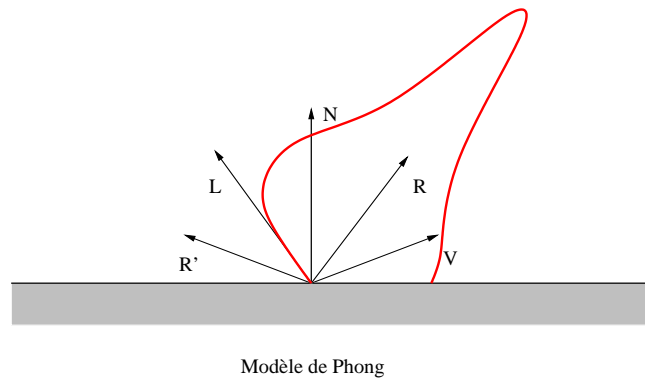


Figure 8.7:

- Définir la normale en un sommet comme la moyenne des normales aux facettes incidentes.
- Définir la normale en un point  $p$  d'une facette comme la moyenne des normales aux sommets de la facette, pondérés par les coordonnées barycentriques de  $p$ .

### 8.5.6 Au delà du tracé de rayon

Il existe des algorithmes d'éclairage totalement différents du tracé de rayons. Les méthodes de radiositè consistent à écrire les équations de conservation de l'énergie. L'inconnue est la *radiositè*, flux énergétique émis par un élément de surface. Voir [PGMR], chapitre 10.

## 8.6 Modèles d'éclairage

Un modèle d'éclairage est nécessaire pour calculer l'influence de phénomènes optiques (diffusion, réflexion, réfraction) sur la couleur et la clarté d'une scène. Il est sensé incorporer des caractéristiques des objets, des sources et de l'espace entre les objets.

L'objectif n'est pas de respecter scrupuleusement les lois de la physique et de la physiologie, mais plutôt d'obtenir un bon rendu (**rendering**).

### 8.6.1 Modèle de Phong

Dans le cadre du tracé de rayons, un modèle d'éclairage est une fonction qu'on met en oeuvre en un point où de la lumière est réfléchi, diffusée ou réfractée.

Le modèle de Phong est un des plus anciens (1975). Supposons, dans un premier temps, qu'on travaille en lumière monochromatique. Dans ce cas, le modèle s'écrit

$$I_{emis} = k_{ambient}I_{ambient} + k_{diffus}I_{diffus} + k_{speculaire}I_{speculaire} + k_{refracte}I_{refracte}.$$

et donne des formules pour les éclairages *diffus*  $I_{diffus}$ , *spéculaire*  $I_{speculaire}$  et *réfracté*  $I_{refracte}$  en fonction des positions relatives des rayons incident, réfléchi et réfracté, et de l'éclairage incident  $I_{incident}$ .

**Notations.** Pour alléger le vocabulaire, on suppose qu'on est en train de calculer l'éclairage à la racine de l'arbre des rayons. L'éclairage incident arrive donc le long des rayons secondaires ou des rayons d'ombre, de vecteur unitaire  $\vec{L}$  (orienté vers la source) et l'éclairage émis est mesuré dans la direction de la caméra  $\vec{V}$ . On note  $\vec{N}$  la normale à la surface de l'objet,  $\vec{R}_s$  la réflexion du rayon secondaire (le symétrique de  $\vec{L}$  par rapport à  $\vec{N}$ ), et  $\vec{R}_r$  la réfraction du rayon secondaire.

- L'éclairage *ambient* est une constante. Il modélise l'éclairage résultant des réflexions et diffusions multiples de la lumière émanant des sources sur les objets non représentés dans la scène.
- L'éclairage *diffus* modélise la fraction de la lumière incidente qui est diffusée uniformément dans toutes les directions, par un matériau non réfléchissant (tissu, béton) ou partiellement réfléchissant (métal brossé, plastique, cuir). La loi de Lambert s'énonce

$$I_{diffus} = I_{incident}(\vec{N} \cdot \vec{L})_+$$

où l'indice  $+$  signifie que  $I_{diffus} = 0$  si  $\vec{N} \cdot \vec{L} < 0$ . Autrement dit, la quantité de lumière émise est indépendante de la direction d'émission.

- L'éclairage *spéculaire*, si on applique rigoureusement la loi de Descartes, devrait être concentré exclusivement dans une direction. Cela conduirait, pour une source ponctuelle, à n'éclairer qu'un pixel. Or on voit plutôt une tache brillante (**highlight**). Phong a choisi de modéliser ce phénomène par la formule

$$I_{speculaire} = I_{incident}(\vec{V} \cdot \vec{R}_s)_{+}^{n_{Phong}},$$

où  $n_{Phong}$ , l'indice spéculaire de Phong, détermine la largeur de la tache. Typiquement,  $n_{Phong}$  varie entre 1 et 200, la tache est étroite et brillante si  $n_{Phong}$  est grand, elle est large et pâle si  $n_{Phong}$  est petit.

**Remarque.** Lorsque le rayon secondaire n'est pas un rayon d'ombre (i.e. aboutit sur un objet), la formule de Phong donne simplement  $I_{speculaire} = I_{incident}$ . Autrement dit, les reflets d'objets sont exacts, seuls les reflets des sources sont élargis en taches.

- L'éclairage *réfracté* obéit à la même règle

$$I_{refracte} = I_{incident}(\vec{V} \cdot \vec{R}_r)_{+}^{n_{Phong}},$$

- Les coefficients  $k_{ambient}$ ,  $k_{diffus}$ ,  $k_{speculaire}$  et  $k_{refracte}$  sont des caractéristiques de l'objet. Leurs valeurs relatives traduisent le caractère plus ou moins réfléchissant du matériau. Ils jouent aussi un rôle dans la couleur, voir plus loin.
- L'éclairage *incident* se calcule en fonction de l'éclairage  $I_{source}$  émis par la source ou par l'objet d'où provient la lumière. Logiquement

$$I_{incident} = \frac{I_{source}}{R^2}$$

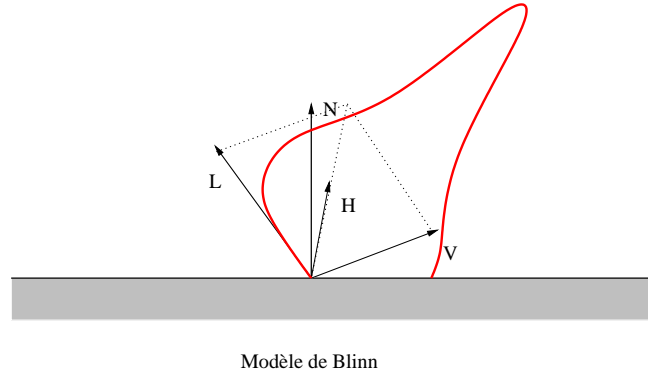
où  $R$  est la longueur parcourue par la lumière depuis la source ou l'objet précédent. Ce facteur produit l'effet de *profondeur*. L'expérience montre qu'un meilleur résultat visuel est obtenu en remplaçant  $R^2$  par  $R + k$ .

### 8.6.2 Une variante

Blinn (1977) suggère de remplacer  $\vec{V} \cdot \vec{R}$  par  $\vec{N} \cdot \vec{H}$  où  $H = \frac{\vec{V} + \vec{R}}{2}$ . On gagne en temps de calcul. En effet, si la caméra et la source sont à l'infini,  $\vec{N} \cdot \vec{H}$  est constant le long d'une face plane d'un objet.

### 8.6.3 Couleur

En théorie, il faudrait un modèle (en fait, un tracé de rayons complet s'il y a réfraction et donc dispersion) pour chaque longueur d'onde. En pratique, on ne travaille qu'avec les trois couleurs R, V et B. Il y a donc des coefficients  $k_{ambient,R}$ ,  $k_{diffus,R}$ ,  $k_{speculaire,R}$  et  $k_{refracte,R}$  etc... pour chaque



Modèle de Blinn

Figure 8.8:

couleur. Typiquement, on choisit des valeurs égales  $k_{\text{speculaire},R} = k_{\text{speculaire},V} = k_{\text{speculaire},B}$ , ce qui donne une tache spéculaire blanche en lumière incidente blanche. En revanche, le vecteur  $(k_{\text{diffus},R}, k_{\text{diffus},V}, k_{\text{diffus},B})$ , couleur diffusée par un objet lorsqu'il est éclairé en blanc, traduit la couleur (au sens du langage courant) de l'objet.

**Exercice 12** On modélise une boule de billard opaque par la sphère unité. Ses coefficients spéculaires sont  $k_{\text{speculaire},r} = k_{\text{speculaire},v} = k_{\text{speculaire},b} = 1$ . En revanche, la boule est rouge, donc ses coefficients diffus valent  $k_{\text{diffus},r} = 1$ ,  $k_{\text{diffus},v} = k_{\text{diffus},b} = 0$ . L'éclairage arrive par devant (dans la direction  $Ox$ ) à distance infinie, en lumière blanche. On prend une photo par en haut (dans la direction  $Oz$ ) de la sphère. Calculer l'image (i.e. les fonctions  $R(x, y)$ ,  $V(x, y)$ ,  $B(x, y)$ ) lorsque la caméra se trouve

- à distance infinie ;
- en  $(0, 0, 2)$  et l'écran à distance 1.

**Solution de l'exercice 12.** Boule de billard rouge éclairée par devant et vue de dessus.

Les rayons primaires sont verticaux. Leur vecteur directeur est  $\vec{V} = (0, 0, 1)$ .

Si  $x^2 + y^2 > 1$ , le rayon primaire ne coupe pas la boule, donc l'éclairage est nul,  $R(x, y) = V(x, y) = B(x, y) = 0$ .

Sinon, le rayon primaire coupe la sphère en premier en  $p = (x, y, \sqrt{1 - x^2 - y^2})$ . En ce point, la normale sortante est

$$\vec{N} = (x, y, \sqrt{1 - x^2 - y^2}).$$

On lance de  $p$  un rayon d'ombre vers la source (il n'y a pas d'autres objets qui pourraient se refléter dans la boule). Son vecteur directeur est  $\vec{L} = (1, 0, 0)$ .

Eclairage incident. Comme la source est à distance infinie, on ne peut pas tenir compte de son éloignement, donc on pose  $I_{\text{incident}} = I_{\text{source}}$ .

Eclairage diffus. On calcule

$$\vec{L} \cdot \vec{N} = x.$$

Par conséquent,  $I_{\text{diffus}} = 0$  si  $x < 0$ ,  $I_{\text{diffus}} = I_{\text{source}}x$  sinon.

Eclairage spéculaire. On calcule la réflexion du rayon d'ombre

$$\vec{R}_s = 2(\vec{L} \cdot \vec{N})\vec{N} - \vec{L} = (2x^2 - 1, 2xy, 2x\sqrt{1 - x^2 - y^2})$$

puis

$$(\vec{R}_s \cdot \vec{V})_+ = 2x\sqrt{1 - x^2 - y^2} \text{ si } x > 0, \quad = 0 \text{ sinon.}$$



Par conséquent  $I_{\text{speculaire}} = 0$  si  $x < 0$  et sinon

$$I_{\text{speculaire}} = I_{\text{source}}(2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}}.$$

L'éclairage est blanc. Cela signifie que  $I_{\text{source}}$  est le même pour les trois couleurs. Il vient

$$I_{\text{emis},R} = I_{\text{source}}(x + (2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}}), \quad I_{\text{emis},V} = I_{\text{source}}(2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}},$$

$$I_{\text{emis},B} = I_{\text{source}}(2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}}.$$

Enfin, on peut ajouter un effet de profondeur en multipliant les éclairagements émis par  $\frac{1}{k-\sqrt{1-x^2-y^2}}$  où  $k$  est une constante à ajuster (de même que  $n_{\text{Phong}}$ ) au vu du résultat visuel. Il vient

$$r(x, y) = I_{\text{source}} \frac{(x + (2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}})}{k - \sqrt{1-x^2-y^2}}, \quad v(x, y) = I_{\text{source}} \frac{(2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}}}{k - \sqrt{1-x^2-y^2}},$$

$$b(x, y) = I_{\text{source}} \frac{(2x\sqrt{1-x^2-y^2})^{n_{\text{Phong}}}}{k - \sqrt{1-x^2-y^2}}.$$

On règle la luminosité  $I_{\text{source}}$  ad libitum. ■

## 8.7 Aliassage

### 8.7.1 Nature

L'aliassage (*aliasing*), c'est l'apparition sur une image

- de marches d'escalier ;
- de moirés ;

et la disparition de petits objets.

L'oeil est particulièrement sensible aux transitions sombre/lumineux, donc il s'attache à décortiquer les marches d'escaliers et moirés (alignements de marches d'escalier dans les régions rayées) qui n'ont pourtant aucune signification. L'oeil voit des motifs qui n'existent pas, d'où la terminologie (*alias* = autre en latin).

### 8.7.2 Origine de l'aliassage

L'aliassage a pour origine les erreurs d'*échantillonnage*, i.e. celles qui se produisent quand on transforme une image continue en une image discrète, i.e. un tableau de pixels. C'est lorsque l'image présente des transitions que l'échantillonnage pose des problèmes.

Les transitions brutales, i.e. les discontinuités, sont bien détectées par la transformation de Fourier.

**Exercice 13** Soit  $k$  la fonction périodique de période  $2\pi$  qui vaut 1 sur  $[0, \pi[$  et  $-1$  sur  $[-\pi, 0[$ . Calculer ses coefficients de Fourier.

**Solution de l'exercice 8.9.** Série de Fourier d'un créneau.

On calcule

$$\hat{k}_n = \frac{1}{2\pi} \int_0^{2\pi} k(x)e^{-inx} dx = \frac{2}{\pi in}$$

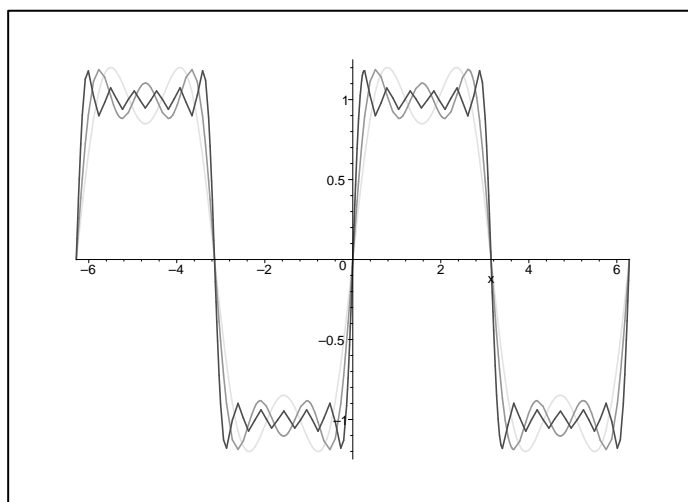


Figure 8.9: Premières harmoniques d'un créneau

si  $n$  est impair, et est nul si  $n$  est pair. Il vient

$$k(x) = \sum_{n \in \mathbf{Z}} \frac{2}{\pi i(2n+1)} e^{i(2n+1)x} = \sum_{n \in \mathbf{N}} \frac{4}{\pi(2n+1)} \sin((2n+1)x).$$

La figure 8.9 représente la somme des 2 (resp. 3, resp. 4) premiers termes de cette série. On constate que la convergence vers la fonction  $k$  est particulièrement lente au voisinage des multiples de  $\pi$ , i.e. des discontinuités de  $k$ . ■

Sur cet exemple, on voit que la discontinuité se traduit par une décroissance lente des coefficients de Fourier, i.e. les hautes fréquences sont très présentes.

### 8.7.3 Echantillonnage

Les hautes fréquences sont difficiles à *échantillonner*, i.e. lorsqu'on transforme un signal continu en un signal discret en prenant ses valeurs à intervalles réguliers, les hautes fréquences donnent lieu à des erreurs. On va expliquer ce principe pour des fonctions d'une variable, la généralisation à deux variables est immédiate.

#### Exemple.

Les fonctions  $x \mapsto e^{i5x}$  et  $x \mapsto e^{-i2x}$  prennent les mêmes valeurs aux points  $x = \frac{2k\pi}{7}$ ,  $k \in \mathbf{Z}$ .

**Théorème 15** *L'échantillonnage aux points  $x = \frac{2k\pi}{N}$ ,  $k \in \mathbf{Z}$ , est injectif sur les sommes d'exponentielles de fréquences  $< N/2$ .*

**Preuve.** Echantillonner  $f$  revient à remplacer  $f$  par la distribution

$$g_N = \sum_{k \in \mathbf{Z}} f\left(\frac{2k\pi}{N}\right) \delta_{\frac{2k\pi}{N}}$$

où  $\delta_x$  désigne la masse de Dirac en  $x$ . Les coefficients de Fourier de  $g_N$  valent

$$(\hat{g}_N)_n = \frac{N}{2\pi} \sum_{p \in \mathbf{Z}} (\hat{f})_{n+pN}$$

(voir exercice 14). On parle de *repliement du spectre*. Si le spectre de  $f$  (i.e. l'ensemble des indices  $n \in \mathbf{Z}$  tels que  $(\hat{f})_n$  est non nul) est contenu dans un intervalle de longueur  $< N$  (par exemple dans l'intervalle  $] -N/2, N/2[$ , l'application  $\hat{f} \mapsto (\hat{g}_N)$  est injective, donc  $f \mapsto g_N$  est injective. ■

**Exercice 14** Soit  $f$  une fonction périodique de période  $2\pi$ . Soit  $N \in \mathbf{N}$ . On note

$$g_N = \sum_{k \in \mathbf{Z}} f\left(\frac{2k\pi}{N}\right) \delta_{\frac{2k\pi}{N}},$$

où  $\delta_x$  désigne la masse de Dirac en  $x$ . Calculer les coefficients de Fourier de  $g_N$ . En utilisant la décomposition de  $f$  en série de Fourier

$$f(x) = \sum_{n \in \mathbf{Z}} \hat{f}_n e^{inx},$$

montrer que

$$(\hat{g}_N)_n = \frac{N}{2\pi} \sum_{p \in \mathbf{Z}} \hat{f}_{n+pN}.$$

**Solution de l'exercice 14.** *Repliement du spectre.*

Par définition, les coefficients de Fourier de la masse de Dirac en  $y$  valent

$$(\delta_y)_n = e^{-ni\frac{2k\pi}{N}}.$$

Il vient

$$(\hat{g}_N)_n = \sum_{k=1}^N \frac{1}{2\pi} f\left(\frac{2k\pi}{N}\right) e^{-ni\frac{2k\pi}{N}}.$$

Or

$$f\left(\frac{2k\pi}{N}\right) = \sum_{p \in \mathbf{Z}} \hat{f}_p e^{ip\frac{2k\pi}{N}}.$$

d'où

$$\begin{aligned} (\hat{g}_N)_n &= \sum_{k=1, \dots, N, p \in \mathbf{Z}} \frac{1}{2\pi} \hat{f}_p e^{i(p-n)\frac{2k\pi}{N}} \\ &= \sum_{p \in \mathbf{Z}} \frac{1}{2\pi} \hat{f}_p \sum_{k=1}^N e^{i(p-n)\frac{2k\pi}{N}} \\ &= \sum_{p \equiv n \pmod{N}} \frac{N}{2\pi} \hat{f}_p \end{aligned}$$

et enfin

$$(\hat{g}_N)_n = \frac{N}{2\pi} \sum_{p \in \mathbf{Z}} (\hat{f})_{n+pN}. \quad \blacksquare$$

Conclusion : avant d'échantillonner, il faut supprimer les hautes fréquences, i.e. appliquer l'opérateur

$$P_N : f \mapsto (x \mapsto \sum_{|n| < N/2} \hat{f}_n e^{inx}).$$

$P_N$  est le projecteur orthogonal sur le sous-espace engendré par les exponentielles de fréquences  $< N/2$ . On peut le voir comme un opérateur de convolution. En effet, annuler les hautes fréquences de  $f$  revient à multiplier  $\hat{f}$  terme à terme par la suite  $\chi_N$  telle que

$$(\hat{\chi}_N)_n = 1 \text{ si } |n| < N/2, \quad (\hat{\chi}_N)_n = 0 \text{ sinon,}$$

i.e. à convoluer  $f$  avec la fonction

$$\chi_N : x \mapsto \frac{\sin(Nx/2)}{\sin(x/2)},$$

(voir exercice 15). Autrement dit,

$$P_N(f)(x) = \int_{\mathbf{R}} f(y)\chi_N(x-y) dy.$$

On obtient une bonne approximation de  $P_N$  en effectuant une convolution avec une approximation de  $\chi_N$ , par exemple en tronquant  $\chi_N$  à son premier 0. En termes d'échantillonnage, cela revient à choisir, comme discrétisation de la fonction  $f$ , les valeurs

$$g\left(\frac{2k\pi}{N}\right) = \sum_{i=-\ell}^{\ell} a_i f\left(\frac{2(k+i)\pi}{N}\right)$$

où les  $a_i$  sont des valeurs de  $\chi_N$ . Autrement dit, on régularise l'échantillonnage en prenant des moyennes pondérées des valeurs aux points voisins.

**Exercice 15** Soit  $N \in \mathbf{N}$ . Quelle est la fonction  $\chi_N$  périodique de période  $2\pi$  dont les coefficients de Fourier valent

$$(\hat{\chi}_N)_n = 1 \text{ si } |n| < N/2, \quad (\hat{\chi}_N)_n = 0 \text{ sinon ?}$$

Tracer sommairement sa courbe représentative (on commencera par tracer la courbe représentative de  $x \mapsto \frac{1}{\sin(x/2)}$ ).

**Solution de l'exercice 15.** Noyau du filtre passe-bas.

Supposons  $N$  impair. Par définition

$$\begin{aligned} \chi_N(x) &= \sum_{n=(1-N)/2}^{(N-1)/2} e^{inx} \\ &= e^{(1-N)ix/2} \sum_{n=0}^{N-1} e^{inx} \\ &= e^{(1-N)ix/2} \frac{e^{iNx} - 1}{e^{ix} - 1} \\ &= \frac{e^{iNx/2} - e^{-iNx/2}}{e^{ix/2} - e^{-ix/2}} \\ &= \frac{\sin(Nx/2)}{\sin(x/2)}. \end{aligned}$$

Lorsque  $N$  est pair,

$$\chi_N(x) = \chi_{N-1}(x) = \frac{\sin((N-1)x/2)}{\sin(x/2)}.$$

Voici la courbe représentative de  $x \mapsto \frac{1}{\sin(x/2)}$ . Pour  $N$  grand,  $\chi_N$  est une sinusoïde de haute fréquence modulée par  $\frac{1}{\sin(x/2)}$ . Aux multiples de  $2\pi$ ,  $\chi_N$  tend vers  $N$  (resp.  $N-1$ ). On en déduit la courbe représentative de  $\chi_N$ .

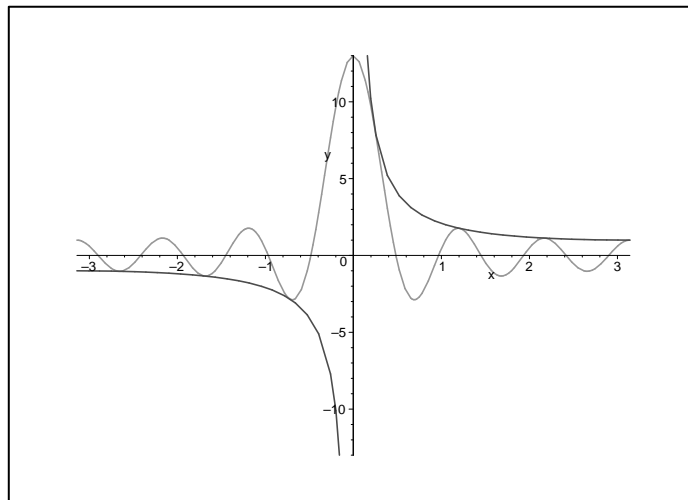


Figure 8.10: Courbe représentative de la fonction  $\chi_{13}$ . ■

#### 8.7.4 Comment réduire l'aliasage

- En suréchantillonnant, i.e. en augmentant  $N$ . Comme cela est coûteux, on peut détecter les régions où un suréchantillonnage est nécessaire (en comparant la couleur en un point avec la couleur aux points voisins) et subdiviser les pixels correspondant. C'est le principe du *suréchantillonnage adaptatif*.
- En supprimant les hautes fréquences dans l'image initiale, i.e. en appliquant l'opérateur  $P_N$  ci-dessus ou une approximation de  $P_N$  dans les deux directions  $x$  et  $y$ .

#### 8.7.5 Antialiasage et lancer de rayons

La disparition de petits objets lors du lancer de rayon peut avoir une autre origine : l'objet passe entre les différents rayons primaires. Pour l'éviter, on peut à nouveau suréchantillonner, ou lancer des cônes plutôt que des rayons, voir [PGMR], page 307.

## 8.8 Textures

### 8.8.1 Nature

Une texture est une modulation des propriétés d'un objet : couleur, réflectivité, transparence...

On distingue

- Les textures 2D : ce sont des étiquettes collées sur la surface de l'objet. La surface étant paramétrée par

$$(u, v) \mapsto s(u, v),$$

il suffit de se donner une texture dans le domaine plan des paramètres, i.e. une fonction  $t(u, v)$  pour l'appliquer sur la surface. Par exemple, modifier la clarté  $c$  de l'objet par une

étiquette revient à poser

$$c(x, y, z) = t(s^{-1}(x, y, z)).$$

- Les textures 3D : ce sont des caractéristiques du matériau volumique dans lequel est taillé l'objet. La texture est directement une fonction de  $(x, y, z)$ .

### 8.8.2 Génération de textures

Que ce soit en 2D ou en 3D, les méthodes de génération de texture se regroupent en trois grandes familles.

- Modélisation physique, chimique biologique. C'est ainsi que A. Turing [?] a proposé un modèle mathématique (équation de réaction-diffusion) qui explique la formation des taches et rayures sur le pelage des animaux.
- Analyse et reconstruction automatique d'image. Par exemple, on examine les coefficients de Fourier d'une image modèle, on sélectionne un petit nombre de fréquences pertinentes et on supprime les autres. Parfois, l'image simplifiée obtenue ressemble suffisamment à l'image initiale.
- Formule explicite. Des formules simples donnent de bons résultats pour le bois, le marbre, les feuillages, le brouillard, les nuages, les montagnes ou le feu.

**Exemple.** Texture bois. Les cernes d'un tronc d'arbre forment approximativement des cylindres concentriques équidistants. La fonction  $(x, y, z) \mapsto \sin(a\sqrt{x^2 + y^2})$  décrit une fonction périodique de la distance à l'axe des  $z$ . Ici  $a$  est un paramètre lié à l'épaisseur des cernes. Pour un bois plus réaliste, on ajoute une perturbation aléatoire

$$(x, y, z) \mapsto c(x, y, z) = \sin(a\sqrt{x^2 + y^2} + b(x, y, z)).$$

Par exemple,  $b$  est un polynôme trigonométrique à coefficients choisis au hasard.

Cela donne une texture en noir et blanc. Si  $C$  désigne le vecteur  $(R, V, B)$  correspondant aux régions les plus claires (jaune) et  $S$  la couleur de la région la plus sombre des cernes (marron), on peut choisir comme vecteur couleur

$$(1 - c(x, y, z))C + c(x, y, z)S.$$

Si la scène comporte plusieurs objets en bois, ne pas oublier de changer l'axe du bois, deux objets n'ayant aucune raison d'être taillés dans le même sens.

# Bibliography

- [Ci] P.G. CIARLET, *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson.
- [CO] J. CORNEY, *3D modeling with the ACIS kernel and toolkit*. Wiley.
- [FVD] J.D. FOLEY, A. VAN DAM, *Fundamentals of interactive computer graphics*. Addison Wesley (1984).
- [F] A. FORTIN, *Analyse numérique pour ingénieurs*. Editions de l'Ecole Polytechnique de Montréal.
- [H] Ch.M. HOFFMAN, *Geometric and solid modelling - an introduction*. Morgan Kauffmann Publishers Inc.
- [HL] J. HOSCHEK, D. LASSER, *Grundlagen der geometrischen Datenverarbeitung*. Teubner, Stuttgart (1989). English translation : *Fundamentals of computer aided geometric design*. A. K. Peters, Wellesley, Mass. (1993).
- [K] R. KRESS, *Numerical Analysis*. Graduate Texts in Math. **181**. Springer, Heidelberg (1998).
- [NS] W.M. NEWMAN, R. SPROUTT, *Principles of interactive computer graphics*. Computation science series, McGraw-Hill.
- [PGMR] B. PEROCHE, D. GHAZANFARPOUR, D. MICHELUCCI, M. ROELEN, *Informatique graphique, méthodes et modèles*. Hermès, Paris (1998).
- [R] D.F. ROGERS, *Algorithmes pour l'infographie*. McGraw-Hill.
- [W] A. WATT, *3D computer graphics*. Addison Wesley (1993).





# Contents

<b>1</b>	<b>Synopsis du cours</b>	<b>3</b>
<b>2</b>	<b>Fondements de la XAO</b>	<b>5</b>
2.1	Qu'est-ce que la XAO ?	5
2.1.1	Attentes de l'utilisateur	5
2.1.2	Un champ d'application élargi	5
2.1.3	A chacun sa XAO	6
2.1.4	Tendances	6
2.2	Un peu d'histoire	6
2.3	Les composantes d'un produit de CAO	7
2.4	Contraintes et difficultés	7
2.4.1	Contraintes	7
2.4.2	Difficultés	8
<b>3</b>	<b>Features</b>	<b>9</b>
3.1	Vue méthodologique de conception	9
3.2	Exemple : le congé d'arête	10
3.3	Features utilisés en mécanique	11
3.3.1	Motifs	11
3.3.2	Transformations	11
3.3.3	Features à contour comportant un ajout de matière	11
3.3.4	Features à contour comportant un enlèvement de matière	12
3.3.5	Features contextuels	12
3.4	Associativité	12
3.5	Problème de la reconnaissance de features	13
<b>4</b>	<b>Modélisation du solide</b>	<b>15</b>
4.1	Distinguer géométrie et topologie	15
4.2	Modèle BRep	16
4.2.1	Courbes	16
4.2.2	Surfaces	17
4.2.3	Faces planes	18
4.2.4	Boucles	20
4.2.5	Caractéristique d'Euler-Poincaré	20
4.2.6	Orientation	21
4.2.7	Faces gauches et BRep	24
4.2.8	Condition de validité d'une BRep plane	25
4.2.9	Lien	26
4.2.10	Condition de validité d'une BRep	28
4.2.11	Variétés	29
4.3	Les classes C++ d'ACIS	31
4.3.1	Classes mathématiques	31

4.3.2	Classes géométriques . . . . .	31
4.3.3	Classes topologiques . . . . .	31
4.3.4	Tolérances . . . . .	32
4.3.5	Bibliographie sur ACIS . . . . .	32
4.4	Modèle procédural (CSG) . . . . .	32
4.4.1	Primitives . . . . .	32
4.4.2	Opérations booléennes régularisées . . . . .	33
4.4.3	Arbre CSG . . . . .	33
4.4.4	Mise à jour . . . . .	34
<b>5</b>	<b>Opérations booléennes</b>	<b>37</b>
5.1	Description de l'algorithme : version 2D . . . . .	37
5.2	Description de l'algorithme en 3D . . . . .	40
5.3	Extension de l'algorithme . . . . .	41
5.3.1	Cas de deux solides . . . . .	41
5.3.2	Réunion, différence régularisée . . . . .	42
5.4	Problèmes de robustesse . . . . .	42
5.4.1	Incohérences . . . . .	42
5.4.2	Zones de confusion . . . . .	44
5.4.3	Autres difficultés . . . . .	44
5.5	Terminologie de ACIS . . . . .	44
<b>6</b>	<b>Cheminement</b>	<b>47</b>
6.1	Champ d'application . . . . .	47
6.2	Intersection de surfaces . . . . .	47
6.2.1	$X_1$ paramétrée, $X_2$ donnée implicitement . . . . .	47
6.2.2	$X_1$ et $X_2$ données par des équations . . . . .	49
6.2.3	$X_1$ et $X_2$ données par des paramétrisations . . . . .	49
6.3	Théorie générale . . . . .	50
6.3.1	Position du problème . . . . .	50
6.3.2	Tangente à la courbe des solutions . . . . .	50
6.3.3	Cas de l'intersection de deux surfaces . . . . .	51
6.3.4	Coeur de l'algorithme . . . . .	51
6.3.5	Rappel sur la méthode de Newton-Raphson en dimension 1 . . . . .	52
6.3.6	Méthode quasi-Newton . . . . .	53
6.3.7	Mise en oeuvre de la méthode quasi-Newton . . . . .	53
6.3.8	Généralisation aux dimensions supérieures . . . . .	54
6.3.9	Le Théorème des submersions . . . . .	54
6.3.10	Cheminement en dimension $n$ . . . . .	55
6.4	Description de l'algorithme complet . . . . .	55
6.4.1	Détermination des inits . . . . .	55
6.4.2	Choix du pas . . . . .	57
6.4.3	Choix du sens de progression et test d'arrêt . . . . .	58
6.4.4	Détection des singularités . . . . .	58
6.4.5	Du discret au continu . . . . .	59
6.5	Alternative au cheminement . . . . .	59
<b>7</b>	<b>Interpolation et approximation</b>	<b>61</b>
7.1	Interpolation . . . . .	61
7.1.1	Polynôme d'interpolation de Lagrange . . . . .	61
7.1.2	Erreur d'interpolation . . . . .	62
7.1.3	Des polynômes aux splines . . . . .	62
7.1.4	Interpolation par des courbes B-splines . . . . .	63
7.1.5	Le problème linéaire . . . . .	63

7.1.6	Estimation de l'erreur d'interpolation . . . . .	64
7.1.7	Choix du vecteur de noeuds . . . . .	65
7.2	Approximation . . . . .	66
7.2.1	Position du problème . . . . .	66
7.2.2	Approximation par une courbe B-spline . . . . .	66
7.3	Résolution numérique d'un problème d'interpolation . . . . .	67
7.3.1	Rappels sur la méthode du gradient conjugué . . . . .	68
7.3.2	Méthode de Jacobi . . . . .	69
<b>8</b>	<b>Synthèse d'images</b> . . . . .	<b>71</b>
8.1	Géométrie affine . . . . .	71
8.1.1	Points et vecteurs . . . . .	71
8.1.2	Notation matricielle des applications affines . . . . .	71
8.2	Perspective . . . . .	73
8.2.1	Vues . . . . .	73
8.2.2	Coordonnées homogènes . . . . .	76
8.2.3	Transformations projectives . . . . .	78
8.2.4	Vues en perspective d'objets . . . . .	78
8.3	Couleur . . . . .	80
8.3.1	Nature physique de la lumière . . . . .	80
8.3.2	Perception de la couleur . . . . .	80
8.4	Elimination des parties cachées . . . . .	81
8.4.1	Algorithme : tampon de profondeur . . . . .	82
8.4.2	Avantages et limitations . . . . .	82
8.5	Tracé de rayons . . . . .	82
8.5.1	Réflexion et réfraction . . . . .	83
8.5.2	Principe de la méthode . . . . .	83
8.5.3	Algorithme . . . . .	84
8.5.4	Avantages et limitations . . . . .	85
8.5.5	Lissage . . . . .	85
8.5.6	Au delà du tracé de rayon . . . . .	86
8.6	Modèles d'éclairément . . . . .	86
8.6.1	Modèle de Phong . . . . .	86
8.6.2	Une variante . . . . .	87
8.6.3	Couleur . . . . .	87
8.7	Aliassage . . . . .	89
8.7.1	Nature . . . . .	89
8.7.2	Origine de l'aliassage . . . . .	89
8.7.3	Echantillonnage . . . . .	90
8.7.4	Comment réduire l'aliassage . . . . .	93
8.7.5	Antialiassage et lancer de rayons . . . . .	93
8.8	Textures . . . . .	93
8.8.1	Nature . . . . .	93
8.8.2	Génération de textures . . . . .	94

<http://www.math.u-psud.fr/~pansu>