



HAL
open science

Contribution à une approche C.A.O. produit : proposition d'un modèleur géométrique

Georges Loriod

► To cite this version:

Georges Loriod. Contribution à une approche C.A.O. produit : proposition d'un modèleur géométrique. Automatique / Robotique. Université de Valenciennes et du Hainaut-Cambrésis, 1989. Français. NNT : 1989VALE0025 . tel-03201969

HAL Id: tel-03201969

<https://uphf.hal.science/tel-03201969v1>

Submitted on 19 Apr 2021

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

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

Numero d'ordre 117-20
Année 1989



THESE

présentée à
L'UNIVERSITE DE VALENCIENNES
ET DU HAINAUT CAMBRESIS

en vue de l'obtention du titre de
Docteur de l'université de Valenciennes
Spécialité: Automatique et Informatique Humaine

par
Georges LORIOD
Ingénieur ENIB

CONTRIBUTION A UNE APPROCHE C.A.O PRODUIT
PROPOSITION D'UN MODELEUR GEOMETRIQUE

Soutenue le 29 novembre 1989 devant
la commission d'examen

M. P. COIFFET
M. Y. GARDAN
M. A. GIBAUD
M. X. HARDY
M. M. LUCAS
M. R. SOENEN

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre 1

Modélisation du produit

1. Introduction.....	4
2. Systèmes intégrés de C.F.A.O.....	4
2.1 Définition.....	4
2.2 Architecture générale.....	5
2.3 Inadéquation des modèles solides.....	6
3. Approche "Features modelling".....	8
3.1 Principe.....	8
3.2 Informations nécessaires.....	9
3.3 Mise en oeuvre.....	11
4. Modèle de produit.....	11
4.1 Caractéristiques géométriques.....	11
4.1.1 Arbre de construction.....	12
4.1.2 Nécessité d'un autre modèle.....	15
4.1.3 Modèle frontières.....	17
4.1.4 Couplage des deux modèles.....	19
4.2 Caractéristiques de précisions.....	20
4.3 Caractéristiques technologiques.....	22
4.4 Caractéristiques matérielles.....	22
5. Conclusion.....	22

Chapitre 2

Opérateurs booléens

Définition

Algorithmes existants

1. Introduction.....	24
2. Opérateurs booléens.....	24
2.1 Définition.....	24
2.2 Importance.....	25
2.3 Méthodes existantes.....	27
3. Algorithme "ligne de section".....	27
3.1 Classe d'objets.....	27
3.2 Principe de l'algorithme.....	28
3.3 Calcul des points d'intersection.....	30
3.4 Jonction des sommets.....	32
3.5 Découpe des objets.....	36
3.5.1 Insertion ligne de section.....	36
3.5.2 Partition des objets.....	37
3.6 Construction du résultat.....	38
3.7 Insuffisances.....	38
4. Algorithme "subdivision de faces".....	41
4.1 Classe d'objet.....	41
4.2 Principe de l'algorithme.....	42
4.3 Division des objets.....	43
4.4 Construction du résultat.....	45
4.5 Evaluation.....	46
5. Conclusion.....	49

Chapitre 3

Opérateurs booléens

Algorithme proposé

1. Introduction.....	50
2. Principe de l'algorithme.....	50
3. Classe d'objets.....	52
4. Création des sommets.....	53
4.1 Calcul du point d'intersection.....	54
4.2 Situation du point.....	55
4.3 Insertion des points d'intersection.....	57
5. Construction des arêtes nouvelles.....	62
6. Construction de la topologie du résultat.....	68
6.1 Classification des arêtes initiales.....	69
6.2 Fermeture des contours.....	71
6.2.1 Faces coplanaires.....	74
6.2.2 Recherche des cycles.....	82
6.2.3 Faces simples.....	84
6.2.4 Type des cycles d'un graphe.....	85
7. Vérification du résultat.....	88
7.1 Vérification de la topologie.....	89
7.2 Vérification de la géométrie.....	90
8. Propagation des liens.....	91
9. Conclusion.....	91

Chapitre 4

Mise en oeuvre Evaluation et perspectives

1. Introduction.....	92
2. Mise en oeuvre.....	92
2.1 Volumes englobants.....	94
2.2 Applications.....	94
3. Evaluation.....	97
3.1 Contexte de l'évaluation.....	97
3.2 Complexité pratique de l'algorithme.....	97
3.3 Résultats.....	98
3.3.1 Critères et modèles choisis.....	98
3.3.2 Cas limites.....	99
3.3.3 Objets de tailles importantes.....	100
3.3.4 Pièces réelles.....	102
3.3.5 Aspect qualitatif.....	102
3.3.6 Aspect quantitatif.....	104
4. Perspectives.....	107
4.1 Arbre des Boîtes englobantes.....	108
4.2 Volumes englobants et englobés.....	109
5. Conclusion.....	112
 Conclusion générale.....	 113
 Bibliographie.....	 117

Introduction

Les vingt dernières années ont été marquées par une évolution profonde du processus de conception et de fabrication d'un produit manufacturé.

Les traditionnelles planches à dessins des bureaux d'études et des méthodes ont ainsi progressivement été supplantées par les écrans graphiques des logiciels de dessin assisté par ordinateur (D.A.O.), phénomène qui, les performances des micro_ordinateurs croissant sans cesse, s'est encore accentué avec l'apparition de ce type de logiciels sur cette gamme de matériel.

Dotés de capacités de modélisation géométrique tridimensionnelle, ils sont ensuite devenus logiciels de conception assistée par ordinateur (C.A.O.) et permettent de prendre en compte les aspects volumiques et massiques d'un objet sans attendre la réalisation, très onéreuse, de prototypes. Dans le même but, d'autres outils ont été développés pour simuler numériquement le comportement d'une pièce soumise à des contraintes mécaniques, électriques ou thermiques.

L'étape de fabrication du produit a également connu cette mutation qui, parallèlement au développement des machines outils à commande numériques et des robots, a donné naissance à trois familles d'assistances informatiques :

- les aides à l'élaboration des gammes de fabrication et de contrôle,

- les générateurs automatiques et simulateurs de trajectoires de bras de robots et d'outils de machines à commande numérique,
- les logiciels intervenant dans le suivi et la planification de la production.

La tendance actuelle est à l'intégration de tous ces outils informatiques au sein d'un système unique afin de permettre un échange d'informations entre ces différentes applications requérant le minimum d'interventions humaines et un suivi continu du produit depuis sa conception jusqu'à sa production.

Cependant, cette intégration, nécessaire pour augmenter la productivité de cette phase de la vie d'un produit, se heurte à un obstacle majeur : une modélisation trop restrictive où seules les informations de nature géométrique sont prises en compte globalement.

La mise en évidence des déficiences d'une modélisation purement géométrique sera le point de départ de notre étude et aboutira à la définition d'un modèle de produit possédant les informations et la structure informationnelle nécessaires au processus de conception et de fabrication par enlèvements de matière d'une pièce.

L'élaboration de ce modèle de produit montre également qu'une représentation par les frontières, même approchée, reste d'actualité dès qu'une connaissance explicite de la géométrie du produit est nécessaire. Toutefois, ce modèle ne peut seul répondre à tous les besoins d'un système intégré de C.F.A.O et nous l'avons donc associé à une représentation par arbre de construction qui possède des caractéristiques très complémentaires.

Afin de maintenir la cohérence de cette double modélisation géométrique, il est indispensable de disposer d'algorithmes permettant de combiner deux objets modélisés par leurs frontières. Nous consacrons donc le second chapitre à cet aspect de la modélisation volumique et l'étude de deux familles d'algorithmes existants montre qu'ils ne sont pas parfaitement

adaptés aux exigences particulières du modèle que nous avons défini.

Dans le troisième chapitre, nous proposons alors une approche différente à ce problème de combinaison de deux objets qui prend en compte les cas problématiques identifiés dans le précédent chapitre et la spécificité du modèle de produit retenu.

Dans le dernier chapitre, nous étudions d'abord les problèmes posés par la mise en oeuvre des méthodes proposées, puis nous évaluons quantitativement et qualitativement l'algorithme global réalisé. Enfin, l'analyse des résultats obtenus nous permettra de définir les directions des développements futurs afin de rendre ce système de modélisation réellement opérationnel.

Chapitre 1

Modélisation du produit

1. Introduction

Les modèles utilisés aujourd'hui dans les systèmes de C.F.A.O commerciaux permettent une bonne représentation de la géométrie d'une pièce mais constituent également un frein à l'intégration réelle des fonctions de fabrication car ils ne supportent pas toutes les informations nécessaires.

Nous mettrons ce fait en évidence dans la première partie de ce chapitre puis nous présenterons le concept de modélisation par caractéristiques (features modelling).

Enfin, nous proposerons un modèle de produit conforme à cette approche.

2. Systèmes intégrés de C.F.A.O

2.1 Définition

Le concept de système intégré de C.F.A.O est l'aboutissement de l'évolution des outils informatiques dédiés aux services techniques de l'entreprise. Initialement les logiciels utilisés étaient totalement indépendants et formaient des îlots d'automatisation et d'optimisation au sein du processus de conception et de fabrication d'un produit manufacturé.

Une approche plus globale, par interfaçage de ces assistances ponctuelles successives, peut être envisagée mais conduit à un système disparate et suppose ce processus purement séquentiel.

Or, l'étude d'un produit est une démarche itérative qui peut être décrite [DAV 81] comme une succession de transformations de l'objet à un certain niveau de définition vers un autre plus concret et plus précis.

L'intégration, dans le contexte de la C.F.A.O, est alors défini [MAR 88] comme un lien direct entre différentes fonctions dissociées, garantissant la transmission automatique des informations et autorisant un enchaînement aléatoire de ces fonctions.

2.2 Architecture générale

Outre le recours à un interface utilisateur commun à toutes les applications, l'effort d'intégration a essentiellement porté sur la définition d'une architecture générale des systèmes de C.F.A.O capable de satisfaire cette définition.

Une telle architecture, employée dans la majeure partie des systèmes actuels, possède deux caractéristiques principales :
(Fig 1.1)

- Une position centrale des informations permettant de réduire notablement le nombre des interfaces nécessaires entre les différentes applications, de supprimer les données redondantes et les incohérences qui en découlent et de garantir leur disponibilité vis a vis de tous les utilisateurs.
- Une structure modulaire, gage de maintenabilité et d'évolutivité du système.

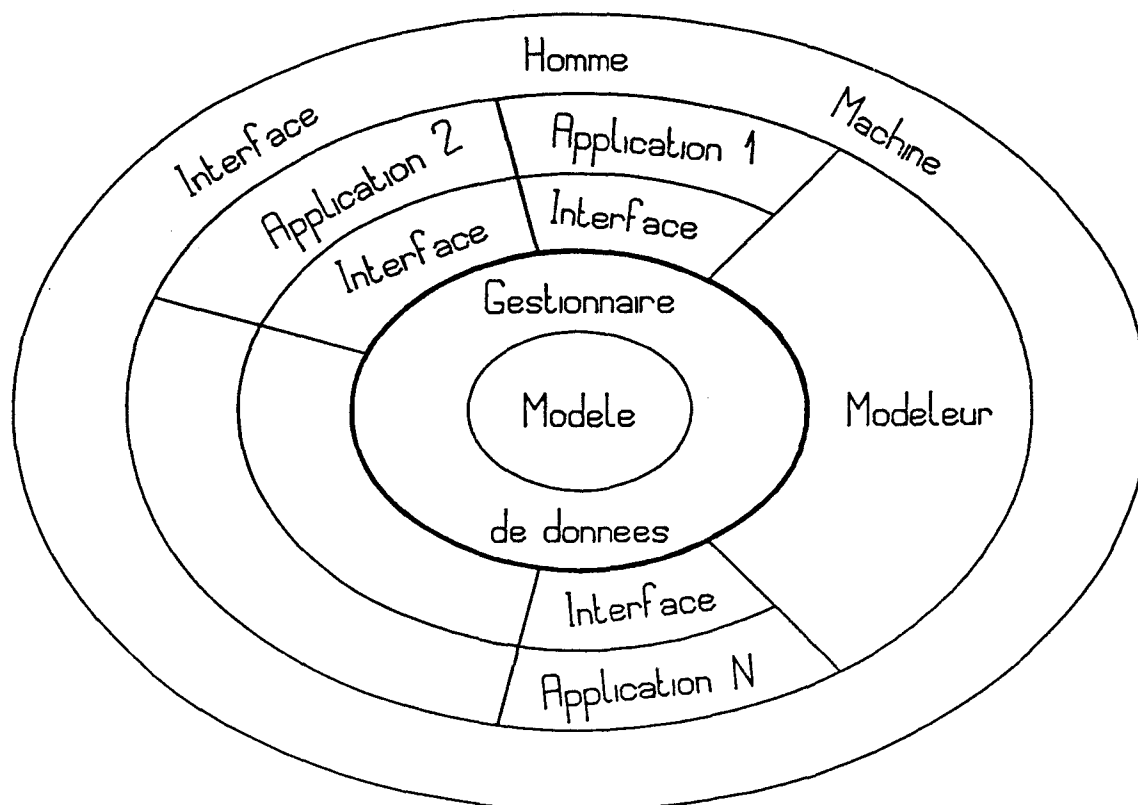


Figure 1.1 : Architecture d'un système Intégré de C.F.A.O.

Noyau de cette architecture, le modèle est la structure abstraite, appréhendable par les algorithmes, qui permet d'exprimer et de traiter les informations dont on dispose sur les objets du monde réel. C'est ce niveau, par l'emploi de modèles restreints aux seuls aspects géométriques, qui constitue aujourd'hui un goulot d'étranglement pour l'intégration des fonctions du processus d'élaboration du produit.

2.3 Inadéquation des modèles solides

Plus proche des besoins du concepteur que les modèles fil de fer et surfaciques apparus antérieurement car ils permettent la prise en compte des aspects volumiques et massiques des pièces, les modèles solides présentent cependant des lacunes importantes [SHA 88] :

- Définition incomplète des produits : En effet, les modèles solides permettent seulement la définition de la géométrie nominale d'une pièce. Or, l'imperfection des moyens de fabrication, les jeux de fonctionnement nécessaires à tout mécanisme et les fonctionnalités diverses assurées par les surfaces d'une pièce (étanchéité, glissement,..) imposent l'emploi de tolérances tant sur les dimensions que sur les formes ou l'état des surfaces des pièces. Ces informations jouent un rôle capital dans le processus de fabrication car elles conditionnent le choix des procédés d'usinages appliqués à la pièce et donc son coût. De plus, altérant les dimensions nominales, elles doivent impérativement être connues des applications de préparation à la fabrication telles que les générateurs automatiques de trajectoires d'outils des machines à commande numérique sous peine de produire des pièces hors tolérances et rebutées ou d'exiger la vérification et la correction manuelle systématique des programmes générés. Dans les systèmes de C.F.A.O contemporains, cette catégorie d'informations apparaît généralement au niveau des applications de cotation de plan, sous forme de symboles ou de textes, sans aucun lien avec le modèle solide central et donc inexploitable.

- Définition de bas niveau des produits : Les modèles solides les plus utilisés décrivent les formes des pièces en terme de sommets, d'arêtes, de faces ou encore par des arbres binaires combinant des primitives simples (cube, cylindre,...) par des opérations booléennes. Or, certains détails de la géométrie d'une pièce, même si leurs formes sont directement assimilables à un cylindre ou à un cube, contiennent plus d'informations car leurs fonctions ou leurs procédés de fabrication sont implicitement définis. Par exemple, une rainure ou un méplat réalisé dans une pièce cylindrique évoquent des méthodes d'usinages différentes mais seront modélisés dans les deux cas par la soustraction d'un parallélépipède rectangle à un cylindre. (Fig 1.2)

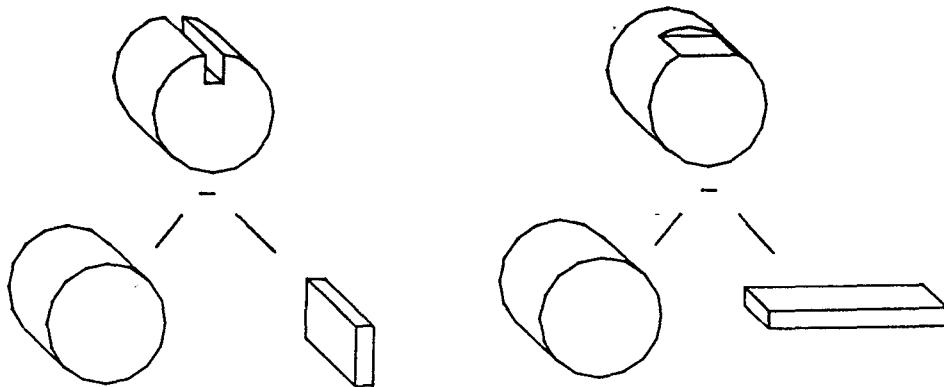


Figure 1.2 : Définition de bas niveau des pièces

En conséquence, les principales applications directes des modèles solides, bien que touchant des secteurs d'activités très divers comme le milieu artistique avec les images de synthèse et les films d'animation ou le domaine médical avec les images tomodensitométriques, sont essentiellement graphiques [WOO 88].

En outre, la majeure partie des avantages potentiels d'un système intégré, qui sont dus à la présence d'un modèle central unique et non ambigu, sont perdus car seule la géométrie du produit est modélisée et donc transmissible aux applications [MARK 87].

Les informations nécessaires à la phase d'industrialisation du produit doivent alors, soit être extraites interactivement par l'interprétation d'affichages graphiques, éventuellement très réalistes, du modèle solide, soit réintroduites manuellement, et en quelque sorte traditionnellement, par le biais des annotations et attributs portés sur le plan de définition de la pièce, plan créé grâce aux applications de cotations.

3. Approche "Features modelling"

3.1 Principe

De manière à incorporer à la maquette informatique du produit toutes les informations nécessaires à sa conception et à sa fabrication, l'approche "features modelling" consiste à

modéliser celui ci non plus en termes d'entités géométriques de bas niveau mais en termes de caractéristiques.

Dans ce contexte, une caractéristique est définie comme un ensemble d'informations relatifs à un aspect particulier d'une pièce, qu'il s'agisse de son esthétique, de son mode de fabrication ou de sa fonction [SHA 88]. Outre un support informationnel complet, cette démarche offre l'avantage d'établir une structuration de la masse d'informations de natures très diverses nécessaires au processus de conception et de fabrication car elles sont alors implicitement regroupées par famille.

3.2 Informations nécessaires

Applicable à de nombreux secteurs d'activité, cette approche doit cependant être particularisée car les caractéristiques d'une boîte de vitesse sont, par exemple, difficilement comparables à celles d'un vêtement et font d'ailleurs appel à des terminologies et des données spécifiques. Dans le cadre de la conception et de la fabrication mécanique, cinq catégories, rassemblant l'ensemble minimal des informations indispensables à l'étude d'un produit, ont été définies [MARK 87] :

- **Caractéristiques géométriques** : Ce sont les éléments géométriques de base, tels que points, segments, courbes et surfaces ainsi que les différentes relations existant entre ces éléments, qui permettent la définition de la géométrie nominale d'une pièce. Pratiquement, c'est le niveau d'informations manipulé par les divers modèles solides actuels.
- **Caractéristiques de précisions** : Elles quantifient les déviations maximales admissibles par rapport à la forme idéale de la pièce, tant au niveau macro_géométrique (tolérances dimensionnelles, de formes, de positions..) qu'au niveau micro_géométrique (états de surface).
- **Caractéristiques technologiques** : Il s'agit de sous ensembles de la géométrie d'une pièce qui assurent une fonction précise ou qui nécessitent un procédé de

fabrication particulier. Ces caractéristiques sont généralement définies par un vocabulaire spécifique (rainure, nervure, poche, taraudage..) plus riche que la seule description de la géométrie correspondante car ces termes évoquent la fonction remplie ou le mode de fabrication de la forme qu'ils décrivent. De plus, ils autorisent parfois le recours à une représentation simplifiée (filetage) ou symbolique (roue dentée) de cette forme.

- **Caractéristiques matérielles** : Elles décrivent les propriétés physiques, chimiques, électriques et mécaniques du matériau dont est constitué la pièce ainsi que les éventuels traitements subi par celle ci et qui ont pu altérer ces propriétés globalement (trempe à coeur, vieillissement) ou superficiellement (peinture, anodisation, cémentation..) et donc affectés à ses surfaces.
- **Caractéristiques administratives** : Les caractéristiques administratives regroupent des informations telles que référence interne ou externe, fournisseurs et délais d'approvisionnement éventuels, stocks, etc, sans influence sur les autres caractéristiques déjà définies mais nécessaires à la gestion de la vie de la pièce au sein de l'entreprise.

Un modèle intégrant toutes ces caractéristiques permet une automatisation beaucoup plus poussée des applications d'ingénierie déjà existantes dans les systèmes de C.F.A.O et, associé à la base de données générales de l'entreprise décrivant ses ressources matérielles et humaines, réalise alors le lien avec les applications de planification et de suivi de la production.

En outre, il offre des perspectives très vastes à une assistance accrue des systèmes intégrés car, enrichi d'une base de connaissance exprimant l'expérience et la connaissance des procédés d'usinages nécessaires à la préparation à la

fabrication, il permettrait la création d'un système expert dédié à cette étape capitale [MAR 88].

3.3 Mise en oeuvre

Un modèle de ce type peut être considéré comme une large extension des modèles solides actuels. A ce titre, deux méthodes sont envisageables pour intégrer à ceux ci les caractéristiques qui leur font défaut et plus particulièrement les caractéristiques technologiques :

- A posteriori : Dans cette approche, un programme de reconnaissance et d'extraction des caractéristiques technologiques examine la base de données produite par un modeleur solide et déduit le type des caractéristiques présentes. Appliquée à un modèle de représentation par les frontières [JOS 88] ou à un modèle par arbre de construction [WOO 88], cette méthode tente de rendre explicite ce qui est implicite dans ces modèles. Elle exige des programmes généralement complexes, spécifiques au modèle traité qui ne peuvent deviner les informations absentes (tolérances, états de surface) et qui ne sauraient prétendre à l'infailibilité. De plus, cette tâche devient inutile si l'on peut retenir ces informations lors de la phase de conception du produit.
- Au fur et à mesure : Cette approche doit offrir au concepteur le moyen de décrire le produit directement par ses caractéristiques et ce depuis le début de son étude jusqu'à son achèvement. Cela implique un modèle capable de manipuler des données à différents niveaux d'abstractions et c'est à l'élaboration d'un tel modèle que nous allons maintenant nous attacher.

4. Modèle de produit

4.1 Caractéristiques géométriques

Si ce type d'informations est celui qui est manipulé par les modèles solides actuels, il est aussi admis qu'un seul de ces

modèles ne suffit pas à faire face à toutes les fonctionnalités d'un système intégré de C.F.A.O [MAR 88].

Nous proposons donc un modèle géométrique hybride, basé sur une représentation par arbre de construction et une représentation par les frontières. Nous montrerons que ces deux modèles possèdent des caractéristiques générales complémentaires, bien adaptées au domaine de la conception et de la fabrication mécanique et demandent respectivement peu de modifications pour être associés.

4.1.1 Arbre de construction

Un objet est représenté par des entités volumiques simples auxquelles sont appliquées des transformations mathématiques et combinées par des opérations booléennes.

La structure sous_jacente est alors un arbre binaire dont les feuilles sont les primitives volumiques et dont les noeuds regroupent les opérateurs booléens et les transformations géométriques. (Fig 1.3)

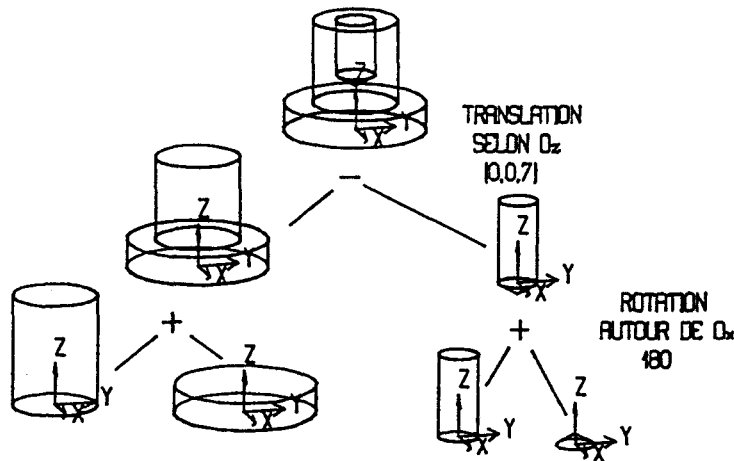


Figure 1.3 : Modélisation par arbre de construction

Ce schéma de représentation est exact, concis et sa validité est aisée à assurer [REQ 80]. De plus, sa structure d'arbre lui confère des avantages déterminants dans le cadre d'un système de C.F.A.O :

- L'utilisation des opérateurs booléens pour décrire la géométrie d'un objet semble être une technique qui conviendrait aux utilisateurs car elle s'apparente au mode d'obtention réel des pièces en fabrication mécanique par enlèvement de matière.
- Mémorisant l'historique de création de la maquette informatique d'un objet, il permet une modélisation simple des composants standards tels que vis, clavette, à la géométrie figée mais aux dimensions variables, par un arbre de construction dont les solides primitifs et les transformations géométriques ne sont pas évalués mais paramétrés (Fig 1.4).

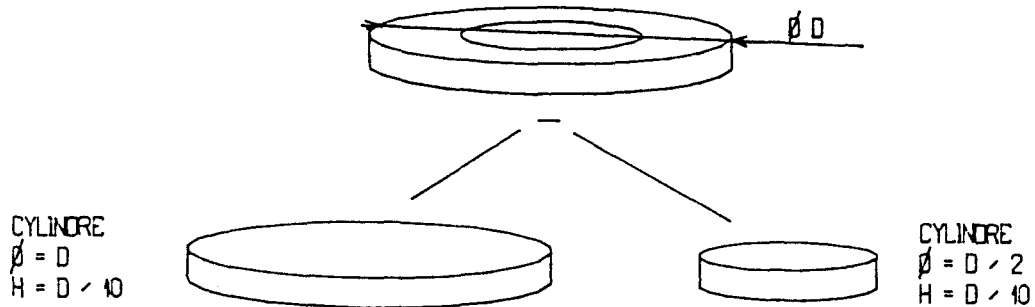


Figure 1.4 : Modélisation des pièces paramétrées

Cette propriété permet également des modifications aisées de la géométrie d'une pièce car il suffit d'agir sur la primitive ou le noeud concerné. Enfin, tous les états intermédiaires d'un objet étant mémorisés, favorisant ainsi la création rapide de variantes ou la reprise d'un projet à un stade antérieur de développement, il s'intègre donc bien au processus itératif essai_erreur de conception et garantit l'évolutivité indispensable des produits dont l'étude est terminée.

Le domaine de représentation de ce modèle étant limité par le type des surfaces des volumes élémentaires utilisés, les primitives suivantes :

- boîte
- pyramide
- coin (prisme à base triangulaire)
- cylindre
- cône
- tronc de cône
- sphère
- tore

permettent la modélisation géométrique de tout objet possédant des surfaces planes, cylindriques, coniques, sphériques ou toriques qui sont les surfaces les plus courantes des pièces obtenues par enlèvement de matière. Toutefois, un nombre aussi restreint de primitives impose une décomposition excessive des pièces à représenter.

Afin de rendre la conception moins fastidieuse et d'exploiter les capacités de paramétrisation inhérentes à ce modèle, il est intéressant d'introduire trois niveaux d'entités [MAR 88] :

- Les solides primitifs qui sont en fait les volumes élémentaires précédemment cités. Définis par leur type et un ensemble d'équations, de contraintes et de paramètres fixés par l'utilisateur, il leur est associé l'ensemble des références des surfaces les délimitant car celles ci seront le support d'autres caractéristiques et sont indispensables à de nombreuses applications.
- Les solides composés sont essentiels au modèle, qu'ils enrichissent sans cesse, car ils permettent d'ajouter de nouveaux solides paramétrés qui sont ensuite manipulés comme des solides primitifs. Construits à partir des solides primitifs, leur définition est semblable, seul le type étant remplacé par une référence d'entrée dans un arbre de construction.
- Les solides générés qui sont une valuation particulière d'un solide composé. Ils sont définis par une référence d'entrée dans un arbre de construction, un ensemble de valeurs à affecter aux paramètres définissant le solide et

l'ensemble des références des surfaces générés le délimitant.

4.1.2 Nécessité d'un autre modèle

L'énumération des nombreuses qualités de l'arbre de construction peut laisser croire qu'il s'agit du modèle géométrique idéal. La plupart de ces propriétés, qui font sa puissance et son intérêt dans un système de C.F.A.O, proviennent du fait qu'il ne décrit pas explicitement la géométrie de l'objet modélisé mais plutôt la façon de la construire [GHA 88].

Cette particularité ne présente pas que des avantages et permet l'élaboration d'arbres très complexes et rigoureusement valides qui ne modélisent rien ou dont une partie de l'arborescence n'a aucune influence sur la géométrie de la racine de l'arbre. (Fig 1.5)

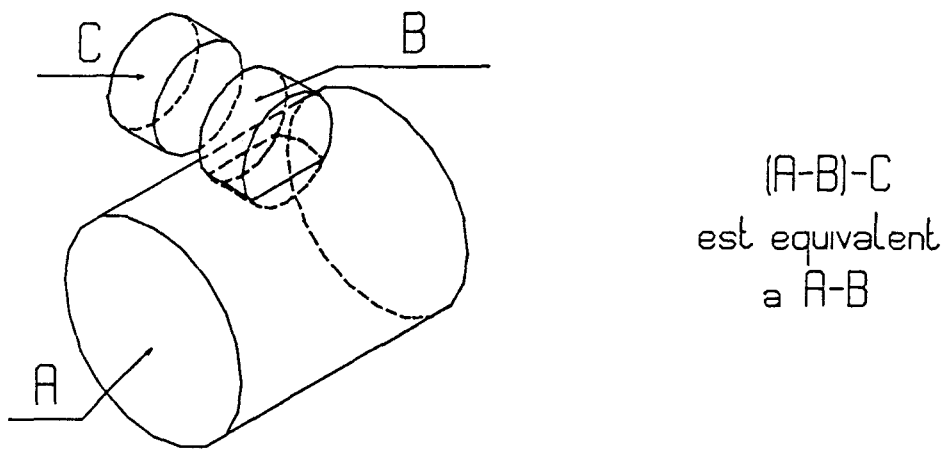


Figure 1.5 : Arbre de construction partiellement inutile

De plus, de nombreuses utilisations de la modélisation d'une pièce requièrent une connaissance plus formelle de sa géométrie. Ainsi, le calcul de caractéristiques physiques telles que le volume, le centre de gravité ou la surface d'un objet n'est pas réalisable directement.

De même, les applications de préparation à la fabrication, générateurs de trajectoires d'outils ou assistances à

l'élaboration des gammes de fabrication, s'appuient sur les surfaces vraies de la pièce. Or, si toutes les surfaces potentielles d'un objet sont connues au niveau des feuilles de l'arbre, certaines sont susceptibles de disparaître lors des opérations booléennes (Fig 1.6) information qui est totalement absente.

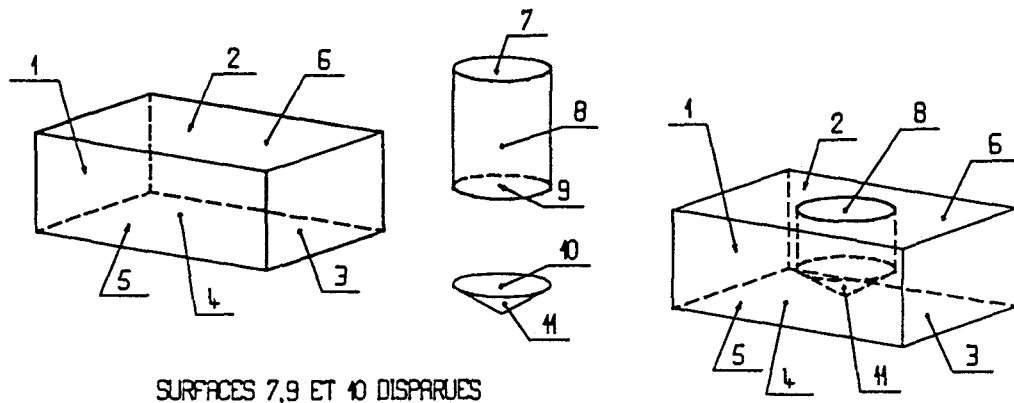


Figure 1.6 : Disparition de surfaces des primitives

Enfin, et c'est probablement l'inconvénient majeur de ce modèle, la visualisation d'une pièce, fonctionnalité essentielle car elle participe largement à l'interactivité et à l'ergonomie du système en permettant une validation immédiate de la géométrie en cours de conception, est possible seulement en appliquant un algorithme du type lancer de rayon (Ray_tracing) [SCH 87] [GHA 88].

Cette famille d'algorithmes fournit des images d'excellente qualité, ils sont d'ailleurs principalement utilisés en synthèse d'image, mais exigent des temps de calculs [FUJ 86] incompatibles avec la notion d'interactivité.

De plus, ce sont des algorithmes d'élimination des surfaces cachées qui produisent des images constituées de taches colorées, certes très réalistes et esthétiques, mais difficilement reproductibles sur un support papier et peu conformes à la normalisation du dessin technique.

En effet, ces dessins, se contentant de représenter les contours apparents des surfaces des pièces, sont formés exclusivement de lignes et sont donc beaucoup plus proches des images générées par les algorithmes d'élimination de lignes cachées.

Des solutions algorithmiques ont été proposées pour compenser certains de ces inconvénients, notamment le calcul des propriétés physiques [TIM 80] ou la détection des surfaces virtuelles [MAR 88]. Cependant, dus au principe même de description des objets par arbre de construction, une solution globale peut être apportée en procédant à l'évaluation de l'arbre par une conversion en un autre modèle solide.

4.1.3 Modèle frontières

Les limitations d'une modélisation par arbre de construction, évoquées au paragraphe précédent, permettent de définir les fonctionnalités attendues du modèle d'évaluation :

- autoriser une visualisation rapide du modèle sous une forme proche du dessin technique et éventuellement sous d'autres formes plus expressives.
- permettre le calcul des propriétés physiques de l'objet modélisé.
- offrir une modélisation explicite de la géométrie de la pièce et plus particulièrement de ses surfaces.
- supporter l'application de transformations géométriques et la combinaison de deux objets.

Ces caractéristiques imposent pratiquement la représentation par les frontières comme modèle d'évaluation. En effet, une représentation par arbre octal (octree) ne saurait convenir car, si la combinaison de deux objets ou le calcul des propriétés physiques sont très simples, l'application de transformations géométriques est beaucoup plus complexe [JAC 80] et la notion de surfaces est totalement absente. De même, un modèle à génération par balayage (sweeping) possède,

dans sa forme la plus élaborée, un très grand pouvoir de description mais il n'existe pas aujourd'hui d'algorithmes permettant la combinaison de deux objets [PLE 88].

Le modèle frontière (boundary representation ou B_REP), à l'opposé de l'arbre de construction, mémorise la géométrie de la frontière d'un objet avec définition explicite des sommets, des arêtes, des faces et de leurs connectivités (Fig 1.7).

Son implémentation la plus répandue n'autorise que des faces planes d'ou une approximation par une surface polyédrique des frontières des solides comportant d'autres types de surfaces. (fig 1.7)

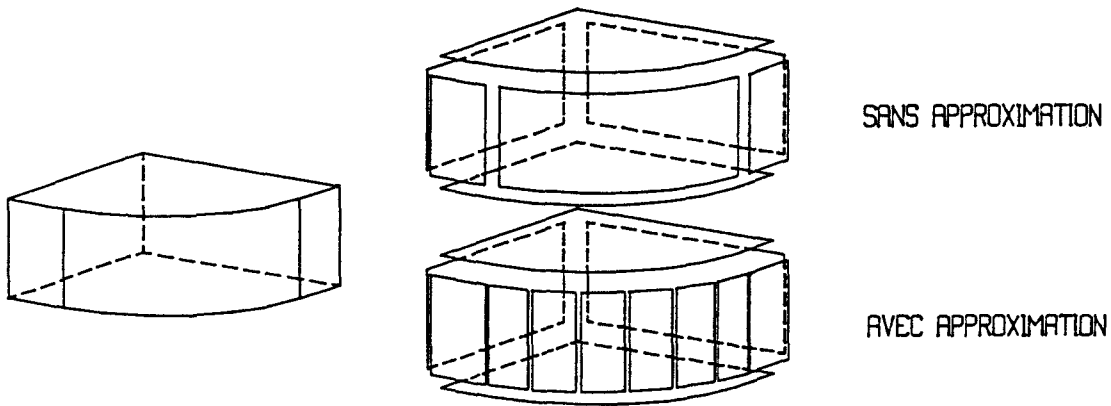


Figure 1.7 : Modélisations par les frontières exacte et approchée

Cette restriction est due au fait qu'il est parfois difficile de déterminer les faces de certains types d'objets (Fig 1.8) et permet de limiter la complexité, déjà élevée, des algorithmes effectuant les opérations booléennes. [MART88]

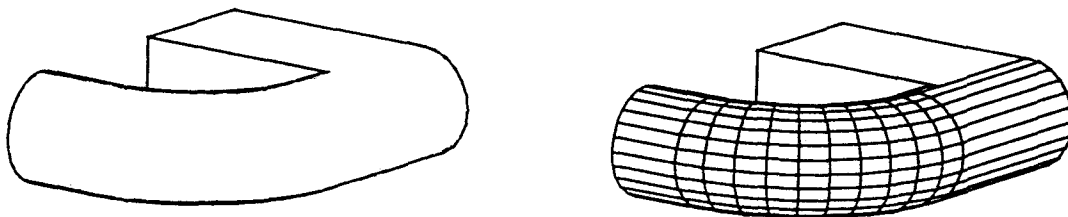


Figure 1.8 : Type des surfaces ?

Ce modèle, inexact sous cette forme, peu concis et dont la validité est délicate à assurer, est par contre directement exploitable pour la visualisation ou le calcul des propriétés physiques. La majorité des algorithmes d'élimination des parties cachées lui sont d'ailleurs dédiés et certaines stations de travail sont capables d'exploiter des scènes décrites selon ce type de représentation.

4.1.4 Couplage des deux modèles

De manière à permettre le filtrage des surfaces virtuelles, on ajoute à chaque face du modèle frontière d'un solide primitif un lien avec la surface originelle approchée (Fig 1.9), liens qui subsisteront ou qui disparaîtront en fonction de l'appartenance des surfaces aux résultats des combinaisons lorsque l'on procédera à l'évaluation des noeuds.

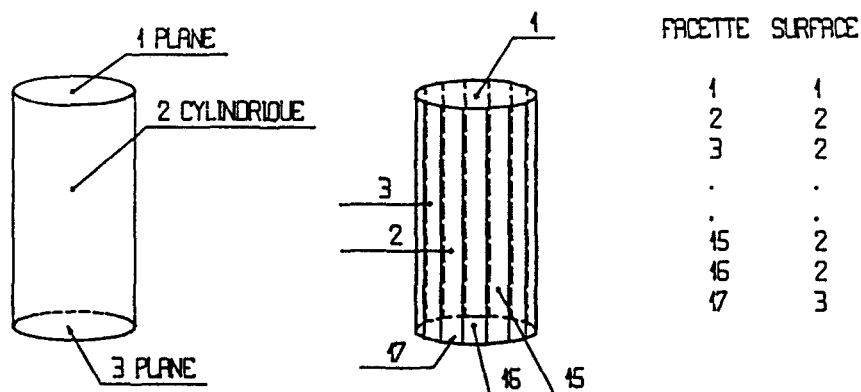


Figure 1.9 : Couplage des modèles frontières et arbre de construction

Les modifications à apporter aux deux modèles sont donc mineures et les liens introduits peuvent avoir en outre d'autres applications que le filtrage des surfaces virtuelles.

Ils offriront, par exemple, la possibilité d'améliorer la qualité de la visualisation du modèle frontière. En effet, ce type de représentation a pour conséquence de faire apparaître un grand nombre d'arêtes indésirables sur les surfaces non planes. La solution généralement proposée à ce problème consiste à ne pas dessiner une arête, théoriquement visible

après le calcul des parties cachées de l'objet, si l'angle formé par les deux plans contenant les faces auxquelles elle appartient est inférieur à une certaine valeur fixée. (Fig 1.10)

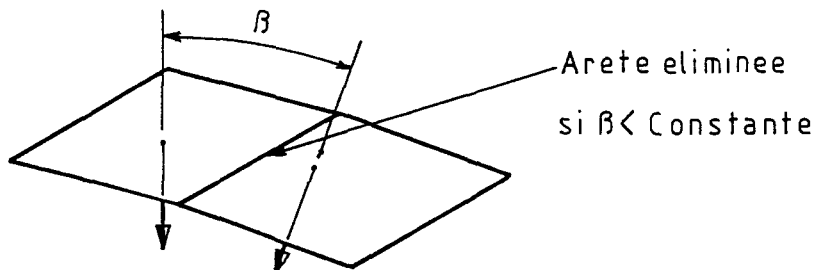


Figure 1.10 : Elimination des arêtes dues à l'approximation polyédrique lors du dessin

Ce procédé, également utilisé pour réaliser une variation continue de l'illumination des faces, malgré la facettisation, lors de la génération d'images ombrées d'un objet [NEW 79], fait aussi disparaître des arêtes réellement présentes. (Fig 1.11)

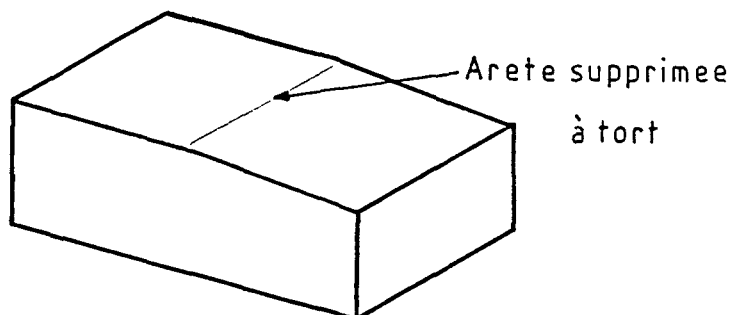


Figure 1.11 : Elimination Intempestive d'arêtes

Le lien de chaque face du modèle frontière avec la surface théorique approchée permet alors d'ajouter une condition supplémentaire, qui interdit toute élimination intempestive, imposant à une arête susceptible d'être supprimée d'appartenir à deux faces liées à la même surface réelle.

4.2 Caractéristiques de précisions

Destinées à fixer les limites admissibles des variations de la géométrie d'une pièce, ces caractéristiques peuvent être

affectées à une surface unique ou établir une relation entre plusieurs d'entre elles.

Bien qu'elles concernent des aspects géométriques très différents, la nature des informations que ces caractéristiques manipulent sont suffisamment proche pour permettre leur représentation par un modèle unique constitué :

- d'un type
- d'une ou plusieurs références à des surfaces du modèle géométrique de l'objet
- d'un ou plusieurs attributs quantifiant les variations tolérées vis à vis de la forme ou de la dimension théorique affectée ou précisant éventuellement la fonction ou le procédé de fabrication de la surface référencée.

Les attributs de ces caractéristiques sont des valeurs numériques ou littérales, directement appréhendables par les calculateurs, il n'est donc pas nécessaire de prévoir une représentation informatique particulière.

Ce modèle permet la modélisation des spécifications géométriques et dimensionnelles normalisées actuellement en vigueur soit, classées selon le nombre de surfaces référencées et d'attributs nécessaires :

- les spécifications d'états de surfaces
- les tolérances de formes (planéité, cylindricité..)
- les tolérances d'orientation (inclinaison, parallélisme..)
- les tolérances de position (coaxialité, symétrie..)
- les tolérances dimensionnelles

L'intégration totale de ces caractéristiques au modèle de produit alliée aux capacités de paramétrisation de l'arbre de construction permettent de déduire un nouveau modèle géométrique de la pièce dont les dimensions seraient centrées dans les intervalles de tolérance ou dont le volume serait minimal ou maximal. Ce modèle peut alors, par exemple, être

utilisé pour vérifier la cohérence des tolérances portées sur les différentes pièces d'un assemblage.

4.3 Caractéristiques technologiques

Leur modélisation est en fait déjà assurée par les différents niveaux d'entités introduits dans l'arbre de construction. En effet, les solides composés, qui peuvent être dotés de caractéristiques de précision génériques, permettent de constituer facilement une bibliothèque de formes technologiques standards paramétrées.

Manipulés ensuite par l'utilisateur comme des super_primitives, ils autorisent bien la description d'une pièce en termes de caractéristiques technologiques et, devenues des composantes à part entière du modèle, sont immédiatement identifiables dans une arborescence.

4.4 Caractéristiques matérielles

Sans influence sur la géométrie de la pièce modélisée, elles ne doivent cependant pas être omises car elles déterminent les paramètres d'usinages. Données numériques ou littérales, leur modélisation est directe, seuls les traitements et revêtements de surface exigeant en plus une référence à la surface affectée.

5. Conclusion

Nous avons proposé un modèle de produit conforme à l'approche "features modelling", approche dont l'objectif est de définir à la fois un support informationnel et une méthodologie de conception permettant une intégration réelle des fonctions de conception et de fabrication et respectant les besoins des utilisateurs.

Ce modèle de produit est basé sur une représentation géométrique hybride, associant un arbre de construction, pour sa puissance de description et de manipulation des objets, à un modèle frontière, pour sa définition explicite de la géométrie.

Si cette association demande peu d'aménagements des deux modèles, elle exige un ou des algorithmes permettant la combinaison de deux objets modélisés par leurs frontières.

Nous consacrerons la suite de cette étude à ces opérateurs booléens car c'est en fait le seul problème posé par l'évaluation d'un arbre de construction.

Le chapitre suivant définira précisément ces opérateurs puis nous analyserons le principe et montrerons les limites de deux algorithmes existants.

Dans le troisième chapitre, nous proposerons et détaillerons un algorithme utilisant une approche différente et nous procéderons à son évaluation qualitative et quantitative dans le dernier chapitre.

Chapitre 2

Opérateurs booléens

Définition

Algorithmes existants

1. Introduction

Pour faire face à toutes les fonctionnalités attendues d'un système intégré de C.F.A.O, nous avons défini un modèle de produit, respectant l'approche "features modelling", qui met conjointement en oeuvre une représentation par arbre de construction et une modélisation par les frontières.

La grande complémentarité de ces deux modèles géométriques rend cette association extrêmement séduisante, mais elle ne peut être viable que si l'on dispose d'opérateurs booléens permettant l'évaluation des noeuds de l'arbre de construction.

Nous consacrerons donc ce chapitre à l'étude de deux familles d'algorithmes permettant de combiner deux objets modélisés par leurs frontières et nous montrerons leurs limites dans ce contexte particulier d'utilisation.

2. Opérateurs booléens

2.1 Définition

Les opérateurs booléens sur les solides sont apparentés aux opérateurs ensemblistes et peuvent être définis comme suit :

- $A \cup B$ (Fusion) Ensemble des points appartenant à l'un ou l'autre des deux solides.

- $A - B$ (Différence) Ensemble des points du solide A à l'extérieur du solide B.
- $A \cap B$ (Commun) Ensemble des points appartenant aux deux solides.

Ces définitions sont malheureusement inexactes car ces opérateurs ne sont pas des lois de composition interne de l'ensemble des solides (Fig 2.1).

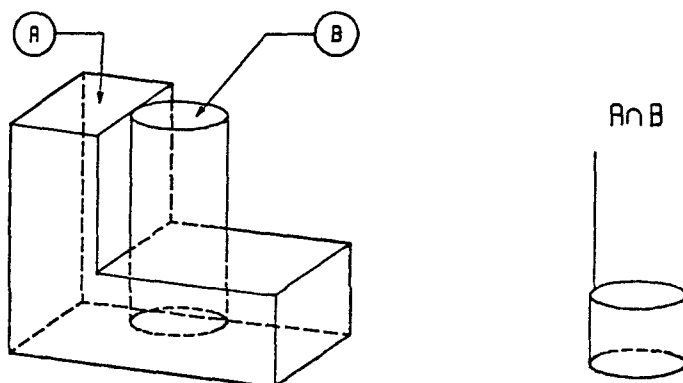


Figure 2.1

De nouveaux opérateurs ont alors été introduits, les opérateurs booléens régularisés ($U^*, -^*, \cap^*$), définis de la façon suivante [TIL 80] :

$$\begin{aligned}
 A \cup^* B &= r(A \cup B) \\
 A -^* B &= r(A - B) \\
 A \cap^* B &= r(A \cap B)
 \end{aligned}$$

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

2.2 Importance

L'arbre de construction ne fait que décrire les opérations permettant de construire l'objet, la conversion de cet arbre en une représentation par les frontières implique d'une part l'évaluation des solides primitifs sous forme B_REP et d'autre part la composition de ces objets selon les opérations définies

aux noeuds de l'arbre. Ces noeuds associent deux types d'opérations sur les objets:

- les transformations géométriques : Dans le cas d'une modélisation B_REP, elles sont particulièrement aisées à mettre en oeuvre, il suffit en effet d'appliquer la transformation à tous les sommets du solide.
- les opérations booléennes : Simples dans le principe, elles posent en fait de nombreux problèmes liés aux imprécisions numériques, à la singularité de certains cas et à la difficulté d'établir la validité d'une telle représentation. De plus les objets résultants sont généralement les arguments d'autres compositions, la stabilité de celles-ci vis à vis de la classe d'objets est donc indispensable. Enfin, le jeu limité de primitives retenu pour notre arbre de construction implique un large recours à ces opérations. Ainsi, il ne faut pas moins de six primitives et cinq opérations booléennes pour modéliser un prisme doté de quatre congés (Fig 2.2).

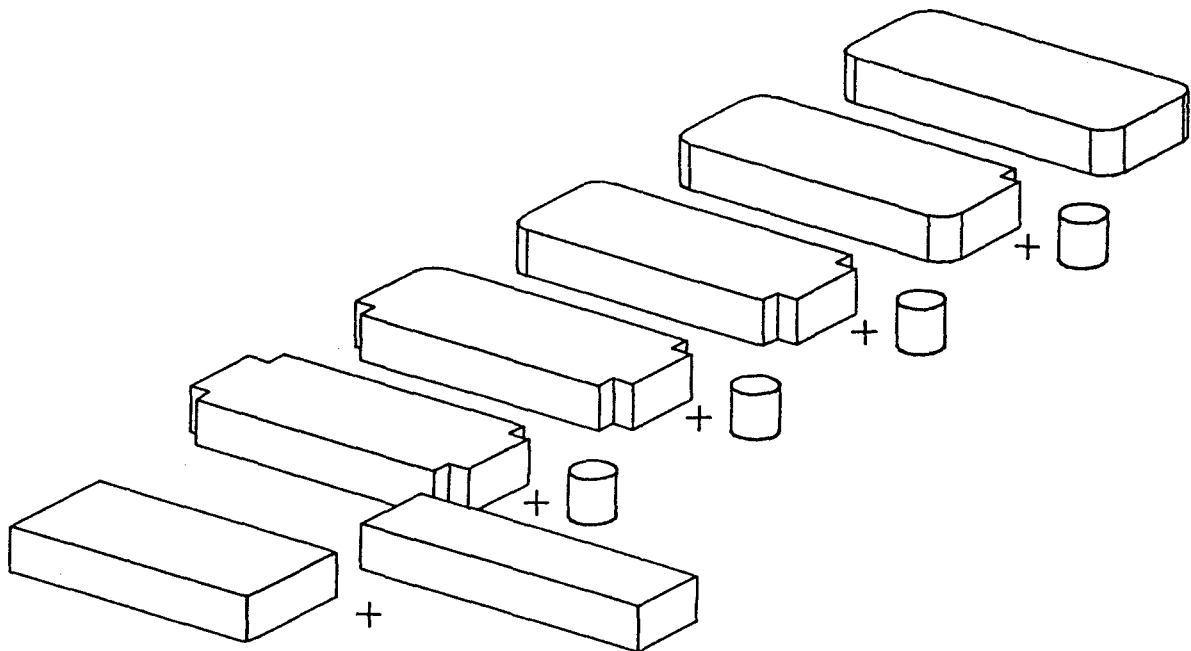


Figure 2.2 : Décomposition d'un objet

2.3 Méthodes existantes

De nombreux systèmes de modélisation mettent en oeuvre une représentation par les frontières de objets du monde réels et possèdent des algorithmes permettant de combiner deux solides virtuels.

Ces algorithmes, dont le but commun est un éclatement des faces des solides opérandes pour élaborer celles de l'objet résultat, peuvent être classés en deux grandes familles selon la manière dont est réalisé cet éclatement :

- Les algorithmes de la première famille procèdent par étapes successives qui respectent les différents niveaux, sommets, arêtes et faces, présents dans ce type de représentation. Ainsi, l'éclatement des faces n'est fait qu'après avoir calculé tous les sommets d'intersections et reliés ensuite ceux ci deux à deux pour former des arêtes. L'algorithme que nous appellerons "Ligne de section" appartient à cette catégorie et est issu de [BRU 80], [MAN 83a], [MAN 83b], [MART87] et [MART88].
- La seconde famille modifie la topologie de chacun des deux arguments en incorporant aux faces des objets initiaux les sommets et les arêtes résultant de l'intersection des deux frontières au fur et à mesure de leur apparition. Le résultat de la combinaison est alors implicitement contenu dans les nouvelles topologies ainsi créées dont il suffit d'unir deux sous-ensembles. L'algorithme que nous avons baptisé "Subdivision de faces" est un représentant de cette famille. C'est une variante de l'algorithme décrit dans [LAI 86] lui même inspiré de [TUR 84].

3. Algorithme "ligne de section"

3.1 Classe d'objets

Les solides admis par ce type d'algorithme appartiennent naturellement à la famille des représentations solides par les frontières, mais il n'y a pas de limitation à la complexité des objets qui peuvent être concaves, ou à celles des faces qui

peuvent également être concaves et posséder des contours intérieurs (Fig 2.3).

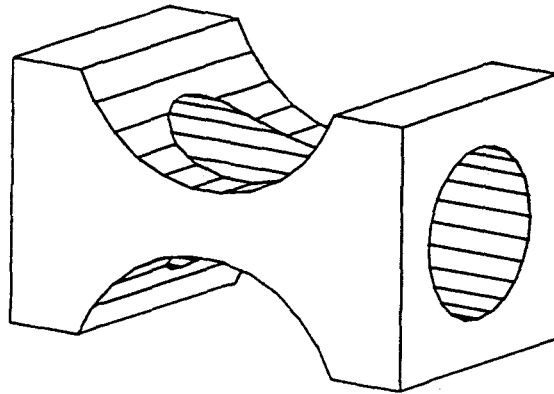


Figure 2.3 : Objets admis

3.2 Principe de l'algorithme

Il est basé sur le fait que les surfaces polyédriques, approchant les frontières des solides manipulés, donnent en se coupant une ou plusieurs lignes polygonales (Fig 2.4a) qui permettent ensuite de scinder chacun des objets en deux sous ensembles: le sous ensemble de l'objet A qui est situé à l'extérieur de l'objet B, celui qui est situé à l'intérieur, et inversement pour le deuxième objet (Fig 2.4b).

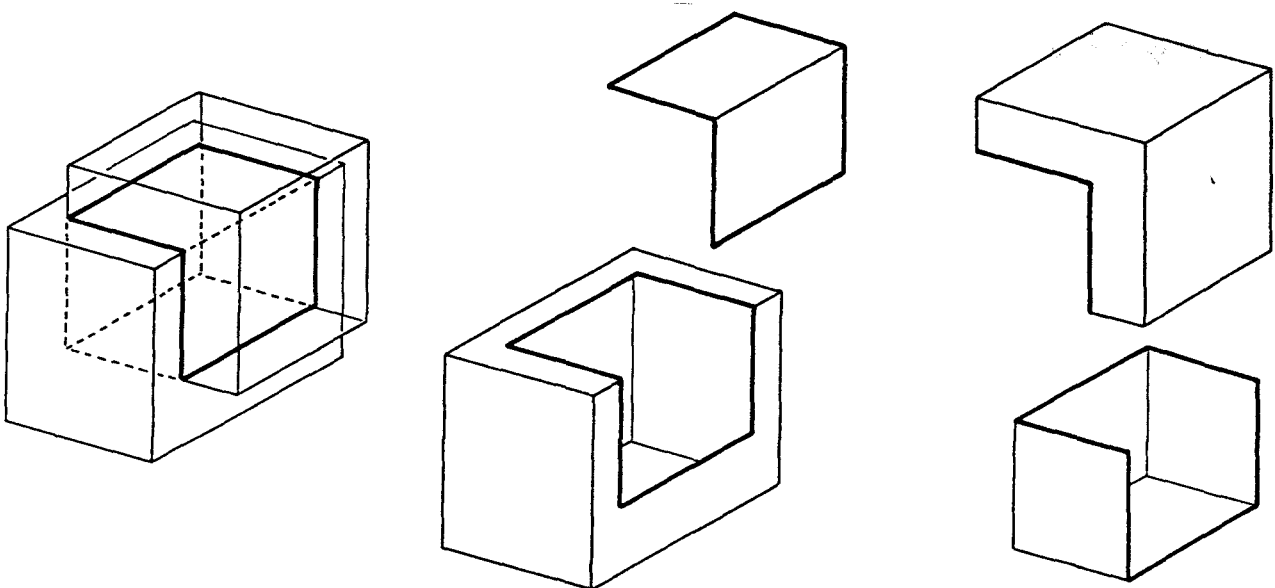


Figure 2.4 : Ligne d'Intersection et partition des objets

On obtient alors quatre sous ensembles d'objets, non fermés, qu'il faut assembler deux à deux le long de la ligne de section pour obtenir l'union, l'intersection ou la différence des deux solides (Fig 2.5).

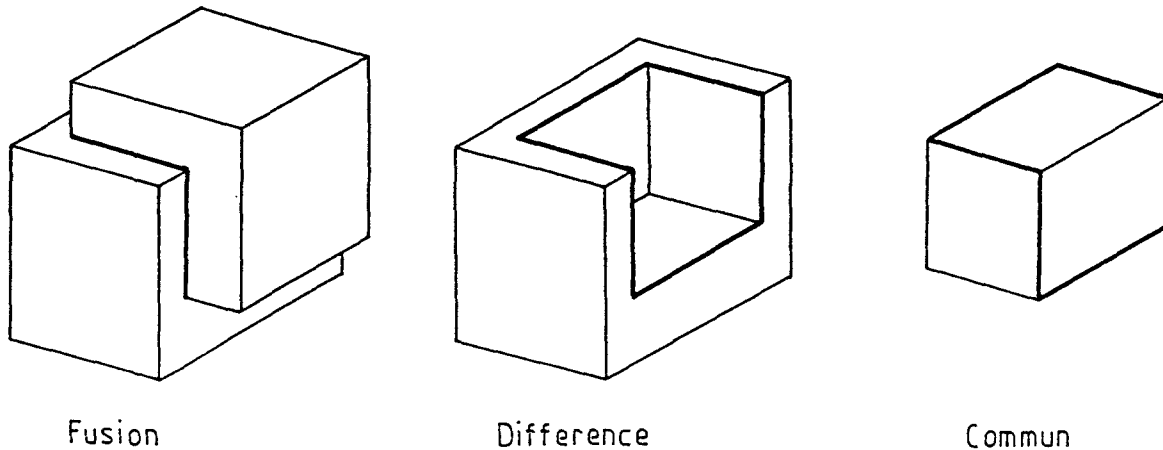


Figure 2.5 : Assemblage des sous_ensembles

L'algorithme des opérations booléennes régularisées peut donc s'écrire de façon générale :

```

OBJET opérations_booléennes(opérateur,A,B)
TYPE_OPERATION opérateur;
OBJET A,B;
{
  OBJET résultat,AextB,AintB,BextA,BintA
  LIGNE ligne_section
  POINT points_intersection

  /* Calcul des intersections A avec B puis B avec A */

  points_intersection = Intersection(A,B)
  points_intersection = Intersection(B,A)

  /* Construction ligne(s) d'intersection des frontières */

  ligne_section = Joindre_points (points_intersection)

  /* diviser objets */

  scinder(A,B,AextB,AintB)
  scinder(B,A,BextA,BintA)

```

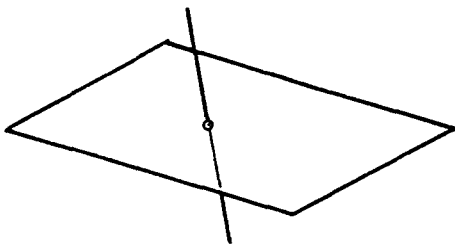
```
/* Construction objet résultant */  
  
selon opérateur  
{  
  FUSION      : résultat = assembler(AextB,BextA)  
              break  
  DIFFERENCE  : résultat = assembler(AextB,BintA)  
              break  
  COMMUN      : résultat = assembler( AintB,BintA )  
              break  
}  
retour résultat  
}
```

3.3 Calcul des points d'intersection

Un sommet de la ligne de section provient toujours de l'intersection d'une arête d'un solide avec une face de l'autre. Pour obtenir tous les points d'intersection, il faut donc parcourir la liste des arêtes d'un solide, et, pour chacune d'elles, déterminer si elle coupe une ou plusieurs faces de l'autre , puis on permute le rôle des objets.

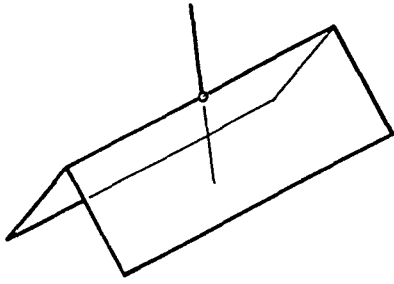
Cependant, afin de pouvoir construire ultérieurement la ligne de section, il est nécessaire de mémoriser sur quelles arêtes et dans quelles faces intervient chaque sommet ainsi calculé.

Or, en fonction du lieu de l'intersection, d'une part sur l'arête et d'autre part sur la face coupée, la quantité d'informations n'est pas la même. On peut en fait recenser six cas différents qui sont:



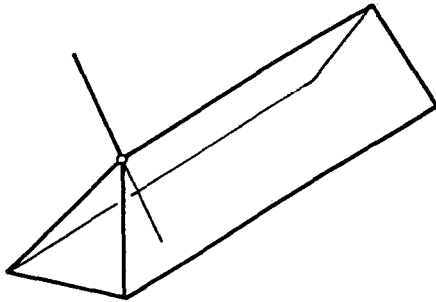
Arête-Face:

Ce point d'intersection affecte une arête et une face.

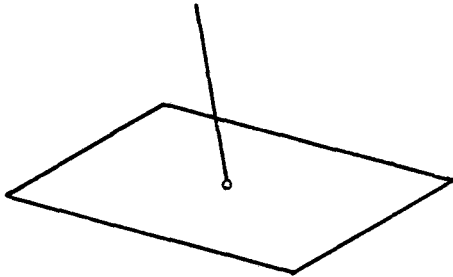


Arête-Arête:

l'arête intersectante et les deux faces connexes à l'arête intersectée sont impliquées.

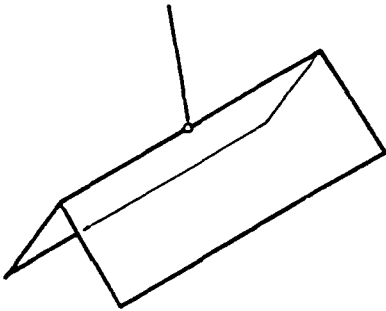


Arête-Sommet: Dans ce cas, cette intersection qui est un des sommets des objets initiaux peut être attribuée à une arête et au moins trois faces.



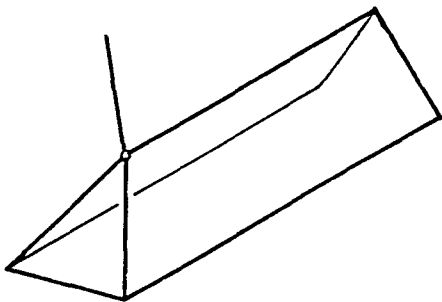
Sommet-Face:

Ce point, appartenant également à l'un des solides intervient dans au moins trois arêtes et une face.



Sommet-Arête:

Symétrique du troisième cas, au minimum trois arêtes et deux faces possèdent ce point.



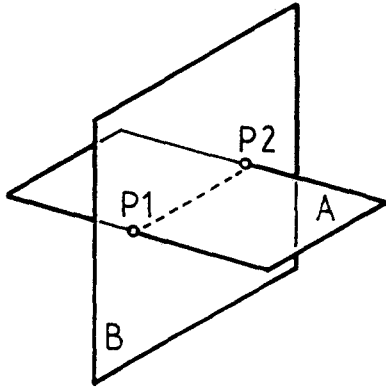
Sommet-Sommet:

Ici, les deux solides partagent un sommet, ce point appartient alors à trois arêtes et trois faces au moins.

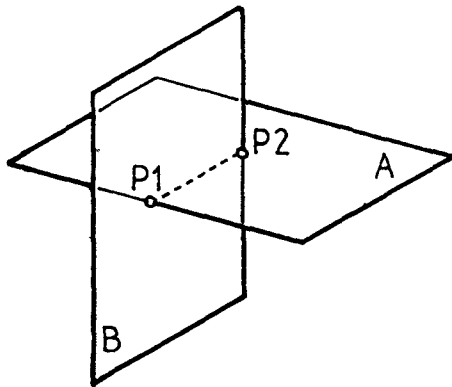
Il faut noter que seul les deux premiers types d'intersection fournissent une quantité invariable de données à mémoriser.

3.4 Jonction des sommets.

L'idée de base, pour construire une arête, est de dire que deux sommets d'intersection peuvent être reliés s'ils appartiennent à la même face aussi bien sur l'objet A que sur l'objet B. Les points d'intersection ayant lieu sur le contour d'au moins une des faces, on peut alors distinguer deux cas :



P1 et P2 sont situés sur le contour de la face de A et dans le face de B, ils permettent donc de construire une arête.



P1 est situé sur le contour de la face de A et dans la face de B, P2 est situé sur le contour de la face de B et dans la face de A, ils peuvent également être joints.

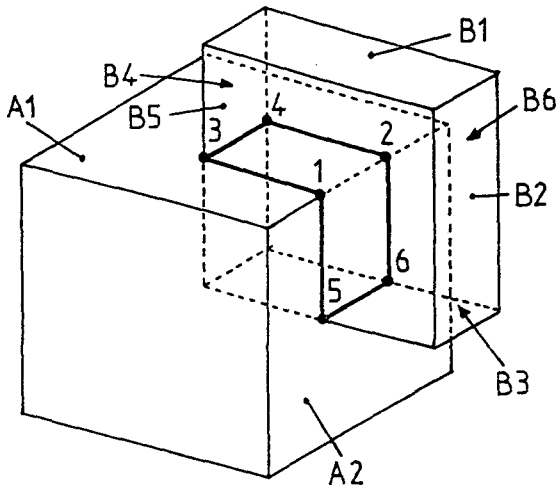
Ces deux cas admettent des variantes si les points d'intersection ne sont pas dans la face intersectée mais sur le contour de celle ci.

En exploitant le principe défini ci dessus, la construction de la ligne de section peut se faire selon la procédure suivante:

On se place sur un point d'intersection (P) quelconque. Celui ci résulte de l'intersection d'une arête partagée par deux faces, que nous appellerons face courante (A) et face connexe (A'), et d'une face de l'autre objet, que nous appellerons (B). On cherche alors si le contour de B possède une intersection (P') avec A'. Dans ce cas, P' est retiré de la liste des points candidats, l'arête PP' est créée, B devient la

face courante (A), la face qui partageait avec B l'arête ou était situé P' devient (A') et A' devient donc (B).

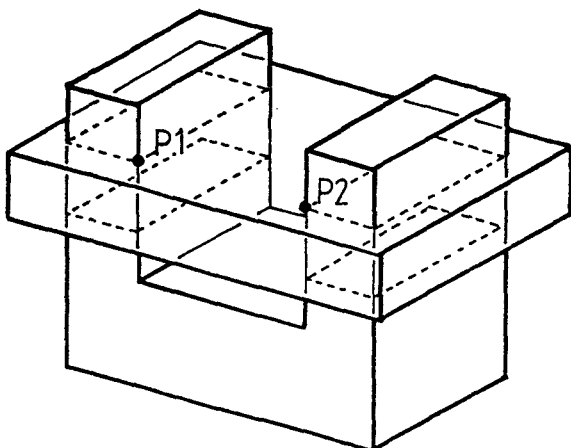
Sinon, le contour de A' doit posséder une autre intersection (P') avec B. A' devient alors la face courante (A), la face connexe (A') et B reste la face intersectée. Ce processus est réitéré jusqu'à ce que la ligne en construction soit fermée.



Point	Face courante	Face connexe	Face coupée
1	A1	A2	B5
5	B5	B3	A2
6	B3	B6	A2
2	A2	A1	B6
4	B6	B4	A1
3	B4	B5	A1
1			

Figure 2.6 : Exemple de construction d'une ligne de section

Cet algorithme ne fournira un résultat correct que si les objets à combiner sont convexes et ont exclusivement des intersections du type Arête_Face. En effet, la condition de joignabilité que nous avons énoncée est nécessaire mais non suffisante si deux faces ont en commun plus de deux points d'intersection.



P1 et P2 vérifie bien la condition d'appartenance mais ne forment pas un segment de l'une des quatre lignes de section qui peuvent être construites dans ce cas.

Pour trouver le successeur correct d'un point d'intersection on peut alors utiliser la méthode suivante [MART87] :

- On ordonne tous les points d'intersection commun à deux faces sur la droite d'intersection des deux plans supports de celles ci.
- Pour un point donné, on pourra alors construire deux segments au plus. On déterminera lequel convient en vérifiant que le point milieu de chaque segment respecte lui aussi la condition d'appartenance. (Fig 2.7)

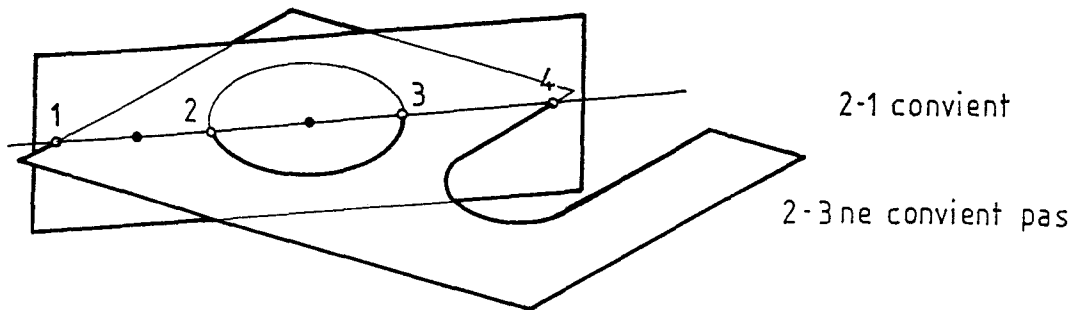
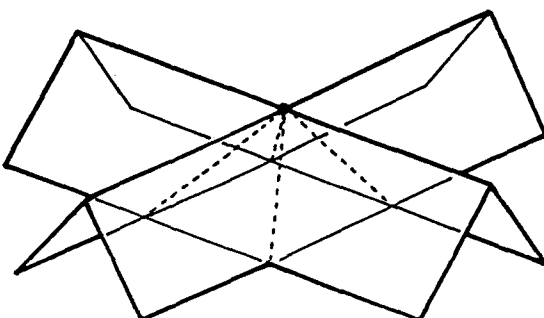


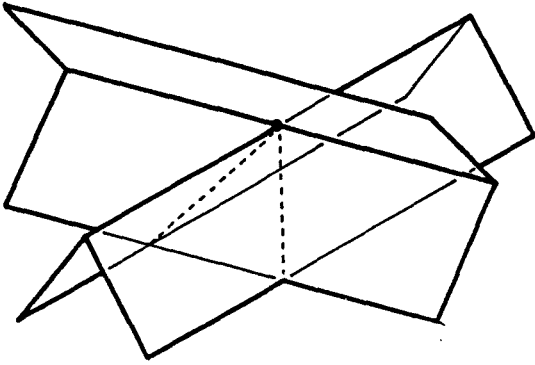
Figure 2.7 : Successeur correct d'un point d'intersection

Ce critère de choix permet de traiter tous les types d'intersection, mais il faut déterminer quel couple de faces utiliser pour chercher le point suivant si l'on se trouve dans le cas où le point courant appartient à plus de trois faces.

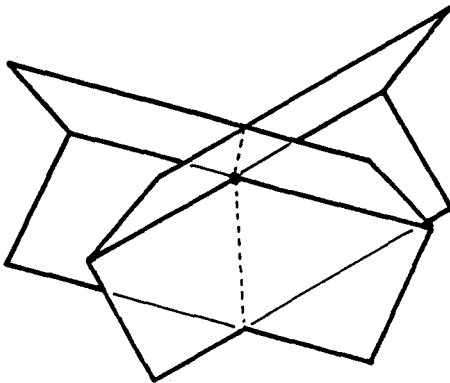
Ainsi, un point d'intersection Arête_Arête appartient à quatre faces qui peuvent prendre différentes configurations, chacune exigeant un traitement particulier :



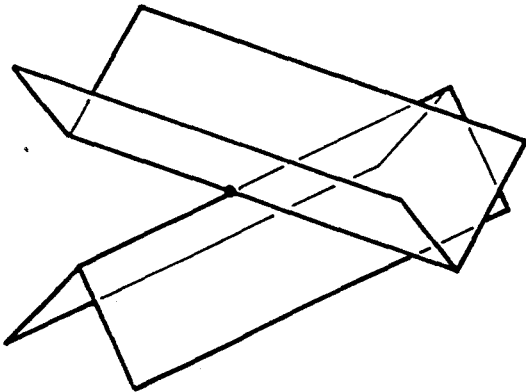
Tous les couples de faces permettent la construction d'un segment.



Une des faces ne peut être traversée par la ligne de section. On se ramène au cas général en n'en tenant pas compte.



Seul deux couples de faces permettent la construction d'une portion de ligne de section. La recherche du point suivant ne se fait pas, dans ce cas, avec la face intersectée mais avec la face adjacente à celle-ci.



Cette configuration ne peut aboutir à la construction d'un segment. Le point est donc simplement ignoré.

Les équations des plans ne suffisent pas à distinguer ces quatre cas. Il faut en effet déterminer la situation de chaque face par rapport au plan contenant les deux arêtes sur lesquelles se trouvent le point d'intersection courant.

Cela peut être réalisé en calculant, pour chaque face, la distance signée d'un de ses sommets n'appartenant pas à l'arête coupée par rapport au plan formée par les deux arêtes intersectantes (Fig 2.8) [MART87].

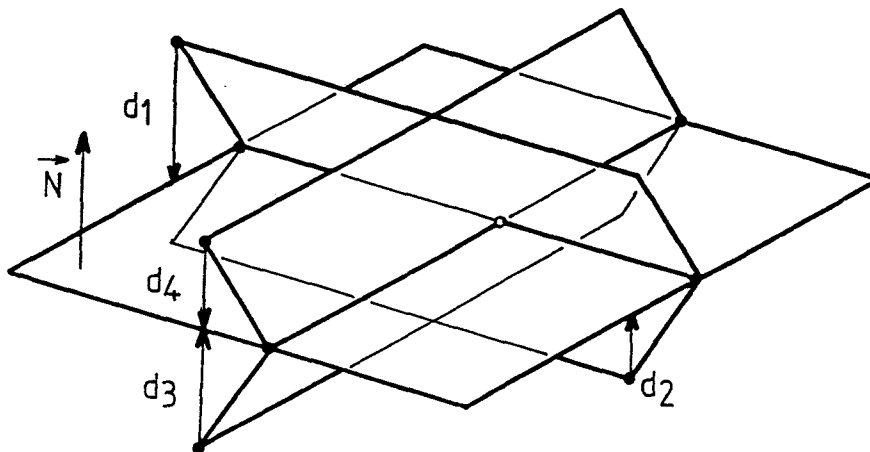


Figure 2.8 : Distinction des différentes configurations d'intersection Arête_Arête

3.5 Découpe des objets.

3.5.1 Insertion ligne de section

La ou les lignes de section étant construites, il faut maintenant les introduire dans la topologie de chaque objet. Cette opération peut être réalisée en utilisant un algorithme de "suivi de contour", dérivé de ceux employés en infographie pour effectuer le clipping polygonal :

- On insère les points d'intersection dans les contours des faces auxquelles ils appartiennent.
- Pour chaque face possédant des points d'intersection, on suit son contour jusqu'à ce que l'on rencontre l'un d'eux. On cherche alors la portion de ligne de section débutant par ce point et dont tous les sommets appartiennent à cette face. Le dernier point vérifiant cette condition doit alors nécessairement appartenir au contour initial de celle-ci, contour que l'on parcourt à nouveau jusqu'au prochain point d'intersection ou jusqu'à ce que le contour en construction soit fermé.

Si une ligne de section appartient dans sa totalité à une face, elle est alors susceptible de former un trou dans celle ci.

Les trous, construits par des lignes de section ou provenant de l'objet originel, sont toujours attribués au contour initial d'une face. Or, celle ci ayant pu être divisée par ailleurs, il est donc indispensable de déterminer à quel contour extérieur appartient chaque trou. (Fig 2.9)

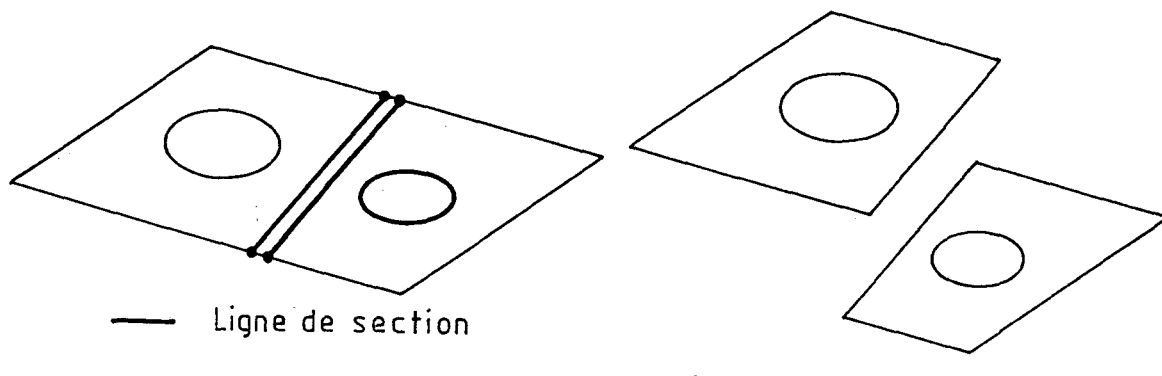


Figure 2.9 : Appartenance des contours internes

3.5.2 Partition des objets

Après division, une face ne peut plus traverser la frontière de l'autre solide. Elle est donc située, soit totalement à l'intérieur, soit totalement à l'extérieur. Chaque objet devant être divisé en deux sous ensembles de faces possédant la même situation, on peut procéder au test d'intériorité pour chacune d'elles. Il est toutefois beaucoup plus rapide de ne faire ce calcul qu'une seule fois puis d'exploiter les connectivités existantes entre les faces pour en déduire leurs situations.

En effet, deux faces partageant une arête initiale ont obligatoirement le même état, tandis que deux faces ayant en commun une arête appartenant à une ligne de section ont des états différents car cette arête est située sur la frontière des deux volumes.

3.6 Construction du résultat

Cette étape ne pose aucun problème algorithmique. Elle consiste simplement à rassembler, en fonction de l'opération désirée, une partie de chaque objet (Tab 2.1).

	AextB	AintB	BextA	BintA
Fusion	*		*	
Différence	*			*
Commun		*		*

Tableau 2.1 : Choix des parties à conserver

Dans le cas de la différence il est toutefois nécessaire d'inverser le sens de description des faces de BintA car l'intérieur de l'objet soustrait est nécessairement l'extérieur de l'objet résultat.

3.7 Insuffisances

Lors de l'étude de l'algorithme construisant la ligne de section, nous n'avons pas traité les cas où intervenaient certains types d'intersections, notamment ceux où des sommets des solides initiaux étaient impliqués.

Si la fréquence d'apparition de ces types particuliers est très faible, il peut être acceptable de les négliger. La recherche de combinaisons d'objets les produisant systématiquement permet de distinguer trois familles :

- Les combinaisons d'objets où le résultat théorique est un volume n'ayant pas de réalité physique. Les objets qui partagent un seul sommet ou une seule arête peuvent être classés dans cette famille [FLA 87] (Fig 2.10).

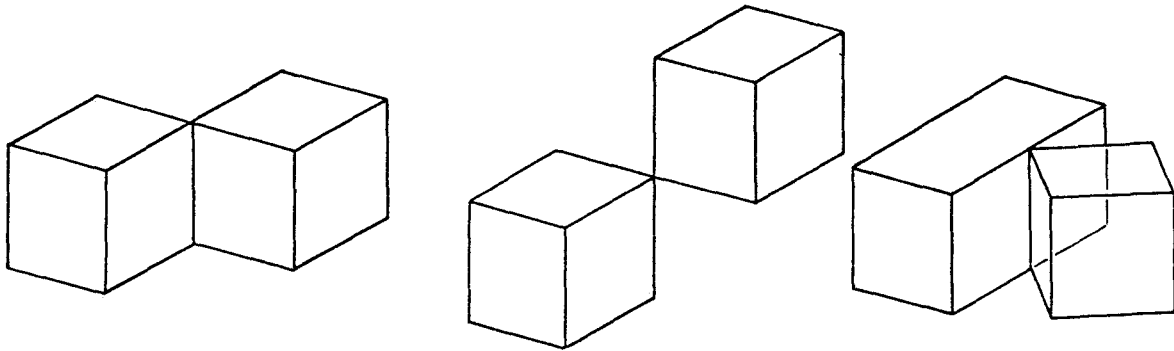


Figure 2.10 : Aberrations géométriques

L'échec de l'algorithme dans de tels cas n'est pas gênant et peut même être exploité pour détecter toutes tentatives de création de telles aberrations géométriques.

- Les combinaisons multiples d'un même objet occupant différentes positions dans l'espace (Fig 2.11a) ou les combinaisons de solides impliquant un raccordement de surfaces théoriques de types différents (Fig 2.11b)

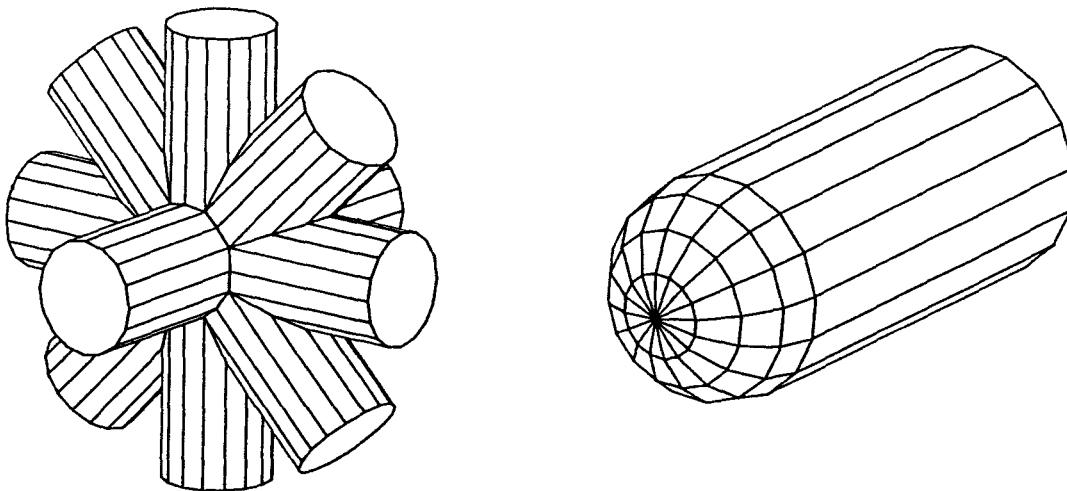


Figure 2.11 : Combinaisons multiples et raccordements de surfaces différentes

De telles combinaisons n'entraînent pas un échec systématique de l'algorithme et, dans ce contexte, il semble qu'un traitement approprié de ces points d'intersection problématiques puisse être trouvé. [MART87]

- Les combinaisons d'objets qui possèdent des faces coplanaires, catégorie qui regroupe également les "collages". (Fig 2.12)

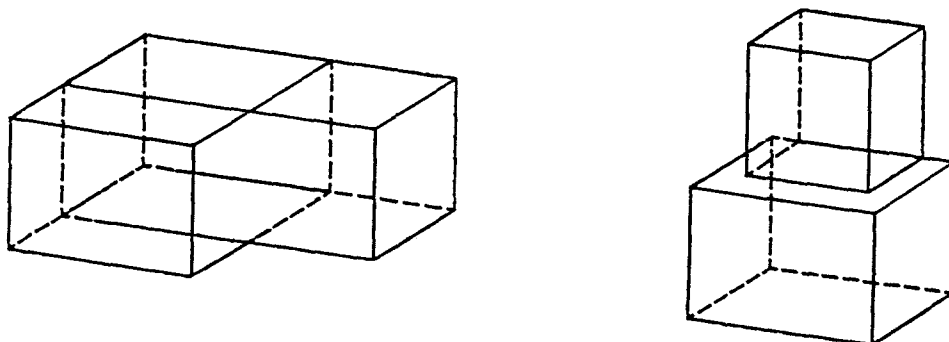


Figure 2.12 : Faces coplanaires

Les problèmes posés par cette famille sont beaucoup plus intéressants car la fréquence des opérations impliquant des objets ayant cette caractéristique est relativement élevée, particulièrement lors de l'évaluation d'arbres de construction possédant peu de solides primitifs. Si il est possible de réduire fortement le nombre de ces cas, c'est au prix de modifications importantes de l'arbre C.S.G (Fig 2.13), imposant ainsi une méthodologie de conception rigide pour l'utilisateur du système ce qui limite considérablement son intérêt.

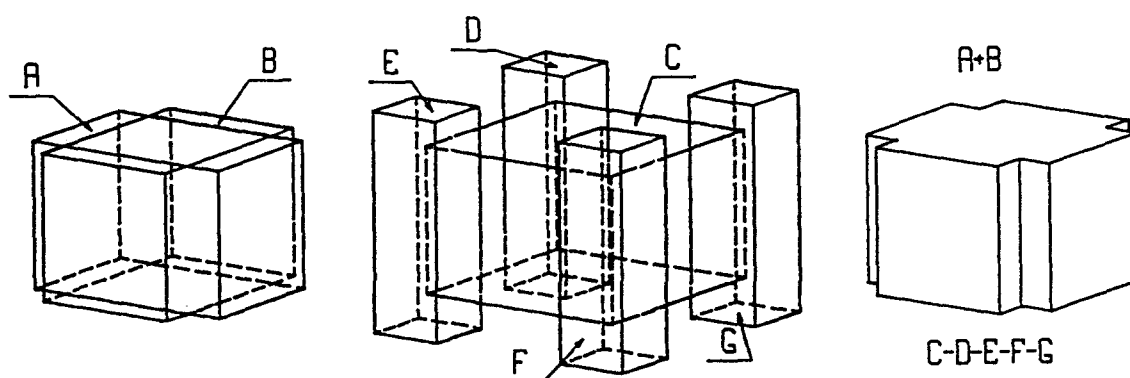


Figure 2.13 : Suppression des cas problématiques

Enfin, traiter ces cas demandera des modifications très profondes de l'algorithme car, dans cette situation, la ligne de section n'existe pas. En effet, l'intersection des frontières des deux solides possédant des faces coplanaires

ne produit plus une ou plusieurs lignes polygonales mais un assemblage de lignes et de surfaces. (Fig 2.14)

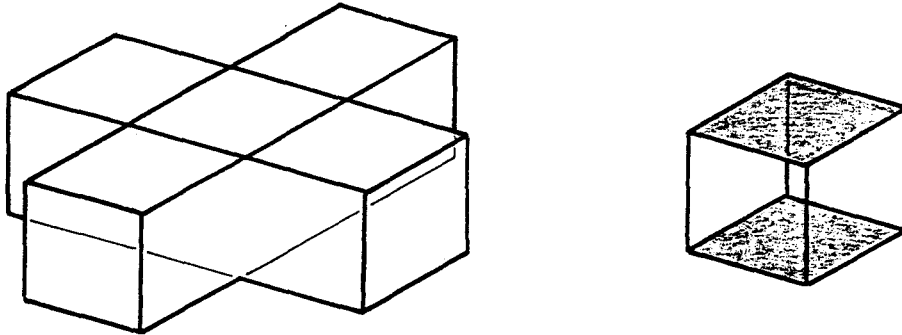


Figure 2.14 : Intersection de solides possédant des faces coplanaires

Il est donc indispensable de concevoir un nouvel algorithme réalisant la jonction des sommets afin qu'il produise cette intersection. De plus, les frontières des objets ne doivent plus être divisées en deux parties mais en trois :

- à l'intérieur de l'autre objet
- à l'extérieur de l'autre objet
- sur la frontière de l'autre objet

L'apparition de ce troisième état pour une face exigera naturellement un traitement approprié lors de la phase de construction du résultat.

La prise en compte des faces coplanaires conduirait donc à refondre en grande partie cet algorithme déjà complexe et délicat à mettre en oeuvre ou, plus radicalement, à envisager une approche différente.

4. Algorithme "subdivision de faces"

4.1 Classe d'objet

Appartenant également à la famille des représentations volumiques par les frontières, les objets supportés par ce type d'algorithme peuvent être concaves mais pas les facettes constituant leurs polyèdres.

Cette restriction n'est pas réellement contraignante car le résultat d'une opération booléenne est un solide conforme à

cette définition qui pourra donc être l'un des arguments d'une autre combinaison.

De plus, la conversion d'un objet possédant des faces concaves est aisée, les algorithmes décomposant un polygone concave en plusieurs polygones convexes ayant déjà été développés pour les besoins de l'infographie, notamment pour effectuer du remplissage de taches [MAR 84].

4.2 Principe de l'algorithme

Cet algorithme modifie la topologie de chaque objet au fur et à mesure de l'apparition des intersections, de telle sorte qu'à la fin de ce calcul il suffit de choisir, sur chaque solide, les faces appartenant au résultat.

Toute face d'un objet coupant une face de l'autre est donc immédiatement divisée en plusieurs faces connexes dont l'un des éléments, sommet ou arête, appartient à l'intersection des polygones initiaux. Ainsi, l'intérieur des nouvelles faces créées ne coupent plus la frontière de l'autre objet (Fig 2.15).

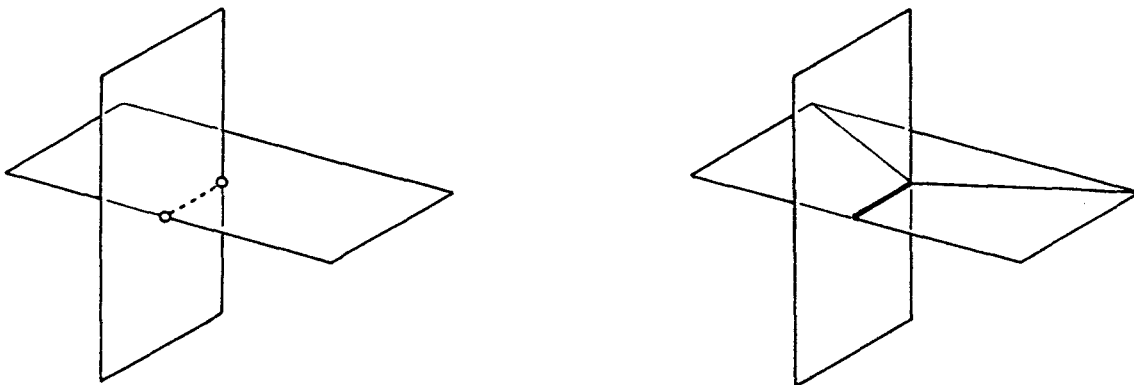


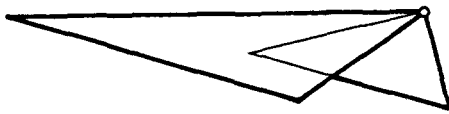
Figure 2.15 : Division d'une face

L'algorithme que nous allons détailler travaille sur des solides dont toutes les faces sont triangulaires. Une représentation de ce type n'altère en rien les principes mis en oeuvre mais apporte une grande simplicité, tant au niveau des données, toutes les faces ont la même structure et sont toujours planes et convexes, qu'au niveau algorithmique. En effet, déterminer, par exemple, si un point est à l'intérieur d'un contour est alors particulièrement aisé.

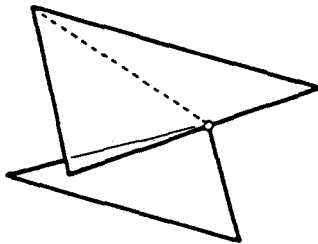
4.3 Division des objets

Nous devons calculer pour chaque face d'un objet son intersection éventuelle avec chaque face de l'autre solide. L'intersection de deux polygones convexes peut être un point, un segment ou même un polygone et, selon la position de cette intersection sur la face, le découpage sera différent.

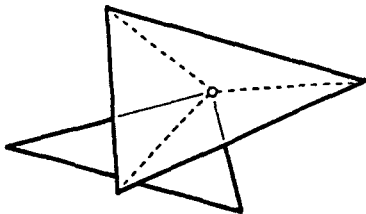
Trois cas différents peuvent être répertoriés lorsque l'intersection de deux faces est un point :



Sur un sommet du contour initial. Aucun découpage n'est donc nécessaire.

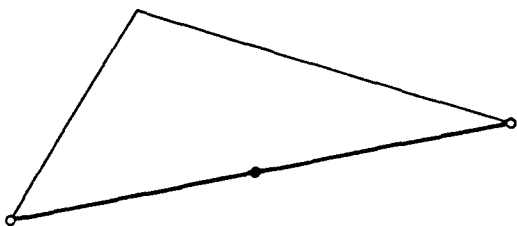


Sur une arête. La face est coupée en deux en créant le segment liant le point trouvé et le sommet n'appartenant pas à l'arête intersectée.



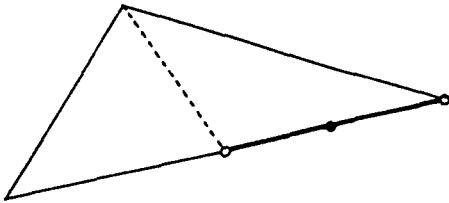
Dans la face. Celle ci est divisée en trois après construction des arêtes joignant le point calculé aux sommets initiaux.

Si l'intersection est un segment, on distingue tous les cas possibles par la position de ses deux extrémités et de son point milieu sur la face. Si l'on exclut les configurations équivalentes, huit cas sont susceptibles d'être rencontrés :

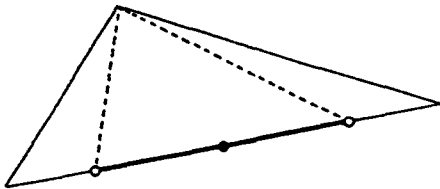


Sommet_Arête_Sommet

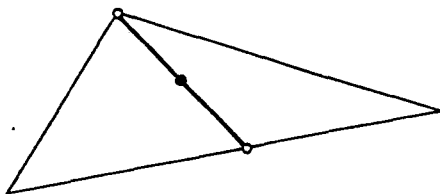
Aucune division n'est nécessaire



Sommet_Arête_Arête

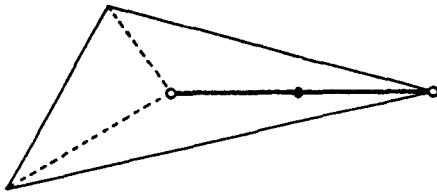


Arête_Arête_Arête

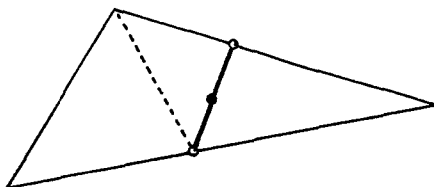


Sommet_Face_Arête

C'est le segment d'intersection qui scinde la face en deux parties.

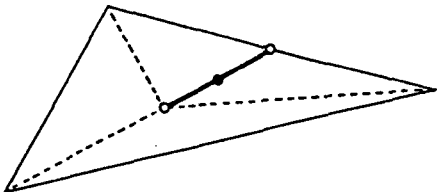


Sommet_Face_Face

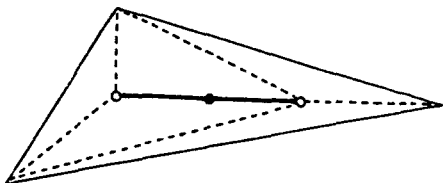


Arête_Face_Arête

Deux possibilités de découpage qui produiront l'une et l'autre quatre faces.



Arête_Face_Face



Face_Face_Face

Deux alternatives, mais cinq triangles seront toujours créés.

Une intersection de type polygone provient de l'intersection de deux faces coplanaires et n'entraîne pas de découpage.

En effet, ce polygone est formé de portions d'arêtes qui peuvent être partagées par des faces non coplanaires à la face traitée et qui produiront alors les subdivisions indispensables.

Les zones coplanaires, communes aux deux objets, ne seront pas nécessairement formées, après subdivision, des mêmes polygones sur chacun d'eux. C'est toutefois sans importance car la surface délimitée est identique et ne sera conservée que sous une forme dans le résultat.

Cet algorithme de subdivision d'objet est appliqué trois fois. La première passe morcelle les faces du premier objet jusqu'à ce qu'il n'y ait plus d'intersections à l'intérieur de celles ci. Lors du second passage, la subdivision des faces du deuxième objet ne peut pas créer d'arêtes coupant l'intérieur d'une face du premier. Par contre, des intersections de segments peuvent être générées et impliquent donc une seconde division de l'objet A. Les nouvelles arêtes créées au cours de cette troisième passe sont toujours situées à l'intérieur des faces du premier objet et ne peuvent donc posséder d'intersections avec l'intérieur d'une face ou avec un segment du second solide.

4.4 Construction du résultat

Au préalable, la situation des faces de chaque objet vis à vis de l'autre solide est calculée. Quatre états peuvent être affectés à une face :

- dehors,
- dedans,
- dessus et de même normale,
- dessus et de normale opposée,

les deux derniers caractérisant les faces coplanaires.

l'objet final est alors construit en regroupant des sous_ensembles de faces de même état de chaque objet comme indiqué dans les tableaux suivants [LAI 86].

Objet A	dedans	dehors	même	opposée
Fusion		*	*	
Différence		*		*
Commun	*		*	

Tableau 2.2 : Critères de choix des faces de l'objet A

Objet B	dedans	dehors	même	opposée
Fusion		*		
Différence	*			
Commun	*			

Tableau 2.3 : Critères de choix des faces de l'objet B

Les faces classées dessus existent sur les deux objets, elles ne sont donc jamais conservées pour l'un deux. Enfin, dans le cas de la différence, le sens de description des contours des faces de l'objet B qui appartiennent au résultat doit être inversé car ce qui est à l'intérieur à cet objet est à l'extérieur de l'objet final.

4.5 Evaluation

Cet algorithme, simple dans son principe et sa mise en oeuvre car systématique, traite théoriquement tous les cas de combinaisons de deux solides.

Cependant, il introduit un découpage excessif des frontières des objets, produisant ainsi un résultat inutilement complexe et gourmand en mémoire. Ce phénomène n'est d'ailleurs pas sans

conséquence sur le bon déroulement de l'algorithme car il provoque :

- Une augmentation des temps de traitements, due à l'accroissement du nombre des faces, lorsque l'on réutilise le résultat dans d'autres combinaisons mais également de l'opération booléenne en cours d'évaluation. En effet, à chaque intersection calculée des faces sont créées et devront être prises en compte lors des phases suivantes. De plus, ces faces ne sont pas facilement éliminées, à priori, par une procédure de filtrage du type "boite englobante" car elles n'apparaissent qu'à proximité du lieu de l'intersection des deux objets.
- Une apparition rapide des problèmes liés à l'imprécision des calculs sur des nombres réels. Afin de compenser ces erreurs, toute comparaison de deux réels est faite avec une tolérance (epsilon) . Ainsi, dans le cas des calculs géométriques, deux points seront considérés comme identiques si leur distance est inférieure à une certaine valeur. Ce procédé a des inconvénients dont le plus important est la perte de transitivité de la relation d'égalité (Fig 2.16).

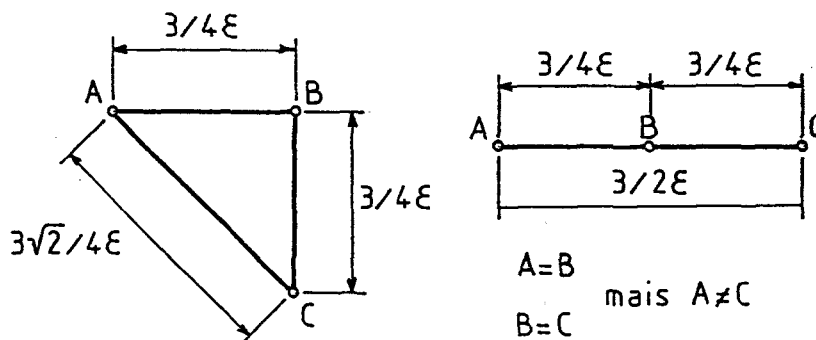
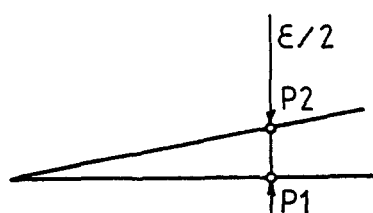


Figure 2.16 : Inconvénients liés à l'emploi des "epsilon"

Dans le contexte de la modélisation géométrique par les frontières, l'utilisation de ces tolérances peut conduire à des contradictions entre les informations issues des calculs géométriques et celles contenues dans la topologie de l'objet (Fig 2.17).



P1 et P2 sont situés sur des arêtes distinctes
mais ils seront considérés comme
identiques dans une comparaison

Figure 2.17 : Contradictions entre les résultats des calculs et les informations topologiques.

La subdivision excessive des frontières des objets produite par cet algorithme engendre nécessairement des faces de faibles dimensions et aboutira alors inévitablement à ce genre de contradiction lors des calculs d'intersections de deux faces provoquant ainsi son échec.

Le choix d'utiliser des facettes exclusivement triangulaires n'est pas obligatoirement la cause d'une apparition plus rapide de ces cas. En effet, même si l'on utilise un procédé de découpages des faces en polygones convexes quelconques, une face possédant initialement un contour simple (3 ou 4 points) et un nombre assez important d'intersections avec l'autre objet produira une subdivision problématique (Fig 2.18). Cette situation est malheureusement fréquente puisque tout perçage d'une boîte est susceptible de l'engendrer, interdisant pratiquement la réutilisation du résultat dans d'autres combinaisons.

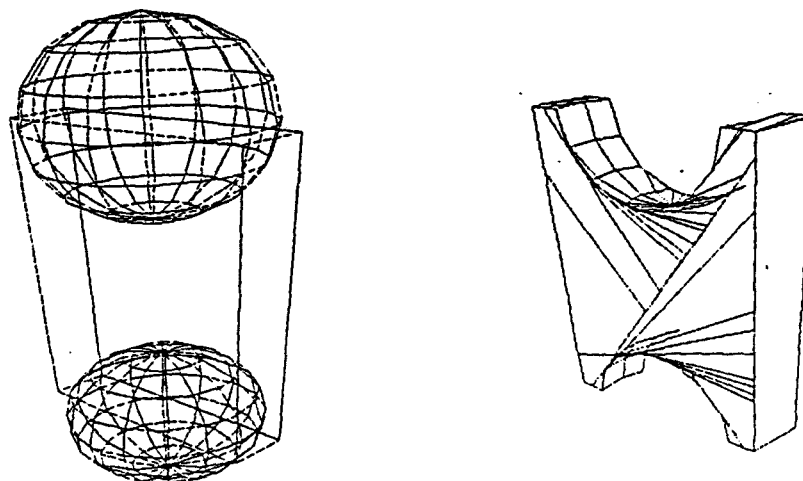


Figure 2.18 : Subdivision excessive des faces (BAD 88)

5. Conclusion

Malgré des approches fondamentalement différentes, aucun des deux algorithmes étudiés n'apporte une solution réellement fiable au problème posé par les combinaisons booléennes de deux objets.

Si il est possible, par un choix judicieux des volumes à traiter en fonction du résultat recherché, de pallier partiellement à leurs défauts, ce procédé peut difficilement être utilisé lors de l'évaluation d'un solide décrit par un arbre de construction sous forme B_REP.

En outre, identifier les cas conduisant systématiquement un algorithme à l'échec n'est pas une tâche aisée car ils sont étroitement liés à son principe mais aussi à sa réalisation qui est une opération délicate de par la taille des programmes à mettre en oeuvre.

L'analyse des comportements des algorithmes face à ces cas conduit à remettre en cause leurs principes mais laisse apparaître une certaine complémentarité de leurs déficiences. Si celle ci ne peut être exploitée par une utilisation alternative, selon le cas, de l'un ou de l'autre de ces deux algorithmes, elle devrait permettre d'établir un concept plus général.

Chapitre 3

Opérateurs booléens

Algorithme proposé

1. Introduction

Les deux familles d'algorithme que nous avons présentées n'ont pas une approche suffisamment générale pour traiter tous les cas particuliers qui se présentent lors de l'évaluation d'un arbre de construction.

Ce fait et la spécificité du modèle de produit hybride que nous avons défini, notamment les liens introduits entre les faces de la représentation par les frontières et les surfaces des primitives de l'arbre de construction, nous ont amené à définir notre propre algorithme dont nous allons expliciter le principe et les différentes étapes.

2. Principe de l'algorithme

Afin de ne pas introduire d'éléments non essentiels dans la topologie du résultat d'une opération booléenne, nous avons élaboré un algorithme qui effectue la partition des surfaces polyédriques des arguments de l'opération postérieurement aux calculs de leurs intersections.

Cependant, pour assurer un traitement efficace et fiable des cas "limites" et plus particulièrement des cas de coplanéité de faces, la notion de "ligne de section" a été abandonnée au profit d'une méthode dérivée des procédés de recherche de cycles dans un graphe.

Ainsi, pour chaque face susceptible d'être modifiée par l'opération menée, un graphe est construit dont les noeuds sont les sommets de cette face et dont les arcs, qui établissent une relation entre deux noeuds, représentent les arêtes.

Les arcs et les noeuds d'un graphe sont typés selon qu'ils résultent du calcul de l'intersection des surfaces polyédriques des solides à combiner ou qu'ils proviennent de ces solides. De plus, seuls les sommets et les arêtes susceptibles d'appartenir au résultat de l'opération, en fonction du type de l'opération booléenne et de leurs situations vis à vis de l'autre objet, sont introduits dans un graphe (Fig 3.1).

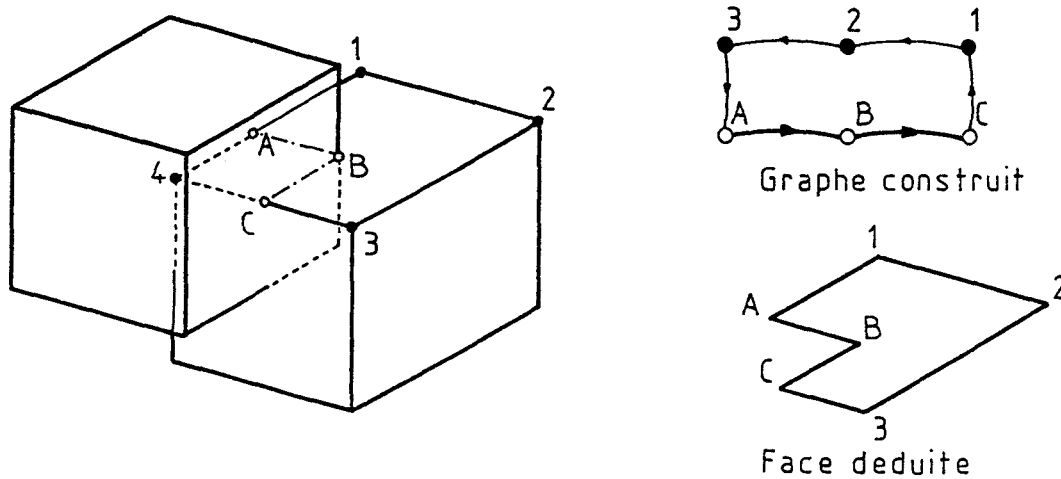


Figure 3.1 : Exemple de graphe construit pour une face

Les ensembles d'arcs qui forment des cycles dans un graphe permettent alors de déduire les contours d'une face modifiée par l'intersection. Lorsque plusieurs arcs partent d'un même noeud, l'arc qui permet de suivre le cycle correct est choisi en fonction de son caractère de nouveauté vis à vis des objets initiaux. Ce critère découle du fait que la combinaison de deux solides fait nécessairement apparaître des éléments nouveaux dans le résultat.

Les faces coplanaires exigent une construction de graphe et une recherche de cycles spécifiques car plusieurs faces des objets à combiner possédant cette propriété ne produisent souvent qu'une face unique (Fig 3.2).

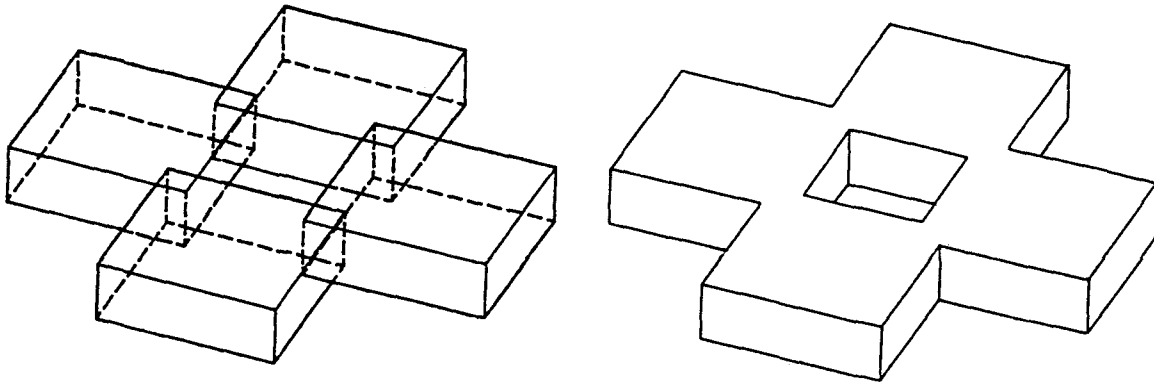


Figure 3.2 : Face produite par l'association de plusieurs faces coplanaires

L'emploi de cette méthode impose certaines contraintes aux étapes préliminaires de calcul des points d'intersection et de création des arêtes les liants. En effet, les composantes d'un graphe doivent être unique, sous peine de voir apparaître des noeuds desquels ne parte ou n'arrive aucun arc ou la construction de cycles bouclés sur des éléments dupliqués.

Pour respecter cette condition, les sommets d'intersection sont insérés dès leur calcul dans la ou les arêtes concernées, provoquant une modification progressive de la topologie des deux objets. Tout calcul d'intersection ultérieur qui tendrait à dupliquer un sommet fournira alors comme résultat le point déjà créé et inséré.

Les options générales faites pour bâtir cet algorithme permettent de distinguer trois grandes phases

- calcul et insertion des points d'intersection,
 - création des arêtes qui lient ces points,
 - construction et recherche des cycles d'un graphe,
- dont nous allons préciser la fonction et détailler les méthodes utilisées pour remplir cette fonction.

3. Classe d'objets

Les solides manipulés par notre algorithme doivent respecter certaines propriétés qu'ils partagent d'ailleurs avec les solides du monde réel [REQ 80] :

- Un solide doit posséder un intérieur et les excroissances planes ou filiformes sont proscrites.
- Un solide doit occuper une portion finie d'espace.
- Un solide est indéformable.

Un solide peut toutefois être en plusieurs parties lesquelles ne sont pas nécessairement convexes. Les facettes polygonales formant les frontières de ces solides peuvent être concaves et posséder des contours intérieurs correspondant à des trous.

La seule restriction que nous ferons concerne le sens de description des contours des faces. Ainsi, pour un observateur situé à l'extérieur d'un objet, le contour externe d'une face devra toujours être décrit dans le sens horaire tandis que les éventuels contours internes seront décrits dans le sens trigonométrique (Fig 3.3). Cette convention permet de diriger la normale à toute face vers l'intérieur de l'objet, normale qui sera exploitée par la suite pour donner une orientation aux arêtes nouvelles lors de leur création

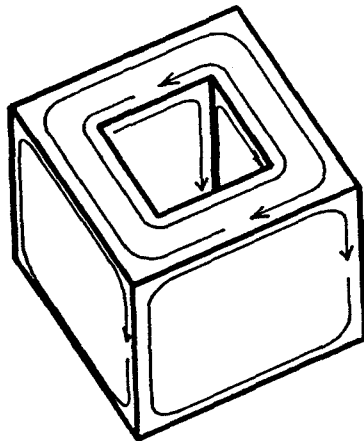


Figure 3.3 : Orientations des contours des faces

4. Création des sommets

Le principe général de l'algorithme confère en fait deux fonctions à cette étape :

- Le calcul des points d'intersection des polyèdres représentant les frontières des deux objets à combiner.
- L'insertion, dès leur apparition, des points trouvés dans les polyèdres.

Un point d'intersection ne pouvant provenir que de l'intersection d'une arête d'un solide avec une face de l'autre objet, le processus suivant est alors appliqué à tous les couples constitué d'une arête et d'une face de chaque solide :

- Calcul du point d'intersection de la droite support de l'arête avec le plan contenant la face.
- Si ce point existe, sa situation vis à vis de l'arête et de la face est calculée.
- Si le point appartient à ces deux entités, il est alors ajouté à la topologie des deux objets.

4.1 Calcul du point d'intersection

Ce calcul ne posant mathématiquement et algorithmiquement aucun problème, nous ne le détaillerons donc pas. Toutefois, le cas où l'arête est contenue dans le plan de la face peut sembler digne d'intérêt car il est susceptible de produire plusieurs intersections (Fig 3.4a). Il est en fait ignoré, les points nécessaires étant générés par les faces connexes (Fig 3.4b).

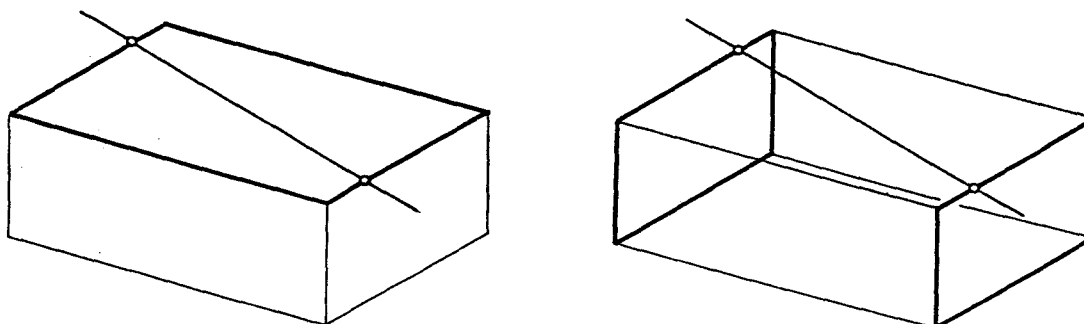


Figure 3.4 : Intersections produites par une arête et une face situées dans le même plan

4.2 Situation du point

L'objectif de ce calcul est double car il permet de vérifier que le point précédemment trouvé constitue bien une intersection de l'arête et de la face et fournit les données indispensables à son insertion. En effet, selon que ce point se trouve sur l'arête ou sur l'un de ses sommets, à l'intérieur de la face ou sur son contour, le traitement sera différent.

Vis à vis de l'arête, un point peut avoir trois situations distinctes :

- à l'extérieur et il est alors rejeté
- sur l'arête
- confondu avec l'une de ses extrémités,

situation qu'il est extrêmement aisée d'établir. Par contre, sa localisation sur la face est beaucoup plus délicate à déterminer, seuls ses contours étant connus, et peut prendre quatre valeurs :

- à l'extérieur et ce n'est pas une intersection réelle,
- à l'intérieur,
- sur une arête,
- sur un sommet.

La méthode que nous avons employée consiste à dénombrer les intersections que possède une demi_droite, dont l'origine est le point à localiser et de direction quelconque mais parallèle au plan de la face, avec le ou les contours de celle ci [HEG 85]. Si ce nombre est pair, le point est situé à l'extérieur, et s'il est impair le point est à l'intérieur. Enfin, si un des points d'intersection est identique au point testé, celui ci est sur le contour. Cependant, quand une intersection a lieu sur un sommet de la face, elle doit être comptabilisée si la demi_droite traverse le contour (Fig 3.5a) et omise si ce n'est pas le cas (Fig 3.5b).

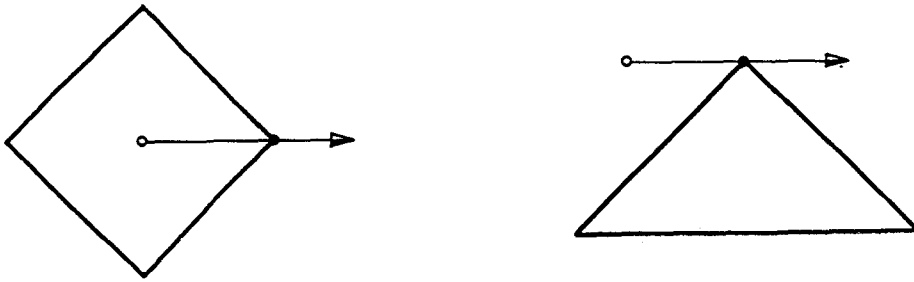


Figure 3.5 : Comptabilisation des Intersections

Plutôt que de recourir à des procédés complexes pour lever cette ambiguïté, nous avons préféré imposer une direction légèrement différente à la demi_droite, cette direction n'ayant pas d'influence sur le résultat, et réitérer tout le processus en prenant soin de vérifier au préalable que le point n'est pas sur un sommet du contour. Cette précaution est nécessaire car, dans ce cas, quelque soit la direction donnée à la demi_droite, on trouverait toujours une intersection sur un sommet et les calculs pourraient être poursuivis indéfiniment. Ce principe, conçu à l'origine pour résoudre ce type de problèmes dans un espace bidimensionnel, peut être étendu de trois façons différentes à un espace tridimensionnel :

- En effectuant tous les calculs (intersection demi_droite contour) dans cet espace.
- En projetant les contours et le point sur le plan contenant la face pour revenir à un problème plan.
- En utilisant une des trois projections sur les plans principaux (XoY, XoZ , YoZ) de l'espace tridimensionnel implicitement existantes des éléments à traiter : Il suffit de négliger une coordonnée choisie en fonction de l'orientation du plan contenant la face.

La dernière méthode est la plus simple à mettre en oeuvre et la plus rapide, mais l'imprécision des calculs et les altérations des distances entre les points dues à la projection peuvent conduire à des conclusions erronées si le point à localiser est proche de l'un des contours de la face. Dans ce

cas, une vérification dans l'espace originel, par le calcul de la distance du point au contour, permettra de confirmer ou d'infirmier le résultat obtenu.

4.3 Insertion des points d'intersection

La représentation par les frontières d'un objet est une structure fragile et ajouter un sommet à un objet entraîne des modifications à tous les niveaux d'entités présents dans cette structure afin qu'elle reste cohérente.

Cependant, c'est une structure hiérarchisée dans laquelle un point, qui est l'élément de plus bas niveau, pourra être introduit de deux manières différentes seulement. De ce fait, deux opérations élémentaires d'insertion d'un point sont définies :

- Division d'une arête : L'extrémité d'arrivée de l'arête est remplacée par le point d'intersection, une nouvelle arête est créée dont l'origine est le point d'intersection et l'extrémité d'arrivée celle de l'arête divisée. Enfin, les contours des deux faces auxquelles l'arête appartenait sont modifiés pour intégrer l'arête créée.
- Remplacement d'un sommet par un point d'intersection : il faut alors procéder à la modification de toutes les arêtes qui possèdent ce sommet.

De plus, pour chaque face, une liste des intersections ayant eu lieu à l'intérieur de ses contours est maintenue car ces points ne sont pas insérés dans la topologie de l'objet bien qu'ils lui appartiennent.

Ces trois opérations élémentaires sont appliquées selon le nombre et la nature des entités concernées par une intersection et dépendent de la situation du point calculé. Celui ci pouvant avoir deux situations différentes sur l'arête et trois vis à vis de la face, on pourra donc être confronté à six cas distincts, chacun exigeant une combinaison spécifique des trois opérateurs, comme indiqué dans le tableau suivant :

Lieu intersection		Opérateur appliqué à l'objet possédant l'arête	Opérateur appliqué à l'objet possédant la face
arête	Face		
Sur	Intérieur	Division arête	Ajout point
Sur	Contour	Division arête	Division arête
Sur	Sommet	Division arête	Remplacement point
Extrémité	Intérieur	Remplacement point	Ajout point
Extrémité	Contour	Remplacement point	Division arête
Extrémité	Sommet	Remplacement point	Remplacement point

Tableau 3.1 : Insertion des points d'Intersection

Parmi les six types d'intersection répertoriés, quatre impliquent des sommets qui peuvent alors être des points d'intersection déjà insérés. Ces cas particuliers ne doivent pas être négligés car, si certains sont valides et ne demandent alors aucun traitement, d'autres dénotent une incohérence des objets initiaux ou une localisation erronée du point d'intersection.

Deux configurations peuvent être correctes lorsqu'elles font réapparaître un point déjà calculé :

- **Extrémité_Intérieur** si le point d'intersection, extrémité de l'arête, a déjà été imputé à la face. Dans le cas contraire, on recherchera la présence éventuelle du point dans l'un des contours de la face, présence qui impliquerait une défaillance sans conséquence de l'algorithme de localisation car les deux volumes restent cohérents. Enfin, si cette deuxième condition n'est pas satisfaite, le point peut être attribué à la face, mais la validité des objets et du résultat devra être vérifiée car cette situation peut indiquer deux faces d'un même objet qui se recouvrent.
- **Extrémité_sommet** si l'extrémité et le sommet ont déjà été remplacés par le même point d'intersection. Sinon, on peut être confronté à trois situations anormales :

- * L'extrémité de l'arête est un point d'intersection mais pas le sommet. La localisation de l'intersection pourra seule être mise en cause si le point d'intersection appartient à l'un des contours ou à l'une des faces connexes au sommet incriminé.
- * Le sommet est un point d'intersection mais pas l'extrémité de l'arête. Si ce point a été affecté à l'une des faces partageant cette arête, on pourra, la encore, conclure à une mauvaise localisation du point.
- * Il s'agit de deux points d'intersection distincts, situation qui indique une incohérence des deux solides.

Les deux types restants, **Sur_Sommet** et **Extrémité_Contour**, sont en fait similaires car ils font apparaître tous les deux une arête non divisée par un point d'intersection.

La même procédure, destinée à vérifier la cohérence des objets, peut donc leur être appliquée :

- On contrôlera que le point d'intersection retrouvé n'est pas, en fait, l'extrémité de l'arête sur laquelle il est censé être, indiquant une situation de ce point inexacte sur celle ci.
- Dans le cas contraire, on vérifiera que ce point d'intersection n'a pas été attribué à la liste des intersections internes de l'une des faces adjacentes à l'arête problématique impliquant encore une mauvaise localisation.

Enfin, lors du calcul des intersections d'une arête du deuxième objet avec les faces du premier, trois types d'intersection ne devraient pas surgir :

- arête sur contour,
- arête sur sommet,
- extrémité sur contour,

En effet, ils matérialisent des intersections qui ont lieu simultanément sur le contour d'une face de chaque objet. Les

points d'intersection correspondant devraient donc déjà avoir été calculés et insérés lorsque le rôle des objets était inversé.

Des vérifications de validité, semblables à celles évoquées pour les cas précédents et dont le but est de retrouver, sous une autre forme, l'intersection improbable dans les entités voisines, doivent être employées.

Les nombreux cas difficiles énumérés ne doivent pas être considérés comme autant de source d'erreurs possibles car ils permettent, au contraire, d'offrir une certaine insensibilité de l'algorithme à l'imprécision affectant les résultats des calculs effectués sur des "réels". En effet, lors de l'analyse de ces cas, nous avons souvent mis en évidence une erreur de localisation du point d'intersection comme cause possible de ces problèmes, erreur généralement due à l'emploi de tolérances sur les résultats des calculs afin de pallier à leur imprécision.

L'apparition de ces cas et les vérifications de validité associées permettent de diagnostiquer immédiatement une incohérence des objets ou une erreur de calcul et de ne pas propager cette erreur par la création de points d'intersection surnuméraires, eux mêmes générateurs d'arêtes de longueur nulle, entités qu'il faut ensuite isoler et éliminer.

De plus leur fréquence d'apparition est en pratique très faible car deux facteurs doivent être réunis :

- Un cas "limite" et donc une forte probabilité de présence de points d'intersection situés à proximité des contours des faces.
- Des erreurs répétées dans les résultats produits par la procédure de localisation alors qu'elle fait appel à une méthode fiable et éprouvée.

Schématiquement, l'étape de création et d'insertion des points d'intersection peut être formulée comme suit :

Pour toute Arête de ObjetA

```

{
  Pour toute Face de ObjetB
  {
    Point = Intersection_Droite_Plan( AreteA , FaceB )
    Si Point existe
    {
      situation = Localiser( Point, ArêteA, FaceB, &Extrémité,&Arête, &Sommet )

      Selon Situation
      {
        Sur_Intérieur      : Insérer( Point, ArêteA, ObjetA )
                           Ajouter_Liste(Point, FaceB )
        Sur_Contour        : Insérer( Point, ArêteA, Objet A )
                           Insérer( Point, Arête .ObjetB )
        Sur_Sommet         :
          Si Sommet est une Intersection
          { /* Situation anormale */
            Si Sommet ∉ à Listes Intersection faces connexes ArêteA
            {
              Objet incohérent
            }
          }
          Sinon
          {
            Insérer( Point, ArêteA, ObjetA )
            Remplacer( Point, Sommet, ObjetB )
          }
        Extrémité_Intérieur :
          Si extrémité est une Intersection
          { /* Situation anormale */
            Si Sommet ∉ Liste Intersections FaceB
            {
              Si Sommet ∉ Contours de FaceB
              {
                Objet incohérent
              }
              Sinon
              {
                Erreur de localisation
              }
            }
          }
          Sinon
          {
            Remplacer( Point, ArêteA, ObjetA)
          }
        }
      }
    }
  }
}

```

```
        Ajouter_Liste(Point, FaceB)
    }
    Les autres cas ne sont pas énumérés car leurs traitements
    sont similaires à ceux ci.
}
}
}
```

Le traitement des arêtes de l'objet B avec les faces de l'objet A est également similaire dans sa structure, mais les rôles des objets ne peuvent pas être simplement permutés car les cas d'incohérence potentielle sont plus nombreux.

Enfin, quand une arête est divisée, la recherche des intersections avec les faces non encore balayées par la boucle est poursuivie avec la première arête résultant de la division, tandis que la seconde, ajoutée en fin de la liste des arêtes de l'objet, sera traitée ultérieurement.

5. Construction des arêtes nouvelles

Nous disposons maintenant de tous les points d'intersection des frontières des objets à combiner. Nous devons donc créer les arêtes qui appartiendront au résultat de l'opération mais qui n'existent pas dans les solides initiaux.

Les principes régissant la construction de ces arêtes sont simples et ont déjà été énoncés lorsque nous avons analysé l'algorithme "Ligne de section" :

- Une arête nouvelle lie toujours deux points d'intersection
- Une arête nouvelle est située sur la frontière des deux objets, et, de ce fait, est obligatoirement contenue dans une face de chacun d'eux.

L'application de ces règles élémentaires à des faces de complexité quelconque nous a conduit à définir la procédure de construction suivante :

- Rechercher chaque couple formé d'une face de l'objet A et d'une face de l'objet B susceptible de générer des arêtes. Deux faces répondant à ce critère doivent :

- * Posséder au moins deux points d'intersection communs, ceux ci pouvant être insérés dans les contours ou situés à l'intérieur des faces.

- * Ne pas être coplanaires.

- Ordonner ces points communs sur la droite d'intersection des plans supports des deux faces, droite dont la direction est donnée par le produit vectoriel de leurs normales. Les normales sont, par convention, toujours dirigées vers l'intérieur des objets et donneront donc une direction constante.

- Vérifier que chaque arête pouvant être construite entre deux points consécutifs sur la droite est bien à l'intérieur des deux faces.

L'exclusion des faces coplanaires de la première étape a deux justifications :

- Leurs intersections ne sont pas des arêtes mais des surfaces polygonales (Fig 3.6)

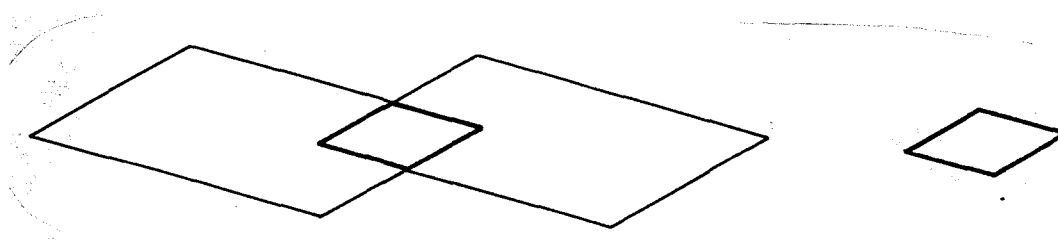


Figure 3.6 : Intersection de deux faces coplanaires

- Les arêtes délimitant ces surfaces seront créés par les faces connexes (Fig 3.7)



Figure 3.7 : Création des arêtes nécessaires par les faces adjacentes

Le classement des points sur la droite est nécessaire car les faces concaves peuvent posséder plus de deux points d'intersections qu'il faut alors relier correctement (Fig 3.8).

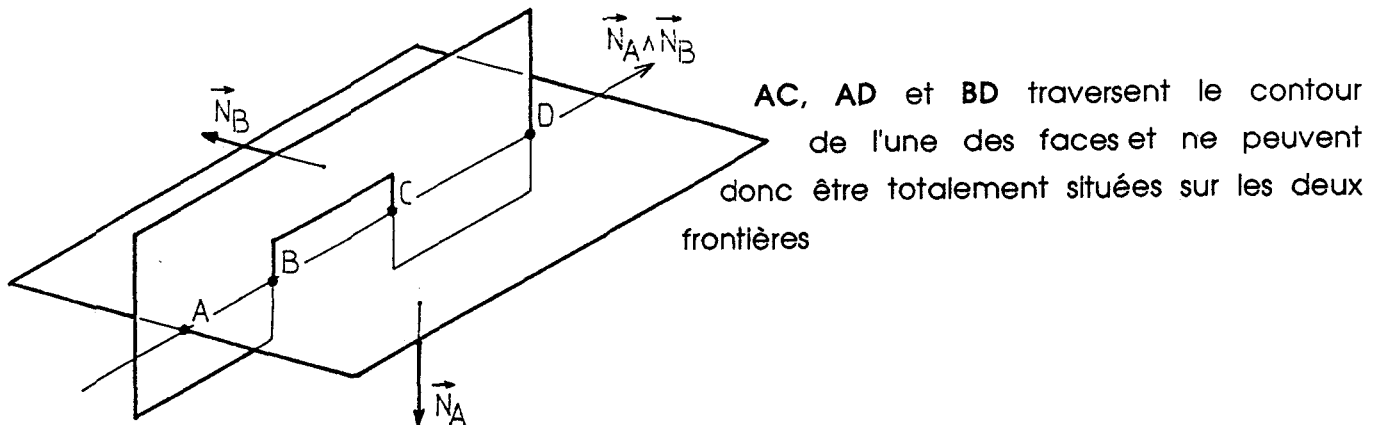


Figure 3.8 : Classement des points d'intersection de deux faces complexes

La vérification d'appartenance des arêtes ainsi générées est la seule méthode qui garantisse la création des arêtes nécessaires et d'elles seules. Par exemple, le procédé qui exploite la parité du rang des points sur la droite en construisant les arêtes dont l'origine est un point de rang impair (Fig 3.9a) [BRU 80] n'est valable que dans les cas d'intersections franches des deux faces (Fig 3.9 b,c,d).

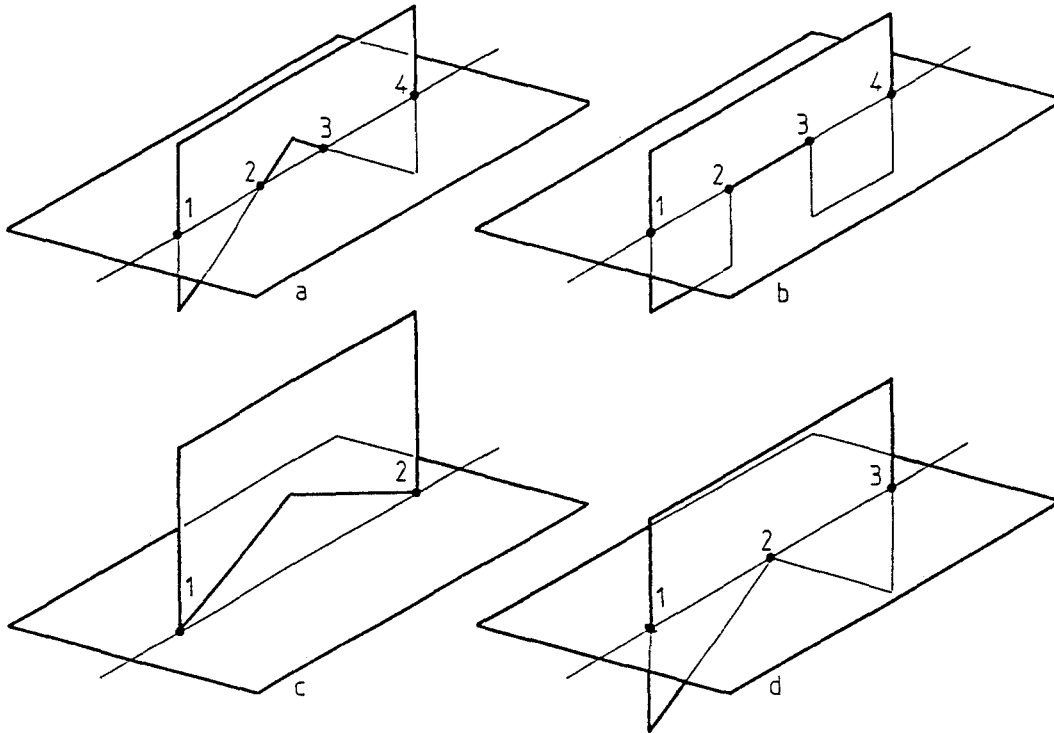
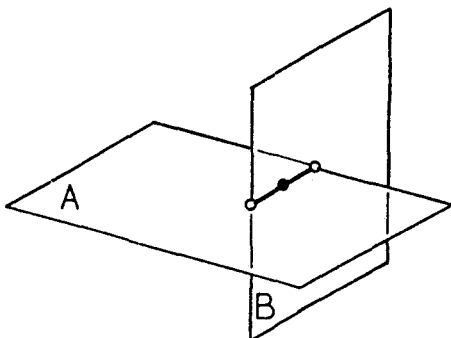


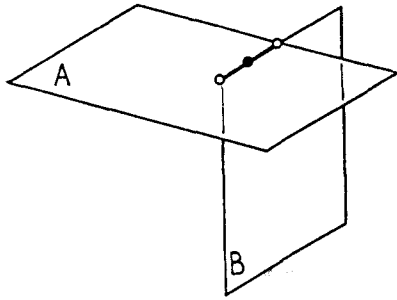
Figure 3.9 : Différents cas d'intersection de deux faces

Cette vérification est effectuée en calculant la situation du point milieu de l'arête à tester vis à vis des deux faces, l'algorithme développé pour la localisation d'un point d'intersection étant évidemment réutilisé. De plus, les informations complémentaires qu'il fournit permettent de déterminer le caractère de nouveauté de l'arête créée.

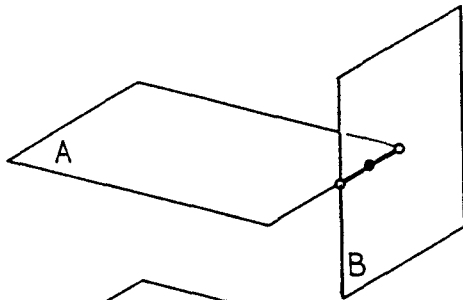
Ce caractère, essentiel car il est utilisé par la suite pour choisir le chemin correct dans le graphe employé pour construire les faces de l'objet résultant de la combinaison, peut prendre quatre valeurs :



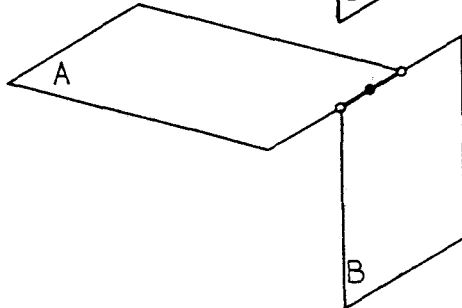
Totalement nouvelle car elle n'existe dans aucun des objets. Son point milieu est situé à l'intérieur des deux faces.



Nouvelle pour l'objet A et ancienne pour l'objet B. Son point milieu est à l'intérieur de la face de A et sur le contour de la face de B.



Ancienne pour l'objet A et nouvelle pour l'objet B, son point milieu est donc sur le contour de la face de A et à l'intérieur de la face de B.



Ancienne pour les deux objets, son point milieu est donc sur le contour des deux faces.

La situation du point milieu suffit à déterminer le type d'une arête "nouvelle" car, avec la méthode de construction utilisée, elle ne peut traverser le contour d'une face et est donc totalement à l'intérieur de cette face ou sur son contour.

Dans ce dernier cas, les points d'intersection ayant été insérés dans les objets, l'arête construite doit déjà exister dans la topologie de l'objet concerné. Comme il s'agit bien d'une arête d'intersection, l'arête construite remplacera alors l'arête existante ce qui implique la modifications des deux faces auxquelles elle appartient. L'insertion des arêtes a le même objectif que l'opération correspondante portant sur les points d'intersections : éviter ou supprimer la duplication d'entités. Deux situations peuvent alors se présenter selon que l'arête construite remplace une arête d'un seul des objets, type Nouvelle A ou B, ou des deux objets, type Ancienne. Dans ce dernier cas, l'insertion est systématiquement effectuée, en tenant compte du sens de parcours de l'arête remplacée, dans les quatre faces concernées (Fig 3.10).

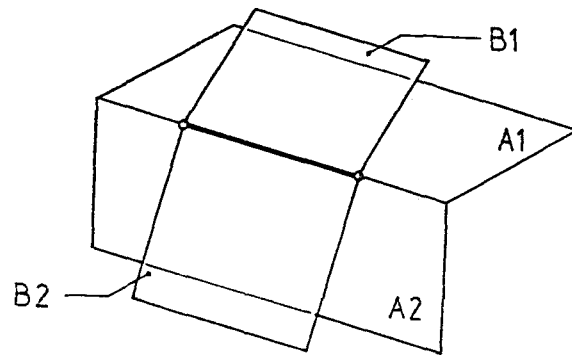


Figure 3.10 : Remplacement d'une arête

La création ultérieure de la même arête par un autre couple de face ,par exemple A2 et B2 dans notre figure, sera alors détectée et ignorée.

Dans le premier cas, une seule arête devrait être construite et substituée à l'arête initiale équivalente. Cependant, deux configurations différentes sont susceptibles de générer ce type d'arête nouvelle. La première ou l'une des faces traverse la frontière de l'autre objet (Fig 3.11a) et devra donc être morcelée, la seconde ou elle se contente de l'effleurer (Fig 3.11b).

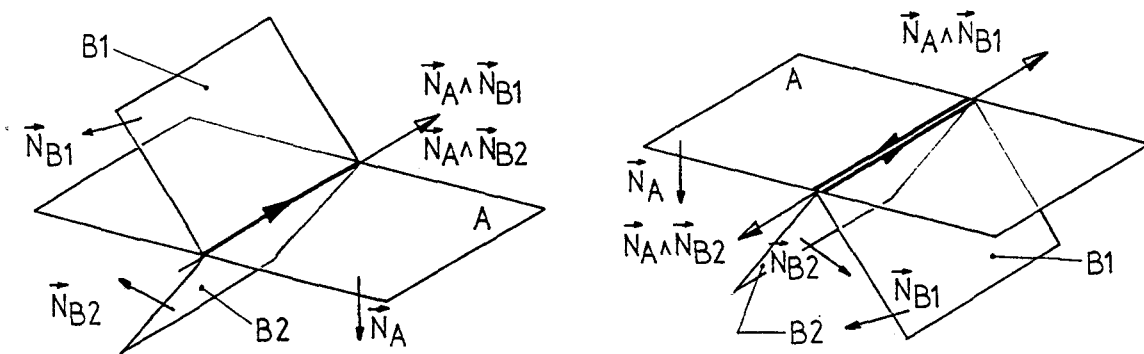


Figure 3.11 : Intersections sur une arête

Afin de distinguer ultérieurement ces deux configurations, l'arête construite sera volontairement dupliquée lorsque les orientations qui lui sont successivement attribuées sont différentes (Fig 3.11b).

Cette phase de création des arêtes peut être schématisé comme suit :

```

Pour toute Face de l'Objet_A
{
  Pour toute Face de l'Objet_B
  {
    Si Face_A et Face_B peuvent engendrer des arêtes
    {
      Calculer direction droite
      Ordonner points d'intersection communs sur droite
      Pour toute Arête joignant deux points consécutifs
      {
        Situer point milieu de l'Arête dans Face_A et Face_B
        Selon situation
        {
          Intérieur_A Intérieur_B : Créer Arête

          Intérieur_A Contour_B : Créer Arête
                                Modifier Objet_B

          Contour_A Intérieur_B : Créer Arête
                                Modifier Objet_A

          Contour_A Contour_B : Créer Arête
                                Modifier Objet_A
                                Modifier Objet_B
        }
      }
    }
  }
}

```

la procédure de modification des objets ayant en outre la fonction de détection et de traitements des cas particuliers énoncés.

6. Construction de la topologie du résultat

La fonction de cette étape est de construire une représentation par les frontières cohérente de la combinaison de deux objets, connaissant leurs frontières, les points d'intersection de ces frontières et les arêtes liant ceux ci.

Son rôle est donc essentiel et son "intelligence topologique" déterminera la capacité de l'algorithme à traiter les cas "limites".

Cependant, une face du résultat de la combinaison étant nécessairement constituée de parties des faces des solides initiaux, ce problème très vaste peut être décomposé en un ensemble de problèmes, plus simples à résoudre, de partitions et de fermetures des contours des faces originelles.

L'éclatement des contours est déjà partiellement réalisé car les points d'intersection ont été insérés lors de leur création. Il reste donc à déterminer quelles parties des contours d'une face sont susceptibles d'appartenir au solide résultat, puis à refermer les contours ouverts obtenus par les arêtes d'intersection.

6.1 Classification des arêtes initiales

La présence d'une partie d'une face dans le résultat de la combinaison dépend de sa situation vis à vis de l'autre objet (à l'intérieur, à l'extérieur, dessus) et du type de la combinaison effectuée.

Une face pouvant être scindée en plusieurs parties de situations différentes et seul ses contours étant réellement connus, c'est en calculant la situation des arêtes dont ils sont formés que nous parviendrons à effectuer cette partition.

Après l'insertion des points d'intersection, une arête d'un objet ne peut plus traverser la frontière de l'autre. Sa situation, vis à vis de l'objet auquel elle n'appartient pas, est alors déterminée en calculant celle de son point milieu qui ne peut prendre que deux valeurs (Interne, Externe) car les arêtes situées "Sur" ont été remplacées par des arêtes "Nouvelles".

Le procédé employé pour situer un point par rapport à un solide est une extension aux volumes de la méthode utilisée pour localiser un point vis à vis d'un polygone. On dénombrera donc les intersections que possède une demi_droite, dont l'origine est le point à situer et de direction quelconque, avec les faces formant le polyèdre approchant la frontière de l'objet.

Si ce nombre est pair le point est situé à l'extérieur, et s'il est impair le point est à l'intérieur. Si une intersection de la demi_droite a lieu sur une arête ou un sommet, on recommence les calculs en affectant une direction différente à la demi_droite, car, comme dans le plan pour un sommet, ces cas sont ambigus (Fig 3.12) et exigent des calculs complémentaires complexes pour être correctement traités.

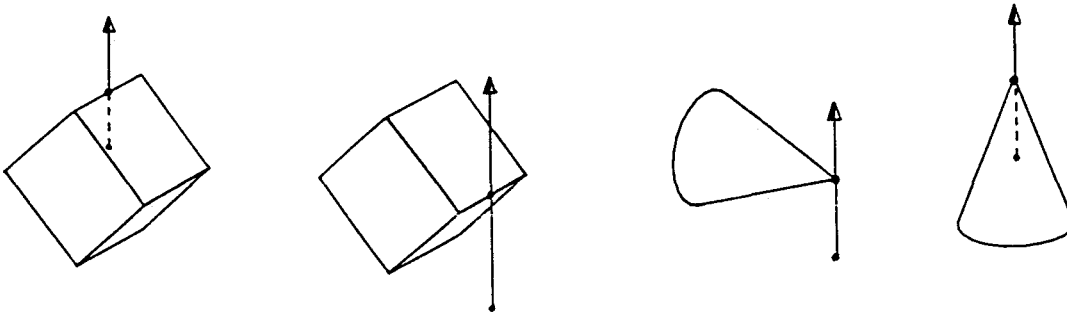


Figure 3.12 : Intersections ambiguës

Ce calcul ne doit pas nécessairement être fait pour chaque arête d'un objet car les connectivités existantes dans une représentation par les frontières permettent de déduire la situation des arêtes voisines à celle que l'on vient de localiser. Ainsi, la position d'une arête sera propagée à toutes celles qui partagent avec elle un sommet de l'objet original car un tel sommet ne peut entraîner un changement de situation. Par contre, la position des arêtes qui partagent un point d'intersection avec une arête déjà située devra être calculée, car il n'indique pas obligatoirement un changement d'état (Fig 3.13).

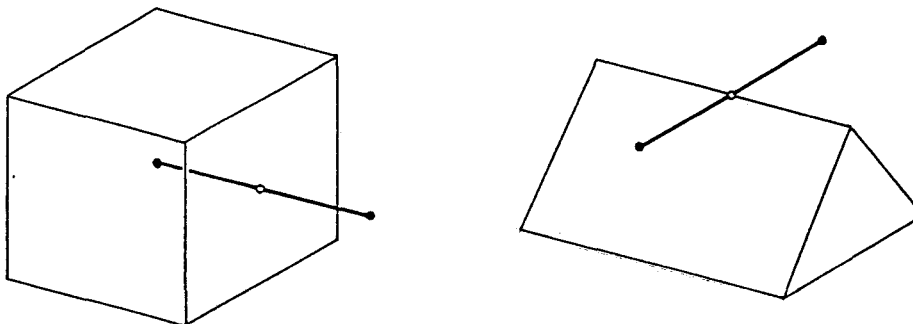


Figure 3.13 : Changement d'état à un point d'intersection

Malgré cette restriction, la diminution du nombre des calculs de localisation réellement effectués reste substantielle, particulièrement si les objets possèdent un grand nombre de faces. Ainsi, dans le cas de la figure 3.14a trois calculs de localisation sont fait au lieu des 78 théoriquement nécessaire, et, dans celui de la figure 3.14b, quatre seulement alors que chaque sphère possède 240 arêtes.

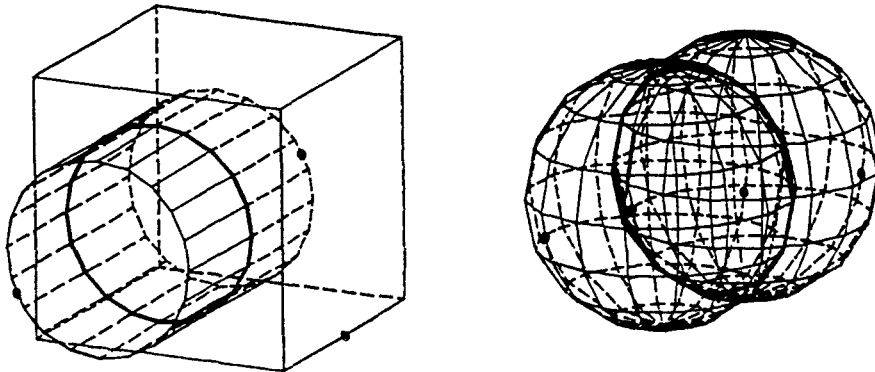


Figure 3.14 : Exploitation des connectivités des faces

L'algorithme calculant la situation des arêtes est alors le suivant :

```
Pour toute Arête de Objet
{
  Si Arête non située alors
  {
    Localiser Arête
    Propager Situation
  }
}
```

6.2 Fermeture des contours

Les propriétés des points et des arêtes calculées dans les étapes précédentes permettent de discerner deux catégories de faces correspondant à des traitements spécifiques :

- les faces intactes qui ne possèdent pas de point d'intersection et dont toutes les arêtes ont donc la même situation,
- les faces modifiées.

Une face intacte n'a pas à être reconstruite, car, si elle appartient au résultat de la combinaison, c'est sous sa forme initiale. La construction d'un graphe est alors inutile et la décision de conserver cette face est prise en fonction de l'objet auquel elle appartient, de sa situation vis à vis de l'autre objet et du type de l'opération effectuée. Les différents cas possibles sont indiqués dans le tableau suivant qui est naturellement analogue à ceux utilisés par les deux algorithmes dont nous avons fait la présentation.

Opération	Objet A	Objet B
Fusion	Externe	Externe
Différence	Externe	Interne
Commun	Interne	Interne

Tableau 3.2 : Critères de sélection des faces intactes

Notons que la situation d'une face est déterminée par celle, déjà calculée, de l'une des ses arêtes. Enfin, le sens de description d'une face de l'objet B conservée dans une opération de soustraction doit être inversé car l'intérieur de cet objet est l'extérieur du résultat.

Les faces modifiées, dont les contours doivent être reconstruits, sont scindées en deux sous_ensembles :

- Les faces "simples" pour lesquelles un graphe sera créé par face.
- Les faces coplanaires dont certaines doivent être associées pour construire un seul graphe.

La notion de faces coplanaires est alors différente de la notion mathématique équivalente car nous attribuerons cette propriété à deux faces si elles sont contenues dans le même

plan et si elles sont susceptibles de produire une face unique de l'objet final.

Pour vérifier cette deuxième condition, leur intersection devra être un segment ou un polygone, ce qui implique qu'elles doivent partager au moins deux points d'intersections et une arête les liant (Fig 3.15).

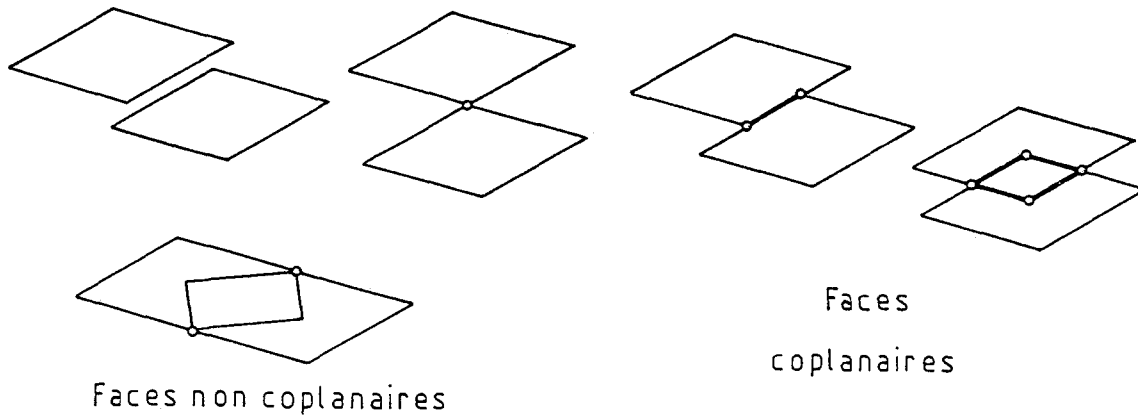


Figure 3.15 : Conditions de coplanéité

Cette propriété n'est pas limitée à deux faces et peut s'appliquer à tout un groupe de faces qui remplissent deux à deux les conditions de coplanéité que nous avons définies (Fig 3.16).

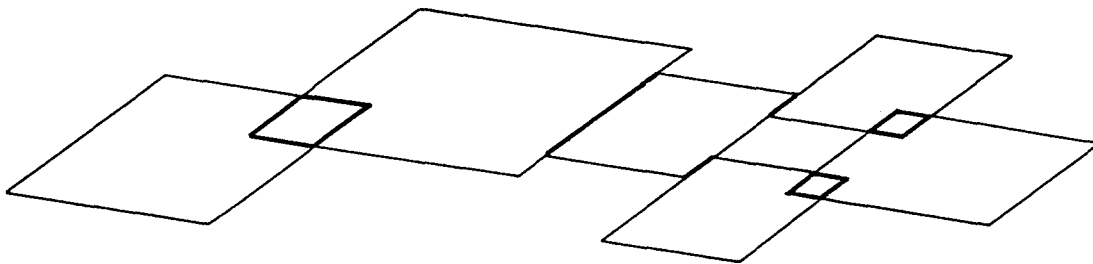


Figure 3.16 : Groupe de faces coplanaires

Bien que ce ne soit pas le cas général, nous aborderons directement le traitement des groupes de faces coplanaires. En effet, comme nous le montrerons par la suite, il doit chronologiquement être effectué avant celui des faces simplement modifiées car il permet d'éliminer certaines arêtes

nouvelles qui ne peuvent appartenir au résultat et qui sont susceptible de générer des contours incorrects.

6.2.1 Faces coplanaires

La construction d'une ou plusieurs faces du résultat à partir d'un groupe de faces coplanaires des solides initiaux se fait par une recherche de cycles, qui matérialiseront des contours, dans un graphe préalablement créé à l'aide des entités connues ou calculées. Ce graphe est constitué de deux éléments fondamentaux:

- Le noeud, homologue d'un sommet de l'une des faces.
- L'arc, joignant deux noeuds, image d'une arête.

Les noeuds sont typés selon l'origine et la situation des sommets qu'ils représentent:

- Noeud "Initial" si le sommet correspondant appartenait à l'un des solides à combiner.
- Noeud "Contour" s'il s'agit d'un point d'intersection sur le contour de l'une des faces.
- Noeud "Interne" dans le cas d'un point d'intersection situé à l'intérieur de l'une des faces.

Les arcs d'un graphe sont typés et orientés. Leur type, comme celui d'un noeud, est fonction de la provenance de l'arête dont il est l'image. Nous distinguons ainsi cinq types d'arcs:

- Initial si il est l'image d'une arête, divisée ou non, de l'un des solides originels
- Nouveau
- Nouveau A
- Nouveau B qui correspondent aux types des arêtes nouvelles
- Ancien

L'orientation d'un arc dépend de son type et de l'opération booléenne effectuée. Un arc "Initial" conserve généralement l'orientation définie par le sens de parcours de l'arête qu'il

représente, sauf si il s'agit d'un arc associé à une arête de l'objet B dans une opération de soustraction. En effet, l'intérieur de cet objet étant nécessairement l'extérieur du résultat de la combinaison, il faut donc inverser le sens de description de ses faces pour diriger leurs normales vers l'intérieur du futur objet. Pour cela, les arcs associés aux arêtes de cet objet sont introduit dans le graphe avec une orientation inverse au sens de parcours de l'arête.

Toutes les arêtes Initiales d'une face qui ne peuvent appartenir au solide final ne sont pas introduites dans le graphe. Le choix est fait en fonction de leur objet d'origine, de leur situation vis à vis de l'autre solide et du type de l'opération booléenne selon des critères identiques à ceux employés pour la sélection des faces intactes.

Les arêtes réellement nouvelles sont toujours introduites dans le graphe, mais l'orientation qui leur a été donnée, si elle est constante, est également arbitraire. Ainsi, cette orientation est toujours correcte si l'arête permet de refermer la partie externe d'une face de l'objet A ou la partie interne d'une face de l'objet B (Fig 3.17).

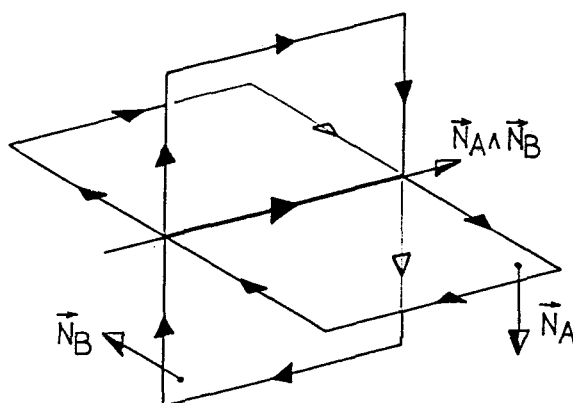


Figure 3.17 : Orientation Initiale des arêtes nouvelles

Le type de l'opération booléenne effectuée permet de définir, pour chaque objet, qu'elle doit être la situation des parties à conserver des faces, il fixera également l'orientation des arcs représentant des arêtes nouvelles :

Opération	Partie gardée	Orientation arc	Partie gardée	Orientation arc
Fusion	Externe =>	Initiale	Externe =>	Inverse
Différence	Externe =>	Initiale	Interne =>	Inverse *
Commun	Interne =>	Inverse	Interne =>	Initiale

Tableau 3.3 : Orientation des arcs de type "Nouveau"

* Dans ce cas particulier, le sens de description de la face est inversé, il est donc obligatoire de procéder à la même opération sur les arêtes "Nouvelles".

Les arêtes de type Nouvelle A, Nouvelle B et Ancienne, déjà introduites dans les contours des faces et extrêmement fréquentes dans les cas de faces coplanaires, suivent les mêmes règles d'orientation que les arêtes initiales de la face dans laquelle elles apparaissent.

Cependant, une attention particulière doit être portée sur ces arêtes car elles n'appartiennent pas toujours au résultat de la combinaison [FLA 87] (Fig 3.18) :

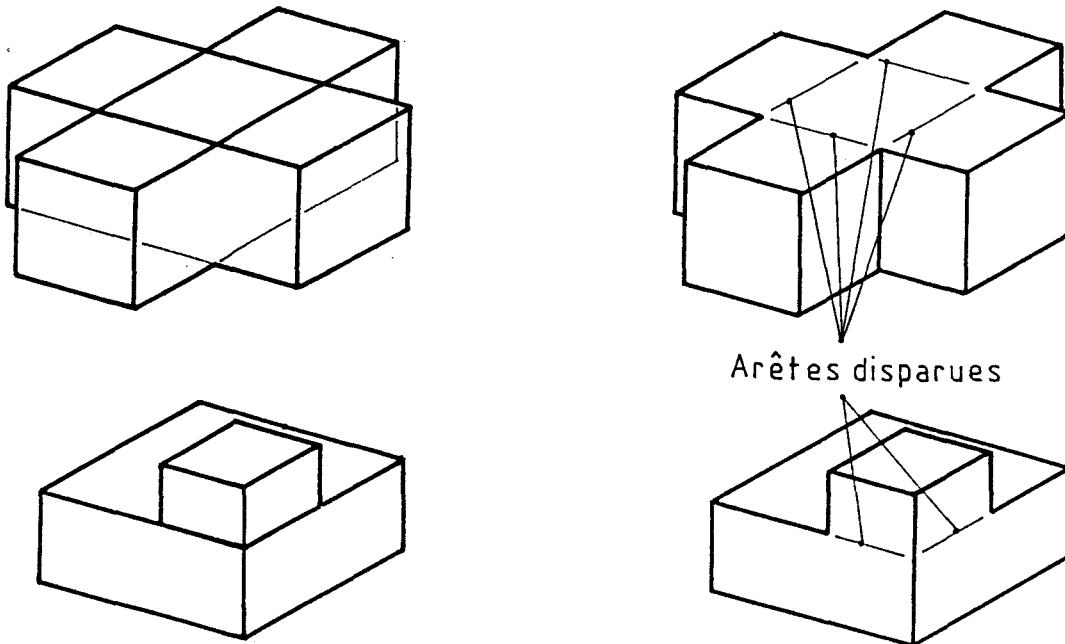


Figure 3.18 : Appartenance des arêtes au résultat de la combinaison

Elles ne doivent donc pas être placées dans le graphe en cours de construction et elles seront, en outre, retirées de la liste des arêtes d'intersection afin d'éviter qu'elles soient malencontreusement utilisées ultérieurement.

Dans le cas des faces coplanaires, un traitement spécifique de ces arêtes particulières était indispensable car, pouvant être présentes dans deux faces, elles auraient alors été introduites deux fois dans le graphe et parfois avec des orientations différentes.

Le sélection de ces arêtes ne peut être faite simplement selon des critères de type ou d'origine car leur appartenance au solide résultant de la combinaison dépend de la position relative des deux objets à traiter et de l'opérateur booléen appliqué (Fig 3.19)

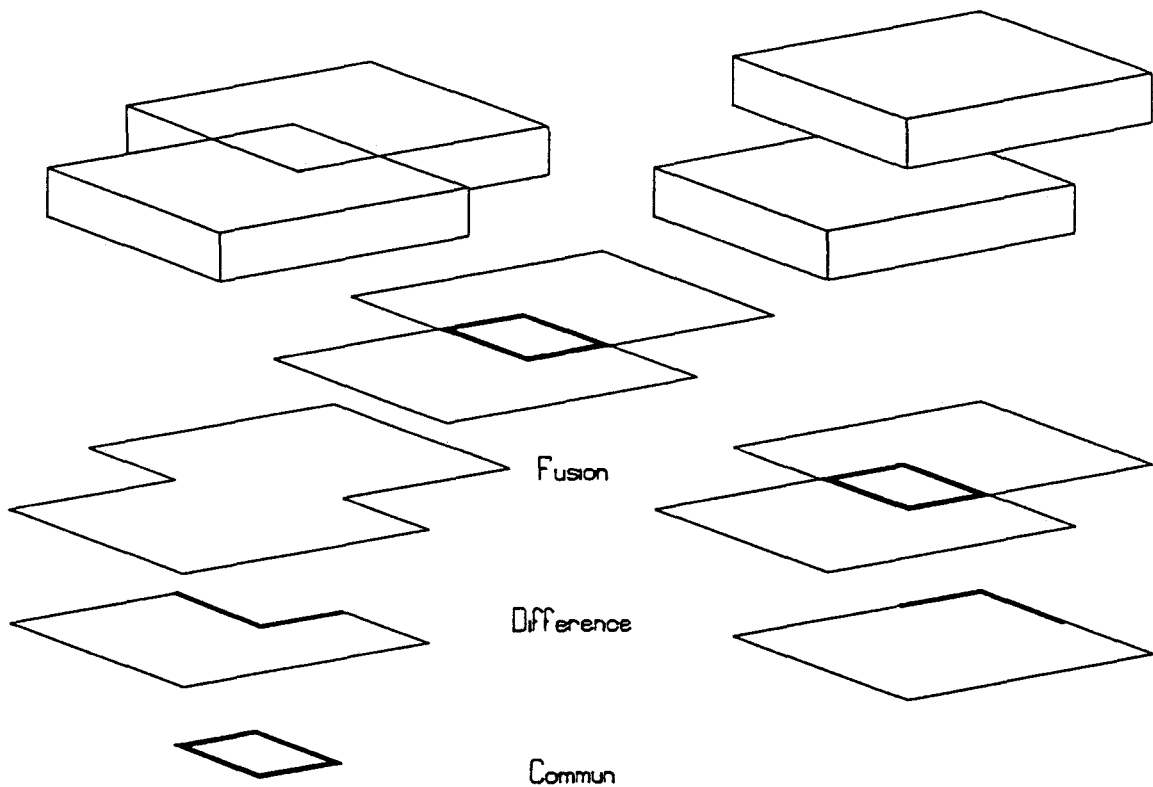
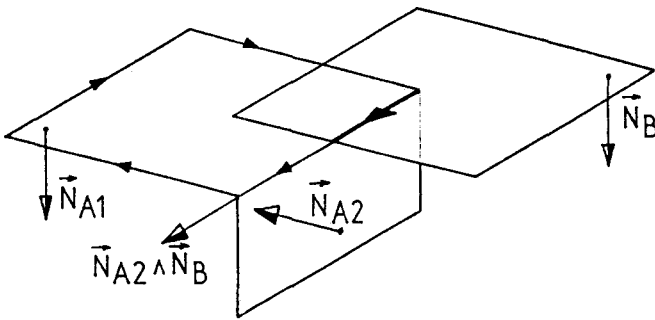


Figure 3.19 : Influence de la position des objets combinés

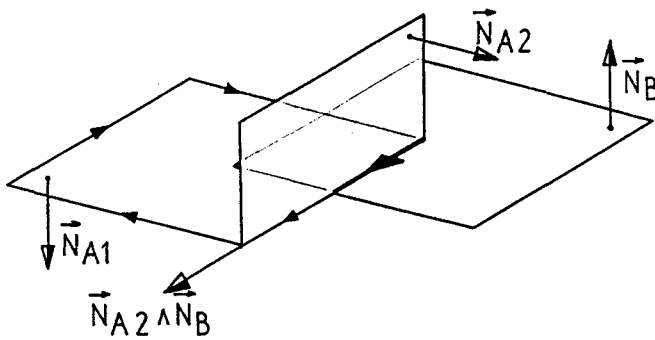
La position relative des objets est étroitement liée à la notion de "matière" qui est caractérisée par le sens de description du contour d'une face duquel découle la direction de sa normale. Cette normale est également présente dans

l'orientation initiale d'une arête nouvelle, combinée toutefois à celle de la seconde face génératrice. L'ensemble de ces facteurs est reflété par le sens d'emploi, vis à vis de son orientation initiale, d'une arête de type Nouvelle A ou Nouvelle B dans les contours des faces où elle a été insérée. C'est donc ce sens de parcours qui fournira la donnée supplémentaire nécessaire au choix.

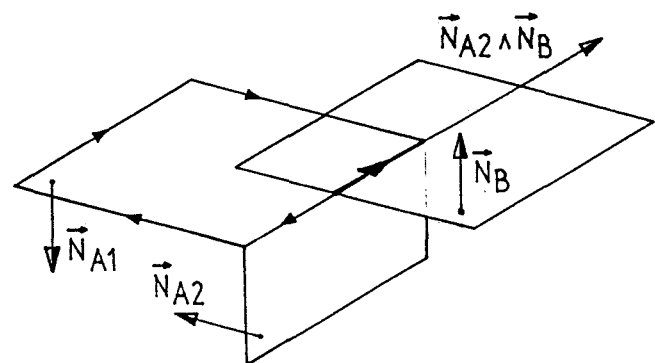
Une arête ne pouvant être parcourue de plus de deux manières, nous serons confrontés à deux situations distinctes, correspondant toutefois à quatre configurations des faces impliquées :



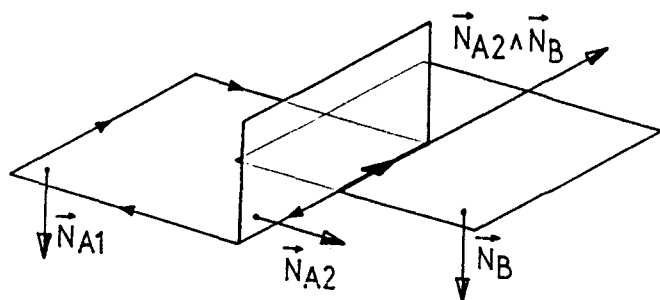
Dans ces deux configurations, l'orientation attribuée à l'arête est correcte quand elle est insérée dans le contour de la face coplanaire.



Selon la convention d'orientation retenue, la face génératrice, qui ne peut être la face coplanaire, est donc située, à proximité de cette arête, à l'intérieur du second objet. Nous considérerons alors que cette arête possède également cette qualité.



Dans ce cas, configurations inverses des deux précédentes, le même raisonnement conduit à une conclusion opposée.



L'arête est donc considérée comme étant à l'extérieur de l'objet B.

Un raisonnement similaire est également tenu lorsque l'arête est de type NouvelleA et appartient donc à l'objet B, mais les situations déduites doivent être inversées car, lorsque une arête "nouvelle" est parcourue selon son sens de création dans le contour de la face génératrice, cette face se trouve partiellement à l'extérieur de l'objet A.

Ce procédé revient à attribuer à ces arêtes une situation Interne ou Externe de même signification que celle calculée pour les arêtes initiales des solides. Il est alors logique de leur appliquer les mêmes critères de sélection, d'où le tableau suivant :

Type de l'arête :		Nouvelle B		Nouvelle A	
Type opération	Sens de parcours de l'arête sur la face coplanaire				
	Initial => Interne	Inverse => Externe	Initial => Externe	Inverse => Interne	
Fusion	Rejetée	Gardée	Gardée	Rejetée	
Différence	Rejetée	Gardée	Gardée	Rejetée	
Commun	Gardée	Rejetée	Rejetée	Gardée	

Tableau 3.4 : Sélection des arêtes Nouvelle A et Nouvelle B

Quand on traitera un groupe de faces coplanaires, ces critères de sélection seront appliqués à toutes les arêtes de ces types associées à un couple de faces.

Les arêtes d'intersection de types Ancienne partagées par deux faces coplanaires apparaissent dans quatre situations différentes :

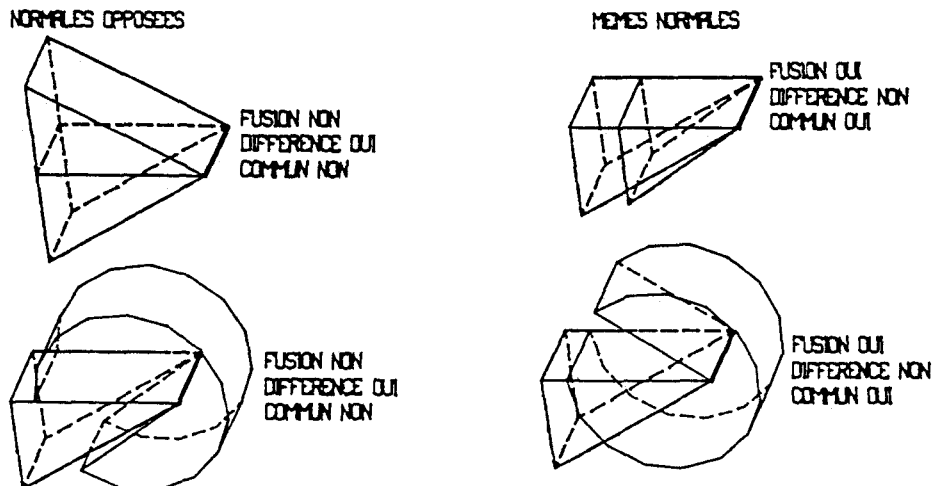
- Faces superposées de même normales
- Faces superposées de normales opposées
- Faces juxtaposées de même normales
- Faces juxtaposées de normales opposées

Ces configurations sont identifiées d'une part par les directions des normales et d'autre part par le sens de parcours de l'arête commune sur chacun des contours des deux faces coplanaires, comme indiqué dans le tableau suivant :

	Directions des normales	Sens de parcours de l'arête
Faces superposées	identiques	même sens
	opposées	sens opposé
Faces juxtaposées	identiques	sens opposé
	opposées	même sens

Tableau 3.5 : Identification des diverses configurations de faces coplanaires partageant une arête

Dans le cas des faces superposées, le type de l'opération booléenne et de la configuration suffisent à déterminer si l'arête "Ancienne" doit être introduite dans le graphe construit pour les deux faces coplanaires (Fig 3.20) :



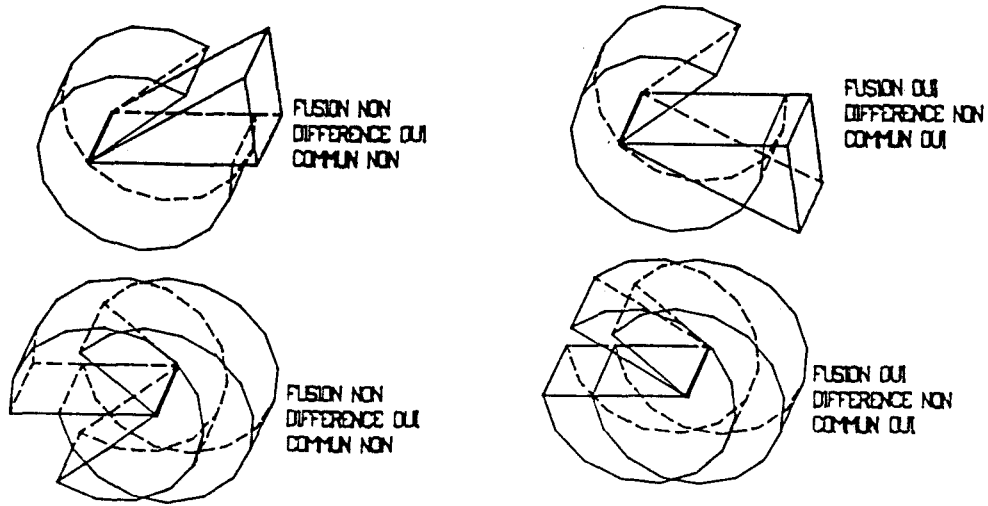


Figure 3.20 : Cas des faces superposées

Par contre, dans le cas des faces juxtaposées, il faut également tenir compte de la situation de l'arête partagée sur une portion convexe ou concave des objets (Fig 3.21) :

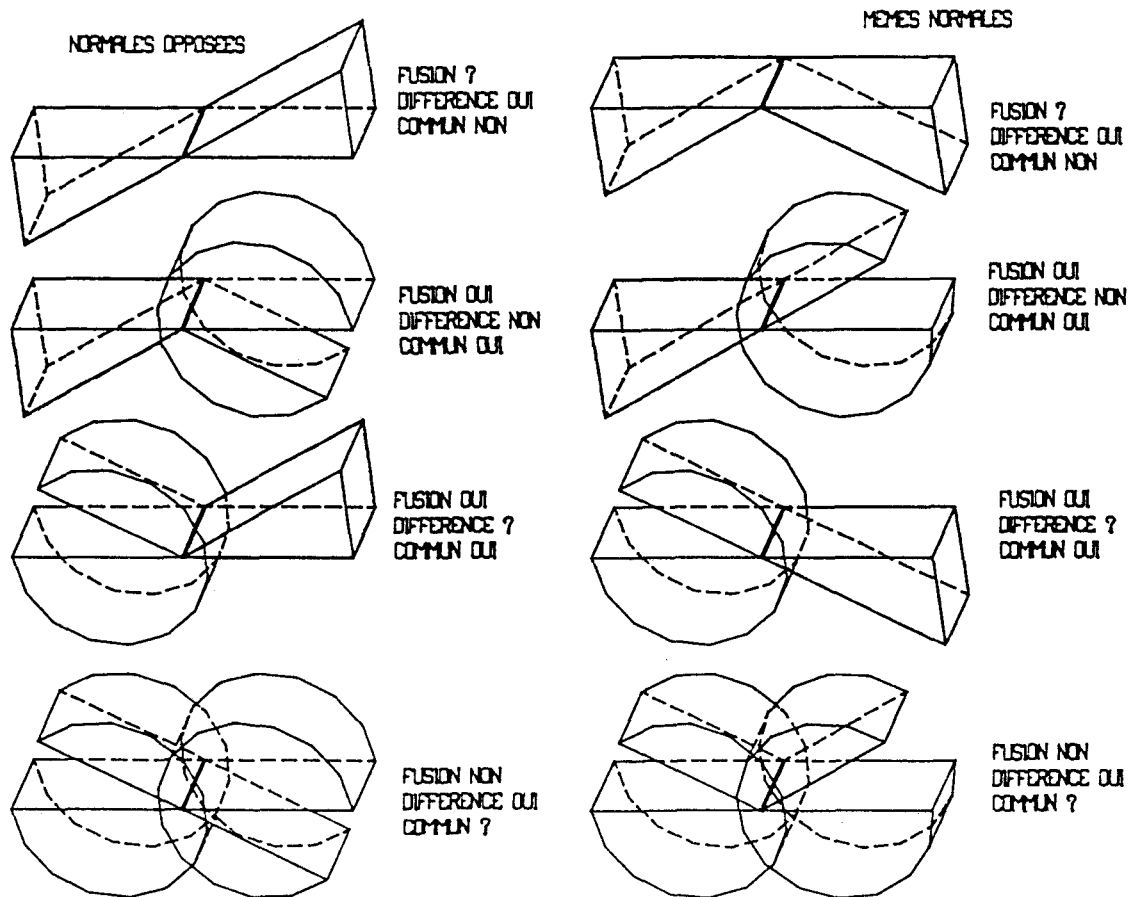


Figure 3.21 : cas des faces juxtaposées

La présence d'une concavité ou d'une convexité est liée à la notion de "matière" ou d'intérieur d'un objet. Les normales des quatre faces qui partagent cette arête devraient donc nous permettre de distinguer ces différentes configurations.

En effet, lorsque l'arête "Ancienne" est située sur des parties de même convexité sur les deux objets, le plan contenant les deux faces coplanaires sépare les deux faces connexes (Fig 3.22), information qui est reflétée par leurs normales.

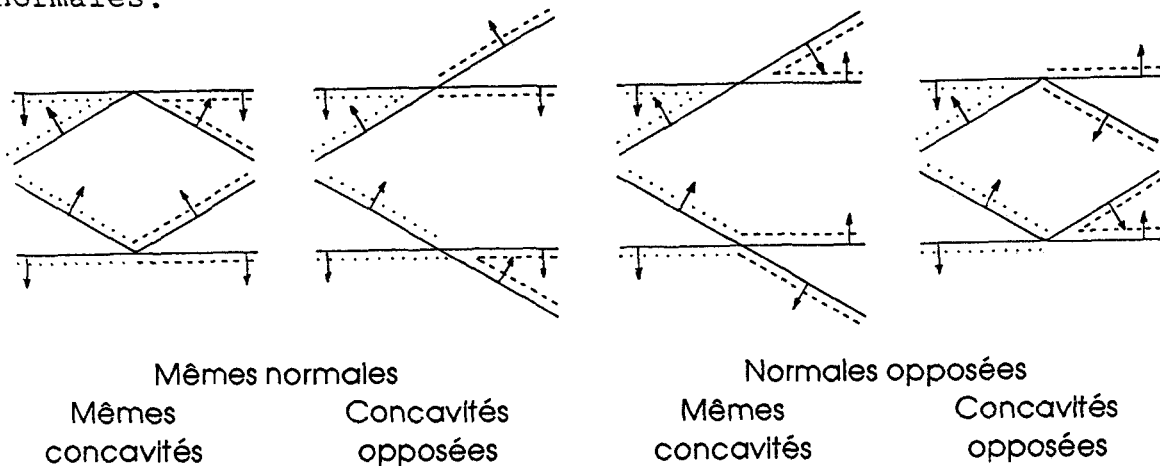


Figure 3.22

6.2.2 Recherche des cycles

Un cycle d'un graphe est une liste ordonnée d'arcs, deux arcs successifs de cette liste partageant un noeud, l'origine du premier arc et l'extrémité du dernier étant confondue. La recherche d'un cycle peut alors se ramener à la recherche du successeur d'un arc donné parmi ceux qui ont pour noeud d'origine son extrémité.

On distingue alors quatre cas possibles selon la nature de l'arc et celle de son noeud d'arrivée :

- **Arc Initial et noeud Initial** : Ce type de noeud ne peut être l'origine que d'un seul arc car il représente un sommet situé sur une partie non modifiée du contour d'une face originelle. Le successeur de l'arc courant étant unique, il est donc inutile de procéder à un choix.

- **Arc Initial et noeud Contour** : Cette situation représente un point d'intersection situé sur le contour d'une face et donc l'intersection de celle ci avec la frontière de l'autre objet. Ce type de noeud peut, dans les cas "limites" et donc des faces coplanaires, être l'origine de plusieurs arcs. On choisira alors en priorité un arc image d'une arête d'intersection nouvelle pour cette face car elle est commune aux deux frontières et réalise la coupure de la face en partie interne et externe. C'est donc le caractère de nouveauté de cet arc qui sera le critère de choix et, selon l'origine de l'arc initial courant, les priorités suivantes sont attribués aux successeurs potentiels :


Arc courant	Initial A	Initial B
 <p>Priorité décroissante</p>	<p>Nouveau NouveauA NouveauB Ancien InitialB InitialA</p>	<p>Nouveau NouveauB NouveauA Ancien InitialA InitialB</p>

Tableau 3.6 : Priorité affectées aux arcs

Les deux dernières priorités, affectées à des types d'arcs images d'arêtes des solides initiaux, sont nécessaires pour permettre le raccordement de parties non modifiées de faces distinctes, éventualité qui peut se présenter dans les cas de faces coplanaires.

- **Arc non Initial et noeud Interne** : Ce cas reflète le parcours d'une partie de cycle formée d'éléments qui représentent des entités nouvelles pour la face reconstruite. Un tel noeud est nécessairement l'origine d'arcs images d'arêtes d'intersection et le choix du successeur est alors fait selon les mêmes critères que ceux utilisés dans le cas précédent.

- **Arc non Initial et noeud Contour** : Ce type d'arc, représentant une arête d'intersection terminée par un point sur le contour d'une face, indique généralement la fin de la découpe d'une face et le retour à une partie de son contour originel. Le choix du successeur privilégiera donc un arc image d'une arête appartenant aux solides initiaux. C'est en fait la configuration symétrique du cas "Arc Initial noeud contour" et l'ordre des priorités allouées aux arcs est donc inversé.

L'arc initiateur d'un cycle est choisi, lorsqu'il en existe, parmi les arcs représentant des arêtes appartenant obligatoirement au solide final. Il s'agit des arcs de types Initial et Nouveau car ces types correspondent à des arêtes dont la situation est non ambiguë.

Un cycle est complet quand le noeud d'arrivée de l'arc courant est identique au noeud de départ de l'arc initiateur. Un arc ne peut être présent dans plus d'un cycle d'un même graphe car cela entraînerait la présence de deux contours situés dans le même plan et partageant une arête.

La recherche des cycles dans un graphe est poursuivie tant qu'il possède plus de deux arcs libres et, quand un cycle ne peut être fermé, les arcs utilisés par celui-ci sont restitués au graphe. La recherche reprend alors avec un autre arc initiateur, mais il ne doit pas rester d'arc de type Initial ou Nouveau inemployés dans le graphe après la construction de tous les cycles possibles.

6.2.3 Faces simples

Les éléments constituant le graphe construit pour chaque face modifiée sont naturellement semblables à ceux définis pour les groupes de faces coplanaires.

Les critères d'introduction et d'orientation des arcs dans le graphe sont également identiques, mais les procédures de tri applicables aux arêtes de types NouvelleA, NouvelleB et Ancienne n'ont pas à être employées car ces arêtes ne peuvent être partagées par deux faces. De plus, elles ont été retirées

de la liste des arêtes d'intersection lorsque cela était nécessaire.

Le seul cas exigeant un traitement différent d'une simple sélection sur le type est celui où deux arêtes d'intersection distinctes relient les mêmes sommets mais avec des orientations opposées. Il s'agit nécessairement d'arêtes de types NouvelleA ou NouvelleB volontairement dupliquées lors de leur création afin de permettre la distinction entre une intersection vraie d'une face et un cas "limite" d'effleurement. Ces deux arêtes ne sont pas introduites dans le graphe en cours de construction car elles pourraient donner naissance à deux faces coplanaires de même orientation et partageant une arête (Fig 3.23).

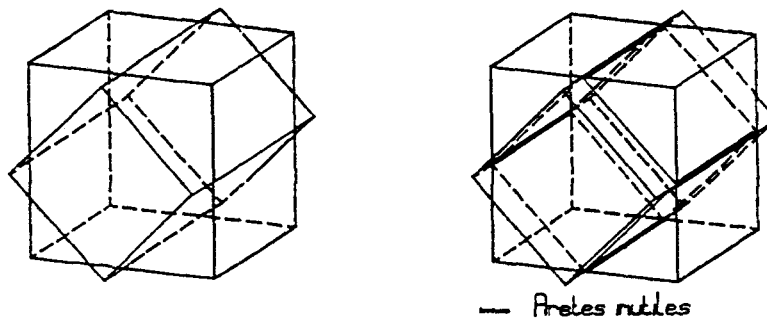


Figure 3.23 : Faces coplanaires partageant une arête

Cette situation peut également apparaître lors du traitement d'un groupe de faces coplanaires, la même opération est alors effectuée.

La recherche des cycles suit les principes énoncés dans les cas de coplanéité. Toutefois, le choix du successeur d'un arc est plus restreint car tous les arcs d'un graphe sont les images d'arêtes appartenant à un seul des deux objets combinés.

6.2.4 Type des cycles d'un graphe

Un graphe peut contenir un ou plusieurs cycles qui seront transformés en contours de faces du résultat de l'opération booléenne. Il faut donc déterminer si un cycle est un contour externe ou interne puis établir les relations d'appartenance entre ces contours car un contour interne ne peut exister seul.

Dans la plupart des cas, le type d'un cycle est aisément déterminé. En effet, l'orientation donnée aux arcs lors de leur introduction dans un graphe fait que le contour déduit d'un cycle possède son sens de description définitif. Par conséquent, il est possible de calculer, pour chaque cycle d'un même graphe, la normale du contour associé et de déterminer leur type en comparant cette normale à celle de la face dont ils sont tous issus.

Cependant, le graphe construit pour un ensemble de faces coplanaires peut éventuellement rassembler deux groupes de faces de normales opposées. Selon la normale choisie comme référence, ce qui est cycle interne pour l'une sera cycle externe pour l'autre et réciproquement.

Même dans ce cas particulier un cycle valide ne peut décrire un contour formé d'arêtes initiales appartenant à deux faces de normales opposées ou d'arêtes d'intersection situées à l'intérieur de faces possédant la même propriété car cela impliquerait la construction d'un contour dont la matière changerait de côté selon la partie considérée. (Fig 3.24)

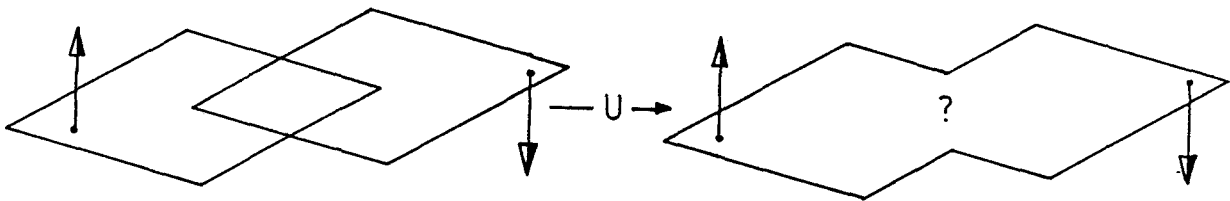


Figure 3.24 : Combinaison de faces de normales opposées.

Chaque arc d'un cycle est l'image d'une arête. En fonction de la normale de la face d'origine de l'un de leurs arcs, on pourra former deux listes de cycles, chaque liste possédant alors une normale de référence unique qui permettra d'attribuer un type à tous ses cycles.

Lorsque le type de chaque contour est défini, il faut établir les relations devant exister entre contours internes et externes pour convertir ensuite un ou plusieurs contours en une face de l'objet résultat. Le simple test d'intériorité d'un point situé sur un contour interne vis à vis d'un contour externe est une condition nécessaire mais non suffisante si l'imbrication des contours est multiple (Fig 3.25).

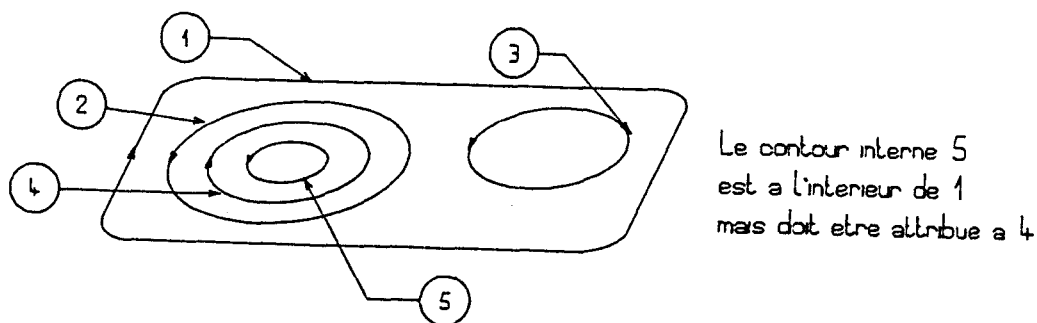


Figure 3.25 : Imbrication multiple de contours

Cette situation particulière fait apparaître une relation d'appartenance à plusieurs niveaux dans laquelle tous les contours externes ne jouent pas le même rôle. Or, notre structure de modélisation par les frontières est telle qu'une face ne peut posséder qu'un seul contour externe et donc un seul niveau d'imbrication.

Il faut donc distinguer parmi les différents niveaux d'imbrication le niveau le plus bas. Pour un contour interne situé à l'intérieur de plusieurs contours externes, c'est nécessairement la relation existant entre celui ci et le contour externe de plus petite taille car c'est le seul qui puisse être contenu par tous les autres.

On classe donc les contours externes et internes par ordre de surfaces croissante puis on effectue le test d'intériorité pour chacun des contours internes dont l'aire est inférieure à celle du premier contour externe.

Lorsque toutes les relations sont établies, le premier contour externe et ses éventuels contours internes forment une face du solide final. Ils sont alors retirés des listes de

contours et ce processus est réitéré avec le contour externe suivant. Naturellement, s'il n'existe pas de contours internes ou un seul contour externe, il est inutile de procéder aux test d'intériorité car la face peut être créé directement.

Quand toutes les faces intactes et modifiées des solides initiaux ont été traitée, on dispose normalement de la représentation par les frontières du résultat de la combinaison. Toutefois, la topologie construite référence toujours les tables des arêtes et des sommets des deux arguments de l'opération ainsi que celles contenant les points et les arêtes d'intersection. De nouvelles tables, propres à cet objet, sont alors créés en recopiant les éléments originels et calculés effectivement utilisés, ce qui implique la modification des faces et des arêtes de manière à référencer ces nouvelles tables.

7. Vérification du résultat

Nous avons évoqué à plusieurs reprises, lors de la description des différentes étapes de notre algorithme, la nécessité de procédures de vérification de la modélisation d'un solide.

Si les propriétés d'une modélisation correcte d'un objet du monde réel sont parfaitement définies, la satisfaction de ces contraintes par un modèle est plus délicate à établir car une représentation par les frontières comporte deux niveaux distinct d'information :

- Le niveau topologique concerne toutes les relations d'appartenance et de connectivité existant entre les faces, les arêtes et les sommets.
- Le niveau géométrique relatif à la position dans l'espace des sommets de l'objet et par conséquent des plans contenant les faces.

Ces deux niveaux possèdent une certaine indépendance, exploitée, par exemple, pour appliquer une transformation géométrique à un solide, mais déterminent conjointement sa

validité. Ainsi, la modification de la position d'un seul des sommets d'un objet correct peut rendre celui ci totalement incohérent (Fig 3.26).

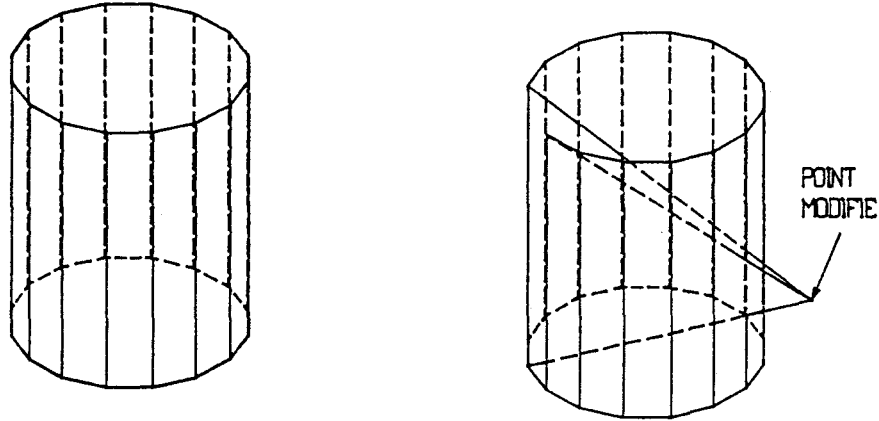


Figure 3.26 : Objet devenu incohérent

La vérification globale du solide étant en pratique irréalisable, la hiérarchie présente dans une représentation par les frontières nous a conduit à contrôler la validité du niveau topologique seul, puis, celle ci établie, la cohérence du niveau géométrique.

7.1 Vérification de la topologie

Deux méthodes permettent la vérification de la topologie d'un objet. La première repose sur les propriétés mathématiques d'une surface polyédrique fermée, propriétés qui ont permis de définir une relation liant le nombre de ses constituants. Cette relation, connue sous le nom de loi d'EULER-POINCARRE, s'exprime comme suit [TAN 87]:

$$F - A + S - T = 2 (O - P) \text{ où}$$

F est le nombre de faces
A le nombre d'arêtes
S le nombre de sommets
T le nombre total de contours internes des faces

O le nombre de parties distinctes de l'objet
P le nombre de "poignée" ou trou traversant effectivement l'objet

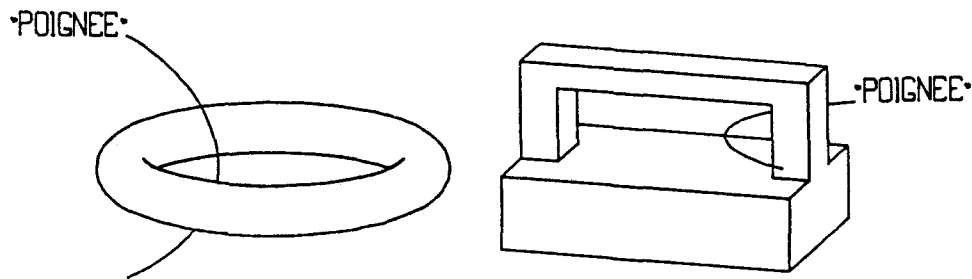


Figure 3.27 : "Poignée" d'un objet

Cette dernière valeur, contrairement aux autres qui peuvent directement être extraites du modèle ou aisément calculées, n'est pas évaluable automatiquement. On doit donc soit faire appel à l'utilisateur soit se contenter de vérifier que cette valeur est entière en introduisant les données connues dans la relation.

La seconde méthode exploite l'orientation constante des contours des faces qui, alliée à la fermeture de la surface polyédrique, fait qu'une arête doit nécessairement être parcourue une fois dans chaque sens (Loi de MOEBIUS) [PRE 82]. Simple à mettre en oeuvre, cette loi, si elle n'est pas satisfaite, assure de l'incohérence de la topologie d'un objet mais la réciproque n'est pas vraie.

7.2 Vérification de la géométrie

La méthode utilisée construit, pour chaque face, un vecteur colinéaire à sa normale et dont la norme est égale à l'aire de la face. Tous les points du ou des contours de la face interviennent dans ce calcul permettant ainsi de construire un vecteur même si la face n'est pas plane ou si l'un de ses contours se coupe lui même.

Si l'objet possède une géométrie cohérente avec sa topologie, la somme de ces vecteurs doit alors être nulle [MEZ 86].

8. Propagation des liens

Lors de la construction de la topologie de l'objet résultat d'une combinaison, nous pouvons, pour maintenir le lien établi entre une face et la surface exacte quelle approche, être confronté à trois cas :

- La face traitée appartient intégralement au solide final. Elle lui est donc ajoutée avec son lien.
- La face est modifiée. Le lien qu'elle possédait sera propagée à toutes les faces qu'elle a permis de construire.
- La face est modifiée et appartient à groupe de faces coplanaires. Dans ce cas le problème est plus complexe car une face du résultat peut naître de la combinaison de plusieurs faces des objets initiaux et devra donc posséder plusieurs liens. En fonction de l'origine des arêtes formant les contours de chaque face construite dans ce cas, on peut déterminer quelles faces des objets combinés ont été associées et propager ainsi les liens nécessaires.

9. Conclusion

Lors de la définition de chaque étape de notre algorithme nous avons accordé une très grande attention au problèmes posés par les cas "limites".

Identifiées et exploitées par les méthodes que nous proposons, les configurations spécifiques qui apparaissent dans ces cas particuliers devraient donc être correctement traitées par l'algorithme proposé.

Le chapitre suivant, consacré à l'implémentation et à l'évaluation de ces méthodes permettra de situer leurs capacités réelles de résolution des problèmes posés par les combinaisons booléennes de deux objets modélisés par leur frontière.

Chapitre 4

Mise en oeuvre

Evaluations et perspectives

1. Introduction

Dans le chapitre précédent, nous avons fait une présentation des différentes étapes de notre algorithme d'un point de vue purement méthodologique.

L'implémentation directe de ces méthodes conduirait inévitablement à des temps de traitements excessifs si l'on combinait des objets complexes en nombres de faces et d'arêtes.

Dans la première partie de ce chapitre, nous étudierons donc les problèmes posés par la mise en oeuvre de certaines étapes de l'algorithme et nous les modifierons dans le but de réduire leur complexité théorique moyenne et les temps de calculs qui en découlent.

Puis, dans la seconde partie, nous procéderons à une évaluation des performances de l'algorithme pratique .

Enfin, dans la dernière partie, les conclusions tirées de l'analyse des résultats obtenus lors de l'évaluation nous permettront de définir les orientations des développements futurs.

2. Mise en oeuvre

L'analyse de la complexité d'un algorithme à deux objectifs [GAUD89]:

- Déterminer si il est réellement utilisable en estimant son utilisation des ressources de l'ordinateur hôte.
- Permettre sa comparaison avec d'autres algorithmes résolvant différemment le même problème.

L'évaluation de la complexité de chacune des quatre étapes que compte notre algorithme montre que deux d'entre elles possèdent une complexité quadratique:

- Le calcul des points d'intersection qui demande théoriquement la comparaison de chaque arête d'un solide avec toutes les faces de l'autre objet. Sa complexité est donc, en première estimation, de l'ordre de $(A_1 * F_2 + A_2 * F_1)$ ou A_1 , A_2 , F_1 et F_2 sont les nombres d'arêtes et de faces respectifs des deux polyèdres combinés.
- La classification des arêtes qui exige dans le meilleur des cas (objets disjoints) le calcul de la situation d'une arête de chaque objet vis à vis de l'autre solide (complexité de l'ordre de $F_1 + F_2$) et dans le cas le plus défavorable le calcul de la situation de toutes les arêtes des deux solides (complexité alors supérieure à $A_1 * F_2 + A_2 * F_1$, les arêtes ayant été divisées).

Ces deux étapes pénaliseront donc le comportement global de l'algorithme lorsque l'on combinera des objets de tailles importantes, d'autant plus que la majeure partie des calculs nécessaires sont effectués sur des nombres réels.

Or, ces comparaisons systématiques et coûteuses sont souvent inutiles car la probabilité d'intersection d'une demi-droite, et à fortiori d'un segment, avec toutes les faces d'un polyèdre est nulle.

Un procédé simple de localisation des entités dans des espaces disjoints permettrait alors d'éviter leur comparaison effective et donc une réduction généralement importantes des calculs .

2.1 Volumes englobants

Cette localisation spatiale peut être réalisée en plaçant chaque arête, chaque face et chaque objet (Fig 4.1) dans la boîte de volume minimal capable de contenir cette entité [MART88b].

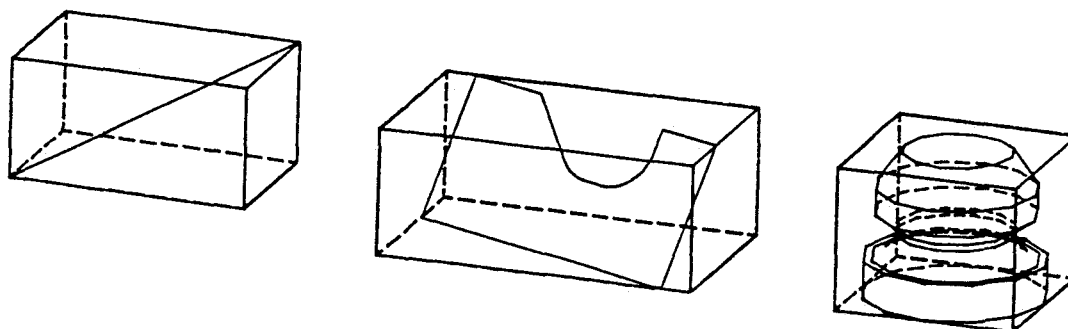


Figure 4.1 : Boîtes englobantes

Avant de procéder aux calculs d'intersections, on s'assurera que les boîtes englobantes possèdent une partie commune et donc que les entités contenues ont une bonne probabilité de se couper.

2.2 Applications

2.2.1 Représentation et comparaison des boîtes englobantes

Une boîte englobante est un parallélépipède rectangle dont les faces sont parallèles aux plans principaux de l'espace.

Une représentation par les frontières d'un tel objet est inutile car il est complètement défini par l'une de ses diagonales.

Deux points suffisent donc à sa modélisation et, de manière à pouvoir les construire et les comparer aisément, on choisit habituellement :

- Le coin inférieur droit antérieur correspondant aux composantes minimales.
- Le coin supérieur gauche postérieur correspondant aux composantes maximales.

Construire la boîte englobante d'une entité revient alors à déterminer les valeurs minimales et maximales des coordonnées des sommets de cette entité.

Enfin, deux boîtes seront disjointes si l'une des composantes maximales de la première est inférieure à la composante minimale correspondante de la seconde ou si l'une des composantes minimales de la première est supérieure à la composante maximale correspondante de la seconde.

2.2.2 Calcul des points d'intersection

Afin d'exploiter au mieux le principe de localisation spatiale, la comparaison de la boîte englobante d'une arête avec celle d'une face n'est pas simplement ajoutée avant la recherche de leur intersection éventuelle.

Il est en effet beaucoup plus intéressant d'établir d'abord la liste des faces de l'objet intersecté dont les boîtes englobantes coupent celle de l'objet intersectant. Ainsi, après la comparaison de sa boîte englobante avec celle de l'objet coupée, chaque arête de l'objet coupant ne devra plus être comparée qu'aux faces présentes dans cette liste.

L'algorithme pratique de calcul et d'insertion des points d'intersection est alors le suivant :

```
Liste_Faces = Etablir_liste( Boîte_A , Objet_B )
pour toute Arête de A
{
  Si Boîte_Arête coupe Boîte_B alors
  {
    Pour toute face appartenant à Liste_Faces
    {
      Si Boîte_Arête coupe Boîte_Face alors
      {
        Point = Intersection_Droite_Plan( Arête_A , Face_B )
      }
    }
  }
}
```

```

    traitement identique à l'algorithme initial
  }
}
}
}

```

2.2.3 Classification des arêtes

La méthode retenue pour calculer la situation d'une arête vis à vis d'un objet consistait à dénombrer les intersections que possédait une demi-droite issue de son point milieu et de direction quelconque avec les faces de cet objet.

Ce procédé rend difficile l'utilisation directe des volumes englobants, une demi_droite ne pouvant être contenue dans une boîte dont les deux coins sont connus.

La procédure de localisation d'une arête est donc modifiée et possède maintenant deux phases :

- On détermine d'abord la situation du point milieu de l'arête vis à vis de la boîte englobant l'objet. Un point pouvant être assimilé à une boîte de volume nul, il suffit alors de comparer ses coordonnées à celles des coins de la boîte.
- S'il est situé à l'extérieur, la localisation de l'arête est immédiate : elle doit nécessairement être située à l'extérieur de l'objet. Dans le cas contraire, on construit un segment partant du point milieu de l'arête, de direction quelconque et dont la longueur est légèrement supérieure à celle de la diagonale de la boîte englobant l'objet. Ce segment possèdera obligatoirement le même nombre d'intersection que la demi-droite de même support et de même direction. Par contre, il peut être inclus dans une boîte finie et permet alors de procéder à la comparaison des boîtes avant de calculer effectivement son intersection avec une face.

3. Evaluation

3.1 Contexte de l'évaluation

L'implémentation de cet algorithme a été réalisée en langage C sur une station de travail DEC/GPXII.

De façon à fournir un élément de référence, il sera comparé au système de modélisation du logiciel de C.F.A.O EUCLID_IS, implanté sur la même machine, qui utilise également une représentation par les frontières et dispose d'opérateurs booléens.

3.2 Complexité pratique de l'algorithme

L'ordre de complexité de ce type d'algorithme est délicat à évaluer, la taille des objets, seule information dont on dispose, n'étant pas le seul paramètre qui doit être pris en compte.

Ainsi, le nombre et le type des points d'intersection font évoluer la taille des objets car ils divisent les arêtes initiales, conditionnent le nombre d'arêtes nouvelles pouvant être construites, influencent l'étape de classification et déterminent enfin la complexité des graphes permettant de construire la topologie du résultat de la combinaison.

De plus, les optimisations apportées rendent cette évaluation plus problématique encore parce qu'elles modifient le nombre des calculs réellement effectués dans des proportions à priori inconnues et difficilement quantifiables.

Un exemple simple permet de mettre cet état de fait en évidence tout en montrant l'intérêt des filtres introduits dans l'algorithme. Ce test calcule la différence de deux sphères de même rayon, initialement disjointes et dont la distance entre leur centre est progressivement réduite.

L'ordre de complexité de toutes ces configurations, si on l'exprime en fonction du nombre de sommets, d'arêtes et de faces des objets impliqués, est rigoureusement identique alors

que les temps de calculs varient sensiblement comme le montre le tableau (Tab 4.1) résumant les mesures effectuées.

Dans tous les cas, la surface polyédrique, approchant les sphères de rayon 10, comportait 114 sommets, 240 arêtes et 128 faces.

Distance centres	Intersections calculées	Arêtes créés	Localisations effectuées	Temps (Seconde)
22	0	0	0	2.27
18	200	12	2	3.46
16	225	12	2	3.93
14	345	20	2	5.07
12	389	20	2	6.28
10	517	16	4	8.52
8	609	20	4	9.32
6	1122	28	4	12.51
4	1509	28	4	13.42
2	2428	28	4	15.86

Tableau 4.1

L'utilisation des boîtes englobantes a été particulièrement bénéfique lors de la phase de calcul des points d'intersection, le nombre des calculs réellement effectués étant toujours largement inférieur au nombre théoriquement nécessaire (61440). Dans une moindre mesure, elle a permis une réduction du nombre de localisations d'arêtes effectivement calculées, nombre qui, à l'exception du premier cas, est normalement de quatre car chaque sphère est scindée en deux parties.

3.3 Résultats

3.3.1 Critères et modèles choisis

Les utilisateurs d'un système de modélisation jugent les outils algorithmiques qui leur sont proposés en fonction [MEZ 86]:

- de leurs capacités à résoudre correctement les problèmes pour lesquels ils sont conçus (aspect qualitatif),
- des temps de traitements nécessaires pour parvenir à un résultat (aspect quantitatif).

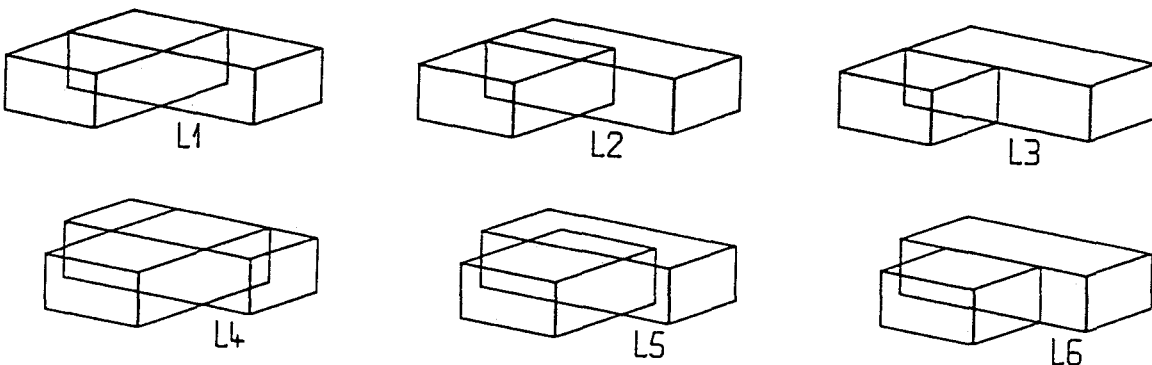
Nous reprendrons donc ces deux critères et, pour cela, trois catégories différentes de modèles ont été employées :

- Des combinaisons d'objets simples qui font apparaître systématiquement les types d'intersections problématiques (faces coplanaires, arêtes ou sommets confondus).
- L'évaluation d'arbres de construction simples mais produisant des objets de grandes tailles.
- L'évaluation d'arbres de construction modélisant des pièces réelles qui, en associant les deux premières catégories de tests, sont susceptibles de faire apparaître d'autres situations d'échecs.

3.3.2 Cas limites

Il n'est pas nécessaire de combiner des objets complexes pour engendrer les types d'intersections les plus difficiles à traiter.

Ainsi, les douze configurations de deux parallélépipèdes rectangles de la figure 4.2 permettent une étude relativement complète, bien qu'elle soit un peu artificielle, des cas de coplanéité de faces associés, pour certains d'entre eux, à des problèmes de sommets ou de parties d'arêtes confondues.



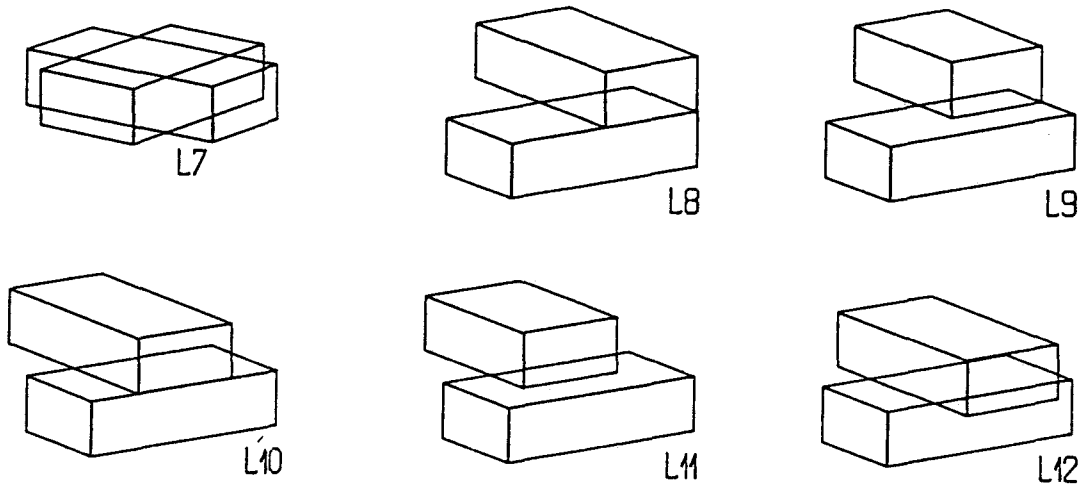


Figure 4.2 : Cas limites

Les quatre objets de la figure 4.3 donnent également naissance à des problèmes d'arêtes ou de sommets confondus, mais sans faire intervenir de faces coplanaires. Toutes ces configurations sont bien différentes car l'approximation polyédrique d'une sphère n'est pas totalement symétrique.

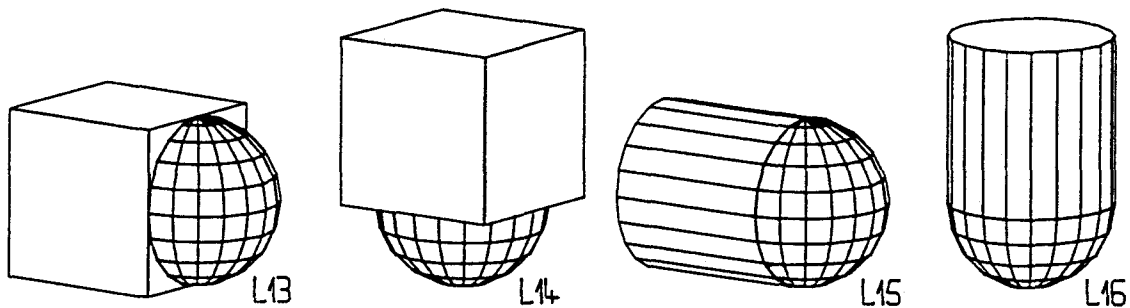


Figure 4.3 : Cas limites sans faces coplanaires

3.3.3 Objets de tailles importantes

Deux méthodes ont été utilisées pour produire simplement des objets de grandes tailles.

La première consiste à percer un objet initialement cubique par des cylindres perpendiculaires à ses faces et uniformément répartis sur celles-ci (Fig 4.4).

La taille de l'un des arguments de chaque opération est donc de plus en plus importante tandis que celle de l'autre, avant

insertion des points d'intersection, est fixe. L'objet final possède 744 sommets, 1356 arêtes et 582 faces avec la polygonalisation adoptée pour les cylindres.

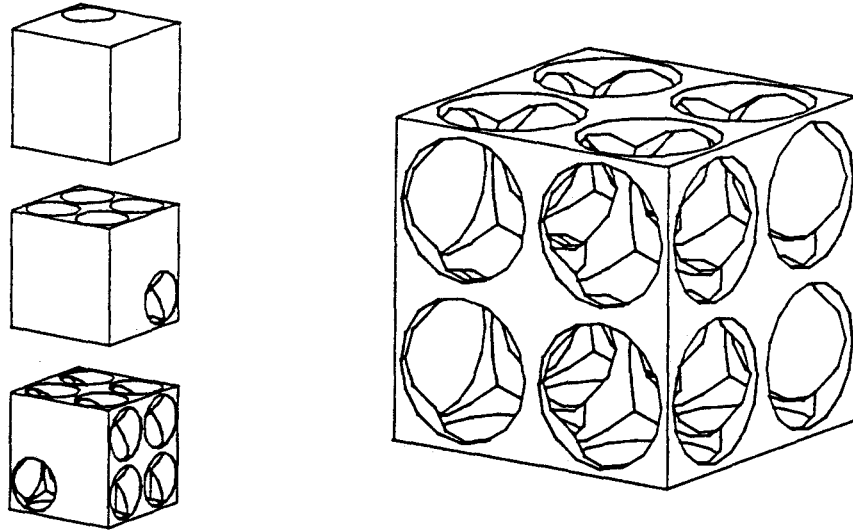


Figure 4.4

La seconde exploite la possibilité offerte par l'arbre de construction d'utiliser un même solide à différentes positions de l'espace (Fig 4.5) de manière à combiner deux objets de tailles importantes et proches. Le résultat comporte 1208 sommets, 2304 arêtes et 1088 faces.

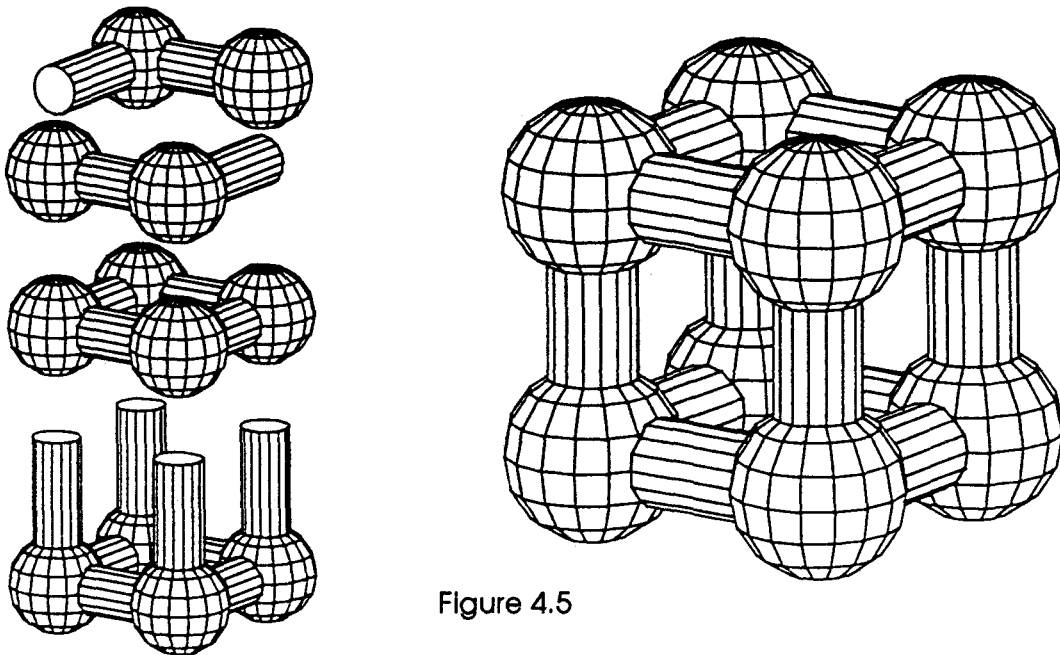


Figure 4.5

3.3.4 Pièces réelles

Les deux pièces de la figure 4.6 ont été modélisées en utilisant exclusivement les primitives solides énumérées lors de la description du modèle de produit proposé au premier chapitre.

Dans ces conditions, la première pièce (R1) exige un arbre de construction comportant 34 opérations booléennes et celui de la seconde pièce (R2) demande l'évaluation de 40 noeuds.

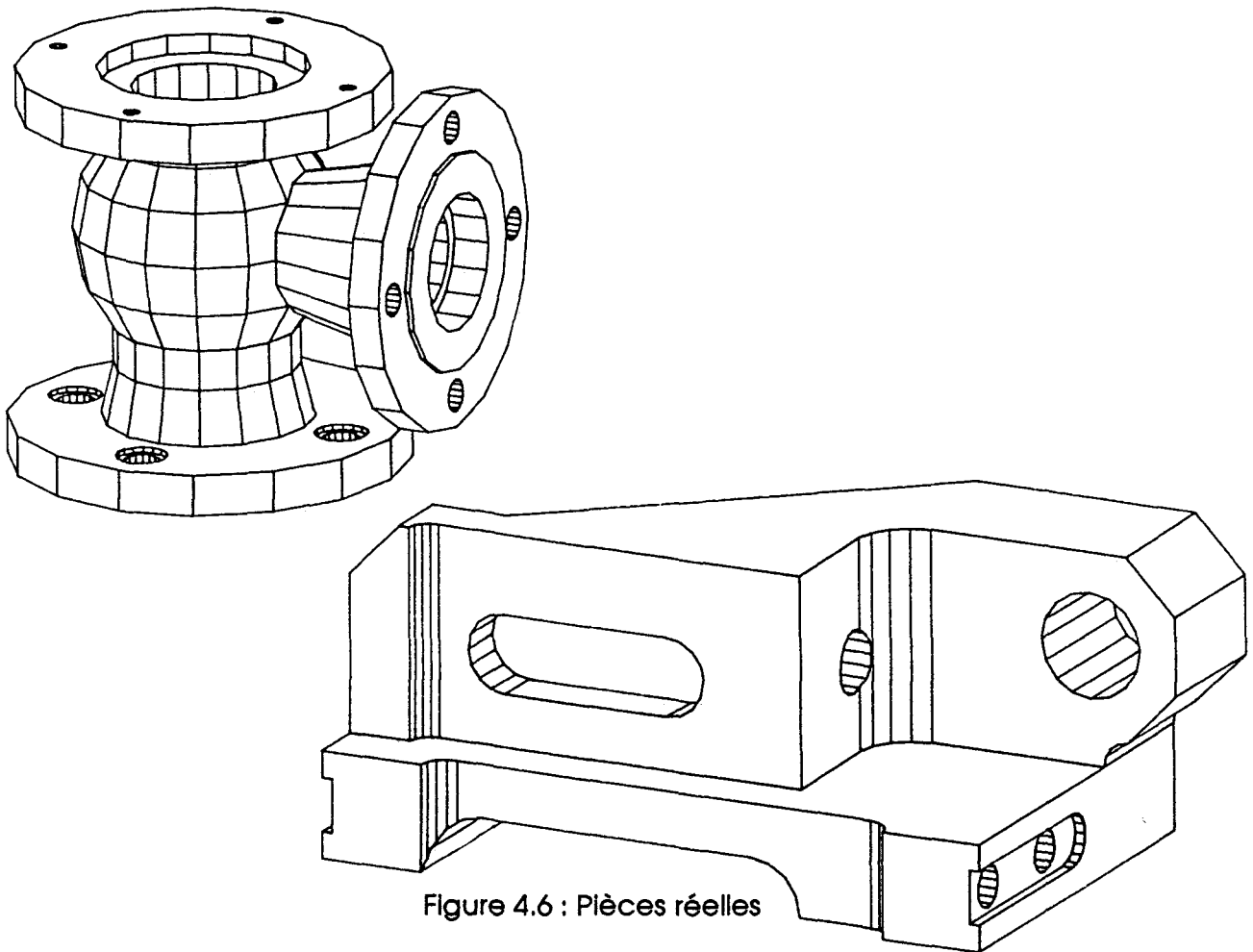


Figure 4.6 : Pièces réelles

3.3.5 Aspect qualitatif

Le tableau 4.2 présente synthétiquement les résultats obtenus avec les trois opérateurs booléens de chaque algorithme lorsqu'on leur soumet les cas "limites". De manière à situer leurs capacités de traitement réelles, seules les combinaisons devant aboutir à un objet valide ont été calculées.

Cas	Fusion		Différence		Commun	
	EUCLID		EUCLID		EUCLID	
L1	+	-	+	-	+	-
L2	+	+	+	+	+	+
L3	+	+		Non calculé		
L4	+	-	+	-	+	-
L5	+	+	+	+	+	+
L6	+	+		Non calculé		
L7	+	-	+	-	+	-
L8	+	+		Non calculé		
L9	+	+		Non calculé		
L10	+	+		Non calculé		
L11	+	+		Non calculé		
L12	+	+		Non calculé		
L13	+	-	+	-	+	+
L14	+	-	+	+	+	+
L15	+	-	+	-	+	-
L16	+	-	+	-	+	-

+ : Traité

- : Non traité

Tableau 4.2

L'attention particulière portée sur ces cas, lors de la définition de notre algorithme, n'a donc pas été vaine car ils ont tous été correctement traités alors que celui d'EUCLID_IS échouait sur certains d'entre eux.

L'interprétation de ces résultats doit cependant être quelque peu nuancée :

- Bien qu'aucun des modèles choisis n'ait provoqué un échec de notre algorithme, il serait illusoire de croire que sa fiabilité est absolue car sa mise en oeuvre nécessite des programmes de tailles importantes, inévitablement imparfaits et dépendants, de surcroît, de jeux subtils d'"epsilon" destinés à compenser l'imprécision des calculs.

Il est alors plus réaliste de conclure que ses limites sont plus éloignées que celles de l'algorithme d'EUCLID_IS mais qu'elles seront certainement atteintes.

- La plupart des combinaisons qui n'ont pas été calculées par EUCLID_IS représentent des objets pouvant être modélisés soit par un arbre de construction équivalent (L1, L2 et L3 aboutissent à des objets géométriquement semblables), soit par l'utilisation de primitives de création par balayage linéaire ou circulaire, permettant ainsi de contourner ces cas. Cependant, des configurations de faces coplanaires similaires peuvent survenir lors de combinaisons d'objets, eux mêmes issus de l'évaluation d'arbres de construction complexes, dans lesquels il sera alors difficile d'isoler et de remplacer les primitives sources de ces problèmes. Enfin, l'opération de fusion d'une sphère et d'un parallélépipède rectangle du cas L15, qui n'a pu être calculée, est plus gênante car il paraît difficile de modéliser différemment cet objet.
- A l'exception de la pièce R2, dont l'arbre de construction comportait une opération sur deux boîtes dans une configuration similaire au cas L4 et qui pouvait donc être évitée, tous les autres objets choisis ont été correctement évalués par les deux algorithmes. Les cas "limites" sont donc généralement plus exigeant avec ce type d'algorithmes que l'utilisation réelle d'un système de modélisation par un opérateur conscient de ses limitations.

3.3.6 Aspect quantitatif

Les histogrammes des figures 4.7, 4.8 et 4.9 présentent comparativement, pour chaque opérations booléennes, les mesures effectuées lors de l'évaluation des arbres de construction modélisant les pièces V1, V2 et R1 par notre algorithme et celui d'EUCLID_IS.

Objet V1

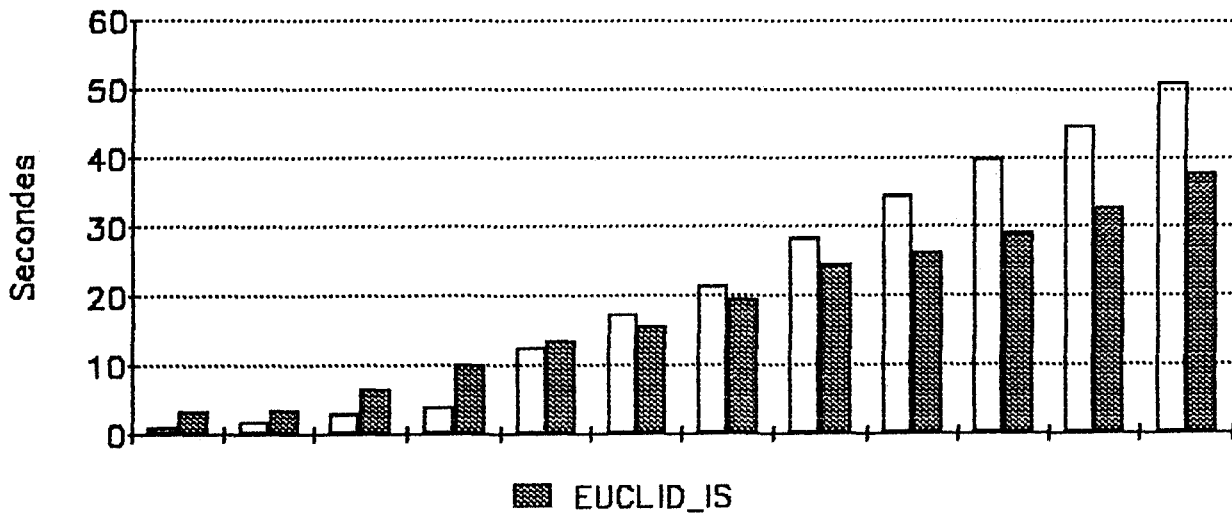


Figure 4.7

Objet V2

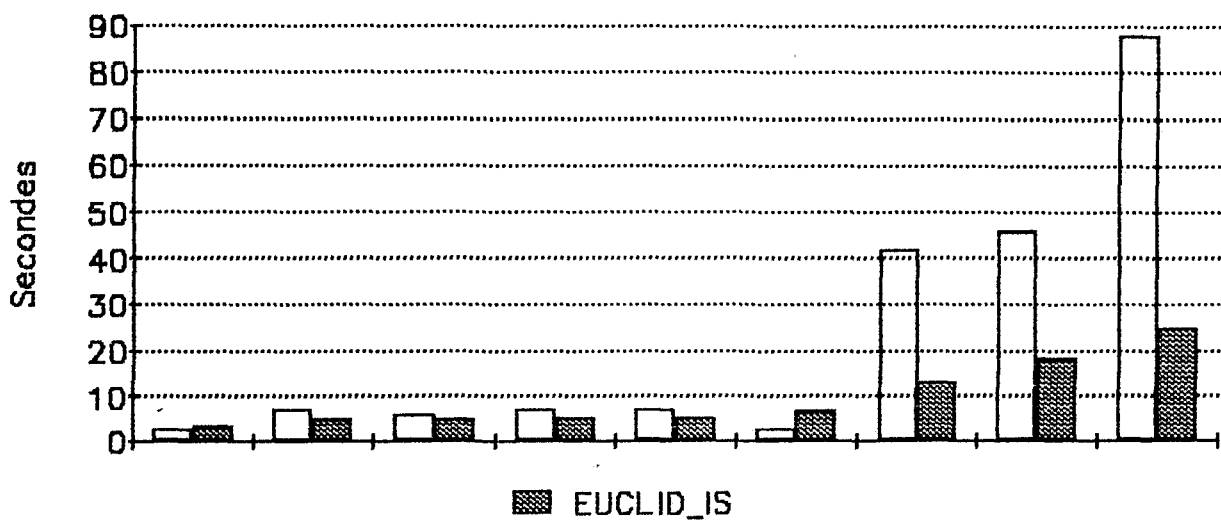


Figure 4.8

L'analyse de ces histogrammes montre une évolution plus favorable des temps de traitements de l'algorithme d'EUCLID_IS lorsque la taille des objets combinés croît.

Bien que le principe exact sur lequel cet algorithme est basé ne soit pas connu, il est cependant possible de formuler trois hypothèses sur les origines de cette progression moins rapide :

- Les deux algorithmes n'effectuent pas les mêmes traitements comme l'indique leurs comportements sensiblement différents face à certains cas (Fig 4.9) et leurs pouvoirs de résolution inégaux précédemment mis en évidence.

Objet R1

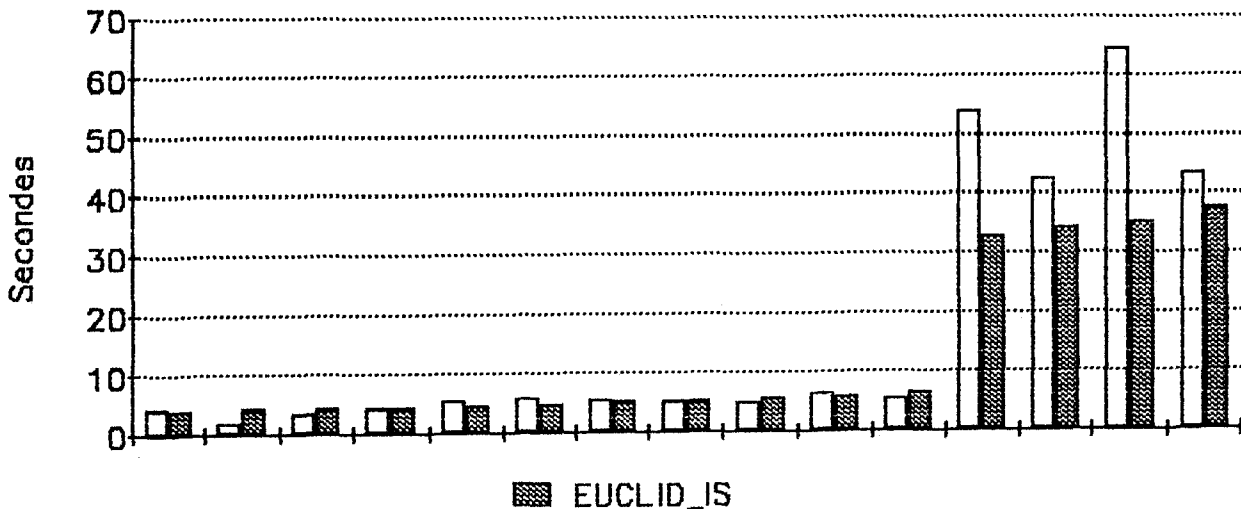


Figure 4.9

- Ecrit en FORTRAN, l'algorithme d'EUCLID_IS conduit probablement ses calculs sur des nombres réels en arithmétique flottante simple précision (codage des nombres réels sur 32 bits). Le langage C, que nous avons utilisé, privilégie lui l'arithmétique flottante double précision (codage sur 64 bits) qui fournit des résultats plus précis mais qui est aussi plus lente (30% environ sur VAX [AVEL89]).
- La représentation particulière des surfaces polyédriques d'EUCLID_IS introduit un niveau supplémentaire dans la classique hiérarchie Objet->Faces->Arêtes->Sommets des modèles frontières car elles sont considérées comme des "Figures" (Association logique) de polygones. Or, ces "Figures", par le principe de codage des entités utilisé, ne peuvent associer plus de 63 éléments. Si l'approximation polyédrique de la frontière de l'objet modélisé comporte plus de 63 faces, elle est alors mémorisée comme une "Figure" de "Figures" de faces. Cette structure, si la répartition des faces dans les différentes "figures" est faite en fonction de leurs connectivités, doit permettre,

associée à un procédé de localisation spatiale par volumes englobants, de limiter le nombre de comparaisons inutiles et coûteuses des constituants des deux solides et donc une réduction des temps de traitements.

Enfin, l'implémentation de notre algorithme a été faite, avant tout, dans le but de valider la démarche adoptée et n'est donc certainement pas optimale en terme de rapidité.

4. Perspectives

Les résultats obtenus démontrent la validité des choix effectués mais font également apparaître une évolution rapide des temps de calculs lorsque la taille des arguments d'une opération augmente.

L'examen de la répartition moyenne de ces temps entre les différentes étapes de l'algorithme (Fig 4.10) confirme ce que l'étude de leurs complexités théoriques avait déjà indiqué : les phases de calculs des points d'intersection et de classification des arêtes sont les plus coûteuses.

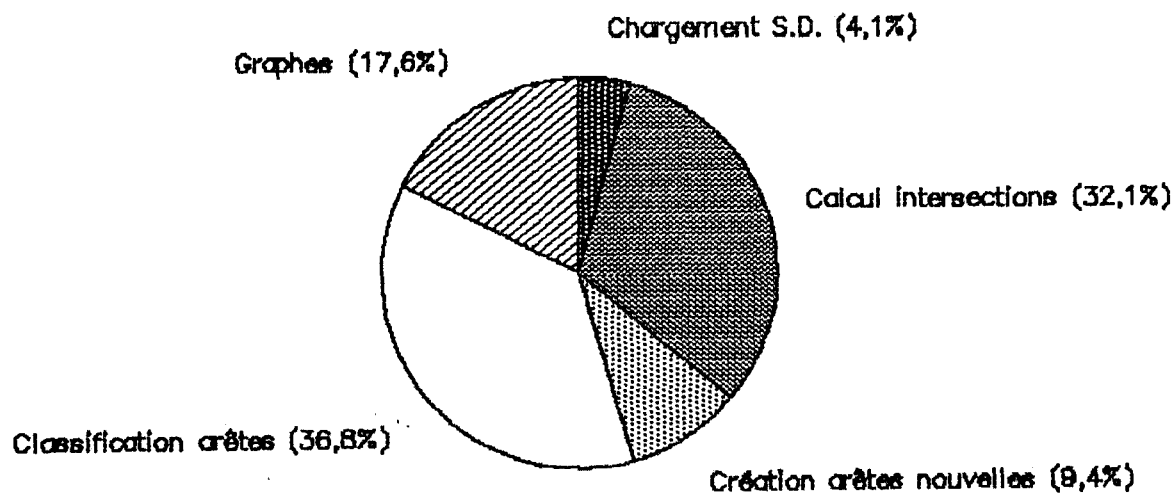


Figure 4.10 : Répartition des temps de traitements

Il est donc certain que l'on obtienne les gains les plus substantiels en optimisant ces deux étapes et cela pour deux raisons :

- Le nombre de calculs effectués inutilement dans ces deux étapes reste élevé car la sélectivité des filtres utilisés est trop faible, les boîtes englobantes maximisant

généralement l'espace occupé par l'entité qu'elles contiennent (Fig 4.11).

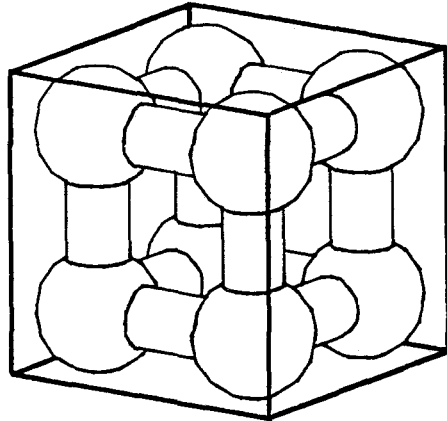


Figure 4.11 : Maximisation de l'espace occupée par un objet

- Les traitements effectués par les phases de création des arêtes nouvelles et de reconstruction des faces peuvent difficilement être réduits sans remettre en cause leurs principes.

Nous proposerons donc deux voies de développement, reposant toutes deux sur une meilleure approximation de l'espace réel occupé par un objet composé, qui devraient permettre de réduire le nombre des calculs effectués dans ces deux étapes et plus particulièrement dans celle de classification des arêtes.

4.1 Arbre des Boites englobantes

Tous les solides primitifs qui sont combinés pour aboutir à un objet complexe sont parfaitement connus au niveau de l'arbre de construction qui le décrit.

Il est donc possible de calculer la boîte englobant chacun de ces solides et l'ensemble de ces boîtes fournira alors une approximation de l'espace occupée par l'objet beaucoup plus précise que celle offerte par sa boîte englobante.

D'un point de vue pratique, les boîtes englobantes des primitives d'un objet ajoutent un niveau à la hiérarchie de ces boîtes proche de celui introduit dans EUCLID_IS par la notion de "figure". Elles permettront donc d'éliminer globalement toutes les faces d'un objet composé provenant d'une primitive

dont la boîte englobante ne coupe pas celle de l'entité à comparer à l'objet, grâce aux liens ajoutés entre les faces d'un polyèdre et la surface théorique représentée, liens qui sont ensuite maintenus lors de l'évaluation des opérations booléennes. Dans notre algorithme, cette propriété peut être exploitée:

- à l'étape de calcul des points d'intersection, lors de la constitution de la liste des faces d'un objet susceptibles de couper la frontière de l'autre,
- à l'étape de classification des arêtes, quand on comptabilise les intersections possédées par un segment avec toutes les faces d'un objets.

Enfin, l'arborescence des boîtes englobantes offre la possibilité de localiser directement un point qui serait à l'intérieur de la boîte englobant un solide mais à l'extérieur de celles englobants ses primitives car il est alors nécessairement à l'extérieur de l'objet.

4.2 Volumes englobants et englobés

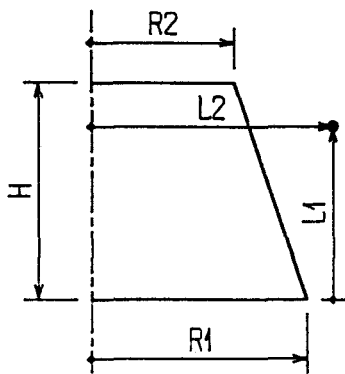
La connaissance des primitives composant un objet complexe, offerte par l'arbre de construction, peut encore être exploitée plus finement pour localiser un point vis à vis de cet objet.

On associe à chaque feuille de l'arbre de construction des volumes englobant et englobé exacts. En utilisant les définitions mathématiques de ces volumes exacts, il est possible d'établir des conditions simples permettent de déterminer si un point est à l'intérieur de l'un de ces volumes :

- Boite Un point sera à l'intérieur de l'un de ces
 - Pyramide solides si les distances signées du point
 - Coin aux plans supports de ses faces sont
- toujours positives

Notons que chacun de ces trois solides primitifs est identique à ses volumes englobant et englobé car il s'agit de solides polyédriques qui sont modélisés sans approximation.

- Sphère La distance du point au centre de la sphère doit être inférieure à la valeur du rayon.
- Cylindre En projetant le point sur l'axe, on peut
- Cône calculer les distances L_1 et L_2 (Fig 4.12)
- Tronc de cône qui devront satisfaire les deux conditions suivantes:



$$0 < L_1 < H$$

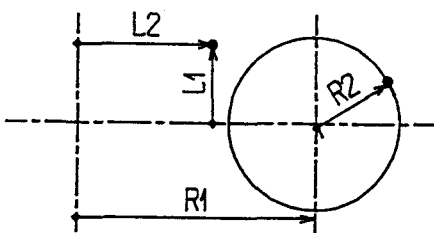
$$|L_2| < \frac{R_2 - R_1}{H} L_1 + R_1$$

Dans le cas du cylindre R_1 et R_2 sont égaux.

Dans le cas du cône R_2 sera nul.

Figure 4.12

- Tore On projette également le point à situer sur l'axe du tore (Fig 4.13), les distance L_1 et L_2 devant vérifier l'équation suivante :



$$L_1^2 = R_2^2 - (L_2 - R_1)^2$$

Figure 4.13 : Appartenance à un tore

Pour chaque opérateurs booléens, il est également possible de définir des règles permettant de connaître la situation du point vis à vis du résultat de la combinaison en fonction de la situation du point vis à vis des volumes englobants et englobés de ses arguments:

- Chapitre 4 -

- Fusion
 - * Le point sera à l'intérieur du résultat si il est à l'intérieur de l'un des volumes englobés par les objets à combiner.
 - * Le point sera à l'extérieur si il est à l'extérieur des deux volumes englobants.

- Différence
 - * Le point sera à l'extérieur du résultat si il est à l'extérieur du volume englobant l'objet subissant la soustraction ou à l'intérieur du volume englobé par l'objet soustrait.
 - * Il sera à l'intérieur si il est à l'extérieur du volume englobant l'objet soustrait et à l'intérieur du volume englobé par l'autre argument de la soustraction.

- Commun
 - * Le point est à l'extérieur du résultat si il est à l'extérieur de l'un des volumes englobants les arguments de l'opération.
 - * Il sera à l'intérieur si il se trouve à l'intérieur des deux volumes englobés par les objets combinés.

En parcourant l'arbre de construction d'un objet composé et en appliquant ces règles simples, on pourra alors déterminer, sans faire de calculs supplémentaires, la situation de tout point qui n'est pas entre les volumes englobant et englobé d'une primitive.

Utilisé à l'étape de classification des arêtes, cette méthode devrait donner de bons résultats, car seule la situation des arêtes situées totalement à l'extérieur ou à l'intérieur d'un objet doit être calculée. La probabilité pour que le point milieu de ces arêtes soit situé entre les volumes englobant et englobé d'une primitive doit donc être généralement très faible, d'autant plus que cette zone d'indétermination n'existe pas pour trois des solides élémentaires.

5. Conclusion

Nous avons consacré la première partie de ce chapitre aux problèmes posés par l'implémentation des différentes phases de l'algorithme élaboré dans le chapitre précédent.

Cette étude nous a conduit à introduire un procédé de filtre par volumes englobants au sein de deux étapes.

L'évaluation des performances de notre algorithme, comparativement à celui d'EUCLID_IS, a permis de montrer la validité des méthodes choisies pour traiter de manière satisfaisante les cas "limites". Cependant, elle a également révélé une évolution des temps de traitements moins favorable lorsque la taille des arguments d'une combinaison augmentait.

Cette situation indique la direction des développements nécessaires afin de rendre cet algorithme réellement opérationnel. Nous avons donc fait deux propositions dans ce sens qui exploitent l'association étroite des modèles frontières et arbre de construction du modèle de produit que nous avons défini.

Conclusion

Notre mémoire présente un système de modélisation destiné à un système intégré de C.F.A.O à orientation mécanique. Il est essentiellement consacré aux algorithmes permettant de calculer les combinaisons booléennes de deux objets modélisés par leurs frontières.

Dans le premier chapitre, l'étude des limitations des systèmes de C.F.A.O actuels montre qu'une modélisation du produit restreinte aux seuls aspects géométriques est le principal obstacle à une intégration réelle des applications participant à l'élaboration d'un produit.

Afin de prendre en compte toutes les informations nécessaires au processus de conception et de fabrication d'un produit manufacturé et de respecter les besoins des concepteurs, nous proposons un modèle de produit basé sur l'approche "features modelling".

Cette approche consiste à décrire une pièce non plus en termes d'entités géométriques de bas niveau mais en termes de caractéristiques, une caractéristique étant définie comme un ensemble d'informations relatifs à un aspect particulier d'une pièce.

Dans le secteur de la conception et de la fabrication mécanique, cinq catégories d'informations apparaissent indispensables à la modélisation d'une pièce:

- les caractéristiques géométriques qui définissent sa forme générale
- les caractéristiques technologiques qui sont des sous_ensembles de sa géométrie liés à une fonction ou un procédé de fabrication particulier

- les caractéristiques de précisions quantifiant les variations admises vis à vis de la forme idéale
- les caractéristiques de sa matière
- les caractéristiques administratives rassemblant les données permettant la gestion du produit au sein de l'entreprise

La mise en oeuvre de ce modèle fait appel à une double modélisation géométrique:

- l'arbre de construction pour sa puissance de description, de manipulation et de paramétrisation des solides,
- la représentation par les frontières qui apporte à l'arbre de construction la connaissance explicite des surfaces de la pièce qui lui manque pour de nombreuses applications telles que le calculs des caractéristiques physiques ou la visualisation sous différentes formes de l'objet conçu.

Afin d'exploiter la complémentarité de ces deux modèles, chaque surface d'une primitive solide de l'arbre de construction est référencée par les faces correspondantes du polyèdre de la représentation approchée par les frontières de cette primitive.

Le maintien constant de la cohérence entre les deux modèles implique que l'on dispose des algorithmes nécessaires à l'évaluation des noeuds de l'arbre de construction et permettant:

- l'application de transformations géométriques à une représentation par les frontières,
- la combinaison, en assurant la propagation des liens introduits entre les deux modèles, d'objets modélisés par leurs frontières.

l'application de transformations géométriques ne posant aucun problème, nous nous sommes donc intéressé exclusivement aux opérateurs booléens.

Le second chapitre précise tout d'abord la notion d'opérateurs booléens dans le contexte de la modélisation solide, puis, par l'étude de deux familles d'algorithmes existant et la mise en évidence de leurs limitations, il permet de mieux cerner la nature des problèmes posés par ces opérateurs:

- les cas "limites" associés aux objets dont les frontières se recouvrent partiellement font apparaître des configurations particulières pour lesquelles un traitement adapté, lié à la notion de régularisation des opérateurs booléens, doit être prévu.
- l'emploi de tolérances sur les résultats des calculs effectués en arithmétique flottante afin de pallier à leur imprécision aboutit à des contradictions entre ces résultats et les informations géométriques et topologiques dont on dispose sur les objets à combiner, contradictions qu'il faut détecter et interpréter correctement.

Afin d'offrir à l'algorithme que nous proposons dans le troisième chapitre la fiabilité qui fait défaut aux deux familles d'algorithmes que nous avons étudiées pour résoudre ces problèmes, celui ci:

- insère, dès leur construction, les entités géométriques qui proviennent de l'intersection des frontières des objets dans leur topologie. Ce procédé permet d'une part d'éviter la duplication de ces entités et d'autre part de détecter et de corriger la plupart des erreurs pouvant apparaître dans les calculs nécessaires à leur création.
- construit chaque face de la surface polyédrique du résultat de la combinaison par une méthode de recherche de cycles dans un graphe. Un graphe est construit pour chaque face altérée par l'intersection à partir des entités existantes ou calculées et les cas limites sont identifiés et exploités lors de la construction du graphe et de la recherche des cycles pour parvenir à la solution correcte.

L'implémentation de cet algorithme est alors étudié au quatrième chapitre et certaines modifications lui sont apportées afin de prendre en compte les contraintes de temps de réponse qui pèsent sur tout système interactif.

Enfin, l'évaluation qualitative de l'algorithme final montre la validité des concepts avancés car tous les modèles tests choisis ont été évalués avec succès, ce qui n'a pas toujours été le cas lorsque ces modèles ont été soumis à l'algorithme équivalent du logiciel de C.F.A.O EUCLID_IS.

L'aspect temps de traitements est cependant moins favorable et nous avons donc proposé des procédés de filtre destinés à limiter le volume de calculs inutilement effectués. Ces techniques sont plus sélectives que la méthode par "boîtes englobantes" actuellement utilisée.

Dotés de l'algorithme ainsi optimisé, le système de modélisation sera alors opérationnel et permettra la réalisation d'applications offrant une assistance accrue à chaque étape du processus de fabrication du produit telles que:

- Une assistance à l'élaboration des gammes de fabrication qui proposera automatiquement des associations de surfaces pertinentes soit en les déduisant directement des caractéristiques technologiques éventuellement présentes dans la modélisation de la pièce, soit en regroupant des surfaces contiguës qui possèdent des caractéristiques de précisions identiques. En outre, les caractéristiques de précisions permettront également de déterminer le nombre et le type des opérations d'usinages nécessaires à la réalisation des surfaces ainsi associées.
- Des générateurs de trajectoires d'outils qui pourront utiliser les capacités de paramétrisation de l'arbre de construction pour demander l'évaluation d'une pièce dont les dimensions sont recadrées dans les intervalles de tolérances définis par les caractéristiques de précisions afin d'exploiter ceux ci au mieux en fonction de l'usure prévisible des outils et de la forme à réaliser.

Bibliographie

- [AVE 89] Evaluation du Transputer sous OCCAM et TDS
J G AVELIN
Rapport interne LGIL Valenciennes Mars 1989
- [BAD 88] Operations booléennes sur les polyèdres :
Evaluation d'arbres C.S.G.
D BADOUEL, G HEGRON
IRISA publication interne n°404 Avril 1988
- [BRU 80] Aide à la conception de bras de robots et de la
cinématique associée avec étude de non interference
spatiale
J M BRUN, M EVRARD, P MACE
Compte rendu de fin d'étude DGRST Fevrier 1980
- [DAB 87] Interactive solid modelling
I DABNEY
Computer_aided engineering journal October 1987
- [DAV 81] Méthodologie pour la construction de système de CAO
B DAVID
Thèse docteur es science informatique
Grenoble 1981
- [FLA 87] Edge_edge relationships in geometric modelling
J FLAQUER, A CARBAJAL, M A MENDEZ
CAD Vol 19 n°5 June 1987
- [FUJ 86] ARTS: Accelerated ray_tracing system
A FUJIMOTO, T TANAKA, K IWATA
IEEE computer graphics and applications April 1986

- [GAU 89] Types de données et algorithmes volume 1
Analyse d'algorithmes
Définition des types de données
M.C. GAUDEL, M SORIA, C FROIDEVAUX
INRIA collection didactique
- [GHA 88] La synthèse d'images
D GHAZANFARPOUR, B PEROCHE, J ARDENCE
D MICHELUCCI
Edition Hermes 1988
- [GIB 83] Méthodologie de conception des systèmes de C.A.O.
Application à la confection
A GIBAUD
Thèse docteur de troisième cycle
Valenciennes 1983
- [GOS 88] Representing dimensions, tolerances and features in
MCAE systems
D.C. GOSSARD, R.P. ZUFFANTE, H SAKURAI
IEEE Computer graphics and applications
Vol 8 n°2 March 1988
- [HEG 85] Synthèse d'image: Algorithmes élémentaires
G HEGRON
Dunod informatique Aout 1985
- [JAC 80] Oct_trees and their use in representing
three_dimensional objects
C L JACKINS, S L TANIMOTO
Computer graphics and image processing n°14 1980
- [JOS 88] Graph_based heuristics for recognition of machined
features from a 3D solid model
S JOSHI, T C CHANG
CAD Vol 20 n°2 March 1988
- [LAI 86] Constructive solid geometry for polyhedral objects
D H LAIDLAW, W B TRUMBORE, J F HUGHES
SIGGRAPH '86
ACM Vol 20 n°4 1986

- [MAN 83a] Set operation of GWB
M MANTYLA
Computer graphics forum North holland Vol 2/3 1983
- [MAN 83b] Localized set operations for solid modelling
M MANTYLA, M TAMMINEN
Computer graphics Vol 17 n°3 1983
- [MAR 85] La synthèse d'image
F MARTINEZ
Editest 1984
- [MAR 87] Etude de la faisabilité d'un système de C.F.A.O.
intégré autour d'une base de données relationnelle
R MARANZANA
MICAD 87 Paris
- [MAR 88] Intégration des fonctions de conception et de
fabrication autour d'une base de données
relationnelle
R MARANZANA
Thèse de doctorat Valenciennes 1988
- [MARK87] What do solid models need?
P MARKS
Machine design March 12, 1987
- [MART87a] Les difficultés et les erreurs possibles dans les
algorithmes de calcul de l'intersection de solides
définis par leur bord
P MARTIN, D MARTIN
CESTA-MARI 1987
- [MART87b] Les algorithmes de calcul de l'intersection de
solides définis par leur bord 1
P MARTIN, D MARTIN
Revue internationale de CFAO et d'infographie
Vol 2 n°4 1987

- [MART88] Les algorithmes de calcul de l'intersection de solides définis par leur bord 2
P MARTIN, D MARTIN
Revue internationale de CFAO et d'infographie
Vol 3 n°2 1988
- [MEA 82] Geometric modeling using octree encoding
D MEAGHER
Computer graphics and image processing
Vol 19 n°2 1982
- [MEZ 86] SIROCO Conception et validation de l'interface de communication et des algorithmes
A MEZIANE
Thèse docteur de troisième cycle
Valenciennes 1986
- [NEW 79] Principles of interactive computer graphics
W M NEWMAN, R F SPROULL
Mc GRAW_HILL book company
Computer science series 1979
- [PLE 88] La modélisation de volumes
D PLEMENOS
Lyon Avril 1988
- [PRE 82] Shape representation and solid modelling in CAD system
K PREISS
CAD systems framework, North holland, June 1982
- [REQ 80] Representation for rigid solids: Theory, methods and systems
A A G REQUICHA
ACM computing surveys Vol 12 n°4 December 1980
- [REQ 83] Solid modeling: Current status and research directions
A A G REQUICHA, H B VOELCKER
IEEE computer graphics and applications
Vol 3 n°7 October 1983

- [ROY 88] Feature_based representational scheme of a solid modeler for providing dimensioning and tolerancing information
U ROY, C R LIU
Robotics and computer_integrated manufacturing
Vol 4 n°3/4 pp.335-345 1988
- [SCH 87] Infographie volume 1 et 2
P SCHWEIZER
Presses polytechniques romandes 1987
- [SHA 88a] Functional requirements and conceptual design of the feature_based modelling system
J J SHAH, M T ROGERS
Computer_aided engineering journal February 1988
- [SHA 88b] Expert form feature modelling shell
J J SHAH, M T ROGERS
CAD Vol 20 n°9 November 1988
- [SZI 84] An algorithm for determining the interaction of two simple polyhedra
M SZILVASI_NAGY
Computer graphics Vol 3 n°3 September 1984
- [TAN 87] Etude et mise en oeuvre d'une nouvelle approche de validité et de cohérence des informations volumiques et surfaciques pour la modélisation géométrique de solides
Y TANG
Thèse de doctorat UTC Compiègne 1987
- [TIL 80] Set membership classification: A unified approach to geometric intersection problems
R B TILOVE
IEEE transactions on computers
Vol 29 n°10 October 1980

- [TIM 80] Computation of global geometric properties of solid
objects
TIMER
CAD Vol 12 n'6 1980
- [TUR 84] A set operation algorithm for two and three
dimensional geometric objects
J A TURNER
Architecture and planning research laboratory
University of Michigan August 1984
- [WOO 88] Some speculations on feature recognition
J R WOODWARD
CAD Vol 20 n'4 May 1988

Bibliothèque Universitaire de Valenciennes



00904914