

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



**Mémoire pour l'obtention
D'un diplôme d'ingénieur d'état en informatique.**
Option : Système d'Information (SI)

Sujet :

**Automatisation de l'Usinage des
Surfaces Gauches par la Méthode
I.T.O**

Présenté par : MAHDI Asmaa
TCHANTCHANE Zahida

Promoteurs : BEY Mohamed
BENDIFALLAH Hassene

Organisme d'accueil : CDTA (Centre de Développement des Technologies Avancées).
Division Productique et Robotique

Soutenue le: date soutenance, devant le jury composé de :

Nom. Président du jury, Mer Bennouar

Président

Nom examinateur 1, Mmed Boustia

Examinateur

Nom examinateur 2, Mer ouledaissa

Examinateur

*R*emerciements:

“ Ce n'est pas dans la connaissance qu'est le bonheur, mais dans l'acquisition de la connaissance ”

Edgar Allan Poe

Avant toute chose, le véritable remerciement revient à grand Dieu tout puissant qui nous a donné la force de réaliser ce travail.

La réussite des travaux de recherche est pour beaucoup le fruit de l'encadrement dynamique et le suivi de nos promoteurs Mr BEY Mohamed et Mr BENDIFALLAH Hassene. Nous les remercions du plus profonds, pour le soutien, l'aide, les conseils, et surtout de nous avoir supporter durant la réalisation de ce projet.

Nous remercions également l'ensemble des personnes du Centre de Développement des Technologies Avancées (CDTA).

Nous tenons tout particulièrement à remercier Mr MESSAID le directeur de notre département d'informatique et MME WAHRANI et tous les professeurs et les membres de l'université SAAd DAHLEB de BLIDA.

Nos remerciements à tous qui nous ont aidé pour réaliser ce travail de prêt ou de loin.

Dédicace

*A toute personne chère à mon cœur je dédie ce
travail.....*

A ma merveilleuse mère pour son amour

A mon père

A mon frère, A Amine

*A mes sœurs « Hadjer, Hind,
Noussaïba »*

A toute ma famille et mes amis

*A vous spécialement Hanen, Sanae & Amira
.....pour le soutien, l'aide et l'amour que vous
m'avez apporté durant mon chemin
universitaire.*

Mahdi Asmaa

إهداء

الحمد لله رب العالمين و الصلاة و السلام على سيدنا محمد و على اله
وأصحابه أجمعين.

يا ربنا لك الحمد و الشكر كما ينبغي لجلال وجهك و عظيم سلطانك
أسالك اللهم صدقا في القول و إخلاصا في العمل و كرامة في الدارين.

أهدي هذا العمل إلى أمي الحنون على دعمها المستمر لي طوال سنوات
حياتي.

إلى أبي على مساندته.
إلى إخوتي محفوظ, عواوش, بشرى و محمد أمين.

إلى ابن أختي محمد هاني.

إلى جدتي العزيزتين على دعواتهما.

إلى صديقتي لمياء, فلة, شافية, سمية, أنيسة, هدى و أحلام

إلى نزيهة, نسيبة و نسرين

الأميين

إلى العزيزين تشانسان محمداً و زوجته الكريمة.

إلى السيدين باي محمد و بن ضيف الله حسان.

و إلى كل من ساعدني من قريب أو من بعيد.

تشانسان زهيدك

Résumé

Ce travail s'insère dans le cadre de développement d'outils de conception et de fabrication des surfaces gauches initié par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) au niveau de la Division Robotique et Productique du Centre de Développement des Technologies Avancées (CDTA).

Dans ce projet nous nous intéressons à l'automatisation de l'usinage des surfaces gauches en finition par la méthode ITO sur des fraiseuses à commande numérique à 3 axes. Le but de notre travail est le développement d'une application logicielle graphique et interactive sous Windows permettant de trianguler les surfaces à usiner, de déterminer le rayon d'outil optimum qui évite les interférences et ensuite de générer la surface des points centre outil en utilisant la méthode ITO et enfin la génération du trajet d'outil par la stratégie des plans verticaux avec les deux modes de balayage One-Way et Zig-Zag ainsi que la génération des programme d'usinage « G-Code ».

Summary

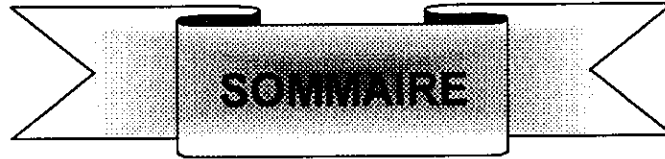
This work is included in development of tools for the design and manufacture of free form surfaces initiated by the team Computer Aided Design and Manufacture of the Robotics and Production Division in the Advanced Technology Development Center (CDTA).

In this project, we are interested in the automation of machining of free form surfaces in the finishing stage using ITO method on 3 axes CNC milling machines. The goal of our work is to develop a graphic and interactive software under Windows to approximate the surfaces with a set of triangles, to find the radius of optimum tool that avoid the interferences, and to generate the center point surface using ITO method and in the end the generation of the tool path and the machining program.

ملخص

إنّ هذا العمل يندرج في إطار تطوير وسيلة، قادرة على تصميم و تصنيع الأسطح ذات الأشكال الحرّة و التي يقوم بتطويرها فرقة البحث الخاصة بالتصميم و التصنيع عن طريق الكمبيوتر على مستوى فرع التصنيع الآلي الخاص بمركز تطوير التكنولوجيا المتقدمة.

نقوم في هذا المشروع، ببرمجة عملية تصنيع المساحات الحرّة باستعمال طريقة القاطعة المقلوبة ITO في آلة التقطيع الرقمية، ذات الثلاث محاور. إنّ الهدف من هذا العمل هو تطوير برنامج آلي تصويري قابل للاستعمال على نظام التشغيل ويندوز و الذي يسمح بما يلي:
ايجاد الشكل الهندسي لمساحة ما، إيجاد نصف قطر القاطعة الملائم و الذي يجنبنا تداخل مساحات التقطيع في ما بينها، تشكيل مساحة نقط مراكز القاطعة، تشكيل مسار التصنيع باستعمال طريقة المحاور العمودية و إحدى الإستراتيجيتين ذهاب و ذهاب- رجوع و أخيراً كتابة برنامج التصنيع آلياً انطلاقاً من السطح المصنّم و المسار المصحح



SOMMAIRE

INTRODUCTION GENERALE	01
------------------------------	-----------



CHAPITRE 1 : METHODES DE CONCEPTION DES SURFACES

I. INTRODUCTION.	03
II. METHODES DE REPRESENTATION DES SURFACES.	03
II.1. Surfaces non paramétriques.	03
II.1.1. Forme explicite.	03
II.1.2. Forme implicite.	03
II.2. Surfaces paramétriques.	04
II.3. Propriétés géométriques des surfaces paramétriques.	04
II.3.1. Vecteurs tangents et vecteur normal.	04
II.3.2. Courbe isoparamétrique.	05
II.3.3. Courbure d'une surface.	06
III. METHODE DE CONCEPTION DES SURFACE EN CAO.	07
III.1. Surfaces basées sur l'utilisation des courbes.	07
III.2. Surfaces basées sur l'utilisation des points.	08
IV. ETUDE DES SURFACES B-SPLINE ET SURFACES NURBS.	08
IV.1. Surface B-Spline.	09
IV.1.1. Définition d'une surface B-Spline.	09
IV.1.2. Structure particulière d'un vecteur nodal.	10
IV.1.3. Formes possibles d'une surface B-Spline .	10
IV.1.4. Propriété des surfaces B-Spline.	11
IV.2. surfaces NURBS.	11
IV.2.1. Propriété des surfaces NURBS.	12
V. CONCLUSION.	12

CHAPITRE 2 : MACHINES OUTILS A COMMANDE NUMERIQUE.

I. INTRODUCTION.	13
II. GENERALITES SUR LE FRAISAGE.	13
II.1. Opérations "en bout".	13
II.2. Opérations "en roulant".	13
III. MODES D'USINAGE.	14
III.1. Usinage en avalant.	14
III.2. Usinage en opposition.	14
IV. MOUVEMENTS GENERATEURS DES FORMES.	14
V. PARAMETRES DE COUPE.	15
V.1. Vitesse de coupe.	15
V.2. Détermination de la fréquence de rotation.	15
V.3. Vitesse d'avance.	15
VI. DEFINITION D'UNE MACHINE-OUTIL A COMMANDE NUMERIQUE.	16
VI.1. Définition.	16
VI.2. Architecture des fraiseuses à commande numérique.	16
VI.2.1. Partie opérative.	16
VI.2.2. Partie commande.	17
VII. AXES DE DEPLACEMENT.	18
VII.1. Mouvement de translation.	18
VII.2. Mouvement de rotation.	18
VII.3. Types d'axes d'Une MOCN.	19
VIII. TYPES DE FRAISEUSES.	19
IX. PROGRAMMATION DES MOCN	19
IX.1. programme d'usinage.	20
IX.1.1. Format d'un mot.	20
IX.1.2. Format d'un bloc.	21
IX.2. Caractéristique d'un programme.	21
IX.3. Structure d'un programme ISO.	21
X. CONCLUSION.	22

CHAPITRE 3 : USINAGE DES SURFACES GAUCHES EN FINITION.

I. INTRODUCTION.	23
I. CHOIX DES FORMES D'OUTILS.	23
III. PARAMETRES DE PASSE.	24
III.1. Pas longitudinal.	24
III.2. Pas transversal.	24
IV. PROBLEMES D'INTERFERENCES ET DE COLLISIONS.	25
IV.1. Problème d'interférence.	25
IV.2. Problème de collision.	25
V. METHODES DE GENERATION DE CHEMIN D'OUTIL.	26
V.1. Méthode du copiage informatique.	26
V.2. Méthode basé sur le point de contact CC (CC-Based Method).	26
V.3. Méthode basée sur le point d'extrémité CL (CL-Based Method).	27
V.3.1. Méthode de calcul par offset de la forme.	27
V.3.2. Méthode de l'offset inverse (ITO).	28
VI. PRINCIPALES STRATEGIES D'USINAGE DES SURFACES GAUCHES.	30
VI.1. Courbes isoparamétriques.	30
VI.2. Plans parallèles.	31
VI.2.1. Plans verticaux.	31
VI.2.2. Plans horizontaux (Z-CONSTANT).	32
VII. CONCLUSION.	32

CHAPITRE 4 : TRIANGULATION DES SURFACES ET CALCUL DES INTERSECTIONS

I. INTRODUCTION.	33
II. TRIANGULATION.	33
II.1. Méthodes de triangulation.	34
II.1.1. Triangulation uniforme.	34
II.1.2. Triangulation adaptative.	34
III. INTERSECTIONS	36
III.1. Cas d'intersection d'un triangle et un plan dans l'espace 3D.	36
III.2. Equation de la droite dans l'espace.	38
III.3. Equation du plan dans l'espace.	38
III.4. Intersection entre un plan et une droite dans l'espace.	39

III.5. Appartenance d'un point à un segment de droite.	39
III.6. Equation d'un cylindre dans l'espace.	39
III.7. Intersection entre une droite et un cylindre .	40
III.8. Equation d'une sphère.	41
III.9. Intersection d'une droite et d'une sphère.	41
III.10. Equation du tore.	41
III.11. Intersection d'une droite et un tore.	42
IV. CONCLUSION	43

CHAPITRE 5 : CONCEPTION DE L'APPLICATION LOGICIELLE

I. INTRODUCTION.	44
II. METHODE DE MODELISATION.	44
III. REALISATION DE L'APPLICATION.	44
III.1. Cahier de charges.	44
III.1.1. Présentation du projet.	44
III.1.2. Problématique.	44
III.1.3. Objectifs visés.	45
III.1.4. Plateforme exigée.	46
III.2. Solution de la problématique.	46
III.3. Modélisation de l'application en UML.	46
III.3.1. Diagramme cas d'utilisation.	47
III.3.2 Diagramme de séquence.	49
III.3.3. Diagramme de collaboration.	54
III.3.4. Diagramme d'activité.	57
III.3.5. Diagramme de classes.	59
IV. CONCLUSION.	68

CHAPITRE 6 : IMPLEMENTATION

I. INTRODUCTION.	69
II. PRESENTATION DE L'APPLICATION.	69
III. IMPLEMENTATION.	69
III.1. Fenêtre principale.	69
III.2. Barre du menu principal.	70
III.3. Rubrique d'usinage en finition par la méthode ITO.	70

IV. PRESENTATION DES FENETRES.	70
IV.1. Fenêtre de triangulation des surfaces.	70
IV.2. Fenêtre du choix d'outils.	75
IV.2.1. Calcul du rayon théorique.	76
IV.2.2. Visualisation des sommets selon la forme.	76
IV.2.3. Génération des régions.	77
IV.2.4. Détection et correction des interférences.	78
IV.3. Fenêtre du choix d'outils final.	80
IV.4. Fenêtre de méthode I.T.O.	82
IV.5. Paramètres de la méthode des plans parallèles.	84
IV.6. Trajet d'outil.	87
V. CONCLUSION.	89

CHAPITRE 7 : TESTS ET VALIDATIONS.

I. INTRODUCTION.	90
II. TESTS ET VALIDATIONS.	90
II.1. Surface ondulée.	91
II.1.1. Approximation du modèle CAO.	91
II.1.2. Choix d'outil théorique.	93
II.1.3. Choix d'outil optimum.	94
II.2. Surface cavité_électrode.	95
II.2.1. Approximation du modèle CAO.	95
II.2.2. Choix d'outil théorique.	97
II.2.3. Choix d'outil optimum.	97
II.2.4. Choix d'outil définitif.	98
II.2.5. Paramètre d'usinage.	100
II.2.6. Méthode ITO.	101
II.2.7. Usinage par la méthode des plans parallèles.	101
III. CONCLUSION.	105
CONCLUSION GENERALE.	106
ANNEXE A « G-Code ».	107
ANNEXE B « UML ».	111
REFERENCES BIBLIOGRAPHIQUES.	118

Liste des figures

CHAPITRE 1 : METHODES DE CONCEPTION DES SURFACES

Figure 1 : Surface paramétrique.	04
Figure 2 : Vecteur tangent et vecteur normal d'une surface paramétrique.	05
Figure 3 : Courbes isoparamétriques d'une surface paramétrique.	05
Figure 4 : Courbure d'une courbe en un point x.	06
Figure 5 : Relations entre les différentes représentations.	09
Figure 6 : Paramètres de définition d'une surface B-Spline.	10
Figure 7. Paramètres de définition d'une surface NURBS.	12

CHAPITRE 2 : MACHINES OUTILS A COMMANDE NUMERIQUE.

Figure 1 : Fraisage en bout.	13
Figure 2 : Fraisage en roulant.	14
Figure 3 : Modes d'usinage.	14
Figure 4 : Paramètres de coupe en fraisage.	16
Figure 5 : Machines outils à commande numérique.	16
Figure 6 : Partie commande et partie opérative d'une MOCN.	17
Figure 7 : Modèle de la structure d'une machine outil.	17
Figure 8 : Orientation des axes X, Y et Z.	18
Figure 9 : Axes linéaires et axes de rotation.	18
Figure 10 : Sens positif d'un axe de rotation.	19
Figure 11: Types de fraiseuses.	19
Figure 12 : Format général d'un programme d'usinage.	20
Figure 13 : Format général d'un mot.	20
Figure 14 : Format général d'un bloc.	21
Figure 15 : Structure d'un programme ISO.	22

CHAPITRE 3 : USINAGE DES SURFACES GAUCHES EN FINITION.

Figure 1 : Processus de réalisation d'un usinage.	23
Figure 2 : Types de fraises.	24
Figure 3 : Erreur de flèche.	24

Figure 4 : Distance maximale entre passes et hauteur de crête.	24
Figure 5 : Outil en position d'interférence et en collision.	25
Figure 6 : Différents cas de collisions.	25
Figure 7 : Copiage informatique.	26
Figure 8 : Position de la fraise hémisphérique par rapport à la surface.	27
Figure 9 : Surface offset d'une surface.	28
Figure 10 : Outils utilisés dans la méthode ITO.	28
Figure 11 : Génération de la surface offset par la méthode ITO.	29
Figure 12 : Génération de la surface offset par la méthode ITO pour des outils toriques et cylindriques.	29
Figure 13 : Méthode ITO pour un outil hémisphérique.	30
Figure 14 : Usinage par des courbes isoparamétriques.	30
Figure 15 : Modes de balayage suivant les courbes isoparamétriques.	31
Figure 16 : Méthodes des plans verticaux.	31
Figure 17 : Stratégie d'usinage par plans horizontaux (Z-Constant).	32

CHAPITRE 4 : TRIANGULATION DES SURFACES ET CALCUL DES INTERSECTIONS

Figure 1 : Triangulation d'une surface.	33
Figure 2 : Choix de la bonne triangulation.	34
Figure 3 : Triangulation uniforme d'une surface nominale.	34
Figure 4 : Critères de la triangulation adaptative.	35
Figure 5: Différents cas de subdivision d'un triangle.	35
Figure 6 : Triangulation adaptative d'une surface nominale.	36
Figure 7 : Intersection entre un triangle et un plan.	37
Figure 8 : Droite dans l'espace.	38
Figure 9 : Plan vertical.	38
Figure 10 : Position des différents points par rapport au segment [AB].	39
Figure 11 : Cylindre dans l'espace.	40
Figure 12 : intersection d'une droite (D) avec un cylindre.	40
Figure 13 : Sphère de centre P_0 et de rayon R.	41
Figure 14 : Intersection d'une droite et une sphère.	41
Figure 15 : Un tore dans l'espace.	42
Figure 16 : Différentes formes d'un tore.	42
Figure 17 : intersection d'un tore avec une droite.	43

CHAPITRE 5 : CONCEPTION DE L'APPLICATION LOGICIELLE

Figure 1 : Diagramme de cas d'utilisation.	48
Figure 2 : Diagramme de séquence d'ouverture du modèle CAO.	49
Figure 3 : Diagramme de séquence d'approximation du modèle.	59
Figure 4 : Diagramme de séquence de choix d'outils théorique.	50
Figure 5 : Diagramme de séquence de choix d'outil optimum.	51
Figure 6 : Diagramme de séquence de choix d'outil définitif.	51
Figure 7 : Diagramme de séquence de choix d'outil automatique.	52
Figure 8 : Diagramme de séquence de choix d'outil manuel	52
Figure 9 : Diagramme de séquence de la génération de surface offset.	53
Figure 10 : Diagramme de séquence de la génération de trajet d'outils	54
Figure 11 : Diagramme de collaboration.	56
Figure 12 : Diagramme d'activité d'ouverture du modèle CAO.	57
Figure 13 : Diagramme d'activité d'approximation du modèle CAO	57
Figure 14 : Diagramme d'activité de choix d'outil.	58
Figure 15 : Diagramme d'activité de l'usinage par la méthode plan parallèle.	58
Figure 16 : Diagramme de classes.	59
Figure 17 : Classe de triangulation.	61
Figure 18 : Classe de LIST_SOMMET.	62
Figure 19 : Classe de NŒUD_SOMMET.	62
Figure 20 : Classe de SOMMET.	62
Figure 21 : Classe de LIST_SEGMENT.	63
Figure 22 : Classe de NŒUD_SEGMENT.	63
Figure 23 : Classe de SEGMENT.	64
Figure 24 : Classe de LIST_TRIANGLE.	64
Figure 25 : Classe de NŒUD_TRIANGLE.	65
Figure 26 : Classe de TRIANGLE.	65
Figure 27 : Classe d'INFORMATION.	65
Figure 28 : Classe de RAYON D'OUTIL.	65
Figure 29 : Classe de PLAN VERTICAL.	66
Figure 30 : Classe de LIST_REGION.	66
Figure 31 : Classe de LES REGION.	67
Figure 32 : Classe de POINT_D'INTERFERENCE.	68

CHAPITRE 6 : IMPLEMENTATION

Figure 1: Fenêtre principale.	69
Figure 2 : Accès à l'usinage en finition par la méthode ITO.	70
Figure 3: Fenêtre de triangulation des surfaces.	71
Figure 4 : Critères de triangulation.	72
Figure 5 : Calcul du nombre de lignes et de colonnes.	72
Figure 6 : Création des sommets.	73
Figure7 : Création des segments.	73
Figure8 : Génération des triangles.	74
Figure 9 : Fenêtre d'affichage des résultats de la triangulation.	75
Figure 10: Fenêtre paramètres de dessin.	75
Figure 11 : Fenêtre choix d'outils.	76
Figure 12 : Couleurs des formes locales de la surface.	77
Figure 13: Régions de la surface avec ces points.	77
Figure14 : Détection des interférences.	79
Figure15 : Correction des interférences.	79
Figure 16 : Fenêtre d'affichage des rayons.	80
Figure 17 : Fenêtre de choix final des outils.	80
Figure 18: Modes de choix d'outils.	81
Figure 19 : Fenêtre des paramètres d'usinage.	81
Figure 20: régions des segments.	82
Figure 21: principe de génération de la surface offset par la méthode I.T.O.	83
Figure 22 : Fenêtre Méthode de l'outil inverse (I.T.O).	84
Figure 23 : Fenêtre paramètre de la méthode des plans parallèles.	85
Figure 24 : fenêtre trajet d'outil.	87
Figure 25 : Mode de balayage One-Way.	89
Figure 26 : Mode de balayage Zig-Zag.	89

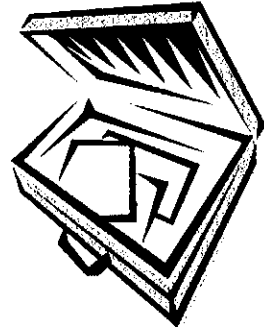
CHAPITRE 7 : TESTS ET VALIDATIONS.

Figure 1 : Surface ondulée.	90
Figure 2 : Surface cavité_électrode.	90
Figure 3 : Scénario d'approximation du modèle CAO.	91
Figure 4 : Résultats de la triangulation uniforme.	92

Figure 5 : Résultats de la triangulation adaptative.	92
Figure 6: Triangles voisins.	93
Figure 7 : Résultats des différents cas.	93
Figure 8 : Scénario du choix d'outil théorique.	93
Figure 9 : Affichage du rayon théorique.	94
Figure 10 : Différentes formes locales de la surface.	94
Figure 11 : Scénario de choix d'outil optimum.	94
Figure 12 : Régions et points des régions.	95
Figure 13 : Triangulation uniforme.	95
Figure 14 : Triangulation adaptative.	96
Figure 15 : Résultats des différentes surfaces.	96
Figure 16 : Rayon théorique propre pour chaque surface.	97
Figure 17 : Rayon théorique identique pour toutes les surfaces.	97
Figure 18 : Régions et points des régions.	97
Figure 19 : Différentes formes locales de la surface.	97
Figure 20 : Points d'interférence.	98
Figure 21 : Résultat du rayon corrigé.	98
Figure 22 : Scénario de choix d'outil définitif.	98
Figure 23 : Choix automatique de l'outil.	99
Figure 24 : Choix manuel de l'outil.	99
Figure 25 : Diagramme de séquence de paramètres d'usinage.	100
Figure 26 : Surface offset par la méthode ITO.	101
Figure 27. Diagramme de séquence de la méthode des plans parallèles.	101
Figure 28 : Différentes positions des centre d'outils.	103
Figure 29 : Simulation du mouvement.	104
Figure 30 : Programme G-Code.	105

ANNEXE B « UML »

Figure 1: Principales étapes de la définition d'UML.	112
Figure 2: les différentes vues du système.	113
Figure 3: Un exemple de diagramme de cas d'utilisation.	114
Figure 4: Un exemple de diagramme de classes.	115
Figure 5: Exemple de diagramme de collaboration entre objets.	115
Figure 6: Exemple de diagramme de séquence.	116
Figure 7: Exemple de diagramme d'activité.	117



INTRODUCTION
GENERALE

INTRODUCTION GENERALE

I. SITUATION DU PROBLEME

Avec le développement qu'a connu l'outil informatique en terme de fiabilité et de rapidité de calcul, celui-ci est devenu ces dernières années un outil nécessaire dans tous les domaines de la technologie et de l'automatisation puisqu'il permet aux industriels d'augmenter la productivité tout en assurant une meilleure qualité des produits. Comme toute autre spécialité, l'industrie mécanique a été aussi développée par l'utilisation de cet outil. Cette industrie ne cherche pas seulement la quantité et la qualité mais avec la concurrence industrielle, elle cherche à réduire le cycle de développement d'un produit (temps de développement) depuis la conception, l'analyse jusqu'à la fabrication. Pour réduire les temps de fabrication, les grandes et même les petites et moyennes entreprises utilisent des logiciels de CFAO. Ces logiciels permettent d'assister les ingénieurs dans les phases de conception, d'analyse et de planification de l'usinage sur des machines à commande numérique.

Les pièces de formes gauches ou libres sont très utilisées dans notre vie quotidienne. Comme toute pièce utilisée en mécanique, les pièces de formes gauches sont conçues dans le but d'assurer des fonctions inscrites dans le cahier des charges. Vu la complexité des formes de ces pièces et les différents paramètres à prendre en compte lors de l'usinage, ces pièces ne peuvent être usinées que sur des machines outils à commande numérique en particulier les fraiseuses à 03 ou à 05 axes. L'usinage de ces surfaces passe généralement par trois étapes : ébauche qui permet d'enlever le maximum de matière, demi finition où on s'approche de la forme finale et finition où on obtient la forme voulue. Toutes ces étapes nécessitent la génération d'un ensemble d'instructions écrites dans un langage propre à la machine appelé programme « G-Code ».

Le travail que nous présentons dans ce mémoire s'inscrit dans le cadre de développement d'outils de conception et de fabrication des surfaces de formes libres (surfaces gauches) initié par l'équipe CFAO de la Division Productique et Robotique du Centre de Développement des Technologies Avancées (CDTA). Notre projet est une continuité des travaux précédents traitant de la :

- Modélisation et conception des courbes et des surfaces B-Spline et NURBS;
- Reconstruction des courbes et des surfaces B-Spline et NURBS par interpolation et par approximation à partir d'un nuage de points;
- Usinage des surfaces gauches par les courbes isoparamétriques,
- Usinage des surfaces gauches par la stratégie des plans parallèles,
- Usinage des surfaces gauches par la stratégie Z-Constant,
- Ebauchage des surfaces gauches,
- Simulation de l'usinage des surfaces gauches.

Nous pouvons synthétiser l'objectif global de notre travail de la façon suivante :

**« Automatisation de l'usinage des surfaces gauches en finition
par la méthode ITO »**

Deux solutions de fabrication existent, l'usinage par balayage d'une fraise et l'usinage de forme par électroérosion. Pour l'usinage par balayage, deux grandes approches sont utilisées. L'approche basée sur la notion de « Cutter Contact » et l'approche « Cutter Location ». La dernière approche est plus fiable et n'a pas les problèmes de la première approche. Parmi les méthodes de la deuxième approche, nous trouvons la méthode ITO qui permet l'obtention de la surface des centres outils avec n'importe qu'elle type d'outil et évite les problèmes d'interférence.

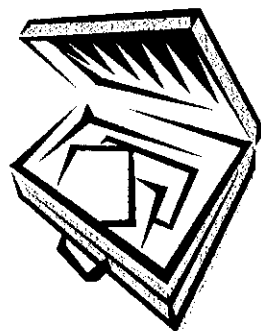
II. OBJECTIF DU TRAVAIL

Dans ce projet, nous nous intéresserons à l'automatisation de l'usinage des surfaces gauches en finition par la méthode ITO sur des fraiseuses à commande numérique à 03 axes. Le but de notre projet est le développement d'une application logicielle graphique et interactive sous Windows permettant l'obtention d'une bonne approximation d'un modèle CAO des surface NURBS, le calcul des outils optimums qui évitent les problèmes d'interférences, la génération de la surface centre outil par la méthode ITO pour des outils hémisphériques, et enfin la génération du chemin d'outils en utilisant la stratégie d'usinage par plans parallèles pour les modes de balayage One-Way et Zig-Zag.

III. DESCRIPTION DU TRAVAIL

Le présent mémoire est composé de sept chapitres :

- Dans le premier chapitre, nous allons étudier les méthodes de représentation et de conception des surfaces et en particulier les surfaces NURBS et B-Spline.
- Le deuxième chapitre est consacré à la présentation de l'architecture et de la programmation des machines outil à commande numérique (MOCN) et en particulier les fraiseuses.
- Dans le troisième chapitre, nous allons présenter les différentes formes d'outils et les différentes stratégies d'usinage des surfaces gauches..
- Dans le quatrième chapitre, nous allons étudier les différentes méthodes d'approximation des surfaces gauches et l'intersection d'une droite avec quelques objets géométriques.
- La conception de notre application logicielle est présentée dans le cinquième chapitre.
- L'implémentation de notre application logicielle est détaillée dans le sixième chapitre.
- Le dernier chapitre présente les tests et la validation des résultats.



CHAPITRE 1
MÉTHODES DE CONCEPTION
DES SURFACES

I. INTRODUCTION :

La conception des surfaces de formes libres (surfaces gauches) en mécanique est devenue une spécialité très importante dans le domaine de la conception assistée par ordinateur (CAO). Ces surfaces sont très utilisées dans le vécu industriel pour la conception et la fabrication des moules, matrices, formes aérodynamiques et des formes esthétiques. Pour cela, nous avons besoins des outils et des méthodes permettant la conception, l'édition et la modification de ces surfaces d'une manière interactive et en temps réel afin d'obtenir la forme voulue.

Dans ce chapitre, nous allons étudier les différentes méthodes utilisées dans la conception des surfaces gauches et en particuliers les surfaces paramétriques.

II. METHODES DE REPRESENTATION DES SURFACES :

Pour modéliser les surfaces en CAO, deux types de surfaces peuvent être utilisés:

- Surfaces non paramétriques,
- Surfaces paramétriques.

II.1. Surfaces non paramétriques :

Ces surfaces peuvent être représentées sous deux différentes formes :

II.1.1. Forme explicite [1, 3] :

Une surface explicite est donnée par l'équation :

$$Z = F(X, Y) \quad (I.1)$$

Où à chaque valeur de X et Y correspond une et une seule valeur de Z.

Cette forme n'est pas utilisée dans la CAO à cause de :

- Complexité des traitements (la translation ou la rotation),
- Impossibilité de représenter toutes les surfaces.

II.1.2. Forme implicite [1, 2] :

Une surface implicite est définie par une fonction de trois variables :

$$F(X, Y, Z) = 0 \quad (I.2)$$

La forme implicite n'a pas la limitation de la représentation explicite, mais la manipulation interactive des surfaces est difficile et elle ne permet pas de représenter toutes les surfaces.

II.2. Surfaces paramétriques [1,8] :

Cette représentation nécessite :

- Deux paramètres u et v appartenant à l'intervalle $[0,1]$,
- Un ensemble de fonctions f , g et h pour déterminer les coordonnées X , Y et Z d'un point de coordonnées u et v .

Ce type de surface est donné par la formule suivante :

$$F(u, v) = (f(u, v), g(u, v), h(u, v)) \quad (1.3)$$

La figure suivante représente la transformation d'un point de coordonnées paramétriques u et v vers un point en 3D de coordonnées X , Y et Z .

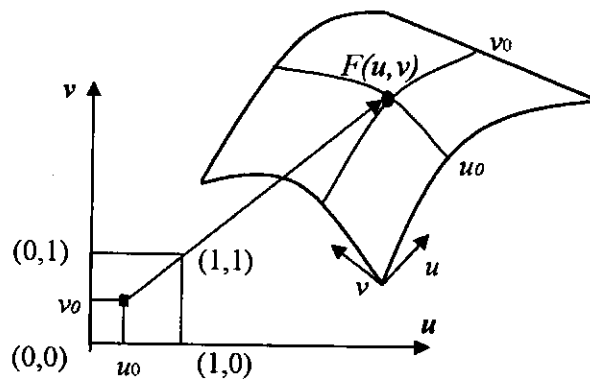


Figure 1 : Surface paramétrique.

En CAO, pour obtenir la forme d'une pièce donnée il faut utiliser plusieurs surfaces paramétriques avec des conditions de raccordement.

II.3. Propriétés géométriques des surfaces paramétriques [1,2] :

Les principales propriétés géométriques des surfaces paramétriques sont:

II.3.1. Vecteurs tangents et vecteur normal [1-3, 8] :

En chaque point de la surface paramétrique, nous pouvons définir un plan tangent à la surface à partir des vecteurs tangents \vec{T}_u et \vec{T}_v au même point.

$$\vec{T}_u = \frac{\partial F}{\partial u} = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) \quad (1.4)$$

$$\vec{T}_v = \frac{\partial F}{\partial v} = \left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right) \quad (1.5)$$

Le vecteur normal en un point de la surface paramétrique est le produit vectoriel des deux vecteurs tangents \vec{T}_u et \vec{T}_v et il est donné par :

$$\vec{n} = \frac{\vec{T}_u \times \vec{T}_v}{|\vec{T}_u \times \vec{T}_v|} \tag{1.6}$$

La figure suivante représente les vecteurs tangents \vec{T}_u et \vec{T}_v , le vecteur normal \vec{n} et le plan tangent Π en un point $F(u, v)$ d'une surface paramétrique.

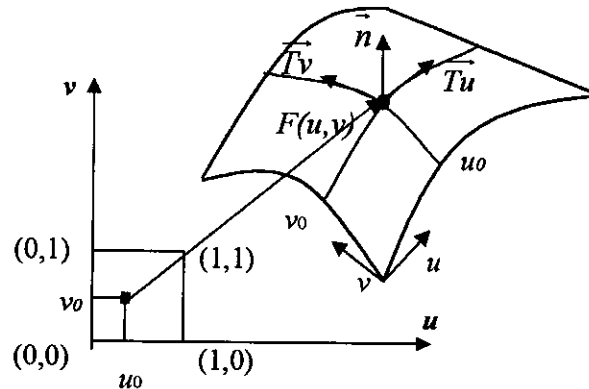


Figure 2 : Vecteur tangent et vecteur normal d'une surface paramétrique.

II.3.2. Courbe isoparamétrique [1,4] :

Une surface paramétrique peut être considérée comme l'union d'un nombre infini de courbes appelées courbes isoparamétrique. La surface paramétrique $F(u, v)$ est obtenue en fixant l'un des deux paramètres et en faisant varier l'autre paramètre ce qui génère une courbe isoparamétrique dans la direction de la variable variée. La figure 3 montre deux courbes isoparamétriques (une pour chaque direction).

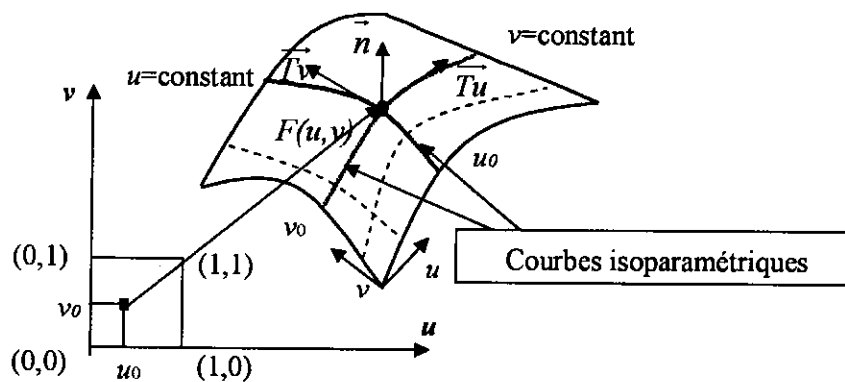


Figure 3 : Courbes isoparamétriques d'une surface paramétrique.

II.3.3. Courbure d'une surface [1,5] :

II.3.3.1. Notion de courbure :

En un point d'une courbe, la courbure k est l'inverse du rayon R du cercle osculateur (le cercle qui approxime le mieux la courbe que tout autre cercle). Ce rayon est appelé rayon de courbure (voir figure 4). Cette courbure est donnée par l'équation suivante:

$$K = \frac{1}{R} \quad (1.7)$$

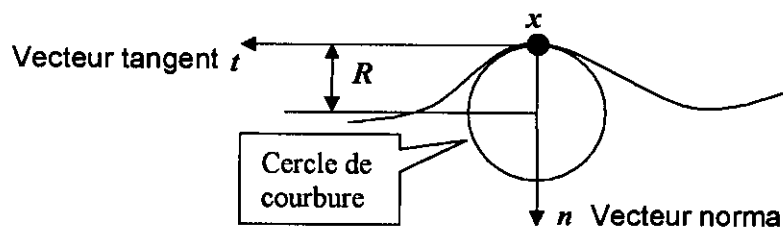


Figure 4 : Courbure d'une courbe en un point x

II.3.3.2. Courbures principales d'une surface paramétrique [1] :

Les surfaces sont caractérisées par deux courbures contrairement aux courbes qui sont caractérisées par une seule courbure. Les courbures principales K_1 et K_2 en un point d'une surface paramétrique sont calculées à partir du calcul des rayons de courbure principaux qui sont les solutions de l'équation du second degré suivante :

$$(LM - M^2) * K^2 + (2MF - GL - EN) * K + EG - F^2 = 0 \quad (1.8)$$

Où L , E , N , G , M et F sont donnés par les relations suivantes :

$$E = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial u} \quad (1.9)$$

$$F = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \quad (1.10)$$

$$G = \frac{\partial P}{\partial v} \times \frac{\partial P}{\partial v} \quad (1.11)$$

$$L = n \times \frac{\partial^2 P}{\partial u^2} \quad (1.12)$$

$$M = n \times \frac{\partial^2 P}{\partial u \partial v} \quad (1.13)$$

$$N = n \times \frac{\partial^2 P}{\partial v^2} \quad (I.14)$$

A partir des courbures principales, la courbure moyenne H et la courbure gaussienne K sont données par :

$$H = \frac{K_1 + K_2}{2} \quad (I.15)$$

$$K = K_1 \times K_2 \quad (I.16)$$

En un point x de la surface et en fonction des valeurs des deux courbures (H, K) la surface aura les propriétés suivantes (forme locale):

- Si $K_1 > 0$ et $K_2 > 0$ alors x est un point elliptique (concave),
- Si $K_1 < 0$ et $K_2 < 0$ alors x est un point elliptique (convexe),
- Si $K < 0$ alors x est un point hyperbolique (selle de cheval),
- Si $K = 0$ et ($k_1 > 0$ ou $k_2 > 0$) alors x est un point parabolique (développable concave),
- Si $K = 0$ et ($k_1 < 0$ ou $k_2 < 0$) alors x est un point parabolique (développable convexe),
- Si $K = 0$ et $H = 0$ alors x est un point plat.

III. METHODE DE CONCEPTION DES SURFACE EN CAO [1,4] :

Dans plusieurs situations de conception de nouvelles surfaces, le concepteur n'a en tête que la forme de la surface qu'il veut obtenir sans connaître ni sa représentation mathématique ni ses propriétés géométrique. Donc, il faut qu'il dispose de méthodes lui permettant d'obtenir les surfaces voulues. Pour cela deux classes de méthodes sont utilisées dans la conception des surfaces.

- Méthodes basées sur l'utilisation des courbes.
- Méthodes basées sur l'utilisation des points.

III.1. Surfaces basées sur l'utilisation des courbes [2,4] :

La détermination des surfaces dans cette méthode est basée sur la connaissance de quelques courbes. Cette méthode est utilisée dans la conception des surfaces suivantes :

- Surface balayée,
- Surface de Coons,
- Surface de Gordon,

- Surface de révolution,
- Surface extrudée,
- Surface lissée,
- Surface réglée.

III.2. Surfaces basées sur l'utilisation des points [1,6] :

Dans cette méthode, la conception de la surface repose sur la donnée d'un ensemble de points. Cette classe englobe :

- Interpolation d'un nuage de point (globale ou locale),
- Approximation d'un nuage de point (globale ou locale),
- Surfaces de Bézier,
- Surfaces de Bézier Rationnelle,
- Surfaces B-Spline,
- Surfaces NURBS (Non-Uniform Rational B-Spline).

Dans notre travail nous allons considérer les surfaces les plus utilisées en CAO à savoir les NURBS et les B-Spline.

Le choix d'une méthode de conception repose sur les critères suivants :

- La représentation de la surface d'une manière uniforme,
- La représentation des surfaces fermées,
- La modification de la surface à concevoir doit être intuitive et en temps réel,
- Les transformations géométriques (changement d'échelle, de repère.) n'influent pas sur la représentation de la surface,
- Moins coûteuse en source informatique et la plus représentative de l'objet,
- Peut satisfaire au maximum les contraintes de dessin et certaine, contrainte de continuité.

IV. ETUDE DES SURFACES B-SPLINE ET SURFACES NURBS [1,5] :

La modélisation des surfaces en CAO repose sur une technique interactive faisant appel à des fonctions paramétriques polynomiales ou rationnelles à coefficients vectoriels. Le premier modèle de surfaces sous cette forme fut celui de Bézier. Ce dernier fut détrôné par les courbes et surfaces B-Spline qui sont aujourd'hui à leur tour généralisées par les NURBS (Abréviation anglaise pour B-Spline rationnelles non uniformes). La relation qui existe entre les trois types de représentation des surfaces est montrée par la figure 5.

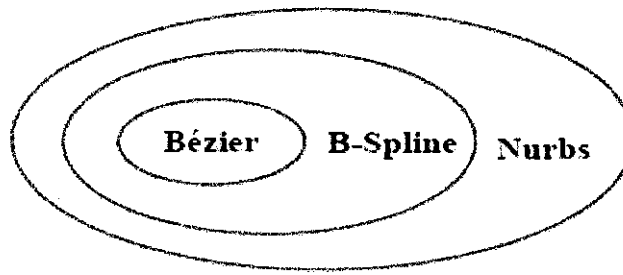


Figure 5 : Relations entre les différentes représentations.

IV.1. Surface B-Spline :

IV.1.1. Définition d'une surface B-Spline [2,3] :

Une surface B-Spline est définie par les paramètres suivants (voir figure 6):

- Un réseau de $(m+1) \times (n+1)$ points de contrôles $p_{i,j}$, où $0 \leq i \leq m$ et $0 \leq j \leq n$. Ce réseau est appelé « réseau de contrôle »,
- Un vecteur nodal de $(h+1)$ nœuds dans la direction u , $U = \{u_{0=0}, u_1, \dots, u_{n=1}\}$,
- Un vecteur nodal de $(k+1)$ nœuds dans la direction v , $V = \{v_{0=0}, v_1, \dots, v_{m=1}\}$,
- Un degré p dans la direction u et un degré q dans la direction v .

La surface de B-Spline est donnée par l'équation suivante :

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) \cdot N_{j,q}(v) \cdot P_{i,j} \quad (I.17)$$

Où $N_{i,p}(u)$ et $N_{j,q}(v)$ sont des fonctions de base B-Spline de degré p et q respectivement. Ces fonctions sont données par [4,5] :

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1} \\ 0 & \text{sinon} \end{cases} \quad (I.18)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (I.19)$$

$$N_{j,0}(v) = \begin{cases} 1 & \text{si } v_j \leq v < v_{j+1} \\ 0 & \text{sinon} \end{cases} \quad (I.20)$$

$$N_{j,q}(v) = \frac{v - v_j}{v_{j+q} - v_j} N_{j,q-1}(v) + \frac{v_{j+q+1} - v}{v_{j+q+1} - v_{j+1}} N_{j+1,q-1}(v) \quad (I.21)$$

Il est à noter que les identités fondamentales, une pour chaque direction, doivent être vérifiées :

$$\begin{cases} h = m + p + 1 \\ k = n + q + 1 \end{cases} \quad (1.22)$$

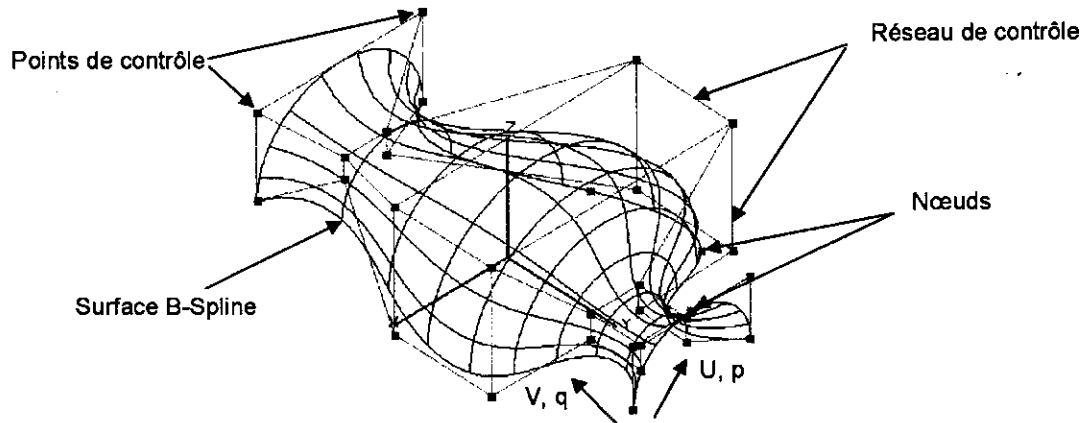


Figure 6 : Paramètres de définition d'une surface B-Spline.

IV.1.2. Structure particulière d'un vecteur nodal [1,8] :

Un vecteur nodal est un ensemble de $(h+1)$ nœuds dont les valeurs nodales sont croissantes $u_0 \leq u_1 \leq \dots \leq u_h$.

- Si $(u_i = u_{i+1} = \dots = u_{i+k-1})$ ce nœud est dit « nœud multiple » de multiplicité k , où k est le nombre d'apparition de ce nœud, dans le cas contraire le nœud est dit « nœud simple ».
- Si $(u_{i+1} - u_i)$ est égale à constante pour $0 \leq i \leq h-1$ alors le vecteur nodal est dit « uniforme », sinon il est dit « non uniforme ».

IV.1.3. Formes possibles d'une surface B-Spline [1,9] :

Une surface B-Spline peut avoir trois formes possibles dans chaque direction:

- Surface ouverte: si le vecteur nodal dans une direction n'a aucune structure particulière alors la surface produite est ouverte dans cette direction et ne passe pas par les points de contrôle $p_{0,0}$, $p_{m,0}$, $p_{0,n}$, $p_{m,n}$.
- Surface pincée: elle est dite pincée si les premiers $P+1$ nœuds sont nuls et les derniers $P+1$ nœuds sont égaux à 1. Si la surface est pincée dans les deux directions, alors la surface passe par les points de contrôle $p_{0,0}$, $p_{m,0}$, $p_{0,n}$, $p_{m,n}$ et de plus elle est tangente en ces points de contrôle aux huit segments du réseau de contrôle.

- Surface fermée: la surface est fermée si les premiers et les derniers points de contrôle sont égaux. Si la surface est fermée dans une direction, alors toutes les courbes isoparamétriques sont fermées dans cette direction.

IV.1.4. Propriété des surfaces B-Spline [2,5] :

Les principales propriétés des surfaces B-Spline sont les suivantes :

- Propriété de l'enveloppe convexe: toute la surface est contenue dans l'enveloppe de son réseau de contrôle.
- Schéma de modification locale : le déplacement d'un point de contrôle n'affecte la surface que localement au voisinage du point déplacé.
- Invariance affine : lorsque une surface B-Spline subit une transformation affine, on peut utiliser les images des points de contrôle pour construire la surface transformée.
- Si $m = p$, $n = q$, et $U = \{0, 0, \dots, 0, 1, 1, \dots, 1\}$, alors la surface B-Spline devient une surface de Bézier.
- Toute courbe isoparamétrique sur la surface B-Spline est une courbe B-Spline définie par un ensemble de points de contrôle.
- La surface est indépendante du système de coordonnées.
- Les deux identités fondamentales doivent être satisfaites (équation II.22)

IV.2. surfaces NURBS [1, 7, 11,12] :

Comme les surfaces B-Spline sont des surface polynomiales, donc ne permettent pas de concevoir les surfaces coniques (sphère, cylindre,...etc.). Donc, il est impératif de passer à la formulation rationnelle des surfaces d'où l'introduction des surfaces NURBS.

Une surface NURBS est définie avec les mêmes paramètres de définition d'une surface B-Spline, sauf qu'à chaque point de contrôle est affecté un poids (voir figure 7).

Une surface NURBS est donnée par :

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (I.23)$$

Où les $w_{i,j}$ sont les poids des point de contrôles $P_{i,j}$ ($0 \leq i \leq m$, $0 \leq j \leq n$), et $N_{i,p}$ et le $N_{j,q}$ sont des fonctions de base B-Spline de degré p et q dans les directions U et V respectivement [2,7].

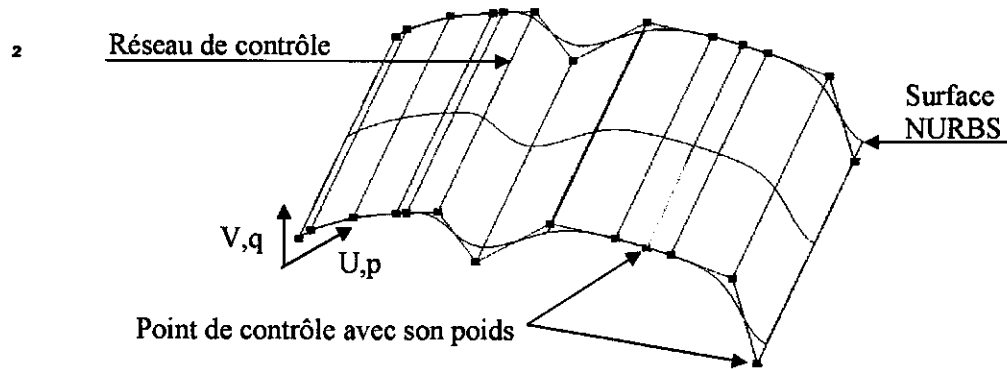


Figure 7 : Paramètres de définition d'une surface NURBS.

Comme pour les surfaces B-Spline, les surfaces NURBS peuvent être ouvertes, fermées ou pincées sous les mêmes conditions.

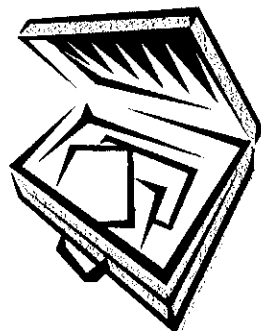
IV.2.1. Propriété des surfaces NURBS :

En plus des propriétés des surfaces B-Spline, les surfaces NURBS ont les propriétés suivantes :

- Si tous les poids sont égaux, la surface NURBS devient une surface B-Spline.
- Une surface NURBS est une surface rationnelle, donc, elle permet de représenter toutes les formes coniques (sphère, ellipsoïde, cylindre, etc.)
- Invariance projective : si une transformation projective est appliquée à une surface NURBS, alors la surface projetée est construite à partir de la projection des points de contrôle.

V. CONCLUSION :

Dans ce chapitre nous avons étudiés les différentes méthodes de conception des surfaces gauches et en particulier les surfaces les plus utilisées dans les logiciels de CFAO à savoir les B-Spline et les NURBS ainsi que leurs importantes propriétés. Pour la suite de notre travail, nous allons considérer les surfaces NURBS en raison de leurs importantes propriétés par rapport aux autres formulations. Dans le chapitre suivant nous passerons à l'étude des machines outils à commande numérique permettant d'usiner ces surfaces.



CHAPITRE 2
MACHINES OUTILS À
COMMANDE NUMERIQUE



I. INTRODUCTION :

Une fois que les surfaces gauches sont conçues, il est nécessaire de choisir les machines outils à utiliser pour les usiner, qui sont en général des fraiseuses, ainsi que l'outillage nécessaire. En raison de la géométrie complexe de ces surfaces, les machines outils utilisées doivent être commandées numériquement. D'où la nécessité de comprendre le mode de fonctionnement de ces machines et leur langage de programmation.

Dans ce chapitre, nous allons présenter les différents composants d'une machine outil à commande numérique et en particulier la fraiseuse à 3 axes, ainsi que la structure générale des programmes d'usinage.

II. GENERALITES SUR LE FRAISAGE [12, 13, 15] :

Usiner une pièce c'est la soumettre à une action d'enlèvement de matière en réalisant les mouvements de coupe nécessaires en utilisant des machines outils. Le choix de la machine dépend de la forme de la pièce et de l'opération d'usinage. Traditionnellement, les machines sont classées en fonction des formes des surfaces à réaliser. Les machines employées dans les ateliers de fabrication mécaniques comprennent en particuliers les tours, les perceuses et les fraiseuses. Mais, dans les ateliers de mécanique, les machines les plus utilisées dans la fabrication des pièces de formes libres (moules, matrices, ...etc.) sont les fraiseuses. Le fraisage, comme son nom l'indique, regroupe les opérations d'usinage pouvant être effectuées sur une fraiseuse et nous pouvant distinguer deux types d'opérations d'usinage.

II.1. Opérations "en bout" :

Cette catégorie regroupe le surfacage et l'usinage 3D à la fraise cylindrique, boule ou torique. Dans ce type de fraisage, l'axe Z de la fraise est perpendiculaire à la surface à usiner (voir figure 1).

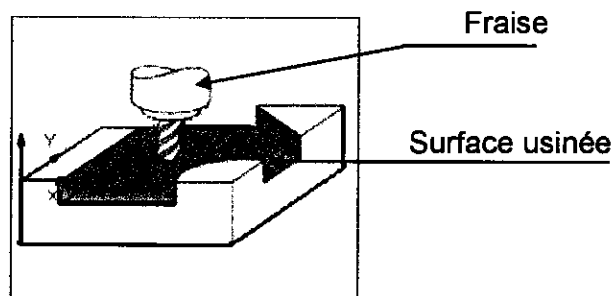


Figure 1 : Fraisage en bout.

II.2. Opérations "en roulant" :

Dans ce type de fraisage, la surface à usiner est tangente à la génératrice de la fraise. C'est donc le profil de la fraise qui va déterminer la forme. Ainsi dans ce type de fraisage, la fraise peut travailler soit en avalant (conseillé pour les pièces minces

ou les opérations de finition) soit en opposition (conseillé pour l'écroûtage ou l'ébauchage). (voir figure 2).

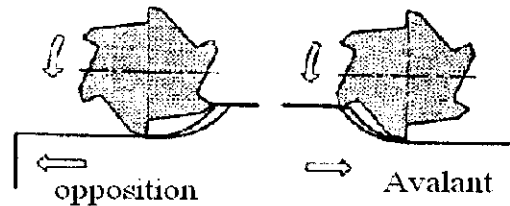


Figure 2 : Fraisage en roulant.

III. MODES D'USINAGE :

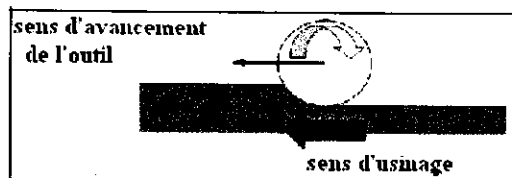
En fraisage, deux modes d'usinage peuvent être utilisés.

III.1. Usinage en avalant :

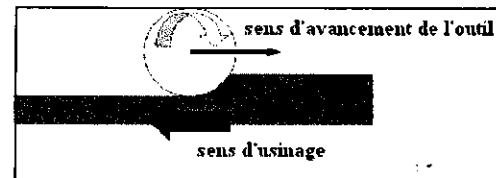
Dans ce type d'usinage, les dents de la fraise sont prises avec la matière avançant dans le sens inverse du sens d'usinage, l'outil attaque la pièce par une épaisseur de copeau maximale et termine le travail avec une épaisseur nulle du copeau (voir figure 3.a).

III.2. Usinage en opposition :

Dans ce type d'usinage, les dents de la fraise sont prises avec la matière avançant dans le même sens d'usinage, l'outil attaque la pièce par une épaisseur du copeau nulle et il termine le travail avec une épaisseur de copeau maximale. (voir figure 3.b).



a. Fraisage en avalant.



b. Fraisage en opposition.

Figure 3 : Modes d'usinage.

IV. MOUVEMENTS GENERATEURS DES FORMES [14]:

Le fraisage s'obtient par les mouvements de coupe et d'avance :

- Mouvement de coupe : c'est le mouvement de rotation donné aux outils d'usinage (fraise).
- Mouvement d'avance : c'est les mouvements de translation relatifs de l'outil par rapport à la pièce selon les trois axes.

V. PARAMETRES DE COUPE [10] :

L'outil suit une trajectoire par rapport à la pièce à usiner, et pour obtenir un travail satisfaisant (bon état de surface usiné, rapidité de l'usinage), on doit fixer les paramètres de la coupe à savoir la vitesse de coupe et la vitesse d'avance (voir figure 4).

V.1. Vitesse de coupe V_c [mm/min] [10, 12] :

C'est la vitesse linéaire du déplacement de l'outil par rapport à la pièce. Cette vitesse est liée principalement à :

- La nature du matériau constituant l'outil.
- La matière constituant la pièce à usiner.

La vitesse de coupe V_c est donnée par :

$$V_c = \frac{\pi * d * N}{1000} \quad (1)$$

Où :

- N : fréquence de rotation en tours par minutes.
- d : diamètre de la fraise en millimètres.

V.2. Détermination de la fréquence de rotation N [tr/mm] :

La vitesse de coupe V_c étant donnée par des tableaux (valeurs empiriques, abaques et autres), il convient de déterminer la fréquence de rotation N par la formule suivante :

$$N = \frac{1000 * V_c}{\pi * d} \quad (2)$$

- V_c : vitesse de coupe en millimètres par minute.
- d : diamètre de la fraise en mm.

V.3. Vitesse d'avance V_f [mm/min] :

L'avance de l'outil s'exprime par la vitesse de déplacement de l'outil sur la trajectoire d'usinage. La vitesse d'avance est donnée par :

$$V_f = fz * Z * N \quad (3)$$

Avec :

- f_z : avance par dent.
- N : fréquence de rotation.
- Z : nombre de dents de la fraise

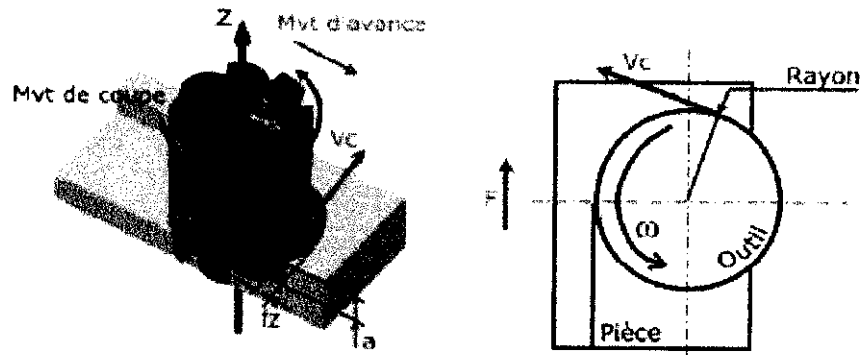


Figure 4 : Paramètres de coupe en fraisage.

VI. DEFINITION D'UNE MACHINE-OUTIL A COMMANDE NUMERIQUE « MOCN »

VI.1. Définition [12, 14] :

Une machine outil à commande numérique « MOCN » (voir figure 5) est une machine d'usinage à cycle automatique programmable, commandée par des "consignes" numériques fournies par un ordinateur. Elle assure l'obtention d'une surface par enlèvement de la matière en réalisant les mouvements de coupe et d'avance nécessaires décrits dans un programme.

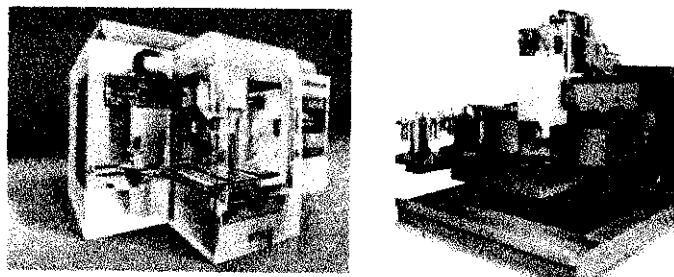


Figure 5 : Machines outils à commande numérique.

VI.2. Architecture des fraiseuses à commande numérique [12] :

Ce type de machine se compose de deux parties complémentaires (voir figure 6) :

VI.2.1 Partie opérative :

Elle est composée d'outils qui agissent directement sur le produit final à réaliser et comporte les éléments suivants :

- Les axes de la machine : assurent la mise en position de l'outil par rapport à la pièce et les mouvements d'avance.
- La broche : elle porte l'outil (fraise) et réalise le mouvement de coupe par la mise en rotation de l'outil.
- Le bâti : c'est la plateforme de la machine qui permet d'assurer le lien entre ces systèmes et de supporter la chaleur et le guidage des axes de mouvements.
- Le porte outils : porte le montage des outils d'usinage.

VI.2.2 Partie commande :

Elle permet d'envoyer des ordres à travers des commandes numériques. Parmi les composant électroniques de cette partie :

- Un système de contrôle-commande : qui permet le suivi automatique du programme de la machine.
- Le pupitre de commande.
- L'armoire électronique.

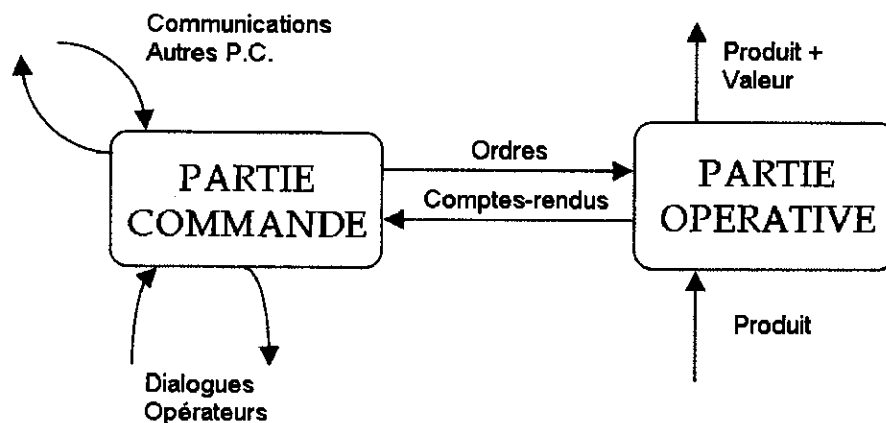


Figure 6 : Partie commande et partie opérative d'une MOCN.

La structure de la machine est représentée par un système bouclé d'éléments comme le montre la figure suivante [11].

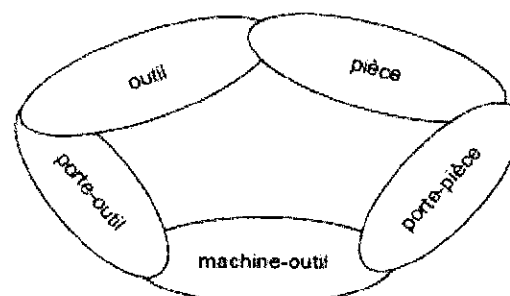


Figure 7 : Modèle de la structure d'une machine outil.

VII. AXES DE DEPLACEMENT : [12-15]

Les déplacements de l'outil ou du porte pièce s'effectuent par combinaisons de translations et/ou de rotations. Chaque mouvement élémentaire (axe) est défini par une lettre affectée du signe « + » ou « - » indiquant le sens du déplacement. Les mouvements définis par la mise en position d'une pièce placée sur la machine se fait par rapport à un repère de 3 axes de translation X, Y et Z.

VII.1. Mouvement de translation [15] :

Les mouvements de translation sont définis par rapport à trois axes perpendiculaires:

- **Axe Z** : parallèle à la broche principale .C'est l'axe du système normal.
- **Axe X** : correspond au plus grand déplacement et il est perpendiculaire à l'axe Z. Pour les fraiseuses :
 - Si l'axe Z est horizontal, le sens X positif est dirigé vers la droite lorsqu'on regarde de la broche principale vers la pièce.
 - Si l'axe Z est vertical, le sens X positif est dirigé vers la droite, lorsqu'on regarde de la broche principale vers le montant de la machine.
- **Axe Y** : forme avec les axes X et Z un trièdre de sens direct. La règle de trois doigts de la main droite permet de trouver facilement l'orientation des axes X, Y et Z. (voir figure 8)

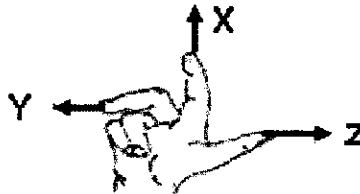


Figure 8 : Orientation des axes X, Y et Z.

VII.2. Mouvement de rotation :

Les mouvements de rotation effectués respectivement autour des axes parallèles à X, Y et Z sont les rotations A, B et C (voir figure 9).

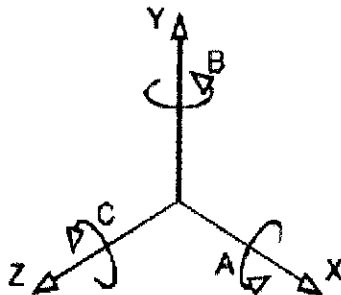


Figure 9 : Axes linéaires et axes de rotation.

En fermant la main droite, le pouce indique le sens positif d'un axe de rotation (voir figure 10).

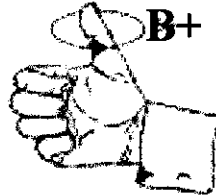


Figure 10 : Sens positif d'un axe de rotation.

VII.3 Types d'axes d'Une MOCN : [13]

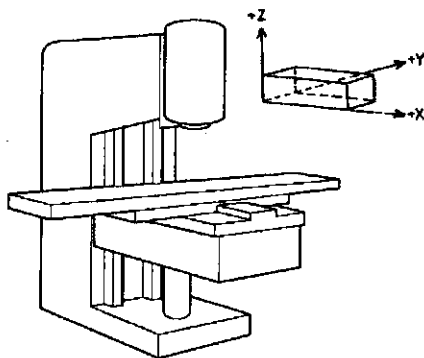
Pour les axes de la machine, nous avons deux types d'axes :

- Demi axe numérique : c'est est un axe de déplacement qui atteint un ensemble fini de positions ou un axe de déplacement asservi en position ou en vitesse.
- Axe numérique : c'est un axe de déplacement pour lequel une infinité de positions peuvent être atteintes à la résolution de positionnement près ou un axe de déplacement asservi en position et en vitesse.

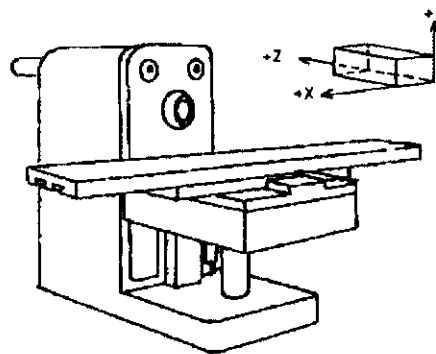
VIII. TYPES DE FRAISEUSES :

On distingue deux types de fraiseuses selon la position des axes :

- Fraiseuse verticale : la broche est verticale (voir figure 11.a).
- Fraiseuse horizontale : la broche est horizontale (voir figure 11.b).



a. Fraiseuse verticale.



b. Fraiseuse horizontale.

Figure 11: Types de fraiseuses

IX. PROGRAMMATION DES MOCN [12-14, 16, 17] :

L'usinage sur des MOCN, nécessite la génération d'un programme d'usinage. Un programme d'usinage est constitué de lignes d'instructions appelées "blocs". Un bloc

correspond aux instructions relatives à une séquence d'usinage. Chaque bloc est constitué d'un groupe de mots (voir figure 12).

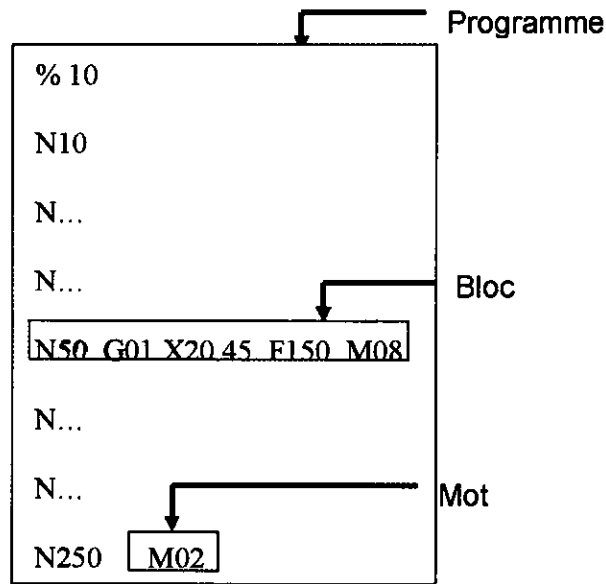


Figure 12 : Format général d'un programme d'usinage

IX.1. programme d'usinage :

Un programme est une suite d'instructions écrites dans un langage codé propre à la commande numérique (le plus utilisé est le code ISO : International Standardization Organization) décrivant les opérations d'usinage.

IX.1.1. Format d'un mot :

Un mot est un ensemble de caractères composé d'une adresse suivie de chiffres constituant une information (voir figure 13).

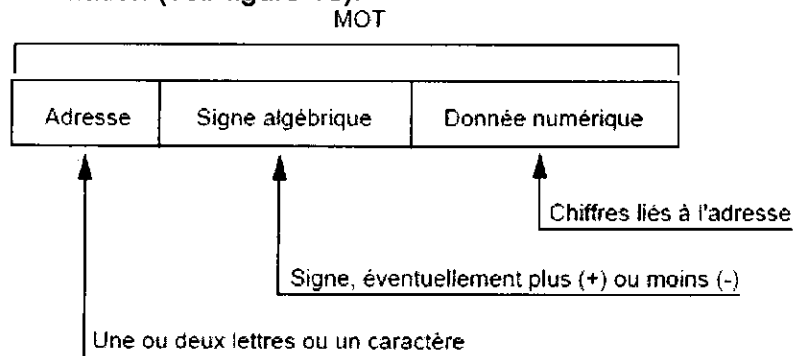


Figure 13 : Format général d'un mot.

Un mot définit une instruction ou données à transmettre au système de commande. Il peut définir une dimension ou une fonction.

Un mot peut être une fonction ou un déplacement suivant un axe :

- **G...** : fonctions préparatoires : qui préparent la logique à une action donnée ou à un type de calcul.
- **F...** : fonctions vitesse d'avance ("Feedrate" = avance).
- **S...** : fonctions vitesse de broche ("Speed" = vitesse).
- **T...** : fonctions outils ("Tools" = outils).
- **M...** : fonctions auxiliaires ("Miscellaneous" = varié, divers) : qui déterminent les mouvements de la broche, la sélection de vitesse, l'arrosage, etc.
- **X, Y, Z.** : translation suivant les axes X, Y et Z respectivement.
- **A, B, C.** : angles de rotation autour des axes X, Y et Z respectivement.

IX.1.2. Format d'un bloc :

Un bloc (ou séquence) définit une ligne d'instructions composée de mots codés à transmettre au système de commande (voir figure 14). Le format de bloc définit la syntaxe des mots de fonctions et de dimensions composant chaque bloc de programmation.

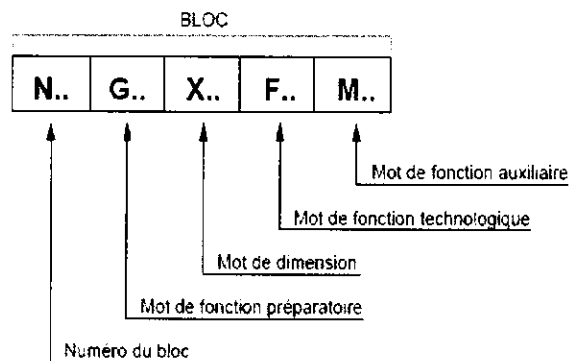


Figure 14 : Format général d'un bloc.

IX.2. Caractéristique d'un programme [13, 17] :

Un programme d'usinage est caractérisé par:

- Des caractères obligatoires de début et de fin.
- Exécute dans l'ordre d'écriture des blocs situés entre les caractères de début et de fin de programme.
- Avoir une numération de blocs.

IX.3. Structure d'un programme ISO :

Quelque soit le langage de programmation utilisé pour l'écriture des programmes pièces, le seul langage compréhensible par la machine est le langage ISO. Le

passage d'un langage de haut niveau au langage ISO est possible en utilisant un logiciel de traduction. La programmation peut être faite de deux manières : manuelle ou assistée par l'utilisation de la méthode conversationnelle ou un logiciel de F.A.O.

Un programme ISO est structuré comme suit (voir figure 15) :

- Début de programme : caractère % suivi du numéro de programme et éventuellement d'un commentaire entre parenthèses,
- Ensemble des blocs à exécuter (programme effectif),
- Fin de programme : code M02,
- Fin de chargement de programme: caractère XOFF.

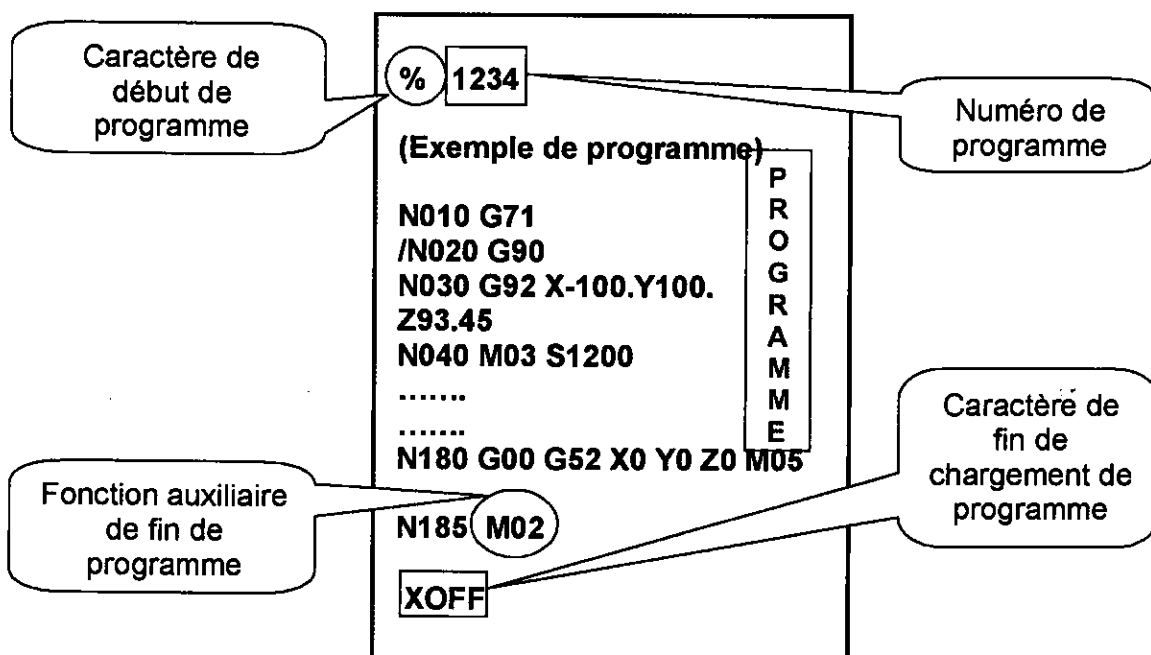
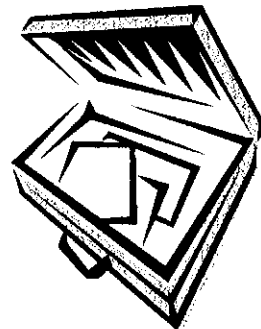


Figure 15 : Structure d'un programme ISO.

X. CONCLUSION :

Dans ce chapitre, nous avons vu les paramètres de coupe nécessaires à l'opération d'usinage en fraisage, la constitution des fraiseuses à commande numérique et leurs modes de fonctionnement ainsi que la syntaxe générale des programmes d'usinage. Dans notre travail, nous allons considérer uniquement les fraiseuses à commande numérique à 3 axes pour l'usinage en finition des surfaces gauches.



**CHAPITRE 3
USINAGE DES SURFACES
GAUCHES EN FINITION**

I. INTRODUCTION :

Après l'obtention de la forme voulue de la pièce et par conséquent le modèle CAO, l'opération logique suivante est la planification des opérations de l'usinage. L'usinage d'une pièce en finition consiste à déterminer la trajectoire d'outil permettant de satisfaire les différentes contraintes de conception et de fabrication. Pour ce la, pour assurer l'obtention d'une pièce correcte, il faut que le processus de génération des trajets d'outils respecte quelques contraintes que nous allons décrire dans ce chapitre. La figure 1 montre la succession des différentes étapes permettant de passer de la conception jusqu'à l'obtention d'une pièce réelle [18].

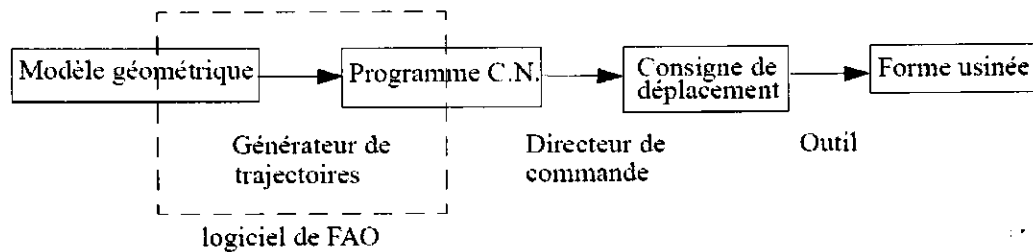


Figure 1 : Processus de réalisation d'un usinage.

I. CHOIX DES FORMES D'OUTILS [20-22] :

Pour l'usinage d'une surface de forme libre, il n'existe pas de formes d'outils qui permettent l'obtention de la forme désirée exactement en un seul mouvement élémentaire. Donc, une surface de forme libre est générée par un enchaînement de déplacements d'outil avec la condition qu'il doit être constamment tangent à la surface à usiner en chaque point de contact entre l'outil et la surface. Le choix de l'outil est le résultat d'un compromis entre la rigidité de l'outil, la cinématique de la machine et la forme de la pièce à usiner.

Pour usiner des surfaces gauches, trois formes d'outils peuvent être utilisés :

- Torique : l'arête coupante prend la forme d'un tore. Cet outil présente l'avantage d'avoir un petit rayon de bout de la fraise « r » afin d'usiner les petites parties concaves et un grand rayon « R » de corps afin d'enlever plus de matière (voir figure 2.a).
- Hémisphérique : la partie active est en forme d'une demi sphère de rayon R (voir figure 2.b). Cet outil possède une vitesse de coupe très faible sur les zones proches de l'axe d'outil et nulle au centre ce qui écrase la matière et par conséquent un mauvais état de surface.
- Cylindrique : la partie active est en forme d'un cylindre de rayon R (voir figure 2.c). Cet outil permet l'enlèvement de la matière avec la même vitesse de coupe à chaque point de contact.

Dans le cas de l'usinage en finition sur des fraiseuses à 3 axes, les fraises hémisphériques sont les plus utilisées, mais en ébauche, c'est les outils cylindriques et toriques qui sont utilisés.



Figure 2 : Types de fraises.

III. PARAMETRES DE PASSE [18, 22, 23] :

Pour usiner une surface gauche, il faut choisir un mode de balayage de l'outil (stratégies d'usinage) qui caractérise la forme de la trajectoire que doit suivre l'outil. A la direction de balayage, il faut associer deux paramètres d'usinage :

III.1. Pas longitudinal :

Il est calculé en respectant un critère de tolérance d'usinage qui est la valeur maximale de l'erreur de flèche entre le segment reliant deux positions successives de l'outil et la surface nominale (voir figure 3). Ce pas permet de garantir le respect de la position relative de l'outil par rapport à la surface.

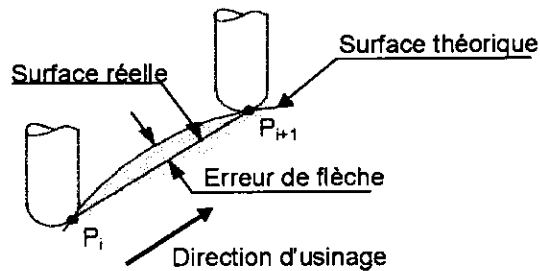


Figure 3 : Erreur de flèche.

III.2. Pas transversal :

Afin d'usiner la totalité des surfaces, les trajets sont juxtaposés selon un pas transversal calculé en respectant un critère de distance maximale entre passes ou un critère de hauteur de crête (voir figure 4). De plus la distance maximale entre passes peut être calculée entre les surfaces de guidage ou sur la surface nominale.

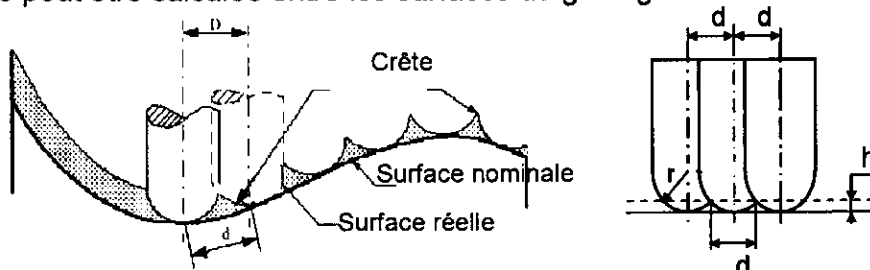


Figure 4 : Distance maximale entre passes et hauteur de crête.

IV. PROBLEMES D'INTERFERENCES ET DE COLLISIONS [7, 21] :

Parmi les problèmes qui se posent lors de la génération des trajets d'usinage des surfaces gauches sont les interférences et les collisions.

IV.1. Problème d'interférence (interférence locale) :

Le phénomène d'interférences entre l'outil et la surface se produit dans le cas où l'outil usine une partie concave dont le rayon de courbure est inférieure à celui de l'outil (voir figure 5.a). Donc, pour éviter les interférences, il faut choisir un rayon d'outil qui soit plus petit ou égal au plus petit rayon de courbure de la surface.

IV.2. Problème de collision (interférence globale) :

En usinage sur des fraiseuses à commande numériques verticales à 3 axes, l'axe de la fraise hémisphérique est dans une position verticale, et par conséquent, tous les points de contact entre l'outil et la surface dont l'angle α entre l'axe de l'outil \vec{u} et la normal \vec{n} est supérieur à 90° ne peuvent être usinés (points en contre dépouille). Ces points sont des points de collision (voir figure 5.b).

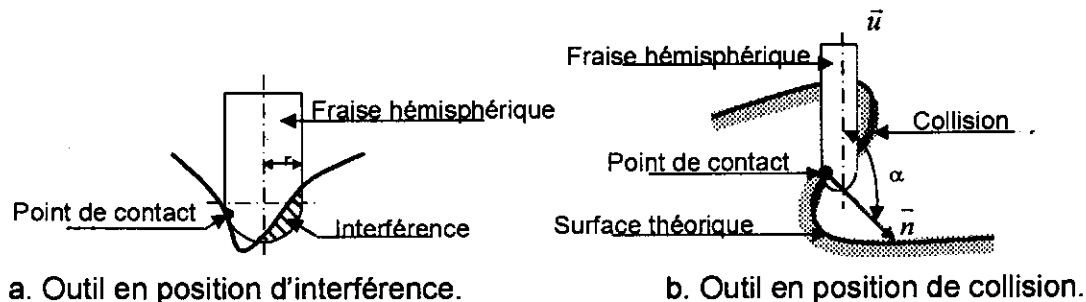


Figure 5 : Outil en position d'interférence et en collision

Ce type d'interférence concerne le corps de l'outil ainsi que le porte outil et l'environnement d'usinage et dépend de la longueur de l'outil ainsi que de la forme de la pièce (voir figure 6).

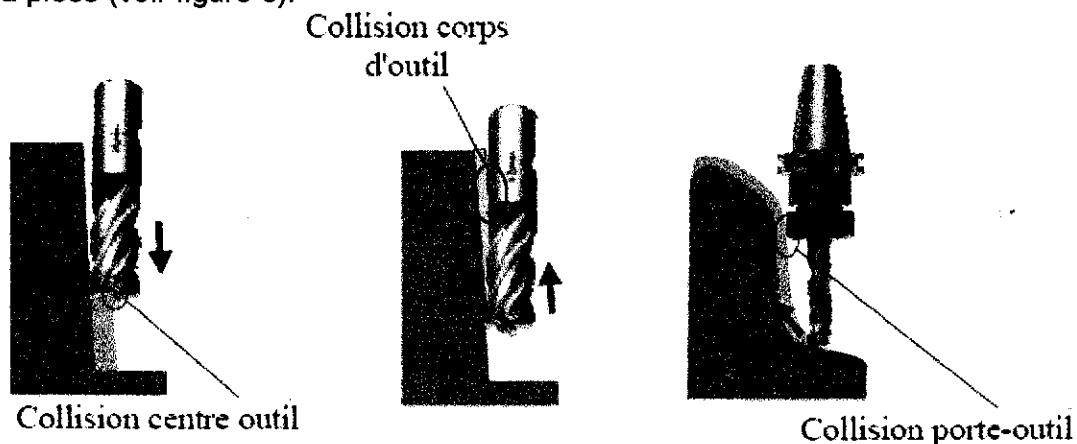


Figure 6 : Différents cas de collisions.

V. METHODES DE GENERATION DE CHEMIN D'OUTIL [18] :

Pour usiner une surface gauche, il faut déterminer le chemin d'outil (trajet d'usinage). L'objectif principal de la génération de chemin d'outil est de calculer une suite ordonnée de points (points CL –point extrémité de l'outil-) que l'outil doit suivre. Pour générer le chemin d'outil, différentes méthodes peuvent être utilisées.

V.1. Méthode du copiage informatique :

C'est une méthode de passage par le centre d'outil. L'outil glisse le long d'une droite jusqu'à ce qu'il touche la surface à usiner (voir figure 7). Ce glissement est traduit par un algorithme de balayage d'une direction.

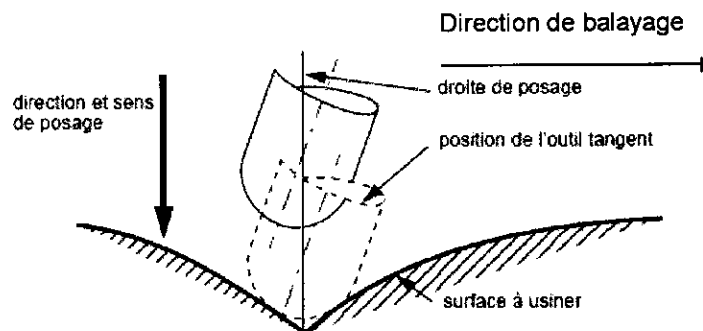


Figure 7 : Copiage informatique.

V.2. Méthode basé sur le point de contact CC (CC-Based Method) [20, 23]

Dans la méthode du point de contact CC (CC- Based Method), le chemin d'outil est généré à partir du modèle CAO de la surface à usiner où chaque point de contact entre l'outil et la surface est converti en un point d'extrémité de l'outil (point CL). Pour cette méthode, la génération du chemin d'outil passe par les étapes suivantes

- Calcul de la normale à la surface au point de contact considéré,
- Calcul de la position du centre de l'outil,
- Calcul de la position de l'extrémité de l'outil (point CL).

Pour trouver les positions successives de l'outil, nécessaires et suffisantes permettant d'usiner la surface, les paramètres suivants sont définis (voir figure 8) [19] :

- CC (Cutter Contact) : point de contact entre l'outil et la fraise,
- CE : point centre de l'outil,
- CL (Cutter Location) : point extrémité de l'outil,
- Le vecteur \vec{n} : vecteur normal à la surface au point de contact,
- Le vecteur \vec{u} : vecteur d'orientation de l'axe de l'outil,
- r : rayon de l'outil hémisphérique et le petit rayon de l'outil torique,

➤ R le grand rayon de l'outil torique.

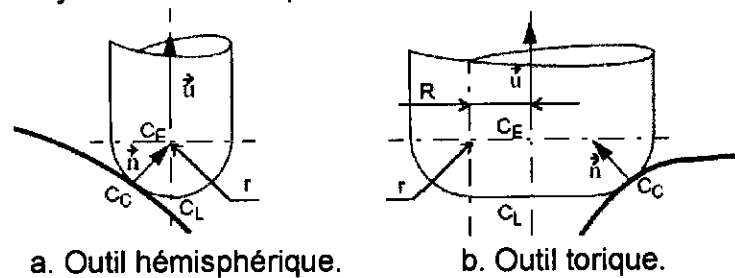


Figure 8 : Position de la fraise hémisphérique par rapport à la surface.

Les positions des points de l'outil hémisphérique sont données par les équations suivantes [22] :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} \end{cases} \quad (\text{III.1})$$

Les positions des points de l'outil torique sont données par les équations suivantes [8] :

$$\vec{k} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|} \quad (\text{III.2})$$

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} + R \cdot \frac{\vec{k} \wedge \vec{u}}{\|\vec{v} \wedge \vec{u}\|} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} + R \cdot \frac{\vec{k} \wedge \vec{u}}{\|\vec{v} \wedge \vec{u}\|} \end{cases} \quad (\text{III.3})$$

V.3. Méthode basée sur le point d'extrémité CL (CL-Based Method) [23, 25-27]

Dans la méthode des points d'extrémités de l'outil (CL-Based Method), la surface CL est utilisée comme surface de génération de chemin. Une surface CL est produite en déplaçant la surface nominale le long de la normale avec une certaine distance.

V.3.1. Méthode de calcul par offset de la forme [19,20] :

Cette méthode est basée sur le calcul de l'offset de la surface à usiner avec une valeur égale à la valeur du rayon de l'outil hémisphérique utilisé. La position d'outil tangente à la surface usinée appartient à la surface offset. L'avantage de cette méthode est qu'elle est simple et efficace, ainsi elle évite les interférences locales, mais elle ne peut être appliquée que pour des outils hémisphériques (voir figure 9).

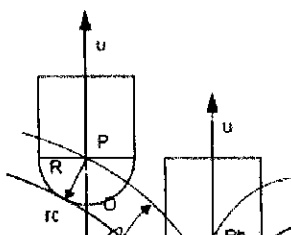


Figure 9 : Surface offset d'une surface.

Deux problèmes se posent lors du calcul de l'offset de la surface [20] :

- La surface offset n'est pas toujours générée suivant le même format de description que la surface nominale;
- la surface offset peut interférer avec elle-même et générer des boucles d'auto intersection dans des zones à faible rayon de courbure. Une position de l'outil tangente en un point de la surface localisée dans ces boucles présente une interférence avec la surface à usiner. Il faut donc mettre en oeuvre une méthode de correction de la trajectoire d'usinage.

V.3.2. Méthode de l'offset inverse (ITO) [20, 23] :

La méthode de l'offset inverse est une méthode simple et efficace et permet l'obtention des positions centre outil de sorte que l'outil soit tangent à la surface et cela quelque soit la forme de l'outil utilisé.

V.3.2.1 : Différents outils utilisés [28]:

La méthode de l'offset inverse (Inverse Tool Offset (ITO)) est plus générale, elle permet l'obtention de la surface centre outil (point de référence de l'outil) pour n'importe quelle forme d'outil (hémisphérique, cylindrique ou torique) (voir figure 10).

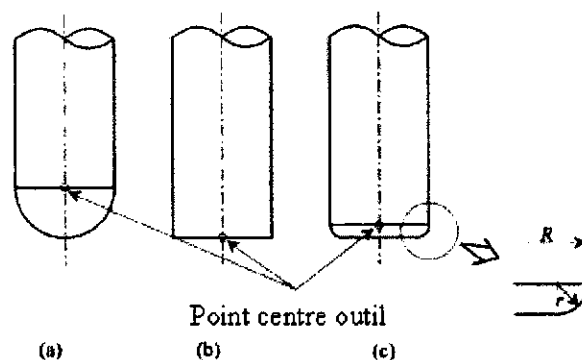


Figure 10 : Outils utilisés dans la méthode ITO.

Si le centre d'un outil appartient à la surface offset, l'outil usine la surface (voir figure 11.a), et si le centre d'outil appartient à la surface, il usine une surface offset

(voir figure 11.b) de la valeur du rayon de l'outil. Donc, la surface offset est l'enveloppe des différentes positions de l'outil inverse.

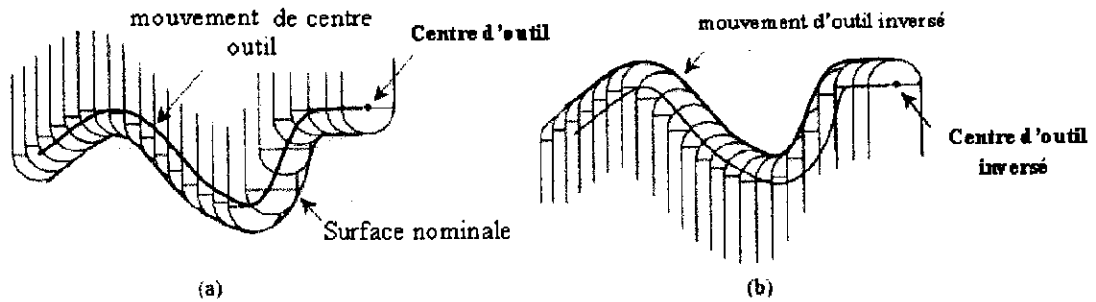
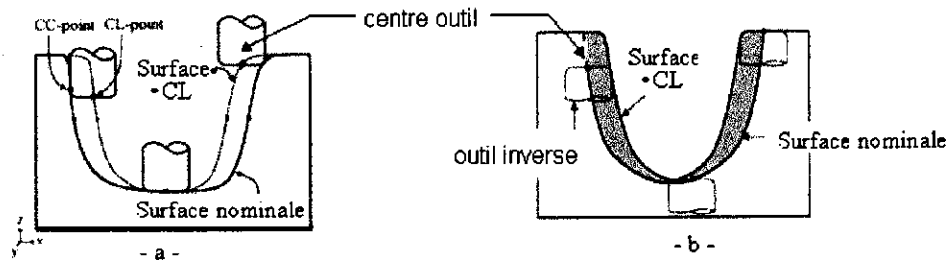
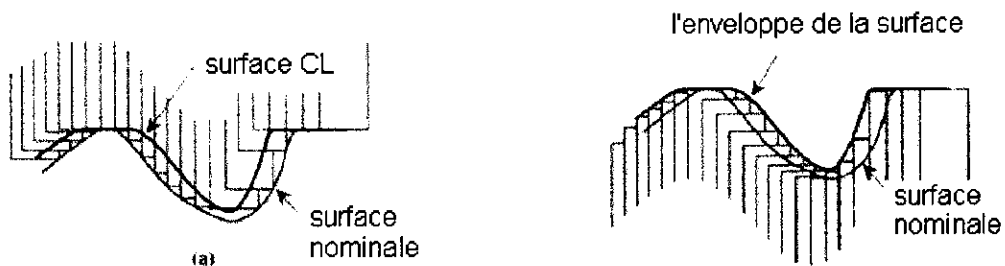


Figure 11 : Génération de la surface offset par la méthode ITO.

La figure 12 montre la génération de la surface CL pour les deux formes d'outil torique (voir figure 12.1) et cylindrique (voir figure 12.2)



1. Cas : outil torique.



2. Cas : outil cylindrique.

Figure 12 : Surface offset par la méthode ITO pour des outils toriques et cylindriques.

V.3.2.2 Démarche de calcul :

La démarche de calcul de la surface offset (CL) lors de l'usinage avec un outil hémisphérique est la suivante :

- On considère une surface discrétisée où en chaque point on fait passer une droite verticale (voir figure 13.a).
- On centre en chaque point une sphère de même rayon que l'outil (voir figure 13.b). Dans le cas d'un outil torique, c'est un tore qui est centré en chaque point, tandis que pour un outil cylindrique c'est le centre du cylindre.

- On calcule les points d'intersection entre chaque droite et une sphère. (voir figure 13.c).
- La surface offset est enfin obtenue par la sélection des points d'intersection les plus hauts sur chaque droite (voir figure 13.d).

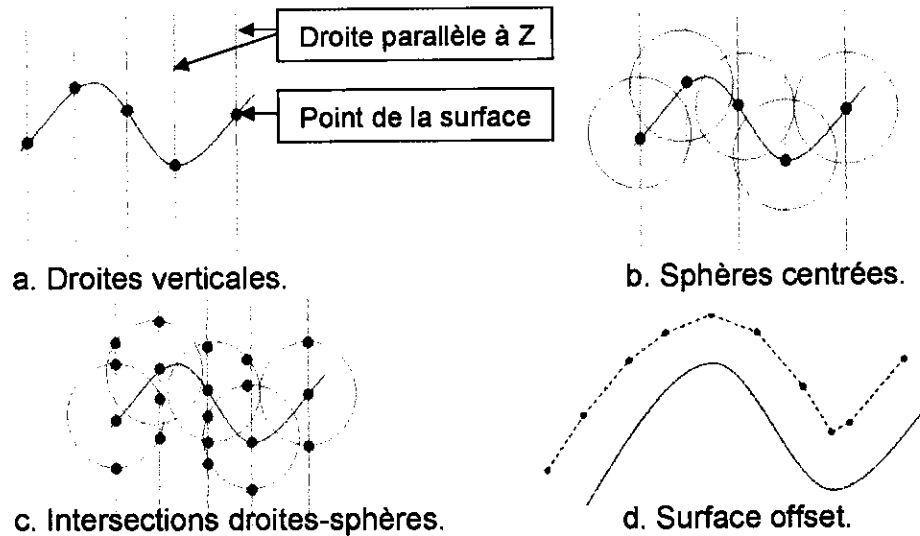


Figure 13 : Méthode ITO pour un outil hémisphérique.

VI. PRINCIPALES STRATEGIES D'USINAGE DES SURFACES GAUCHES [19,20]

Lors de l'usinage des surfaces gauches, l'outil suit des courbes 3D qui représentent le trajet d'usinage pouvant être déterminé par différentes méthodes. Les deux principales méthodes sont :

- Les courbes isoparamétriques.
- Les plans parallèles.

VI.1. Courbes isoparamétriques [19,21] :

Cette méthode est basée sur les courbes isoparamétriques des surfaces à usiner suivant \vec{u} ou suivant \vec{v} ou sur des portions de courbes isoparamétriques (voir figure 14). Le trajet d'outil est planifié dans le plan paramétrique est en suite il est transformé vers l'espace 3D

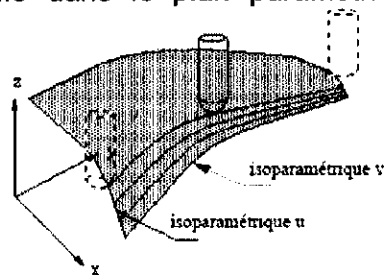


Figure 14 : Usinage par des courbes isoparamétriques.

Plusieurs modes de balayage peuvent être utilisés pour usiner des surfaces gauches en utilisant les courbes isoparamétriques où le choix d'un mode dépend de

la forme de la surface, du temps d'usinage et des possibilités d'accès. Les différents modes d'usinage selon les courbes isoparamétriques sont illustrés par la figure 15.

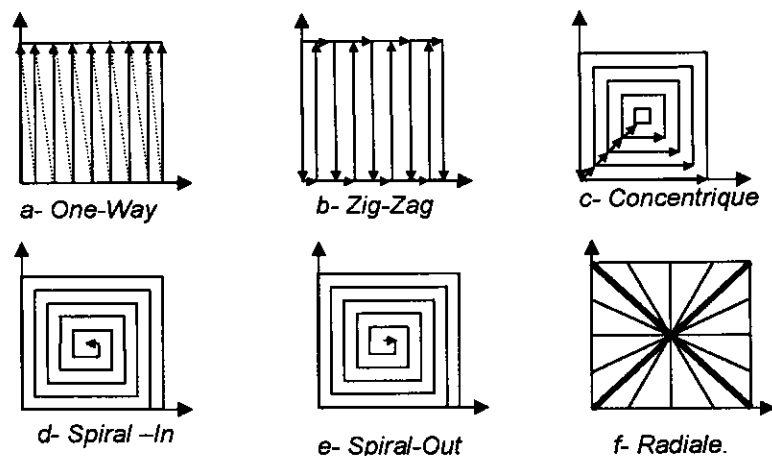


Figure 15 : Modes de balayage suivant les courbes isoparamétriques.

VI.2. Plans parallèles :

VI.2.1. Plans verticaux [18,19, 21] :

Cette stratégie d'usinage permet la construction des plans parallèles à l'axe Z, formant n'importe quel angle avec le plan (XY). La trajectoire est générée à partir des courbes d'intersection de la surface à usiner avec ces plans parallèles (voir figure 16). Pour cette stratégie, il est possible d'usiner en One-Way ou en Zig-Zag.

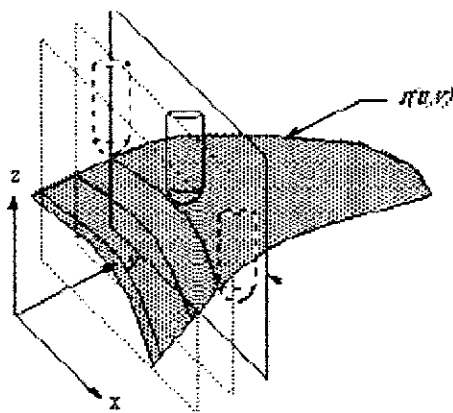


Figure 16 : Méthodes des plans verticaux.

VI.2.2. Plans horizontaux (Z-CONSTANT) [21 ,22] :

Ce type de trajectoire est obtenu par l'intersection de la surface à usiner avec des plans horizontaux. Ces intersections sont appelées « des contours » et les courbes obtenues sont appelées « courbes de niveau » (voir figure 17).

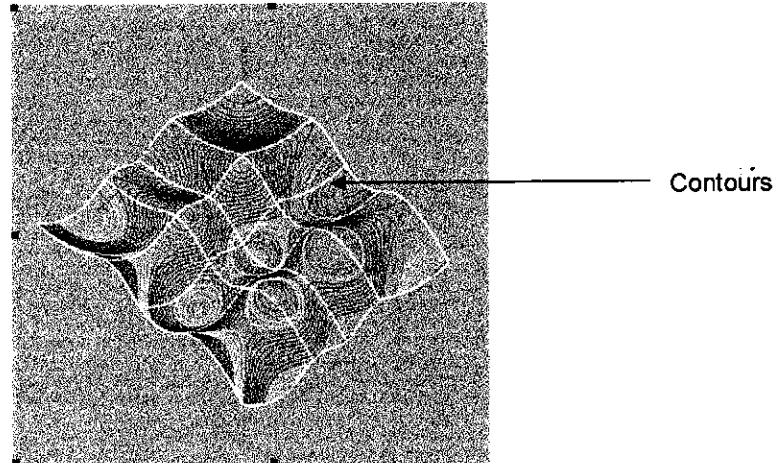
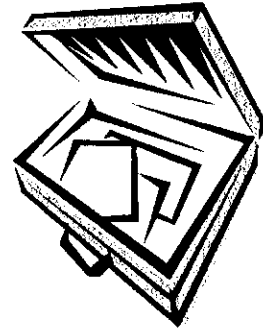


Figure 17 : Stratégie d'usinage par plans horizontaux (Z-Constant).

VII. CONCLUSION :

Dans ce chapitre, nous avons présentés les différentes formes d'outils utilisés dans l'usinage des surfaces gauches, les problèmes d'interférence et de collision, les critères à respecter afin d'obtenir une surface usinée qui s'approche plus de la surface théorique ainsi que les méthodes de génération de chemin d'outil. Dans le chapitre suivant, nous allons étudier les méthodes d'approximation des modèles CAO et le calcul des intersections entre une droite et des objets 3D.



CHAPITRE 4
TRIANGULATION DES
SURFACES ET CALCUL DES
INTERSECTIONS



I. INTRODUCTION :

L'usinage des surfaces gauches sur des MOCN est une pratique courante dans beaucoup d'industries telles que dans la conception des carrosseries des voitures, les coques des avions et d'autres applications où les surfaces douces sont nécessaires pour réduire la résistance de l'air ou l'eau ou simplement pour des raisons esthétiques.

Les trajets d'outils nécessaires à l'usinage des surfaces gauches sont généralement déterminés par l'intersection de la surface à usiner avec un ensemble de plans (plans horizontaux ou plans verticaux ou même les deux types de plans). Il est facile de déterminer ces points d'intersection si nous avons une surface simple, ce qui n'est pas le cas avec les surfaces gauches qui possèdent des formes géométriques très complexes et par conséquent la détermination de ces points devient plus difficile et plus complexe. Afin de simplifier le calcul de ces points, il faut procéder à l'approximation de la surface à usiner par des modèles géométriques simples.

Dans ce chapitre, nous allons voir la triangulation qui est une des méthodes d'approximation des modèles CAO des surfaces par un ensemble de triangles.

II. TRIANGULATION [29] :

Une surface est généralement approximée par des éléments géométriques simples tels que les quadrilatères et les triangles. Cependant, le triangle est l'élément géométrique le plus utilisé pour sa simplicité et l'approximation par cet élément est appelée triangulation. La figure 1 représente le modèle CAO d'une surface et sa triangulation.

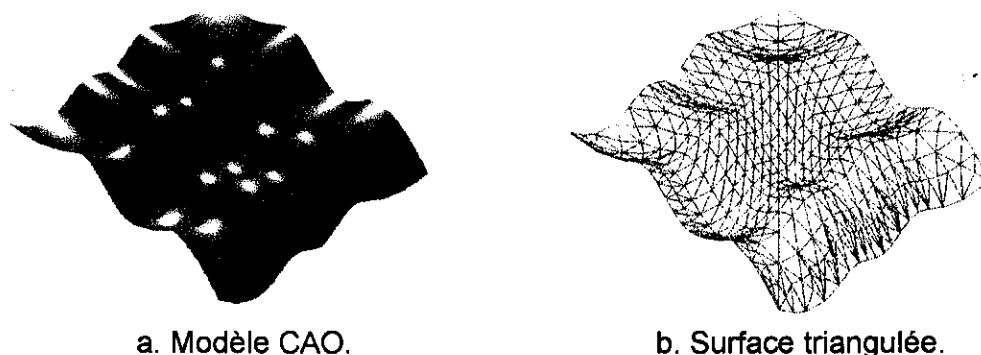


Figure 1 : Triangulation d'une surface.

La triangulation est un problème de la géométrie informatique dont l'objectif est de discrétiser un objet (en 3D) et de l'approximer par un ensemble de triangles. Le problème qui se pose est de trouver la bonne triangulation possible sans avoir une sur qualité de l'approximation. Donc, il s'agit de trouver le nombre minimum de triangles les moins éloignés possibles par rapport à la surface nominale (bonne triangulation) (voir figure 2).

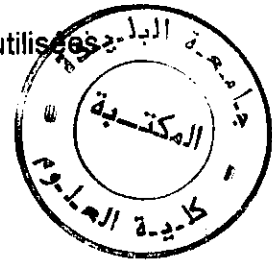


Figure 2 : Choix de la bonne triangulation.

II.1. Méthodes de triangulation :

Pour trianguler une surface, deux méthodes peuvent être utilisées.

- Triangulation uniforme.
- Triangulation adaptative.

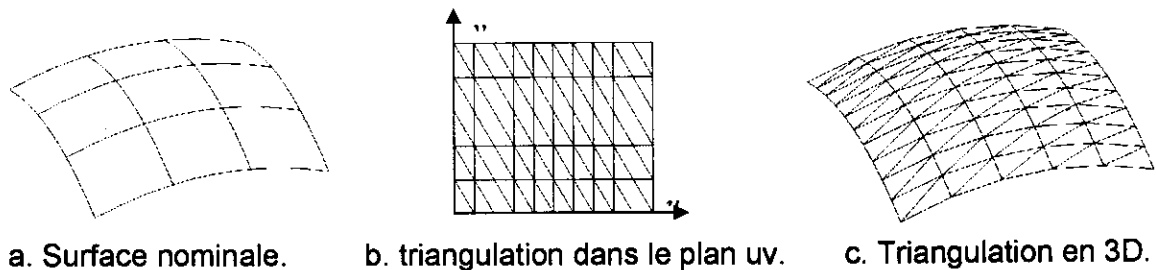


II.1.1. Triangulation uniforme [1] :

Dans cette triangulation, le nombre de triangles, pour chaque surface, sont fixés à l'avance suivant la direction u et la direction v dans le plan paramétrique. Ensuite, cette information est utilisée pour transformer la triangulation du plan paramétrique vers l'espace 3D.

Avec cette méthode de subdivision, la triangulation est rapide, mais pour avoir une bonne approximation il faut choisir un nombre important de triangles dans chaque direction, ce qui augmente l'espace mémoire nécessaire au stockage des différentes informations de chaque triangle et augmente le temps de calcul des intersections.

La figure 3 montre la triangulation uniforme d'une surface nominale théorique.



a. Surface nominale. b. triangulation dans le plan uv. c. Triangulation en 3D.

Figure 3 : Triangulation uniforme d'une surface nominale.

II.1.2. Triangulation adaptative [1,32] :

Pour cette méthode, nous cherchons à adapter dynamiquement la triangulation à partir d'un ensemble de critères fixés par l'utilisateur. C'est la méthode la plus utilisée car elle permet de réduire le nombre de triangles générés et par conséquent le temps de calcul d'intersections. De plus elle s'adapte mieux aux variations géométriques de la forme de la surface.

La triangulation adaptative nécessite la spécification des critères d'approximation de la surface afin que l'ensemble des triangles donne une bonne représentation de la surface théorique. Les conditions à respecter pour chaque triangle sont les suivantes :

- La longueur maximale de chaque côté du triangle doit être inférieure à une tolérance « d1 » spécifiée par l'utilisateur. Donc, trois vérifications sont nécessaires (voir figure 4.a).
- La distance entre le milieu de chaque côté du triangle et la surface nominale doit être inférieure à une tolérance « d2 » spécifiée par l'utilisateur. Donc, trois vérifications sont nécessaires (voir figure 4.b).
- La distance entre le centre de gravité du triangle et la surface nominale doit être inférieure à une tolérance « d3 » spécifiée par l'utilisateur. Donc, une vérification est nécessaire (voir figure 4.c).

Donc, pour trianguler une surface donnée, nous avons sept conditions à vérifier pour chaque triangle généré. Si une des conditions n'est pas vérifiée, le triangle est subdivisé d'une manière adaptative jusqu'à la vérification de toutes les conditions.

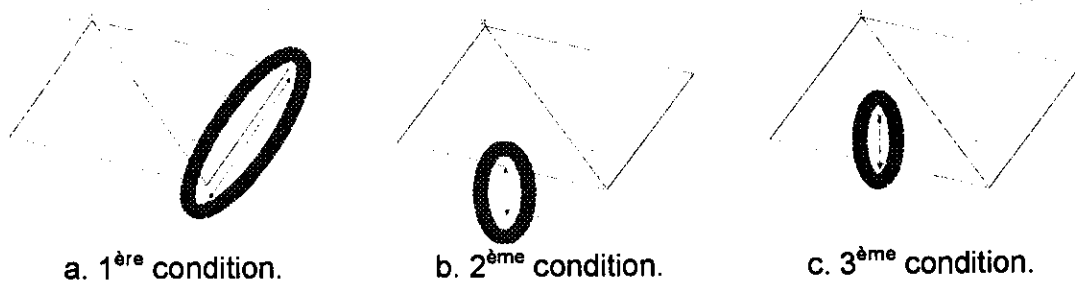


Figure 4 : Critères de la triangulation adaptative.

Le triangle est subdivisé dans les cas suivants (voir figure 5) :

- Si un coté ne vérifie pas une condition (« d1 » ou « d2 »), le triangle est subdivisé en deux triangles (voir figure 5.a).
- Si deux cotés ne vérifient pas une condition (« d1 » ou « d2 »), le triangle est subdivisé en trois triangles (voir figure 5.b).
- Si trois cotés ne vérifient pas une condition (« d1 » ou « d2 »), le triangle est subdivisé en quatre triangles (voir figure 5.c).
- Si la distance entre le centre de gravité du triangle et la surface est plus grande que « d3 », le triangle est subdivisée en trois triangles (voir figure 5.d).

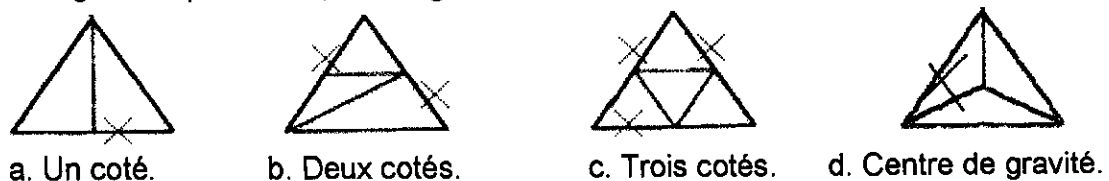
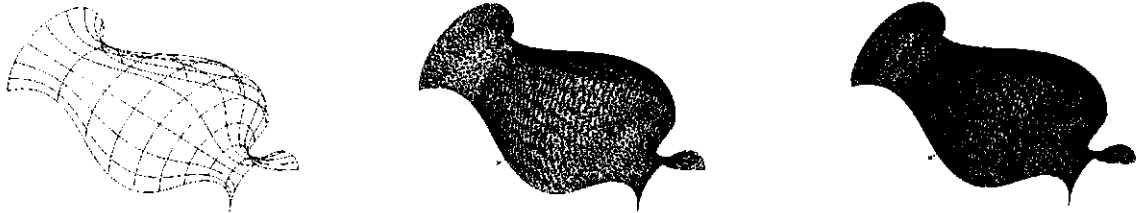


Figure 5: Différents cas de subdivision d'un triangle.

A la fin de la triangulation, nous obtenons un ensemble de triangles qui donnent une bonne approximation de la surface où tous les sommets des triangles appartiennent à la surface nominale. Les triangles générés seront utilisés dans l'étape de calcul des intersections.

La figure 6 montre une surface nominale et sa triangulation adaptative pour deux cas différents :



a. Surface nominale.

b. $d_1=d_2=d_3=2\text{mm}$.c. $d_1=d_2=e=3\text{mm}$.

Figure 6 : Triangulation adaptative d'une surface nominale.

III. INTERSECTIONS :

III.1. Cas d'intersection d'un triangle et un plan dans l'espace 3D [30] :

Pour qu'un plan (π) intersecte un triangle il faut et il suffit qu'il intersecte au moins un segment du triangle (figure 7).

La figure 7 illustre les différents cas de l'intersection entre un plan et un triangle :

1. Un point (sommets du triangle) (voir figure 7.a),
2. Un segment (voir figure 7.b),
3. Trois segments (le triangle) (voir figure 7.c),
4. Deux points (milieu de deux segments) (voir figure 7.d),
5. Deux points (un sommet et milieu d'un autre segment) (voir figure 7.e),
6. Un ensemble vide (le triangle est parallèles au plan et les sommets n'appartiennent pas au plan) (voir figure 7.f).

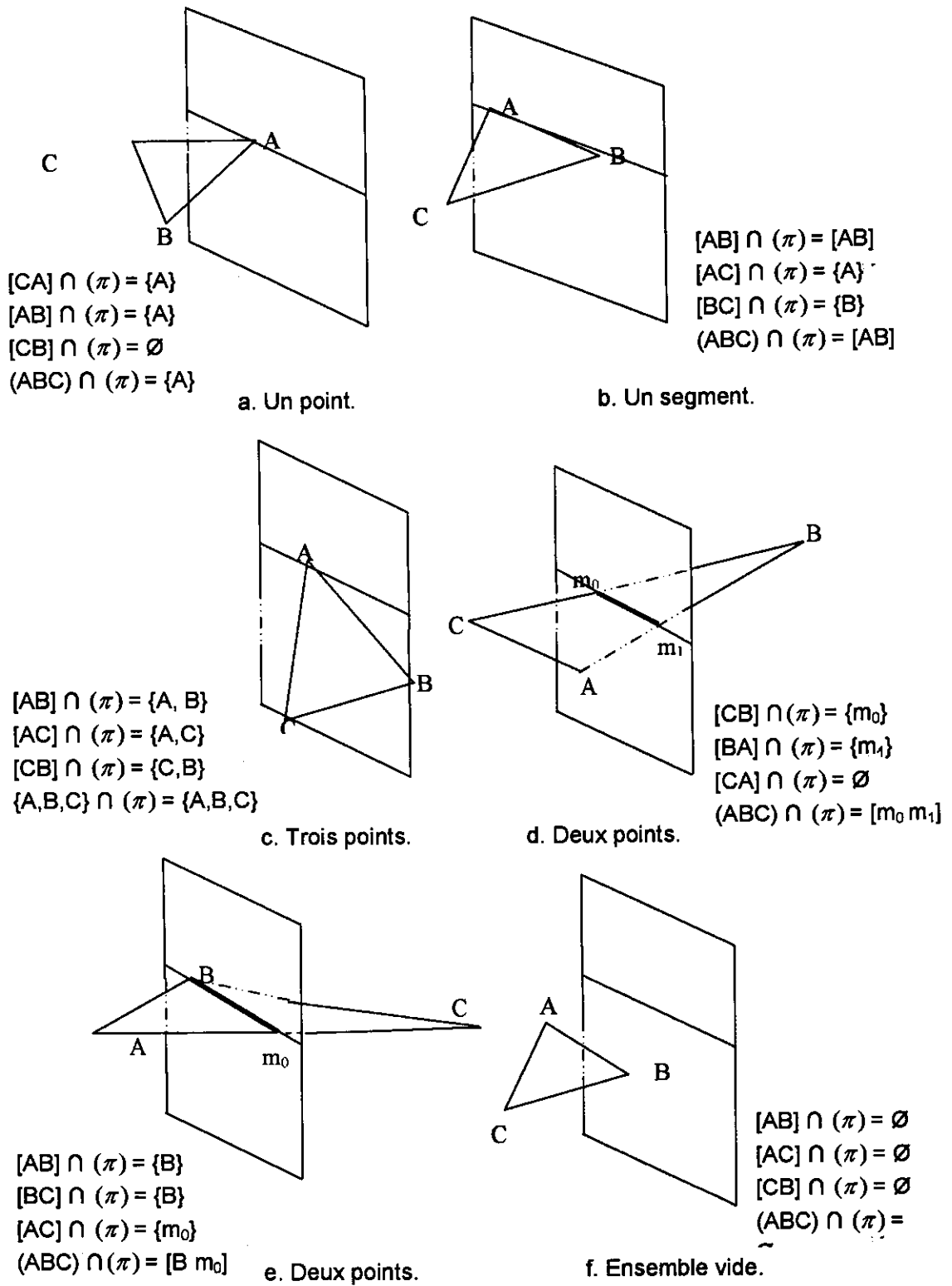


Figure 7 : Intersection entre un triangle et un plan.

Pour trouver l'intersection entre un plan et un triangle il faut avoir l'équation du segment du triangle et l'équation du plan.

III.2. Equation de la droite dans l'espace [1] :

Par deux points P_0 et P_1 dans l'espace \mathfrak{R}^3 , on ne peut faire passer qu'une seule droite. La droite qui passe par P_0 et P_1 est définie comme l'ensemble des points M qui vérifient la relation suivante (voir figure 8) :

$$\overrightarrow{P_0M} // \overrightarrow{P_0P_1} \quad (IV.1)$$

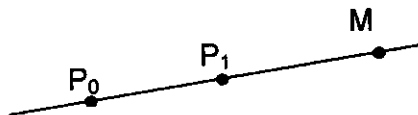


Figure 8 : Droite dans l'espace.

Soient $P_0(x_0, y_0, z_0)$ et $P_1(x_1, y_1, z_1)$ deux points de l'espace, soit $M(x, y, z)$ un point de la droite Δ_M qui passe par les deux points P_0 et P_1 .

$$M \in \Delta_M \Leftrightarrow \overrightarrow{P_0M} // \overrightarrow{P_0P_1} \Leftrightarrow \exists \alpha \in \mathbb{R} / \overrightarrow{P_0M} = \alpha \overrightarrow{P_0P_1} \quad (IV.2)$$

Le développement de l'équation (IV.2) donne :

$$\begin{cases} X = X_0 + \alpha(X_1 - X_0) \\ Y = Y_0 + \alpha(Y_1 - Y_0) \\ Z = Z_0 + \alpha(Z_1 - Z_0) \end{cases} \quad (IV.3)$$

III.3. Equation du plan dans l'espace [8] :

L'équation d'un plan dans l'espace est donnée par :

$$AX + BY + CZ + D = 0 \quad (IV.4)$$

Où A , B et C sont les composantes du vecteur normal au plan. Dans le cas où le plan est vertical (voir figure 9), son équation est donnée par :

$$AX + BY + D = 0 \quad (IV.5)$$

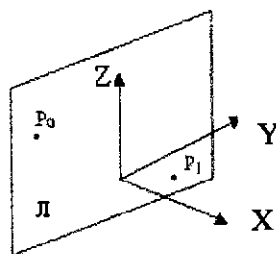


Figure 9 : Plan vertical.

III.4. Intersection entre un plan et une droite dans l'espace [1] :

L'intersection entre un plan et une droite dans l'espace donne les résultats suivants:

1. Ensemble vide,
2. La droite elle même,
3. Un point.

Pour déterminer l'intersection entre un plan et une droite, il suffit de considérer les équations (IV.3) et (IV.4) et de déterminer la constante α et par la suite X, Y et Z.

La triangulation permet de rendre le problème d'intersection entre une surface de forme quelconque et un plan un problème linéaire.

III.5. Appartenance d'un point à un segment de droite :

Soit $[AB]$ un segment de droite dans l'espace et M un point.

$$M \in [AB] \Leftrightarrow \exists \alpha / 0 \leq \alpha \leq 1, \overline{AM} = \alpha \cdot \overline{AB} \quad (\text{IV.6})$$

La figure 10 montre les positions de trois points par rapport au segment de droite $[AB]$.

$$\begin{cases} M_1 \in [AB] & \text{et } 0 \leq \alpha \leq 1 \\ M_2 \notin [AB] & \text{car } \alpha > 1 \\ M_3 \notin [AB] & \text{car } \alpha < 0 \end{cases} \quad (\text{IV.7})$$

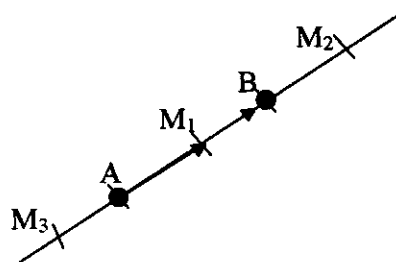


Figure 10 : Position des différents points par rapport au segment $[AB]$.

III.6. Equation d'un cylindre dans l'espace [21, 30] :

Le cylindre qui est défini par l'axe passant par les deux points $P_0(x_0, y_0, z_0)$ et $P_1(x_1, y_1, z_1)$ et le rayon R (voir figure 11) est l'ensemble des points $M(x, y, z)$ dont la distance à l'axe du cylindre est égale au rayon R. Son équation est donnée par la formule suivante :

$$R = \frac{\| \overrightarrow{P_0 P_1} \wedge \overrightarrow{P_0 M} \|}{\| \overrightarrow{P_0 P_1} \|} \tag{IV.8}$$

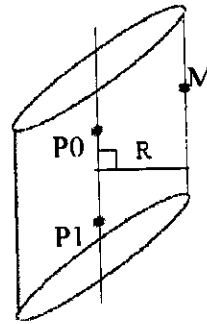


Figure 11 : Cylindre dans l'espace.

Dans le cas d'un cylindre vertical et son axe c'est l'axe Z, son équation est donnée par :

$$X^2 + Y^2 = R^2 \tag{IV.9}$$

III.7. Intersection entre une droite et un cylindre :

Pour déterminer l'intersection entre un cylindre et une droite, il suffit de résoudre le système d'équations formé par les équations (IV.3) et (IV.9).

L'intersection d'un cylindre avec une droite peut être:

1. Un point (voir figure 12.a).
2. Un ensemble vide (voir figure 12.b).
3. Deux points (figure 12.c).
4. La génératrice (droite tangente au cylindre) (voir figure 12.d)

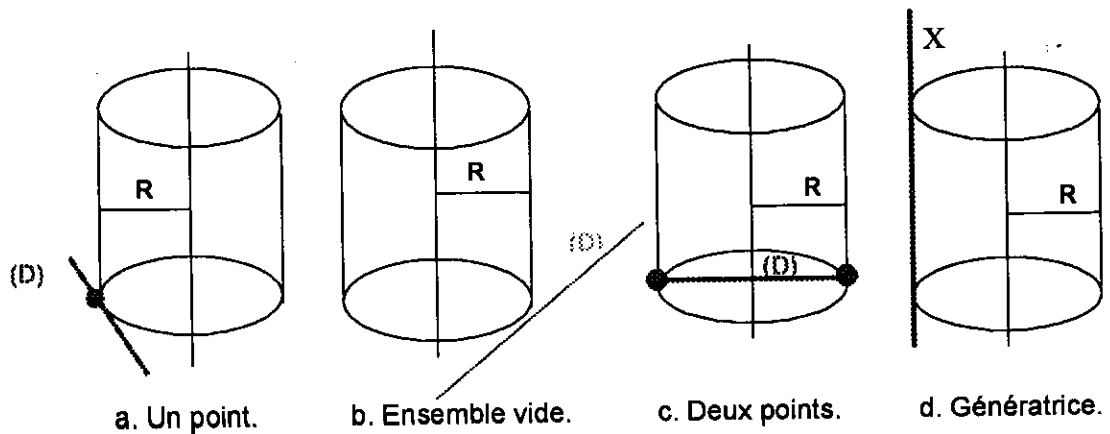


Figure 12 : Intersection d'une droite avec un cylindre.

III.8. Equation d'une sphère [30,31] :

Une sphère est une surface à 3 dimensions dont tous les points sont situés à une même distance d'un point appelé centre. La valeur de cette distance commune au centre est appelée le rayon de la sphère. Elle n'inclut donc pas les points situés à une distance inférieure au rayon (cas de la boule).

En géométrie cartésienne, une sphère de centre $P_0(X_0, Y_0, Z_0)$ et de rayon R (voir figure 13) est l'ensemble des points (X, Y, Z) tels que [31] :

$$(X_1 - X_0)^2 + (Y_1 - Y_0)^2 + (Z - Z_0)^2 = R^2 \quad (\text{IV.10})$$

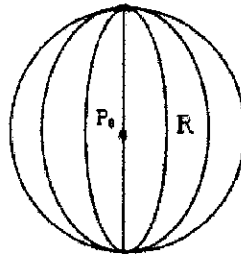


Figure 13 : Sphère de centre P_0 et de rayon R .

III.9. Intersection d'une droite et d'une sphère [30] :

Pour déterminer l'intersection entre une sphère et une droite, il suffit de résoudre le système formé par les équations (IV.3) et (IV.10).

L'intersection d'une sphère avec une droite du triangle de la surface peut être :

1. Un point (voir figure 14.a).
2. Un ensemble vide (voir figure 14.b).
3. Deux points (voir figure 14.c).

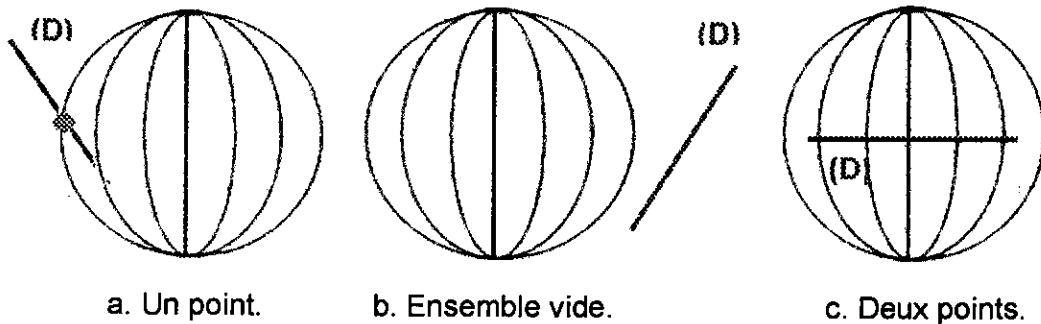


Figure 14 : Intersection d'une droite et une sphère.

III.10. Equation du tore [33] :

Le tore est une surface de révolution engendrée par la rotation d'un cercle autour d'un axe situé dans son plan et ne le coupant pas (voir figure 15).

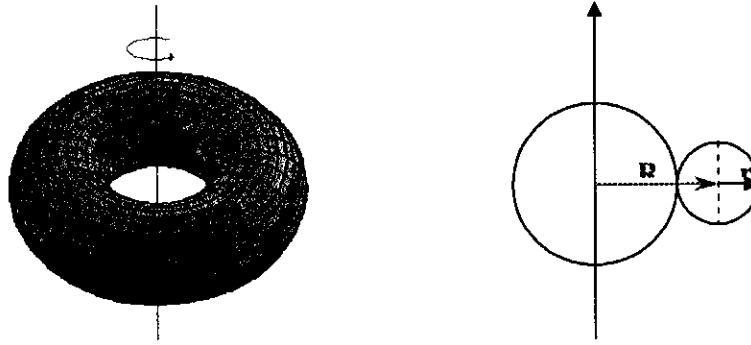


Figure 15 : Tore dans l'espace.

L'équation d'un tore est définie par :

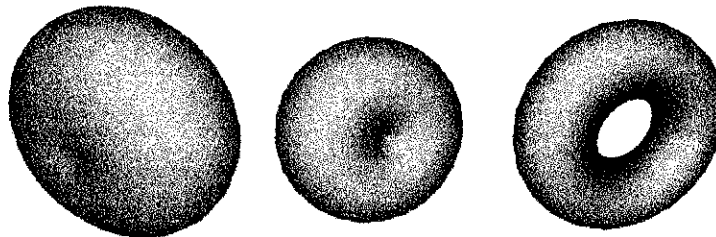
$$(X^2 + Y^2 + Z^2 + R^2 - r^2)^2 = 4R^2(X^2 + Y^2) \quad (\text{IV.11})$$

Avec :

- R : le rayon major du tore,
- r : le rayon mineur du tore.

En fonction de R et r, le tore peut prendre trois formes différentes [31] :

- Si $R < r$, le tore est dit « croisé » et ressemble visuellement à une citrouille (voir figure 16.a).
- Si $R = r$, le tore est dit « à collier nul » (voir figure 16.b).
- Si $R > r$, le tore est dit « ouvert » et ressemble à une chambre à air (voir figure 16.c).



a. $R < r$.

b. $R = r$.

c. $R > r$.

Figure 16 : Différentes formes d'un tore.

III.11. Intersection d'une droite et un tore [30] :

L'intersection d'un tore et une droite est obtenue en considérant les équations (IV.3) et (IV.11).

Cette intersection peut être :

1. Un point (voir figure 17.a).
2. Un ensemble vide (voir figure 17.b).
3. Deux points (voir figure 17.c).
4. Trois points (voir figure 17.d).
5. Quatre points (voir figure 17.e).

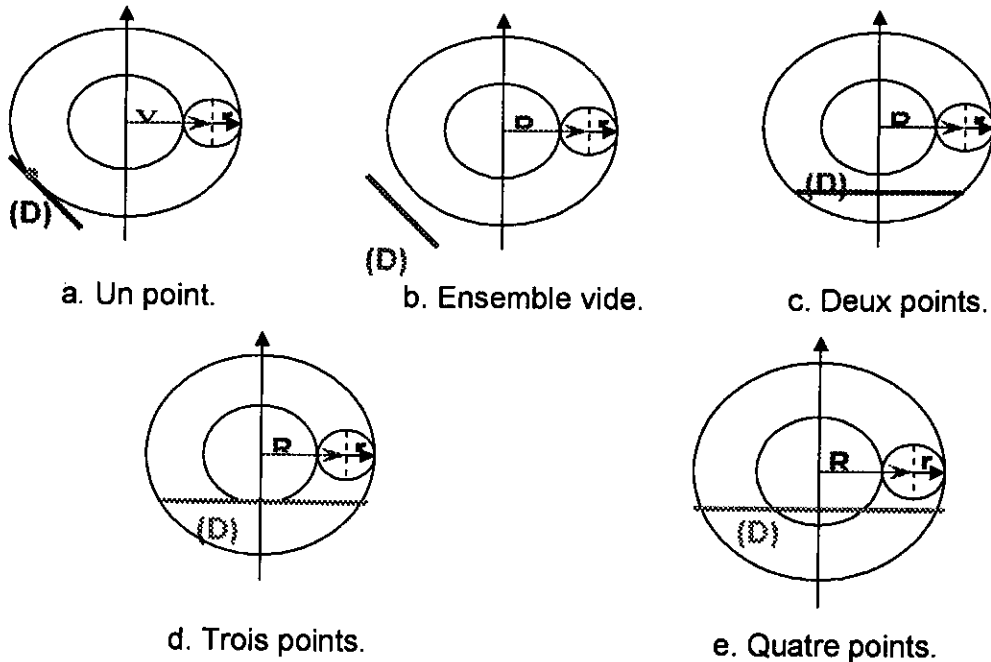
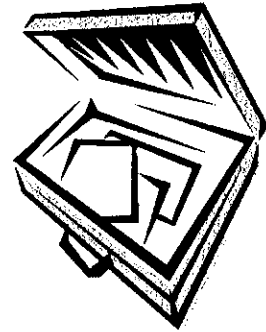


Figure 17 : Intersection d'un tore avec une droite.

IV. CONCLUSION :

Dans ce chapitre, nous avons considéré une des méthodes d'approximations des surfaces qui est la triangulation ainsi que les équations de quelques objets géométriques nécessaires à la suite du travail et leurs intersections avec une droite. Dans la suite de notre travail, nous allons utiliser la triangulation uniforme et la triangulation adaptative pour approximer les surfaces à usiner ainsi que l'intersection d'une sphère avec une droite pour générer la surface offset. Dans le chapitre suivant, nous allons présenter la conception de notre application.



CHAPITRE 5
CONCEPTION DE L'APPLICATION
LOGICIELLE

I. INTRODUCTION :

Dans les chapitres précédents nous avons vu les différentes notions fondamentales de l'usinage des surfaces gauches en finition ainsi que les machines utilisées. Nous allons maintenant entamer la conception de notre application en utilisant le modèle orienté objet, le langage de modélisation UML et la démarche en cascade.

II. METHODE DE MODELISATION [1,2] :

Pour le développement de notre application logicielle, nous avons utilisé le Builder **C++** comme langage de programmation et pour faire la conception, nous avons utilisé le modèle orienté objet et le langage de modélisation UML (*Unified Modeling Language*) qui est un langage graphique qui permet de représenter, visualiser, construire un système informatique vu qu'il est compatible avec toutes les techniques de réalisation des applications et indépendant du langage de programmation.

III. REALISATION DE L'APPLICATION :

Pour réaliser notre application, nous devons passer par plusieurs étapes :

III.1. Cahier de charges :

Pour structurer les besoins de notre système, nous allons créer un cahier de charge.

III.1.1. Présentation du projet :

Notre projet s'insère dans le cadre de développement d'outils de conception et de fabrication des surfaces gauches initiés par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) au niveau de la Division Productique et Robotique du Centre de Développement des Technologies Avancés (CDTA).

Le but de ce travail est l'automatisation de l'usinage des surfaces gauches par la méthode « I.T.O ». La finalité de ce travail est le développement d'une application logicielle graphique et interactive sous Windows qui permet d'approximer un modèle géométrique, calculer le rayon d'outil optimal théorique qui évite les interférences et ensuite de choisir l'outil manuellement ou à partir d'une base de données d'outils. Après cette étape, la suite construction de la surface offset (surface de centres outil) par la méthode « I.T.O » et enfin la génération du trajet d'usinage à partir des différentes positions d'outil.

III.1.2. Problématique :

Les pièces de formes libres (moules, matrices, formes esthétiques, formes aérodynamiques, ...etc.) sont très utilisées dans notre vie quotidienne, à cause de leurs géométries très complexes, ces pièces ne peuvent être usinées que sur des

fraiseuses à commande numérique à 03 ou à 05 axes. Le trajet d'outils sera traduit en un programme « G-Code » compréhensible par la machine. Ce programme contient un nombre très important de lignes d'instructions et donc la génération manuelle est impossible. Les surfaces usinées contiennent toujours des erreurs qui peuvent être soit l'existence des zones non usinées soit l'existence des zones d'interférences et de collisions. La génération de surface offset par la méthode « I.T.O » permet de générer le trajet d'usinage pour n'importe quelle forme d'outil et permet aussi d'éliminer les problèmes d'interférences.

III.1.3. Objectifs visés :

Notre objectif principal dans l'application est l'Automatisation de l'Usinage des Surfaces Gauches par la Méthode « I.T.O ». Cet objectif peut être subdivisé en plusieurs sous objectifs :

- **Premier objectif** : approximation des modèles CAO des surfaces à usiner par des éléments géométriques simples (triangles). Pour atteindre cet objectif, nous devons passer par les étapes suivantes :
 1. Approximation uniforme de la surface par un certain nombre de triangles.
 2. Une deuxième approximation de surface par la triangulation adaptative en fonction des paramètres introduits par l'utilisateur.
- **Deuxième objectif** : choix d'outils optimums afin d'éliminer les interférences. Cet objectif est aussi divisé en sous objectifs.
 1. Calcul du rayon d'outil théorique pour la surface isolée à partir des rayons de courbures en chaque point de la surface.
 2. Calcul du rayon d'outil optimum qui évite les interférences en considérant toutes les surfaces.
 3. Choix définitif de l'outil manuellement ou à partir d'une base de données des outils.
 4. Détermination des paramètres d'outils (vitesse d'avance, vitesse d'engagement et vitesse de dégagement).
- **Troisième objectif** : génération de la surface offset par l'approche « I.T.O ».
- **Quatrième objectif** : usinage par plans parallèles. Cet objectif est divisé en sous objectifs.
 1. Création des plans verticaux.
 2. Calcul et tri des points d'intersections pour chaque plan.
 3. Génération des positions d'outils en One-Way ou en Zig-Zag.
 4. Simulation des mouvements de l'outil par rapport à la surface.
 5. Génération du programme d'usinage « G-Code ».
 6. Usinage réel.

III.1.4. Plateforme exigée :

- Recommandations matérielles (configuration minimale nécessaire) :
 1. Intel Pentium 4 de 3 GHz ou équivalent.
 2. Une RAM de 1Go minimum.
 3. Carte accélératrice 3D de 128 Mo compatible OpenGL ou équivalente.
 4. Lecteur de CD-ROM/DVD-ROM.
 5. Souris PS/2.
 6. Clavier.
 7. Ecran 19 pouces.
- Système d'exploitation : Windows XP
- Logiciels requis :
 1. Builder C++ V.6.
 2. Bibliothèque graphique OpenGL.

III.2. Solution de la problématique :

Pour atteindre les quatre objectifs cités précédemment, nous allons développer une application logicielle graphique et interactive sous Windows et nous allons procéder comme suit :

- Développement des algorithmes d'approximations des surfaces gauches.
- Développement des algorithmes de détection et correction des interférences.
- Développement des algorithmes de génération des surfaces offset par la méthode « I.T.O ».
- Développement des algorithmes de génération du trajet d'outil.

III.3. Modélisation de l'application en UML :

Pour exprimer le fonctionnement du système, nous allons utiliser dans cette modélisation les cinq diagrammes de l'ensemble des diagrammes du langage UML. Nous allons détailler le diagramme de cas d'utilisation, les diagrammes de séquence, le diagramme de collaboration, les diagrammes d'activité et le diagramme de classes.

Le cycle de vie d'un logiciel désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. Pour cela, nous allons utiliser le cycle de vie en cascade en commençant par l'analyse, la conception, l'implémentation, les tests et la validation et enfin la maintenance.

III.3.1 Diagramme cas d'utilisation :

Il a pour objet de décrire les situations auxquelles le système doit répondre. On se place du point de vue de l'utilisateur.

Le diagramme cas d'utilisation est caractérisé par :

- **Acteurs** : ce sont des personnes ou systèmes extérieurs à l'application et qui interagissent avec elle.
- **Cas d'utilisation** : représentent une fonctionnalité (un objectif à atteindre) du système à construire. Ils sont en relation avec des acteurs et d'autres cas d'utilisation.
- **Relations** : généralement orientées.

Dans ce qui suit, nous allons présenter les principaux cas d'utilisations en commençant par « l'ouverture du modèle CAO » que nous voulons usiner par une action d'un acteur (utilisateur du système), qui doit intervenir dans « l'approximation du modèle CAO » et « le choix d'outils ». Cette dernière partie est subdivisée en trois étapes :

- Choix d'outil théorique,
- Choix d'outil optimum,
- Choix d'outil définitif à partir d'une base de données des outils (automatique), ou bien manuellement par l'utilisateur

Par la suite, l'utilisateur a la possibilité de lancer la génération de la surface offset par la méthode ITO et la génération du trajet d'outil pour pouvoir simuler l'usinage et générer le « G-Code ».

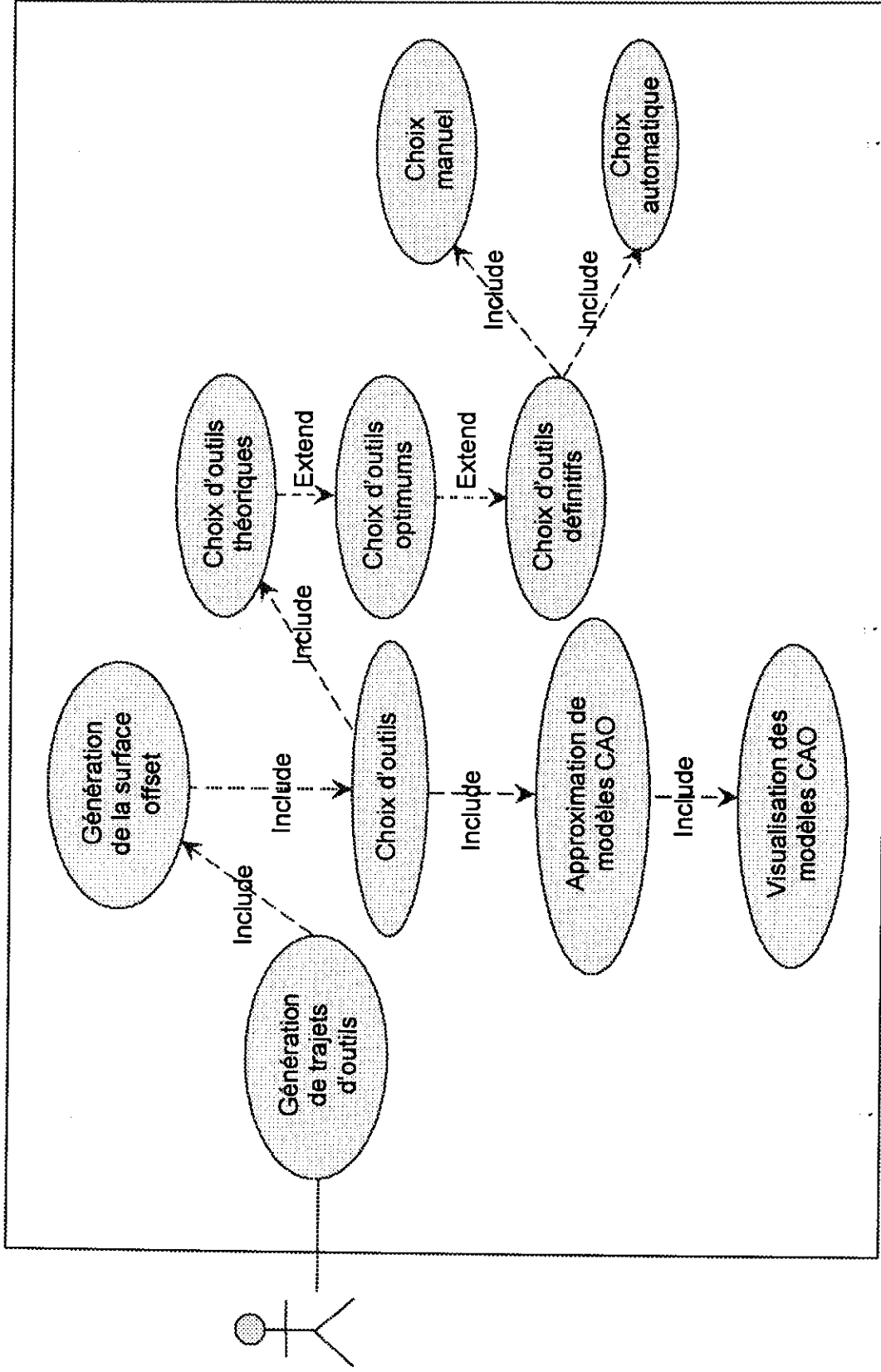


Figure 1. Diagramme de cas d'utilisation.

III.3.2 Diagramme de séquences :

Le diagramme de séquence met en évidence l'aspect chronologique de l'envoi des messages. Souvent, le diagramme de séquence permet de compléter le diagramme des cas d'utilisation en mettant en évidence les objets et leur interaction.

La visualisation des modèles CAO passe par les étapes suivantes :

- L'utilisateur demande l'ouverture des modèles CAO.
- Le système ouvre et visualise les modèles CAO.

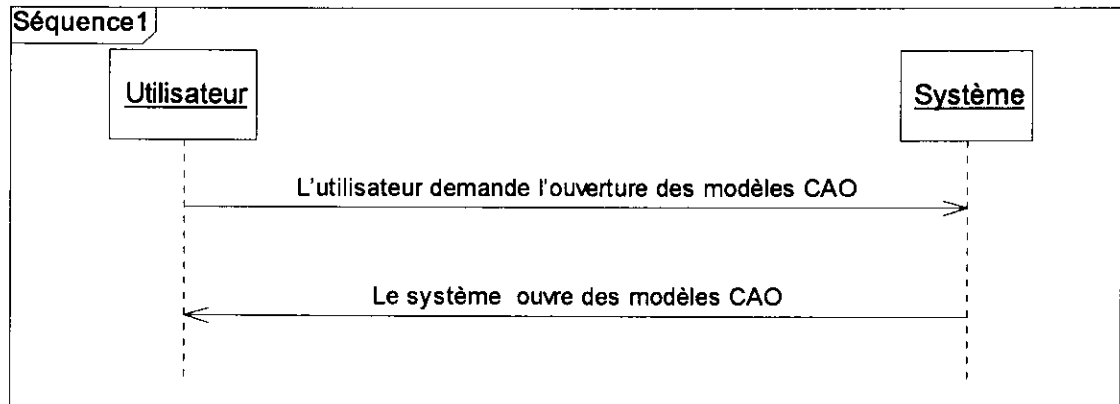


Figure 2 : Diagramme de séquence d'ouverture du modèle CAO.

L'approximation de la surface passe par les étapes suivantes :

- L'utilisateur demande l'approximation du modèle.
- Le système ouvre la fenêtre d'approximation.
- L'utilisateur introduit les paramètres de précision et lance Uniforme.
- Le système offre une première approximation du modèle.
- L'utilisateur lance la triangulation adaptative.

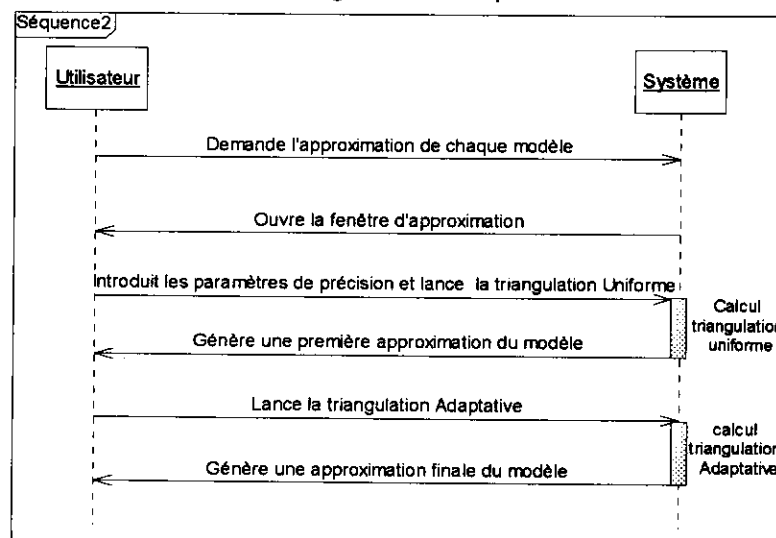


Figure 3 : Diagramme de séquence d'approximation du modèle.

Le choix d'outil théorique passe par les étapes suivantes :

- L'utilisateur demande le choix d'outils théorique.
- Le système ouvre la fenêtre du choix d'outils.
- L'utilisateur introduit le mode de choix et demande de calculer le rayon théorique.
- Le système calcule le rayon théorique.
- Le système affiche le rayon théorique calculé.

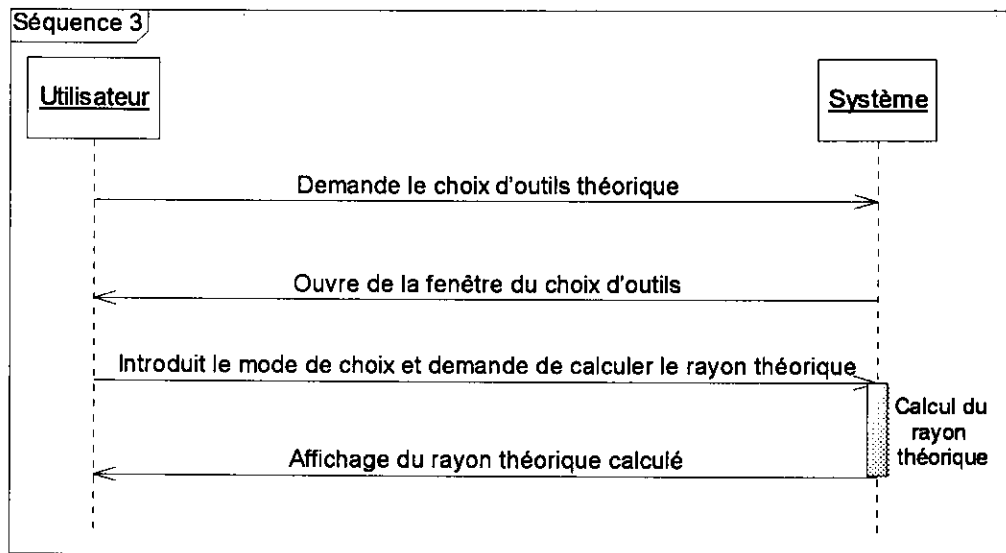


Figure 4 : Diagramme de séquence de choix d'outils théorique.

Le choix de l'outil optimum passe par les étapes suivantes :

- L'utilisateur demande la génération des régions.
- Le système génère les régions.
- L'utilisateur demande la détection et la correction des interférences.
- Le système calcule un rayon optimum qui élimine les interférences.
- Le système affiche le rayon optimum.

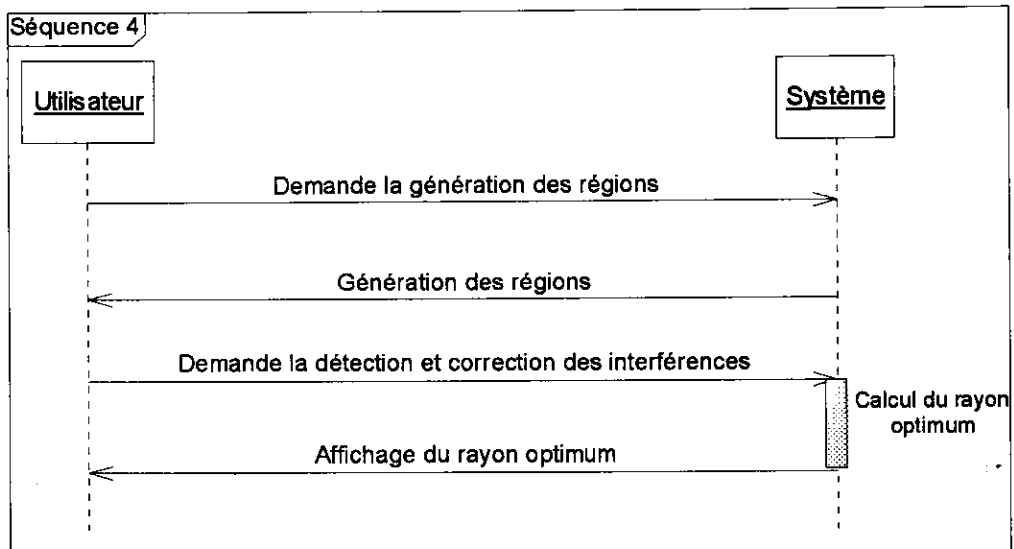


Figure 5 : Diagramme de séquence de choix d'outil optimum.

Le choix d'outil définitif passe par les étapes suivantes :

- L'utilisateur demande le choix d'outil définitif.
- Le système ouvre la fenêtre du choix d'outils définitif.
- L'utilisateur choisit un mode de choix.

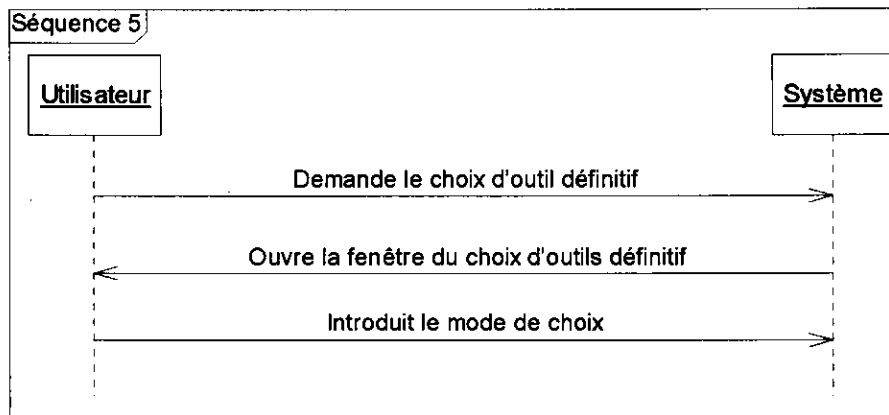


Figure 6 : Diagramme de séquence de choix d'outil définitif.

Le choix d'outil automatique passe par les étapes suivantes :

- L'utilisateur demande le choix automatique de l'outil.
- Le système affiche la base de données des outils.
- L'utilisateur introduit le choix à partir de la base de données.
- Le système affiche l'outil choisi ainsi que ces paramètres.

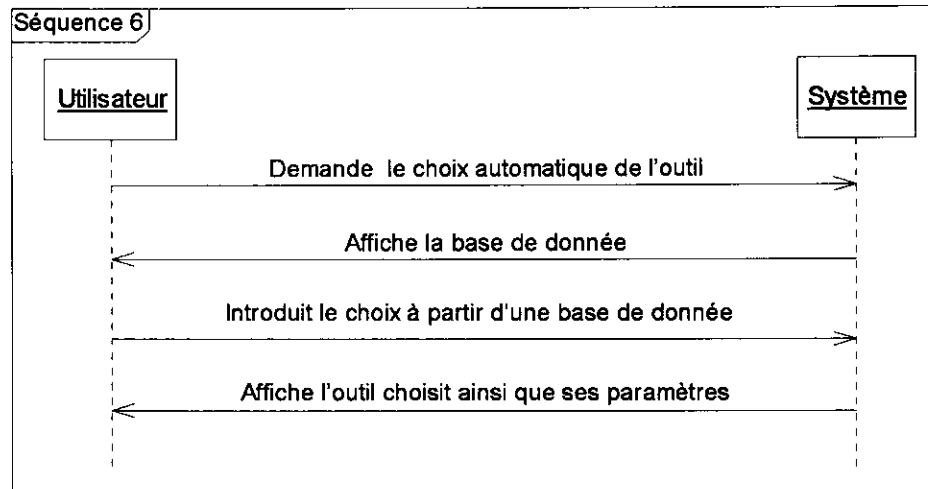


Figure 7 : Diagramme de séquence de choix d'outil automatique.

Le choix d'outil manuel passe par les étapes suivantes :

- L'utilisateur demande de choix manuel de l'outil.
- Le système offre les champs des paramètres de l'outil et affiche les outils existant dans la base de données des outils.
- L'utilisateur introduit les paramètres de l'outil.
- Le système affiche les paramètres de l'outil choisi.

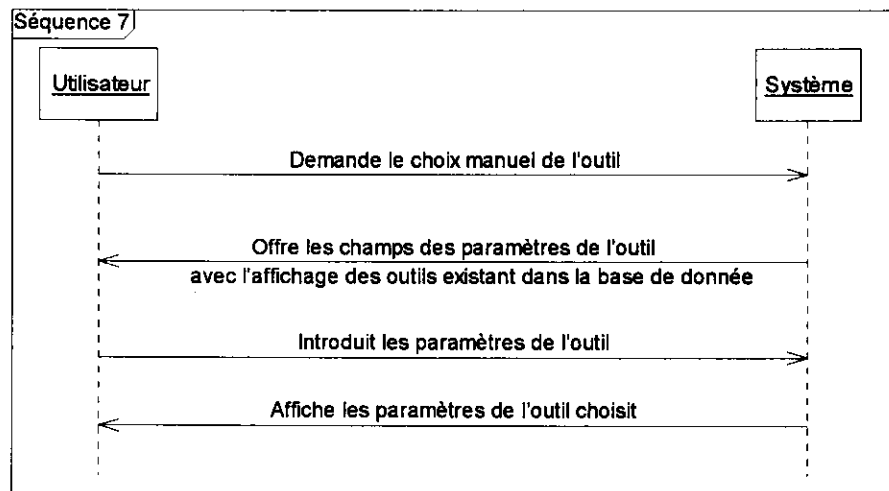


Figure 8 : Diagramme de séquence de choix d'outil manuel.

La génération de la surface offset passe par les étapes suivantes :

- L'utilisateur demande l'introduction des paramètres d'usinage,
- Le système ouvre la fenêtre de paramètres d'usinage,
- L'utilisateur introduit les paramètres d'usinage,

- L'utilisateur demande la fenêtre de la méthode ITO,
- Le système ouvre la fenêtre de la méthode ITO,
- L'utilisateur lance la méthode ITO,
- Le système génère les nouveaux sommets, segments et triangles de la surface offset.

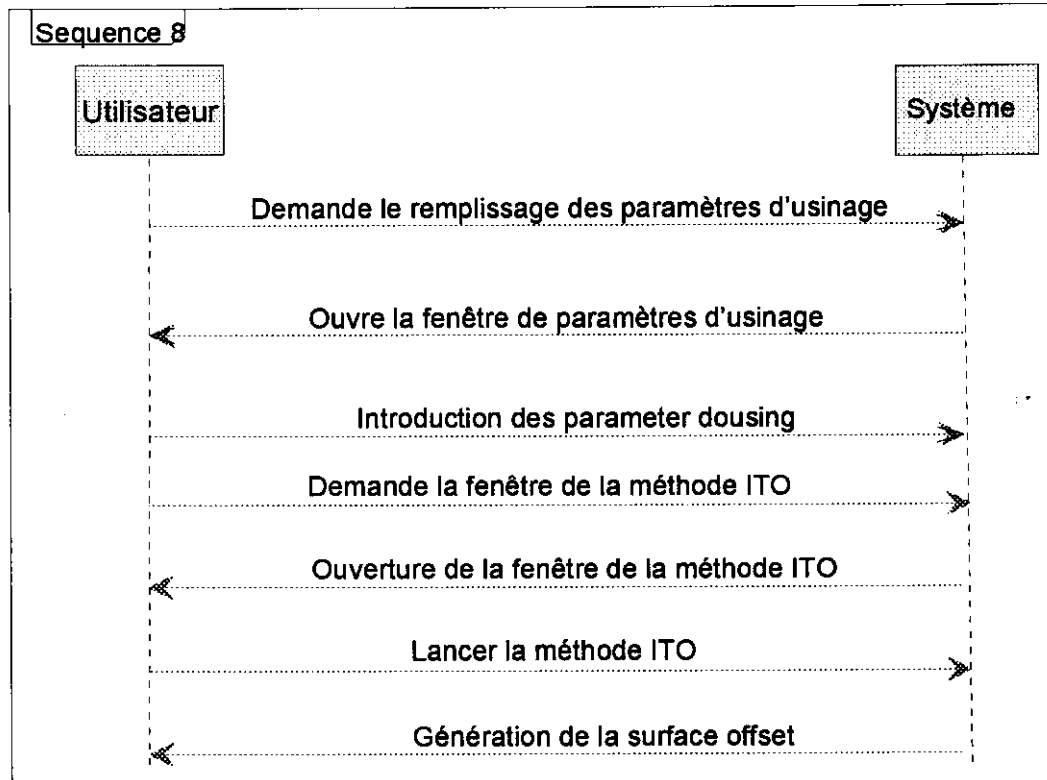


Figure 9 : Diagramme de séquence de la génération de surface offset.

La génération de trajet d'outils par la méthode des plans parallèles passe par les étapes suivantes :

- L'utilisateur demande l'usinage par la méthode des plans parallèles.
- Le système ouvre la fenêtre des paramètres de la méthode.
- L'utilisateur introduit les paramètres de la méthode et lance la génération des points d'intersection.
- Le système génère et tri les points d'intersection.
- L'utilisateur demande la génération du trajet d'outil.
- Le système ouvre la fenêtre de trajet d'outil.
- L'utilisateur introduit le mode de balayage et lance la simulation des mouvements.

- Le système lance la simulation des mouvements.
- L'utilisateur introduit les paramètres du « G-Code » et lance la génération du « G-Code ».
- Le système génère et affiche le « G-Code ».

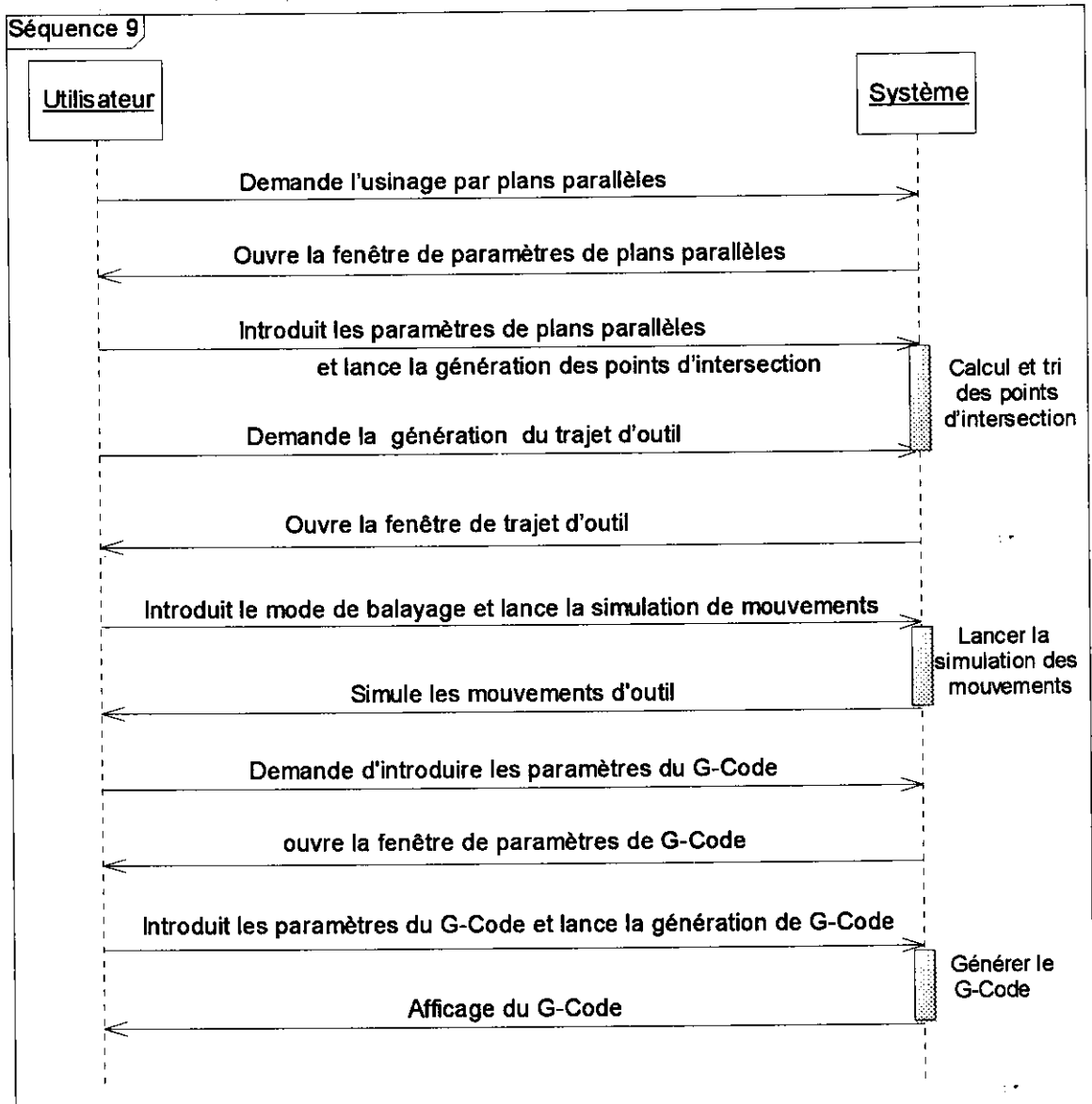


Figure 10 : Diagramme de séquence de la génération de trajet d'outils

III.3.3 Diagramme de collaboration :

Ce diagramme montre les interactions entre les objets. Ils insistent particulièrement sur la structure des liens permettant de mettre les objets en collaboration, en communication. Ils représentent à la fois les objets et les messages que ceux-ci s'envoient.

Un lien sert de support de transmission pour le message. Le message déclenche une action dans l'objet destinataire. Il est également possible de préciser pour un message ses arguments et valeurs de retour. Il complète et détaille un cas d'utilisation.

Dans notre système, la forme de triangulation des surfaces **Form1200** s'occupe du calcul des sommets, des segments, des triangles et des triangles voisins. Tous ces résultats seront affichés dans la forme de résultats de la triangulation (**Form1201**). La **Form1200** envoie les résultats à la forme de choix d'outil (**Form1203**) qui effectue le calcul du rayon théorique et du rayon optimum et les envoie à la forme de choix d'outils à utiliser (**Form1206**) qui elle-même fait le choix définitif de l'outil d'une manière automatique ou manuel. Par la suite, les paramètres d'usinage pour chaque outil utilisé sont introduits dans la forme de paramètres d'usinage (**Form1210**). La forme (**Form1200**) envoie à la forme principale d'affichage (**Form1**) des messages pour la visualisation des points de la surface, des segments, des triangles et des triangles voisins pour chaque triangle. La forme (**Form1203**) envoie des messages d'affichage à la forme (**Form1**) pour la visualisation des régions, des points des régions et des points d'interférences. Toute modification de l'aspect visuel des objets affichés dans la forme (**Form1**) est incorporée dans la forme de paramètres de dessin (**Form1204**). La forme de la méthode de l'outil inverse - ITO - (**Form1007**) récupère les paramètres de l'outil utilisé à partir de la forme (**Form1206**) puis génère la surface offset. Elle envoie des messages d'affichage à la forme (**Form1**). Le calcul des points d'intersection est effectué par la forme **Form1208** responsable des paramètres des plans parallèles tandis que la forme (**Form1209**) permet la génération du trajet d'outil, la simulation du mouvement de l'outil et la génération du programme « G-Code ».

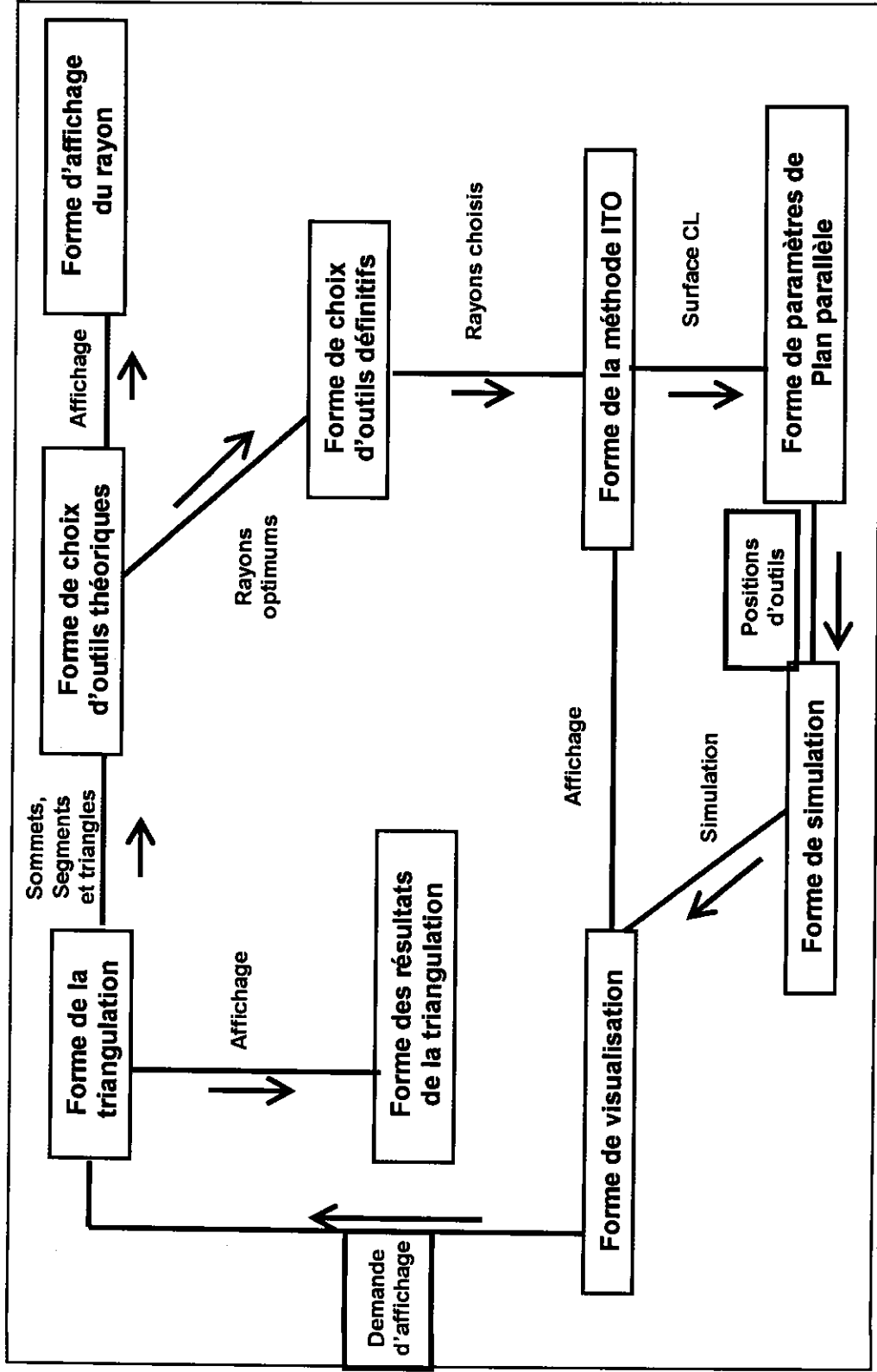


Figure 11 : Diagramme de collaboration.

III.3.4 Diagramme d'activité :

Le diagramme d'activités est un modèle détaillé, il montre l'activité et le fonctionnement d'une opération d'une classe. Dans notre système nous avons le diagramme d'activité d'ouverture du modèle CAO (voir figure 17), le diagramme d'activité d'approximation du modèle (voir Figure 18), le diagramme d'activité du choix d'outil (voir figure 19) et enfin le diagramme d'activité de l'usinage par la méthode des plans parallèles (voir figure 20).

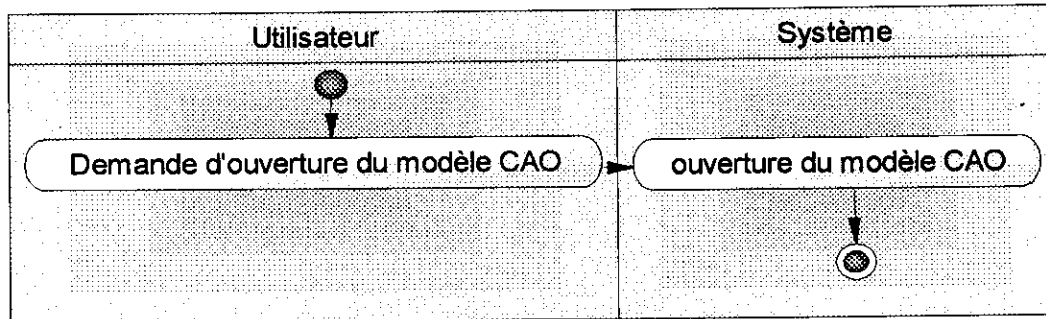


Figure 12 : Diagramme d'activité d'ouverture du modèle CAO.

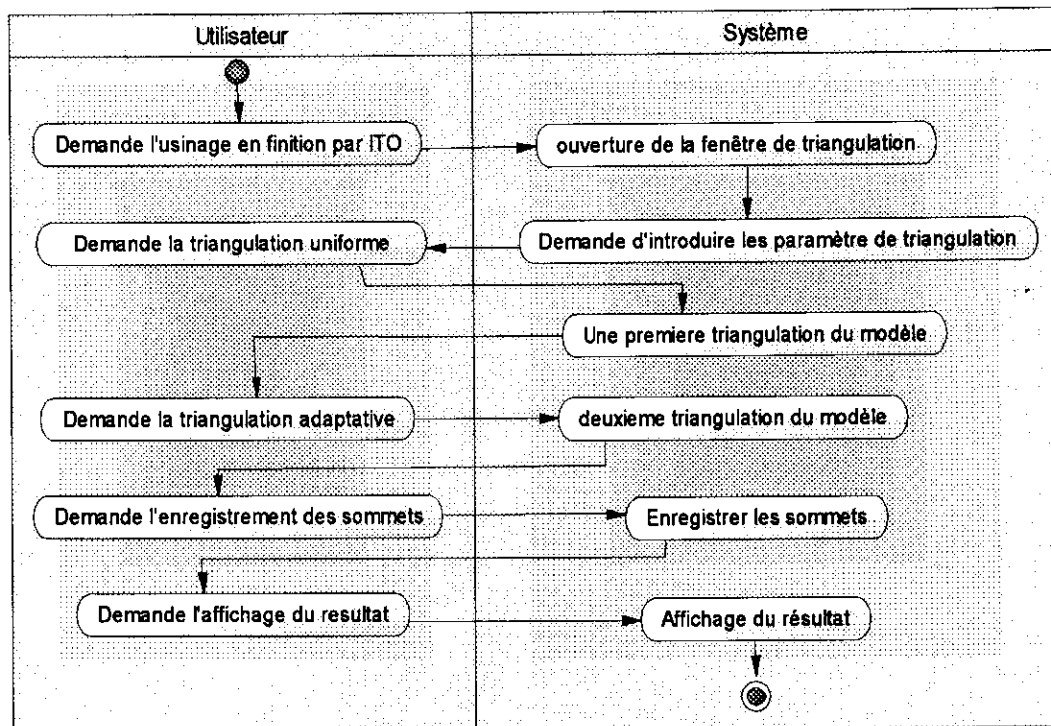


Figure 13 : Diagramme d'activité d'approximation du modèle CAO.

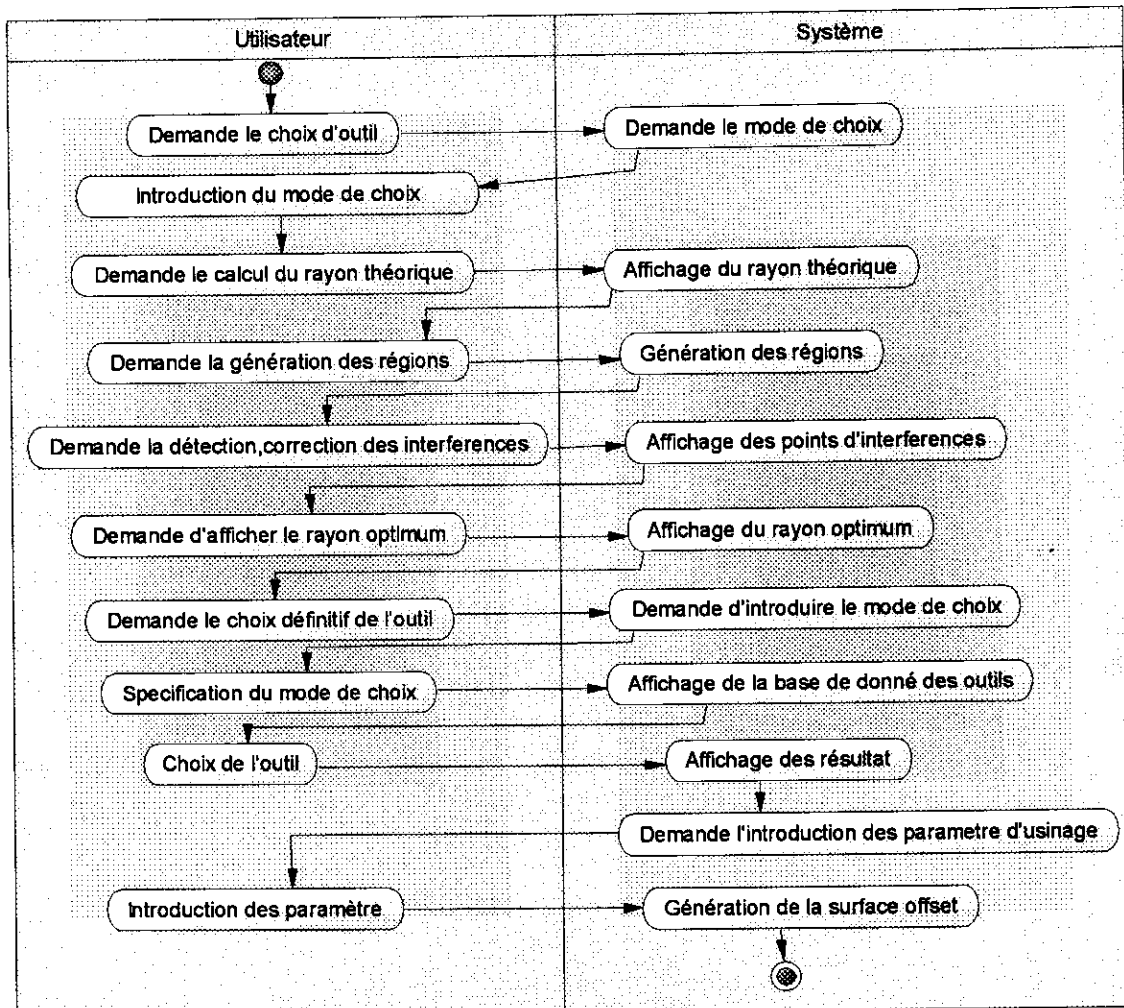


Figure 14 : Diagramme d'activité de choix d'outil.

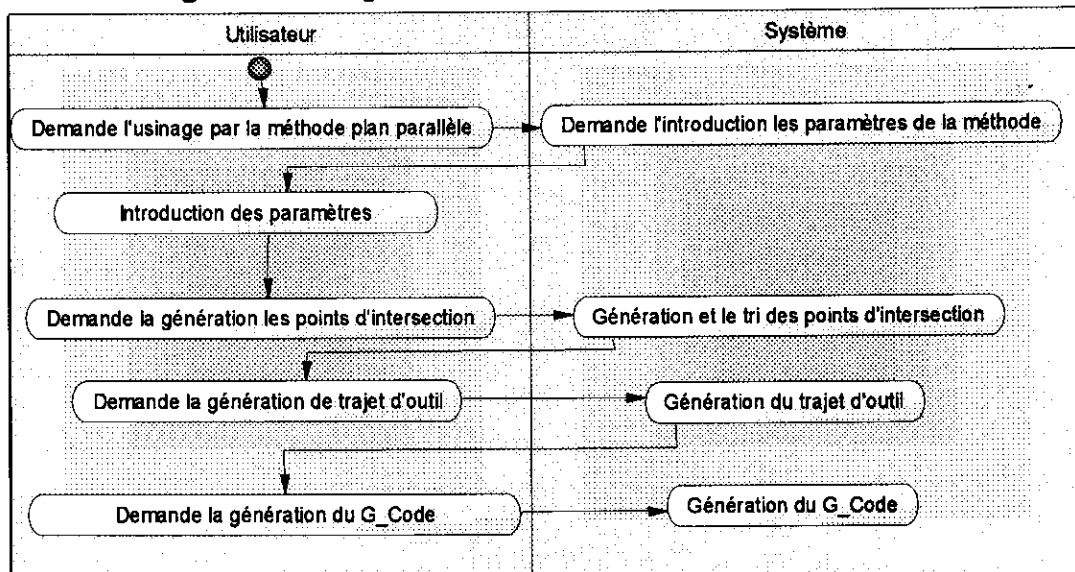


Figure 15 : Diagramme d'activité de la génération de trajet d'outils

Les différentes classes utilisées dans notre conception sont les suivantes :

- **Classe de triangulation (TForm1200)** : c'est une classe mère qui englobe toutes les classes que nous avons utilisées. Dans cette classe, nous allons générer les sommets, les segments et les triangles pour chaque modèle CAO de surface. Cette classe comporte plusieurs attributs et plusieurs fonctions.
- **Classe de résultats de triangulation (TForm1201)** : dans cette classe, nous allons récupérer tous les résultats obtenus de la TForm1200.
- **Classe de choix d'outil théorique (TForm1203)** : cette classe permet le calcul du rayon d'outil théorique et la génération des régions pour l'accélération du calcul du rayon optimum qui élimine les interférences.
- **Classe d'affichage des différents rayons (TForm1202)** : dans cette classe, nous allons récupérer les rayons calculés (théorique, optimum et choisi).
- **Classe de paramètres de dessin (TForm1204)** : toute modification de couleur et épaisseur des éléments géométriques des dessins est faite dans cette classe.
- **Classe de choix d'outil définitif (TForm1206)** : dans cette classe, nous allons choisir définitivement l'outil pour l'usinage.
- **Classe de la méthode ITO (TForm1207)** : la génération de la surface offset est faite dans cette classe.
- **Classe de paramètres de la méthode des plans parallèles (TForm1208)** : dans cette classe, nous allons créer des plans verticaux suivant un angle d'inclinaison donné et une distance entre deux plans successifs spécifiés par l'utilisateur et par la suite la génération et le tri des points d'intersection.
- **Classe de trajet d'outil (TForm1209)** : la génération du trajet d'outil est faite dans cette classe. Nous avons utilisé aussi d'autres classes pour la simulation des mouvements d'outils et pour la génération du programme « G-Code ».

Les principaux attributs de la classe Triangulation sont les suivants :

Triangulation	
+ Liste sommet surface	: LIST_SOMMET
+ Liste segment surface	: LIST_SEGMENT
+ Liste triangle	: LIST_TRIANGLE
+ Triangle voisin	: NOEUD_TRIANGLE
+ Liste info uniforme	: INFORMATION
+ Liste info adaptative	: INFORMATION
+ Liste rayon d'outil	: RAYON_D'OUTIL
+ Liste région surface	: LIST_REGION
+ Ensemble surface	: ENSEMBLE_SURFACE
+ Plan verticaux	: TPLAN_VERTICAL

Figure 17. Classe de triangulation.

Parmi les attributs de cette classe :

1. Liste sommet surface : est une liste de sommets propre pour chaque surface qui permet de stocker tous les sommets que nous allons générer, elle est de type LIST_SOMMET.

Le type LIST_SOMMET est une classe définie par un pointeur sur le début de la liste et un pointeur sur la fin de la liste. Ces deux pointeurs sont de type NOEUD_SOMMET (voir figure 18).

Chaque type NOEUD_SOMMET est une classe définie par un sommet de type TSOMMET et un pointeur sur le nœud suivant (voir figure 19).

Le type TSOMMET est une classe qui définit le sommet caractérisé par (voir figure 20) :

- u, v : coordonnées du point dans le plan paramétrique,
- x, y, z : coordonnées du point dans l'espace 3D,
- tux, tuy, tuz : composantes du vecteur tangent T_u ,
- tvx, tvy, tvz : composantes du vecteur tangent T_v ,
- nux, nuy, nuz : composantes du vecteur normal,
- Rmax, Rmin : rayons de courbures principaux.

La classe LIST_SOMMET comporte plusieurs fonctions et les plus importantes sont les suivantes :

- Créer sommet () : calcul des coordonnées en 3D à partir de ces coordonnées paramétriques u et v.
- Créer liste sommet : création de la liste des sommets pour une surface donnée.

- Ajouter sommet () : permet d'insérer un sommet à la liste des sommets.
- Ajouter sommet _région () : permet d'insérer un sommet à une région.
- Ajouter sommet _interférence () : permet d'insérer un sommet d'interférence à la liste des points d'interférence pour chaque surface.
- Dessiner sommet () : permet de dessiner les points de la surface générée.
- Vider sommet () : permet de vider la liste des sommets de la surface.

LIST_SOMMET	
- *debut	: NOEUD_SOMMET
+ crier_liste_sommet ()	: LIST_SOMMET
+ dessiner_sommet ()	: void
+ ajouter_sommet ()	: LIST_SOMMET
+ vider_sommet ()	: void
+ ajouter_sommet_region ()	: void
+ ajouter_sommet_interférence ()	: void

Figure 18 : Classe de LIST_SOMMET.

NOEUD_SOMMET	
- s	: SOMMET
- *suivant	: NOEUD_SOMMET
- *suivant_région	: NOEUD_SOMMET
- *suivant_interférence	: NOEUD_SOMMET
+ &opérateur= ()	: NOEUD_SOMMET
+ opérateur== ()	: bool

Figure 19 : Classe de NOEUD_SOMMET.

SOMMET	
- u	: double
- v	: double
- x	: double
- y	: double
- z	: double
- tux	: double
- tuy	: double
- tuz	: double
- nx	: double
- ny	: double
- nz	: double
- Rmax	: double
- Rmin	: double
+ opérateur== ()	: bool

Figure 20 : Classe de SOMMET.

2. Liste segment surface : est une liste de segments de type LIST_SEGMENT propre pour chaque surface. Elle permet de stocker les segments générés.

Ce type LIST_SEGMENT est une classe définie par un numéro de la surface et deux pointeurs de type NOEUD_SEGMENT un pour le début de la liste et un pour la fin de la liste (voir figure 21).

Chaque type NOEUD_SEGMENT est une classe définie par un segment de type TSEGMENT et un pointeur sur le nœud suivant, le nœud précédent, le nœud suivant dans la région et le nœud précédent dans la région (voir figure 22).

Chaque type TSEGMENT est une classe définie par le numéro de la surface et par deux nœuds sommet (voir figure 23).

Chaque segment est caractérisé par une erreur de flèche et une longueur.

Parmi les fonctions de la classe LIST_SEGMENT :

- Créer liste segment () : création de la liste des segments pour une surface donnée.
- Ajouter segment : insertion d'un segment dans la liste des segments.
- Dessiner segment () : permet l'affichage de l'ensemble des segments.
- Double distance () : calcul de la distance entre deux sommets.

LIST_SEGMENT	
- *tete	: NOEUD_SEGMENT
- *fin	: NOEUD_SEGMENT
- num_surface	: int
+ creer liste segment ()	: LIST_SEGMENT
+ ajouter segment ()	: LIST_SEGMENT
+ dessiner segment ()	: void
+ vider segment ()	: void

Figure 21 : Classe de LIST_SEGMENT.

NOEUD_SEGMENT	
- sg	: SEGMENT
- *suivant	: NOEUD_SEGMENT
- *précédent	: NOEUD_SEGMENT
+ &operator= ()	: NOEUD_SEGMENT
+ operator== ()	: bool

Figure 22 : Classe de NOEUD_SEGMENT.

SEGMENT	
- *p	: NOEUD_SOMMET
- *q	: NOEUD_SOMMET
- hauteur	: double
- longueur	: double
- verifie_longueur	: bool
- verifie_hauteur	: bool
- numero surface	: int
<hr/>	
+ &operator= ()	: SEGMENT

Figure 23 : Classe de SEGMENT.

3. Liste triangle surface : est une liste des triangles générés propre pour chaque surface. Elle est de type LIST_TRIANGLE.

La classe LIST_TRIANGLE est définie par deux pointeurs de nœud triangle, un pour le début de la liste et un pour la fin de liste (voir figure 24).

Chaque NOEUD TRIANGLE est défini par un triangle de type TRIANGLE et deux pointeurs, un sur le nœud suivant et un sur le nœud précédent, et trois pointeurs sur ces voisins (voir figure 25).

Chaque TRIANGLE est donné par trois nœuds sommet, une erreur flèche au centre de gravité et un booléen de vérification des sept conditions pour accélérer les calculs (voir figure 26).

Parmi les fonctions de la classe LIST_TRIANGLE, nous avons :

- Triangulation adaptative () : permet la subdivision des triangles ne vérifiant pas une des sept conditions. Cette fonction utilise d'autres fonctions comme trois segment (), deux segment (), un segment () et centre de gravité ().
- Tester () : permet de vérifier si les voisins d'un triangle sont correct ou non.
- Dessiner (noeud triangle) : permet le dessin des triangles voisins.

LIST_TRIANGLE	
- *parc	: NOEUD_TRIANGLE
- *last	: NOEUD_TRIANGLE
<hr/>	
+ crier liste triangle ()	: LIST_TRIANGLE
+ dessiner triangle ()	: void
+ ajouter triangle ()	: LIST_TRIANGLE
+ triangulation adaptative ()	: LIST_TRIANGLE
+ &trois segment ()	: LIST_TRIANGLE
+ &deux segment ()	: LIST_TRIANGLE
+ &un segment ()	: LIST_TRIANGLE
+ ¢re de gravité ()	: LIST_TRIANGLE
+ tester ()	: void

Figure 24 : Classe de LIST_TRIANGLE.

NOEUD_TRIANGLE	
- T	: TRIANGLE
- *suivant	: NOEUD_TRIANGLE
- *precedent	: NOEUD_TRIANGLE
- *voisin1	: NOEUD_TRIANGLE
- *voisin2	: NOEUD_TRIANGLE
- *voisin3	: NOEUD_TRIANGLE

Figure 25 : Classe de NOEUD_TRIANGLE.

TRINGLE	
- *sg1	: NOEUD_SEGMENT
- *sg2	: NOEUD_SEGMENT
- *sg3	: NOEUD_SEGMENT
- erreur cg	: double
- vérifie cg	: bool
- vérifie 7cond	: bool
- num	: int
+ crier triangle () : TRIANGLE	

Figure 26 : Classe de TRIANGLE.

4. Liste d'information : est une liste de type INFORMATION qui permet de stocker toutes les informations obtenues à partir du calcul de la triangulation uniforme et de la triangulation adaptative (voir figure 27).

INFORMATION	
- num surface	: int
- nombre sommet	: int
- nombre triangle	: int
- nombre segment	: int

Figure 27 : Classe d'INFORMATION.

5. Liste rayon d'outil : est une liste de type RAYON_OUTIL qui permet de stocker le rayon d'outil théorique, optimum et choisis ainsi que sa longueur (voir figure 28).

RAYON_D'OUTIL	
- numero surface	: int
- Rayon théorique	: double
- Rayon choisis	: double
- Rayon optimum	: double
+ calculer rayon théorique () : void	

Figure 28 : Classe de RAYON D'OUTIL.

6. Plans verticaux : est une liste de type plan vertical qui permet de stocker les points d'intersections (points de passage du centre d'outil) (voir figure 29). Parmi les fonctions de cette classe, nous avons :

- Générer chemin () : permet de calculer, de trier les points d'intersection et de générer le chemin d'outil.
- Dessiner plan vertical () : permet de dessiner le plan vertical en différentes positions en fonction de son inclinaison et de sa position par rapport à la surface.

plan vertical	
- D1	: double
- s1	: double
- s2	: double
- s3	: double
- num_surf	: int
- taille	: int
- chemin	: LIST_SOMMET
+ générer chemin ()	: void
+ dessiner plan vertical ()	: void

Figure 29 : Classe de PLAN VERTICAL.

7. Liste région surface : est une liste de type LIST_REGION qui permet de stocker les régions pour chaque surface (voir figure 30).

LIST_REGION	
- x_max	: double
- x_min	: double
- y_max	: double
- y_min	: double
- z_max	: double
- z_min	: double
- numero surface	: int
- région surface	: LES_REGION
+ crier les régions ()	: void
+ visualiser région ()	: void
+ visualiser limite surface ()	: void
+ visualiser point région ()	: void
+ visualiser segment région ()	: void

Figure.30 : Classe de LIST_REGION.

Parmi les attributs de cette classe une région de la surface de type REGION qui permet de stocker les sommets de cette région.

La figure 31 représente les attributs du type LES REGION.

LES REGION	
- X_max	: double
- X_min	: double
- Y_max	: double
- Y_min	: double
- Z_max	: double
- Z_min	: double
- sommets	: LIST_SOMMET

Figure 31 : Classe de LES REGION.

Parmi les fonctions de la classe LIST_REGION, nous avons :

- Visualiser les régions () : permet la visualisation de l'ensemble des régions.
- Visualiser les points de région () : permet la visualisation des points d'une région.
- Visualiser les segments de région () : permet la visualisation des segments d'une région.

8. Liste points d'interférence : est une liste de type INTERFERENCE_POINT qui permet de stocker les points qui créent un problème d'interférence (voir figure 32).

Parmi ses fonctions :

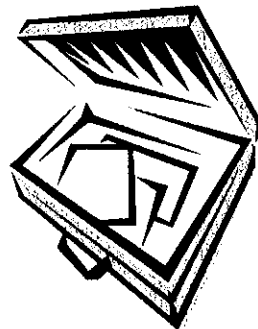
- Distance () : qui calcule la distance entre le centre de la sphère et les sommets des régions afin de déterminer les points d'interférence.
- Position d'outil () : elle permet de pointer sur un sommet de la liste des sommets et de calculer le centre d'outil et le comparer avec l'ensemble des sommets des régions appartenant à l'enveloppe d'outil en faisant appel à la fonction comparer_avec_surface () .
- Comparer_avec_surface () : elle compare un sommet avec les sommets des régions ; si ce dernier crée une interférence il sera inséré dans la liste des sommets d'interférence. Cette fonction fait appel à une autre fonction pour calculer le nouveau rayon qui élimine les interférences.
- Correction d'interférence () : permet de calculer le rayon d'outil qui élimine les interférences.

POINT_D'INTERFERENCE	
- sommet_interference	: int
- num_surface	: int
- R_theorique	: int
- X_CE	: int
- Y_CE	: int
- Z_CE	: int
<hr/>	
+ distance cg ()	: double
+ position d'outil ()	: void
+ correction ()	: void
+ Comparer_avec_surface ()	: void

Figure 32 : Classe de POINT_D'INTERFERENCE.

IV. CONCLUSION :

Dans ce chapitre nous avons défini notre système en commençant par la problématique et les différents objectifs à atteindre, ensuite les solutions que nous avons proposées avec les différents diagrammes de conception. Dans le chapitre suivant, nous allons présenter l'implémentation de notre application.



CHAPITRE 6
IMPLÉMENTATION

I. INTRODUCTION :

Après l'étude de la conception de notre application ainsi que les objectifs fixés et les solutions proposées dans le chapitre précédent, nous allons maintenant présenter notre application logicielle où son implémentation a été faite en utilisant un langage évolué doté d'une interface qui facilite à l'utilisateur la manipulation et l'exécution des différentes tâches. En plus, nous allons faire une description des principaux algorithmes développés.

II. PRESENTATION DE L'APPLICATION :

Pour atteindre les différents objectifs fixés, nous avons intégrés les modules que nous avons développés à une application logicielle graphique de FAO qui permet la génération du trajet d'outils pour l'usinage en finition des modèles CAO en utilisant la méthode I.T.O. A cet effet, nous avons utilisés un ensemble de fenêtres Windows permettant à l'utilisateur d'interagir avec l'application en lui permettant de visualiser tous les objets manipulés via la bibliothèque graphique OpenGL.

III. IMPLEMENTATION :

III.1. Fenêtre principale :

Elle est composée de deux parties (voir figure 1); une pour l'affichage des différents objets créés et l'autre pour la manipulation. Cette dernière est constituée d'une barre de menu et d'un ensemble de boutons de manipulation des différents paramètres des modèles CAO des courbes et des surfaces (création d'une surface, suppression d'une surface, modification des points de contrôle, ... etc.).

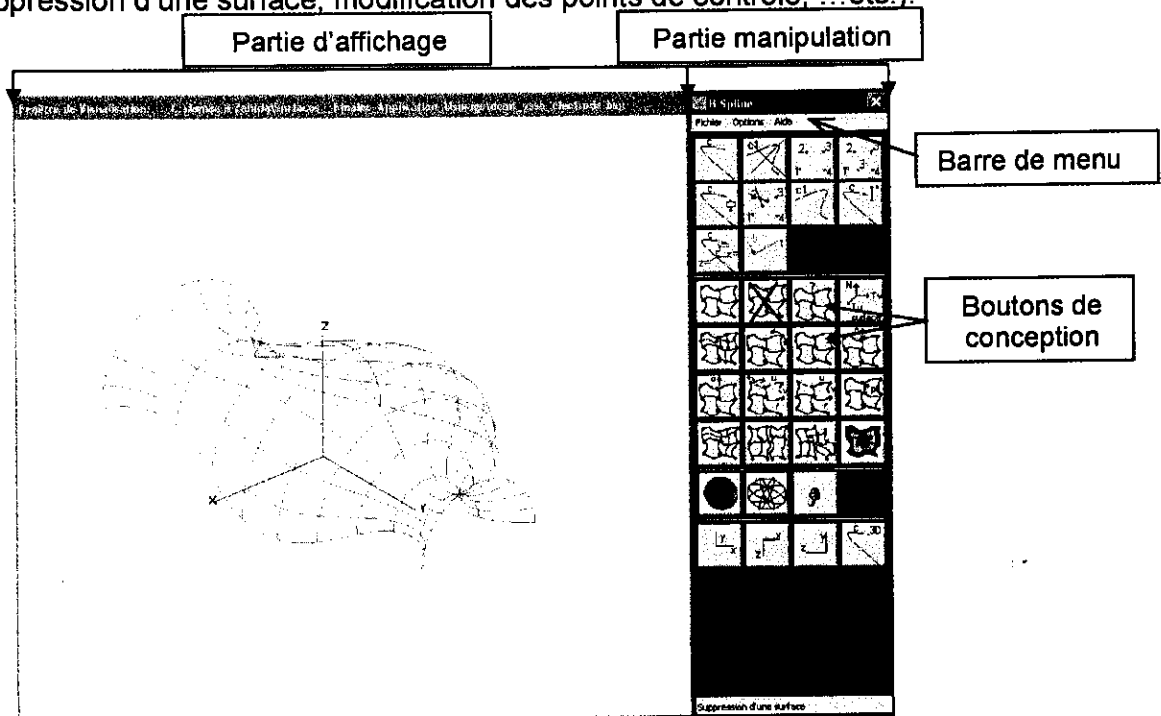


Figure 1: Fenêtre principale.

III.2. Barre du menu principal :

Elle est composée de trois rubriques qui sont :

- Rubrique Fichier : comporte toutes les fonctionnalités de manipulation des fichiers (ouverture, création, sauvegarde, ...etc.).
- Rubrique Option : permet la modification des différents paramètres des courbes ainsi que les paramètres des surfaces et les paramètres d'usinage et permet aussi l'usinage en finition par la méthode I.T.O.
- Rubrique Aide : pour l'affichage des informations concernant la réalisation et l'utilisation de l'application.

III.3. Rubrique d'usinage en finition par la méthode ITO :

Les modules que nous avons intégrés à l'application sont accessibles à partir du menu dans la rubrique Option, Usinage et ensuite « finition par la méthode I.T.O » (voir figure 2). Pour que l'utilisateur puisse accéder aux différentes fonctions, il faut d'abord qu'il ouvre un fichier contenant les modèles CAO des surfaces à usiner.

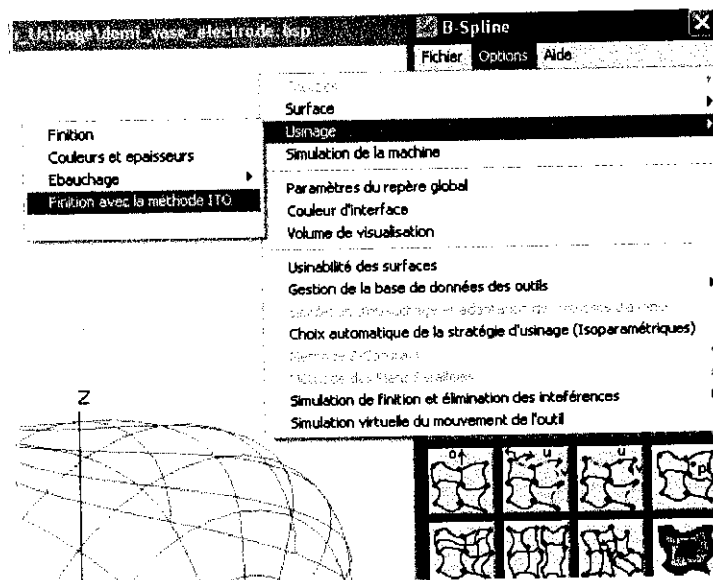


Figure 2 : Accès à l'usinage en finition par la méthode ITO.

IV. PRESENTATION DES FENETRES :

Nous allons présenter maintenant toutes les fenêtres créées dans notre travail.

IV 1. Fenêtre de triangulation des surfaces :

C'est la première fenêtre qui apparaît à l'utilisateur (voir figure 3) qui permet de générer une triangulation uniforme et ensuite adaptative des surfaces à usiner. De

même, elle permet aussi la visualisation, des sommets, des segments et des triangles ainsi que les voisins d'un triangle donné.

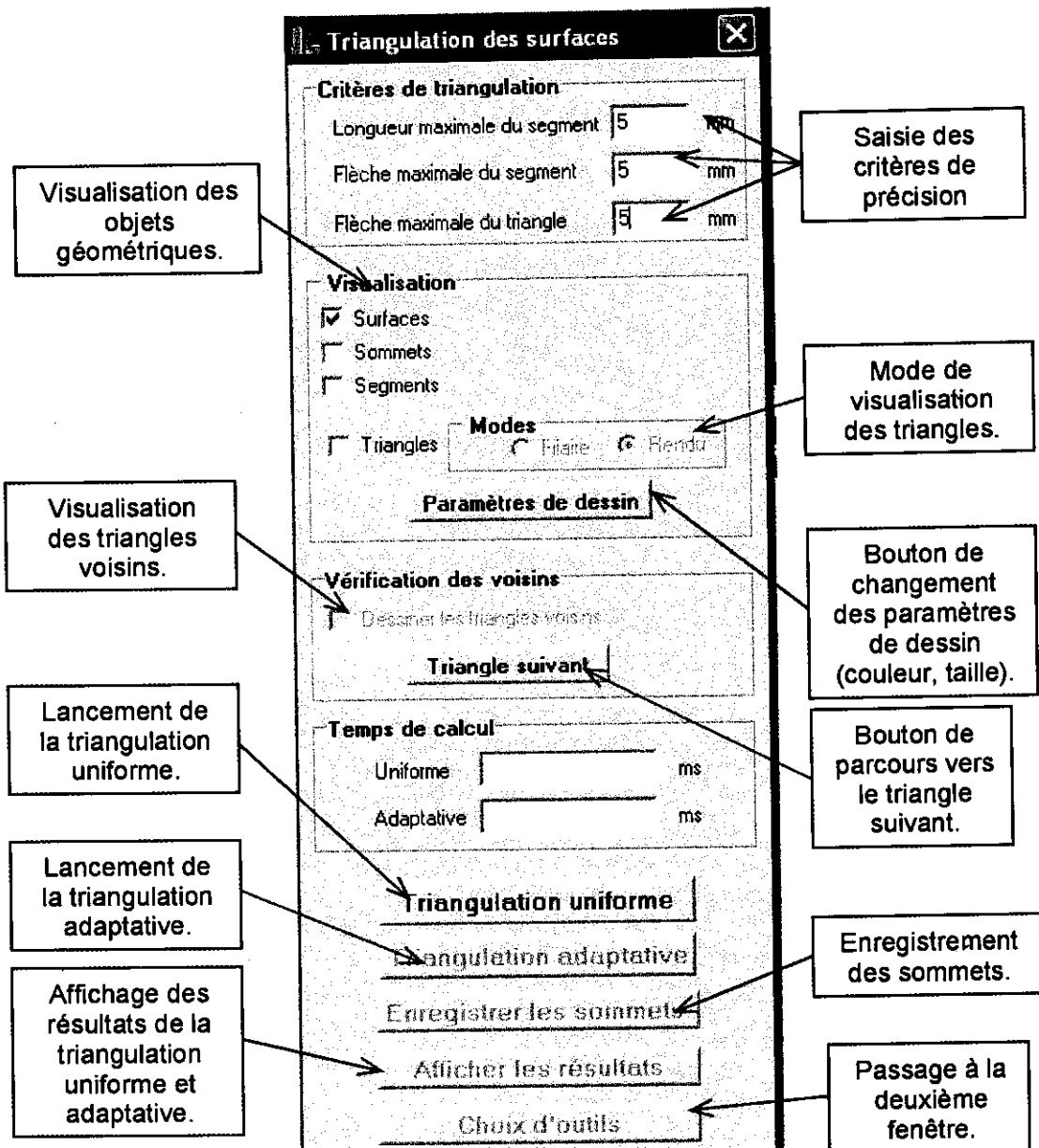


Figure 3: Fenêtre de triangulation des surfaces.

Avant de lancer la triangulation, l'utilisateur doit spécifier en premier lieu les différents critères de triangulation (paramètres de précision) :

- Longueur maximale du segment (voir figure 4.a),
- Flèche maximale du segment (voir figure 4.b),
- Flèche maximale du triangle (voir figure 4.c).

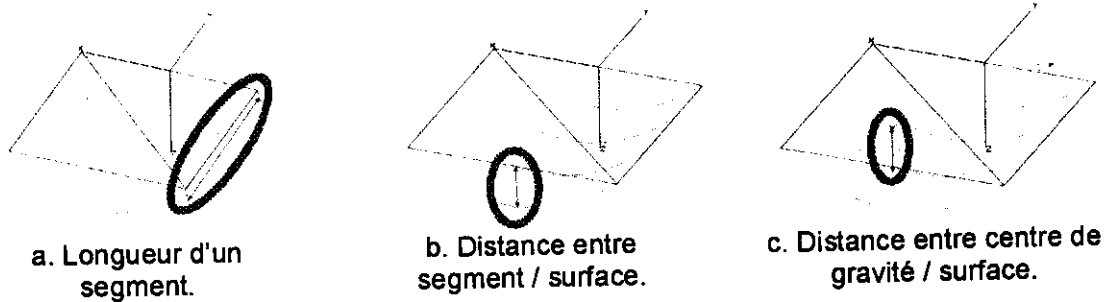


Figure 4 : Critères de triangulation.

Une fois que ces paramètres sont introduits, l'utilisateur peut lancer la triangulation uniforme à partir du bouton « triangulation uniforme ». C'est une première approximation de la surface.

Avant d'entamer la triangulation uniforme, le nombre de lignes et de colonnes pour chaque surface sont calculés en passant par les étapes suivantes :

- Nous supposons que le premier sommet P_0 possède les coordonnées $(0,0)$ dans le plan paramétrique (\vec{u}, \vec{v}) (voir figure 5),
- Pour chaque direction (\vec{u} ou \vec{v} du plan paramétrique), nous choisissons un deuxième sommet P_1 suivant \vec{u} de coordonnée 0.5 ($dist_u$) et un autre sommet P_2 suivant \vec{v} de coordonnée 0.5 ($dist_v$) (voir figure 5),
- Pour chaque direction (\vec{u} ou \vec{v}), nous calculons la longueur et l'erreur de flèche du segment formé par P_0P_1 et P_0P_2 ,
- La longueur et la flèche sont comparées avec les valeurs introduites par l'utilisateur. Si l'une des valeurs ne vérifie pas la contrainte imposée, nous utilisons la dichotomie pour affiner les calculs et cela par le choix d'un point au milieu de P_0P_1 ou P_0P_2 dans le plan paramétrique pour refaire les calculs.

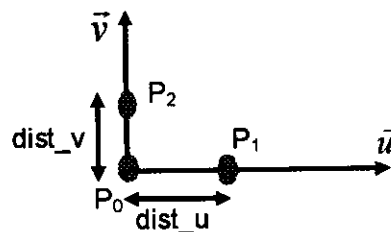


Figure 5 : Calcul du nombre de lignes et de colonnes.

Une fois que le pas suivant \vec{u} et le pas suivant \vec{v} sont calculés, les coordonnées des sommets, les segments et les triangles sont générés dans le plan paramétrique et par suite sont transformés vers l'espace 3D. La triangulation uniforme est faite par l'algorithme suivant :

Algorithme : Triangulation uniforme.

1. Création de la liste des sommets :

- Soit pas_u et pas_v , u et v des réels tels que :
 $pas_u = 1/nbr_colonnes$, $pas_v = 1/nbr_lignes$; u et v initialisés à 0,
- Pour i de 0 à nbr_lignes
 $v = pas_v * i$,
- Pour j de 0 à $nbr_colonnes$
 $u = pas_u * j$,
- Créer un sommet de coordonnées (u, v) .

Le principe de calcul est illustré par la figure 6.

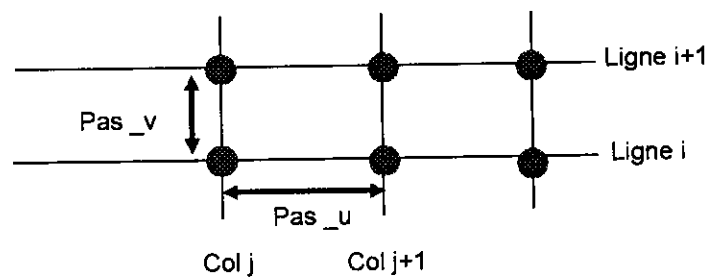


Figure 6 : Création des sommets.

2. Création de la liste des segments :

- Trouver les quatre sommets ayant deux à deux les mêmes coordonnées suivant \bar{u} et suivant \bar{v} (voir figure 7).
- Créer les cinq segments formés par ces sommets et les insérer dans la liste des segments (voir figure 7).
- Refaire le travail jusqu'à la fin de la liste des sommets.

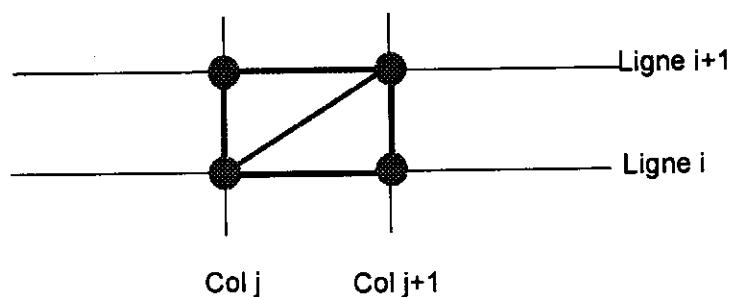


Figure 7 : Création des segments.

γ. Création de la liste des triangles :

- Trouver tous les segments possédant un sommet en commun (voir figure 8),
- Créer les deux triangles formés par ces segments,
- Refaire le calcul jusqu'à la fin de la liste des sommets et générer à chaque fois les triangles voisins (voir figure 8).

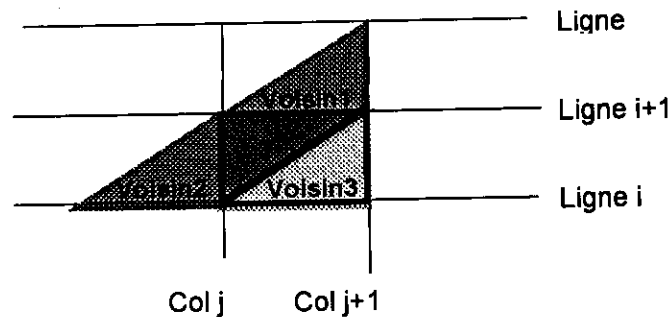


Figure 8 : Génération des triangles.

A la fin de la triangulation uniforme, le bouton « triangulation adaptative » est activé et il est donc possible de passer à l'étape d'amélioration des résultats d'approximation. L'algorithme suivant donne les principales étapes de la triangulation adaptative :

Algorithme : Triangulation adaptative

- Vérifier si tous les triangles respectent les critères de précision spécifiés par l'utilisateur.
- Pour chaque triangle qui ne vérifie pas une des conditions :
 1. Si les trois segments ne vérifient pas une des conditions de longueur ou de flèche, chaque segment est subdivisé en deux segments.
 2. Si la condition du centre de gravité n'est pas vérifiée, chaque segment est subdivisé en deux segments avec l'ajout d'un sommet au centre de gravité du triangle.
 3. Pour deux segments du triangle qui ne vérifient pas une des deux conditions, chacun des deux segments est subdivisé en deux segments.
 4. Si un segment ne vérifie pas une des deux conditions, il est subdivisé en deux segments.

Pour chaque subdivision, de nouveaux sommets, segments et triangles sont créés et sont ajoutés aux différentes listes et les voisins de chaque triangles sont mis à jour.

L'utilisateur a la possibilité de visualiser les différents objets de la triangulation (sommets, segments, triangles) pendant la triangulation s'il a coché les cases de visualisation avant le lancement d'une des triangulations, comme il peut les visualiser

à la fin des calculs. Il a aussi la possibilité d'afficher les triangles voisins de chaque triangle de la liste des triangles.

A la fin de chaque type de triangulation, l'utilisateur peut enregistrer les sommets en cliquant sur le bouton « enregistrer les sommets » ou afficher les différents résultats (voir figure 9) concernant le nombre de sommets, segments, triangles ...etc., en appuyant sur le bouton « afficher les résultats », ou passer à la fenêtre suivante en choisissant le bouton « choix d'outils ». Dans la fenêtre des résultats, il est possible de voir le résultat global pour l'ensemble des surfaces ou les résultats d'une surface donnée.

	Surface 0		
	Uniforme	Adaptative	Rapport
nombre de lignes	4	/	/
nombre de colonnes	16	/	/
nombre de sommets	85	276	3.247
nombre de segments	212	775	3.656
nombre de triangles	128	500	3.906
nombre de segments dont la longueur est vérifiée	114	775	6.798

Figure 9 : Fenêtre d'affichage des résultats de la triangulation.

Dans la fenêtre de triangulation, l'utilisateur peut aussi changer les paramètres de dessin (couleurs et épaisseurs) des différents objets en cliquant sur le bouton « paramètres de dessin » (voir figure 10).

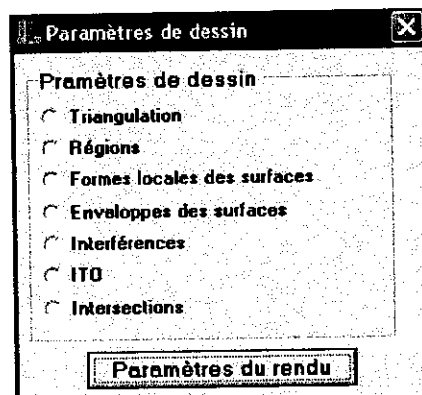


Figure 10 : Fenêtre paramètres de dessin.

IV.2. Fenêtre du choix d'outils :

Cette fenêtre permet le calcul du rayon théorique de l'outil, la génération des régions et la détection et la correction des interférences ainsi que le calcul du rayon optimum (voir figure 11).

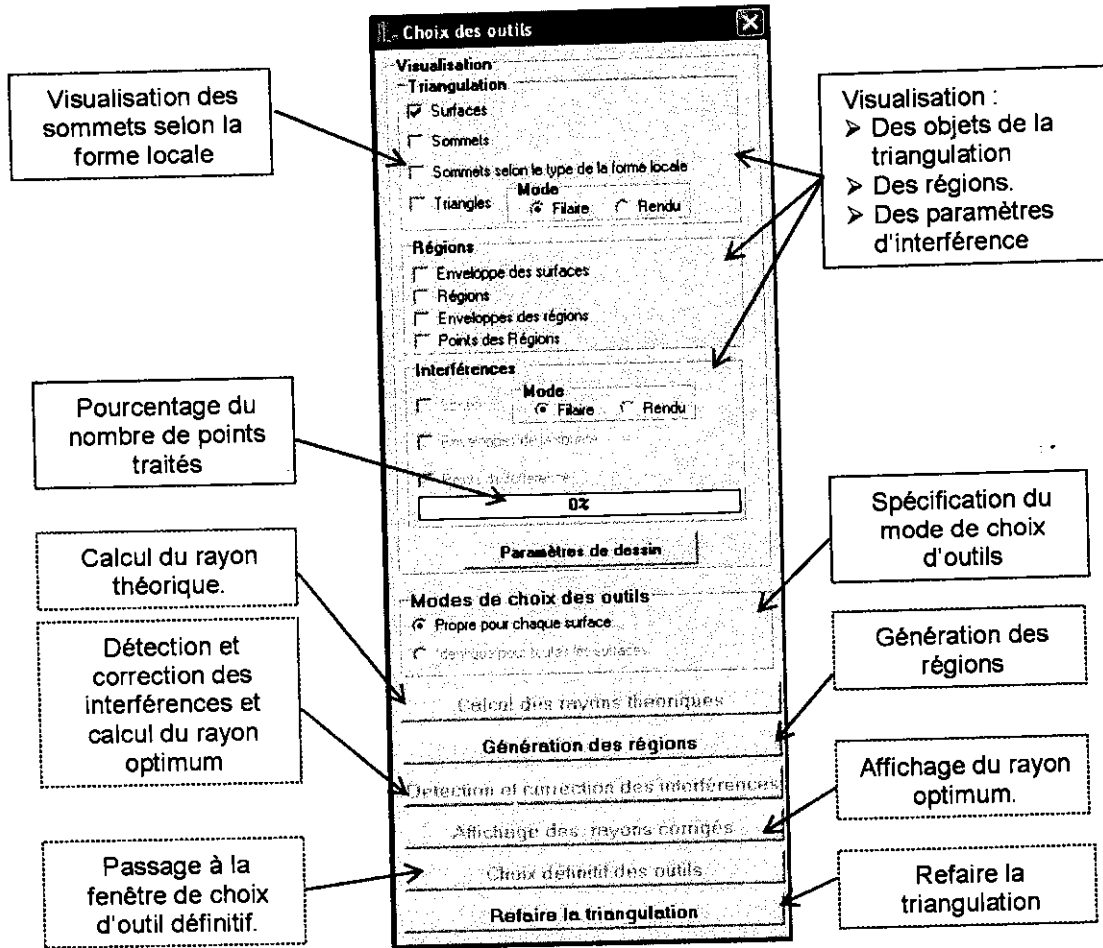


Figure 11 : Fenêtre choix d'outils.

IV.2.1. Calcul du rayon théorique :

A partir des rayons de courbures maximum et minimum de chaque sommet de la liste des sommets, nous pouvons définir le rayon théorique comme étant le plus petit rayon trouvé.

IV.2.2. Visualisation des sommets selon la forme locale :

L'utilisateur a la possibilité de visualiser les sommets selon leurs formes locales à la fin de calcul du rayon théorique s'il a coché la case prise dans la partie de visualisation. La figure 12 montre les différentes couleurs des formes locales et comme nous pouvons avoir six formes locales, cette fenêtre affiche six rectangles ou chacun est associé à un type de forme locale. Les couleurs des régions peuvent être modifiées en cliquant sur le bouton « paramètres de dessin ».

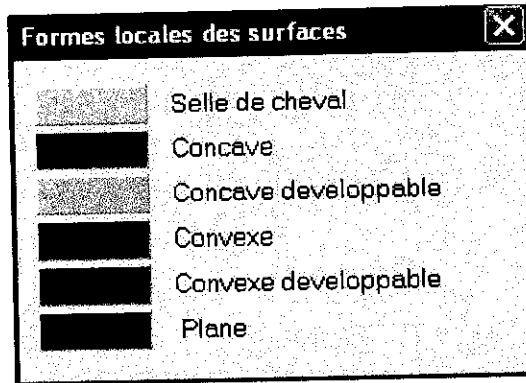


Figure 12 : Couleurs des formes locales de la surface.

IV.2.3. Génération des régions :

Afin d'accélérer les calculs de détection et de correction des inférences et par conséquent minimiser les temps de calcul, les sommets de chaque surface sont groupés dans des régions (voir figure 13).

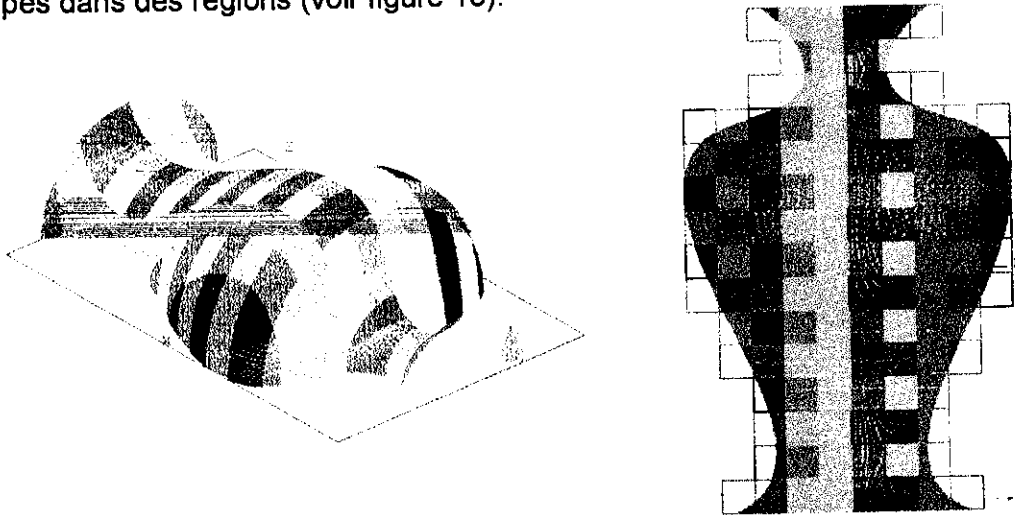


Figure 13: Régions de la surface avec ces points.

Nous définissons une région comme étant l'ensemble des sommets appartenant au même intervalle délimité par x_{min} , x_{max} , y_{min} et y_{max} . Pour générer les régions, nous avons appliqués l'algorithme suivant :

Algorithme : Génération des régions de points

- Le nombre de colonnes est égal à la longueur de la surface suivant \bar{x} divisée par le rayon théorique plus un,
- Le nombre de lignes est égal à la longueur de la surface suivant \bar{y} divisée par le rayon théorique plus un,

- La taille de la région suivant \bar{x} est la longueur de la surface suivant \bar{x} divisée par le nombre de colonnes,
- La taille de la région suivant \bar{y} est la longueur de la surface suivant \bar{y} divisée par le nombre de lignes,
- Pour chaque sommet de coordonnées x et y de la liste des sommets, ces positions i et j sont données par :
 - La position $j = x / \text{longueur de la région suivant } \bar{x}$
 - La position $i = y / \text{longueur de la région suivant } \bar{y}$

IV.2.4. Détection et correction des interférences :

Dans cette étape, nous cherchons dans la liste des sommets ceux qui créent des problèmes d'interférence et par la suite nous procédons à la correction du rayon d'outil par le calcul du rayon optimum. Un point crée une interférence s'il appartient à la sphère tangente à la surface en un sommet donné.

Les étapes de détection des interférences sont détaillées dans l'algorithme suivant :

Algorithme : Détection des interférences.

- Pour chaque sommet S de la liste des sommets (point de contact) :
 1. Calculer les coordonnées du centre outil,
 2. Déterminer l'enveloppe de la sphère,
 3. Vérifier si la surface peut avoir des interférences (si elle n'est ni plane, ni convexe, ni convexe développable),
 4. Tester s'il existe un chevauchement entre l'enveloppe de la sphère et les régions de la surface,
 5. Si oui, trouver les régions $[i, j]$ qui se chevauchent avec l'enveloppe de la sphère,
 6. Pour chaque sommet q appartenant à une de ces régions faire :
 7. Si q est différent de S et la distance entre les deux sommets S et q est inférieure au rayon théorique :
 - a. Ajouter le sommet S à la liste des interférences,
 - b. Appeler la fonction de correction du rayon,
 - c. Passer au sommet suivant dans la région.
 8. Refaire le calcul pour chaque région et avec les surfaces voisines.

La figure 14 explique ce principe de calcul.

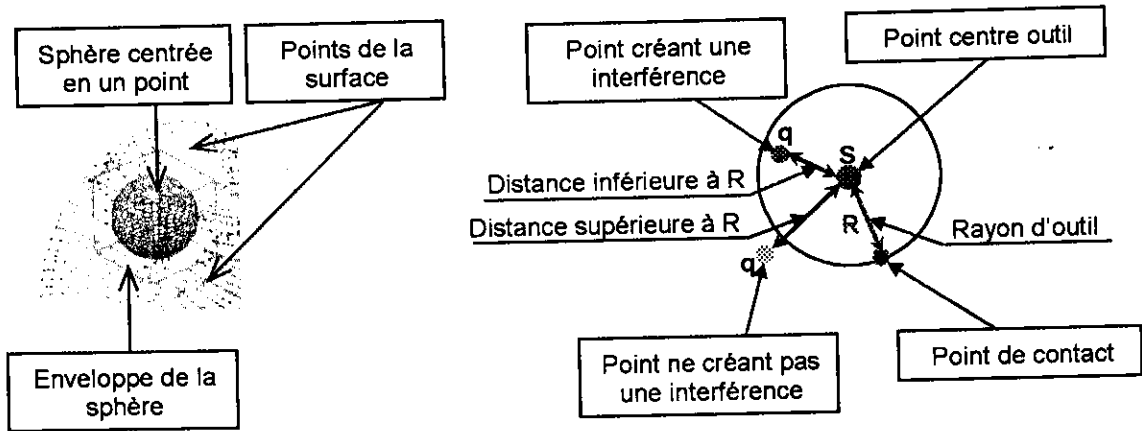


Figure 14 : Détection des interférences.

A chaque étape de détection des interférences, le nouveau rayon corrigé est déterminé par l'algorithme suivant :

Algorithme : Correction des interférences.

- Pour chaque sommet q créant une interférence avec le sommet S :
 1. Trouver le milieu M de la droite (Sq) ,
 2. Calculer le vecteur \overline{Sq}
 3. Déterminer l'équation du plan passant par le point M et ayant comme vecteur normal le vecteur \overline{Sq} ,
 4. Calculer le point d'intersection entre le plan déterminé dans l'étape précédente et la droite passant par S et ayant comme vecteur directeur le vecteur normal en S . Le point calculé est le nouveau centre outil C_E ,
 5. Calculer la distance entre S et C_E qui est le nouveau rayon corrigé.

La figure 15 exprime les détails de calcul.

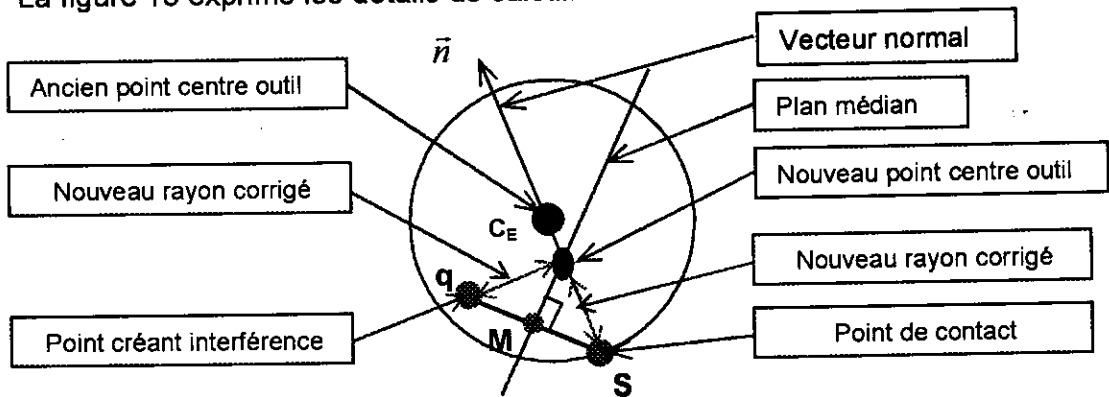


Figure 15 : Correction des interférences.

Nous avons maintenant le nouveau rayon corrigé pour chaque surface qui sera affiché en cliquant sur le bouton « affichage des rayons corrigés » (voir figure 16).

N° Surface	Rayon théorique	Rayon utilisé	Type de surface
0	5.0640	5.0640	La Surface est composée de plusieurs formes locales

Figure 16 : Fenêtre d'affichage des rayons.

IV.3. Fenêtre du choix d'outils final :

Dans cette fenêtre, l'utilisateur choisit le rayon final qu'il va utiliser dans l'usinage (voir figure 17). Une fois que le rayon final est choisi, l'utilisateur a la possibilité soit de refaire le choix d'outil théorique ou final ou modifier les paramètres de dessin avec l'activation du bouton « paramètre d'usinage ».

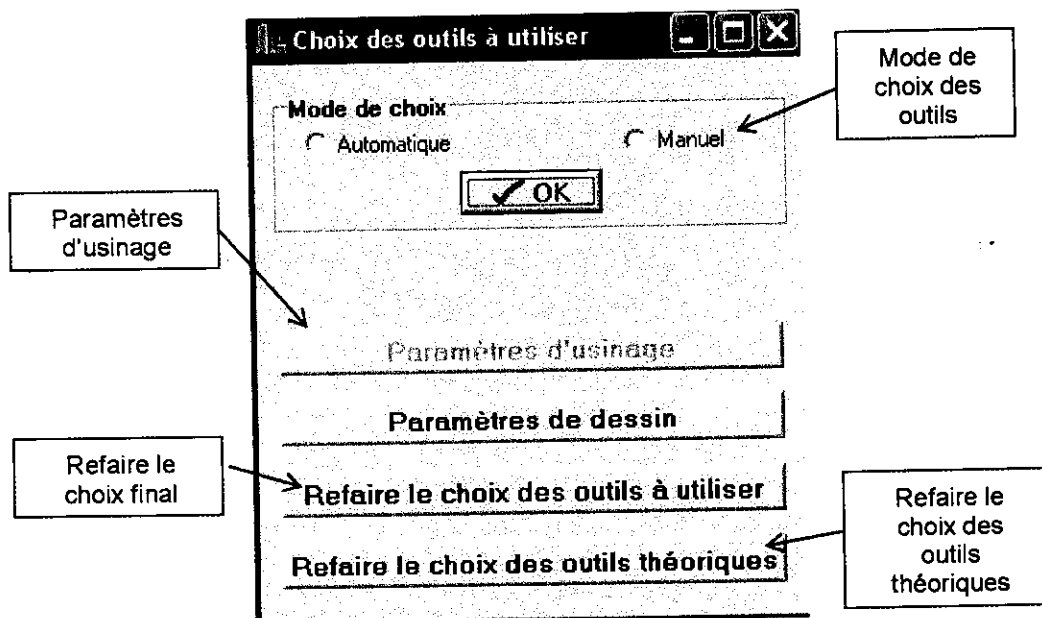
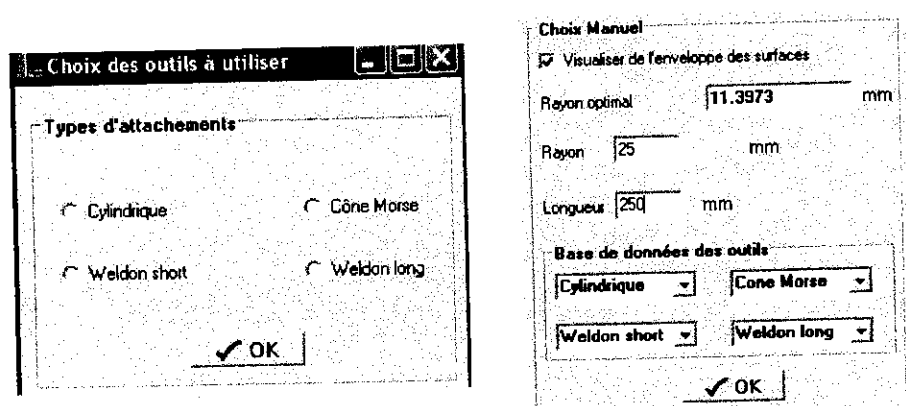


Figure 17 : Fenêtre de choix final des outils.

Le choix des outils peut être fait soit automatiquement à partir de la base de données des outils en spécifiant le type d'attachement (voir figure 18.a), soit manuellement en consultant la base de données des outils ou en spécifiant d'autres valeurs du rayon et de longueur (voir figure 18.b).



a. Choix automatique.

b. Choix manuel.

Figure 18: Modes de choix d'outils.

Après le choix des outils, le bouton paramètres d'usinage permet de spécifier les autres paramètres nécessaires pour l'usinage (vitesse d'avance, vitesse d'engagement, ... etc.) et le passage vers la méthode ITO (voir figure 19) en générant de nouvelles régions des points avec le nouveau rayon choisi ainsi que les régions des segments (voir figure 20) afin d'accélérer le calcul des points intersections et par conséquent la génération du trajet d'outils.

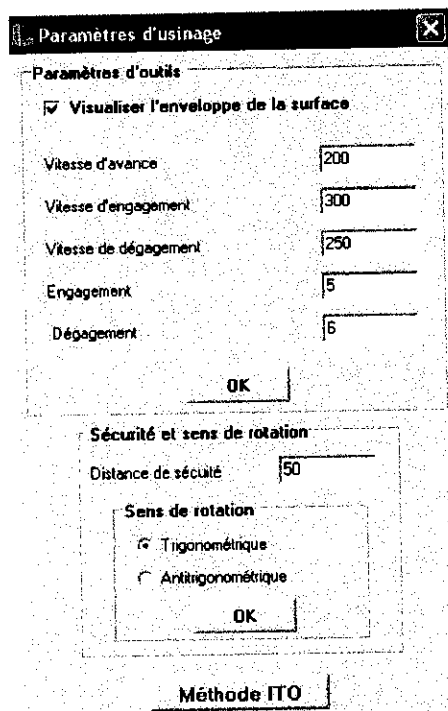


Figure 19 : Fenêtre des paramètres d'usinage.

Algorithme : Génération des régions de segments

- Le nombre de colonnes, de lignes et la taille de la région suivant \bar{x} ou \bar{y} sont déterminés de la même manière qu'avec les régions des points,
- Pour chaque sommet d'un segment, les positions i et j sont données par :
 - La position $j = x / \text{longueur de la région suivant } \bar{x}$
 - La position $i = y / \text{longueur de la région suivant } \bar{y}$
- Si les deux sommets appartiennent à la même région, ce segment est inséré dans la liste des segments de la région (i, j) .
- Si un des deux sommets n'appartient pas à la même région, le segment est inséré dans une autre liste de segments.

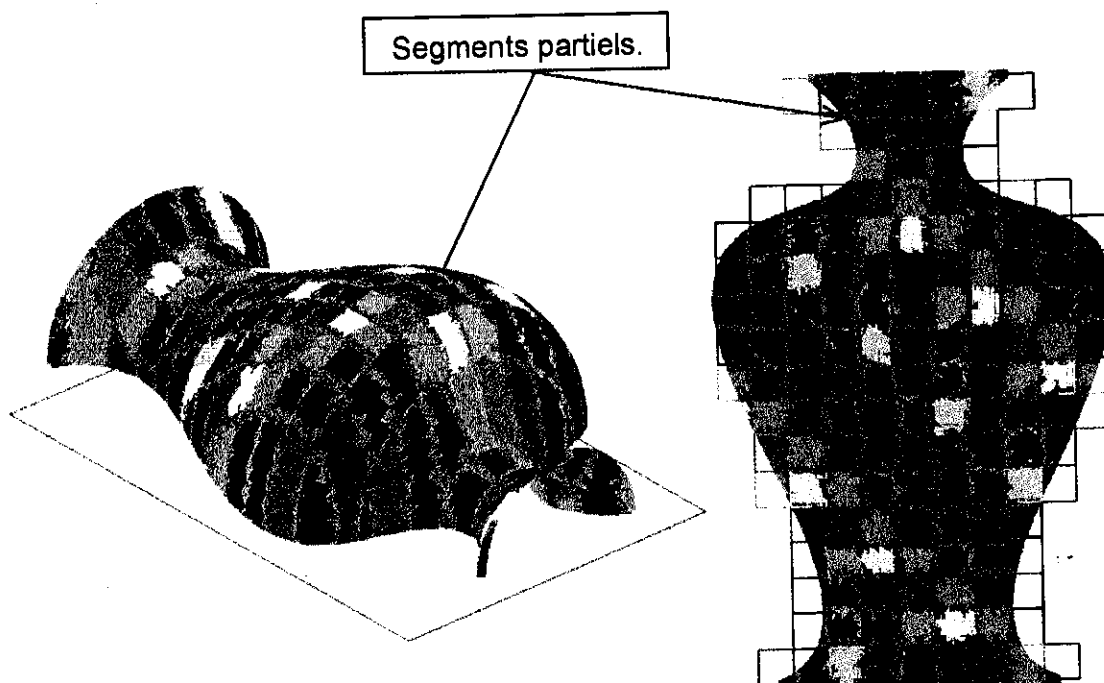


Figure 20: Régions des segments.

IV.4. Fenêtre de la méthode I.T.O :

Dans cette fenêtre, nous générons la surface des centres outil par la méthode ITO en utilisant l'algorithme suivant :

Algorithme : Méthode I.T.O

1. Pour chaque sommet de la liste des sommets, nous traçons une droite parallèle à l'axe \bar{z} (voir figure 21.a),
2. Pour chaque sommet nous centrons une sphère ayant le rayon que l'outil final (voir figure 21.b).
3. Pour chaque sommet d'une région appartenant à l'enveloppe de la sphère, nous calculons les différentes intersections entre droites et sphère (voir figure 21.c),
4. Pour chaque droite nous prenons le point le plus haut et l'ensemble de tous les points forme la surface centre outil (voir figure 21.d).

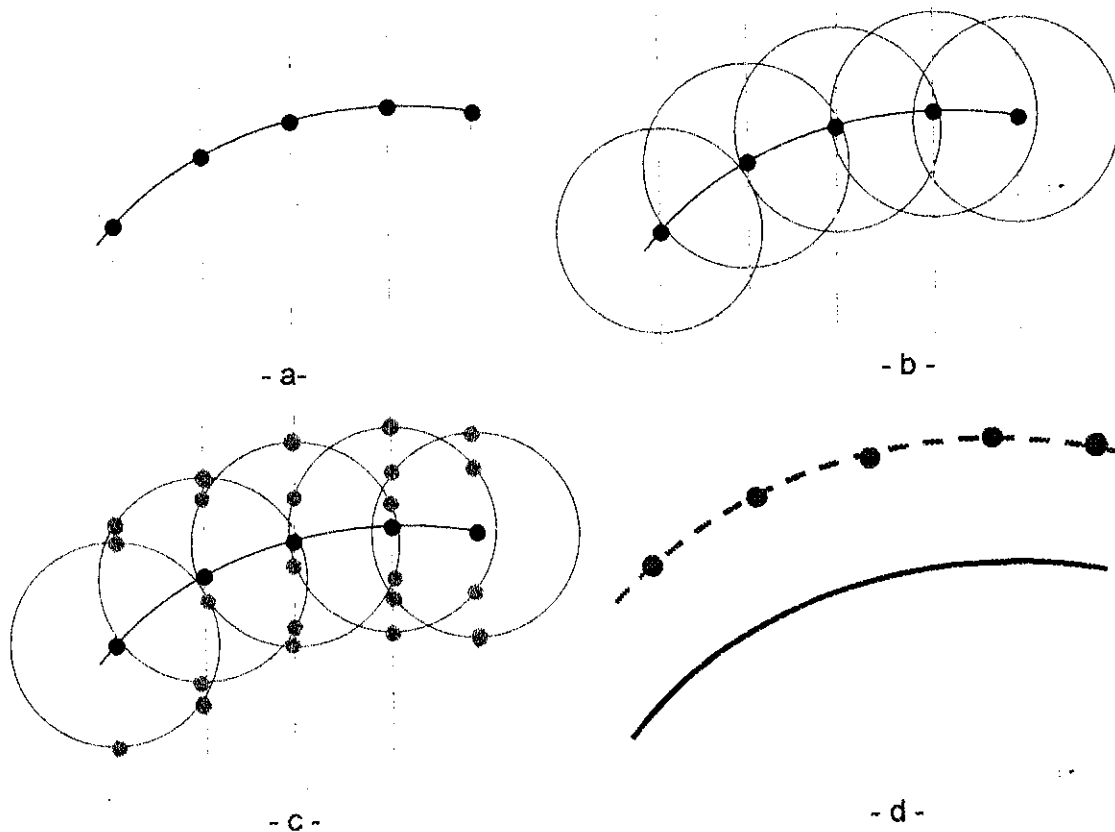


Figure 21: Principe de génération de la surface centre outil par la méthode I.T.O.

À la fin, l'utilisateur a la possibilité de refaire le choix d'outil final ou changer les paramètres de dessin et le bouton « usinage par plan parallèles » est activé pour le passage vers la fenêtre suivante (voir figure 22)

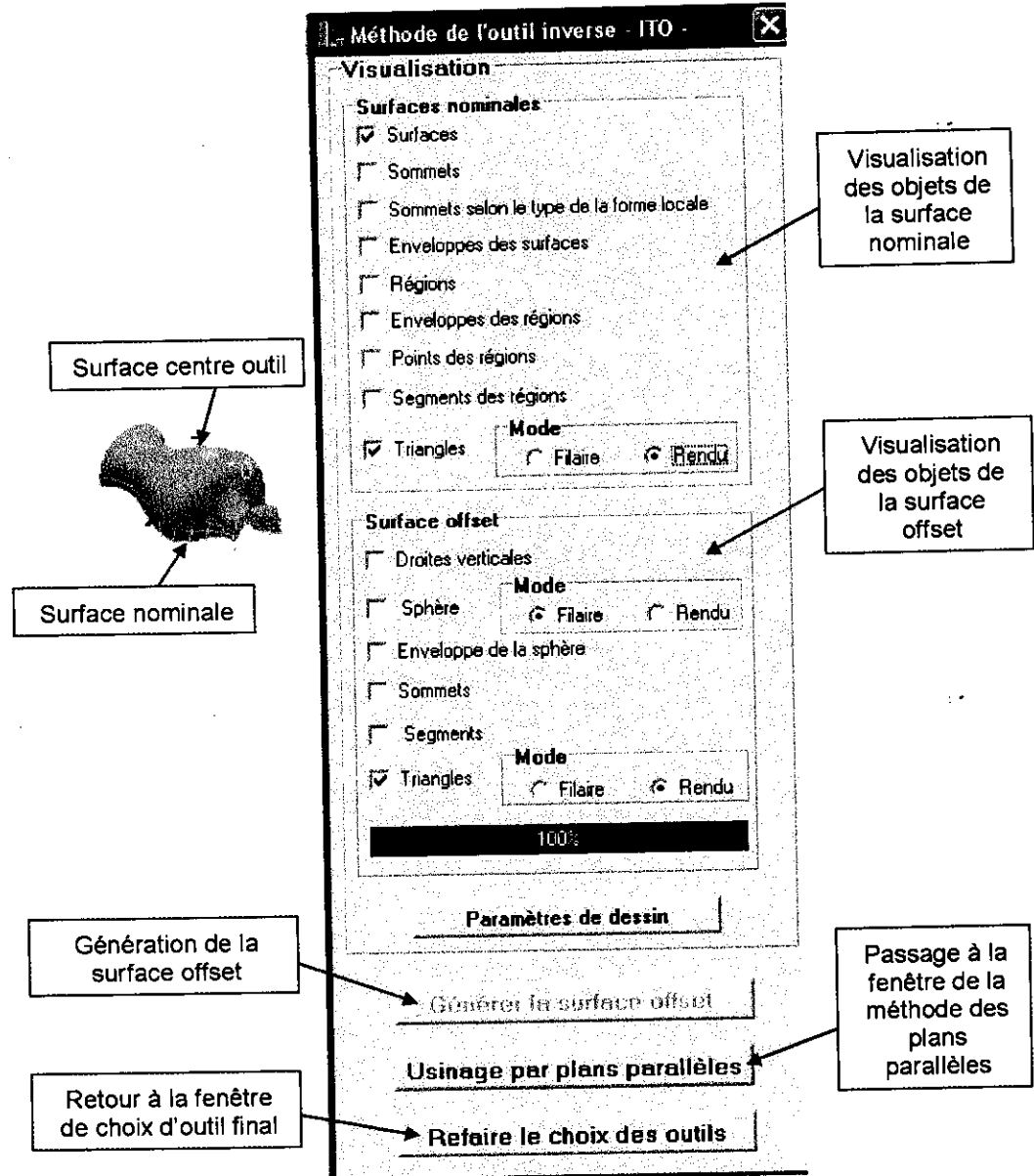


Figure 22 : Fenêtre Méthode de l'outil inverse (I.T.O).

IV.5. Paramètres de la méthode des plans parallèles :

Dans cette fenêtre, l'utilisateur doit spécifier les paramètres d'usinage de la méthode des plans parallèles : la distance entre deux plans successifs et l'angle d'orientation du plan vertical par rapport au plan XZ (figure 23). Ces paramètres peuvent être identiques pour l'ensemble des surfaces ou spécifiées pour chaque surface.

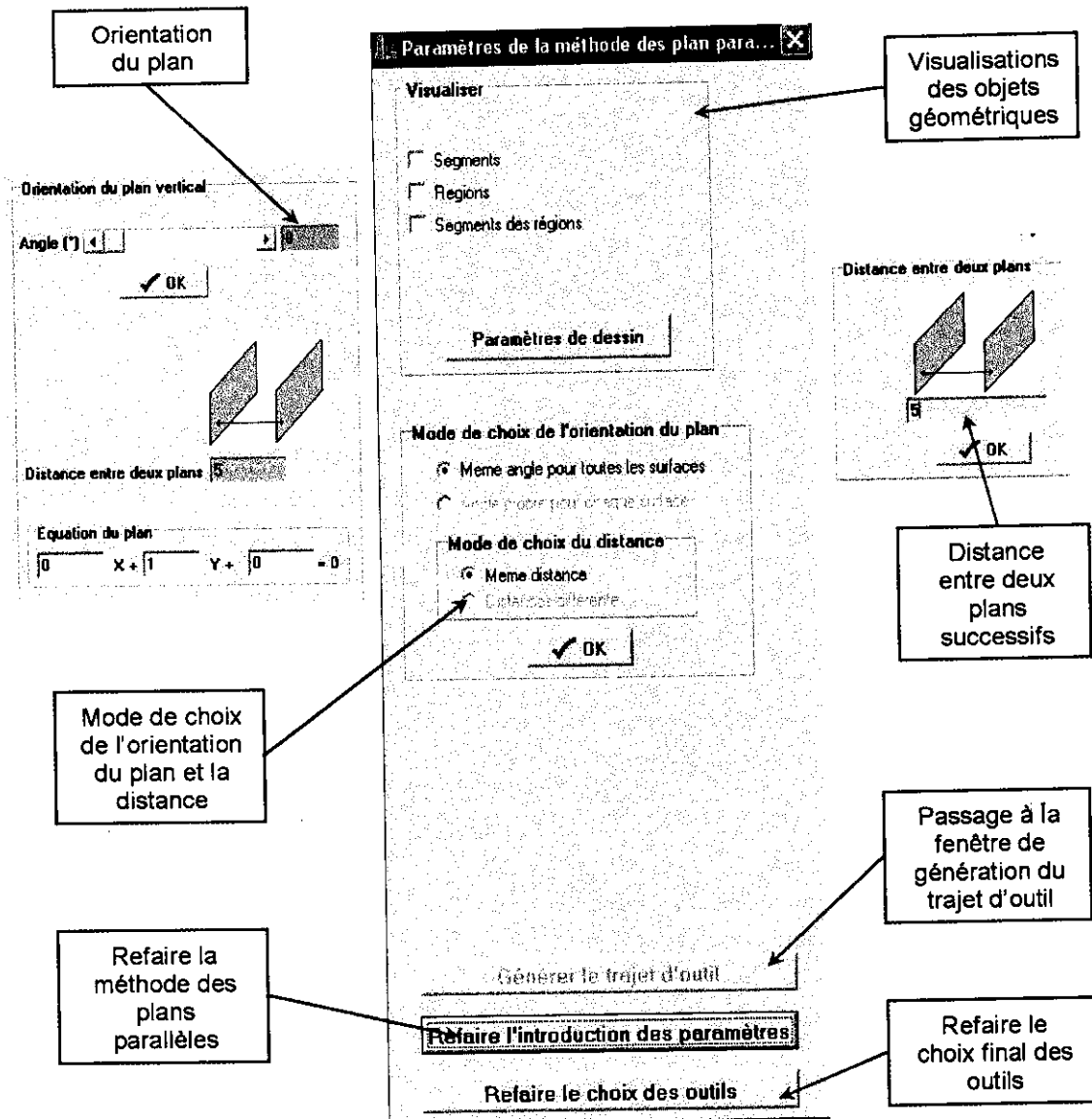


Figure 23 : Fenêtre paramètre de la méthode des plans parallèles.

Une fois que les paramètres de la méthode des plans parallèles sont introduits, il est maintenant possible de passer à l'étape de génération du trajet d'outil en cliquant sur le bouton « générer le trajet d'outil ».

Le principe est exprimé par l'algorithme suivant :

Algorithme : génération de trajet d'outil.

1. soit distance la distance spécifiée par l'utilisateur et α l'angle d'inclinaison,
2. détermination de l'équation du plan vertical d'équation $Ax+By+D=0$ passant par le point centre de la partie supérieure de l'enveloppe de la surface ou de l'enveloppe de l'ensemble des surfaces et faisant un angle α avec le plan XZ,

3. Calcul du nombre de plans,
4. soit $B_{initial} = \text{distance} / \cos(\alpha)$,
5. pour chaque plan faire :
 - Calculer $D_y = l$ 'intersection du plan avec l'axe des Y
 - $B_{initial} = B_{initial} + D_y$,
 - $D = -B_{initial} * B$,
 - Trouver les deux points de la limite du plan par le calcul des différentes intersections avec les segments de l'enveloppe,
 - Trouver les régions qui se chevauchent avec le plan,
 - Pour chaque région, calculer l'intersection de chacun de ses segments avec le plan en passant par les étapes suivantes :
 - Remplacer les paramètres des deux sommets du segment dans l'équation du plan,
 - Si les résultats sont tous les deux positifs ou négatif, alors il n'existe pas d'intersection et donc passer vers le segment suivant,
 - Si les résultats sont tous les deux nuls, alors les deux sommets du segment sont des points d'intersection et sont insérés dans la liste des points d'intersection et passer au segment suivant,
 - Si au moins un résultat n'est pas nul, calculer le point d'intersection et l'insérer dans la liste des points d'intersection et passer au segment suivant,
 - Refaire les mêmes étapes avec la liste des segments partielle.

La fonction qui permet d'insérer les différents points d'intersection permet aussi le tri de ces point suivant \bar{x} si l'angle d'inclinaison est différent de 90° et suivant \bar{y} si non. Le positionnement d'un point est donné par l'algorithme suivant :

Algorithme : génération de trajet d'outil.

Soit le point S à insérer dans la liste des points d'intersection,

1. si S est le premier élément dans la liste, alors insérer le et il devient le début de la liste,
2. si S est le deuxième élément, comparer son X (respectivement y) avec le début de la liste, s'il est inférieur il devient le début et le début devient la fin de la liste sinon c'est le contraire,
3. sinon
 - soit deux nœuds sommets P et Q tel que P pointe vers le début et Q vers le pivot qui est le milieu de la liste,
 - Tant que le X (respectivement Y) est supérieur au point P, P=suivant,

- Tant que X (respectivement Y) du point S est inférieur au point Q, Q=précédent,
- Refaire le calcul jusqu'à trouver la position où X (respectivement Y) du point S est inférieur au point P, alors le nœud P devient le suivant de S et dans ce cas il faut décrémenter le pivot, ou une position X (respectivement Y) supérieure au nœud Q, alors le point S devient le suivant de Q et donc il faut incrémenter le pivot.

IV.6. Trajet d'outil :

La dernière fenêtre permet la génération du trajet d'outil, la simulation des mouvements de l'outil et la génération du « G-Code » (voir figure 24).

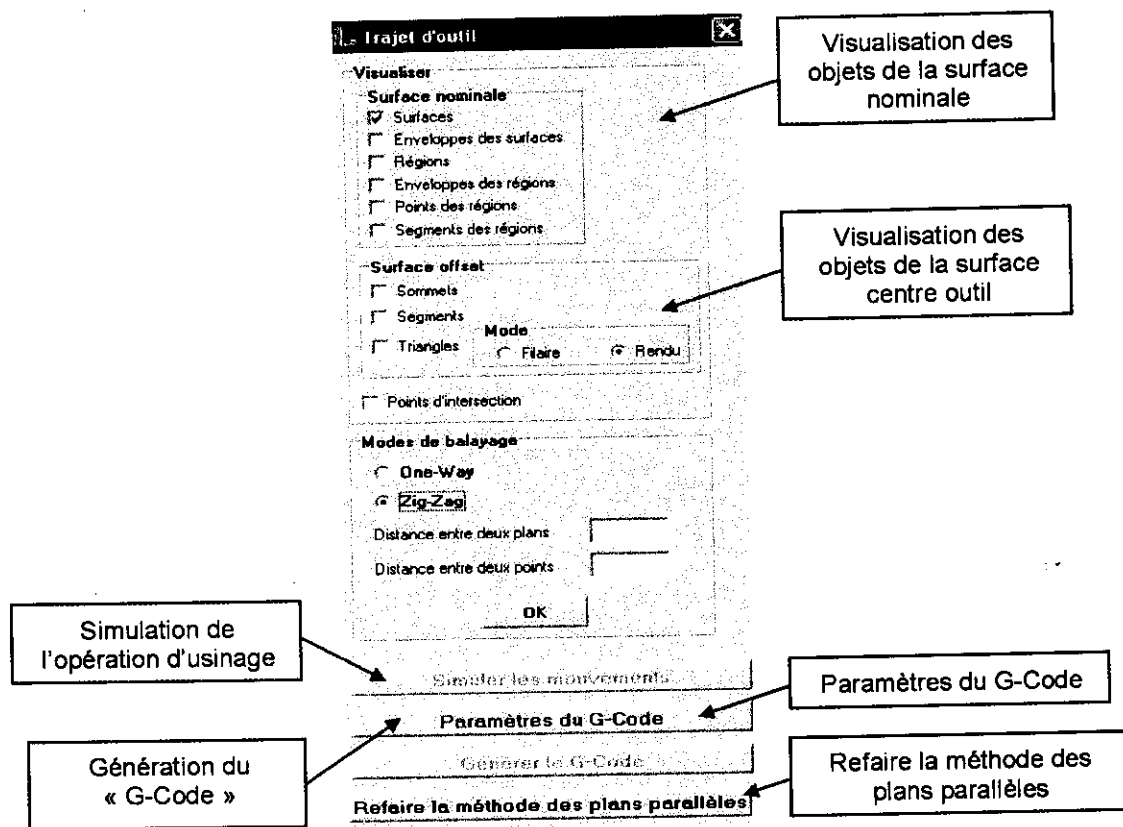


Figure 24 : Fenêtre trajet d'outil.

L'utilisateur doit d'abord spécifier le mode de balayage One-Way ou Zig-Zag ainsi que les paramètres nécessaires comme la distance de saut entre deux points ou entre deux plans successifs et spécification des paramètres du « G-Code ».

Le mode de balayage One-Way est exprimé par l'algorithme suivant :

Algorithme : mode de balayage One-Way

Soit $Z_{\text{outil}} = Z_{\text{point-rayon}}$

1. pour chaque plan faire,
 - insérer un point de même x et y que le point P ; le premier élément de la liste des points d'intersection et d'un $Z = Z$ de l'enveloppe+distance de sécurité et d'une vitesse égale à une constante (11111),
 - insérer un autre point de même position et vitesse avec un Z égale à $Z_{\text{outil}} + \text{l'engagement}$,
 - insérer le troisième point de même position X et Y que le point P et d'un Z égale à Z_{Outil} avec une vitesse égale à la vitesse d'engagement,
 - passer au point suivant ($P=\text{suivant}$)
2. pour chaque point faire
 - si la distance entre ces deux points est inférieure à la distance spécifiée par l'utilisateur faire :
 - insérer un point de même x et y que le point P et de Z égale à $Z_{\text{outil}} + \text{dégagement}$ et d'une vitesse égale à la vitesse de dégagement,
 - insérer un autre point avec un $Z = Z$ de l'enveloppe+distance de sécurité et d'une vitesse égale à une constante (11111),
 - passer au point suivant ($p=\text{suivant}$)
 - insérer un point de même x et y que le point P ; le premier élément de la liste des points d'intersection et d'un Z égale à Z de l'enveloppe+distance de sécurité et d'une vitesse égale à une constante (11111),
 - insérer un autre point de même position et vitesse avec un Z égale à $Z_{\text{outil}} + \text{l'engagement}$,
 - insérer un autre point de même position X et Y que le point P et d'un Z égale à Z_{Outil} avec une vitesse égale à la vitesse d'engagement,
 - sinon, insérer un point de même position X et Y que le point P avec un Z égale à Z_{outil} et une vitesse égale à la vitesse d'avance,
 - passer au point suivant ($P= \text{suivant}$)
 - si P est le dernier point faire :
 - insérer un point de même x et y que le point P et de Z égale à $Z_{\text{outil}} + \text{dégagement}$ et d'une vitesse égale à la vitesse de dégagement,
 - insérer un autre point avec un Z égale à Z de l'enveloppe+distance de sécurité et d'une vitesse égale à une constante (11111).

La figure 25 exprime le mode de balayage One-Way

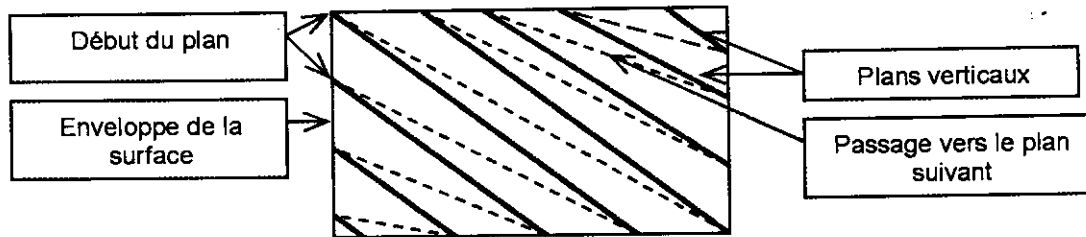


Figure 25 : Mode de balayage One-Way.

Le mode de balayage Zig-Zag est fait de la même manière qu'avec One-Way, sauf qu'à la fin de chaque plan i , nous passons à la fin du plan suivant j et nous parcourant les points vers le précédent (au lieu le suivant comme dans le plan i) jusqu'à atteindre le début de la liste des points du plan j et ensuite nous passons au début du plan suivant i . Nous répétons les étapes i, j jusqu'à terminer tous les plans (voir figure 26).

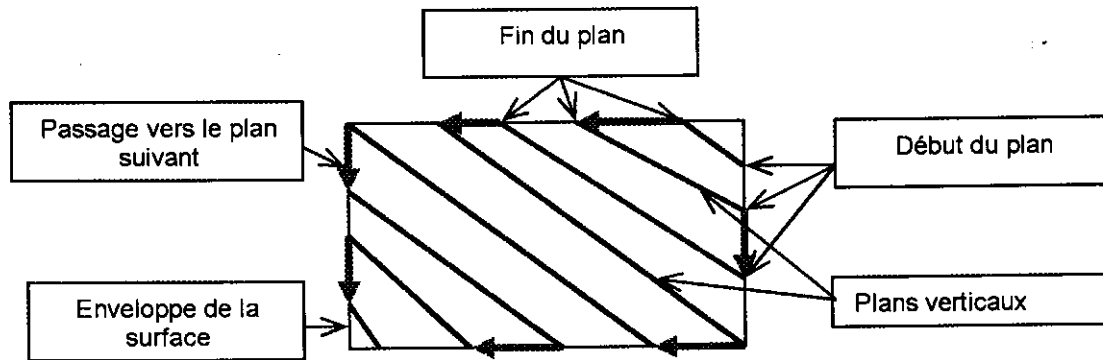
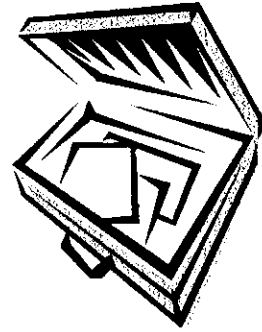


Figure 26 : Mode de balayage Zig-Zag.

V. CONCLUSION :

Dans ce chapitre, nous avons présentés toutes les formes que nous avons créées et intégrées à une application de CFAO permettant de trianguler la surface, de choisir l'outil théorique optimum et final, de générer la surface centre outil, de générer le trajet d'outil par la stratégie des plans parallèles et enfin de simuler les mouvements d'outils et de générer le « G-Code ». Dans le chapitre suivant, nous allons considérer des surfaces afin de tester et de valider notre application.



©

CHAPITRE 7
TESTS ET VALIDATIONS

I. INTRODUCTION :

Après l'étape d'implémentation et d'intégration des différents modules que nous avons développés, nous sommes arrivés maintenant à la dernière phase dans la démarche en cascade qui est la phase test et validation des résultats des différentes étapes précédentes. Dans cette phase, nous pouvons vérifier la comptabilité de notre travail ainsi la correction des problèmes pouvant apparaître dans l'exécution.

II. TESTS ET VALIDATIONS :

Cette étape nous permet de simuler et de tester les cas d'utilisation de l'usinage en finition par la méthode ITO à l'aide du déroulement des différents scénarios représentés par les diagrammes de séquence.

Pour les tests, nous allons considérer deux exemples d'application. Considérons la surface gauche « surface ondulée » (voir figure 1) et la surface « cavité_électrode » (voir figure 2).

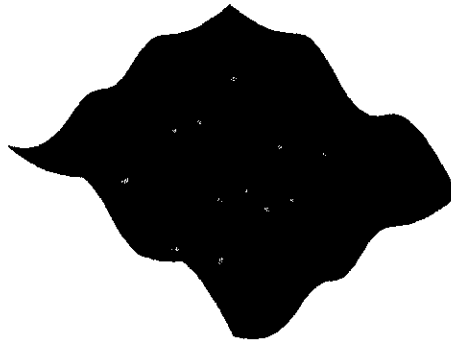


Figure 1 : Surface ondulée.

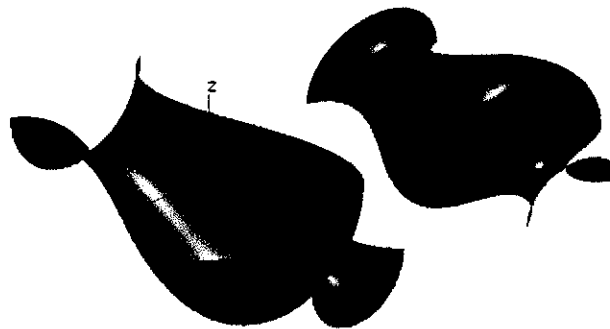


Figure 2 : Surface cavité_électrode.

Après l'ouverture du modèle CAO, le premier test concerne l'approximation de la surface par un ensemble de triangles (triangulation) (voir figure 3).

II.1. Surface ondulée :

II.1.1. Approximation du modèle CAO :

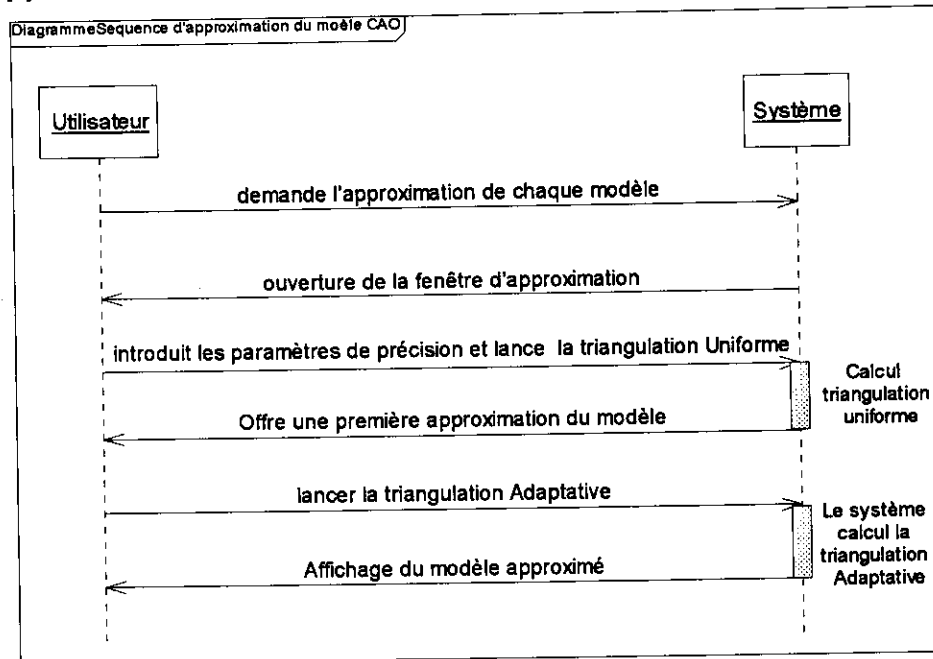


Figure 3 : Scénario d'approximation du modèle CAO.

Après l'ouverture de la fenêtre d'approximation, nous allons trianguler notre surface en deux étapes, triangulation uniforme (voir figure 4) et triangulation adaptative (voir figure 5). Pour cette approximation, nous avons considérés deux cas de triangulation pour chaque surface où les paramètres de précision sont les suivants :

1. Cas 1 :

- Longueur maximale du segment est égale à 25mm.
- Flèche maximale du segment est égale à 25mm.
- Flèche maximale du triangle est égale à 25mm.

2. Cas 2 :

- Longueur maximale du segment est égale à 3mm.
- Flèche maximale du segment est égale à 3mm.
- Flèche maximale du triangle est égale à 3mm.

A la fin de la triangulation, nous obtenons comme résultats un ensemble de points de la surface (sommets), un ensemble de segments et un ensemble de triangles.

Pour vérifier que les voisins de chaque triangle sont corrects, il est possible de parcourir la liste des triangles et visualiser les triangles voisins à un triangle donné (voir figure 6).

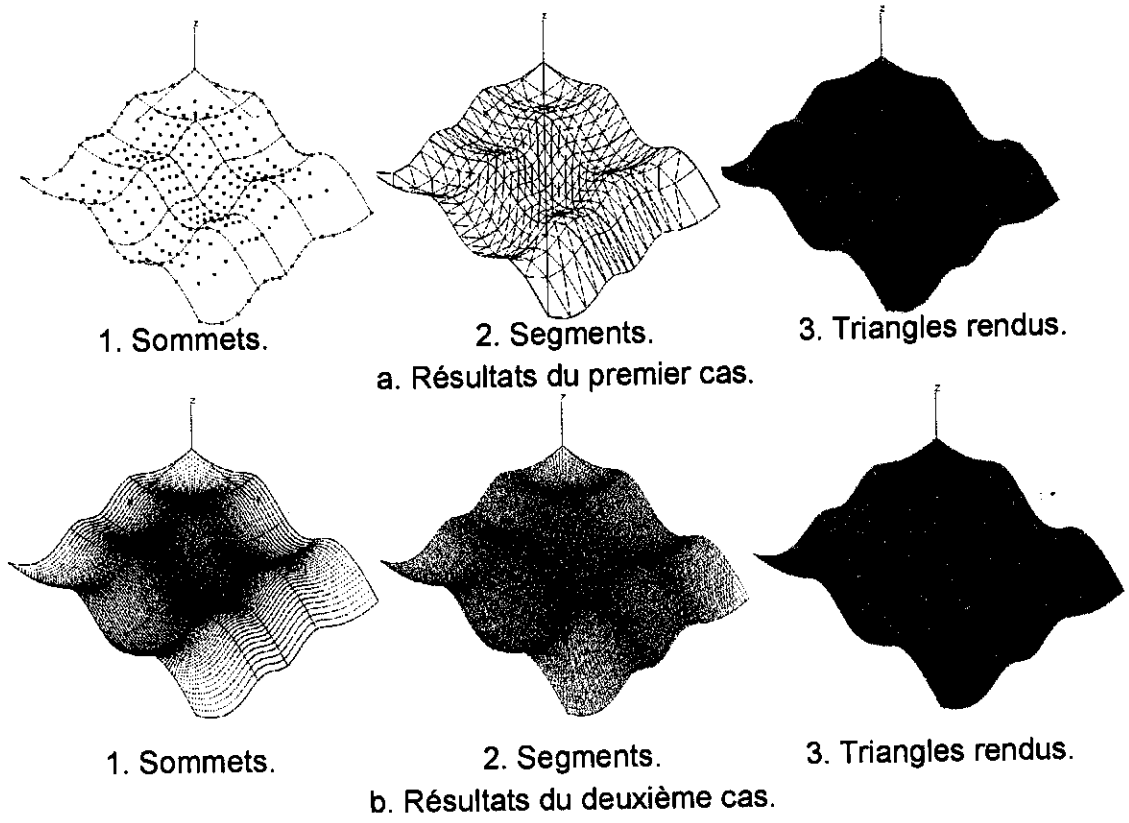


Figure 4 : Résultats de la triangulation uniforme.

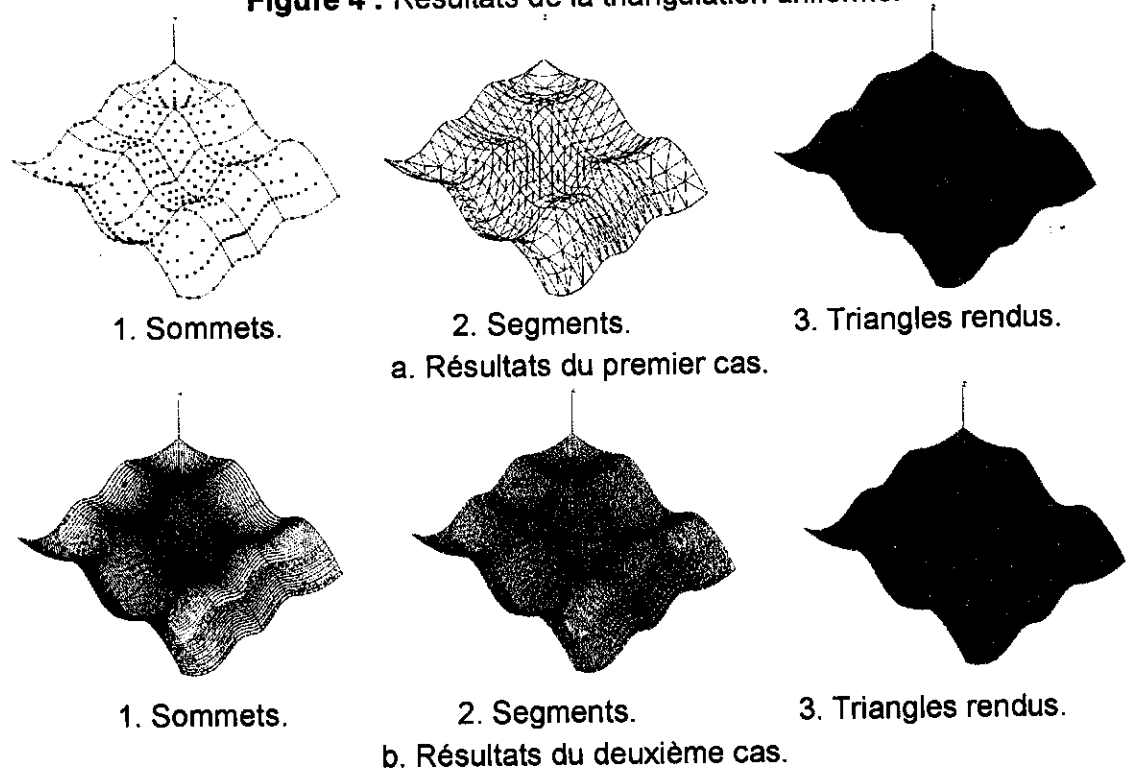


Figure 5 : Résultats de la triangulation adaptative.

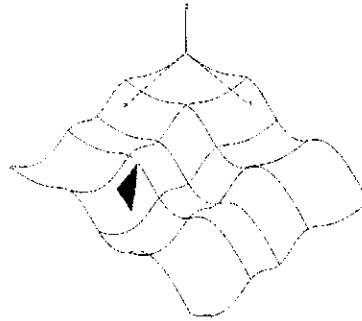


Figure 6: Triangles voisins.

A partir du bouton « afficher résultats », nous pouvons connaître les résultats de la triangulation pour les deux cas (voir figure 7).

	Uniforme	Adaptative
nombre de sommets	436	271
nombre de segments	800	1042
nombre de triangles	512	672
longueur maximale du segment	41.4226	12.935
longueur minimale du segment	5.9542	2.904
riche maximale du segment	4.0121	0.711
riche minimale du segment	0.0465	0.010
riche maximale du triangle	3.6489	3.649
riche minimale du triangle	0.0817	0.042

a. Cas 1.

	Uniforme	Adaptative
nombre de sommets	1941	2152
nombre de segments	3949	4364
nombre de triangles	2739	4326
longueur maximale du segment	5.5318	2.679
longueur minimale du segment	0.7048	0.253
riche maximale du segment	0.0741	0.514
riche minimale du segment	0.0009	0.000
riche maximale du triangle	0.0561	0.066
riche minimale du triangle	0.0006	0.001

b. Cas 2.

Figure 7 : Résultats des différents cas.

II.1.2. Choix d'outil théorique :

Après avoir approximer la surface et visualiser ses points, nous allons passer maintenant aux choix d'outil théorique (voir figure 8).

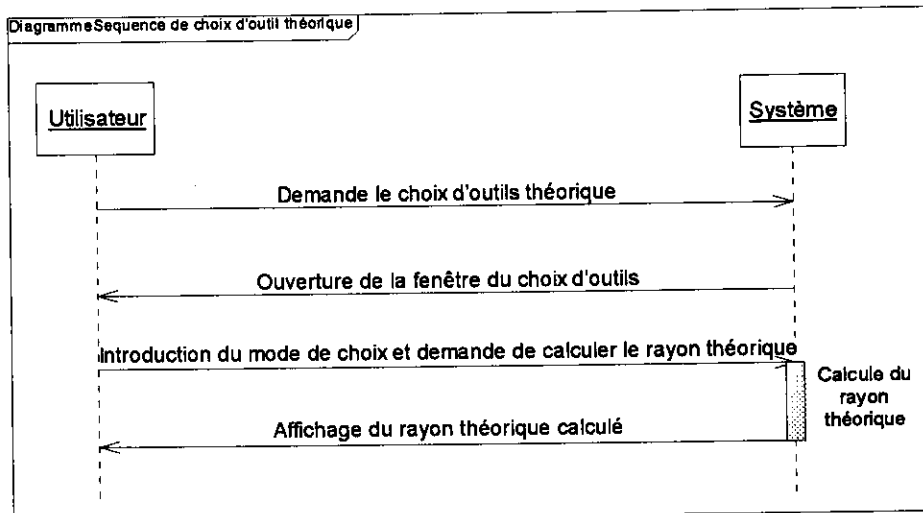


Figure 8 : Scénario du choix d'outil théorique.

En cliquant sur le bouton « choix d'outil », une nouvelle fenêtre de choix d'outil théorique apparaît où l'utilisateur est invité à sélectionner le mode de choix. En

cliquant sur le bouton « calculer les rayons théoriques », les différents rayons sont affichés dans une nouvelle fenêtre (voir figure 9).

N°Surface	Rayon théorique	Rayon utilisé	Type de surface
0	5.0640	5.0640	La Surface est composée de plusieurs formes locales

Figure 9 : Affichage du rayon théorique.

Nous pouvons aussi visualiser les formes locales en chaque point de la surface en fonction des courbures principales (voir figure 10).

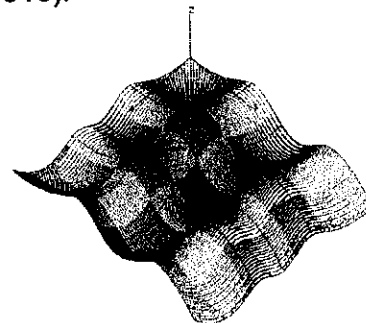
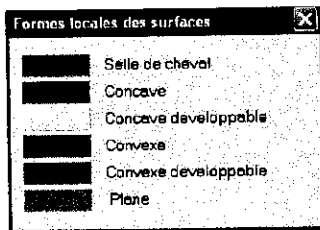


Figure 10 : Différentes formes locales de la surface.

Ce rayon théorique calculé peut créer des interférences et pour cela une deuxième opération de calcul du rayon d'outil (rayon optimum) sera lancée afin d'éliminer les interférences (voir Figure 11).

II.1.3. Choix d'outil optimum :

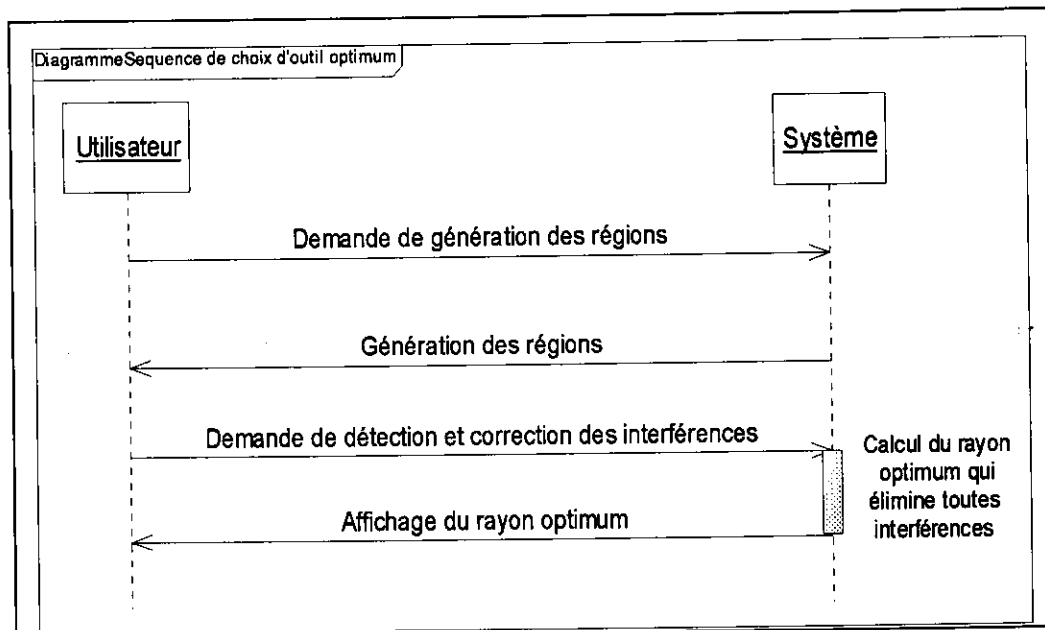
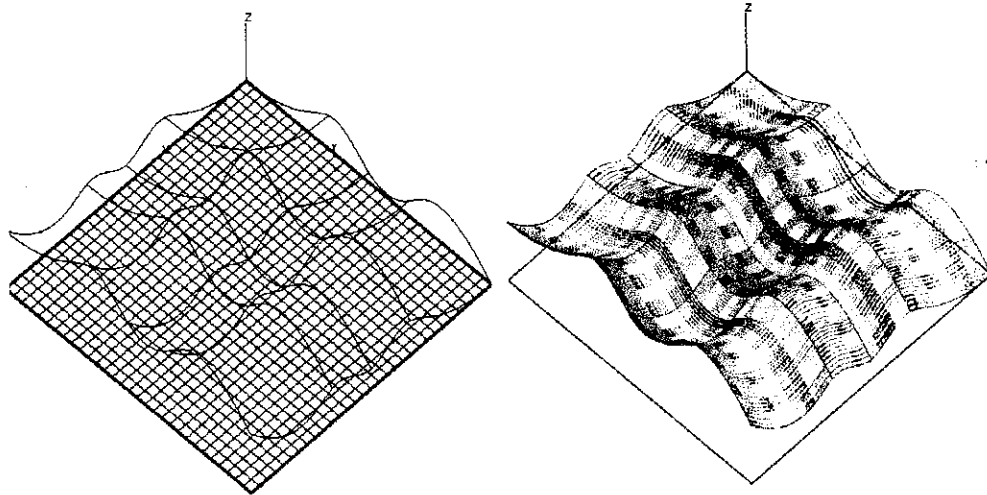


Figure 11 : Scénario de choix d'outil optimum.

Le bouton « générer régions » permet de regrouper les sommets en des régions pouvant être visualisées avec ses points (voir figure 12).



a. Ensemble des régions.

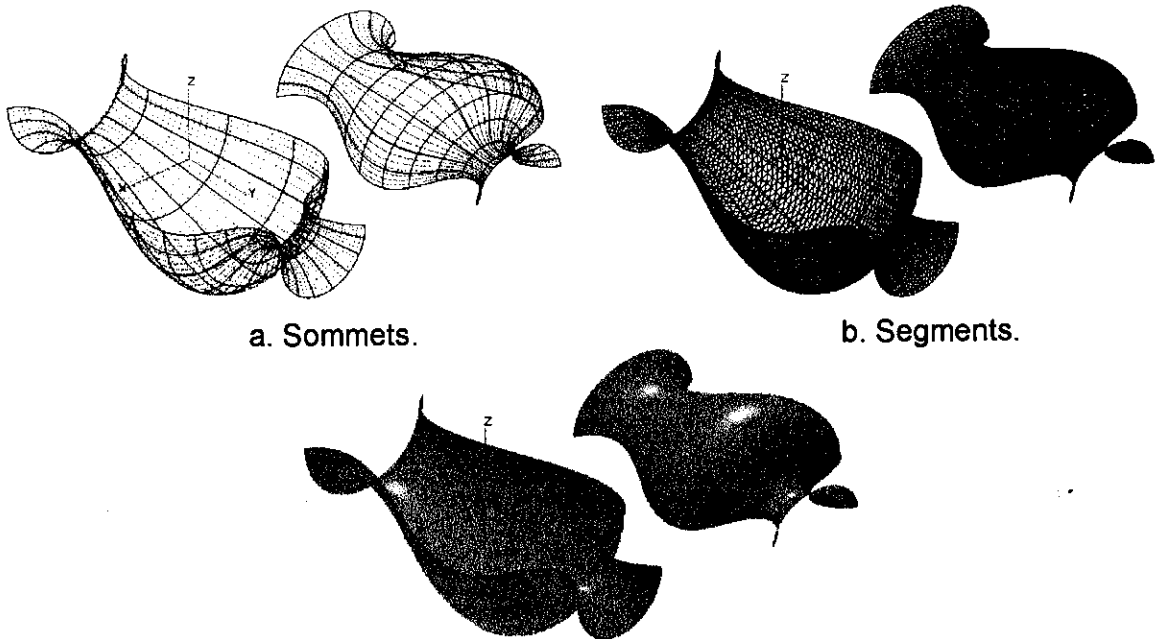
b. Points des régions.

Figure 12 : Régions et points des régions.

II.2. Surface cavité_électrode :

II.2.1. Approximation du modèle CAO :

En considérant le deuxième cas, la figure 13 montre la triangulation uniforme et la figure 14 montres la triangulation adaptative.

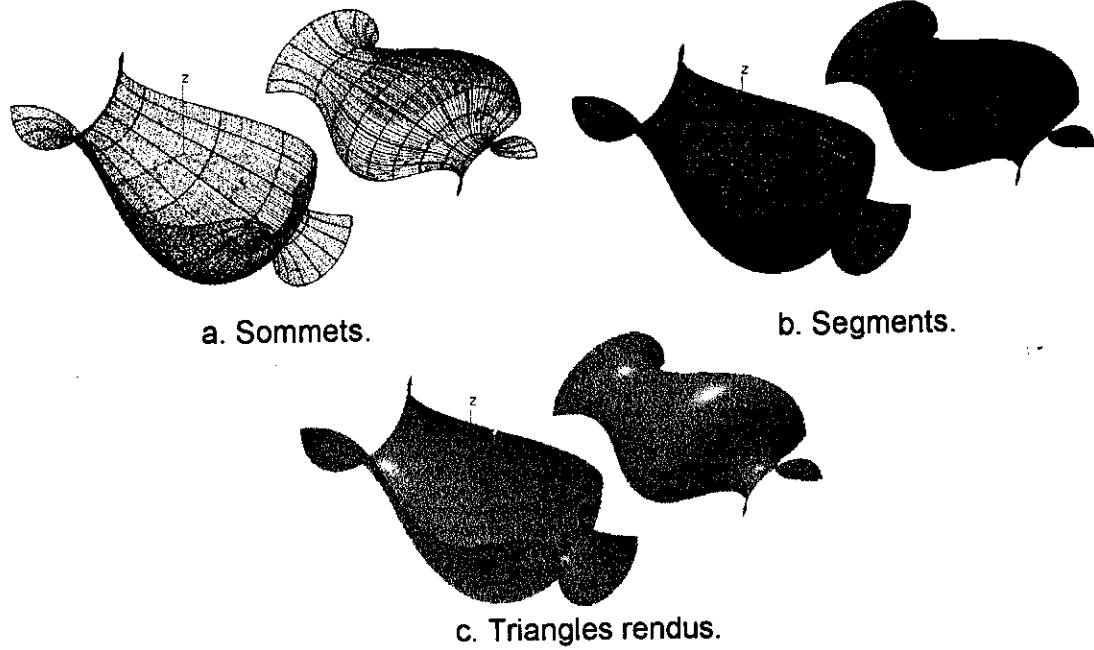


a. Sommets.

b. Segments.

c. Triangles rendus.

Figure 13 : Triangulation uniforme.



a. Sommets.

b. Segments.

c. Triangles rendus.

Figure 14 : Triangulation adaptative.

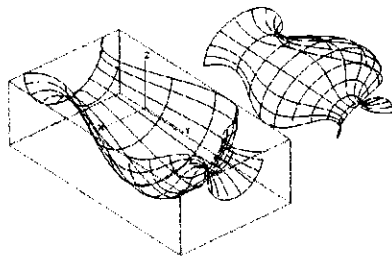
La figure 15 donne les résultats de la triangulation pour chaque surface.

	Uniforme	Adaptative
nombre de sommets	2544	14407
nombre de segments	2425	35247
nombre de triangles	4726	42691
longueur maximale du segment	7,236	1,758
longueur minimale du segment	0,9121	0,125
riche maximale du segment	0,1267	0,011
riche minimale du segment	0,0028	0,002
riche maximale du triangle	0,1231	0,123
riche minimale du triangle	0,0025	0,001

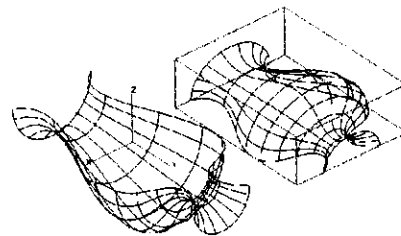
	Uniforme	Adaptative	Rapport
nombre de lignes	52	7	7,4
nombre de colonnes	7	7	1
nombre de sommets	4267	18120	4,250
nombre de segments	12418	33853	4,354
nombre de triangles	6112	35220	4,355
nombre de segments dont la longueur est valide	3906	33953	8,694
nombre de segments dont la riche est valide	12418	33853	4,324
nombre de triangles dont la riche est valide	7	3922	516,000
longueur maximale du segment	7,236	1,758	0,221
longueur minimale du segment	1,226	0,226	0,267
riche maximale du segment	0,126	0,011	0,079
riche minimale du segment	0,002	0,002	0,002
riche maximale du triangle	0,123	0,123	1,000
riche minimale du triangle	0,002	0,001	0,004

	Uniforme	Adaptative
nombre de sommets	4574	31207
nombre de segments	2425	35247
nombre de triangles	4726	42691
longueur maximale du segment	7,236	1,758
longueur minimale du segment	0,9121	0,125
riche maximale du segment	0,1267	0,011
riche minimale du segment	0,0028	0,002
riche maximale du triangle	0,1231	0,123
riche minimale du triangle	0,0025	0,001

	Uniforme	Adaptative	Rapport
nombre de lignes	52	7	7,4
nombre de colonnes	7	7	1
nombre de sommets	4267	18120	4,250
nombre de segments	12418	33853	4,354
nombre de triangles	6112	35220	4,355
nombre de segments dont la longueur est valide	3906	33953	8,694
nombre de segments dont la riche est valide	12418	33853	4,324
nombre de triangles dont la riche est valide	7	3922	516,000
longueur maximale du segment	7,236	1,758	0,221
longueur minimale du segment	1,226	0,226	0,267
riche maximale du segment	0,126	0,011	0,079
riche minimale du segment	0,002	0,002	0,002
riche maximale du triangle	0,123	0,123	1,000
riche minimale du triangle	0,002	0,001	0,004



a. Première surface.



b. Deuxième surface.

Figure 15 : Résultats des différentes surfaces.

II.2.2. Choix d'outil théorique :

Pour le choix du rayon d'outil théorique, nous pouvons choisir un outil pour chaque surface (voir figure 16) ou le même outil pour toutes les surfaces (voir figure 17).

Affichage des différents rayons (théorique, utilisé, corrigé, choisi)

N° Surface	Rayon théorique	Rayon utilisé	Type de surface
0	14.2273	14.2273	La Surface est composée de plusieurs formes locales
1	9.6992	9.6992	La Surface est composée de plusieurs formes locales

Figure 16 : Rayon théorique propre pour chaque surface.

Affichage des différents rayons (théorique, utilisé, corrigé, choisi)

N° Surface	Rayon théorique	Rayon utilisé	Type de surface
0	14.2273	9.6992	La Surface est composée de plusieurs formes locales
1	9.6992	9.6992	La Surface est composée de plusieurs formes locales

Figure 17 : Rayon théorique identique pour toutes les surfaces.

II.2.3. Choix d'outil optimum :

Les régions et les formes locales de chaque surface sont représentées respectivement par la figure 18 et la figure 19.

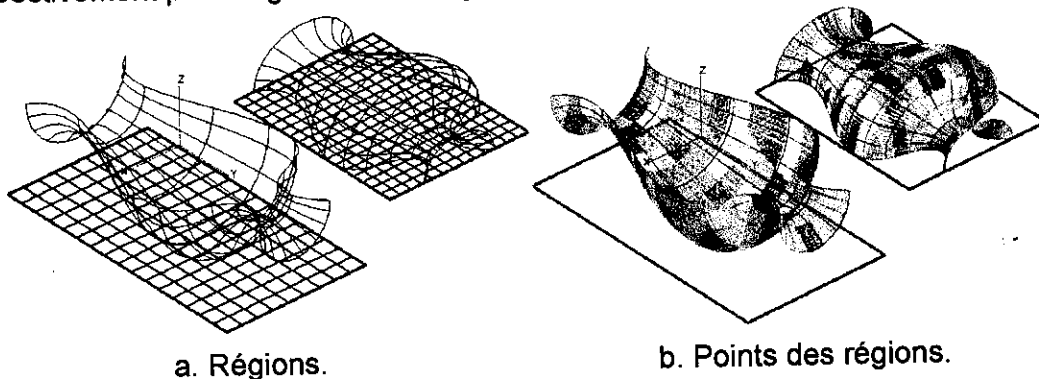


Figure 18 : Régions et points des régions.

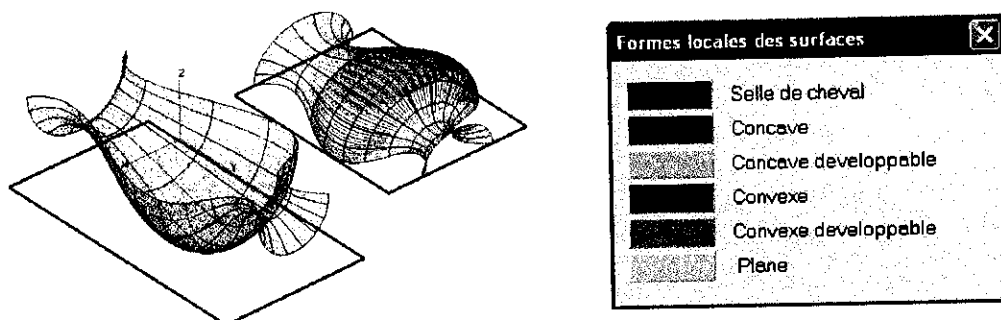


Figure 19 : Différentes formes locales de la surface.

Le bouton « détection et correction des interférences » permet de déterminer les sommets qui créent une interférence avec les autres sommets de la surface ou les surfaces voisines (voir figure 20). Le rayon optimum qui ne crée pas d'interférence sera affiché en cliquant sur le bouton « affichage du rayon corrigé » (voir figure 21).

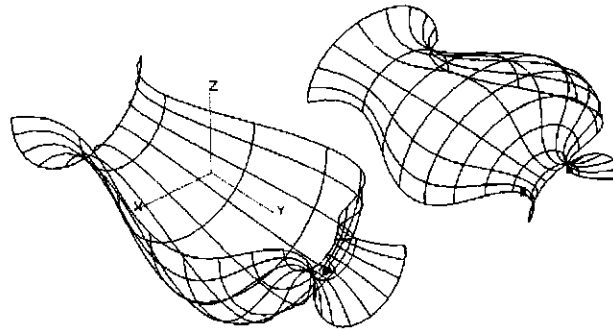


Figure 20 : Points d'interférence.

Affichage des différents rayons (théorique, utilisé, corrigé, choisi)

N° Surface	Rayon théorique	Rayon utilisé	Rayon corrigé	Type de surface
0	14.2273	14.2273	14.2155	La Surface est composée de plusieurs formes lo
1	9.8992	9.8992	9.8536	La Surface est composée de plusieurs formes lo

Figure 21 : Résultat du rayon corrigé.

II.2.4. Choix d'outil définitif :

En cliquant sur le bouton « choix d'outil définitif », une nouvelle fenêtre apparaît et l'utilisateur est invité à sélectionner le mode de choix. La figure 22 explique ce scénario.

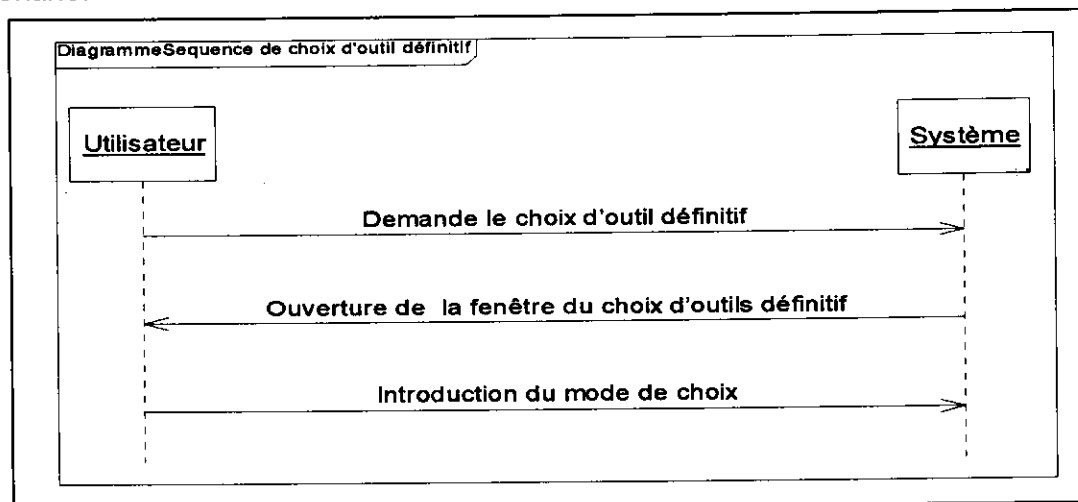
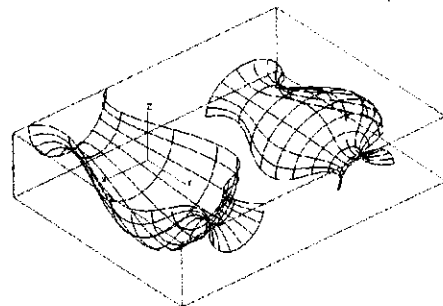


Figure 22 : Scénario de choix d'outil définitif.

Dans ce test, nous avons pris le cas où chaque surface a son propre outil. L'utilisateur peut faire un choix automatique à partir d'une base de données (voir figure 23) avec spécification du type d'attachement ou bien faire un choix manuel (voir figure 24).

Affichage des différents rayons (théorique, utilisé, corrigé, choi

N° Surface	Rayon optimal	Rayon Choisi	Longueur d'outil
0	14.2155	12.5000	250.0000
1	9.8536	8.0000	200.0000



a. Même type d'attachement.



Affichage des différents rayons (théorique, utilisé, corrigé, choisi)

N° Surface	Rayon optimal	Rayon Choisi	Longueur d'outil
0	14.2155	12.5000	117.0000
1	9.8536	8.0000	200.0000

b. Type d'attachements différents

Figure 23 : Choix automatique de l'outil.



Affichage des différents rayons (théorique, utilisé, corrigé, choisi)

N° Surface	Rayon optimal	Rayon Choisi	Longueur d'outil
0	14.2155	14.0000	200.0000
1	9.8536	9.0000	250.0000

Figure 24 : Choix manuel de l'outil.

II.2.5. Paramètre d'usinage :

Pour chaque outil choisi, il faut introduire certains paramètres (voir figure25).

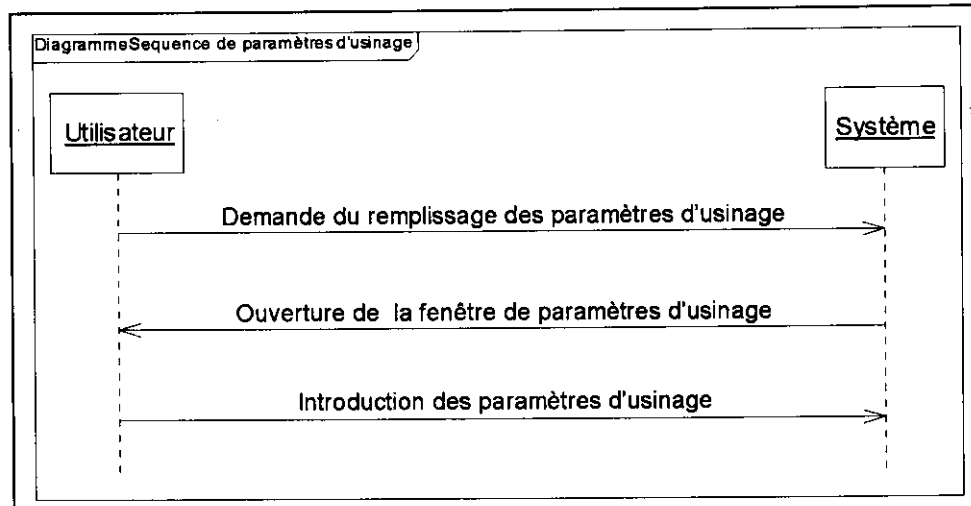
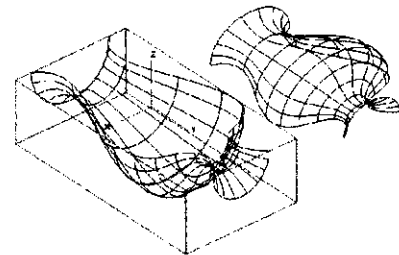


Figure 25 : Diagramme de séquence de paramètres d'usinage.

Nous allons maintenant cliquer sur le bouton « paramètre d'usinage » pour faire apparaître la fenêtre des paramètres d'usinage. Les paramètres d'usinage utilisés pour chaque surface sont les suivants :

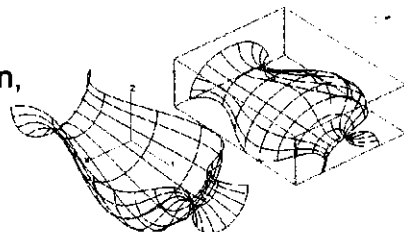
Première surface :

- Vitesse d'avance égale à 200mm/min,
- Vitesse d'engagement égale à 30mm/min,
- Vitesse de dégagement égale à 30mm/min,
- Engagement égal à 20mm,
- Dégagement égal à 10mm,
- Distance de sécurité égale à 30mm,
- Sens de rotation : trigonométrique.



Deuxième surface :

- Vitesse d'avance égale à 400mm/min,
- Vitesse d'engagement égale à 30mm/min,
- Vitesse de dégagement égale à 30mm/min,
- Engagement égal à 25mm,
- Dégagement égal à 15mm,
- Distance de sécurité égale à 30mm,
- Sens de rotation : trigonométrique.



II.2.6. Méthode ITO :

La surface centre outil « surface offset » est générée en cliquant sur le bouton « générer la surface offset » (voir figure 26).

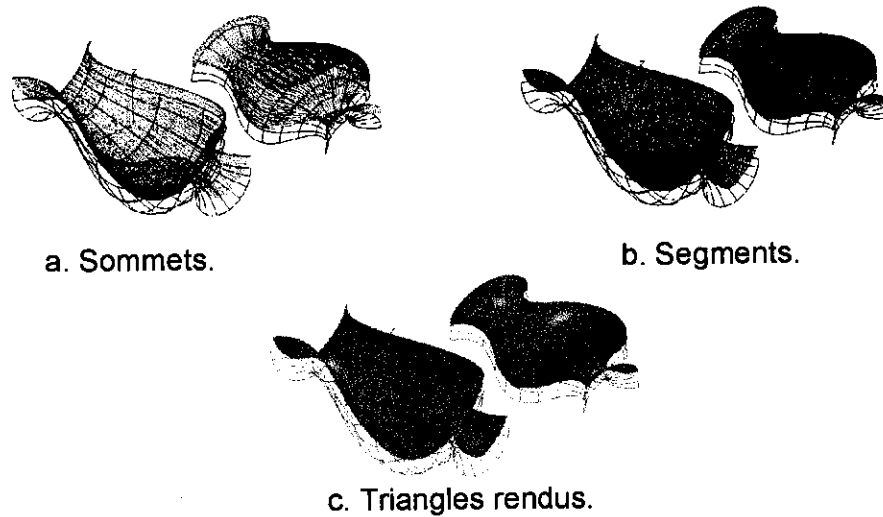


Figure 26 : Surface offset par la méthode ITO.

II.2.7. Usinage par la méthode des plans parallèles :

Lorsque l'utilisateur lance l'usinage par la méthode des plans parallèles, une fenêtre de paramètres de la méthode apparaît pour générer les différentes positions d'outil comme il est montré dans le scénario de la figure 27.

Diagramme de séquence pour l'usinage par la méthode des plans parallèles

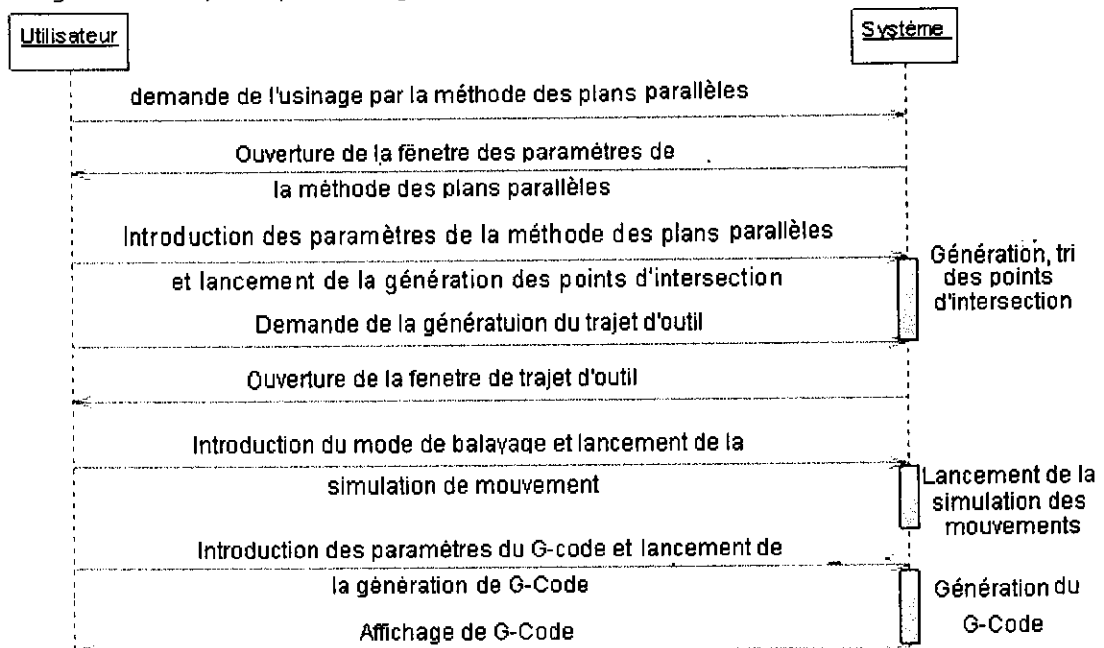


Figure 27. Diagramme de séquence de la méthode des plans parallèles.

Nous allons considérer quatre cas dans nos tests (voir figure 28) :

Cas 1 :

Utiliser un même angle pour les deux surfaces avec une distance différente entre deux plans successifs :

- Angle de rotation de 0° ,
- Distance entre deux plans successifs est égale à 5 mm pour la première surface et 10 mm pour la deuxième surface.

Cas 2 :

Chaque surface à son propre angle avec distance identique entre deux plans successifs :

- Angle de rotation de 90° pour la première surface et de 45° pour la deuxième surface,
- Distance entre deux plans successifs égale à 5mm.

Cas 3 :

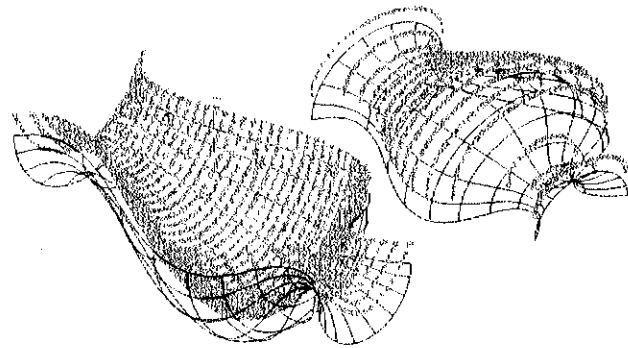
Chaque surface à son propre angle avec distance différente entre deux plans successifs :

- Angle de rotation de 45° pour la première surface et de 90° pour la deuxième surface,
- Distance entre deux plans successifs égale à 5mm pour la première surface et 10 pour la deuxième surface.

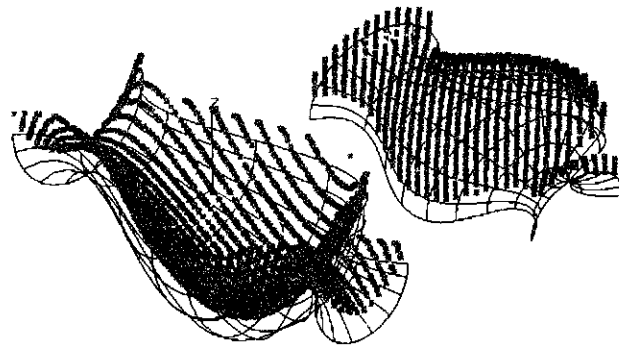
Cas 4 :

Utiliser un même angle pour les deux surfaces avec une même distance entre deux plans successifs :

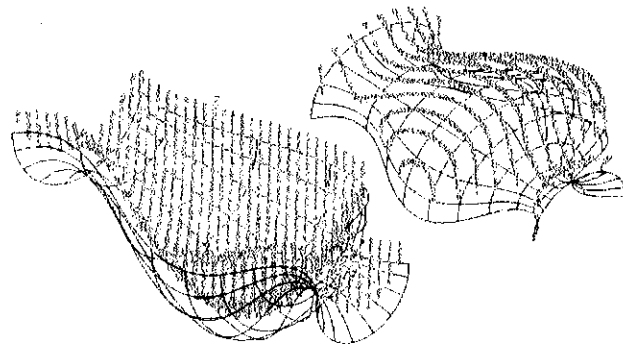
- Angle de rotation de 120° ,
- Distance entre deux plans successifs égale à 5mm



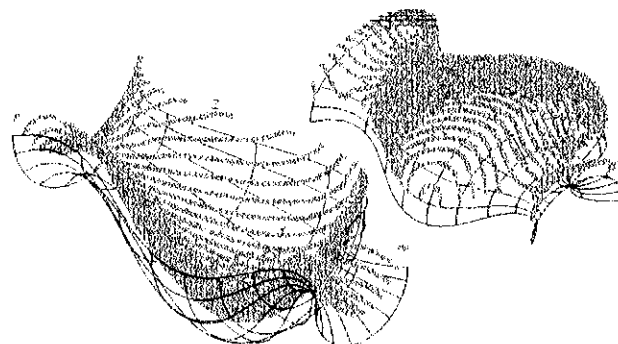
a. 1^{er} cas.



b. 2^{ème} cas



c. 3^{ème} cas.



d. 4^{ème} cas.

Figure 28 : Différentes positions des centre d'outils.

Pour générer le trajet d'outil, nous avons pris le cas où chaque surface a son propre angle avec une même distance entre deux plans successifs. Pour choisir le mode de balayage, il faut cliquer sur le bouton « générer le trajet d'outil » (voir figure 29).

- Cas 1 : nous avons pris le balayage ZIG-ZAG avec un pas de dégagement entre deux plans successifs égal à 30mm et un pas de dégagement entre deux points successifs égal à 20 mm,
- Cas 2 : Le balayage ONE-WAY avec un pas de dégagement entre deux points successifs égal à 30mm.

Après l'introduction du mode de balayage, nous passons à la phase de simulation du mouvement de l'outil. Le lancement de la simulation fait apparaître la fenêtre de simulation où l'utilisateur peut choisir le mode automatique ou le mode manuel pour la simulation virtuel du mouvement des outils par rapport aux surfaces à usiner.

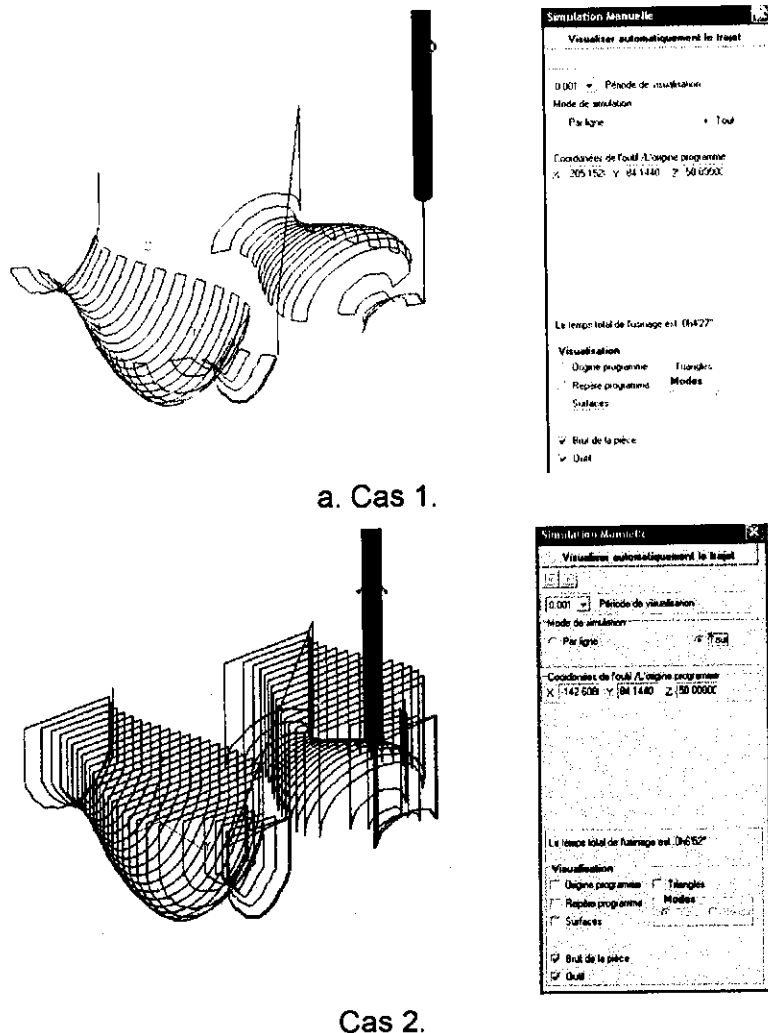
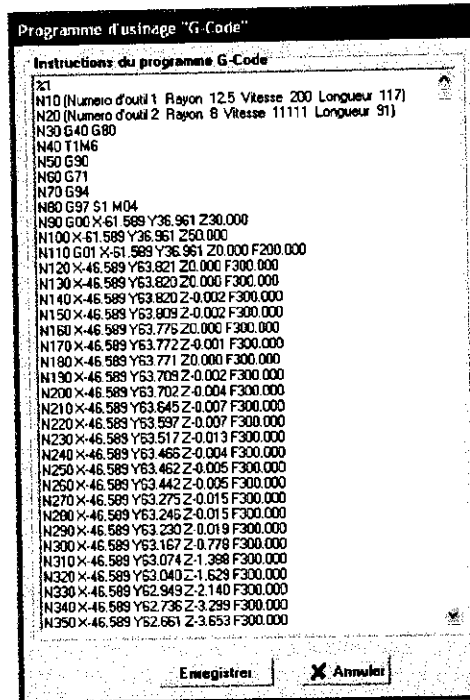


Figure 29 : Simulation du mouvement.

Après la génération du trajet d'outil, vient l'étape de génération du programme d'usinage « G-Code » (voir figure 30). Ce programme sera transmis à la machine pour faire l'usinage réel de la surface.

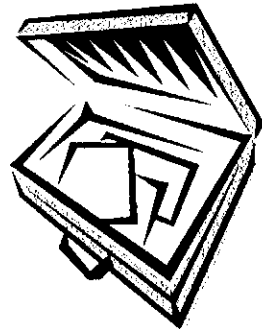


```
Programme d'usinage "G-Code"
Instructions du programme G-Code
%1
N10 (Numero d'outil 1 Rayon 125 Vitesse 200 Longueur 117)
N20 (Numero d'outil 2 Rayon 8 Vitesse 11111 Longueur 91)
N30 G40 G80
N40 T1M6
N50 G90
N60 G71
N70 G94
N80 G97 S1 M04
N90 G00 X-61.589 Y36.961 Z30.000
N100 X-61.589 Y36.961 Z50.000
N110 G01 X-61.589 Y36.961 Z0.000 F200.000
N120 X-46.589 Y63.821 Z0.000 F300.000
N130 X-46.589 Y63.820 Z0.000 F300.000
N140 X-46.589 Y63.820 Z-0.002 F300.000
N150 X-46.589 Y63.809 Z-0.002 F300.000
N160 X-46.589 Y63.775 Z0.000 F300.000
N170 X-46.589 Y63.772 Z-0.001 F300.000
N180 X-46.589 Y63.771 Z0.000 F300.000
N190 X-46.589 Y63.709 Z-0.002 F300.000
N200 X-46.589 Y63.702 Z-0.004 F300.000
N210 X-46.589 Y63.645 Z-0.007 F300.000
N220 X-46.589 Y63.597 Z-0.007 F300.000
N230 X-46.589 Y63.517 Z-0.013 F300.000
N240 X-46.589 Y63.466 Z-0.004 F300.000
N250 X-46.589 Y63.462 Z-0.005 F300.000
N260 X-46.589 Y63.442 Z-0.005 F300.000
N270 X-46.589 Y63.275 Z-0.015 F300.000
N280 X-46.589 Y63.245 Z-0.015 F300.000
N290 X-46.589 Y63.230 Z-0.019 F300.000
N300 X-46.589 Y63.167 Z-0.078 F300.000
N310 X-46.589 Y63.074 Z-1.398 F300.000
N320 X-46.589 Y63.040 Z-1.623 F300.000
N330 X-46.589 Y62.943 Z-2.140 F300.000
N340 X-46.589 Y62.736 Z-3.299 F300.000
N350 X-46.589 Y62.661 Z-3.653 F300.000
Enregistrer Annuler
```

Figure 30 : Programme « G-Code ».

III. CONCLUSION :

Nous avons présentés dans ce chapitre les différents scénarios des fenêtres de notre application logicielle en prenant un exemple de deux surfaces pour le testé et valider toutes les fonctions et les étapes de la triangulation jusqu'à la génération du G-Code.



**CONCLUSION
GENERALE**

CONCLUSION GENERALE

Le travail que nous avons présenté dans ce mémoire consiste en la conception et le développement d'une application logicielle pour l'automatisation de l'usinage en finition des surfaces gauches par la méthode ITO sur des fraiseuses à commande numérique à 03 axes en considérant des outils hémisphériques et la stratégie d'usinage par plans parallèles.

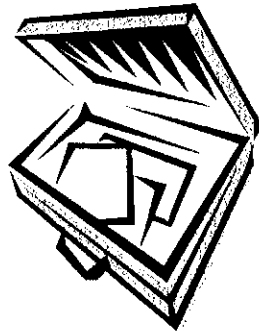
Lors de la réalisation de ce projet, nous avons commencés par une étude bibliographique sur les surfaces paramétriques et leurs propriétés et en particulier les surfaces B-Spline et NURBS. Ensuite, nous avons présentés les machines outils à commandes numérique utilisées dans l'usinage de ces surfaces et la syntaxe des programmes d'usinage « G-Code », les méthodes de générations de trajet d'outil et la méthode ITO et enfin les méthodes d'approximation des modèles CAO et le calcul des intersections entre les différents objets géométriques. A la fin de notre mémoire, nous avons présentés la conception et l'implémentation de notre application logicielle ainsi que les tests et les validations.

Le résultat de notre application est l'enrichissement de l'application logicielle graphique par l'intégration des modules permettant de réaliser les tâches suivantes :

- Triangulation adaptative des surfaces gauches et détermination du nombre minimum de triangles,
- Détection et correction des interférences,
- Calcul des outils optimums permettant d'éviter les problèmes d'interférences,
- Génération de la surface centre outil (surface offset) en utilisant la méthode ITO (Inverse Tool Offset) et un outil hémisphérique,
- Détermination et tri des points d'intersection entre un ensemble de plans verticaux et les surfaces à usiner,
- Génération de la trajectoire d'outil en utilisant la stratégie d'usinage par plans parallèles suivant les deux modes de balayage One-Way et Zig-Zag,
- Génération du programme d'usinage « G-Code »,
- Simulation virtuelle des mouvements des outils.

En perspective, nous recommandons de traiter les points suivants :

- Génération de la surface centre outil pour les outils torique et cylindrique,
- Développement d'une application distribuée (sur un réseau de PC) afin d'accélérer le processus d'exécution et d'augmenter la productivité,
- Usinage en ébauche et en finition à partir d'un nuage de points,
- Usinage sur des fraiseuses 5 axes,
- Adaptation des vitesses d'avance en finition.



ANNEXES

ANNEXE A

PRINCIPALES FONCTIONS DU G-CODE

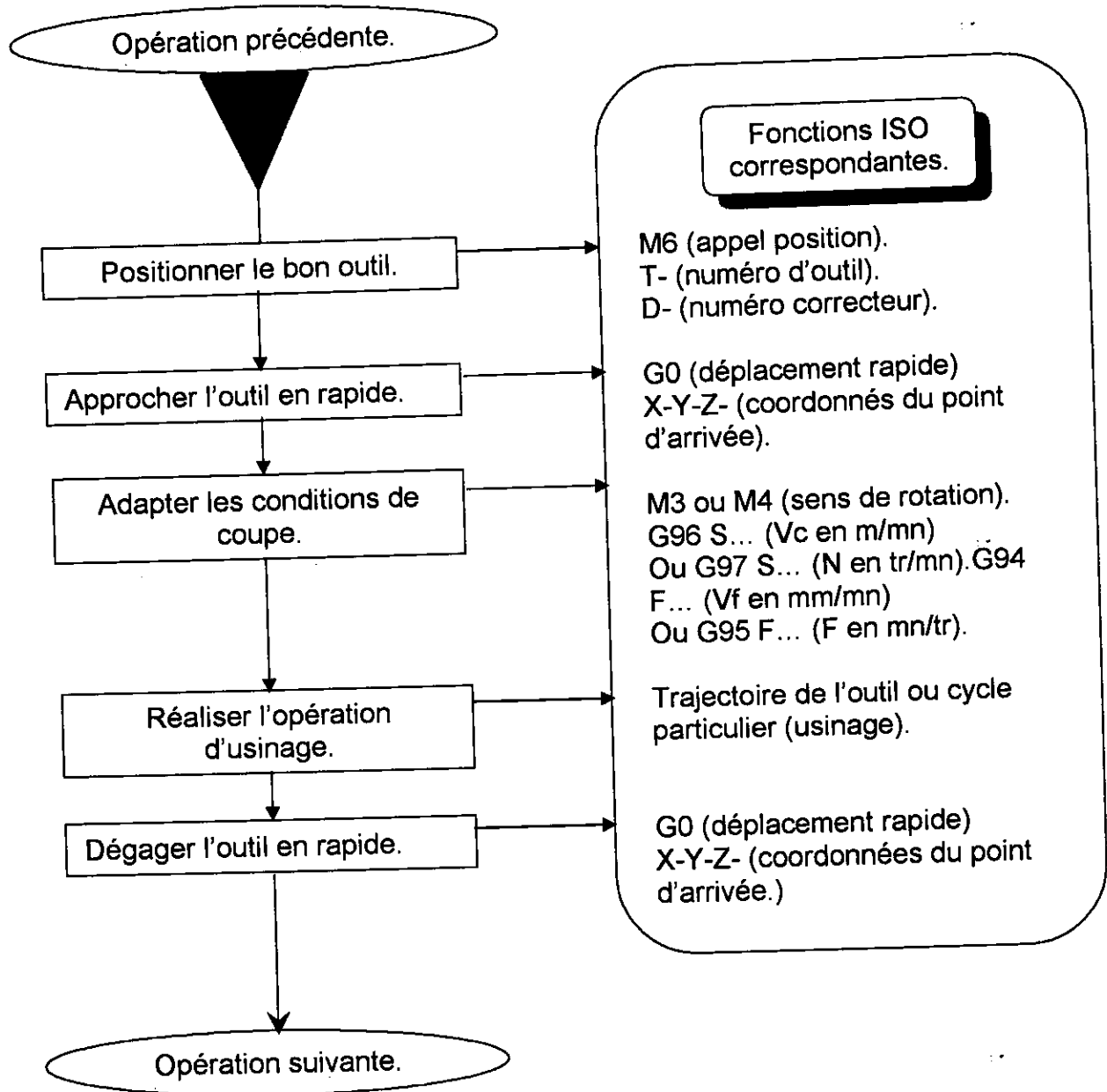
I. INTRODUCTION :

Un programme d'usinage est la transcription dans un langage compréhensible par le directeur de commande numérique d'une machine des opérations d'usinage à effectuer sur une pièce.

II. STRUCTURE D'UN PROGRAMME

Structure	Exemple
<pre> graph TD A([Numéro de programme identification.]) --> B[Initialisation] B --> C[Position dégagement.] C --> D[Opération 1.] D --> E[Position dégagement.] E -.-> F[Opération N.] F --> G[Position dégagement.] G --> H([Fin de programme]) </pre>	<pre> %4723 (Tour SMI) (Poignée) (Ph 10 coté cone) (11/04/03) N28 G40 G80 M5 M9 N29 G0 G52 X0 Z0 (OPERATION CENTRER) N30 T5 D5 M6 (FORET A CENTRER) N40 G0 X0 Z52 N45 G95 G97 S2500 F0.05 M3 M8 N50 G1 Z40 N60 G0 Z52 N70 G0 G52 X0 Z0 M9 (FINITION PROFIL) N240 T3 D3 M6 (OUTIL DE FINITION) N250 G92 S4000 N260 G0 X4 Z52 F0.05 M8 N270 G1 G42 X5 Z46 N280 G96 S250 N290 G1 X15.961 Z46 N300 X24 Z31 N310 Z26 N320 X26 N330 G3 X34 Z22 I26 K22 N340 G1 Z18 N350 X42 N360 G97 S1000 N370 G0 G52 G40 X0 Z0 M5 M9 N 710 M2 </pre>

III. STRUCTURE D'UNE OPERATION



IV. FONCTIONS ISO :

Indication	Adresses	Signification
début d'un programme	%	Début de programme (paramètre d'entrée).
Ordre.	N H	Numéro de bloc Numéro de sous-programme.
Fonctions préparatoires.	G	Elle prédisposent la machine pour un ordre précis (peuvent varier suivant le DCN).
Déplacements suivant les axes.	X Y Z A B C	Valeur du déplacement. Positif ou négatif sur le ou les axes choisis.
Avances.	F	Vitesse d'avances exprimée en mm/min.
Fonctions auxiliaires.	M	Fonction particulière désignant un ordre.
Outils.	T.. D	Sélection de l'outil et de son (ou ses) correcteur (s).
Rayon. plan de sécurité.	R ER	Rayon du cercle en G02 G03. Plan de référence en cycle.
Vitesse. Répétition.	S	N ou Vc suivant G96 ou G97 placé avant. Répétition d'une séquence.

V. LES FONCTIONS PREPARATOIRES G DE FRAISAGE

Code	Désignation	Révocation
G00.	Interpolation linéaire à la vitesse rapide.	G01-G02-G03
G01	Interpolation linéaire à la vitesse programmée.	G00-G02-G03
G02.	Interpolation circulaire sens anti-trigo. G2 X..Y..I..J.. ou R	G00-G01-G03
G03.	Interpolation circulaire sens trigo. G3 X..Y..I..J.. ou R..	G00-G01-G02
G40*	Annulation d'une correction d'outil suivant le rayon.	G41-42
G41.	Positionnement de l'outil à gauche de la trajectoire programmée d'une valeur égale au rayon.	G40-G42
G42.	Positionnement de l'outil à droite de la trajectoire programmée d'une valeur égale au rayon.	G40-41
G52.	Programmation absolue des coordonnées par rapport à l'origine mesure.	fin de bloc

G59.	Changement d'origine programme	Fin de bloc
G80*	Annulation d'un cycle d'usinage.	Tout cycle
G81.	Cycle de perçage - centrage. G81 X...Y...Z...ER...	G80
G82.	Cycle de perçage - chambrage G82 X.... Y ... Z... FR... EF...	G80
G83.	Cycle de perçage avec déburrage G83 X... Y...Z...ER... P... Q...	G80
G84.	Cycle de taraudage. G84 X...Y...Z...ER... (F S x pas)	G80
G85.	Cycle d'alésage. G85 X...Y... Z... ER...	G80
G86.	Cycle d'alésage avec arrêt de broche en fond de trou.G86 X... Y...Z... ER...	G80
G87.	Cycle de perçage avec brise_copeaux G87 Y...Z... ER... P... Q... ER...	G80
G88.	Cycle d'alésage et dressage de face. G 8 X...Y...Z...ER...	G80
G89.	Cycle d'alésage avec arrêt temporisé en fond de trou. G89 X... Y...Z... ER... EF...	G80
G90*.	Programmation absolue des coordonnées.	
G92.	Limitation de la fréquence de rotation (N) en tr/min	M02
G94.	Vitesse d'avance en mm/min	G95
G95.	Vitesse d'avance en mm/tr	G94
G97.	Fréquence de rotation (N) constante en tr/min	G96

VI. LES FONCTIONS AUXILIAIRES

code	désignation	Révolution Action sur DCY
M00	Arrêt programmé.	
M02	Fin de programme pièce.	%
M03	Rotation de broche sens horaire.	M0-M4-M5
M04	Rotation de broche sens anti- horaire.	M0-M3-M5
M05	Arrêt de broche.	M3-M4
M06	Changement d'outil.	Compte rendu
M07	Arrosage N°1.	M0-M9
M08	Arrosage N°2.	M0-M9
M09	Arrêt des arrosages.	M8
M41	Gamme de rotation 1.	
M42	Gamme de rotation 2.	

ANNEXE B

UML « UNIFIED MODELING LANGUAGE »

I. GENERALITE :

UML (Unified Modeling Language, « langage de modélisation objet unifié ») est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, OOD et OOSE.

II. NAISSANCE DU LANGAGE UML :

Les premières bases d'UML résultent de la fusion de trois méthodes en 1995 :

II.1. OMT (Object Modeling Technique) :

OMT a été développée par James Rumbaugh dans le centre de recherche et développement de la société « General Electric » à la fin des années 80.

II.2. OOD (Object Oriented Development):

Au départ, elle est conçue pour le développement d'applications en langage Ada à destination du « Department of Defense ». Ensuite, elle sera étendue au C++.

II.3. OOSE (Object Oriented Software Engineering) :

La méthode OOSE de Ivar Jacobson couvre tout le cycle de développement du logiciel. Créée dans un centre de recherche d'Ericsson (Suède).

II.4. Unification et normalisation d'UML :

Pendant la période allant de 1995 à 1997, les différentes méthodes composant UML ont été unifiées et normalisées (voir figure1). Un certain nombre d'autres méthodes ont également fourni des idées. La version finale d'UML, version 1.1, a été adoptée par l'OMG (Object Management Group) en novembre 1997.

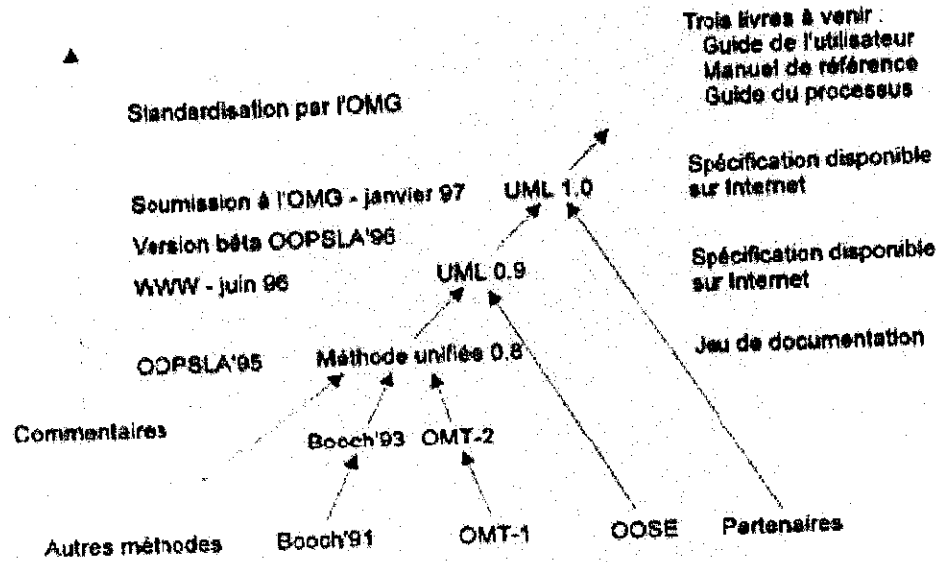


Figure 1: Principales étapes de la définition d'UML

III. DIFFERENTS DIAGRAMMES DE UML :

Le métamodèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relient. Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes.

On distingue deux types de vues (voir figure2):

➤ **Vues statiques :**

1. Diagrammes de cas d'utilisation,
2. Diagrammes d'objets,
3. Diagrammes de classes,
4. Diagrammes de composants,
5. Diagrammes de déploiement.

➤ **Vues dynamiques :**

1. Diagrammes de séquence,
2. Diagrammes de collaboration,
3. Diagrammes d'états transitions,
4. Diagrammes d'activités.

Les diagrammes des cas d'utilisation représentent également une vue fonctionnelle du système.

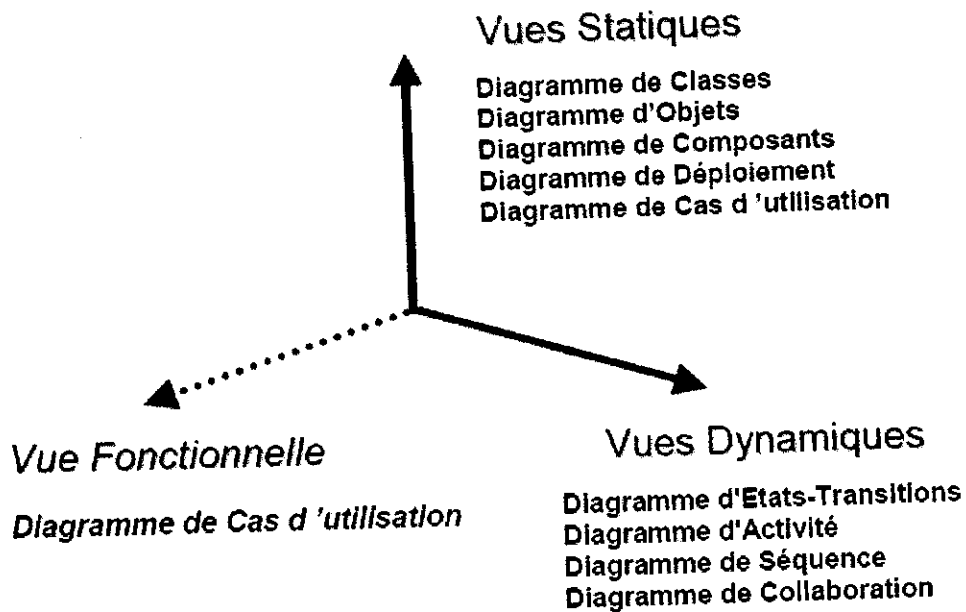


Figure 2 : Différentes vues du système.

III.1. Diagramme de cas d'Utilisation :

En UML, les diagrammes de cas d'utilisation (*Use Case*) constituent le moyen essentiel pour saisir les fonctionnalités d'un système du point de vue de l'utilisateur.

Le diagramme de cas d'utilisation décrit la fonctionnalité proposée pour le système à développer.

Le diagramme cas d'utilisation est caractérisé par :

- **Acteurs** : ce sont des personnes ou systèmes extérieurs à l'application et qui interagissent avec elle.
- **Cas d'utilisation** : représentent une fonctionnalité (un objectif à atteindre) du système à construire. Ils sont en relation avec des acteurs et d'autres cas d'utilisation.
- **Relations** : généralement orientées. Elles peuvent être de trois types :
 1. La relation d'utilisation : une relation d'utilisation permet de décomposer un cas d'utilisation en sous cas d'utilisation.
 2. La relation d'inclusion : est une dépendance stéréotypée dans laquelle le cas d'utilisation source incorpore de façon explicite le comportement du cas d'utilisation cible à un emplacement spécifié dans le cas d'utilisation source.

γ. La relation d'extension : cas d'utilisation cible enrichit le comportement de cas d'utilisation source au point d'extension spécifié.

La figure 3 montre un exemple de diagramme de cas d'utilisation.

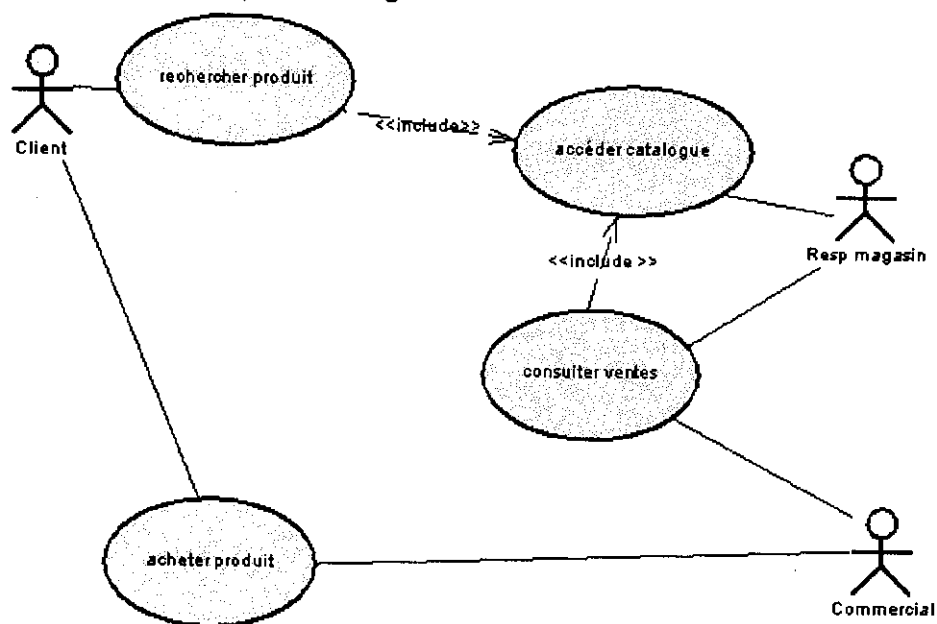


Figure 3 : Exemple de diagramme de cas d'utilisation.

III.2. Diagramme de classes :

Une classe est un type abstrait caractérisé par des propriétés (attributs et méthodes) communes à un ensemble d'objets et permettant de créer des objets ayant ces propriétés.

Classe = attributs + méthodes.

Un diagramme de classe exprime de manière générale la structure statique du système (voir figure4) ; les classes et les relations qui existent entre elles. De même qu'une classe décrit un ensemble d'objets, une association entre classes décrit un ensemble de liens entre objets. Un objet est donc une instance de classe, au même titre qu'un lien est une instance d'association. Ce diagramme ne décrit rien en particulier sur les objets, mais exprime les liens potentiels d'un objet vers les autres objets. Chaque extrémité d'une association permet de préciser le rôle joué par chaque classe dans l'association. Il est également possible de préciser pour une association :

- Un nom de rôle,
- Une cardinalité,
- L'agrégation,
- La spécialisation ou la généralisation.

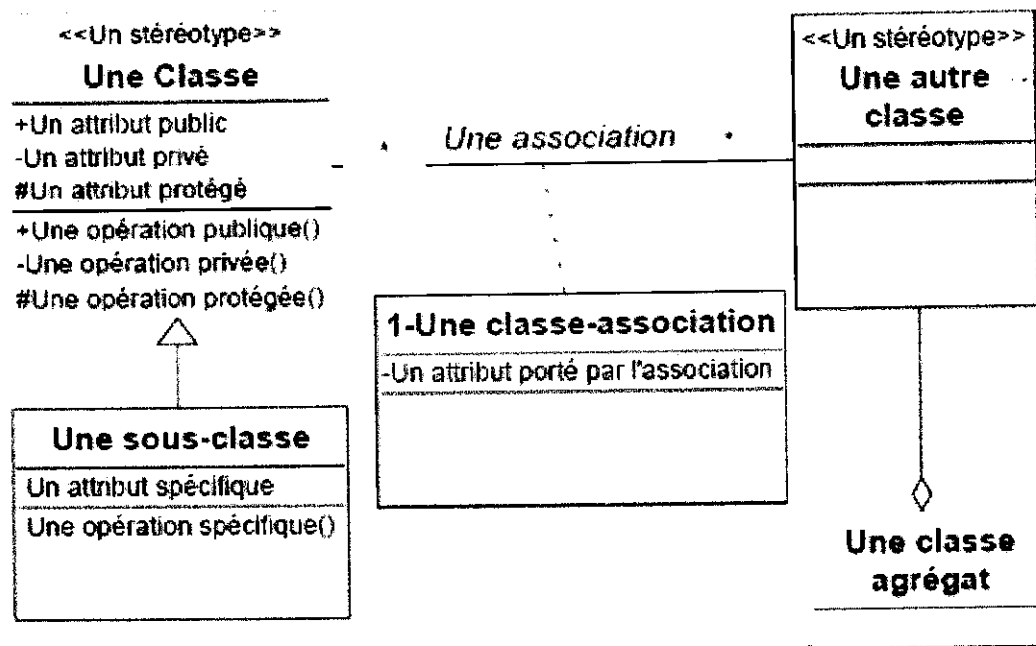


Figure 4 : Exemple de diagramme de classes.

III.2. Diagramme de collaboration :

Ces diagrammes montrent les interactions entre les objets (voir figure5). Ils insistent particulièrement sur la structure des liens permettant de mettre les objets en collaboration, en communication. Ils représentent à la fois les objets et les messages que ceux-ci s'envoient. Les diagrammes de collaboration sont une extension des diagrammes d'objets. Un lien sert de support de transmission pour le message. Le message déclenche une action dans l'objet destinataire. Il est également possible de préciser pour un message ses arguments et valeurs de retour.

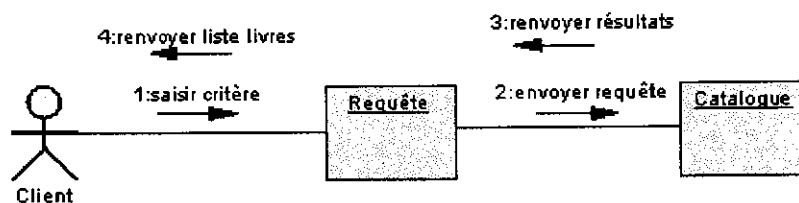


Figure 5: Exemple de diagramme de collaboration entre objets.

III.3. Diagramme de séquences :

Par rapport au diagramme de collaboration, le diagramme de séquence (voir figure6), met en évidence l'aspect chronologique de l'envoi des messages. Ce type de diagramme insiste sur l'aspect temporel, alors que le diagramme de collaboration insiste sur l'aspect spatial. C'est l'équivalent du scénario de la méthode OMT. Souvent, le diagramme de séquence permet de compléter le diagramme des cas d'utilisation en mettant en évidence les objets et leur interaction.

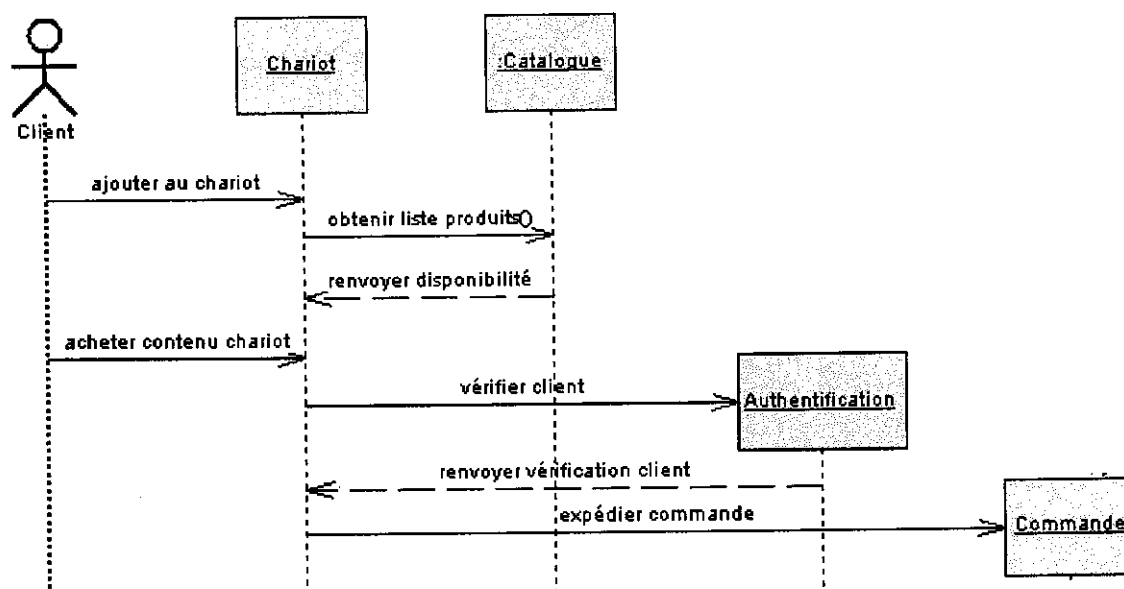


Figure 6: Exemple de diagramme de séquence.

III.4. Diagramme des activités :

Le diagramme d'activités (voir figure7) est un modèle détaillé qui représente les flux de contrôle (appelés transitions) entre les actions effectuées par le système (appelées activités) depuis un point de début jusqu'à plusieurs points de fin potentiels

La notion d'activité peut être considérée vis à vis des éléments suivants:

- Une opération
- Une étape dans une opération
- Une action d'un scénario d'un cas d'utilisation

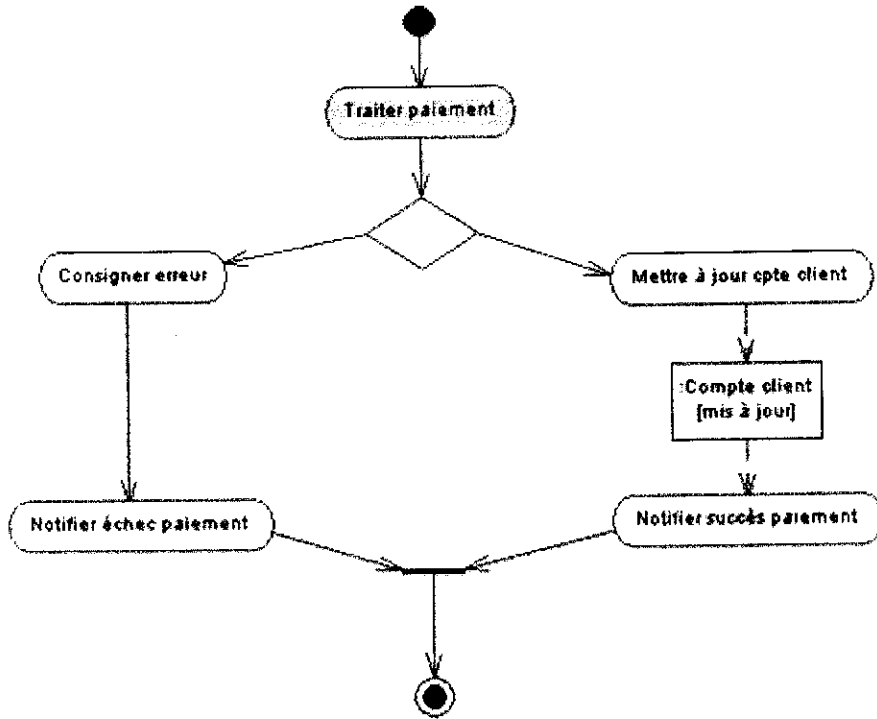


Figure 7: Exemple de diagramme d'activité.

REFERENCES BIBLIOGRAPHIQUES

- [1] S. Boudjouad, N. Tafat-Bouزيد, « Méthodes des plans parallèles pour l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes », PFE, Université USTHB, 2004.
- [2] C. K. Schene « Introduction to Computing with Geometry Notes ». Department of Computer Science ; Michigan Technological University, 2003.
- [3] J. C. Léon. « Modélisation et Conception de Surfaces pour la CFAO ». Edition Hermès, Paris, 1991.
- [4] M. Bey, « Modélisation des Courbes et des Surfaces », Rapport de Recherche, Laboratoire de Robotique et d'Intelligence Artificielle. Avril 2000.
- [5] J. C. Chambelland, « Approche Déclarative de la Modélisation de Surfaces », Laboratoire d'Informatique de Marseille.
- [6] M. P. Cani, « Modélisation et Animation », cours 3^{ème} année ENSIMAG IRV, 2006.
- [7] L. Piegl, « On NURBS : A Survey », University of South Florida.
- [8] D. Chabane, L. Mouterfi, « Automatisation de l'Opération d'Ebauchage des Surfaces sur des Fraiseuses à Commande Numérique à 3 Axes », PFE Université Saad Dahlab, Blida, 2005.
- [9] M. Ouaret, F. Bouabdellah, « Conception et réalisation d'un outil de modélisation des surfaces B-Splines et NURBS », PFE, Université USTHB, 2002.
- [10] Y. Martinet, « Introduction a la C.F.A.O », novembre 2001.
- [11] E. Duc, E. Lefur, « Machines _Outils à Commande Numérique Structure Modélisation et Réglage », septembre 1997.
- [12] <http://www.iutp.univ-poitiers.fr/COURS-DOCS/cours/MOCN-programmation>.
- [13] S. Rezzak, H. Taibi, « Méthode de Z constant pour l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes », PFE, Université Saad Dahlab, Blida, 2005.
- [14] « Commande Numérique par Calculateur », cours 1^{ère} année GMP 2003-2004.
- [15] « La MOCN ou machine outil à commande numérique », La lettre du centre de technologies nouvelles, 1996. (<http://www.ctn.asso.fr>).
- [16] J. Richard, V. K. Nguyen, « STEP-NC: Une Interface Intelligente pour l'Intégration CAO-FAO-MOCN ». Ecole d'Ingénieurs de Genève.
- [17] « Manuel de Programmation VOLUME 1. NUM : 1020/1040/1060M ».

- [18] C. Tournier, « Evaluation des Modes de Génération des Trajets Outil pour l'Usinage de Formes Gauches ». Mémoire de Recherche. Ecole Normale Supérieure de Cachan, Juillet 1996.
- [19] A. Cherfi, « Conception et Développement d'une Application d'Optimisation du Trajet d'Usinage par la méthode Z-constant », PFE, Université Saad Dahlab, Blida, 2006.
- [20] E. Duc, « Usinage de Formes Gauches Contribution à l'Amélioration de la Qualité des Trajectoires d'Usinage », Thèse de doctorat, Ecole Normale Supérieure de Cachan, 1998.
- [21] H. Nebbat, A. Chourar, « Simulation et vérification des programmes d'usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes », PFE, Université Saad Dahlab, Blida, 2004.
- [22] M. Bey, « Automatisation de la Sélection de la Stratégie et de la Direction d'Usinage pour la Finition des Surfaces Gauches », rapport de recherche, novembre 2004.
- [23] D. Misra, V. Sundararajan, P. K. Wright, « Zig-Zag Tool Path Generation for Sculptured Surface Finishing », Workshop in CAD/CAM, 2003.
- [24] <http://www.mathcurve.com/surfaces/rationnelle/rationnelle.shtml>.
- [25] Y. Ren, « Computational Techniques to Improve Efficiency and Accuracy for High Performance Machining of Polyhedral Models », Thèse de Doctorat, University North carolina, 2002.
- [26] B. K. Choi, B. H. Kim, « Die-cavity Pocketing via Cutting Simulation ».
- [27] B. Furet, « Génération des Trajectoires d'Outils - principes et paramètres » Ecole polytechnique Montreal.
- [28] M. Inui, « Fast Inverse Offset Computation Using Polygon Rendering Hardware », CAD 2003.
- [39] B. Ageunini, H. Messaoudi, « Simulation et vérification de l'opération d'ébauchage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes », PFE, Université Saad Dahlab, Blida, 2006.
- [30] B. Salvatore, « Réseau National de Ressources des Structures Métalliques ».
- [31] <http://fr.wikipedia.org/wiki/>.
- [32] P. Martinon. « Résolution Numérique de Problèmes de Contrôle Optimal par une Méthode Homotypique Simplicité ». Thèse de Doctorat, 2005.
- [33] <http://serge.mehl.free.fr/anx/Lemniture.html>.