

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

**Université Saad Dahlab Blida 1**

N° D'ordre : .....



Faculté des sciences

**Département d'informatique**

Mémoire Présenté par :

Lakrib Ali      Boutagga M'hamed

**En vue d'obtenir le diplôme de master**

**Domaine : Mathématique et informatique**

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

**Sujet :**

***Conception et réalisation d'une application pour  
finition des surfaces complexes sur des fraiseuses à 05-axes***

**Soutenu le : 24/06/2015**

H. Derrar  
M. Zair  
M. Hamouda  
F. Zahra  
M. Bey

Président  
Examineur  
Examineur  
Promotrice  
Encadrante

**Promotion**  
2014 / 2015

MA-004-232-1

# **REMERCIEMENT**

*Au terme de ce travail nous tenons à remercier tout d'abord ALLAH tout puissant de nous avoir donné le courage et facilité la réalisation de ce travail.*

*Nous exprimons notre grande reconnaissance et remerciement à nos encadreurs Mr. Bey Mohamed et Mr. Bendhifallah Hassène pour leurs aides, leurs conseils et de nous avoir orienté et mis à notre disposition tous les moyens nécessaires durant notre travail.*

*Nous exprimons également notre gratitude à notre promotrice Mme. Zahra Fatma Zohra.*

*Et enfin, que nos professeurs, nos chers parents et amis, et bien avant tout, trouvent ici l'expression de nos remerciements les plus sincères et les plus profonds en reconnaissance de leurs sacrifices, aides, soutiens et encouragements pour atteindre nos buts.*

*Merci*

*A mes très chers parents qui n'ont jamais cessé de  
m'aider et m'encourager durant tout mon parcours et qui  
m'ont permis d'être aujourd'hui ce que je suis,*

*A mes chers frères, Et à toute ma grande famille  
sans exception.*

*A mon amie, et mon binôme m'hamed avec qui j'ai appris  
tellement de choses et que je remercie d'avoir été patient  
et collaboratif,*

*A tous mes amies, qu'ils m'excusent de ne pas pouvoir les  
citer au risque d'oublier quelqu'un, je vous dis **MERCI**  
pour tous les moments inoubliables qu'on a passé  
ensemble durant ces cinq années,*

*A tous ceux qui me sont chers,*

*Lakrib Ali*

*A mes très chers parents qui n'ont jamais cessé de  
m'aider et m'encourager durant tout mon parcours et qui  
m'ont permis d'être aujourd'hui ce que je suis,*

*A mes chers frères, Et à toute ma grande famille  
sans exception.*

*A mon amie, et mon binôme ali avec qui j'ai appris  
tellement de choses et que je remercie d'avoir été patient  
et collaboratif,*

*A tous mes amies, qu'ils m'excusent de ne pas pouvoir les  
citer au risque d'oublier quelqu'un, je vous dis **MERCI**  
pour tous les moments inoubliables qu'on a passé  
ensemble durant ces cinq années,*

*A tous ceux qui me sont chers,*

*Boutagga mhamed*

## Résumé

Le présent projet s'inscrit dans le cadre du programme triennal 2014-2016 de recherche de l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO », de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA », intitulé « Production des Surfaces de Formes Libres sur des Fraiseuses Numériques à 05 axes ».

Dans ce projet, nous nous intéressons à la génération automatique du trajet d'outil pour la finition des surfaces complexes à partir de leurs modèles STL sur des fraiseuses à commande numérique à 05-axes en utilisant la stratégie d'usinage « Z-Constant » et un outil hémisphérique tout en évitant les problèmes d'interférences et de collisions. L'objectif étant la détermination des points de passage de l'outil et les orientations de l'outil associées à chaque point de passage permettant d'éliminer les interférences et les collisions. La finalité de ce projet est le développement d'une application logicielle graphique et interactive sous Windows permettant d'automatiser ce processus.

**Mots clés :** Modèle STL, Finition, Interférence, Surface Complexes, Collision, Outils, Fraiseuse 5 axes, Z-Constant.

## Abstract

This project is part of the three-year research program 2014-2016 of the team Computer Aided Design and Manufacturing « CAD/CAM » of Production Engineering and Robotics Division « DPR » of Advanced Technology Development Center « CDTA » entitled « Production of Free forms Surfaces on 05-axes CNC Milling Machines »

In this project, we are interested in the automatic generation of the tool path for the finishing of complex surfaces from their STL models on 05-axis CNC milling machines using the machining strategy « Z-Constant » and ball end milling tool with the avoidance of interferences and collisions problems. The objective is the determination of the tool positions and the associated orientations eliminating interferences and collisions. The finality is the development of interactive and graphic software under Windows permitting the automation of the process.

**Keywords:** STL Model, Interference, Collision, Tools, Z-Constant, Complex Surfaces, 05-axis CNC milling machines.

## Table des Matières

Introduction générale.....	1
<b>Chapitre 1</b>	
Introduction : .....	3
PARTIE 1 : MODELS ET PROPRIETES DES SURFACES .....	3
1. Catégories des modèles géométriques :.....	3
1.1. Modèles paramétriques.....	3
1.2. Modèles polyédriques. ....	5
PARTIE 2 : MACHINES D'USINAGE : .....	10
1. Définition d'une MOCN : .....	10
2. Classification des fraiseuses : .....	11
3. Programmation des fraiseuses : .....	12
PARTIE 3 : USINAGE DES SURFACES COMPLEXES.....	13
1. Décomposition du processus d'usinage.....	13
2. Paramètres de coupe .....	14
2.1. Principe de la coupe des métaux.....	14
2.2. Paramètres de coupe.....	15
3. Différentes types d'outils pour les opérations de fraisage.....	16
4. Positionnement de différentes formes d'outil.....	17
5. Types interférences.....	19
5.1. Interférences locales.....	19
5.2. Interférences vers l'arrière.....	19
5.3. Interférences globales(collision).....	19
6. Stratégies d'usinage en finition.....	19
6.1. Plans parallèles.....	19
6.2. Isoparamétriques.....	20
6.3. Z-constant.....	20

6.4. Comparaison des stratégies.....	21
7. Critères de précision.....	21
Conclusion.....	22

## Chapitre 2

Introduction.....	23
1. Outils de modélisation .....	23
2. Conception.....	23
• Cadre du sujet.....	23
• Problématique.....	23
• Objectifs.....	24
3. Modélisation de L'application.....	25
3.1. Diagrammes de cas d'utilisation.....	25
3.2. Diagramme de classes.....	29
3.3. Diagrammes d'activités.....	36
3.4. Diagrammes de séquence.....	37
3.5. Diagramme de collaboration.....	42
Conclusion.....	43

## Chapitre 3

Introduction.....	44
1. Architecture générale de l'application.....	44
2. Solutions proposés.....	45
2.1. Lecture de forme de pièce à partir d'un fichier STL.....	45
2.2. Création des contours.....	47
2.3. Gestion des interférences et collisions.....	53
2.4. Simulation cinématique.....	67
aConclusion.....	70

---

---

## Chapitre 4

1.Introduction.....	71
2.Teste et validation.....	71
2.1.Lecture de fichier STL.....	72
2.2.Creation des contours.....	74
2.3.Points de passage de l'outil.....	76
2.4.Detection et correction des interférences et collisions.....	78
2.5.Simulation des mouvements de l'outil.....	81
3.Conclusion.....	83
Conclusion générale.....	84
Références bibliographique.....	86



Figure 30 : Mode de balayage par Z-constant. ....	21
Figure 31: Erreur de corde. ....	22
Figure 32: Hauteur de crête en 5 axes. ....	22

## Chapitre 2

Figure 1 : Diagramme de cas d'utilisation global. ....	26
Figure 2 : Diagramme de cas d'utilisation « créer contours ». ....	27
Figure 3 : Diagramme de cas d'utilisation « gérer interférence et collision ».....	27
Figure 4 : Diagramme de classes. ....	29
Figure 5 : Classe Point_Stl. ....	30
Figure 6 : Classe Sommet_Stl. ....	30
Figure 7 : Classe Nœud_Sommet_Stl. ....	30
Figure 8 : Classe Nuage_Points. ....	30
Figure 9 : Classe Sommet_Intersection. ....	30
Figure 10 : Classe Triangles_Stl. ....	31
Figure 11 : Classe Nœud_Triangles_Stl.....	31
Figure 12: Classe Maillage_Triangles. ....	31
Figure 13 : Classe Segment_Stl. ....	31
Figure 14 : Classe Nœud_Segment_Stl. ....	31
Figure 15 : Classe Liste_Segment_Stl. ....	32
Figure 16 : Classe Bande. ....	32
Figure 17 : Classe Liste_Bandes.....	32
Figure 18 : Classe Plan_Stl. ....	33
Figure 19 : Classe Centre_Outil. ....	33
Figure 20 : Classe Outil_Stl. ....	34
Figure 21 : Classe Contours_Stl. ....	34
Figure 22 : Classe Nœud_Contours_Stl. ....	34
Figure 23 : ClasseListe_Contours_Stl. ....	35

Figure 24 : Classe Trajectoire. ....	35
Figure 25 : Diagramme d'activité global. ....	36
Figure 26 :Diagramme d'activité pour « créer contours ». ....	36
Figure 27 : Diagramme d'activité pour « gérer interférence et collision ». ....	37
Figure 28 : Lire fichier STL. ....	38
Figure 29 : Diagramme de séquence pour « créer contours ».....	40
Figure 30 : Diagramme de séquence pour « gérer interférence et collision ».....	41
Figure 31 : Diagramme de collaboration. ....	42

### Chapitre 3

Figure 1 : Organigramme global. ....	44
Figure 2 : Organigramme de lecture du fichier STL. ....	47
Figure 3 : Organigramme de création des contours. ....	48
Figure 4: Plans horizontaux. ....	49
Figure 5 : Bandes des triangles. ....	49
Figure 6: Pas d'intersection entre le plan et le triangle. ....	50
Figure 7: Le plan passe par un seul sommet de triangle. ....	50
Figure 8 :Plan en intersection avec une seule arrête de triangle. ....	50
Figure 9 : Plan en intersection avec le plan en deux sommets.....	51
Figure 10 : Le plan passe par un seul sommet et une arrête.....	51
Figure 11 :Le triangle appartient au plan.....	51
Figure 12 : Organigramme de chaînage des segments.....	54
Figure 13 : Gestion des interférences et des collisions.....	55
Figure 14 : Point centre outil. ....	56
Figure 15 :Division de la forme brute en cellules homogènes. ....	57
Figure 16: Dimensions d'une cellule. ....	58
Figure 17: Affectation des sommets aux cellules de la matrice. ....	58
Figure 18: Enveloppe de la sphère de l'outil. ....	60
Figure 19 : Cellules candidates à l'interférence. ....	61
Figure 20: Surface de la sphère. ....	62

Figure 21 :Sommet S est à l'intérieur de la sphère de l'outil. ....	62
Figure 22:Sommet S est sur la surface de la sphère de l'outil.....	62
Figure 23 : Sommet S est à l'extérieur de la sphère de l'outil. ....	63
Figure 24 : Projection d'un sommet sur la droite portant l'axe du cylindre.....	64
Figure 25 : Point en collision avec le cylindre de l'outil.....	64
Figure 26 : Rotation de l'outil autour d'un axe.....	66
Figure 27 : Rotation d'un point P1 autour d'un axe quelconque.....	67
Figure 28 : Décomposition d'un outil hémisphérique.....	67
Figure 29 : Vecteurs de positionnement de l'outil. ....	68
Figure 30 : Détermination de point M de cercle.....	70

## chapitre 4

Figure 1 : Modèle CAO et modèle STL de la pièce de test. ....	71
Figure 2 :Page « Lecture du fichier STL ». ....	73
Figure 3 : Enveloppe de la pièce brute et sommets du modèle STL.....	73
Figure 4 : Triangles du modèle STL. ....	73
Figure 5 : Normales aux sommets. ....	74
Figure 6 :Page « Création des contours ». ....	75
Figure 7 : Plans d'usinage. ....	75
Figure 8 : Bandes de triangles. ....	76
Figure 9 : Points d'intersection (points de contact outil-surface).....	76
Figure 10 : Contours de contact outil-surface. ....	76
Figure 11 :Page « Points de passage de l'outil » ....	77
Figure 12 : Points de passage de l'outil. ....	78
Figure 13 : Page « Détection et correction des interférences et des collisions ».....	79
Figure 14: Points en interférences. ....	80
Figure 15 : Points en collisions. ....	80
Figure 16: Orientation de l'outil avant et après la correction des collisions.....	81
Figure 17 : Page « Simulation ». ....	81

Figure 18 : Outil en position 1. ....82

Figure 19 : Outil en position 2. ....82

Figure 20 : Outil en position 3. ....82

# **INTRODUCTION**

## **GENERALE**

## INTRODUCTION GENERALE

### I. SITUATION DU PROBLEME :

Aujourd'hui, la machine sous toutes ses formes est extrêmement sophistiquée, elle a évolué grâce aux techniques développées pour la conception des pièces. Ces pièces parfois prennent des formes géométriques de base comme un cube, une sphère, etc., mais d'autres fois, les formes deviennent très complexes pour des raisons de performance ou d'apparence. Le plus souvent, ces formes complexes ne sont pas exprimables par des formules mathématiques connues, ce qui donne le nom des surfaces libres (surfaces gauches). C'est la raison pour laquelle, des modèles géométriques ont été développés pour les représenter, ce qui rend la conception de ces pièces plus ou moins simples. Avec l'intégration de l'outil de l'informatique, un nombre important de logiciels de conception « Conception Assistée par Ordinateur » -CAO- sont développés et commercialisés pour aider les concepteurs dans leurs tâches de conception. Une fois la pièce est conçue, il est possible de prendre son modèle en utilisant différents format d'échange de données standards (STL, STEP, IGES, DWG, ...etc.). Dans certains cas où la géométrie de la pièce est tellement complexe que sa conception dans un logiciel de CAO devienne couteuse ou impossible, le processus du Reverse Engineering est utilisé. Dans ce processus, une maquette (prototype) est réalisée manuellement et elle est scannée par un dispositif spécifique pour récupérer un nuage de points très dense. Ce dernier est traité pour le convertir en un modèle triangulé (modèle STL).

Pour la partie production de ces pièces, les machines-outils qui produisent ces pièces ont profité de l'outil informatique. Une commande numérique a été intégrée à ces machines pour automatiser le processus d'usinage. Parmi ces machines, les fraiseuses à 02-axes, à 03-axes, à 04-axes et à 05-axes qui sont utilisées pour la fabrication de ces pièces. Leurs formes finales sont obtenues en trois opérations : ébauchage, demi-finition et finition. La génération des trajets d'outils associés à chaque opération exige la prise en compte de plusieurs paramètres tels que : modèles des surfaces (continus, discrets), outils d'usinage (formes et dimensions), cinématique de la machine (type et nombre d'axes), stratégies d'usinage (en fonction de l'opération d'usinage), précisions exigées (hauteur de crête, erreur de flèche), problèmes d'usinage (interférences, collisions), ...etc. Le nombre important des paramètres à considérer et leurs interdépendances rendent la tâche de génération de trajets d'outils sains et optimums très complexe.

Pendant l'usinage, l'outil suit une trajectoire complexe dans l'espace 3D et de ce fait, des problèmes d'interférence, entre la partie active de l'outil et la surface à usiner, ainsi que des problèmes de collisions, entre les différentes parties de l'environnement d'usinage (machine, outils, pièce, ...etc.), peuvent surgir ce qui peuvent endommager l'environnement d'usinage. Pour cela, il est impératif d'éliminer ces problèmes avant de

lancer l'usinage réel. Dans l'industrie, certaines formes de très grande complexité géométrique telles que aubes de turbines, hélices des navires, rotors, ...etc. ne peuvent être usinées sur des machines 03-axes. Cette contrainte exige le passage à l'usinage en 05-axes. Pour cet usinage, en chaque point de contact, l'outil tout en étant tangent à la surface à usiner, une infinité d'orientations de son axe sont possibles. Les deux degrés de liberté supplémentaires (deux rotations) augmentent les possibilités d'orientation de l'outil et d'accès aux différentes régions de la surface, mais en même temps augmentent considérablement les risques d'interférences et de collisions. Parmi les orientations possibles, il est indispensable de choisir l'orientation la plus optimale évitant ces problèmes et optimisant l'usinage.

Le présent projet s'inscrit dans le cadre du programme triennal 2014-2016 de recherche de l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO », de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA », intitulé « Production des Surfaces de Formes Libres sur des Fraiseuses Numériques à 05 axes ». L'objectif de ce projet est la proposition de solutions logicielles permettant l'automatisation et l'optimisation du processus de production des pièces de formes quelconques sur des fraiseuses numériques à 05-axes.

## **II. OBJECTIF DU TRAVAIL :**

Dans ce projet et à partir des modèles STL des surfaces complexes, nous nous intéressons à l'automatisation de l'opération de finition de ces surfaces sur les fraiseuses à commande numérique à 05-axes en utilisant un outil hémisphérique et la stratégie d'usinage Z-Constant. L'objectif étant la détermination des points de passage de l'outil et les orientations de l'outil associées à chaque point de passage permettant d'éliminer les interférences et les collisions. La finalité de ce projet est le développement d'une application logicielle graphique et interactive sous Windows permettant d'automatiser ce processus.

## **III. STRUCTURATION DU MEMOIRE :**

Le présent mémoire est composé des chapitres suivants :

- Le premier chapitre est consacré à l'étude du processus d'usinage des surfaces complexes.
- L'étude conceptuelle de notre application logicielle est menée dans le deuxième chapitre.
- La démarche de résolution et les solutions proposées sont présentées dans le troisième chapitre.
- Le dernier chapitre est réservé à la phase de tests et de validations des résultats.

# **CHAPITRE I**

## **GENERALITES SUR LE PROCESSUS D'USINAGE DES SURFACES COMPLEXES**



## INTRODUCTION :

Dans la pratique industrielle, pour concevoir de nouveaux produits, l'ingénieur concepteur peut choisir un logiciel de Conception Assistée par Ordinateur -CAO- parmi un nombre très important de logiciels du marché. Une fois la phase de conception est terminée, une deuxième phase commence et qui consiste à analyser, étudier les performances, déterminer les contraintes maximales admissibles, planifier les trajectoires d'usinage, ...etc. Ces études nécessitent généralement l'utilisation de logiciels spécifiques. Le problème qui se pose est l'échange des informations de conception entre ces logiciels. D'où la nécessité de développer des outils permettant de réaliser ces échanges dans un format neutre indépendant des logiciels. Pour cela, plusieurs formats d'échange de données sont créés tels que IGES, STL, VRML, DWG, DXF, STEP, ...etc. Le domaine de la conception et de la fabrication des pièces mécaniques avec des surfaces complexes est un domaine très complexe qui doit répondre aux différents paramètres lors de l'usinage comme le respect de la forme et des contraintes fonctionnelles et/ou esthétiques. En raison de la complexité géométrique de ces pièces, les différentes opérations d'usinage sont effectuées sur des machines-outils à commande numérique. La forme finale des surfaces complexes est obtenue en trois étapes : ébauchage, demi-finition et finition. Pour la finition, différentes stratégies d'usinage, différentes formes d'outils et différentes configurations de machines d'usinage peuvent être utilisées. Donc, pour générer une pièce respectant toutes les exigences imposées dans le cahier des charges, il est indispensable d'étudier et d'analyser le processus d'usinage des surfaces complexes.

Ce chapitre est organisé en trois parties. Dans la première partie, nous allons étudier les différents modèles de représentation des surfaces complexes. La deuxième partie sera consacrée à la présentation des machines d'outils à commande numérique. La dernière partie est consacrée à l'étude du processus d'usinage des surfaces complexes sur des fraiseuses numériques à 5-axes.

## PARTIE 1 : MODELES ET PROPRIETES DES SURFACES

### 1. Catégories des modèles géométriques :

Deux catégories de modèles géométriques peuvent être utilisées pour décrire les surfaces frontières d'une pièce ou d'un objet au cours de sa conception.

**1.1. Modèles paramétriques :** ces modèles permettent de créer une représentation continue des surfaces d'un objet à l'aide de formulations polynômiales ou rationnelles définies dans un espace bi-paramétrique (Figure 1).

**1.1.1. Définition des surfaces paramétriques :** une surface paramétrique est définie par un ensemble de trois fonctions, une pour chaque coordonnée. Ces fonctions dépendent de deux paramètres  $u$  et  $v$ . Une surface paramétrique est donnée par [1]:

$$F(u, v) = (x(u, v), y(u, v), z(u, v)) \quad (1)$$

Où les paramètres  $u$  et  $v$  sont dans l'intervalle  $[0, 1]$ . Donc,  $(u, v)$  est un point d'un carré défini par les sommets  $(0,0)$ ,  $(1,0)$ ,  $(0,1)$  et  $(1,1)$  dans le plan des coordonnées

$uv$ . Pour chaque  $(u, v)$  lui correspond un point  $F(u, v)$  sur la surface paramétrique (Figure 1).

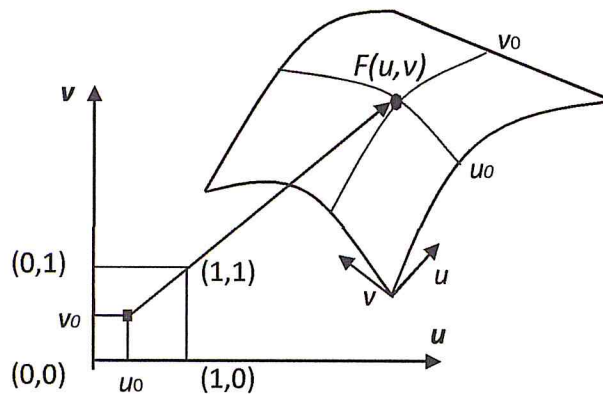


Figure 1 : Surface paramétrique [1].

Généralement, les surfaces paramétriques ne sont pas utilisées individuellement, mais plusieurs morceaux de surfaces paramétriques sont joints ensemble pour former une surface plus complexe. Mathématiquement, si  $x(u,v)$ ,  $y(u,v)$  et  $z(u,v)$  sont des polynômes, alors les surfaces paramétriques ne sont pas capables de représenter plusieurs surfaces usuelles telle que la sphère et l'ellipsoïde. Si  $x(u,v)$ ,  $y(u,v)$  et  $z(u,v)$  sont des fonctions rationnelles, alors les surfaces paramétriques peuvent représenter les sphères, les ellipsoïdes et plusieurs autres surfaces usuelles .

### 1.1.2. Modélisation des surfaces paramétriques:

La forme de la surface est déterminée une fois sa représentation mathématique est connue. Mais pratiquement, dans la plupart des 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. Pour résoudre ce problème, dans le domaine de la conception et de la modélisation des surfaces ils existent deux classes de méthodes qui peuvent aider le concepteur [2] :

- Méthodes basées sur l'utilisation des points : génération d'une surface à partir d'un ensemble de points décrivant la forme de la surface. Dans ce cas, trois approches peuvent être utilisées :
  - Interpolation : génération d'une surface qui passe ces points.
  - Approximation : génération d'une surface qui approxime le mieux un ensemble de points mais qui ne passe pas nécessairement par ces points.
  - Surface paramétrique : c'est une surface continue définie par un réseau de contrôle définissant la forme de la surface. Parmi ces surfaces, Bézier, Bézier Rationnelle, B-Spline et NURBS.

- Méthodes basées sur l'utilisation des courbes : dans ce cas, la surface est générée à partir d'un ensemble de courbes (surface réglée, surface balayée, surface de révolution, surface de Coons, surface de Gordon, ... etc.).

### 1.1.3. Propriétés géométriques des surfaces paramétriques :

- **Vecteurs tangents à la surface paramétrique :**

Les vecteurs tangents à la surface paramétrique au point  $F(u, v)$  sont :

Le vecteur tangent dans la direction  $u$  :

$$\vec{T}_u = \frac{\partial F}{\partial u} = \left( \frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) \quad (2)$$

Le vecteur tangent dans la direction  $v$  :

$$\vec{T}_v = \frac{\partial F}{\partial v} = \left( \frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right) \quad (3)$$

Ces deux vecteurs définissent le plan tangent à la surface au point  $F(u, v)$ .

- **Vecteur normal à la surface paramétrique :**

Le vecteur normal unitaire  $\vec{n}(u, v)$  à la surface au point  $F(u, v)$  est le produit vectoriel des deux vecteurs tangents (Figure 2) et il est donné par :

$$\vec{n} = \frac{\vec{T}_u * \vec{T}_v}{\|\vec{T}_u * \vec{T}_v\|} \quad (4)$$

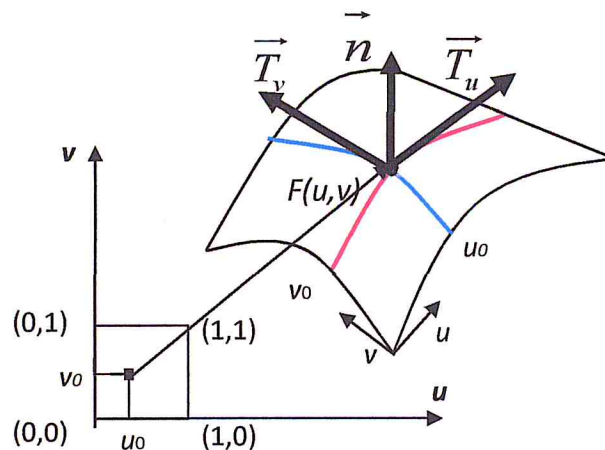


Figure 2 : Vecteurs tangents et vecteur normal d'une surface paramétrique [2].

**1.2. Modèle polyédrique :** il est basé sur une description des surfaces de la pièce ou de l'objet par un ensemble connecté d'entités géométriques simples tels que triangles, carrées, etc. appelées facettes [3].

- **Maillage triangulaire** : il est défini par une paire  $(P, K)$  tel que :
  - $P$  : est l'ensemble de  $N$  points  $P_i = (X_i, Y_i, Z_i) \in \mathbb{R}^3$  avec  $i \in \{1, \dots, N\}$ .
  - $K$  : complexe simplicial portant les informations topologiques du maillage. Il est composé de trois ensembles (Figure 3) :
    - Les sommets  $S = \{s/s = \{i\} \text{ et } i \in \{1, \dots, N\}\}$ .
    - Les arêtes  $A = \{a/a = \{i, j\}\} \text{ et } (i, j) \in \{1, \dots, N\}^2$ .
    - Les faces  $F = \{f/f = (i, j, k)\} \text{ et } (i, j, k) \in \{1, \dots, N\}^3$ .

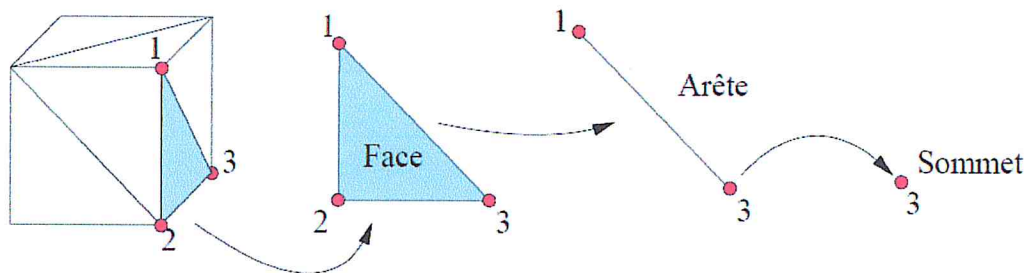


Figure 3 : Détail d'un maillage triangulaire [3].

- **Format STL** : le format STL est un maillage des surfaces de l'objet par un ensemble de triangles. « STL » est le nom court pour le Standard Tessellation Language, développé par la société 3D Systèmes. Il a été également créé pour l'impression 3D. Ce format de fichier décrit uniquement la géométrie des surfaces d'un objet en trois dimensions, car il ne mémorise que les informations géométriques nécessaires à l'impression 3D, développé à l'origine pour les besoins du prototypage rapide par Stéréolithographie, dans lequel la surface d'un solide (ou peau) est représentée par un maillage triangulaire (Figure 4). Donc, le fichier contenant le modèle « STL » est composé d'un ensemble de triangles où chaque triangle est décrit par les coordonnées  $X, Y$  et  $Z$  de ses trois sommets et par un vecteur normal  $\vec{N}$  qui indique l'orientation de la facette et de quel côté est situé l'extérieur (Figure 4). Dans ce format, un triangle partage ses trois segments avec trois triangles voisins ni plus ni moins. De plus, ce format inclut également des règles pour assurer que les triangles sont tous correctement connectés aux sommets. Chaque triangle doit partager deux sommets avec chacun de ses triangles adjacents. Donc, un sommet d'un triangle ne peut pas être localisé sur le côté d'un autre triangle (Figure 5).

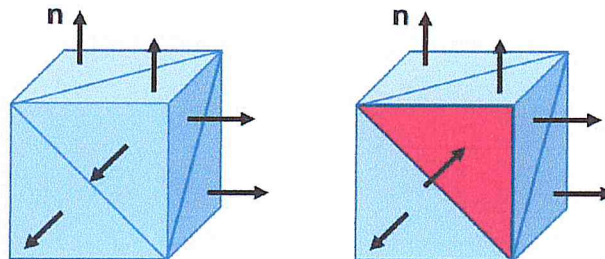


Figure 4 : Maillage triangulaire d'un cube [3].



Figure 5 : Type de maillage.

- **Structure d'un fichier STL (ASCII) :** deux types de fichiers peuvent être générés en fonction du besoin : fichier ASCII (texte) ou fichier binaire. La structure d'un fichier ASCII est la suivante :

```

solid [name]
  facet normal nx ny nz
    outer loop
      vertex x1 y1 z1
      vertex x2 y2 z2
      vertex x3 y3 z3
    endloop
  endfacet
.
.
.
endsolid [name]

```

- **Avantages du format STL :** le format STL a l'avantage d'être simple et très répandu dans l'industrie du prototypage rapide. Il est mathématiquement beaucoup plus simple et facile à trancher car il est facile de calculer les intersections des triangles avec des plans parallèles fictifs.
- **Inconvénients du format STL:** pour représenter fidèlement des formes très complexe, il est indispensable d'augmenter le nombre de triangles, ce qui rend alors le fichier très volumineux.

### 1.2.1. Propriétés géométriques des surfaces polyédriques :

- **Indicatrice et courbure gaussienne :** pour formuler une équivalence discrète à la courbure gaussienne, nous nous appuyerons sur la définition historique basée sur l'utilisation de l'indicatrice sphérique. Soit un sommet P d'un polyèdre (Figure 6),  $F_i$  ses faces voisines et  $N_i$  les normales à ces faces, nous utilisons l'application de Gauss : nous obtenons alors un point par face sur la sphère unité l'ensemble formant un polygone sphérique (Figure 7), dont l'aire est égale à  $e 2\pi - \sum_{k=0}^n \alpha_k$  où les  $\alpha_k$  sont les angles des faces voisines à P. On appelle cette valeur défaut angulaire car elle est égale à l'angle qu'il reste lorsque l'on coupe l'une des arêtes voisines à P pour déplier les faces. C'est en 1961 qu'une première écriture dite classique semble avoir été par Regge :

$$K Dis = \frac{(2\pi - \sum_i \alpha_i)}{(\frac{1}{3} \sum_i \text{aire}(F_i))} \tag{5}$$

Nous justifions cette pondération de 1/3 par la répartition de l'aire du triangle entre ses trois sommets.

En 1995, Boix propose une autre estimation, dite corrigée, qui fait intervenir les longueurs  $L_i$  des côtés des triangles opposés aux angles  $\alpha_i$ .

$$K Dis = \frac{(2\pi - \sum_i \alpha_i)}{(\frac{1}{3} \sum_i \text{aire}(F_i) - \frac{1}{8} \cot \sum_i \cot \alpha_i L_i^2)} \tag{6}$$

Les propriétés de convergence de cette formulation discrète vers la courbure gaussienne d'une surface lisse ont été démontrées.

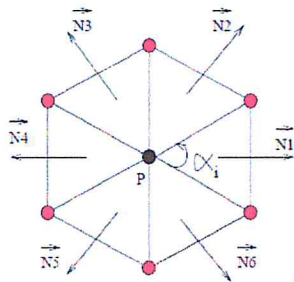


Figure 6 : Polyèdre [3].

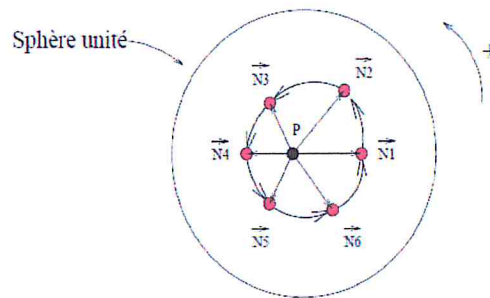


Figure 7 : Polygone sphérique [3].

- **Normale au sommet S** : On définit la normale au sommet S comme étant le vecteur normé calculé en tant que moyenne pondérée des normales aux faces voisines à S (Figure 8).

La pondération se faisant par l'angle au sommet ( $\alpha_i > 0$ ) de la face incidente portant la normale considérée.

$$\vec{n} = \frac{\frac{\sum_i \alpha_i \vec{N}_i}{\sum_i \alpha_i}}{\left\| \frac{\sum_i \alpha_i \vec{N}_i}{\sum_i \alpha_i} \right\|} = \frac{\sum_i \alpha_i \vec{N}_i}{\left\| \sum_i \alpha_i \vec{N}_i \right\|} \quad (\alpha_i > 0 \forall i) \tag{7}$$

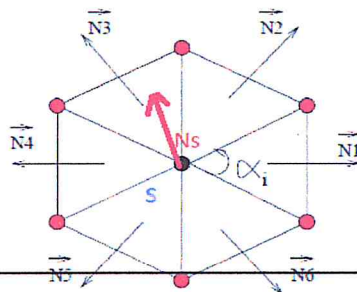
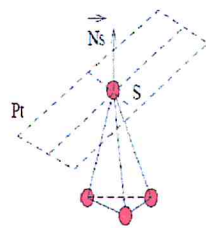


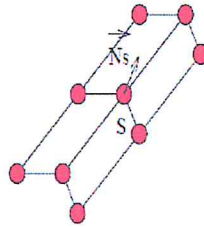
Figure 8 : Normale au sommet [3].

- **Plan tangent** : le plan tangent en un sommet S est ici considéré comme étant le plan passant par S et orthogonale à la normale N définie précédemment.

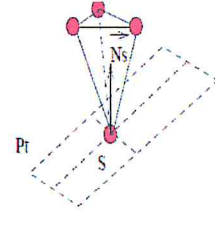
- **Type d'un sommet** : le type d'un sommet dépend de la position des sommets voisins par rapport au plan tangent.
  - Sommet convexe** :  $S$  est convexe si tous les sommets voisins sont au-dessous de son plan tangent  $P_t$  (Figure 9.a).
  - Sommet convexe parabolique** :  $S$  est convexe parabolique s'il est d'abord convexe et  $S$  appartient à une arête frontière de deux plans (Figure 9.b).
  - Sommet concave** :  $S$  est concave si tous ses voisins sont au-dessus de son plan tangent (Figure 9.c).
  - Sommet concave parabolique** : c'est un cas des sommets concaves ou  $S$  appartient à une arête frontière (Figure 9.d).
  - Sommet plat ou plan** : si le cas ou  $S$  est concave et convexe (Figure 9.e).
  - Sommet selle** : c'est sommet ayant la particularité d'être à l'intérieur de l'enveloppe convexe de ses voisins (Figure 10-G).



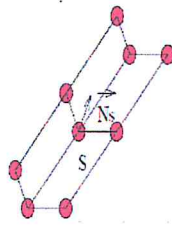
a. Sommet convexe.



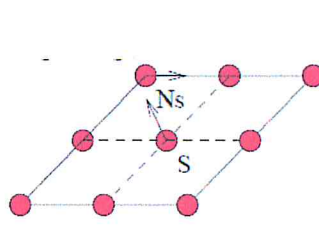
b. Sommet convexe parabolique.



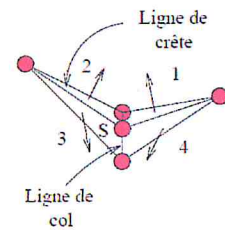
c. Sommet concave.



d. Sommet concave parabolique.



e. sommet plat.



f. Sommet selle.

Figure 9 : Type des sommets [3].

- **Sens relatif au sommet** : le sens positif relatif au sommet est le sens défini par l'ordre de ses faces voisines triées dans le sens trigonométrique autour de la normale à  $S$ . Le sens négatif c'est le sens inverse du sens positif (Figure 10).

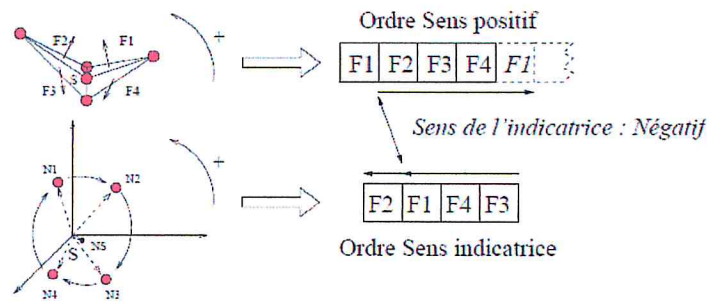


Figure 10 : Sens relatif au sommet [3].

## PARTIE 2 : MACHINES D'USINAGE

### 1. Définition d'une MOCN:

Une Machine-Outil à Commande Numérique (MOCN.) est une machine d'usinage à cycle automatique programmable (Figure 11). Ce type de machines se compose ainsi de deux parties complémentaires : la partie opérative (c'est la machine-outil : elle agit directement sur le produit à réaliser) et la partie commande (c'est la commande numérique qui permet d'élaborer des ordres en fonction des consignes et des comptes rendus) [4]. Les surfaces de formes complexes (surfaces gauches) sont usinées sur des fraiseuses.

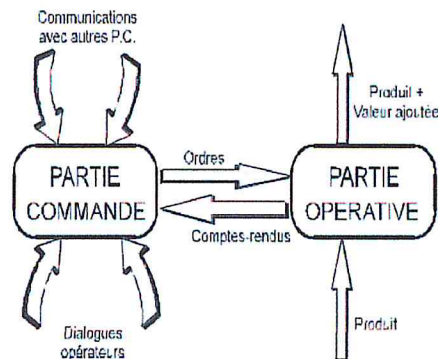


Figure 11 : Schémas d'une MOCN [4].

### 2. Axes des fraiseuses :

- **Mouvements de translation** : lors de l'usinage, l'outil ou la pièce peut être animé de trois mouvements de translation (Figure 12) :
  - **Axe Z** : est parallèle à la broche principale de la machine.
  - **Axe X** : quand cela est possible, l'axe X doit être horizontal et parallèle à la surface de bridage de la pièce.
  - **Axe Y** : cet axe forme avec les axes X et Z un trièdre de sens direct.

Le sens positif des axes est défini de manière telle qu'un mouvement dans une direction positive d'axes de translation augmente les valeurs positives de la position de la pièce par rapport à la machine.



- **Mouvements de rotation A, B, C** : lors de l'usinage, l'outil ou la pièce peut être animé de trois mouvements de rotation (Figure 12). Les angles A, B et C définissent les mouvements de rotation effectués respectivement autour des axes parallèles à X, Y et Z.

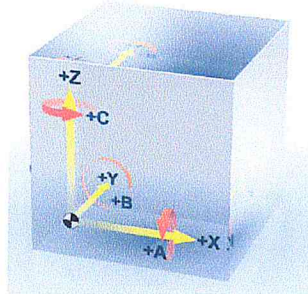


Figure 12 : Axes de la fraiseuse [5].

### 3. Classification des fraiseuses :

- **Fraiseuses à 3 axes** : deux configurations sont possibles : broche verticale (axe Z est vertical) (Figure 13) et broche horizontale (axe Z est horizontal) (Figure 14). Dans les deux cas, la fraise est perpendiculaire à la table. Dans le cas de la broche horizontale, il y a une meilleure évacuation des copeaux et du liquide de lubrification qui, sinon, peut s'accumuler dans les parties creuses.

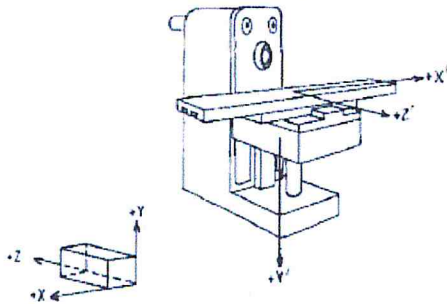


Figure 13 : Fraiseuse horizontale [[4].

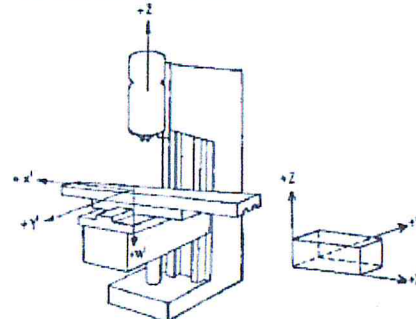


Figure 14 : Fraiseuse verticale [4].

- **Fraiseuses à 4 axes** : c'est souvent une fraiseuse 3 axes dotée d'un plateau tournant. C'est une configuration très pratique en production industrielle mécanique (automobile, aviation etc.)

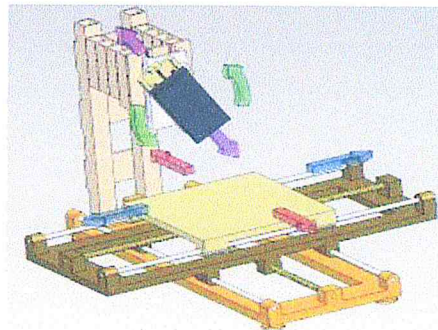


Figure 15 : Fraiseuse à 4 axes [4].

• **Fraiseuses à 5 axes** : nous pouvons trouver sous cette catégorie plusieurs topologies de constructions. Une fraiseuse 5 axes comporte toujours trois axes linéaires (X, Y, Z) et deux axes rotatifs choisis parmi A, B et C. Les machines vont se différencier par la position des axes rotatifs. Les trois types sont :

- Deux axes rotatifs sur la tête (Figure 16.a).
- Deux axes rotatifs sur la table (Figure 16.b).
- Un axe rotatif sur la tête et un axe rotatif sur la table (Figure 16.c).

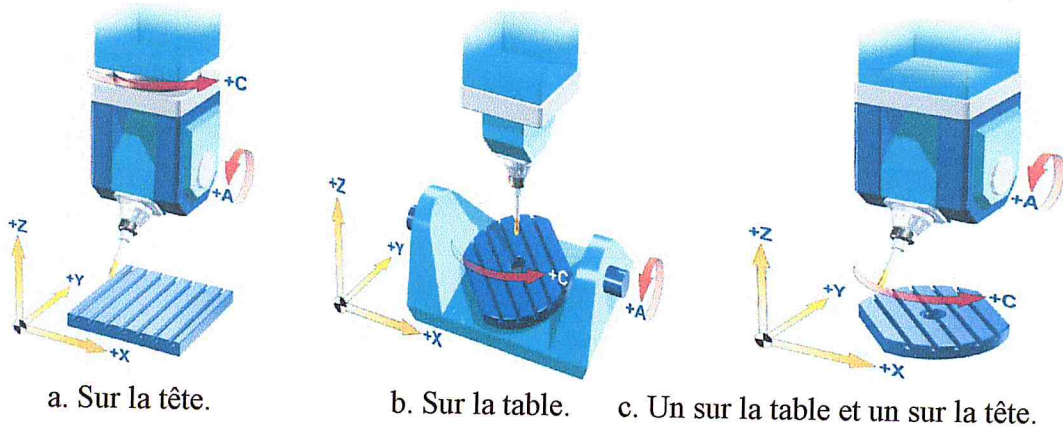


Figure 16 : Répartition des axes rotatifs sur des fraiseuses à 5 axes [5].

#### 4. Programmation des MOCN :

La programmation actuelle des machines-outils se fait par l'intermédiaire du langage G ou G-code dont les principes sont regroupés dans les normes ISO 6983 et annexes. Le passage d'un langage de haut niveau au langage ISO est possible en utilisant un logiciel de traduction (post-processeur) (Figure 17). La programmation s'appuie sur des fonctions préparatoires de type G. Elle est complétée par des fonctions auxiliaires (de type M) et technologiques (F, S, etc....). Un programme en G-code est donc la traduction d'une suite d'actions et de déplacements élémentaires (ligne droite ou arc de cercle par exemple) qui permettent de générer des conditions d'usinage ainsi qu'une trajectoire plus ou moins approximée [6].

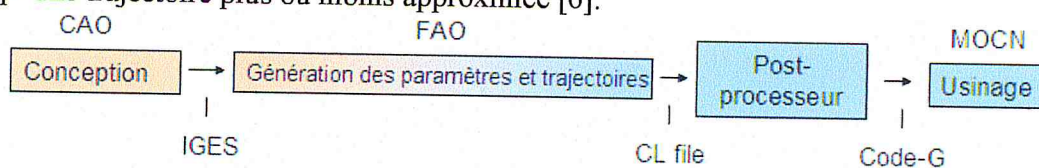


Figure 17 : Chaîne numérique de la conception à l'usinage [6].

#### • Quelques fonctions du code G [7] :

X, Y, Z coordonnées X, Y, Z,

F vitesse d'avance travail en mm/mn.

T appel d'un outil et des correcteurs liés.

G0 : interpolation linéaire sur les trois axes, en rapide.

G1 : interpolation linéaire sur les trois axes, à vitesse de travail.

G2 : interpolation circulaire sens horaire dans le plan XY.

**G3** : interpolation circulaire sens antihoraire dans le plan XY.

**G70** : programmation en pouces.

**G71** : programmation en métrique.

**M0** : arrêt programmé.

**M3** : rotation de la broche.

**M5** : arrêt de la broche.

**M30** : fin du programme.

**N** numéro de bloc.

**%** début de programme.

### PARTIE 3 : USINAGE DES SURFACES COMPLEXES

#### 1. Décomposition du processus d'usinage :

En général, la forme finale d'une pièce est obtenue en trois opérations (Figure 18) :

- **Ebauche** : elle permet d'enlever le maximum de matière (excédent important de matière) en un minimum de temps. L'objectif dans ce cas est d'augmenter au maximum le débit de copeaux. Donc, la machine doit être suffisamment puissante et l'outil devra résister à des efforts de coupe importants (outil massif) [8]. Elle permet d'éliminer la croûte superficielle liée au procédé d'obtention du brut par une opération d'écroûtage.

- **Demi-finition** : elle permet d'avoir une bonne approche de la surface finale, d'assurer la régularité du copeau de finition et la précision géométrique de position [8]. Elle permet de préparer une finition précise en corrigeant les défauts géométriques résultants d'une ébauche et donc d'assurer une surépaisseur constante et faible en finition.

- **Finition** : elle doit être précédée d'une ébauche et d'une demi-finition. L'objectif dans ce cas est l'obtention d'une bonne qualité de surface : dimensions, forme et rugosité dans les tolérances permises par le dessin de définition. Les efforts sont plus faibles que pour une ébauche et une demi-finition, l'outil est donc plus fin [8]. Elle permet de respecter toutes les spécifications imposées par le dessin de définition sur la ou les surfaces concernées par l'opération.

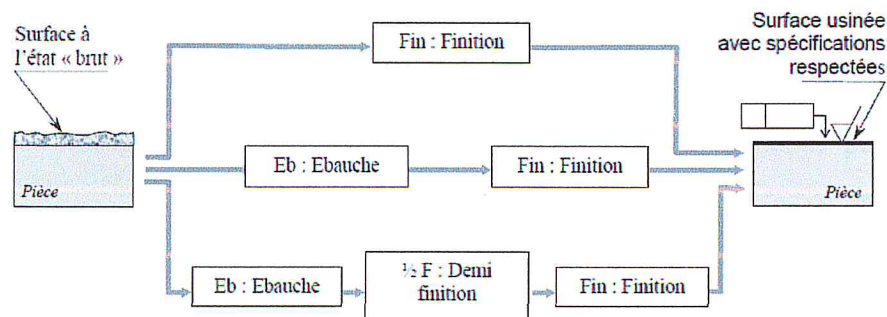


Figure 18 : Différentes phases d'usinage [8].

## 2.2. Paramètres de coupe :

Les paramètres de la coupe sont, d'une part, des grandeurs qui caractérisent les déplacements de l'outil et de la pièce usinée (paramètres de coupe cinématiques) et, d'autre part, les valeurs des surépaisseurs d'usinage et des dimensions de coupe (paramètres de coupe géométriques). La détermination des paramètres de coupe a une grande importance dans la production industrielle, car des régimes réduits augmentent considérablement le temps d'exécution de la pièce à usiner, et élèvent le prix de revient, aussi bien des régimes élevés ne sont pas avantageux non plus, parce que l'outil s'use rapidement, ce qui nous oblige à le changer très souvent, donc on aura toujours le même résultat que précédemment (augmentation du prix de revient de la pièce). Pour cela, les paramètres adoptés doivent être optimums pour assurer un prix de revient minimum de la pièce avec une plus grande productivité [10]. Les paramètres de coupe sont (Figure 21) :

- **Vitesse de coupe  $V_c$**  : c'est le déplacement d'un point de l'arête tranchante de l'outil par rapport à la surface de coupe en une unité de temps. La vitesse de coupe est donnée par :

$$V_c = \pi \times D \times N / 1000 \text{ [m/min]}$$

Où  $D$  est le diamètre de la fraise et  $N$  la vitesse de rotation de la fraise en tour par minutes (tr/min).

- **Vitesse d'avance  $V_f$**  : elle est donnée par :

$$V_f = z \times f_z \times N \text{ [mm/min]}$$

Où  $z$  est le nombre de dents de la fraise,  $f_z$  est l'avance par dent et  $N$  la vitesse de rotation en tour par minutes (tr/min).

- **Profondeur de passe  $a$**  : la profondeur de passe notée  $a$  en [mm], correspond à la longueur de l'arête de coupe engagée dans la matière. La profondeur de passe est toujours mesurée perpendiculairement à la direction de l'avance et non pas suivant l'arête de l'outil.

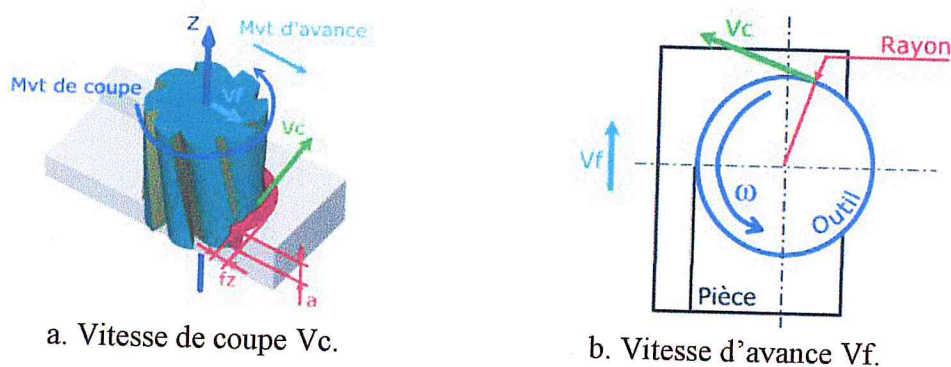


Figure 21 : Vitesses en fraisage [10].

Pour obtenir un bon état de la surface usinée, une rapidité de l'usinage, une usure modérée de l'outil, etc., il est indispensable de choisir judicieusement les paramètres de coupe (Figure 21). Il ressort de la Figure 22 que les paramètres de coupe dépendent de plusieurs paramètres (matière de la pièce, matière de l'outil, forme de l'outil, etc.).

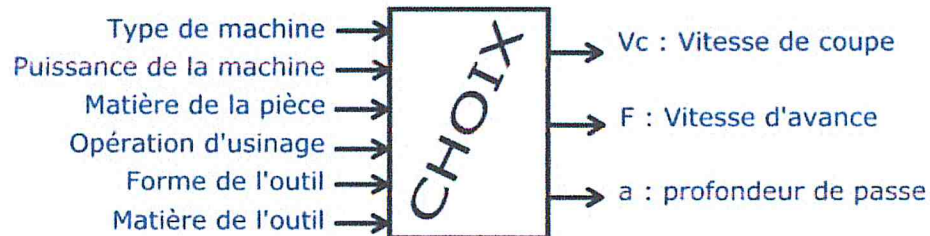


Figure 22 : Paramètres d'usinage [10].

Suivant le type d'opération à réaliser, il faut choisir la méthode d'usinage, et donc choisir la machine à utiliser (tour, fraiseuse, etc.). Pour ce qui est de la puissance de la machine, elle doit être importante pour l'opération d'ébauche et moins importante en finition. Pour la matière il est évident que les efforts de coupe ne sont pas les mêmes si vous usinez une pièce en polystyrène ou en acier. Donc la matière influe sur des choix relatifs à la puissance machine. En ce qui concerne l'opération d'usinage c'est la même idée que pour le type de machine. Même chose pour la forme de l'outil. Cependant, la matière de l'outil influe sur l'usure de l'outil et sa durée du fait que c'est l'outil qui doit usiner la pièce et non l'inverse. Tous ces critères sont intimement liés et étant donné que l'objectif final est d'obtenir une pièce usinée dans de bonnes conditions. Pour cela, il faut déterminer certains paramètres spécifiques (vitesse de coupe  $V_c$ , vitesse d'avance  $V_f$  et profondeur de passe  $a$ ). En ébauche,  $V_c$  est minorée,  $V_f$  ou  $f_z$  majorée et une grande profondeur de passe est prise. En finition  $V_c$  est majorée,  $V_f$  ou  $f_z$  minorée et une faible profondeur de passe est prise.

### 3. Différents types d'outils utilisés pour les opérations de fraisage :

Trois formes d'outils sont utilisées dans le processus d'usinage des pièces de formes complexes :

- **Fraises cylindriques** : pour que ce type d'outil admette le talonnage, il faut considérer les outils à coupe au centre qui autorisent la plongée lors de l'usinage si l'arête de coupe arrive jusqu'au centre de l'outil, la vitesse de coupe  $y$  est faible et l'évacuation des copeaux n'est pas bonne (Figure 23.a) [11].

- **Fraises hémisphériques** : ces fraises sont très intéressantes car elles peuvent pivoter autour du point centre outil sans collision entre l'outil et la pièce. C'est un avantage considérable lors d'un changement de direction d'usinage qui permet de l'effectuer sans perdre le contact outil/pièce (Figure 23.b). Les positionnements adoptés avec ce type d'outil font souvent apparaître un angle d'inclinaison de la broche afin d'amener le point de contact, c'est à dire le point générateur, assez loin de l'axe pour obtenir une vitesse de coupe plus élevée [11].

- **Fraises toriques :** ce sont des outils intermédiaires entre les fraises cylindriques et les fraises hémisphériques qui sont de plus en plus utilisés dans l'industrie. Ils possèdent des avantages considérables et le plus important d'entre eux est d'avoir une partie active, le tore, situé sur un rayon éloigné de l'axe de l'outil. La conséquence directe est que cette partie active voit une vitesse non nulle et donc que la coupe est bonne (Figure 23.c) [5].

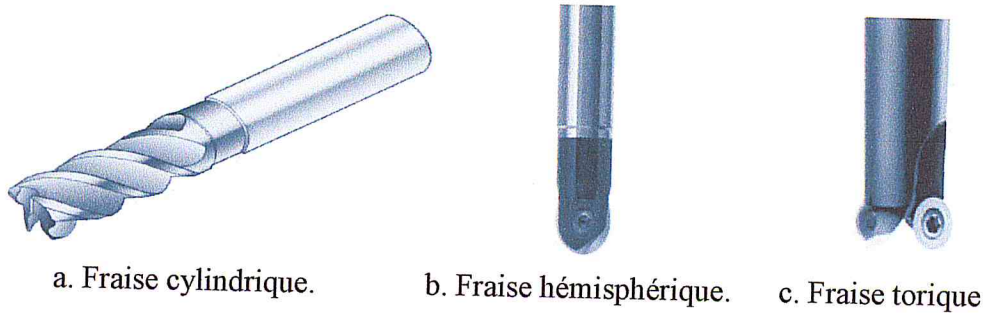


Figure 23 : Fraises d'usinage [11].

#### 4. Positionnement des différentes formes d'outils :

Le positionnement d'un outil sur une surface gauche est décomposé en deux étapes : la définition du point de contact outil/pièce et ensuite l'orientation de l'axe de l'outil. Lors de cette deuxième étape, l'axe de l'outil n'est pas forcément aligné avec la normale à la surface au point considéré. Nous pouvons introduire un angle  $\alpha$  positif entre l'axe de l'outil et la normale dans le plan  $(n, t)$  qui inclinera l'outil vers l'avant (dans le sens du mouvement d'avance); nous parlerons alors de détalonnage ou bien d'épinçage (Figure 24). Il peut aussi introduire un angle  $\alpha$  négatif c'est à dire vers l'arrière de l'outil (dans le sens inverse du mouvement d'avance); et nous parlerons de détalonnage négatif ou encore de talonnage .

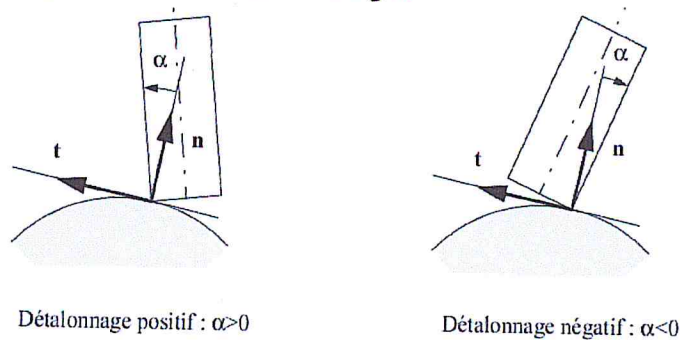
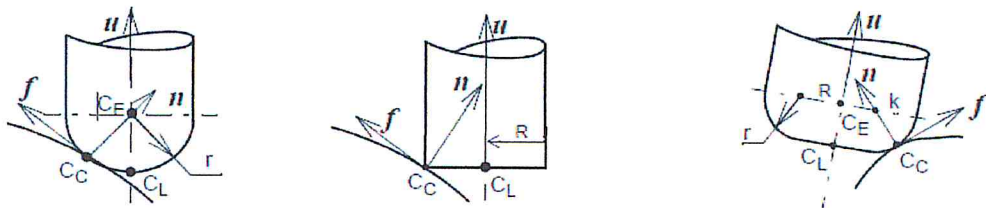


Figure 24 : Types de détalonnage [12].

Le positionnement de l'outil sur la surface consiste à déterminer un couple  $(P, u)$ , où  $P$  représente le point piloté, caractéristique de la géométrie de l'outil, et  $u$  le vecteur directeur de l'axe de l'outil. Classiquement, le point piloté de l'outil est le point CL (Cutter Location), point extrémité. Ce couple est défini localement par rapport à la surface à usiner, au niveau du point de contact outil-pièce CC (Figure 25). Le repère

local, associé  $(CC, f, n, t)$  est tel que  $f$  représente le vecteur tangent à la courbe suivie,  $n$  le vecteur normal à la surface, et  $t$  le vecteur issu du produit vectoriel de  $f$  par  $n$ . La Figure suivante positionne les points caractéristiques de l'outil suivant sa géométrie par rapport à la base locale.  $CE$  représente le centre de l'outil,  $R$  le rayon principal de l'outil et  $r$  le rayon du coin [8].



$$OC_E = OC_C + r \cdot n$$

$$OC_L = OC_E - r \cdot u$$

$$OC_L = OC_C + R \cdot v$$

$$v = \frac{u \wedge n}{\|u \wedge n\|} \wedge u$$

$$OC_E = OC_C + r \cdot n + R \cdot v$$

$$v = \frac{u \wedge n}{\|u \wedge n\|} \wedge u$$

a. Outil hémisphérique.

b. Outil cylindrique.

c. Outil torique.

Figure 25: Points caractéristiques suivant la géométrie de l'outil [12].

Une fois la position de l'extrémité de l'outil déterminée, il reste à définir l'orientation de son axe. Elle est donnée dans le repère local  $(CC, f, n, t)$  par deux angles de rotation (Figure 26). Deux modes sont utilisés dans l'inclinaison de l'axe de l'outil :

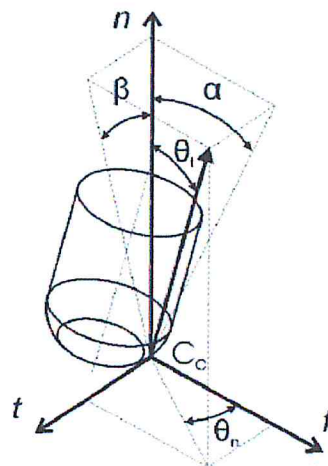


Figure 26 : Angles d'orientation de l'axe outil [12].

- **Premier mode** : il consiste à définir le couple d'angles  $(\theta_t, \theta_n)$ , où  $\theta_t$  représente l'angle d'inclinaison « inclination angle », entre  $u$  et  $n$ ,  $\theta_n$  l'angle de pivotement « screw angle » autour de  $n$ . Donc, l'orientation de l'axe de l'outil est obtenue par une première rotation de  $\theta_t$  autour de  $t$  puis une seconde rotation de  $\theta_n$  autour de  $n$ .
- **Deuxième mode** : il consiste à définir le couple d'angles  $(\alpha, \beta)$ , où  $\alpha$  est l'angle de détalonnage « lead angle », défini dans le plan contenant la direction d'avance, autour de  $t$ , et  $\beta$  est l'angle d'inclinaison latérale « tilt angle », défini dans le plan perpendiculaire à la direction d'avance autour de  $f$  [12].

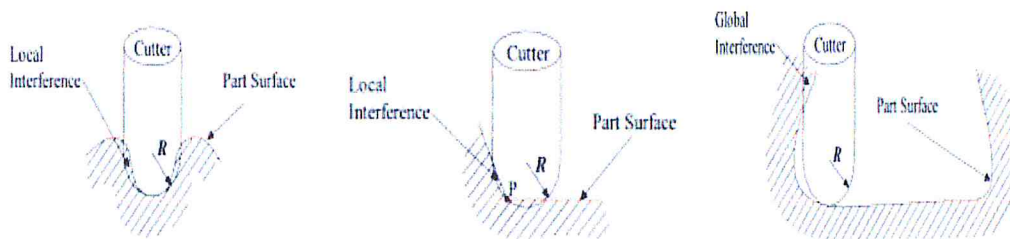
## 5. Types d'interférences :

Lors du positionnement de l'outil en usinage 5-axes, trois problèmes peuvent être rencontrés :

**5.1. Interférences locales :** elles traduisent un enlèvement de matière excessif par la partie active de l'outil sur la surface à usiner. Elles sont principalement dues aux différences de courbures de la partie active de l'outil et de la surface au point de contact (Figure 27.a).

**5.2. Interférences vers l'arrière :** sont des pénétrations intempestives de l'arrière de l'outil dans la surface à usiner (Figure 27.b).

**5.3. Interférences globales (collisions) :** elles représentent de manière générale des collisions entre l'ensemble {corps d'outil, porte outil, broche} et l'ensemble {pièce, porte pièce, éléments de structure...} (Figure 27.c). Du fait des deux rotations supplémentaires, le risque de collision est plus fréquent en usinage 5 axes qu'en usinage 3 axes ; celles-ci peuvent apparaître au niveau des positionnements outils calculés, lors de l'interpolation de la trajectoire ou encore lors des mouvements hors matière. Pour un point de contact outil pièce donné, toute la difficulté consiste à déterminer l'orientation de l'axe de l'outil qui soit la plus efficace.



a. Interférence locale.      b. Interférence vers l'arrière.      c. Interférence globale.

**Figure 27:** Points caractéristiques suivant la géométrie de l'outil [13].

## 6. Stratégies d'usinage en finition :

Pour usiner des surfaces de formes complexes, plusieurs stratégies peuvent être utilisées :

### 6.1. Plans parallèles :

Cette stratégie consiste à déterminer sur la surface  $S(u, v)$  des passes  $C(s)$  pour lesquelles les points de contact outil-pièce  $CC$  ou les points  $CL$  sont contenus dans des plans parallèles  $P_i$  de la surface (Figure 28). Au sein de chaque passe, les positionnements outils sont déterminés par cheminement le long des plans de guidage  $P_i$ , les paramètres de discrétisation associés sont la variation entre deux



positions successives  $(\Delta u, \Delta v)$  dans l'espace paramétrique (sens longitudinal) et la distance entre deux plans successifs  $\Delta p$  (sens transversal) [13]. De plus, elle peut être adoptée pour l'usinage des surfaces définies par leurs modèles STL. Cette stratégie est appropriée pour les surfaces proches du plan horizontal (faibles inclinaisons).

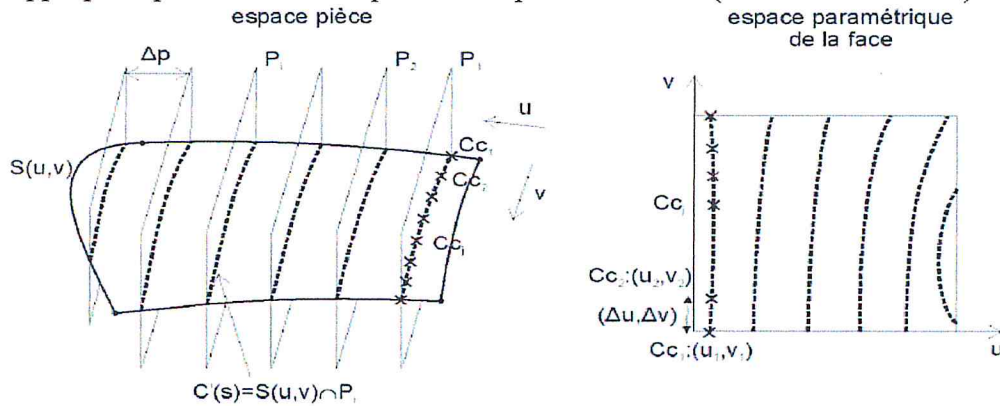


Figure 28: Stratégie par plans parallèles [12].

### 6.2. Isoparamétrique :

Elle consiste à balayer la surface paramétrique  $S(u, v)$  selon une succession de courbes isoparamétriques  $C^*(v) = S(u^*, v)$  obtenues en fixant l'un des paramètres  $u$  ou  $v$ . Dans ce cas, les paramètres de discrétisation de la trajectoire dans les deux directions sont directement les variations  $\Delta u$  et  $\Delta v$  définies dans l'espace paramétrique (Figure 29) [13].

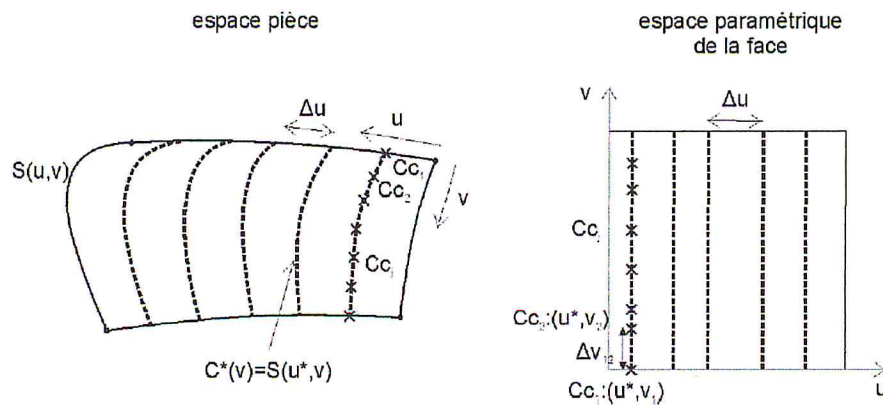


Figure 29 : Stratégie Isoparamétriques [12].

### 6.3. Z-Constant :

Cette stratégie consiste à déterminer sur la surface  $S(u, v)$  des passes  $C(s)$  pour lesquelles les points de contacts outil-pièce  $CC$  ou les points  $CL$  sont contenus dans des plans horizontaux  $P_i$  de la surface (Figure 30). Au sein de chaque passe, les positionnements outils sont déterminés par cheminement le long des plans de guidage  $P_i$ . La distance entre deux plans horizontaux successifs représente la profondeur de passe [9]. Les passes d'usinage en Z-Constant sont générées à partir d'un ensemble de contours définis sur la surface qui décrivent la forme de la pièce à

différentes positions en Z, par découpage horizontal de la géométrie. Cette stratégie est la plus efficace pour les semi-finitions et finitions des surfaces à forte pente. En limitant l'usinage aux pentes comprises entre 30 et 90 degrés, les zones les plus anguleuses sont usinées, laissant les zones les plus plates pour un usinage plus approprié. De plus, elle peut être adoptée pour l'usinage des surfaces définies par leurs modèles STL. Cette stratégie est appropriée pour les surfaces proches du plan vertical (fortes inclinaisons).

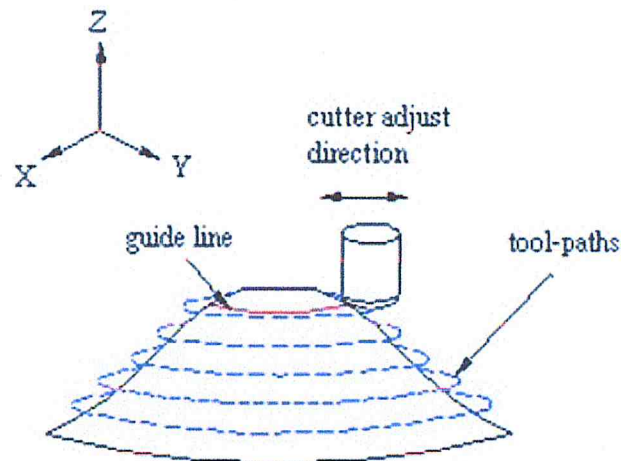


Figure 30 : Mode de balayage par Z-constant [14] .

#### 6.4. Comparaison des stratégies :

Le tableau 1 donne une comparaison entre les trois stratégies d'usinage en fonction des types de surfaces à usiner, des outils utilisés et de la position des surfaces. Dans le cahier de charges imposées au début du projet, les surfaces complexes à usiner sont définies par des modèles STL et la majorité des surfaces à usiner sont proches de la verticale. Donc, il ressort de ce tableau que la stratégie d'usinage la plus appropriée est Z-Constant.

Tableau 1 : Comparaison des trois stratégies d'usinage.

Stratégie d'usinage	Types de surfaces	Outils utilisés	Position de la surface
Isoparamétrique	Paramétrique	Hémisphérique Cylindrique Torique	Quelconque
Plan- parallèle	Paramétrique, STL	Hémisphérique Cylindrique Torique	Proche du plan l'horizontale
Z-constant	Paramétrique, STL	Hémisphérique Cylindrique Torique	Proche du plan verticale

## 7. Critères de précision :

Afin de déterminer la qualité des surfaces usinées, deux critères sont définis [14] :

- Erreur de corde maximale  $e_{max}$  : l'erreur de corde « tolérance d'usinage » représente l'écart maximal entre la surface à usiner et l'enveloppe du mouvement de l'outil généré par l'interpolation des positionnements outils le long d'une passe [8] (Figure 31). Il est important de noter que dans le cas des trajectoires 5 axes, l'évaluation de l'erreur de corde doit prendre en compte l'écart généré par la variation de l'orientation de l'axe de l'outil.
- Hauteur de crête maximale  $h_{c \max}$  : la hauteur de crête qualifie le surplus de matière laissée entre deux passes successives (Figure 32). En 5 axes, l'évaluation de la hauteur de crête est complexe du fait de la libre orientation de l'axe de l'outil ; le profil générateur de la passe est 3D. Pour évaluer la hauteur de crête, l'approximation couramment utilisée dans la littérature consiste à rendre le problème plan en projetant le profil de l'outil dans un plan perpendiculaire à la direction d'avance.

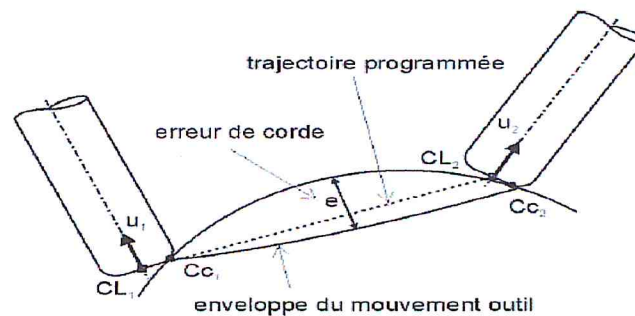


Figure 31: Erreur de corde[12] .

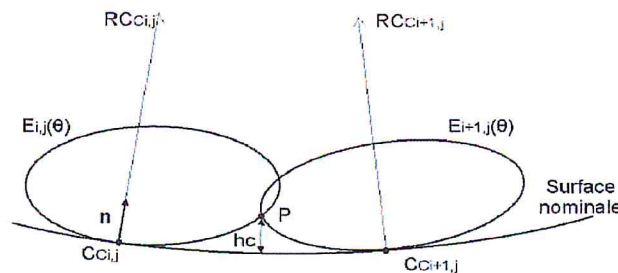


Figure 32: Hauteur de crête en 5 axes [12].

## CONCLUSION :

Dans ce chapitre, nous avons étudié les modèles et propriétés des surfaces complexes (continues et discrètes) ainsi que le format d'échange de données STL. Par la suite, nous avons mis l'accent sur les différentes configurations des fraiseuses numériques et leurs modes de programmation. Dans la dernière partie, nous avons étudié le processus d'usinage des surfaces de formes complexes en 5-axes.

Avec ces connaissances, nous pouvons maintenant modéliser notre problème et proposer des solutions pour le résoudre.

## CHAPITRE II

# ANALYSE ET CONCEPTION

## INTRODUCTION :

Après avoir étudié dans le chapitre précédent le processus d'usinage des pièces de forme complexes, nous passons maintenant à l'étape de conception. Dans ce chapitre, nous allons définir les besoins de notre système et la liste des objectifs à atteindre afin de proposer les solutions adéquates. Nous commençons par la sélection d'une méthode ou un langage pour la modélisation de notre système. Par la suite, nous utilisons cet outil choisi pour détailler la structure de notre système et son déroulement.

### 1. Outil de modélisation :

La réalisation de l'application logicielle nécessite une étape de conception avec une méthode ou un langage de modélisation pour exprimer les besoins et les fonctionnalités de cette application ainsi que son comportement pendant le déroulement de ces fonctionnalités. UML (Unified Modeling Language) est l'un des langages destinés à la modélisation des logiciels. Il est indépendant du langage d'implémentation et il peut décrire les fonctionnalités du système à l'aide des différents diagrammes statiques et comportementaux. Pour modéliser une application avec UML, il est indispensable de définir les besoins de l'utilisateur avant de passer à la conception des solutions qui répondent à nos besoins définis précédemment. L'expression des besoins se fait à l'aide de diagrammes de cas d'utilisation de l'application. La conception est réalisée avec des diagrammes statiques tels que diagramme de classe et diagrammes comportementaux comme diagramme d'activité, diagramme de séquence et diagramme de collaboration ... etc.

### 2. Conception :

- **Cadre du sujet :**

Le présent projet s'inscrit dans le cadre du programme triennal 2014-2016 de recherche de l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO », de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA », intitulé « Production des Surfaces de Formes Libres sur des Fraiseuses Numériques à 05 axes ». L'objectif de ce projet est la proposition de solutions logicielles permettant l'automatisation et l'optimisation du processus de production des pièces avec des surfaces de formes quelconques sur des fraiseuses numériques à 05-axes.

- **Problématique :**

Les pièces mécaniques avec des surfaces complexes utilisées dans différentes industries (automobile, aérospatial, ...etc.) doivent à des contraintes fonctionnelles et/ou de style. En raison de leurs géométries très complexes, elles sont usinées sur des fraiseuses numériques de 03-axes à 05-axes après la génération des trajets d'outils des opérations d'ébauchage, de

demi-finition et de finition. La production de ces pièces exige la prise en compte de plusieurs paramètres tels que : modèles des surfaces (continus, discrets), outils d'usinage (formes et dimensions), cinématique de la machine (type et nombre d'axes), stratégies d'usinage (en fonction de l'opération d'usinage), précisions exigées (hauteur de crête, erreur de flèche), problèmes d'usinage (interférences, collisions), ...etc. Le nombre important des paramètres à considérer et leurs interdépendances rendent la tâche de génération de trajets d'outils sains et optimums très complexe.

Pendant l'usinage, l'outil suit une trajectoire complexe dans l'espace 3D et de ce fait, des problèmes d'interférence entre la partie active de l'outil et la surface à usiner ainsi que des problèmes de collisions entre les différentes parties de l'environnement d'usinage (machine, outils, pièce, ...etc.) peuvent surgir ce qui peuvent endommager la machine. Pour cela, il est impératif d'éliminer ces problèmes avant de lancer l'usinage réel. Dans l'industrie, certaines formes de très grande complexité géométrique telles que aubes de turbines, hélices des navires, rotors, ...etc. ne peuvent être usinées sur des machines 03-axes. Cette contrainte exige le passage à l'usinage en 05-axes. Pour cet usinage, à chaque point de contact, une infinité d'orientations de l'axe de l'outil sont possibles et l'outil est toujours tangent à la surface à usiner. Les deux degrés de liberté supplémentaires (deux rotations) augmentent les possibilités d'orientation de l'outil et d'accès aux différentes régions de la surface, mais en même temps augmentent significativement les risques d'interférences et de collisions. Parmi les orientations possibles, il est indispensable de choisir l'orientation la plus optimale évitant ces problèmes et optimisant l'usinage.

- **Objectifs :**

Notre objectif principal est le développement d'une application logicielle graphique et interactive sous Windows permettant la génération d'un trajet d'outil sain (sans interférences et sans collisions) lors de la finition des surfaces complexes, à partir de leurs modèles STL, sur des fraiseuses numériques à 05-axes par des outils hémisphériques en utilisant la stratégie d'usinage Z-Constant. Cet objectif peut être subdivisé en plusieurs sous objectifs :

1. **Premier objectif :** lecture du modèle STL de la pièce à usiner et récupération de ces paramètres.
2. **Deuxième objectif :** détermination des plans d'usinage et création des contours d'usinage.
3. **Troisième objectif :** calcul des points de passage de l'outil (points centre-outil).
4. **Quatrième objectif :** détermination des outils optimums.
5. **Cinquième objectif :** détection des interférences et des collisions
6. **Sixième objectif:** détermination des orientations de l'outil évitant les collisions.
7. **Septième objectif :** simulation virtuelle des mouvements et des orientations de l'outil.

- **Plateforme exigé :**
  - Intel pentium I3 (2.6GHZ) ou équivalent.
  - Une RAM de 4 GO Minimum.
  - Carte graphique 3D de 1GO compatible OpenGL ou équivalent.
  - System d'exploitation : Windows 7.
  - Logiciel requise : Builder c++ version 6.
  - Bibliothèque graphique OpenGL.

### 3. Modélisation de l'application avec UML :

Dans ce qui suit, nous allons présenter le fonctionnement de notre système en utilisant les diagrammes suivants du langage de modélisation UML :

- Diagramme de cas d'utilisation.
- Diagramme de classes.
- Diagramme d'activités.
- Diagrammes de séquence.
- Diagramme de collaboration.

#### 3.1. Diagrammes de cas d'utilisations :

Un cas d'utilisation (use case) modélise une interaction entre le système informatique à développer et un utilisateur ou un acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur. Nous allons présenter les différentes fonctionnalités de notre système par un diagramme de cas d'utilisation global qui décrit les principales fonctionnalités du système (Figure 1). Les diagrammes d'utilisation pour les cas « créer contours » et « gérer interférence et collision » sont donnés respectivement par les Figures 2 et 3.



Figure 1 : Diagramme de cas d'utilisation global.



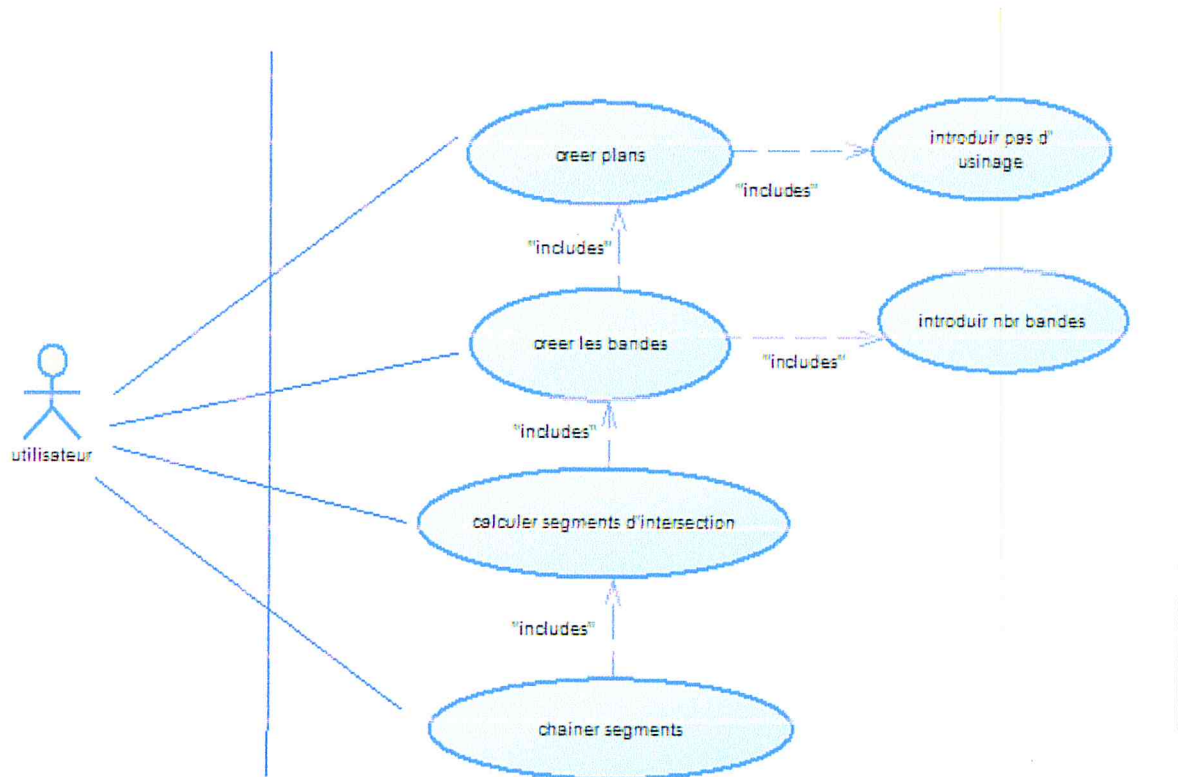


Figure 2 : Diagramme de cas d'utilisation « créer contours ».

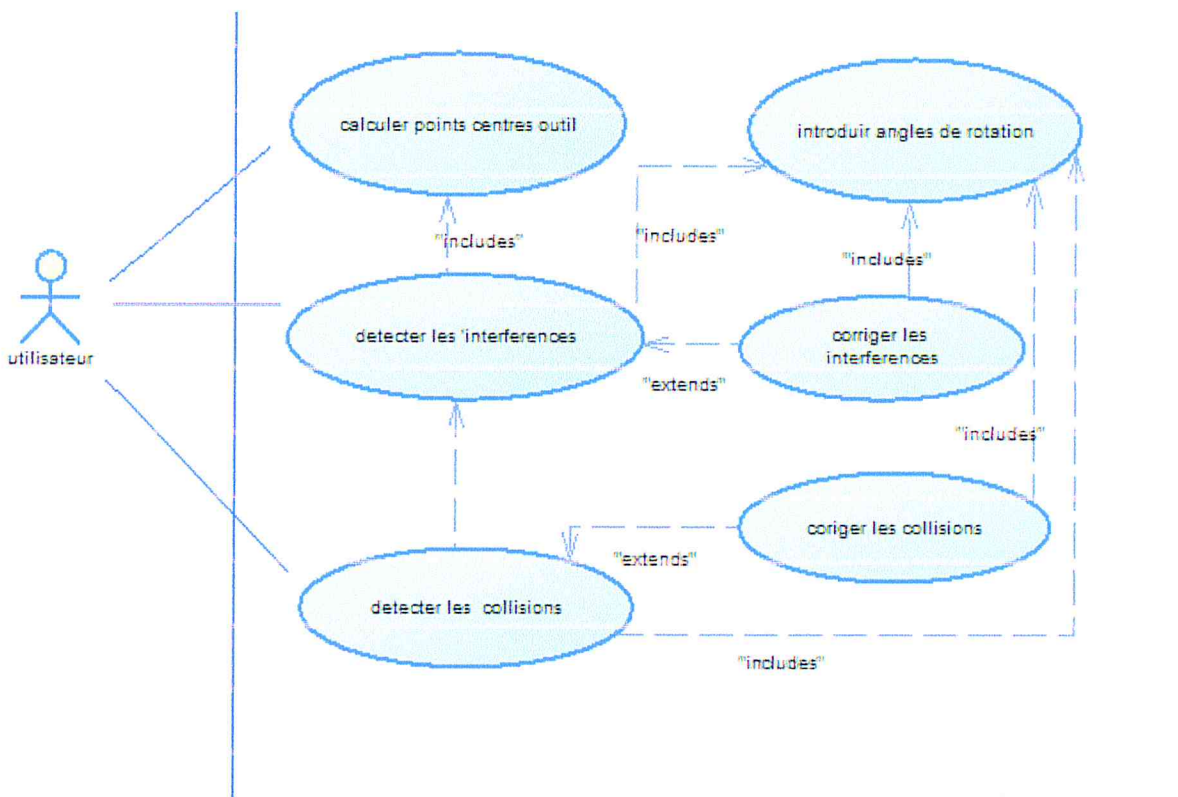


Figure 3 : Diagramme de cas d'utilisation « gérer interférence et collision ».

### 3.1. Diagramme de classes :

Un diagramme de classes décrit le type des objets ou données du système ainsi que les différentes formes de relations statiques qui les relient entre eux. Classiquement, ils existent deux principaux types de relations entre objets :

- Les associations, bien connues des vieux modèles entité/association utilisés dans la conception des bases de données depuis les années 70.
- Les sous-types, particulièrement en vogue en conception orientée objets, puisqu'ils s'expriment très bien à l'aide de l'héritage en programmation.

Après l'étape de définition et d'analyse des besoins, nous allons passer à l'étape suivante pour montrer les solutions que nous avons proposées pour résoudre la problématique en prenant en considération les besoins définis précédemment et en décrivant l'architecture du système et les relations entre ses différents composants à l'aide du diagramme de classes (Figure 4). Les principales classes utilisées dans notre conception sont les suivantes :

- |                        |                       |
|------------------------|-----------------------|
| 1.Point.               | 15.Centre_Outil.      |
| 2.Sommet_Stl.          | 16.Outil              |
| 3.Nœud_Sommet_Stl.     | 17.Contours_Stl.      |
| 4.Nuage_Points.        | 18.Nœud_Contour_Stl.  |
| 5.Sommet_Intersection. | 19.Liste_Contour_Stl. |
| 6.Triangle_Stl.        | 20.Trajectoire.       |
| 7.Nœud_Triangle_Stl.   |                       |
| 8.Maillage_Triangles.  |                       |
| 9 .Segment_Stl.        |                       |
| 10.Nœud_Segment_Stl    |                       |
| 11.Liste_Segment_Stl.  |                       |
| 12.Bande.              |                       |
| 13.Liste_Bandes.       |                       |
| 14.Plan_Stl.           |                       |

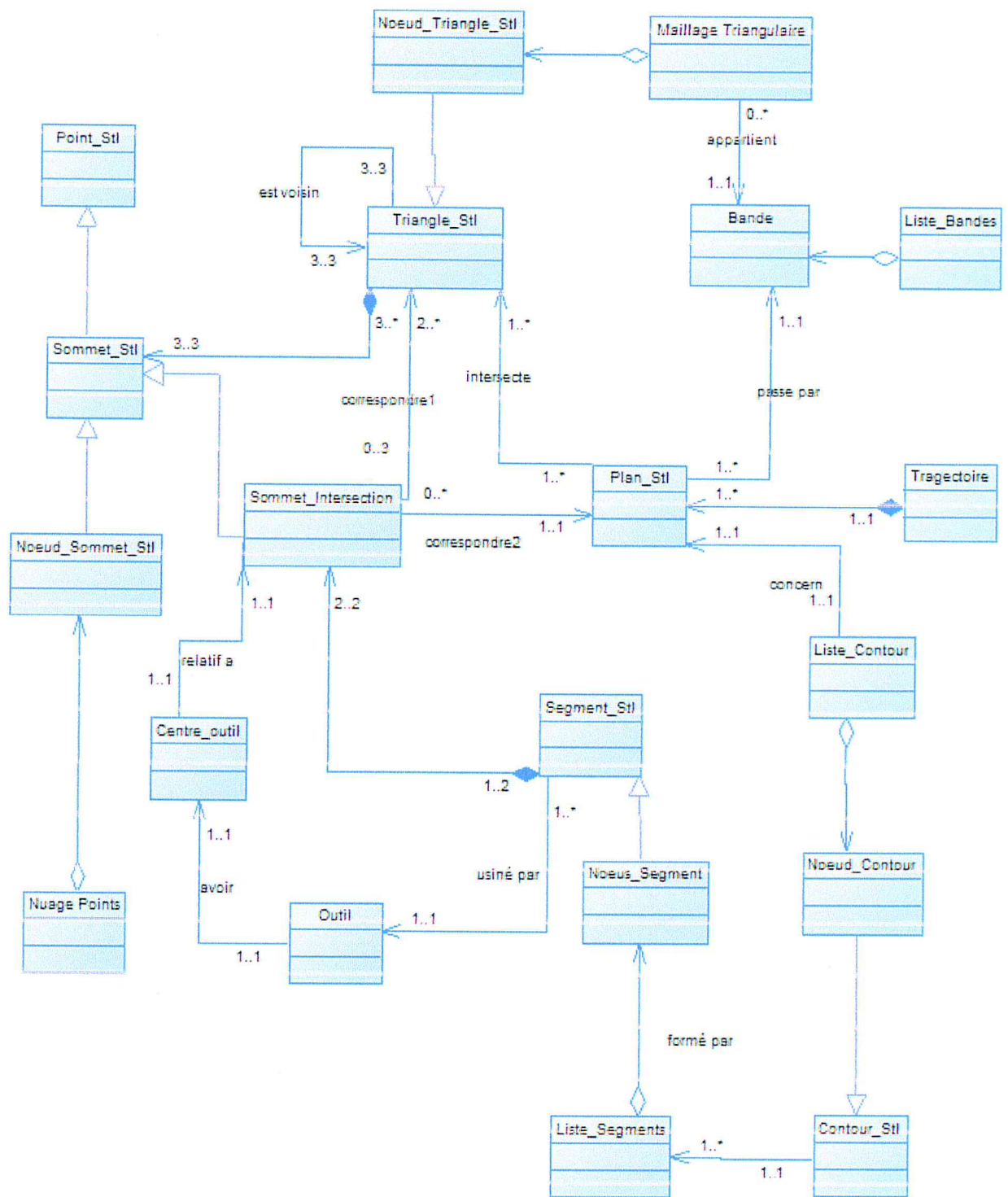


Figure 4 : Diagramme de classes.

**Description des classes :** les principales classes utilisées sont :

1. Classe Point (Figure 5) :

Point_Stl	
- x	: double
- y	: double
- z	: double
+ Point_Stl () : int	

**Figure 5 :** Classe Point\_Stl.

2. Classe Sommet\_Stl (Figure 6) :

Sommet_Stl	
- nx	: double
- ny	: double
- nz	: double
+ Sommet_Stl () : void	
+ visualiser_sommet () : void	
+ est_identique () : bool	
+ dessiner_Normale_Sommet () : bool	

**Figure 6 :** Classe Sommet\_Stl.

3. Classe Nœud\_Sommet\_Stl(Figure 7) :

Nœud_Sommet_Stl	
- tete	: Sommet_Stl*
- suivant	: Nœud_Sommet_Stl *
- Attribut_3	: int
+ Nœud_Sommet_Stl() () : void	

**Figure 7 :** Classe Nœud\_Sommet\_Stl.

4. Classe Nuage\_Points (Figure 8) :

Nuage Points	
- taille	: int
- dernier	: int
- vecteur_sommets	: Sommet_Stl*
+ Liste_Sommets () : void	
+ getposition () : Sommet_Stl*	
+ visualiser_tous_sommets () : void	
+ dessiner_Tous_Normale_Sommet () : void	

**Figure 8 :** Classe Nuage\_Points.

5. Classe Sommet\_Intersection (Figure 9) :

Sommet_Intersection	
+ Sommet_Intersection () : void	

**Figure 9 :** Classe Sommet\_Intersection.

## 6. Triangle\_Stl (Figure 10) :

Triangle_Stl	
- sommet1	: Sommet_Stl *
- sommet2	: Sommet_Stl *
- sommet3	: Sommet_Stl *
- nx	: double
- ny	: double
- nz	: double
+ Triangle_Stl ()	: void
+ visualiser_triangle ()	: void
+ visualiser_aire_triangle ()	: void
+ calculer_intersection ()	: void
+ afficher_normale ()	: void

Figure 10 : Classe Triangles\_Stl.

## 7.. Classe Noeud\_Triangle\_Stl (Figure 11) :

Noeud_Triangle_Stl	
- triangle	: Triangle_Stl *
- suivant	: Noeud_Triangle_Stl*
+ Noeud_Triangle_Stl ()	: void

Figure 11 : Classe Noeud\_Triangle\_Stl.

## 8. Classe Maillage\_Triangles (Figure 12) :

Maillage Triangles	
- taille	: int
- dernier	: int
- vecteur_triangles	: Triangle_Stl *
+ Liste_Triangles ()	: void
+ visualiser_list_triangle ()	: void
+ visualiser_list_aire_triangle ()	: void
+ afficher_Tous_normale ()	: void

Figure 12 : Classe Maillage\_Triangles.

## 9. Classe Segment\_Stl (Figure 13) :

Segment_Stl	
- sommet_debut	: Sommet_Stl *
- sommet_fin	: Sommet_Stl *
+ Segment_Stl ()	: void
+ est_suivant ()	: bool
+ visualiser_Segment ()	: void
+ inverser_Segment ()	: void
+ dessiner_Normale_Segment ()	: void

Figure 13 : Classe Segment\_Stl.

## 10. Classe Noeud\_Segment\_Stl (Figure 14) :

Noeud_Segment_Stl	
- segment	: Segment_Stl *
- suivant	: Noeud_Segment_Stl*
+ Noeud_Segment_Stl ()	: void

Figure 14 : Classe Noeud\_Segment\_Stl.

## 11. Classe Liste\_Segment\_Stl (Figure 15) :

Liste_Segment_Stl	
- tete_liste	: Noeud_Segment_Stl*
- queu_liste	: Noeud_Segment_Stl*
- nombre_segments	: int
<hr/>	
+ Liste_Segment_Stl ()	: void
+ inserer_Segment_Tete ()	: void
+ inserer_Segment_Queue ()	: void
+ visualise_liste_points ()	: void
+ est_vide ()	: bool
+ segment_suivant ()	: Segment_Stl*
+ segment_Precedent ()	: Segment_Stl*
+ chaîner_Noeud_Segment_Queue ()	: void
+ chaîner_Noeud_Segment_Tete ()	: void
+ supprimer_Noeud_Segment ()	: void
+ inverser_Liste_Segments ()	: void
+ supprimer_Segment ()	: void
+ dessiner_liste_normal_points ()	: void

Figure 15 : Classe Liste\_Segment\_Stl.

## 12. Classe Bande (Figure 16) :

Bande	
- zmin_bande	: double
- zmax_bande	: double
- vect_triangle_bande	: Liste_Triangle_Dynamique *
<hr/>	
+ Bande ()	: void
+ affecter_Triangle_bande ()	: bool
+ visualiser_bande ()	: void
+ visualiser_aire_bande ()	: void
+ separer_bande ()	: void
+ separer_aire_bande ()	: void

Figure 16 : Classe Bande.

## 13. Liste\_Bandes (Figure 17) :

Liste_Bandes	
- zmin_globale	: double
- zmax_globale	: double
- largeur	: double
- nombre_bande	: int
- lst_tr	: Maillage_Triangles*
- vect_bandes	: Bande*
<hr/>	
+ Liste_Bandes ()	: void
+ affecter_triangle_liste_bandes ()	: void
+ dessiner_triangles_bande ()	: void
+ visualiser_liste_bande ()	: void
+ visualiser_aire_liste_bande ()	: void
+ separer_tous_bandes ()	: void
+ separer_aire_tous_bandes ()	: void

Figure 17 : Classe Liste\_Bandes.

## 16. Classe Centre\_Outil (Figure 20) :

Outil_Stl	
- limite_cylindre	: void*
- extrimite_cylindre_min	: Point_Stl *
- extrimite_cylindre_max	: Point_Stl *
- extremite_spher_min	: Point_Stl *
- extremite_spher_max	: Point_Stl *
- centre	: Centre_Outil *
- rayon	: double
- longueur	: double
+ Outil_Stl ()	: void
+ calculer_enveloppe_cylindre ()	: void
+ calculer_enveloppe_sphere ()	: void
+ dessiner_enveloppe_cylindre ()	: void
+ rotation_outil ()	: void
+ distance_entre_point_deraction_cylindre()	: double
	()
+ detecter_collision_point ()	: bool
+ detecter_collision_cellule ()	: bool
+ detecter_collision ()	: bool

Figure 20 : Classe Outil.

## 17. Classe Contours\_Stl (Figure 21) :

Contour_Stl	
- list_seg	: Liste_Segment_Stl *
- num_contour	: int
- vecteur_position_outil	: Centre_Outil *
+ Contour_Stl ()	: void
+ visualiser_Point_Contour ()	: void
+ visualiser_Segments_Contour ()	: void
+ visualiser_vecteur_position_outil ()	: void
+ generer_vecteur_position_outil ()	: void
+ visualiser_vecteur_centre_outile_en_	: void
interference ()	
+ visualiser_vecteur_centre_outile_en_	: void
collision ()	

Figure 21 : Classe Contours\_Stl.

## 18. Classe Nœud\_Contours\_Stl (Figure 22) :

Nœud_Contour_Stl	
- cont	: Contour_Stl*
- suivant	: Nœud_Contour_Stl*
+ Nœud_Contour_Stl ()	: void

Figure 22 : Classe Nœud\_Contours\_Stl.

## 19. Classe Liste\_Contours\_Stl (Figure 23) :

Liste_Contours_Stl	
- tete_liste	: Noeud_Contour_Stl*
- nombre_contours	: int
<hr/>	
+ Contours_Stl ()	: void
+ inserer_Contour ()	: void
+ visualiser_Points_Liste_Contour ()	: void
+ visualiser_Segment_Liste_Contour ()	: void
+ supprimer_Noeud_Contour ()	: void
+ generer_position_outil_pour_liste_	: void
contour ()	
+ visualiser_liste_position_outile ()	: void
+ visualiser_liste_centre_outile_	: void
eninterference ()	
+ visualiser_liste_centre_outile_	: void
collision ()	

Figure 23 : Classe Liste\_Contours\_Stl.

## 20. Classe Trajectoire (Figure 24) :

Trajectoire	
- vect_plan	: Plan_Stl *
- taille	: int
- trajectoire_brute	: Liste_Segment_Stl *
- liste_provisoire	: Liste_Segment_Stl *
- pas	: double
- Attribut_6	: int
<hr/>	
+ Trajectoire ()	: void
+ visualiser_tous_plans ()	: void
+ visualiser_aire_tous_plans ()	: void
+ calculer_intesection_plans ()	: void
+ visualiser_tous_points ()	: void
+ visualiser_tous_segments ()	: void
+ creer_contours ()	: void
+ visualiser_tous_segments_contours ()	: void
+ creer_Trajectoire_Brute ()	: void
+ dessiner_normal_segments_plan ()	: void
+ visualiser_tous_point_passge_utile ()	: void
+ visualiser_tous_point_passge_en_	: void
interference ()	
+ visualiser_tous_point_passge_en_	: void
collision ()	

Figure 24 : Classe Trajectoire.

## 3.3. Diagrammes d'activités :

Les diagrammes d'activité sont utilisés pour documenter le déroulement des opérations dans un système, du niveau commercial au niveau opérationnel (de haut en bas). En regardant un diagramme d'activité, nous trouverons les éléments des diagrammes d'état. En fait, le diagramme d'activité est une variante du diagramme d'état où les « états » représentent des opérations et des transitions représentant les activités qui se produisent quand l'opération est terminée. L'usage général des diagrammes d'activité permet de faire apparaître les flots de traitements induits par les processus internes par rapport aux événements externes. Le diagramme d'activité global, le diagramme d'activité pour « créer contours » et le diagramme d'activité pour « gérer interférence et collision » sont représentées respectivement par les Figures 25, 26 et 27.



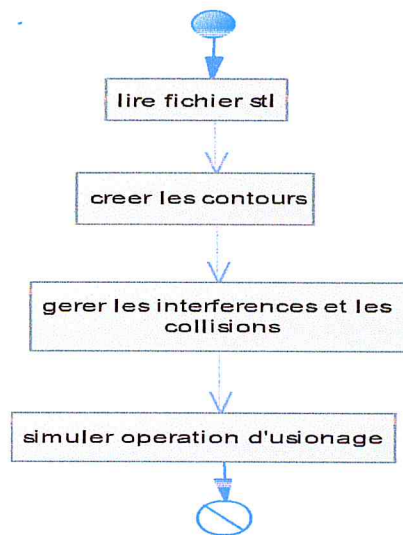


Figure 25 : Diagramme d'activité global.

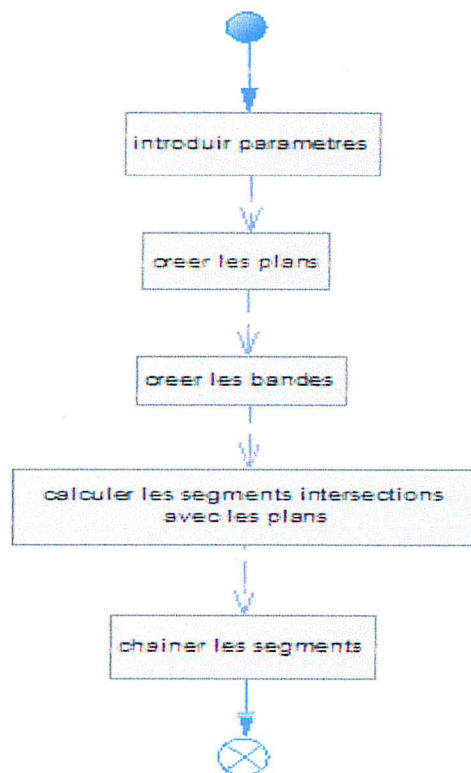


Figure 26 : Diagramme d'activité pour « créer contours ».

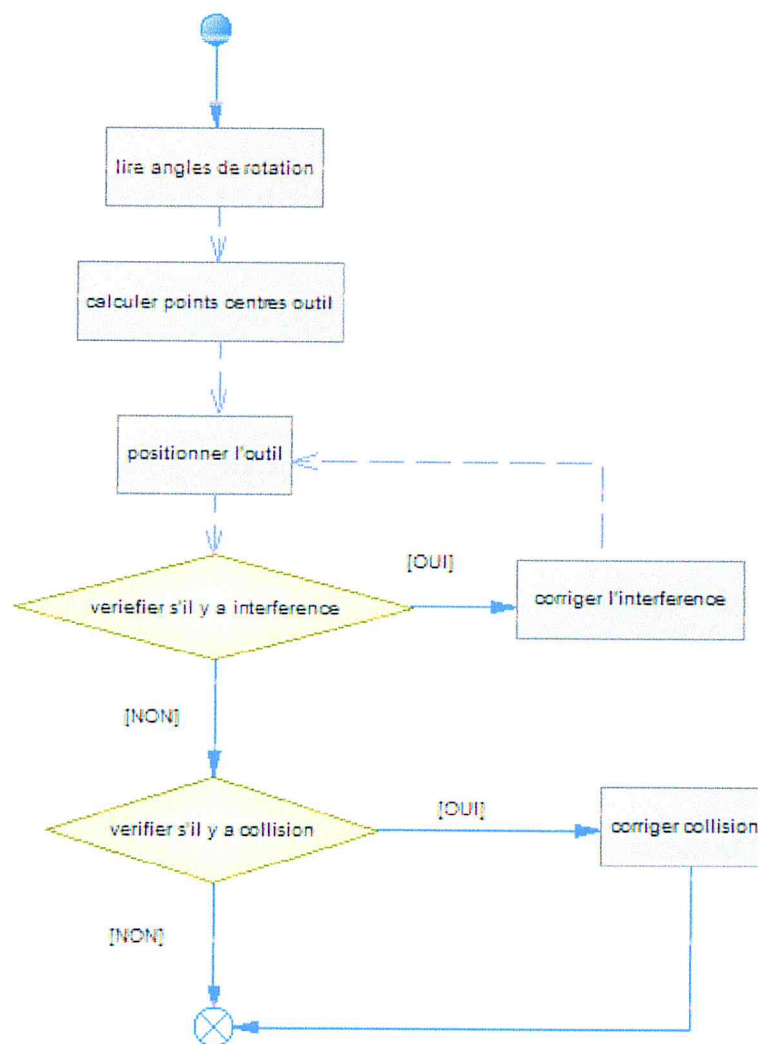


Figure 27 : Diagramme d'activité pour « gérer interférence et collision ».

### 3.4. Diagrammes de séquences :

Les diagrammes de séquences mettent en valeur les échanges de messages (déclenchant des événements) entre acteurs et objets (ou entre objets et objets) de manière chronologique, l'évolution du temps se lisant de haut en bas. Chaque colonne correspond à un objet (décrit dans le diagramme des classes), ou éventuellement à un acteur, introduit dans le diagramme des cas. La *ligne de vie* de l'objet représente la durée de son interaction avec les autres objets du diagramme. Un diagramme de séquences est un moyen semi-formel de capturer le comportement de tous les objets et acteurs impliqués dans un cas d'utilisation. Les diagrammes de séquence pour « lire fichier STL », « créer contours » et « gérer interférence et collision » sont représentés respectivement par les Figures 28, 29, 30.

**La lecture d'un fichier STL se fait par les étapes suivantes (Figure 28) :**

- L'utilisateur demande de récupérer le nuage de points et le maillage de triangles.
- L'objet interface demande la lecture de fichier STL par l'objet lecture.

- L'objet lecture demande la création de liste sommet initiale par l'objet liste de sommets initiale.
- L'objet liste sommets initiale crée une instance de liste de sommets initiale.
- L'objet lecture demande la création de liste initiale de triangles par l'objet liste triangles initiale.
- L'objet liste de triangles initiale crée une instance de liste de triangles initiale.
- L'objet interface demande la création de nuage de points par l'objet Nuage points.
- L'objet nuage points crée une instance de nuage de points.
- L'objet interface demande la création de maillage de triangles par l'objet maillage de triangles.
- L'objet maillage de triangles crée une instance de maillage de triangles.
- L'interface return à l'utilisateur le nuage de points et le maillage de triangles.

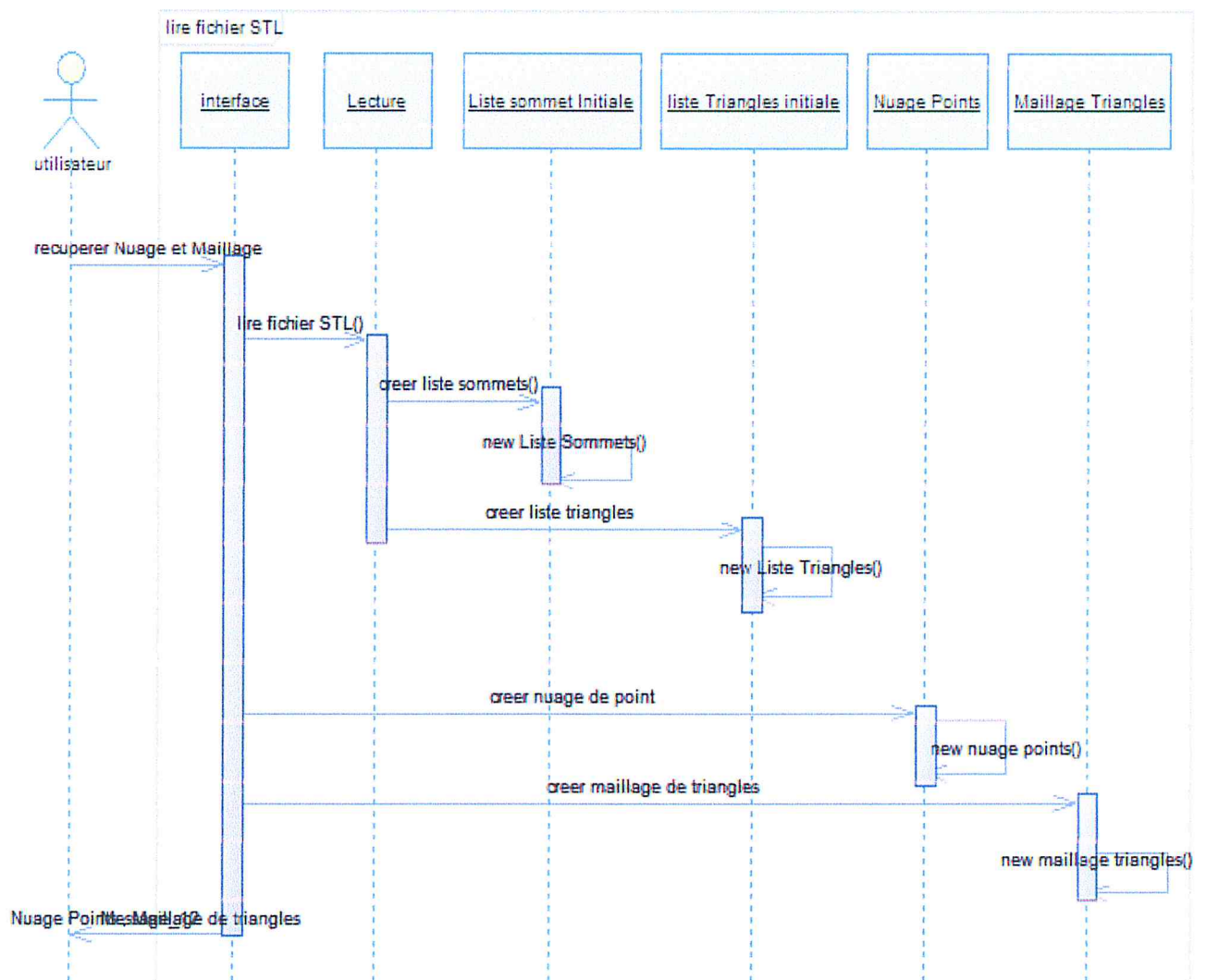


Figure 28 : Lire fichier STL.

**La création des contours passe par les étapes suivantes (Figure 29) :**

- L'utilisateur demande la création des contours par l'objet trajectoire.
- L'utilisateur donne le nombre de bandes et nombre de plans à la trajectoire.
- L'objet trajectoire demande au plan de créer le vecteur des plans.
- L'objet trajectoire demande la création de la liste de bandes par l'objet liste bande.
- L'objet liste bandes crée une instance de liste de bandes.
- Le trajectoire parcourt les plans et chaque fois récupère un plan.
- L'objet plan récupère le maillage de triangles de sa bande correspondante.
- L'objet bande envoie le maillage de triangles au plan.
- Le plan calcule les segments d'intersections.
- Le plan demande la création de liste de contours par l'objet liste contours.
- L'objet liste contours crée une instance de liste de contours.
- Le plan demande la création d'un contour.
- L'objet contour crée une instance de contour.
- Le plan va cahier les segments du contour créé.
- En fin l'objet plan demande de l'objet liste contours d'ajouter le contour créé.
- 

**La gestion des interférences et collision passe par les étapes suivantes (Figure 30) :**

- L'utilisateur envoie un message au l'objet trajectoire pour détecter interférence et collision.
- L'utilisateur envoie les angles de rotations au trajectoire.
- La trajectoire parcourt le vecteur de plans.
- La trajectoire récupère un plan.
- La trajectoire demande au plan de générer les points centre outil.
- Le plan calcule les points centres outil.
- Le plan demande de l'objet outil de vérifier l'interférence et la collision.
- L'outil demande au trajectoire de récupérer la matrice des sommets.
- La trajectoire envoie la matrice des sommets à l'outil.
- L'outil parcourt les sommets.
- L'outil récupère un sommet.
- L'outil vérifie l'interférence avec ce sommet.
- L'outil corrige l'interférence avec ce sommet.
- L'outil vérifie la collision avec ce sommet.
- L'outil corrige la collision avec ce sommet.

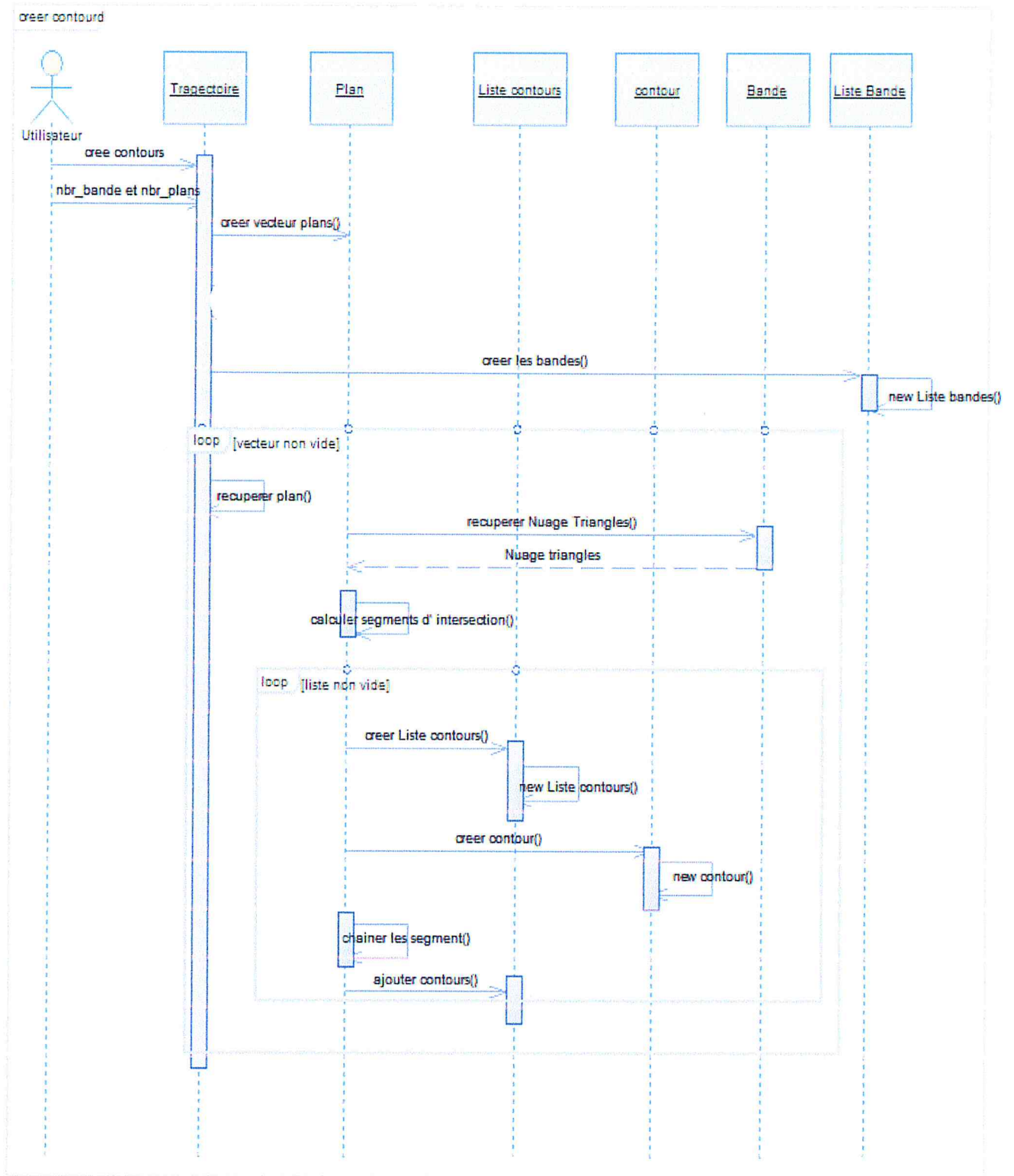


Figure 29 : Diagramme de séquence pour « créer contours ».

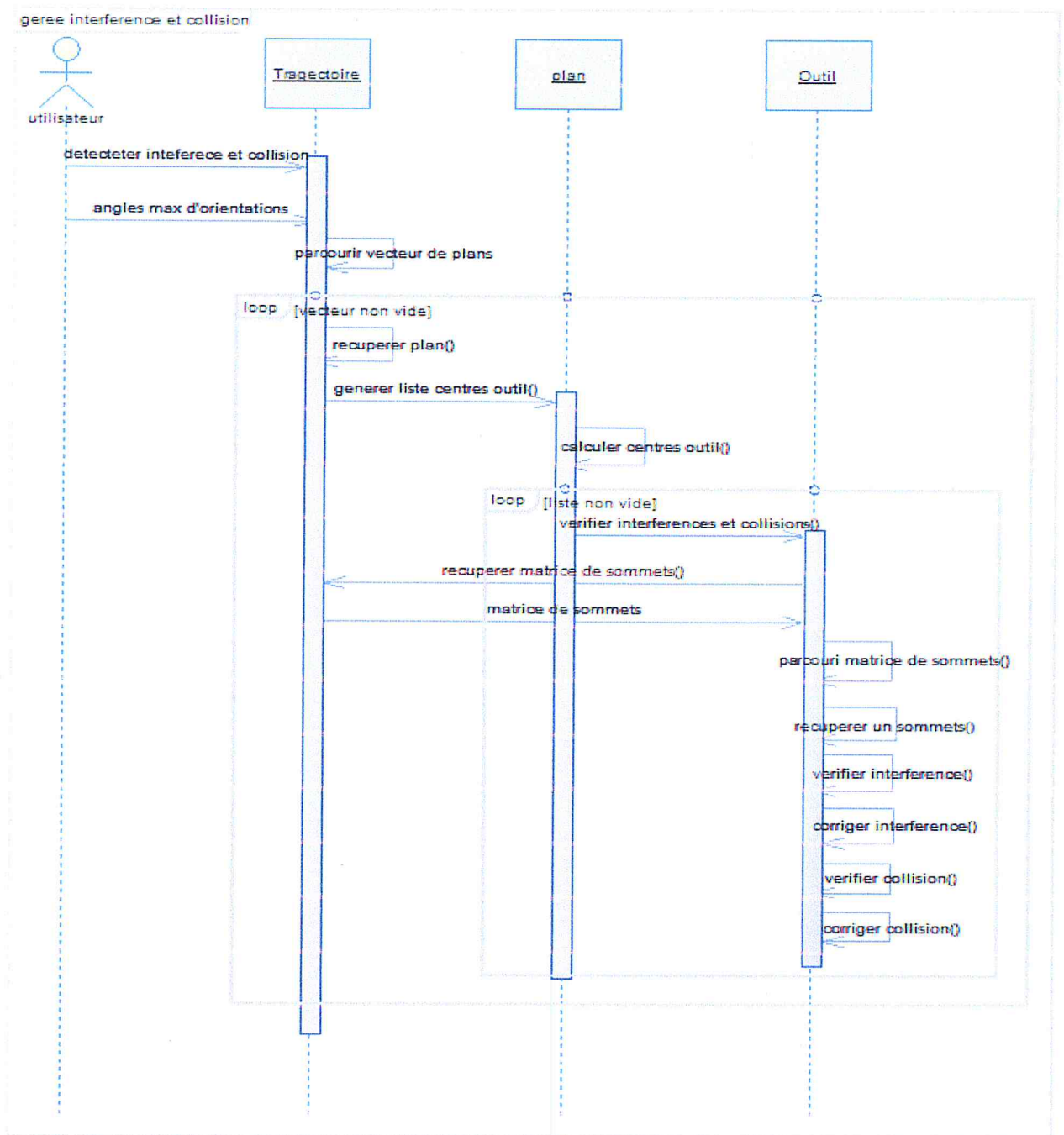


Figure 30 : Diagramme de séquence pour « gérer interférence et collision ».

### 3.5. Diagramme de collaboration :

Le diagramme de collaboration est une extension du diagramme d'objets. C'est une vue dynamique du système où il présente la réalisation des opérations ainsi que l'inscription des interactions entre les objets entre envois et réceptions de messages (Figure 26).

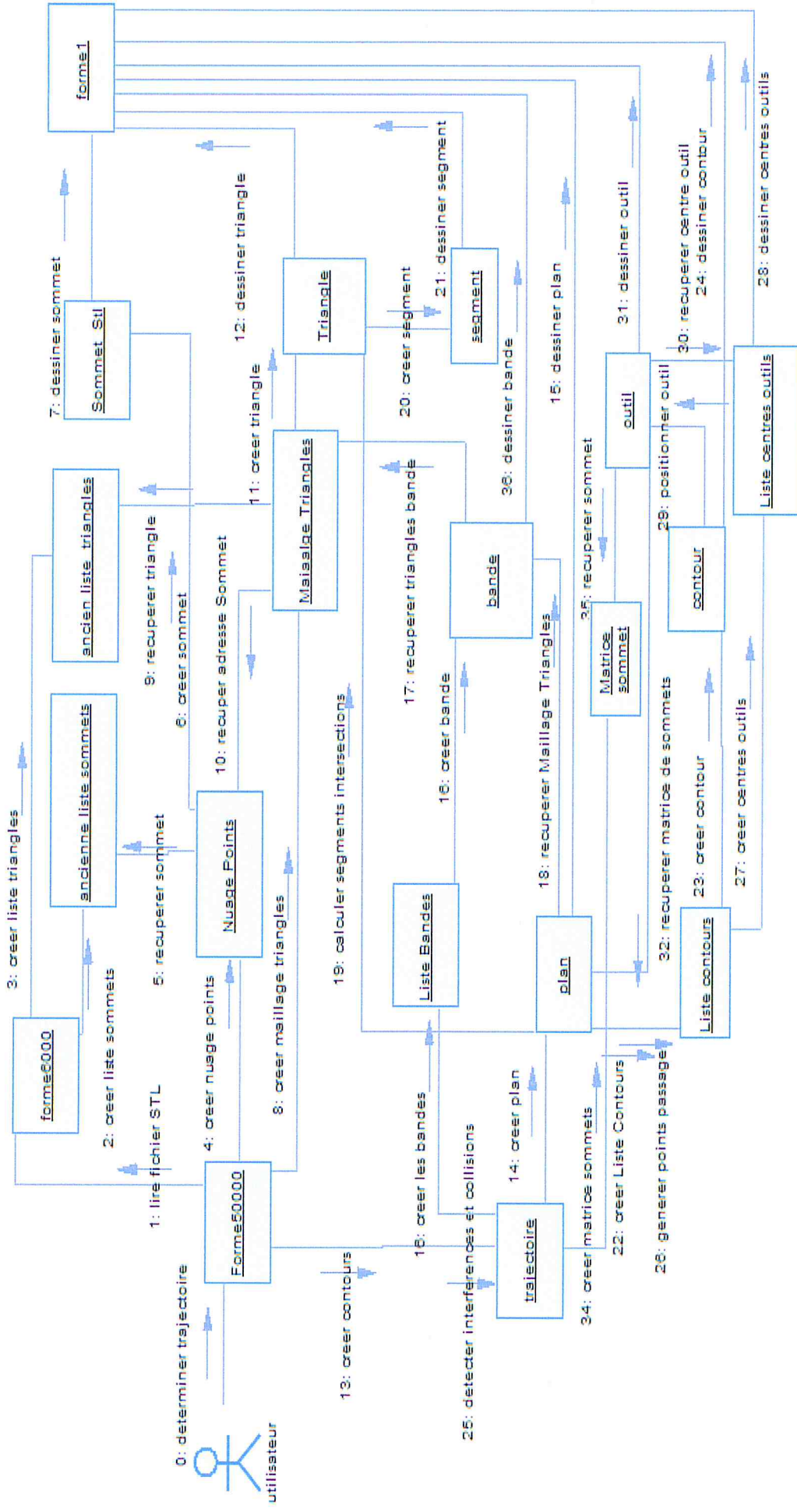


Figure 31 : Diagramme de collaboration.

**CONCLUSION :**

Dans ce chapitre, nous avons conçu notre système avec le langage de modélisation UML après avoir exprimé les besoins au début de ce chapitre, nous avons décrit les différentes fonctionnalités à l'aide des diagrammes disponibles en UM tels que diagramme de classes pour monter la vue structurelle du système, diagramme des cas d'utilisation, diagrammes d'activité, diagramme de séquence et diagramme de collaboration. Nous pouvons maintenant entamer la phase implémentation informatique des différentes fonctionnalités du système.



**CHAPITRE III**

**DEMARCHE DE**

**RESOLUTION**

**ET SOLUTIONS PROPOSEES**

## INTRODUCTION :

Après avoir exprimé les objectifs de notre projet « Finition des Surfaces Complexes en 05-Axes par la Stratégie Z-Constant » ainsi que les solutions proposées en utilisant le langage de modélisation UML (Unified Modeling Language), nous allons passer maintenant à la présentation de l'architecture générale de l'application logicielle que nous allons réaliser et toutes les informations indispensables pour l'implémentation informatique. Dans ce chapitre, nous allons détailler les problèmes mathématiques rencontrés et les complexités trouvées durant le processus de développement en essayant de proposer des réponses (solutions) aux questions posées sous forme d'organigrammes et diagrammes pour toutes les étapes de développement.

### 1. Architecture générale de l'application logicielle :

Les principales étapes de l'application logicielle, dans un ordre chronologique, à développer sont décrites par l'organigramme global donné par la Figure 1.

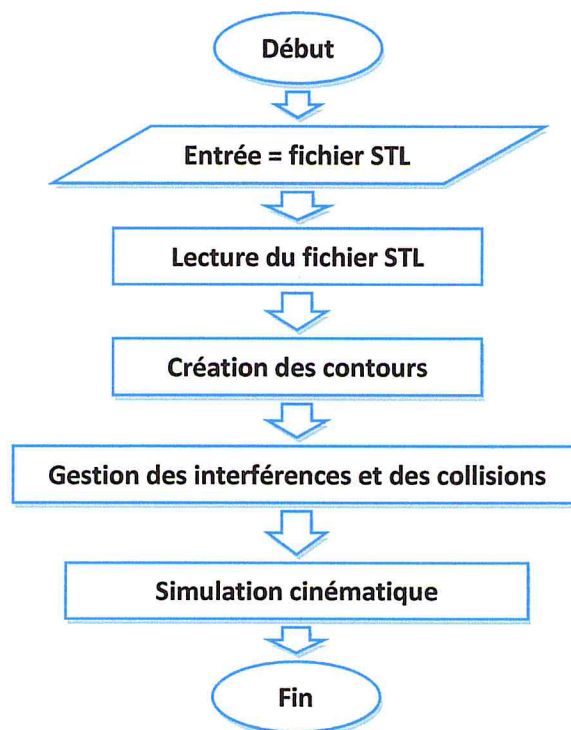


Figure 1 : Organigramme global.

Chaque grande étape comporte plusieurs sous-tâches :

1. Lecture de la forme de la pièce à usiner à partir d'un fichier STL :
  - Récupération des paramètres de la forme brute.
  - Création du vecteur des sommets.
  - Création du vecteur des triangles.

- Estimation du vecteur normale en chaque sommet.
2. Création des contours :
    - Création du vecteur de plans.
    - Création de la liste des bandes.
    - Calcul des points d'intersection entre les plans et les triangles.
    - Estimation de la normale aux points d'intersection.
    - Création de la liste des segments d'intersection dans chaque plan.
    - Chaînage des segments et création des contours.
  3. Gestion des interférences et des collisions entre l'outil et la pièce pour chaque contour :
    - Génération des points centre-outil (points de passage) pour chaque contour.
    - Positionnement de l'outil en chaque point de passage.
    - Détection des interférences de l'outil en chaque point de passage.
    - Correction des interférences.
    - Détection des collisions en chaque point de passage de l'outil.
    - Correction des collisions.
  4. Simulation cinématique des mouvements de l'outil.

## 2. Solutions proposées :

### 2.1. Lecture de la forme de la pièce à usiner à partir d'un fichier STL :

La lecture d'un fichier STL est la première étape dans notre application logicielle. Elle consiste à récupérer les informations et les paramètres définissant la forme de la pièce à usiner (Figure 2). Les informations récupérées sont un vecteur des sommets sans redondance et un vecteur des triangles. L'algorithme de lecture d'un fichier STL est déjà implémenté, alors notre objectif est la récupération de la liste des sommets et de la liste des triangles et les mettre dans nos propres structures de données. A cet effet, nous avons choisi des tableaux de pointeurs vers les sommets et vers les triangles pour économiser l'espace mémoire et pour profiter de l'accès direct et rapide aux données. Après la récupération de ces données, plusieurs autres informations sont déterminées :

- **Paramètres de la forme brute :** chaque pièce à une forme brute. Cette dernière est défini par les dimensions (longueur, largeur et hauteur) et les extrémités

minimales et maximales suivant chaque axe (X, Y, Z). Les valeurs extrêmes des coordonnées des sommets suivant les axes X, Y et Z sont  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Z_{min}$ ,  $Z_{max}$ . Tous ces paramètres ont une grande importance dans la suite.

- **Création du vecteur des sommets** : dans un premier temps, nous procédons à l'insertion de pointeurs vers les sommets de la première liste chaînée générée par l'ancien algorithme dans le nouveau vecteur des sommets.

- **Création du vecteur des triangles** : dans un premier temps, nous parcourons la liste chaînée des triangles, et pour chaque triangle nous déterminons les adresses de ses trois sommets dans le vecteur des sommets. A la fin, nous créons un nouveau triangle défini par les adresses de ses sommets et nous insérons sa référence dans le vecteur des triangles.

- **Estimation de la normale aux sommets** : par la suite, nous avons estimé la normale en chaque sommet car le format STL ne donne que les normales des triangles.

#### ❖ Comment estimer la normale aux sommets ?

Deux modes de calcul sont adoptés pour estimer la normale aux sommets :

- 1<sup>er</sup> mode de calcul : consiste à faire la moyenne des normales des triangles qui partagent le sommet. Soit S un sommet, et  $N_1, N_2, N_3, \dots, N_i, \dots, N_n$  les triangles qui partagent le sommet S. La normale  $N_s$  estimée au sommet S est donnée par :

$$N_s = \frac{\sum N_i}{\sum i}$$

Où n est le nombre de triangles partageant le sommet et i l'indice du triangle.

- 2<sup>ème</sup> mode de calcul : consiste à utiliser la moyenne pondérée par les aires des triangles qui partagent le sommet. La normale  $N_s$  estimée au sommet S est donnée par :

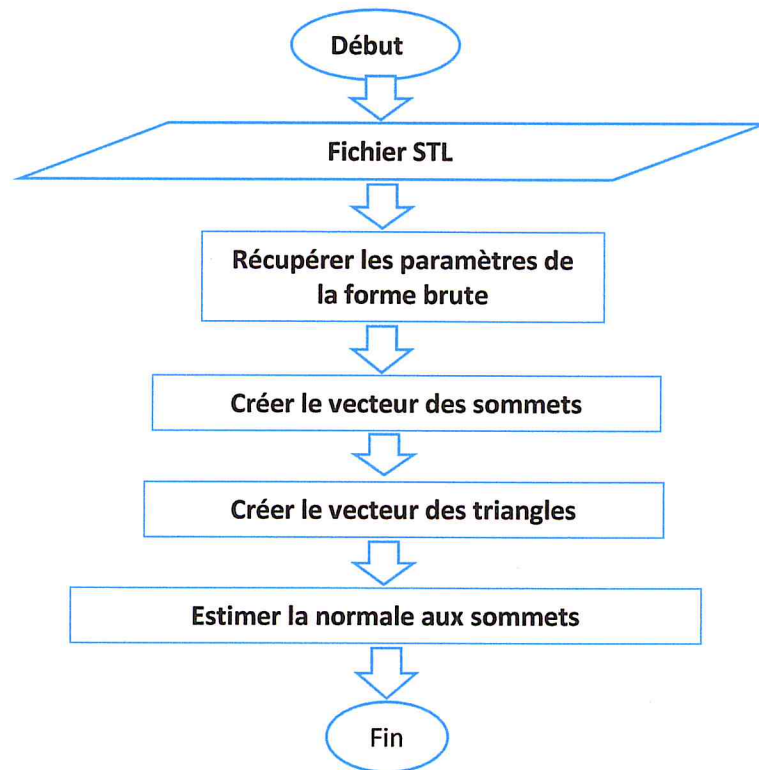
$$N_s = \frac{\sum N_i Aire_i}{\sum Aire_i}$$

Où  $Aire_i$  est l'aire du triangle d'indice i.

Pour les deux modes de calcul, le vecteur normal unitaire au sommet est calculé par :

$$\vec{n} = \vec{N}_s / \|\vec{N}_s\|$$

$$\text{Où } \|\vec{N}_s\| = \sqrt{N_x^2 + N_y^2 + N_z^2}$$



**Figure 2** : Organigramme de lecture du fichier STL.

## 2.2. Création des contours :

La création des contours est une fonctionnalité complexe. Cette étape est composée de plusieurs sous-tâches séquentielles (Figure 3). Un contour est composé par des segments d'intersection des triangles avec un seul plan où chaque deux points voisins doivent avoir un point d'intersection commun.

Pour créer les contours de chaque plan, nous parcourons le vecteur des triangles, et pour chaque triangle nous calculons ses points d'intersection avec le plan d'usinage sélectionné. Généralement, un plan d'usinage passe par un nombre limité de triangles mais pas la totalité des triangles, alors pour minimiser le temps de calcul il est préférable de regrouper les triangles en des bandes disjointes. Par la suite, pour chaque plan nous calculons l'intersection de ce plan juste avec les triangles de la bande qui englobe ce plan.

### 2.2.1. Création des plans :

Notre approche est basée sur la création de plans horizontaux selon l'axe Z pour des valeurs constantes de Z (Z-constant). La distance entre les plans est constante et elle

est égale à la profondeur de la passe d'usinage déterminée par l'utilisateur en fonction de la qualité de finition désirée (erreur de crête). Pour cela, nous créons un vecteur des plans parallèles entre eux et perpendiculaires à l'axe Z. Le premier plan passe par la face supérieure de la forme ( $z=Z_{max}$ ) et dernier plan passe par la face inférieure de la forme ( $z=Z_{min}$ ) (Figure 4).

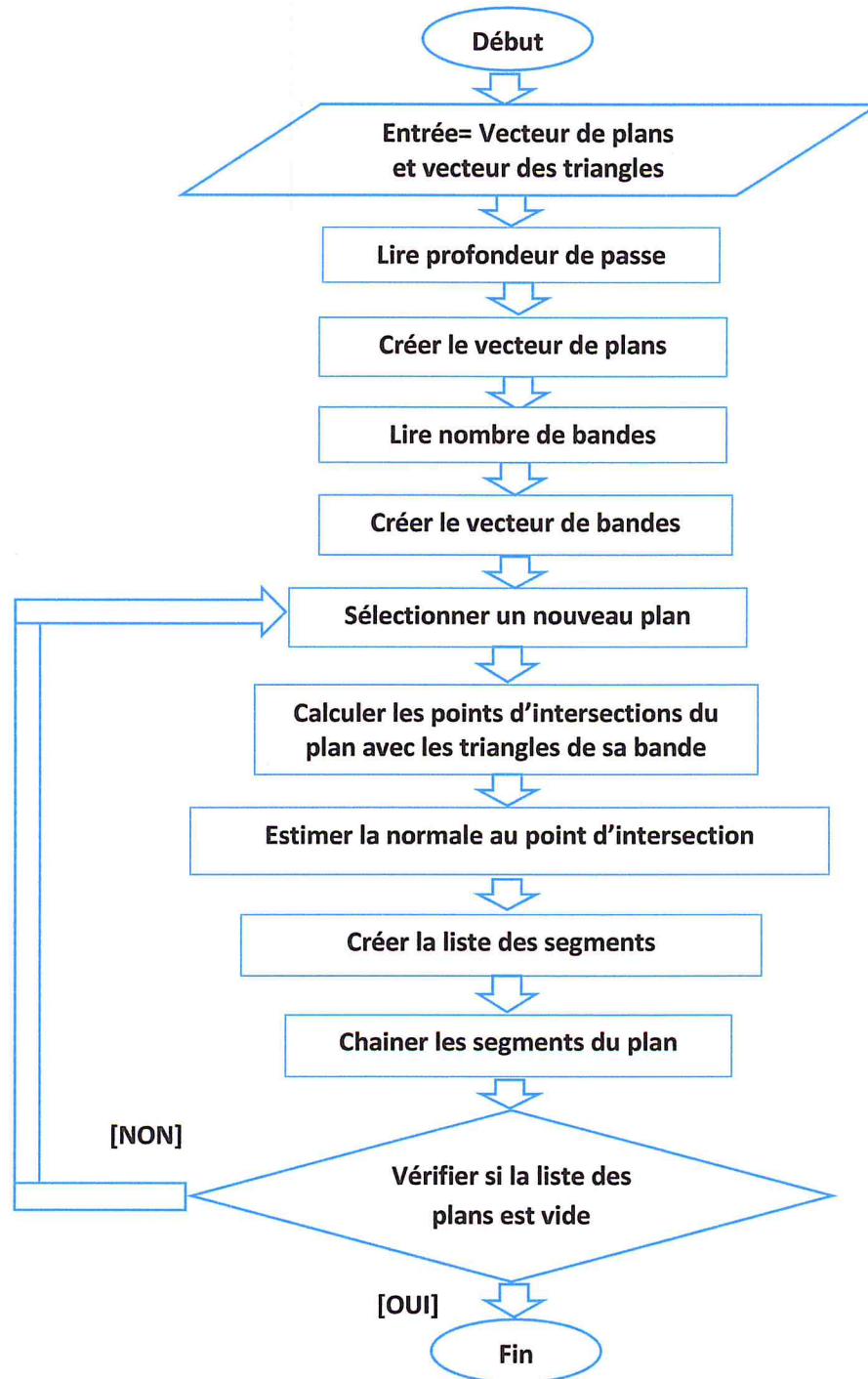


Figure 3 : Organigramme de création des contours.

### 2.2.2. Création du vecteur de bandes :

Pour créer le vecteur de bandes de triangles (Figure 5), nous avons besoin d'une largeur de bande  $L$  qui est calculée par les deux limites de la bande  $Z_{\max\_Bande}$  et  $Z_{\min\_Bande}$ . La largeur  $L$  est calculée automatiquement à partir du nombre de bandes introduit par l'utilisateur par :

$$L = \text{Hauteur de forme} / \text{nombre de bandes}$$

- Une bande ne contient que les triangles compris entre ses limites c'est-à-dire ceux qui se trouvent dans la bande et n'ont pas d'intersection avec les limites supérieure et inférieure de la bande. D'une autre façon, les trois sommets du triangle se trouvent dans la bande.
- Pour tester si un triangle appartient à une bande, il suffit de tester les valeurs des coordonnées  $z$  de ses trois sommets. Le triangle appartient à une bande si :

$$Z_{\max\_Bande} < z < Z_{\min\_Bande}.$$

- Les triangles ayant des sommets dans plus d'une bande sont insérés dans une bande supplémentaire.

Une fois le vecteur de bandes est créé, nous passons au calcul des intersections.

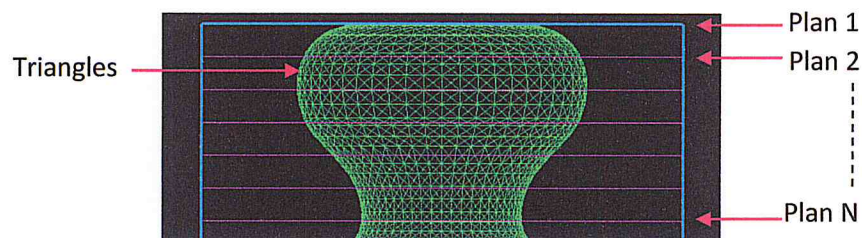


Figure 4: Plans horizontaux.

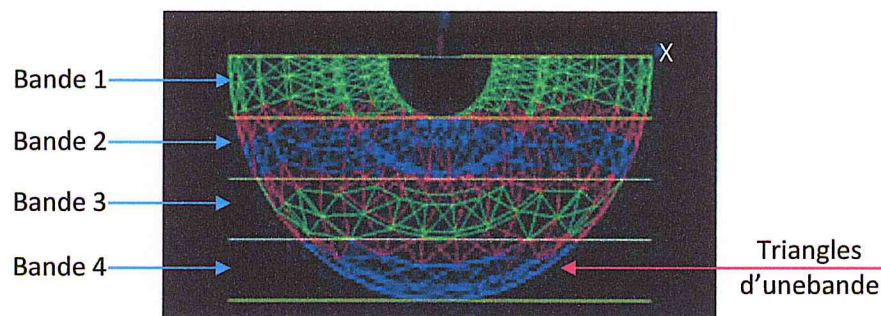
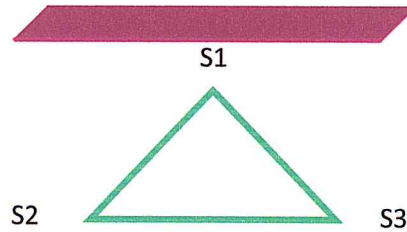


Figure 5 : Bandes des triangles.

### 2.2.3. Calcul des points d'intersection :

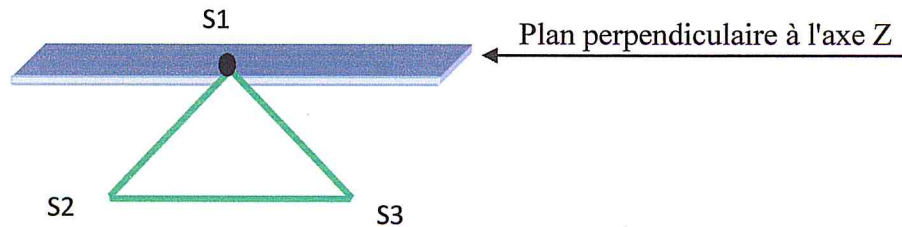
Pour calculer les points d'intersection entre un plan et un triangle, plusieurs cas sont à considérer :

- **1<sup>er</sup> cas** : c'est le cas où le plan ne passe pas par le triangle. Donc, pas d'intersection.



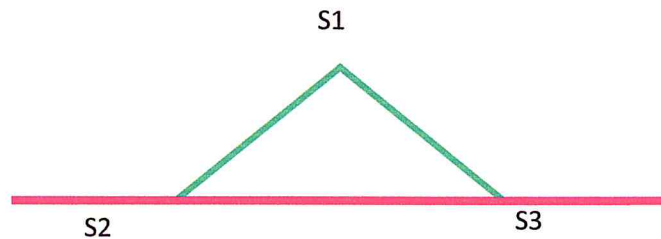
**Figure 6** : Pas d'intersection entre le plan et le triangle.

- **2<sup>ème</sup> cas** : c'est le cas où l'intersection du plan avec un triangle est un seul sommet. Cette situation n'est pas considérée puisqu'elle ne génère pas un segment mais un point.



**Figure 7** : Le plan passe par un seul sommet de triangle.

- **3<sup>ème</sup> cas** : une arête du triangle appartient au plan, c'est-à-dire deux sommets du triangle appartiennent au plan.



**Figure 8** : Plan en intersection avec une seule arête de triangle.

- **4<sup>ème</sup> cas** : le plan passe par deux arêtes. Donc, un seul segment est généré. Il est formé par les deux points d'intersection avec les deux arêtes.



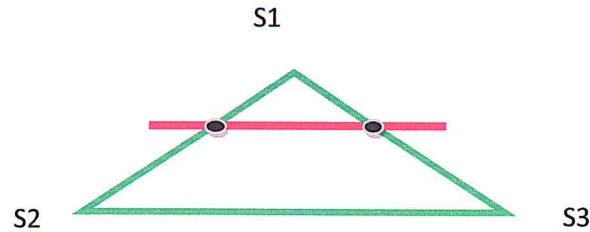


Figure 9 : Plan en intersection avec le plan en deux sommets.

- **5<sup>ème</sup> cas :** le plan passe par un sommet et une arête. Le résultat est un segment formé par le sommet en intersection et le point d'intersection du plan avec l'arête.

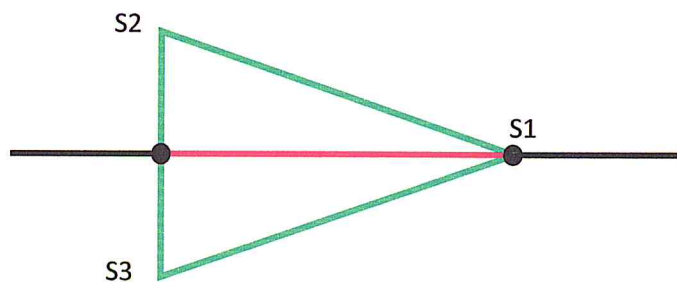


Figure 10 : Le plan passe par un seul sommet et une arête.

- **6<sup>ème</sup> cas :** le triangle se trouve dans le plan. Dans ce cas, il y a une infinité de points d'intersection. Pour notre cas, nous gardons juste les trois sommets du triangle qui forment trois segments.

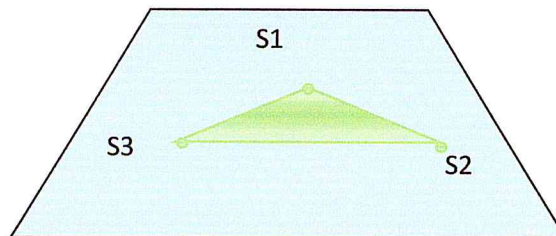


Figure 11 : Le triangle appartient au plan.

#### ❖ Comment calculer les coordonnées des points d'intersection ?

Nous distinguons deux cas possibles. Le premier cas où le plan passe par un sommet et le deuxième cas où le plan passe par une arête en un point différents des sommets des triangles.

- Pour vérifier si un sommet appartient à un plan, il suffit juste de vérifier la coordonnée  $z$  du sommet avec la coordonnée  $Z$  du plan. Si elles sont égales, alors le sommet appartient au plan.
- Le deuxième cas est un peu compliqué. Nous devons d'abord calculer l'équation de la droite passant par l'arête à partir de ses deux

sommets. Par la suite, nous calculons l'intersection d'une droite avec un plan pour déterminer les coordonnées  $x$ ,  $y$  et  $z$  du point d'intersection (Annexe).

#### 2.2.4. Estimation de la normale au point d'intersection :

Si le point d'intersection est un sommet, alors sa normale est déjà estimée dans la première étape (lecture de fichier STL).

Dans le cas où le point est un nouveau point calculé (intersection d'une arête avec un plan), alors la normale est calculée en utilisant la pondération par les distances par rapport aux sommets qui forment l'arête.

#### 2.2.5. Génération des segments :

En réalité, la génération des segments est réalisée au cours de calcul des points d'intersection où un segment est formé par deux points d'intersection entre un plan et un triangle.

#### 2.2.6. Chaînage des segments :

Après génération de la liste globale des segments d'intersection d'un plan, nous devons chaîner les segments voisins (segments ayant un point commun) pour former un contour. Nous devons noter qu'un plan peut contenir plusieurs contours différents et disjoints. Le processus de chaînage des segments est montré par la Figure 12.

La séquence des sous-tâches est détaillée dans les paragraphes suivants :

- Initialiser la liste des segments chaînés d'un contour par un segment retiré de la liste globale des segments non chaînés.
- Parcourir la liste globale et chercher le segment qui suit le dernier segment inséré dans le contour. Pour vérifier si un segment est le suivant, nous devons vérifier si le point de fin du premier segment est le même point de début du deuxième segment. Donc, il suffit de vérifier l'égalité des trois coordonnées  $x$ ,  $y$  et  $z$  des deux points pour dire que les deux segments sont connectés. Une fois le segment suivant déterminé, il est inséré après le premier segment dans le contour (dans la liste des segments) et il est supprimé de la liste globale des segments pour éviter une boucle infinie. Cette procédure est répétée pour chaque dernier segment suivant trouvé et inséré dans le contour jusqu'à ce que le contour devienne fermé.
- Lorsque nous terminons la recherche des segments suivants, nous prenons le premier segment inséré dans le contour et nous cherchons les précédents de ce segment et nous l'insérons avant ce segment avec la même logique que pour les segments suivants mais avec une logique inversée, c'est-à-dire nous

vérifions la coïncidence du point de début du segment sélectionné avec le point de fin de son précédent.

Quand nous déterminons tous les suivants et tous les précédents, nous arrivons à former un contour. Ce contour est inséré dans la liste des contours du plan sélectionné et nous cherchons le prochain contour avec les étapes précédentes jusqu'à la fin de la liste des segments.

### **2.3. Gestion des interférences et des collisions :**

Pour usiner un segment, nous devons positionner l'outil verticalement sur le point de début du segment à usiner et ensuite procéder à la détection des interférences (partie active de l'outil : sphère) et des collisions de l'outil (corps de l'outil : partie cylindrique) avec la forme à usiner. Le processus de détection des interférences et des collisions est donné par la Figure 13.

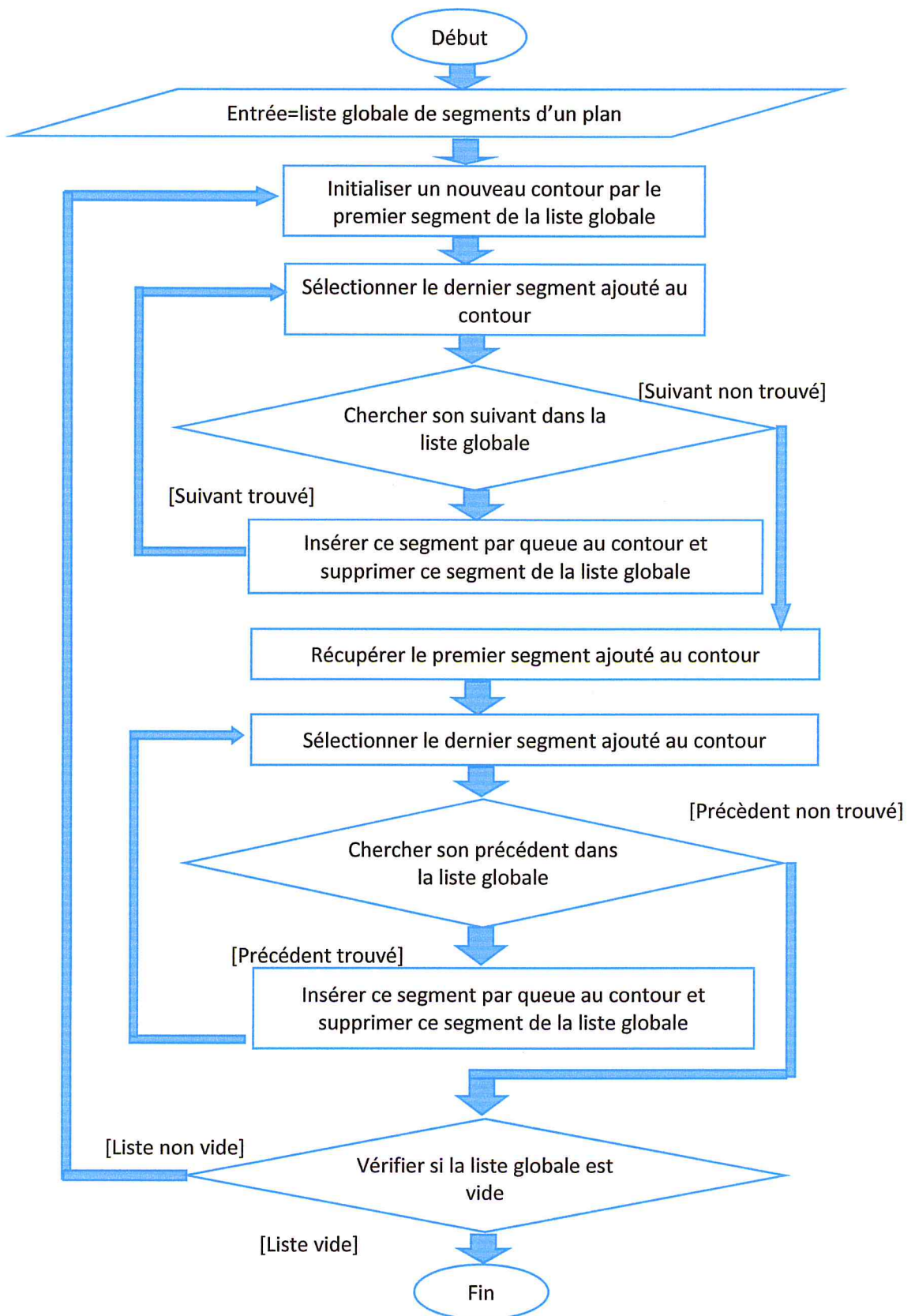


Figure 12 : Organigramme de chaînage des segments

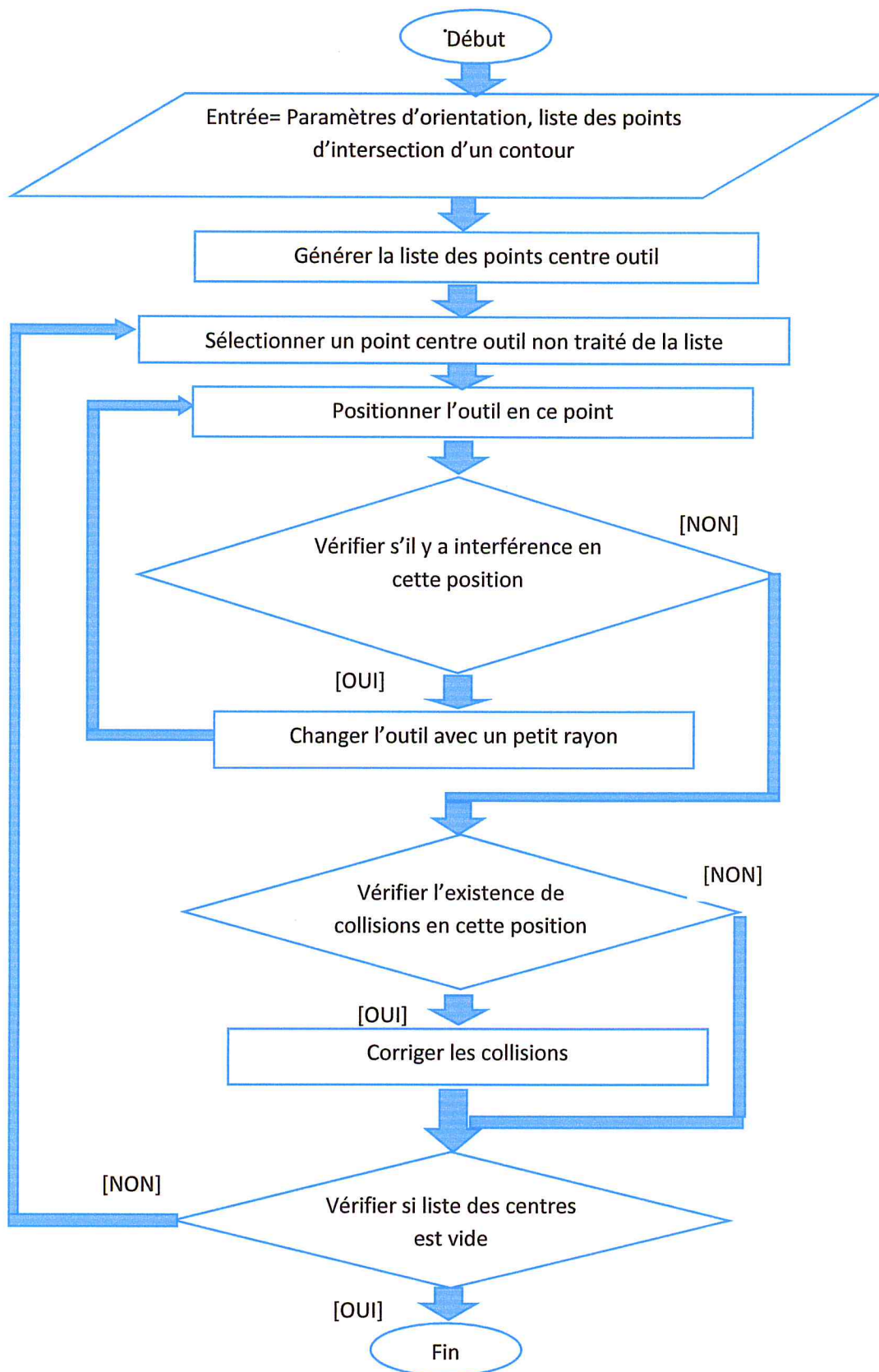


Figure 13 : Gestion des interférences et des collisions.

### 2.3.1. Paramètres d'usinage :

Chaque fraiseuse a des paramètres spécifiques déterminés par le constructeur. Ces paramètres concernent l'outil où chaque outil a son propre rayon de sphère interchangeable et aussi les angles maximums d'orientation :

- Angles alpha : vers le segment (dans le sens d'usinage).
- Angle beta : autour du segment (vers l'extérieur de la pièce).

Donc, avant de traiter les interférences et les collisions, nous devons connaître la liste des outils disponibles (rayons et longueur) et les angles maximums d'orientations puisque l'utilisateur doit saisir ces paramètres au début de cette étape.

### 2.3.2. Création de liste des points de passage de l'outil :

Pour positionner l'outil, nous avons besoin de connaître les coordonnées du centre de la sphère (partie active de l'outil) pour déduire les coordonnées de ses extrémités et de son enveloppe. Les coordonnées du centre sont calculées à partir du point d'intersection (point de contact outil-surface et de la normale en ce point).

#### ❖ Comment calculer les coordonnées du point centre ?

Les valeurs des coordonnées du centre outil sont calculées comme suit :

Soit un outil de rayon  $r$  (rayon de sa sphère) et soit  $P$  le point d'intersection (point de début du segment) de coordonnées  $X_p, Y_p$  et  $Z_p$ . Au point  $P$ , les composantes du vecteur normale unitaire au point  $P$  sont  $U_x, U_y$  et  $U_z$  (Figure 14). Les coordonnées  $X_c, Y_c$  et  $Z_c$  du point  $C$  centre d'outil sont données par :

$$X_c = X_p + U_x \cdot r$$

$$Y_c = Y_p + U_y \cdot r$$

$$Z_c = Z_p + U_z \cdot r$$

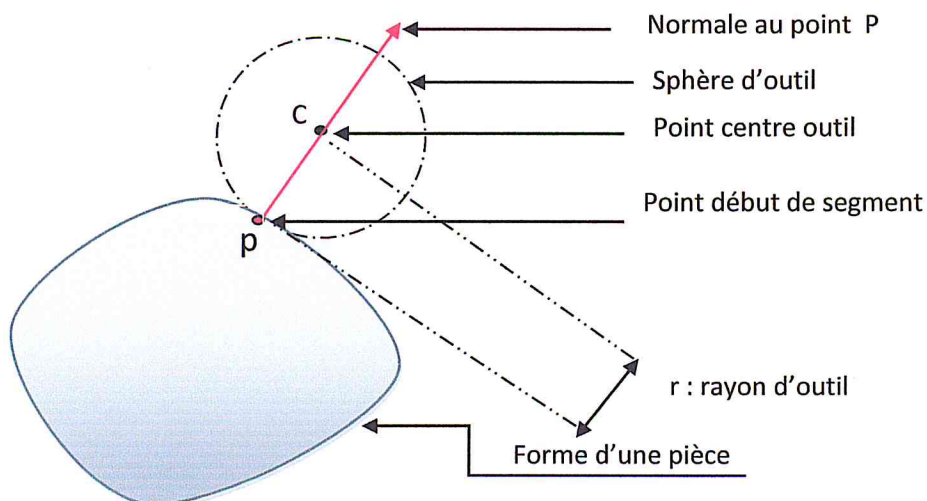


Figure 14 : Point centre outil.

### 2.3.3. Positionnement de l'outil :

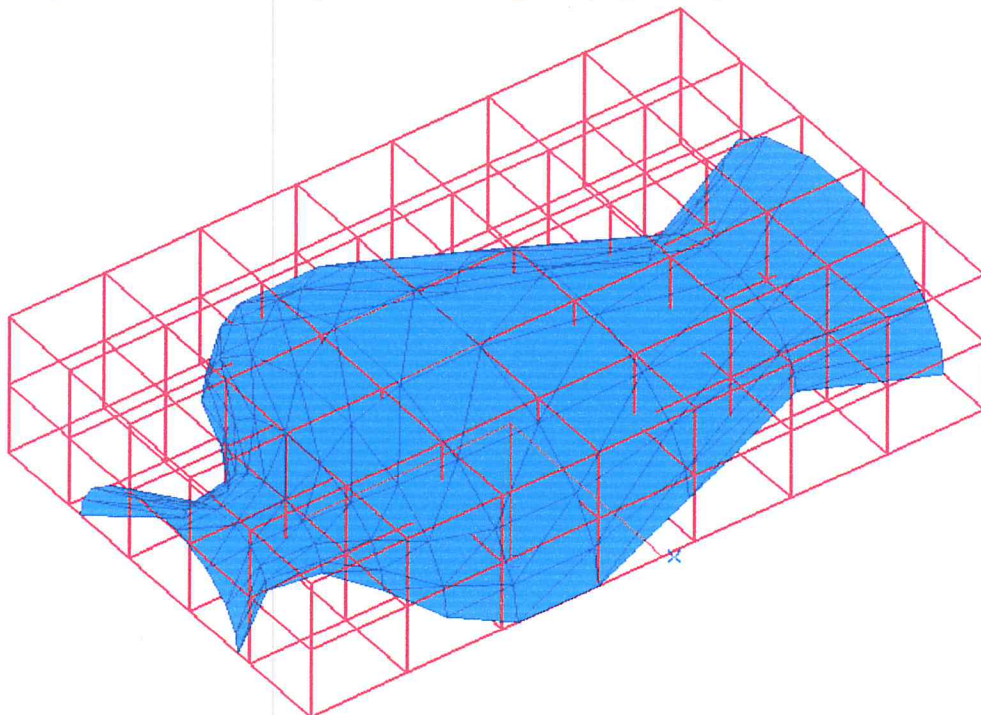
Après le calcul des centres outil, nous pouvons maintenant positionner l'outil pour usiner le segment en question. Initialement, nous positionnons l'outil verticalement au plan, c'est-à-dire son axe est parallèle à l'axe Z.

### 2.3.4. Détection des interférences :

Nous disons que l'outil est en interférence avec la pièce si l'intersection du volume de son sphère (partie active) avec le volume de la pièce est différent de nulle.

#### ❖ Comment détecter les interférences?

Afin de vérifier l'intersection des volumes, il suffit de vérifier l'appartenance des sommets de la forme au volume de la sphère de l'outil. La vérification de l'intersection de la sphère avec tous les sommets est très coûteuse en termes de temps de calcul. Alors, pour minimiser ce temps, il est indispensable de réduire le nombre de sommets à vérifier. D'où, la nécessité d'une nouvelle représentation des sommets. Il est idéal de représenter les sommets par une matrice de trois dimensions de cellules homogènes de dimensions pas forcément égales (Figure 15).



**Figure 15** :Division de la forme brute en cellules homogènes.

- **Création de la matrice de sommets :**

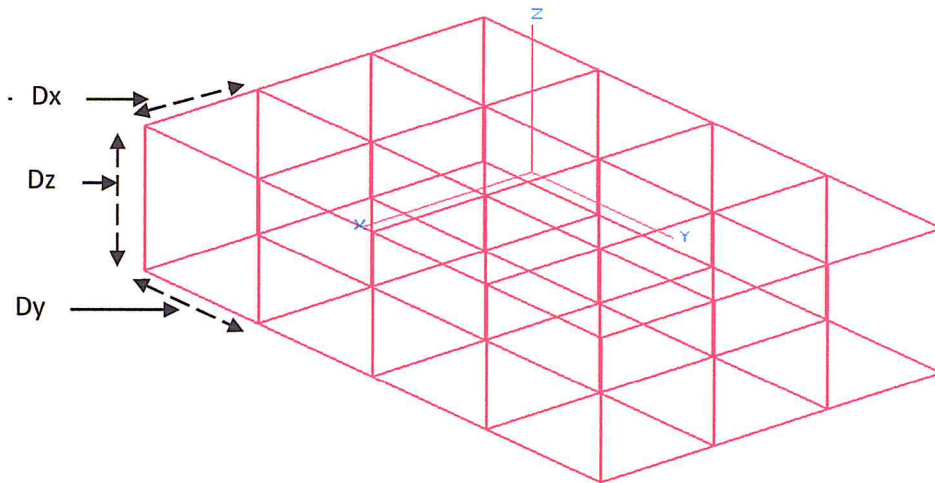
Chaque cellule est définie par ses dimensions et ses extrémités maximales et minimales suivant les trois axes X, Y et Z. Les dimensions (Dx, Dy, Dz) de la cellule sont calculées automatiquement à partir du rayon de l'outil comme suit :

- ❖ **Comment calculer les dimensions de la cellule?**

La dimension Dx de la cellule suivant l'axe X est calculée comme suit (Figure 16) :

$$Dx = (X_{max} - X_{min}) / r$$

Où Xmax et Xmin sont respectivement la coordonnée maximale et la coordonnée minimale de la forme brute suivant l'axe X et r est le rayon de l'outil. Les dimensions Dy et Dz sont calculées de la même manière suivant les axes Y et Z.



**Figure 16:** Dimensions d'une cellule.

- **Affectation des sommets aux cellules correspondantes :**

Après la création de la matrice des cellules, nous devons affecter chaque sommet à la cellule correspondante (Figure 17).

Soit  $S(X_s, Y_s, Z_s)$  un sommet et  $C$  une cellule avec ses limites ( $X_{min}, X_{max}$ ), ( $Y_{min}, Y_{max}$ ) et ( $Z_{min}, Z_{max}$ ) suivant les axes X, Y et Z respectivement.

S appartient à la cellule C si :

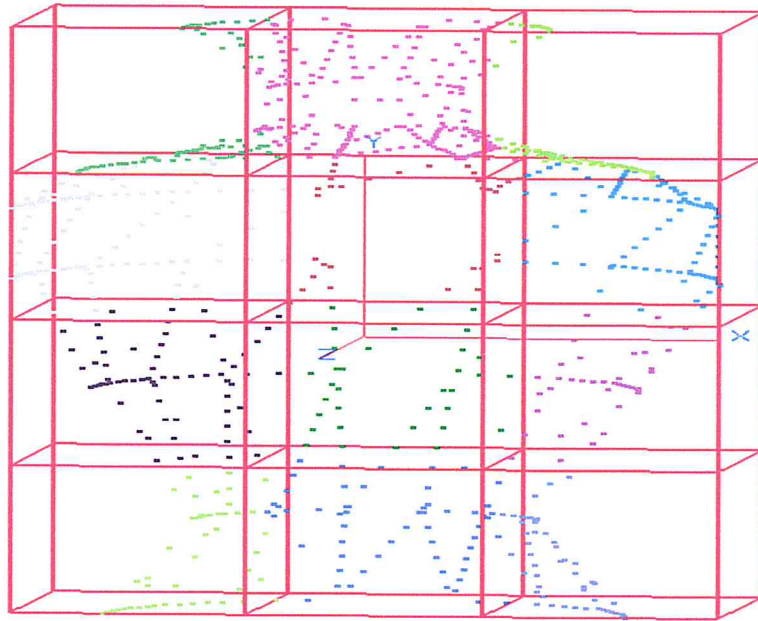
$$X_{min} \leq X_s < X_{max}$$

$$Y_{min} \leq Y_s < Y_{max}$$

$$Z_{min} \leq Z_s < Z_{max}$$

Après la nouvelle représentation des sommets et pour minimiser le nombre des sommets à vérifier dans la détection des interférences, nous allons considérer uniquement les cellules au voisinage de sphère de l'outil.





**Figure 17:** Affectation des sommets aux cellules de la matrice.

❖ **Comment chercher les cellules au voisinage de la sphère ?**

Pour résoudre ce problème, nous allons positionner virtuellement l'enveloppe de la sphère dans la matrice des sommets et par la suite nous allons déterminer les cellules qui se chevauchent avec l'enveloppe de la sphère (Figure 18). Donc, nous allons identifier les indices  $(i,j,k)$  de chaque cellule déterminée.

❖ **Comment calculer l'enveloppe de la sphère (cube) ?**

Les coordonnées des points extrêmes (coin gauche bas et coin droit haut) sont calculées comme suit :

Extrémité minimale (coin gauche bas) :

$$X_{\min} = X_c - r.$$

$$Y_{\min} = Y_c - r.$$

$$Z_{\min} = Z_c - r.$$

Extrémité maximale (coin droit haut) :

$$X_{\max} = X_c + r.$$

$$Y_{\max} = Y_c + r.$$

$$Z_{\max} = Z_c + r.$$

Où  $r$  est le rayon de la sphère  $X_c$ ,  $Y_c$  et  $Z_c$  sont les coordonnées du centre outil.

Pour déterminer les cellules dans une matrice tridimensionnelle, nous devons connaître suivant chaque direction à quelle position (indice) nous commençons la sélection et à quelle position (indice) nous devons terminer la sélection. Par conséquent, six indices doivent être déterminés (Figure 19).

- $I_{min}$  et  $I_{max}$  pour la première dimension.
- $J_{min}$  et  $J_{max}$  pour la deuxième dimension.
- $K_{min}$  et  $K_{max}$  pour la troisième dimension.

Après le calcul des deux extrémités de l'enveloppe de la sphère, nous pouvons facilement extraire les valeurs des indices  $i$ ,  $j$  et  $k$  comme suit :

$$I_{min} = X_{min} / D_x$$

$$I_{max} = X_{max} / D_x$$

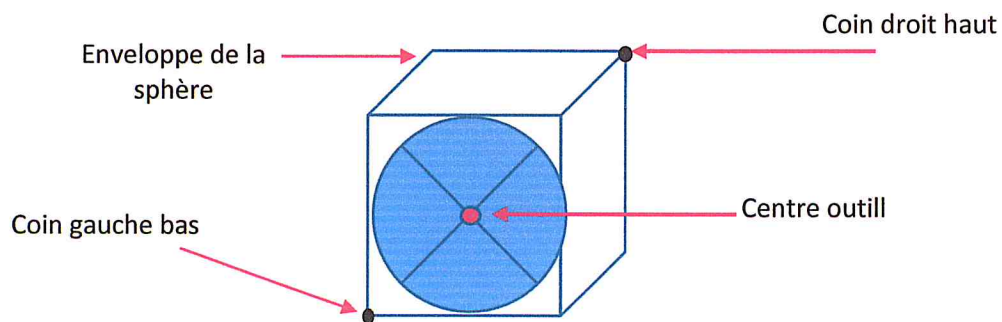
$$J_{min} = Y_{min} / D_y$$

$$J_{max} = Y_{max} / D_y$$

$$K_{min} = Z_{min} / D_z$$

$$K_{max} = Z_{max} / D_z$$

A la fin de cette étape, nous avons déterminé les cellules concernées par la vérification des interférences. Il nous reste que de récupérer les sommets de ces cellules et de vérifier l'appartenance de ces sommets à l'enveloppe de la sphère.



**Figure 18:** Enveloppe de la sphère de l'outil.

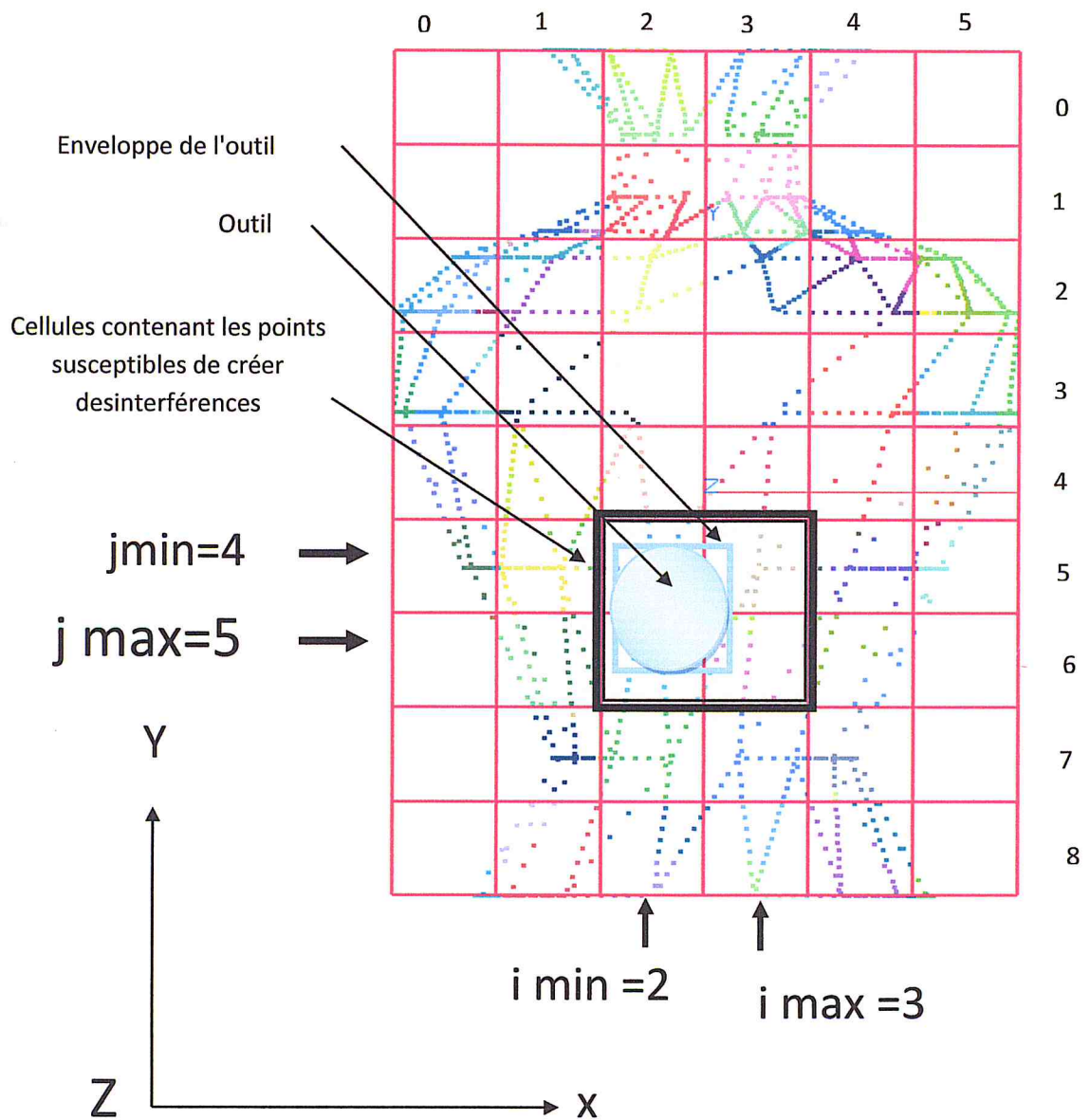
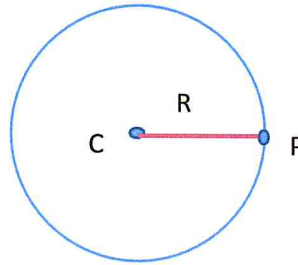


Figure 19 : Cellules candidates à l'interférence.

❖ Comment vérifier si un sommet est en interférence avec la sphère ?

Pour résoudre ce problème, nous utilisons l'équation de la sphère. Soit  $C(X_c, Y_c, Z_c)$  un point qui représente le centre de la sphère et  $R$  le rayon de cette sphère (Figure 20). Un point  $P(X_p, Y_p, Z_p)$  appartient à la surface de la sphère si l'équation suivante est vérifiée :

$$(Xp - Xc)^2 + (Yp - Yc)^2 + (Zp - Zc)^2 = R^2$$



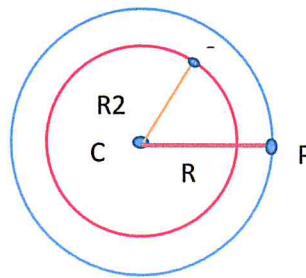
**Figure 20:** Surface de la sphère.

Soit un point  $S(Xs, Ys, Zs)$  appartenant à une deuxième sphère de rayon  $R2$ . Alors, les coordonnées du point  $S$  vérifient l'équation suivante :

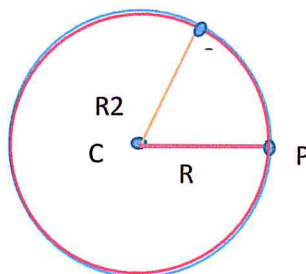
$$(Xs - Xc)^2 + (Ys - Yc)^2 + (Zs - Zc)^2 = (R2)^2$$

Donc :

- si  $R2 < R$ , alors la deuxième sphère se trouve à l'intérieur de la première sphère ce qui implique l'existence d'interférence.
- Si  $R2 = R$ , alors les deux sphères sont identiques ce qui implique absence d'interférence.
- Si  $R2 > R$ , alors la première sphère se trouve à l'intérieur de la deuxième sphère. Dans ce cas, il n'y a pas d'interférence.



**Figure 21 :** Sommet  $S$  est à l'intérieur de la sphère de l'outil.



**Figure 22:** Sommet  $S$  est sur la surface de la sphère de l'outil.

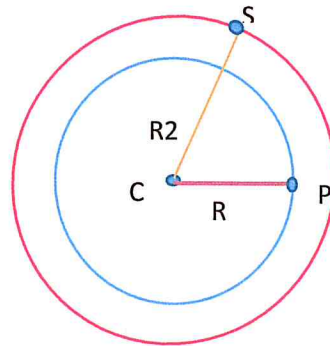


Figure 23 : Sommet S est à l'extérieur de la sphère de l'outil.

- **Changement d'outil :**

En cas d'interférences, nous devons corriger cette interférence par le changement d'outil. Généralement, le magasin d'un centre d'usinage permet la mise en place de plusieurs outils de rayons différents. Au début, nous choisissons le plus grand outil pour avoir des vitesses d'usinage plus importantes et par conséquent des temps d'usinage très faibles. En cas d'interférences, nous diminuons le rayon de l'outil. Si la liste des outils se termine avant la correction totale des interférences, alors nous sommes obligés d'éliminer ce segment de l'usinage de l'opération d'usinage pour ne pas endommager l'outil et la pièce.

### 2.3.5. Détection des collisions :

Après le positionnement de l'outil et la correction des interférences, nous devons vérifier l'existence de collisions entre l'outil et la pièce. Le principe est le même qu'avec la détection des interférences, c'est-à-dire le chevauchement entre l'enveloppe du cylindre de l'outil (partie non active) et le volume de la pièce. Dans ce cas précis, nous devons chercher une méthode pour calculer l'intersection.

- ❖ **Comment détecter la collision ?**

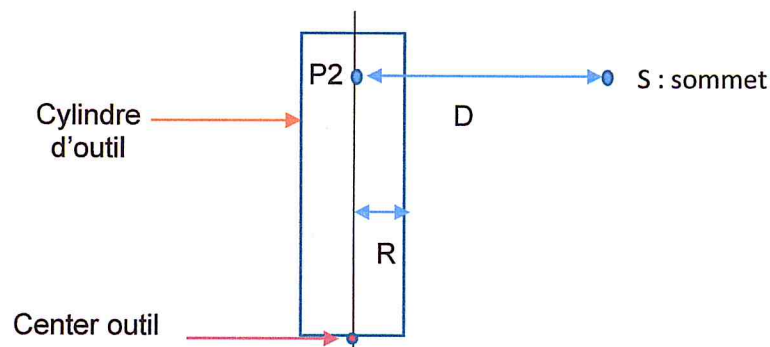
Maintenant l'outil est déjà positionné. Comme nous avons vu précédemment dans la détection des interférences, nous devons déterminer les cellules de la matrices susceptibles de provoquer des collisions de la même manière, c'est-à-dire nous calculons l'enveloppe du cylindre, plus exactement les deux extrémités maximale et minimale et par la suite nous déterminons les indices  $i, j$  et  $k$  des cellules.

Il reste donc la vérification de l'appartenance des sommets des cellules déterminées à l'enveloppe du cylindre. Dans ce cas, le test est différent de celui de l'interférence.

❖ **Comment vérifier si un sommet est en interférence avec le cylindre ?**

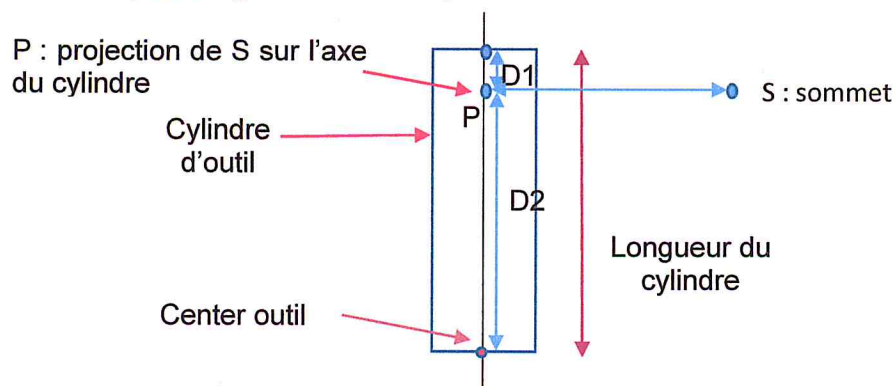
Le test se fait en deux étapes :

- **Première étape :** calculer la distance  $D$  entre le sommet  $S$  et la droite portant l'axe du cylindre, puis la comparer avec le rayon  $R$  du cylindre (rayon de l'outil). Deux cas sont considérés :
  - Si  $D \leq R$ , alors le sommet est candidat à être dans l'enveloppe du cylindre (collision avec l'outil).
  - Si  $D > R$ , alors le sommet n'est pas en collision (Figure 24).



**Figure 24 :** Projection d'un sommet sur la droite portant l'axe du cylindre.

- **Deuxième étape :** si le sommet  $S$  est candidat à une collision, pour terminer le test de vérification des collisions, nous projetons ce sommet sur la droite portant l'axe du cylindre et par la suite nous calculons la distance  $D1$  entre l'extrémité supérieure et le point résultant de la projection de  $S$  et la distance  $D2$  entre l'extrémité inférieure et le point (Figure 25). Si  $L$  est la longueur du cylindre, deux cas sont à considérer :
  - Si  $D1 + D2 = L$ , alors le sommet  $S$  appartient au volume du cylindre et donc une collision avec l'outil.
  - Si  $D1 + D2 > L$ , alors le sommet est en dehors de l'enveloppe du cylindre (pas de collisions).



**Figure 25 :** Point en collision avec le cylindre de l'outil.

La détection des collisions est effectuée en chaque position d'outil (point centre outil). Après cette étape, les angles d'orientation de l'outil pour toutes les positions sont égaux à zéro (pas d'orientations). Les positions d'outil qui provoquent une collision doivent être changées en lançant une procédure de correction des collisions.

- **Correction des collisions :**

Avant de lancer l'opération d'usinage réel, il est indispensable de corriger les collisions en chaque point centre d'outil. Si la correction est impossible, alors nous devons éliminer le segment correspondant au point centre outil de l'opération d'usinage.

- ❖ **Comment corriger les collisions ?**

La collision est due à un mauvais positionnement de l'axe de l'outil. Pour éviter la collision, nous pouvons incliner l'outil avec un angle déterminé afin d'avoir une nouvelle position qui ne provoque pas une collision. Cet angle doit être sauvegardé pour chaque opération de correction.

Pour incliner l'outil et éliminer la collision, nous pouvons orienter l'outil suivant deux directions possibles :

- Orientation vers le segment et dans le sens d'usinage.
- Orientation autour du segment et dans le sens de la normale du point d'intersection générant le point centre outil (vers l'extérieur de la pièce).

- ❖ **Comment orienter l'outil ?**

L'orientation de l'outil a pour objectif de calculer les angles d'inclinaison permettant de positionner l'outil sans collision. L'orientation de l'outil peut être réduite à l'orientation de l'extrémité supérieure de l'outil. Donc, l'orientation ne concerne pas le point centre outil qui constitue l'extrémité inférieure de l'outil (Figure 26). Après la rotation de ce point avec un angle  $\alpha$  et calcul de ses coordonnées nous pouvons positionner l'outil dans cette nouvelle position et par la suite application du test de vérification de collision. S'il n'y a pas de collision, nous sauvegardons l'angle  $\alpha$  et nous passons à un nouveau point centre outil. Si la collision existe toujours, nous incrémentons l'angle  $\alpha$  avec un pas et nous procédons à l'orientation et au test de collision ainsi de suite jusqu'à l'obtention d'un angle adéquat tout en respectant les angles d'orientation maximales de l'outil définis par le constructeur. Si la collision n'est pas éliminée, nous devons éviter l'usinage du segment correspondant.

Dans notre approche, la correction est faite en trois étapes séquentielles en fonction de la situation rencontrée :

- **Première étape :** l'orientation de l'outil est faite uniquement dans le sens d'usinage (vers le segment) avec un angle alpha incrémenté par un pas pas\_alpha. Si nous parcourons tout l'intervalle de variation de alpha sans résoudre le problème, alors nous passons à la deuxième étape.
- **Deuxième étape :** nous repositionnons l'outil dans la position initiale (position verticale) au point centre outil et nous procédons à orienter l'outil uniquement autour du segment (vers l'extérieur de la pièce) avec un angle beta incrémenté chaque fois avec un pas pas\_beta. Si nous parcourons tout l'intervalle de variation de beta sans résoudre le problème, alors nous passons à la troisième étape.
- **Troisième étape :** cette fois, les deux orientations (vers le segment et autour du segment) sont combinées. Donc, une orientation est faite vers le segment avec un pas pas\_alpha et pour chaque nouvelle orientation d'angle alpha calculée, nous orientons l'axe de l'outil autour du segment en considérant tout l'intervalle de variation de beta. Cette procédure est répétée jusqu'à obtention des angles alpha et beta évitant les collisions. Si les collisions persistent toujours, le segment considéré n'est pas usiné et il est retiré du trajet d'outil.

D'après ce qui précède, nous constatons que l'orientation de l'extrémité supérieure (point dans l'espace) se fait autour d'un axe quelconque ne passant pas par l'origine.

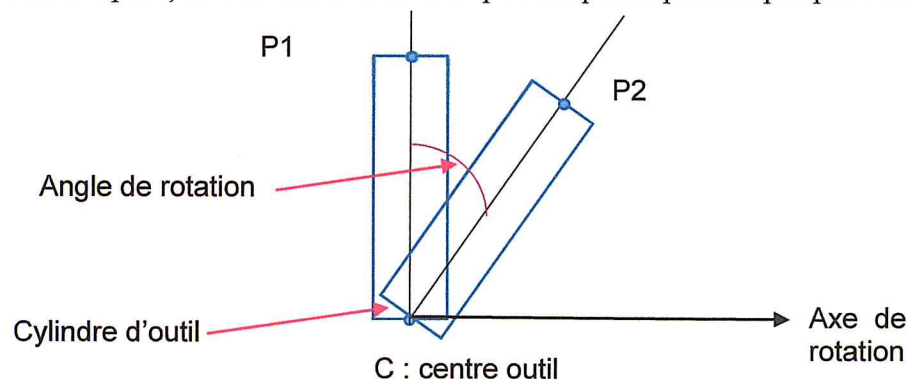


Figure 26 : Rotation de l'outil autour d'un axe.

#### ❖ Comment orienter un point autour d'un axe quelconque ?

Nous avons vu que l'axe d'orientation ne passe pas forcément par l'origine. Ce problème peut être résolu par le changement de base (changement de repère). Le nouveau repère aura comme origine le point centre outil et ses axes sont parallèles aux axes du repère global. Donc, nous effectuons l'orientation de l'extrémité supérieure de l'outil dans le nouveau système d'axes après calcul de ses nouvelles coordonnées (Figure 27).

L'orientation dans le sens d'usinage se fait autour d'un axe  $\vec{U}$  passant par le point centre outil. Cet axe est perpendiculaire à l'axe Z et au segment d'usinage considéré dont les deux points centres outil sont C1 et C2 générés à partir du point du début du segment et du point de la fin du segment. Cet axe est le résultat du produit vectoriel des deux vecteurs unitaires de l'axe Z(0, 0, 1) et de vecteur  $\overrightarrow{C1C2}$ .



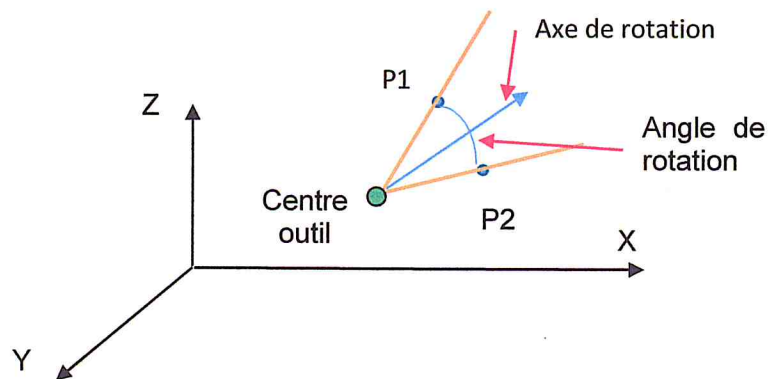
Si nous orientons le point  $P1(X1, Y1, Z1)$  dans le nouveau repère avec un angle  $\alpha$ , alors les coordonnées de son image  $P2(X2, Y2, Z2)$  sont calculées par le produit matriciel de la matrice de transformation avec le vecteur  $\begin{pmatrix} X1 \\ Y1 \\ Z1 \end{pmatrix}$  comme suit :

Soit  $U(u_x, u_y, u_z)$  le vecteur unitaire (**normé**) de l'axe de rotation.

La matrice de transformation des coordonnées est donnée par :

$$\begin{bmatrix} u_x^2 + (1 - u_x^2) * c & u_x * u_y * (1 - c) - u_z * s & u_x * u_z * (1 - c) + u_y * s \\ u_x * u_y * (1 - c) - u_y * s & u_y^2 + (1 - u_y^2) * c & u_y * u_z * (1 - c) - u_x * s \\ u_x * u_z * (1 - c) - u_y * s & u_y * u_z * (1 - c) + u_x * s & u_z^2 + (1 - u_z^2) * c \end{bmatrix}$$

Où  $c = \cos(\alpha)$  et  $s = \sin(\alpha)$



, **Figure 27** : Rotation d'un point P1 autour d'un axe quelconque.

#### 2.4. Simulation cinématique :

La simulation cinématique consiste à visualiser graphiquement l'outil dans toutes les positions de passage de l'outil et cela pour n'importe quelle orientation de son axe. L'outil hémisphérique est composé de deux formes standards (Figure 28) :

- Un cylindre de longueur  $L$  et de rayon  $R$ .
- Une sphère de rayon  $R$ .

Pour simuler les mouvements de l'outil, nous avons besoin de dessiner l'outil dans les différentes positions possibles pour n'importe quelle orientation. Donc le problème est comment représenter graphiquement un cylindre dans l'espace si nous n'avons que le centre de l'outil, son rayon, sa longueur et son point de la limite supérieure du cylindre (Figure 28)?

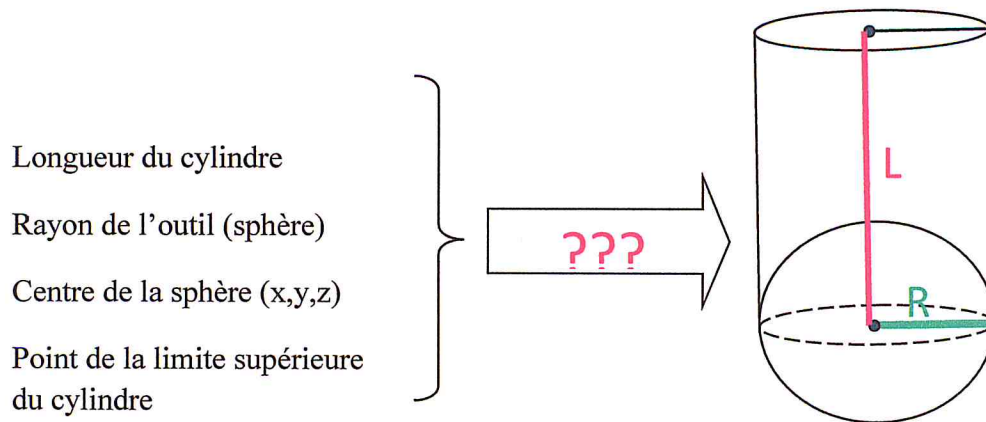


Figure 28 : Décomposition d'un outil hémisphérique.

❖ **Comment visualiser un outil hémisphérique d'orientation quelconque ?**

Pour passer d'un outil hémisphérique réel à une forme graphique, il faut passer par deux étapes :

- **Première étape :** calculer les vecteurs permettant de dessiner l'outil. Les vecteurs sont le vecteur directeur du cylindre  $\vec{w}$  et les deux autres vecteurs orthogonaux entre eux  $\vec{v}$  et  $\vec{u}$  où ces vecteurs appartiennent au plan perpendiculaire à l'axe de l'outil (Figure 29).

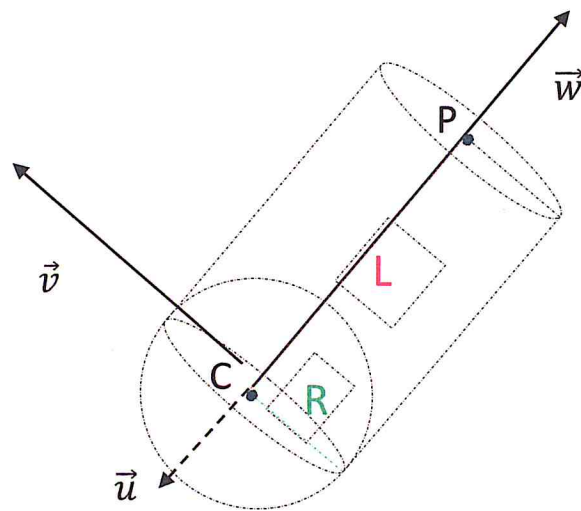


Figure 29 : Vecteurs de positionnement de l'outil.

- **Calcul de  $\vec{w}$  :** ce vecteur est calculé à partir des coordonnées des deux points C (centre outil) et P (point supérieure sur l'axe de l'outil).
- **Calcul de  $\vec{v}$  et  $\vec{u}$  :** pour trouver les composants des vecteurs  $\vec{v}$  et  $\vec{u}$ , il faut déterminer le plan qu'il les contient.

- ✓ Calcul du plan de normale  $\vec{w}$  et passant par le point C :

L'équation d'un plan quelconque est donnée par :

$$a \cdot x + b \cdot y + c \cdot z + d = 0$$

Où a, b et c sont les composantes du vecteur normal au plan. Donc,  $\vec{w}(a,b,c)$  est un vecteur normal à ce plan.

Donc, nous pouvons trouver le plan dont la normale est  $\vec{w}$  et passant par le point C à partir des équations suivantes :

$$C(C_x, C_y, C_z) \in \text{Plan} \Rightarrow d = -(a \cdot C_x + b \cdot C_y + c \cdot C_z)$$

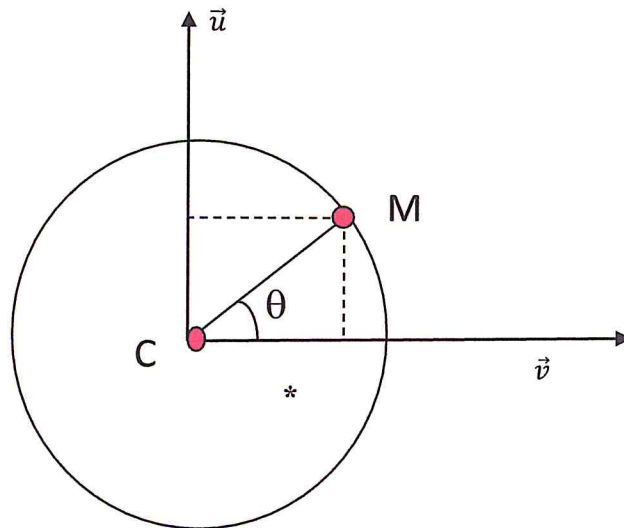
$$\vec{w}(a,b,c) \Rightarrow a \cdot x + b \cdot y + c \cdot z - (a \cdot C_x + b \cdot C_y + c \cdot C_z) = 0$$

- ✓ Détermination d'un point qui appartient au plan : Pour trouver un point P dans le plan calculé, il suffit de fixer deux coordonnées parmi les trois coordonnées du point et déterminer la troisième coordonnée étant donné que ce point appartient au plan. Le calcul de la coordonnée manquante nécessite de considérer trois cas :
  - Si  $a \neq 0$ , alors fixer  $P_z$  et  $P_y$ . Donc,  $P_x = (b \cdot P_y + c \cdot P_z) / a$ .
  - Si  $b \neq 0$ , alors fixer  $P_x$  et  $P_z$ . Donc,  $P_y = (a \cdot P_x + c \cdot P_z) / b$ .
  - Si  $c \neq 0$ , alors fixer  $P_x$  et  $P_y$ . Donc,  $P_z = (a \cdot P_x + b \cdot P_y) / c$ .
- ✓ Déterminer le vecteur  $\vec{v}$  : le vecteur unitaire est calculé à partir des deux points C et P par  $\frac{\vec{CP}}{|\vec{CP}|}$
- ✓ Détermination de  $\vec{u}$  : le vecteur  $\vec{u}$  est perpendiculaire aux vecteurs  $\vec{v}$  et  $\vec{w}$ . Le vecteur unitaire est calculé à partir des vecteurs  $\vec{v}$  et  $\vec{w}$  par  $(\vec{v} \times \vec{w}) / \|\vec{v} \times \vec{w}\|$ .
- **Deuxième étape :** visualisation de la sphère et du cylindre.
  - ✓ Dessin de la sphère : pour dessiner la sphère graphiquement il suffit de connaître son rayon.
  - ✓ Dessin du cylindre : pour dessiner le cylindre, nous devons utiliser une formulation mathématique qui nous permet de trouver les coordonnées x, y et z d'un point M appartenant au plan (P) et appartenant à un cercle de centre C(X0, Y0, Z0) en fonction des vecteurs normés  $\vec{u}$ ,  $\vec{v}$  et  $\vec{w}$  (Figure 30).

$$x = X0 + u1 \cdot \cos(\theta) + \sin(\theta)v1 + t \cdot w1$$

$$y = Y0 + u2 \cdot \cos(\theta) + \sin(\theta)v2 + t \cdot w2$$

$$z = Z0 + u3 \cdot \cos(\theta) + \sin(\theta)v3 + t \cdot w3$$



**Figure 30** : Détermination de point M de cercle.

### CONCLUSION :

Dans ce chapitre, nous avons présenté l'architecture globale de notre application logicielle et les problèmes rencontrés avec les différentes solutions que nous avons proposées pour pouvoir générer un trajet d'outil sain (sans interférences et sans collisions) lors de la finition des surfaces complexes définies par des modèles STL sur des fraiseuses numériques 05-axes en utilisant des outils hémisphériques.

Le test et la validation de notre application sur des surfaces réelles seront considérés dans le chapitre suivant.

# CHAPITRE IV

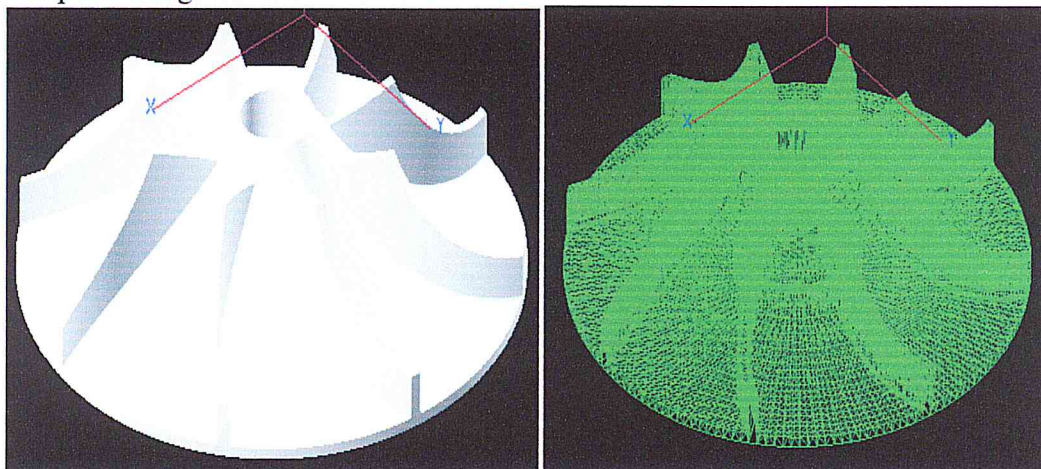
## TEST ET VALIDATION

## 1. INTRODUCTION :

Après les étapes de conception, de proposition des solutions et d'implémentation informatique et d'intégration des différents modules que nous avons développés, nous passons maintenant à l'étape des tests et des validations de notre travail. Pour cela, nous allons tester notre logiciel sur le modèle STL d'une pièce mécanique de formes très complexe. Pendant les tests, l'utilisateur a la possibilité de visualiser et d'effectuer toutes les combinaisons possibles des transformations géométriques (zoom, translations, rotations et projections) en trois dimensions de tous les résultats obtenus (objets géométriques) en utilisant la bibliothèque graphique OpenGL.

## 2. Tests et validations :

Les tests et les validations sont effectués sur la pièce représentée par la Figure 1.a. Cette pièce est conçue dans un logiciel spécialisé de CAO et elle est composée de surfaces de formes très complexes. La pièce est une roue à ailettes utilisées dans les compresseurs. Dans ce même logiciel, la pièce est approximée par un ensemble de triangles. Ces triangles sont stockés dans un fichier STL (Figure 1.b) qui sera utilisé comme une entrée pour le logiciel.



a. Modèle CAO.

b. Modèle STL.

**Figure 1** : Modèle CAO et modèle STL de la pièce de test.

L'interface du logiciel développé contient une seule fenêtre qui est composée de cinq pages représentant la séquence des actions à mener pour aboutir au résultat final. Les pages sont :

- Lecture du fichier STL.
- Création des contours.
- Points de passage de l'outil et rayons optimums.
- Détection et correction des interférences et des collisions.
- Simulation des mouvements de l'outil.

Dans les paragraphes suivants, nous allons suivre cet enchainement.

## 1. Lecture de fichier STL :

La première page est « Lecture du fichier STL » (Figure 2). Dans cette page, l'utilisateur doit cliquer sur le bouton « Ouvrir fichier STL » pour pouvoir sélectionner un fichier STL à ouvrir. Après l'ouverture du fichier, l'utilisateur doit sélectionner un mode de calcul des normales aux sommets et par la suite cliquer sur le bouton « Calculer la normale ». Une fois les résultats sont obtenus, l'utilisateur peut :

- Récupérer les paramètres de la forme brute.
- Visualiser les sommets du modèle (Figure 3).
- Visualiser les triangles du modèle (Figure 4).
- Visualiser les normales estimées aux sommets (Figure 5).

Pour ce modèle : le nombre total des sommets et le nombre total des triangles sont égaux respectivement à 52137 et 104266.

- Les paramètres de la pièce brute sont :

$X_{min} = 0.393$        $Y_{min} = 0.393$        $Z_{min} = 3.6747$   
 $X_{max} = 130.139$        $Y_{max} = 130.139$        $Z_{max} = 33.6747$ .

Longueur = 30mm

Largeur = 129.8mm

Hauteur = 129.8mm

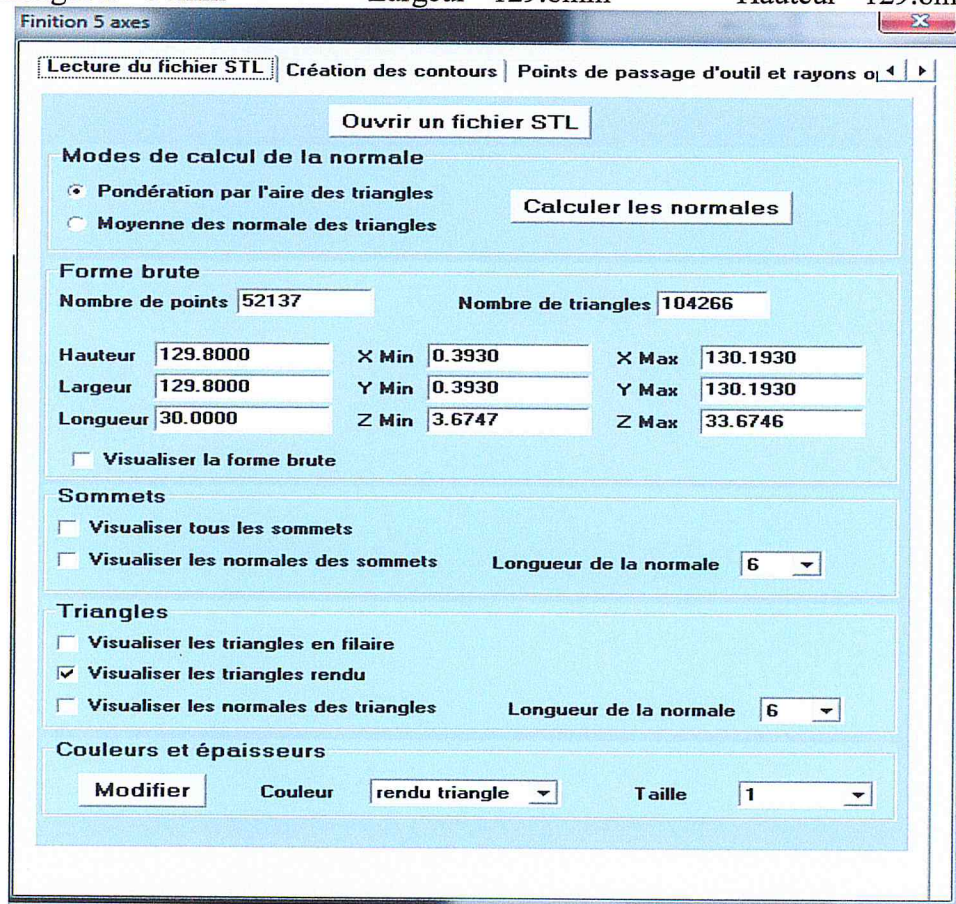
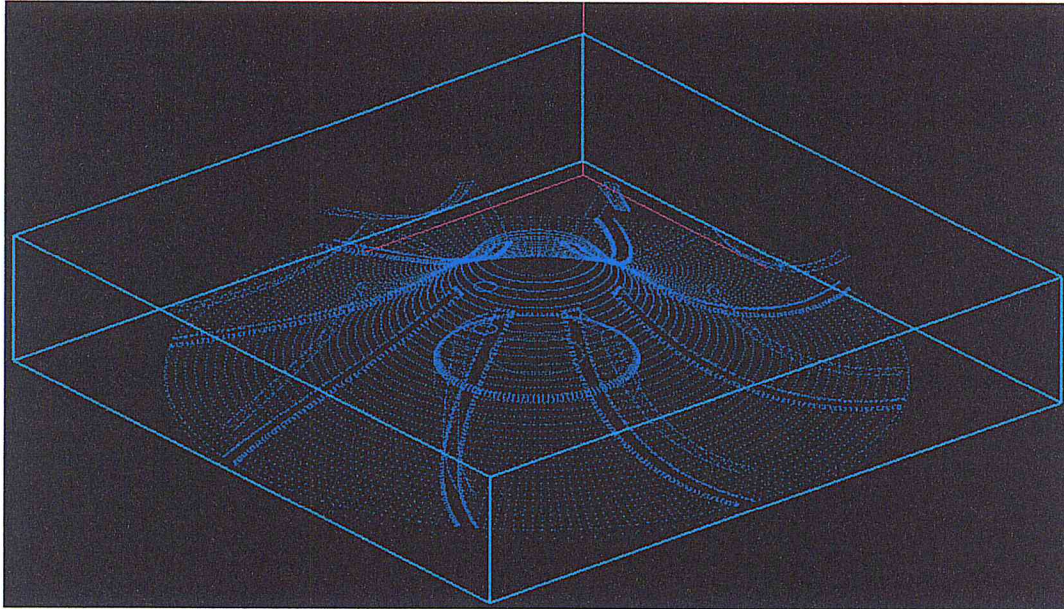
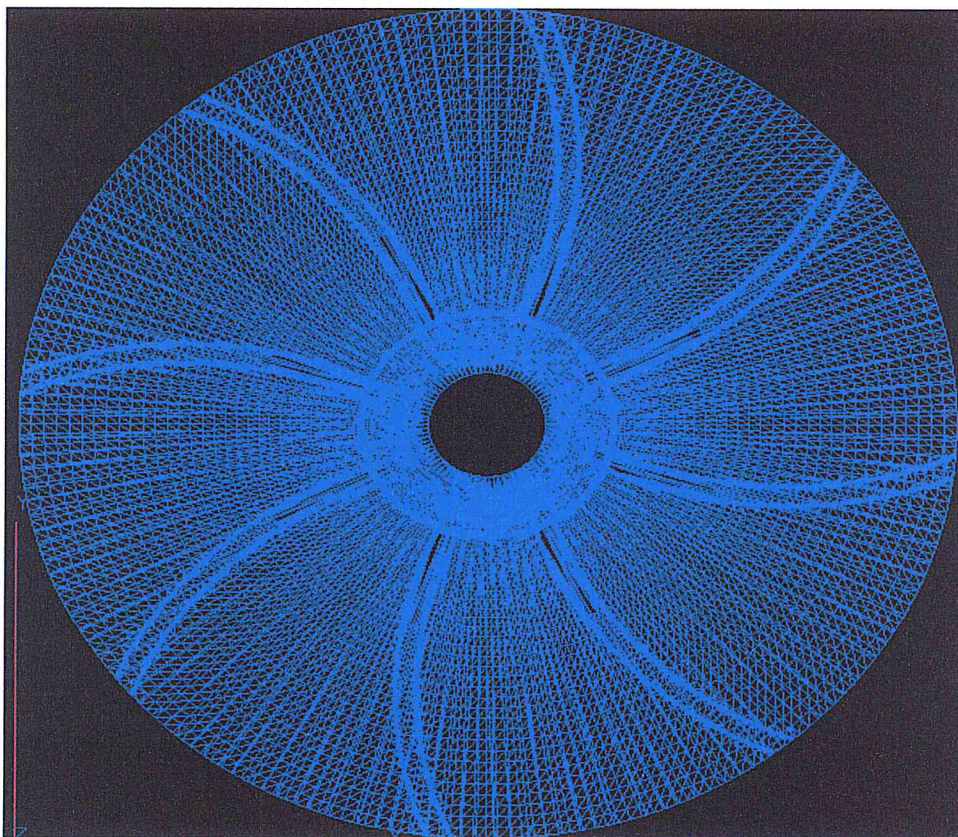


Figure 2 :Page « Lecture du fichier STL ».

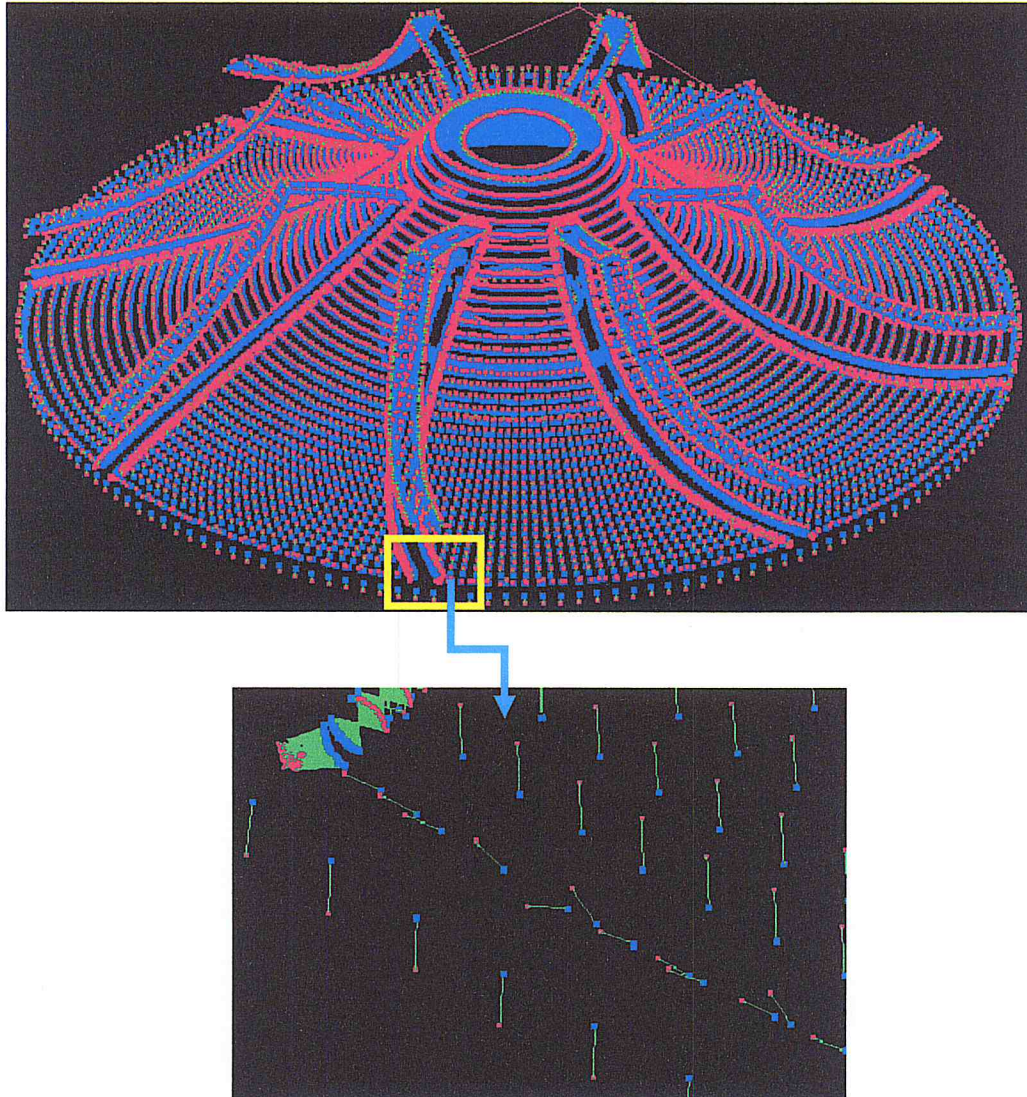


**Figure 3 :** Enveloppe de la pièce brute et sommets du modèle STL.



**Figure 4 :** Triangles du modèle STL.





**Figure 5** : Normales aux sommets.

## 2. Création des contours :

La deuxième page est « Création des contours » (Figure 6). Dans cette page, l'utilisateur doit spécifier le nombre de bandes de triangles à créer et la profondeur de passe et par la suite il doit cliquer sur le bouton « création des contours » pour lancer les calculs.

Pour la création des contours, nous avons choisi une profondeur de passe égale à 5mm et un nombre de bandes égal à 4. Après l'exécution de cette fonction, l'utilisateur peut visualiser plusieurs objets graphiques tels que :

- Les plans d'usinage (Figure 7).
- Les bandes des triangles (Figure 8).
- Les points d'intersections (Figure 9).
- Les contours (Figure 10).

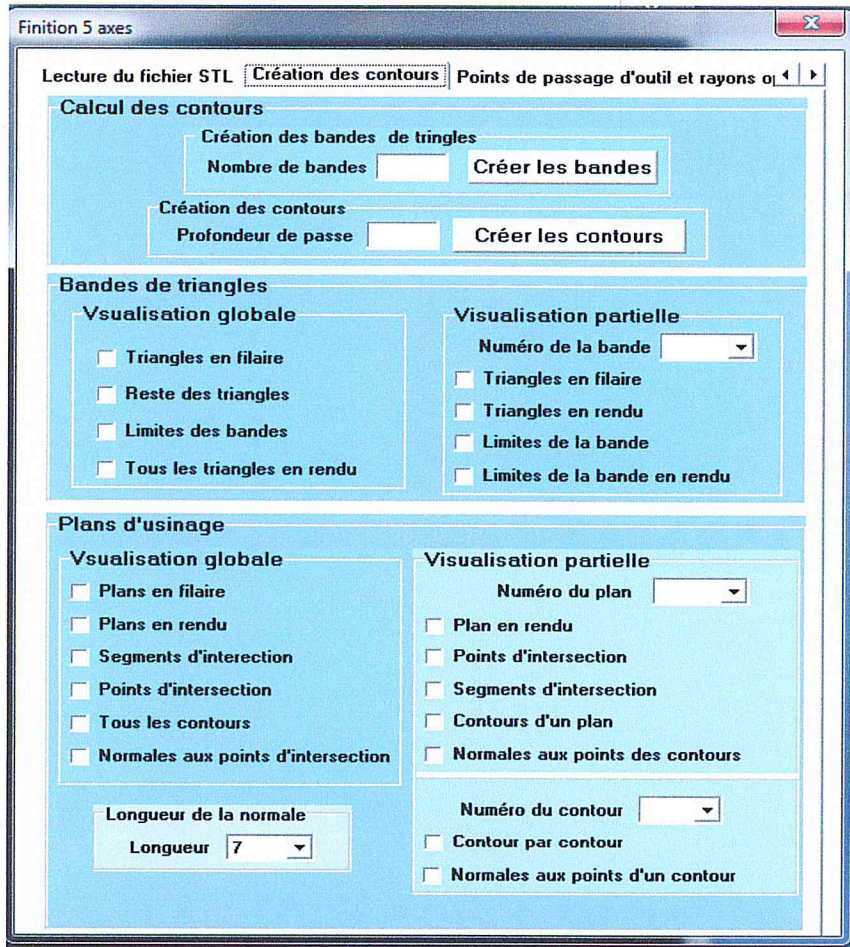


Figure 6 :Page « Création des contours ».

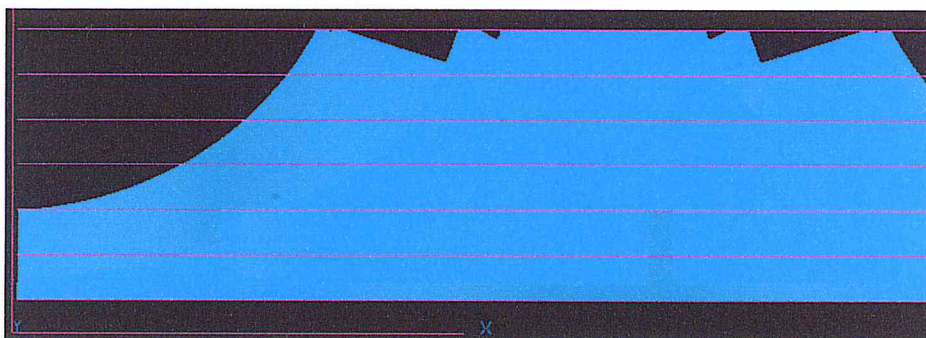
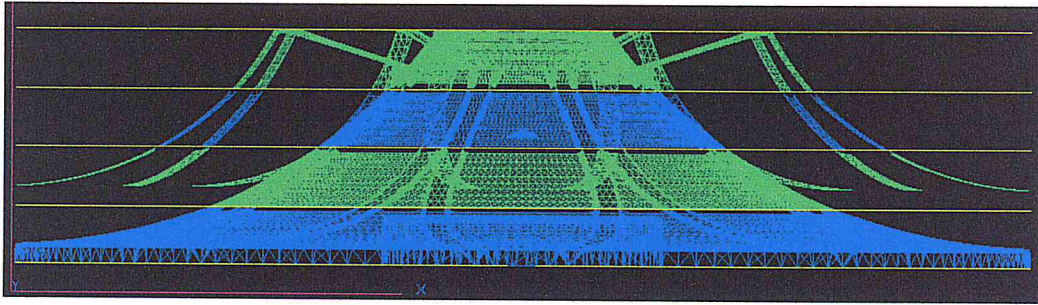
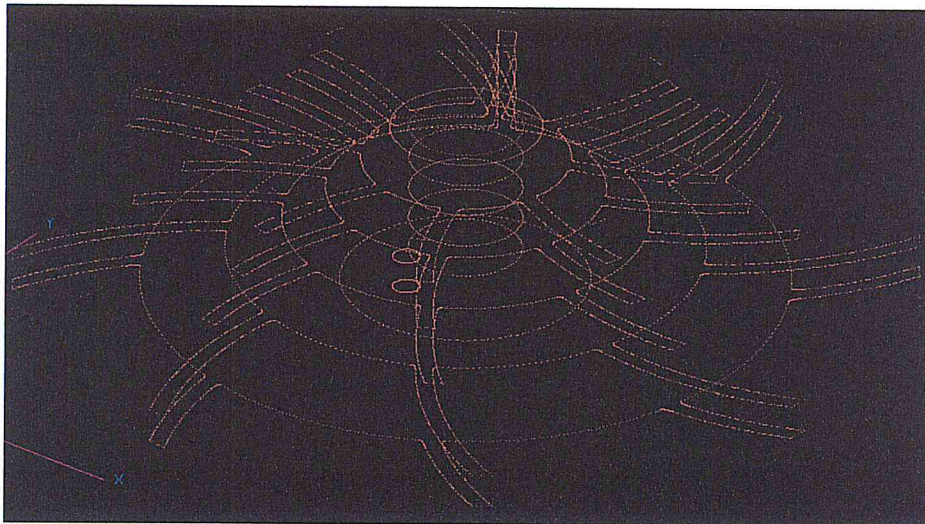


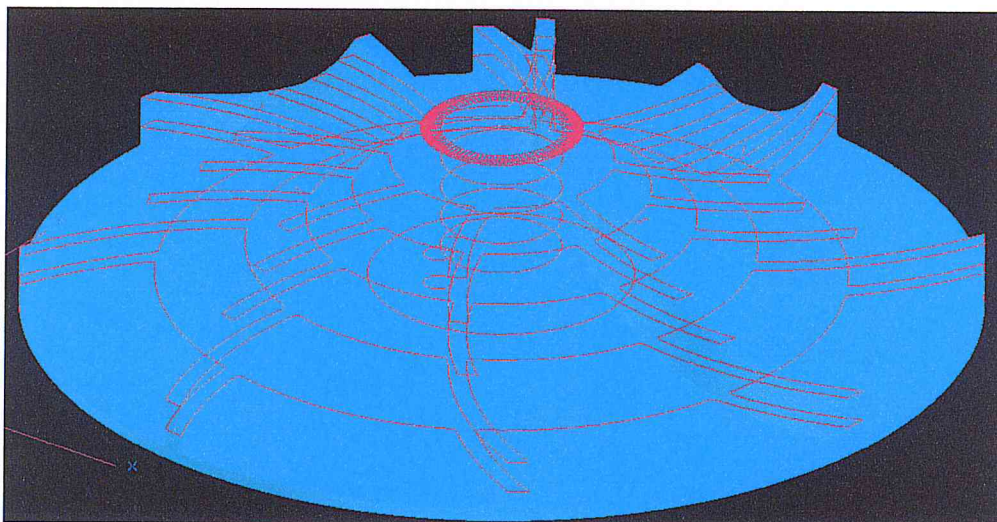
Figure 7 : Plans d'usinage.



**Figure 8** : Bandes de triangles.



**Figure 9** : Points d'intersection (points de contact outil-surface).



**Figure 10** : Contours de contact outil-surface.

### 3. Points de passage de l'outil :

La troisième page est « Points de passage de l'outil » (Figure 11). Dans cette page, l'utilisateur doit spécifier le rayon et la longueur de l'outil hémisphérique et par la suite il doit cliquer sur les différents boutons pour calculer les centres outil.

Pour cette étape, nous avons choisi le rayon de l'outil égal à 5mm et la longueur de l'outil égale à 100mm. Après l'exécution de cette fonction, l'utilisateur peut visualiser plusieurs objets graphiques tels que :

- Points de passage de l'outil (Figure 12).

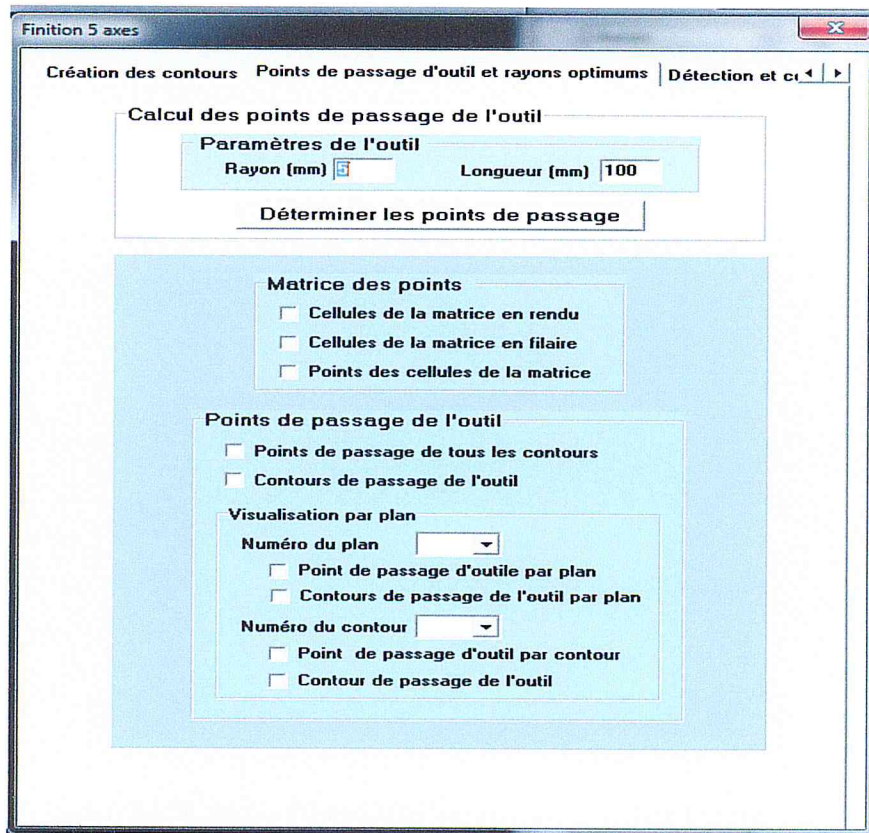
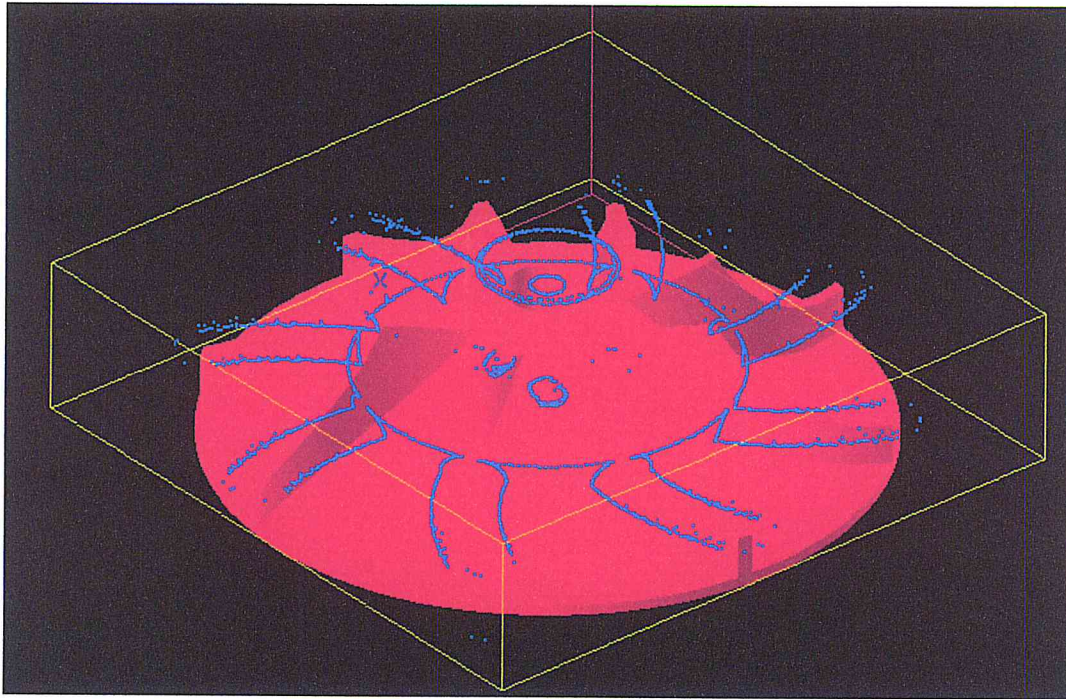


Figure 11 :Page « Points de passage de l'outil »



**Figure 12 :** Points de passage de l'outil.

#### **4. Détection et correction des interférences et des collisions :**

La quatrième page est « Détection et correction des interférences et des collisions » (Figure 13). Dans cette page, l'utilisateur détecte les interférences et les collisions et par la suite il lance les différentes corrections une fois les angles limites et les pas associés sont spécifiés. Après l'exécution de cette fonction, l'utilisateur peut visualiser plusieurs objets graphiques tels que :

- Points créant des interférences (Figure 14).
- Points créant des collisions (Figure 15).
- Orientations de l'outil lors de la correction des collisions (Figure 16).

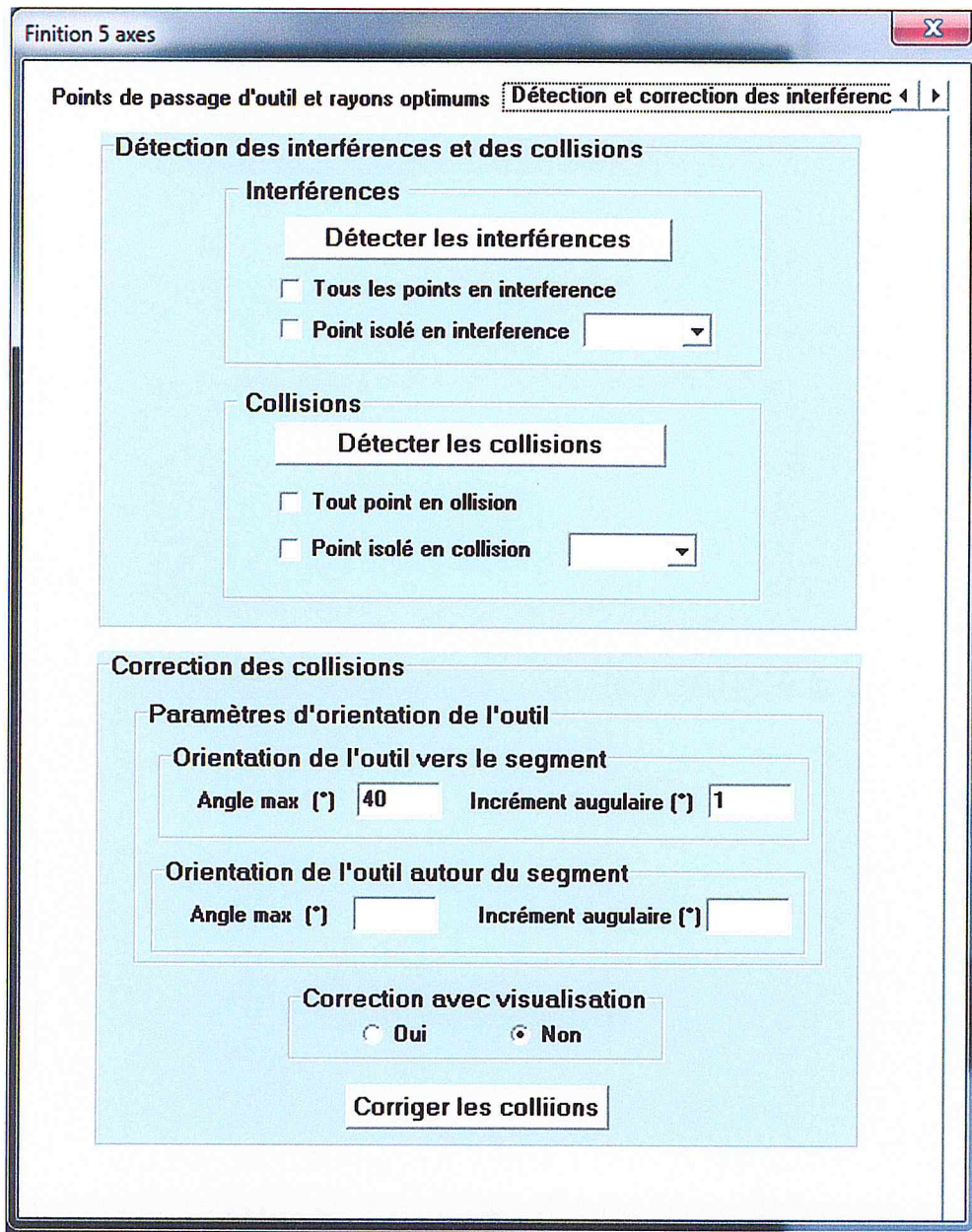


Figure 13 : Page « Détection et correction des interférences et des collisions ».

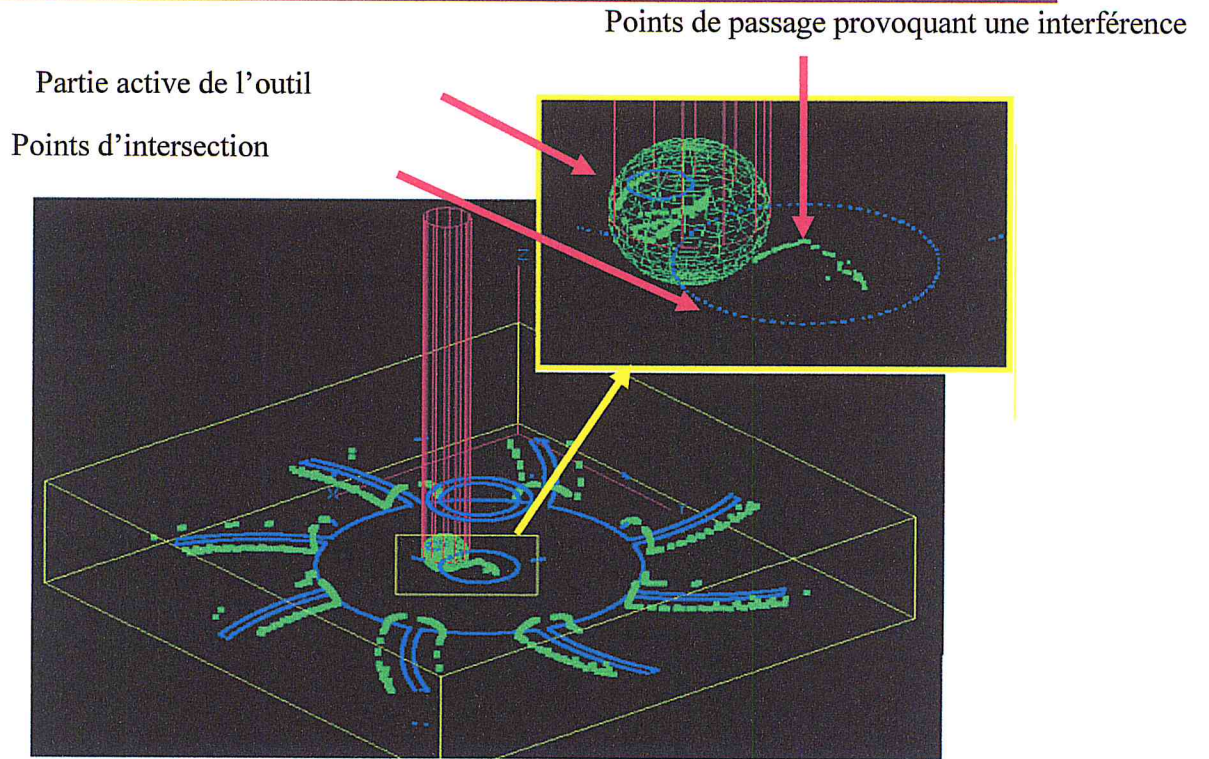


Figure 14: Points en interférences.

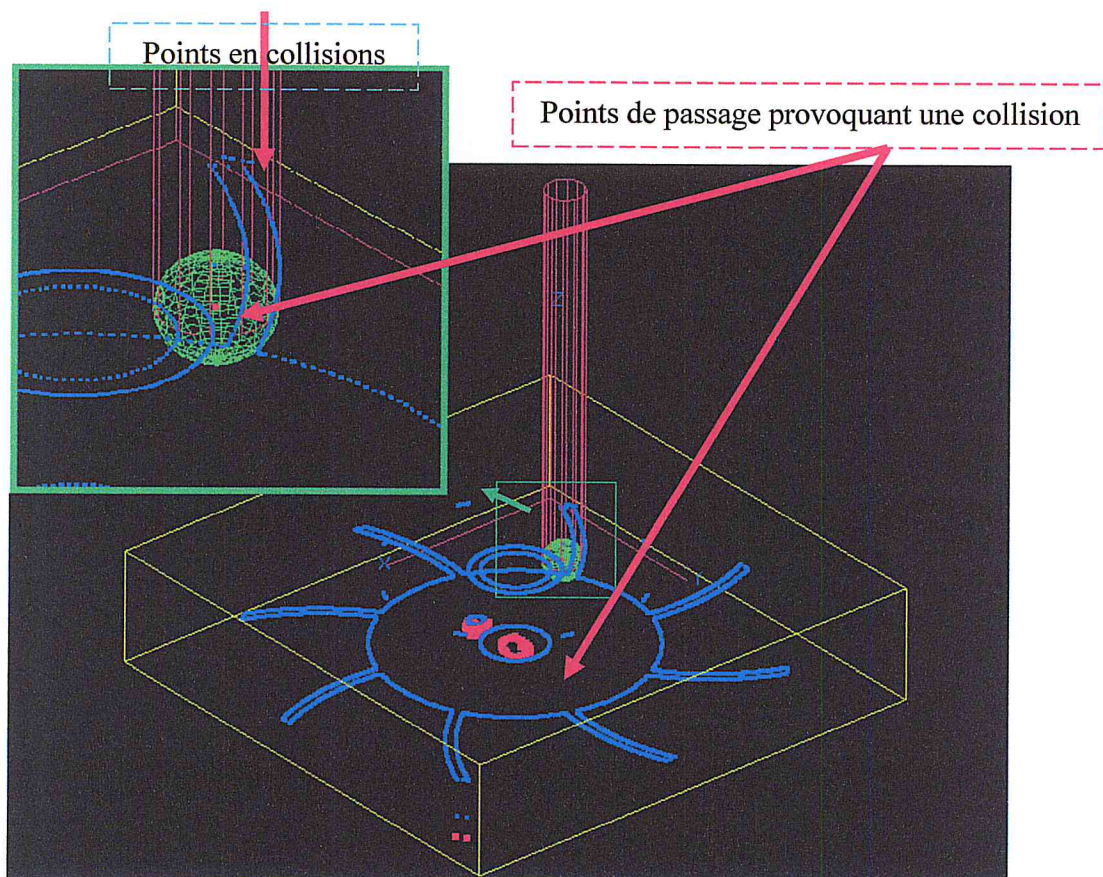


Figure 15 : Points en collisions.

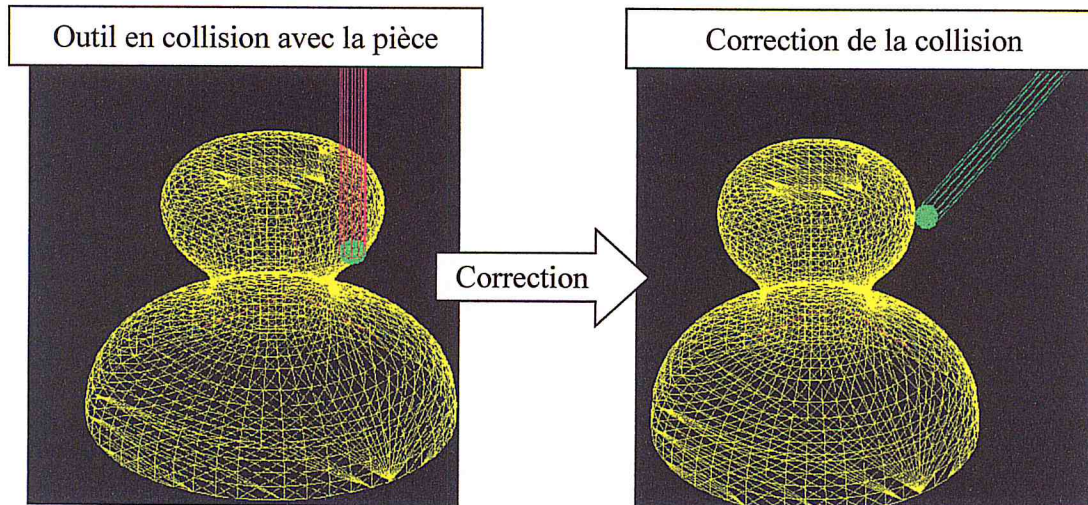


Figure 16: Orientation de l'outil avant et après la correction des collisions.

### 5. Simulation des mouvements de l'outil :

La cinquième page est « Simulation » (Figure 17). Dans cette page, l'utilisateur peut simuler les mouvements de l'outil le long de la trajectoire d'usinage. Il a la possibilité de lancer, de mettre la simulation en pause, de continuer la simulation et de l'arrêter. Lors de la simulation, en chaque position de passage de l'outil, les coordonnées du point centre outil, les angles d'orientation de l'outil et les paramètres de l'outil sont affichés.

- Outil en différentes positions (Figure 18,19,20).

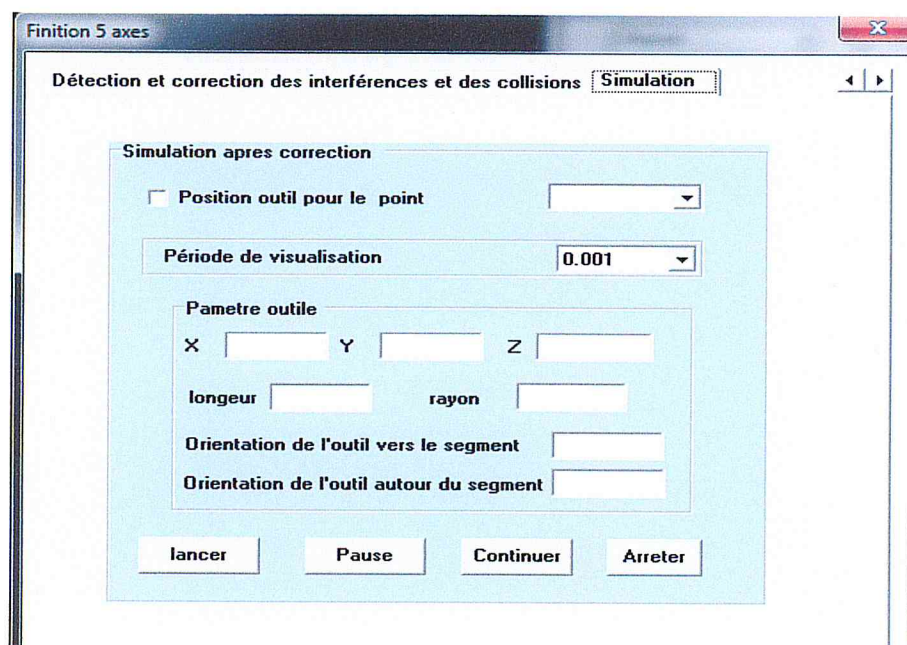


Figure 17 : Page « Simulation ».



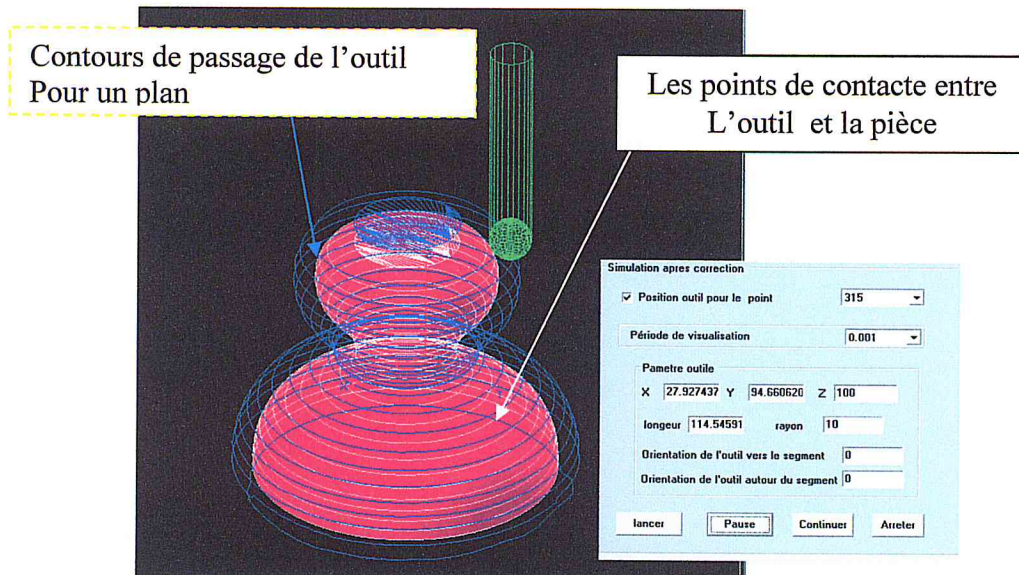


Figure 18 : Outil en position 1.

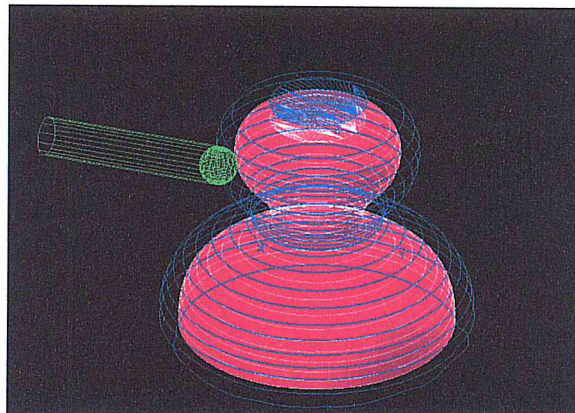


Figure 19 : Outil en position 2.

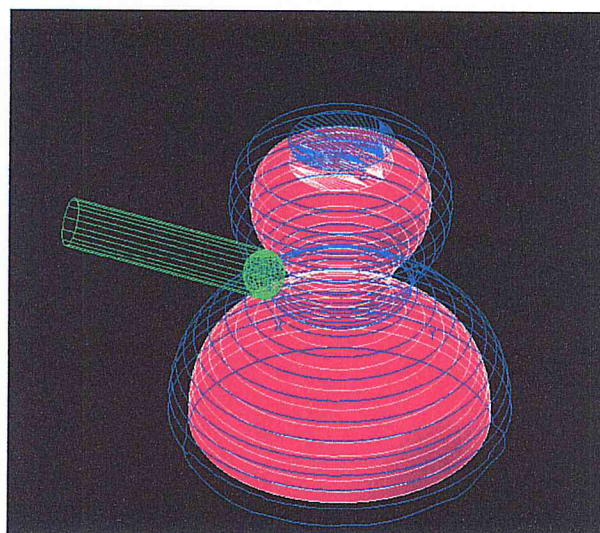


Figure 20 : Outil en position 3.

### 3. CONCLUSION :

Dans ce chapitre, l'application a été testée sur une pièce de forme très complexe pour montrer la succession des étapes de notre approche. L'application a été déroulée depuis la lecture du fichier STL jusqu'à la simulation des mouvements de l'outil le long de la trajectoire d'usinage. L'application logicielle développée est un outil d'aide pour la génération du trajet de finition en 05-axes.

**CONCLUSION**

**GENERALE**

## Conclusion générale :

Nous avons présenté dans ce mémoire notre projet qui consiste à concevoir et à réaliser une application logicielle graphique et interactive pour la finition des surfaces complexes, définies par des modèles STL, par la stratégie d'usinage Z-Constant sur des fraiseuses numériques à 05-axes avec des outils hémisphériques. Cette application logicielle permet de générer une trajectoire d'outil sans interférences et sans collisions.

Lors de la réalisation de ce projet, une étude bibliographique a été faite sur les modèles de représentation des surfaces complexes, les machines-outils à commande numérique et les différentes stratégies d'usinage. Cette étude a été complétée par une étude conceptuelle de l'application et une implémentation informatique des solutions proposées aux différentes questions posées durant la réalisation de cette approche. A la fin, des tests de validation ont été menés sur une pièce de forme très complexe.

Les principaux résultats de notre application logicielle sont les suivants :

1. Lecture du modèle de la pièce donné par un fichier STL et récupération de tous les paramètres géométriques (sommets et triangles) ainsi que les paramètres de de la forme brute.
2. Estimation de la normale pour chaque sommet.
3. Création des plans d'usinage.
4. Groupement des triangles dans des bandes de triangles.
5. Calcul des points d'intersection (points de contact outil-surface) entre les plans d'usinage et les triangles.
6. Minimisation du temps de calcul des points d'intersection.
7. Estimation de la normale aux points d'intersection.
8. Génération des segments d'intersection.
9. Création des contours de chaque plan.
10. Calcul de points de passage de l'outil (points centre-outil).
11. Détection et correction des interférences entre la partie active de l'outil et la pièce en chaque point de passage de l'outil.
12. Détection et correction des collisions en chaque point de passage de l'outil.
13. Minimisation du temps de détection des interférences et des collisions.
14. Simulation virtuelle des mouvements de l'outil avant et après correction des interférences et des collisions.

En perspective de notre travail, nous proposons de traiter les thématiques suivantes :

- Minimisation du temps d'usinage en combinant plusieurs outils hémisphériques.

- Détermination des profondeurs de passes optimales pour respecter la hauteur de crête afin d'obtenir un bon état de surface.
- Détermination des rayons optimums pour éviter les interférences.
- Optimisation du mode d'orientation de l'outil lors de la correction des collisions.
- Détermination de l'inclinaison optimale de l'outil hémisphérique permettant d'augmenter la vitesse de coupe et par conséquent améliorer l'état de surface.
- Détermination du chemin optimal lors de raccordement des différents contours d'usinage pour minimiser le temps de déplacement à vide.
- Application de la stratégie des Plans-Parallèles pour la finition des surfaces complexes.
- Combinaison des deux stratégies d'usinage Z-Constant et Plans-Parallèles.
- Usinage avec des outils cylindriques.
- Usinage avec des outils toriques.
- Combinaison des trois formes d'outils hémisphériques, cylindriques et toriques.
- Combinaison des deux stratégies d'usinage et des trois formes d'outils pour minimiser le temps d'usinage.
- Ebauchage des surfaces complexes sur des fraiseuses numériques à 05-axes.
- Simulation virtuellement l'opération d'usinage (enlèvement de matière).
- Détermination des conditions de coupe optimales (vitesse d'avance et vitesse de coupe).
- Lissage des points de passage l'outil par des courbes continues pour avoir un mouvement plus fluide.

**REFERENCES**

**BIBLIOGRAPHIQUES**

## REFERENCES BIBLIOGRAPHIQUES

1. A. Doneddu, « Géométrie différentielle, intégrales multiples », Tome 6, Editions Vuibert, 1981.
2. Z. Tchantchane, « Automatisation de l'opération de Tréflage des surfaces gauches à partir d'un model STL » université Amar Telidji, Laghouat, 2010/2011.
3. Raphaël LA GRÉCA, « Segmentation de Nuages de Points et Courbures Discrètes pour la Reconstruction », Rapport de stage, DEA Informatique, Laboratoire d'Informatique de Luminy, juin2001.
4. Site Web: [www.michel.baehrel.pagesperso-orange.fr/.../Tournage,%20fraisage,%20mach,15/3/2015](http://www.michel.baehrel.pagesperso-orange.fr/.../Tournage,%20fraisage,%20mach,15/3/2015).
5. Manuel d'utilisation SIEMENS : « SINUMERIK 810D/840D, Fabrication de pièces complexes en fraisage », 2004.
6. Raphael Laguionie, « Programmation avancée des machines-outils : implémentation de l'usinage trochoidal et du Tréflage dans la chaîne numérique Step-NC », Thèse de Doctorat, Ecole Centrale de Nantes, France, 2007.
7. Documentation CNC3 Axes V2.xx, Edition du 27/11/06.
8. Cours sur le Web, «Opération d'usinage et d'assemblage, Réalisation des surfaces, Ebauche, 1/2 Finition et Finition ».
 

Site Web: <http://prof-mpk.fr/Centres.../cours%20ebauche%20finition%20prof.pdf>, 20/02/2015 .
9. Cours sur le Web, « Coupe des Métaux ».
 

Site Web: <http://umc.edu.dz/vf/images/cours/Fabrication Mécanique /chapitreii.pdf>, 12/3/2015 .
10. Patrick GILLES, « Positionnement d'outil torique pour l'usinage de surfaces gauches en fraisage 5-axes avec équilibrage de l'effort de coupe transversal », Thèse de Doctorat, l'Université Toulouse III - Paul Sabatier, France, 2008.
11. Mohamed BEY, « Usinage des surfaces gauches à partir d'un nuage de points », rapport de recherche, CDTA, Algérie, 2012.
12. Sylvain Lavernhe, « Prise en compte des contraintes associées au couple MO-CN en génération de trajectoires 5 axes UGV », Thèse de Doctorat, École normale supérieure de Cachan, France, 2006.
13. D.C.H. Yang, Z. Han, Interference detection and optimal tool selection in 3-axis NC machining of free-form surfaces, January 1999.

14. Chen-Ming Chuang, Hong-Tzong Yau, A new approach to z-level contour machining of triangulated surface models using fillet endmills, National Chung Cheng University, octobre 2004.





The page is otherwise blank and contains no text or other content.