



**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur**

**Et de la Recherche Scientifique**

**CDTA**

**UNIVERSITE SAAD DAHLAB DE BLIDA**

**FACULTE DES SCIENCES**

**DEPARTEMENT D'INFORMATIQUE**

**MEMOIRE DE FIN D'ETUDE**

**Pour l'obtention**

**D'un diplôme de Master en Informatique**

**Option : Systèmes Informatiques et Réseaux**

**Thème :**

**Conception et Développement d'une Application  
Logicielle Pour la Simulation d'Usinage des Pièces  
Complexes sur Fraiseuses Numériques à 03-Axes**

**Réalisé par :**

M<sup>elle</sup>. LEZAMI Samira

M<sup>elle</sup>. MADI Nada

**Soutenu devant :**

Mr. BEY Mohamed	CDTA	Encadreur
Mr. BENDIFALLAH Hassène	CDTA	Encadreur
Mr. HAMOUDA Mohamed	USDB	Promoteur
Mr.	USDB	Président
Mr.	USDB	Examineur

**2019/2020**

## Résumé

Ce travail s'insère dans le cadre de développement d'une plateforme logicielle pour la production des surfaces de formes complexes initié par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) de la Division Productique et Robotique (DPR) du Centre de Développement des Technologies Avancées (CDTA).

Dans ce projet nous nous intéressons à la simulation des opérations d'usinage des pièces de formes complexes définies par leurs modèles STL sur des fraiseuses numériques à 03-axes. Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle, un module logiciel graphique et interactif permettant de simuler toutes les opérations d'usinage (ébauchage, demi-finition et finition) pour tout trajet d'outils en considérant les outils hémisphériques et cylindriques tout en vérifiant la conformité des surfaces résultantes.

**Mots Clés :** Pièce Complexe, Modèle STL, Simulation Usinage, Fichier CL-File, Trajet Outil, Fraiseuse 03-axes.

## Summary

This work is part of a software platform for the production of complex surfaces initiated by the Computer Aided Design and Manufacturing (CAD / CAM) team of the DPR Division of the Center for Development of Advanced Technologies (CDTA).

In this project, we are interested in the simulation of machining operations of complex parts defined by their STL models on 03-axis CNC milling machines. This involves designing, developing and integrating into the software platform, a graphic and interactive software module to simulate all machining operations (roughing, semi-finishing and finishing) for any tool path, considering hemispherical and cylindrical tools while checking the conformity of the resulting surfaces.

**Keywords:** Complex Part, STL Model, Machining Simulation, CL-File, Tool Path, 03-axis CNC milling machine.

## ملخص

هذا العمل جزء من تطوير وحدات برمجية لإنتاج القطع ذات الأشكال المعقدة التي بدأها فريق التصميم والتصنيع بمساعدة الكمبيوتر (CFAO) التابع لقسم الانتاجية والروبوتات (DPR) في مركز تطوير التقنيات المتقدمة (CDTA).

في هذا المشروع، نحن مهتمون بمحاكاة عمليات تصنيع الأجزاء المعقدة الشكل المحددة بواسطة نماذج STL الخاصة بهم على آلات التفريز الرقمية ذات 03 محاور. يتضمن ذلك تصميم، تطوير ودمج في منصة البرامج، وحدة برمجية تفاعلية ورسومات تجعل من الممكن محاكاة جميع عمليات التصنيع (التخشين، نصف التشطيب والتشطيب) لأي مسار أداة، باستخدام أدوات نصف كروية وأسطوانية مع التحقق من مطابقة الأسطح الناتجة

**الكلمات المفتاحية:** قطعة معقدة، نموذج STL، محاكاة القطع، CL-File، مسار الأداة، آلة التفريز ثلاثية المحور.

## ***Remerciements***

*On tient à remercier avant tout Allah le tout puissant de nous avoir donné la foi, la volonté et le courage pour mener à bien ce modeste travail, **El Hamdou Li Allah.***

*Nous adressons le grand remerciement à notre encadreur Monsieur Bey Mohamed qui a bien voulu nous accueillir au sien de l'équipe CFAO. Nous le remercions pour la documentation mise à notre disposition, son aide précieuse et ses conseils tout au long de ce projet.*

*Nous tenons également à remercier notre promoteur Monsieur Hamouda Mohamed pour la confiance qu'il nous a fait en acceptant de diriger ce travail pour son assistance ininterrompue et ses conseils judicieux qui nous ont aidés à mener à bout ce travail.*

*Nos remerciements vont particulièrement à Messieurs les membres du jury pour avoir accepté d'évaluer et de juger notre modeste travail.*

*Nos remerciements s'adressent aussi à tous les enseignants et à tous ceux qui nous ont accompagnés tout au long des deux années de Master.*

*Nous ne pouvons terminer sans remercier nos familles pour leurs soutiens indéfectibles, nos amis et camarades de promotion pour leurs présences et conseils ainsi que tous ceux qui ont contribué de près ou de loin à la bonne réalisation de ce travail.*

## ***Dédicaces***

*Je dédie ce modeste travail :*

*A mes chers parents, **Madi Ali et Madi Hamida,***

*Pour tous leurs sacrifices, leurs amours, leurs tendresses, leurs soutiens et  
leurs prières tout au long de mes études,*

*A mes chers sœurs et frères, pour leurs appuis et leurs encouragements,*

*A mes belles sœurs et beaux-frères.*

*A mes jolis neveux.*

*A mes chères amies **Loubna, Manel, Madina, Ikram...** et à tous celles que je  
n'ai pas citées...*

*A ma sœur et binôme **Samira** et toute la famille **Lezami.***

*A toute la famille **Madi***

*A toute ma famille pour leur soutien tout au long de mon parcours  
universitaire.*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de  
votre soutien infailible,*

*Merci d'être toujours là pour moi.*

**MADI NADA**

# Dédicace

*Tout d'abord, nous remercions DIEU ALLAH, de nous avoir donné la volonté et le courage et la patience afin d'arriver à la finalité de ce modeste travail.*

*À mon père LEZAMI Abderrahmen*

*L'épaule solide, l'œil attentif compréhensif et la personne la plus digne de mon estime et de mon respect. Aucune dédicace ne saurait exprimer mes sentiments. Que Dieu te préserve et te procure santé et longue vie.*

*À ma mère LEZAMI Fadila*

*Tu m'a donné la vie, la tendresse et le courage pour réussir. Tout ce que je peux t'offrir ne pourra pas exprimer l'amour et la reconnaissance que je te porte.*

*À mes très chères sœur Sihem ♥ Asmaa ♥*

*À mes très chères frères Walid ♥ Abdelghani ♥*

*À la personne qui m'a soutenu toute l'année Mon binôme et ma chère collègue Nada j'ai partagé tous ces moments de stress, de joie, de panique, de délire et surtout de guigne ensemble.*

*Une spéciale dédicace aux personnes les plus chère à mon cœur Manel, Ikram et loubna merci d'être toujours à mes côtés, pour les moments agréables que nous avons passés ensemble.*

*Veillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.*

*À toutes chères amies : Amina, Amel, Hayat, Khadija, Madina, Yasmine, Amira, Djohar, Nesrine, Radia, Chaïmaa, Rania, Ghanya, Imene merci pour votre soutien moral.*

*À tous les étudiants de la promotion 2019/2020.*

*Enfin Toute personne qui m'aime et que j'aime ...*

*SAMIRA...*

---

<b>Introduction générale</b> .....	1
------------------------------------	---

## **Chapitre 1 : Etude Bibliographique**

Introduction .....	4
1. Processus de fabrication des pièces de formes complexes .....	4
2. Conception et Fabrication Assistée par Ordinateur « CFAO ».....	5
2.1 Conception Assistée par Ordinateur « CAO » .....	5
2.2 Format d'échange de données .....	6
2.3 Fabrication Assistée par Ordinateur « FAO » .....	8
3. Simulation d'usinage .....	13
3.1 Méthodes d'approximation des surfaces .....	14
3.2 Simulation et vérification et correction.....	15
3.3 Catégories de simulation .....	16
3.4 Echelles de simulation .....	16
3.5 Modèles de la simulation.....	17
3.6 Méthodes de simulation .....	18
Conclusion .....	21

## **Chapitre 2 : Etude Conceptuelle**

Introduction.....	23
1. Solution proposée.....	23

1.1 Récupération des paramètres du fichier STL et création du brut .....	24
1.2 Création des cellules .....	24
1.3 Affectation des sommets et des triangles aux cellules.....	25
1.4 Création des Dexels .....	26
1.5 Calcul des points théoriques.....	28
1.6 Enrichissement du trajet d’outil.....	30
1.7 Calcul des intersections.....	31
1.8 Mise à jour des droites .....	34
1.9 Calcul des écarts .....	35
2. Modélisation UML (Unified Modeling Language).....	36
2.1 Diagramme de cas d’utilisation .....	36
2.2 Diagramme d’activité.....	36
2.3 Diagramme de classes .....	41
Conclusion .....	47

### **Chapitre 3 : Implémentation Informatique et Validation**

Introduction .....	49
1. Présentation des outils utilisés .....	49
1.1 Présentation du langage C++.....	49
1.2 Présentation d’OpenGL (Open Graphics Library) .....	49
1.3 Présentation d’Embarcadero Builder C++ .....	50
2. Présentation de la plateforme logicielle.....	50

2.1 Fenêtres principales .....	50
2.2 Barre du menu principal.....	50
2.3 Simulation 3 Axes.....	52
3. Test et validation .....	56
3.1 Premier modèle STL « modèle de finition » .....	57
3.2 Deuxième modèle STL « modèle d'ébauchage » .....	60
Conclusion .....	64
<b>Conclusion générale .....</b>	<b>65</b>
<b>Références bibliographiques.....</b>	<b>67</b>



**CHAPITRE 1**

Figure 1.1. Processus de réalisation des formes complexes .....	4
Figure 1.2. Echange de données entre deux systèmes. ....	6
Figure 1.3. Modèle STL.....	6
Figure 1.4. Représentation CAO, représentation STL et paramètres d'un triangle. ....	7
Figure 1.5. Formats STL ASCII et BINAIRE. ....	7
Figure 1.6. Phases d'usinage. ....	8
Figure 1.7. Stratégie d'usinage.....	9
Figure 1.8. Stratégies les plus utilisées pour la finition. ....	10
Figure 1.9. Formes de fraises.....	10
Figure 1.10. Positionnement des différentes formes d'outils.....	11
Figure 1.11. Problèmes d'usinage.....	12
Figure 1.12. Fraiseuse horizontale.....	13
Figure 1.13. Approximation d'une surface par des triangles.....	14
Figure 1.14. Types de triangulations .....	14
Figure 1.15. Méthode du Z-Buffer.....	15
Figure 1.16. Simulation d'usinage entre FAO et machine. ....	16
Figure 1.17. Architecture générale de la simulation de l'usinage.....	16
Figure 1.18. Orientation des vecteurs suivant la normale et l'axe de l'outil. ....	18
Figure 1.19. Principe de la méthode du Z-Buffer. ....	19
Figure 1.20. Représentation d'une pièce par le Z-Buffer. ....	19
Figure 1.21. Représentation d'un volume par des Voxels.....	20
Figure 1.22. Représentation d'un solide par des Dexels.....	20

## CHAPITRE 2

Figure 2.1. Organigramme général de la démarche proposée. ....	23
Figure 2.2. Dimensions minimales du brut. ....	24
Figure 2.3. Création des Cellules. ....	24
Figure 2.4. Dimensions minimales d'une cellule. ....	25
Figure 2.5. Affectation des sommets aux cellules. ....	25
Figure 2.6. Affectation des triangles aux cellules. ....	26
Figure 2.7. Points d'une droite. ....	26
Figure 2.8. Création des Dexels. ....	27
Figure 2.9. Affectation des droites aux cellules. ....	27
Figure 2.10. Intersection droite triangle. ....	28
Figure 2.11. Enveloppe du triangle. ....	28
Figure 2.12. Cas d'appartenance et non appartenance d'une droite à l'enveloppe. ....	29
Figure 2.13. Tester l'appartenance d'un point au triangle ....	30
Figure 2.14. Création des positions intermédiaires. ....	31
Figure 2.15. Enveloppe de la sphère d'outil. ....	32
Figure 2.16. Cellules en chevauchement avec l'enveloppe. ....	32
Figure 2.17. Pas d'intersections entre droite et outil. ....	33
Figure 2.18. Intersection entre droite et outil hémisphérique. ....	34
Figure 2.19. Intersection entre droite et outil cylindrique. ....	34
Figure 2.20. Mise à jour des droites. ....	35
Figure 2.21. Calcul des écarts. ....	35
Figure 2.22. Diagramme de cas d'utilisation général. ....	36
Figure 2.23. Diagramme d'activité général. ....	37

Figure 2.24. Diagramme de cas d'utilisation « Gérer le modèle STL » .....	37
Figure 2.25. Diagramme d'activité « Gérer le modèle STL ». .....	38
Figure 2.26. Diagramme de cas d'utilisation « Créer les Dexels ». .....	38
Figure 2.27. Diagramme d'activité « Créer les Dexels ».....	39
Figure 2.28. Diagramme de cas d'utilisation « Simuler l'usinage ».....	39
Figure 2.29. Diagramme d'activité « Simuler l'usinage ».....	40
Figure 2.30. Diagramme de cas d'utilisation « Vérifier le trajet » .....	40
Figure 2.31. Diagramme d'activité « Calculer les erreurs ». .....	41
Figure 2.32. Diagramme de classes. ....	41
Figure 2.33. Classe « Modèle STL ». ....	42
Figure 2.34. Classe « Cellule ». ....	43
Figure 2.35. Classe « sommet ». ....	43
Figure 2.36. Classe « triangle ». ....	43
Figure 2.37. Classe « Dixel ». ....	44
Figure 2.38. Classe « droite ». ....	44
Figure 2.39. Classe « brute » .....	45
Figure 2.40. Classe « Outil ». ....	45
Figure 2.41. Classe « Déplacement ». ....	45
Figure 2.42. Classe « Trajet ». ....	46
Figure 2.43. Classe « Indice Droite » .....	46
Figure 2.44. Classe « Normal » .....	46

### CHAPITRE 3

Figure 3.1. Fenêtres principales de la plateforme logicielle. ....	51
--	----

Figure 3.2. Lancement du logiciel développé « Simulation Usinage 3 Axes ».....	51
Figure 3.3. Onglets du module logiciel développé .....	52
Figure 3.4. Onglet « Gérer le modèle STL ». ....	53
Figure 3.5. Onglet « Créer les Dexels ».....	54
Figure 3.6. Onglet « Simuler l’usinage ».....	55
Figure 3.7. Onglet « Vérifier le trajet ». ....	56
Figure 3.8. Premier modèle STL.....	57
Figure 3.9. Points du trajet d’outils.....	58
Figure 3.10. Résultat de la simulation d’usinage pour différents pas.....	59
Figure 3.11. Zones usinées. ....	60
Figure 3.12. Deuxième modèle STL.....	60
Figure 3.13. Points du trajet d’outil .....	61
Figure 3.14. Résultat de la simulation de l’usinage. ....	62
Figure 3.15. Points intermédiaires et trajets .....	62
Figure 3.16. Zones usinées et zones non usinées.....	63
Figure 3.17. Points intermédiaires et trajet. ....	63
Figure 3.18. Zones non usinées .....	64

# INTRODUCTION GENERALE

## Présentation du sujet :

Aujourd'hui, l'utilisation de l'outil informatique devient de plus en plus très importante dans la résolution des problèmes complexes car il offre l'efficacité, la fiabilité et la rapidité dans le traitement.

Parmi les domaines qui exigent l'utilisation de cet outil, on trouve le domaine de l'industrie, tel que l'industrie automobile, l'industrie aéronautique, l'industrie mécanique, ...etc.

Plusieurs industries ont été contraintes à améliorer et à augmenter la qualité de leurs produits. Particulièrement, l'industrie mécanique qui cherche à réduire le temps et le coût de fabrication, tout en améliorant la quantité et la qualité des produits et les conditions de travail, en offrant aux utilisateurs des ressources intégrées réduisant les tâches itératives et coûteuses.

Pour cela, l'industrie mécanique s'appuie sur la puissance de la CFAO « Conception et Fabrication Assistées par Ordinateur », qui apporte la flexibilité et la souplesse dans la conception et la fabrication des pièces de complexités diverses, comme les pièces de formes gauches. Le processus de réalisation de ces formes est un processus industriel, qui a subi de grandes évolutions ces dernières décennies par l'introduction de la CFAO et de la commande numérique.

Les pièces de formes libres (moules, matrices... etc.) sont très rencontrées dans notre vie quotidienne. Ces pièces sont conçues dans le but d'assurer des fonctions inscrites dans le cahier des charges. Ces pièces ne peuvent être usinées que sur des fraiseuses à commande numérique multiaxes (de 03-axes à 05-axes) en raison de leurs géométries très complexes.

## Problématique

L'obtention de la forme finale de la pièce de formes libres requiert le passage par trois phases d'usinage : ébauchage qui permet d'enlever le maximum de matière, demi-finition où on s'approche de la forme finale, et finition où on obtient la forme voulue. Chacune des phases nécessite la détermination du trajet d'outils par la considération de plusieurs paramètres (outils, stratégies d'usinage, conditions de coupe, problèmes d'usinage, cinématique de la machine, etc.).

La simulation d'usinage est une étape obligatoire avant de passer à l'usinage réel sur machine. Elle permet de vérifier la trajectoire d'outils, de détecter les collisions et de prédire les efforts de coupe et la topographie de la surface finie. Ils existent plusieurs techniques de simulation : solide – CSG, point vecteur, Z-buffer, etc.

Elle peut être une simulation géométrique ou bien une simulation physique. Comme la simulation physique est basée sur la simulation géométrique, trois modèles de représentation géométrique doivent être considérés : modèle de la pièce, modèle de l'outil et modèle du trajet d'outil. Ces modèles peuvent être continus ou discrets (points, triangles « STL », volumes, ...etc.). Les modèles volumiques tels que « Voxel », « Dixel », « Triple-Dixel », etc. sont largement utilisés dans le processus de simulation d'enlèvement de matière.

## Objectif

Ce travail s'insère dans le cadre de développement d'une plateforme logicielle graphique et interactive sous Windows pour la production des pièces de formes complexes initié par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) de la division Productique et Robotique (DPR) du Centre de Développement des Technologies Avancées (CDTA).

Dans ce projet nous nous intéressons à la simulation virtuelle de l'usinage des pièces complexes, définies par leurs modèles STL, sur des fraiseuses numériques à 03-axes. Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle de production des pièces complexes développée par l'équipe CFAO du CDTA, un module logiciel graphique et interactif permettant de simuler l'enlèvement de matière pour toutes les opérations d'usinage (ébauchage, demi-finition et finition) des pièces complexes sur des fraiseuses à 03-axes. Lors de la simulation, le trajet d'outil est généré avec n'importe quelle stratégie d'usinage (Plans Parallèles, Z-Constant, Isoparamétrique, Tréfilage, Contours Décalés, ...etc.) et l'outil utilisé peut-être cylindrique, hémisphérique ou torique.

## Structuration du mémoire :

Le présent mémoire est composé des chapitres suivants :

- Le premier chapitre est consacré à la présentation du processus de CFAO et au format d'échange de données « STL ». Un état de l'art sur les différentes méthodes de simulation est également présenté dans ce chapitre.
- Le deuxième chapitre est réservé à l'étude conceptuelle de notre application logicielle.
- Le dernier chapitre présente l'application logicielle développée, les tests et la validation des résultats.

Ce mémoire sera clôturé par une conclusion générale et des perspectives.

# **Chapitre 1**

---

**Étude**

**bibliographique**

## Introduction

Avec l'évolution de l'industrie dans les domaines énergétique, aéronautique, automobile et autres, les pièces de base sont devenues de plus en plus de formes complexes et leur production demandent de plus en plus des équipements spéciaux, complexes et chers. Cette évolution devait être aussi suivie par une évolution des machines-outils, des outils et des appareils et méthodes de mesure. D'autre part, le processus de réalisation de ces pièces a largement été modifié ces dernières années par l'utilisation des nouvelles techniques comme l'usinage à grande vitesse et la maquette numérique c'est-à-dire la modélisation complète du produit et de son procédé dans un système de CFAO. L'adéquation entre la forme usinée et la forme attendue, dépend de l'aptitude de chacune des activités du processus de conception et de fabrication à modéliser ou produire la géométrie attendue [1].

Dans ce chapitre, nous commençons par introduire le processus de fabrication des pièces de formes complexes. Puis, nous présentons l'outil CFAO ainsi que le format d'échange de données STL. Par la suite, un état de l'art sur les différentes méthodes de simulation.

### 1. Processus de fabrication des pièces de formes complexes :

En général, le processus est défini comme un ensemble de techniques visant l'obtention d'une pièce. L'obtention de la pièce désirée nécessite parfois l'utilisation successive de différents procédés de fabrication [3]. Le processus de fabrication des pièces complexes se découple en une activité de conception et une activité de fabrication.

Ce processus commence par une idée initiale de la pièce. Puis la conception assistée par ordinateur (CAO) pour obtenir un modèle géométrique et l'envoyer à la fabrication assistée par ordinateur (FAO). Ce dernier délivre un fichier exécutable par la machine d'usinage. Ce fichier contient les points de la trajectoire empruntée par l'outil, les tâches d'usinage et les différents changements d'outils nécessaires pour l'obtention de la pièce conçue. Finalement, la pièce est usinée sur des machines-outils à commande numérique (MOCN) (Figure 1.1) [2].

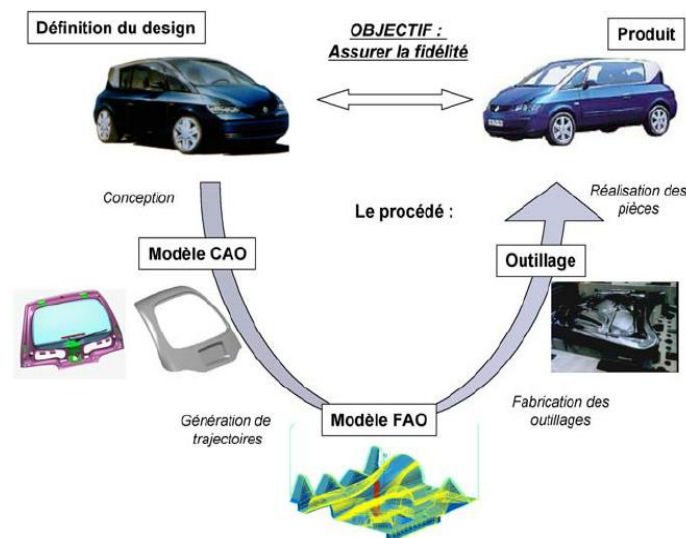


Figure 1.1. Processus de réalisation des formes complexes.



## 2. Conception et fabrication assistées par ordinateur (CFAO) :

La conception et la fabrication assistées par ordinateur (CFAO) sont deux processus virtuels CAO (Conception Assistée par Ordinateur) et FAO (Fabrication Assistée par Ordinateur) qui ont révolutionné les procédés de fabrication. La géométrie finale d'une pièce usinée peut être prédite grâce aux outils numériques de conception et de fabrication qui permettent de simuler le processus de coupe après définition de la pièce brute, de l'outil, du mouvement relatif entre la pièce et l'outil (trajectoire) et des conditions de coupe [2].

### 2.1. Conception assistée par ordinateur (CAO) :

La conception assistée par ordinateur « CAO » comprend l'ensemble des logiciels et des techniques de modélisation géométrique permettant de concevoir, de tester virtuellement à l'aide d'un ordinateur et des techniques de simulation numérique et de réaliser des produits manufacturés et les outils pour les fabriquer. Les surfaces en CAO sont de deux types [4].

- **Modèle continu** : il est basé sur les équations mathématiques des différentes entités géométriques [5].
- **Modèle discret** : l'objet est défini par des entités géométriques simples [5].

#### 2.1.1. Méthodes de représentation des surfaces

➤ **Méthodes de représentation des surfaces du modèle continu** : la plupart des surfaces usinées sont des surfaces complexes (surfaces gauches). En CAO, les méthodes de représentation des surfaces sont classées en deux catégories : surface non paramétrique et surface paramétrique [6].

- **Surface non paramétrique** : elle est de deux formes différentes :

*Formes explicites* : elle est définie par une fonction de deux variables [6] :

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

*Formes implicites* : elle est définie par une fonction de trois variables [6] :

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

- **Surface paramétrique** : elles sont très utilisées dans la modélisation. Elles sont définies par un ensemble de trois fonctions réelles  $X()$ ,  $Y()$  et  $Z()$ . Ces fonctions dépendent de deux paramètres  $u$  et  $v$ . Une surface paramétrique est donnée [6] :

$$F(u, v) = (X(u, v), Y(u, v), Z(u, v)) \quad (3)$$

Les paramètres  $u$  et  $v$  appartiennent à l'intervalle  $[0,1]$ .

➤ **Méthodes de représentation des surfaces du modèle discret** : dans le modèle discret les surfaces sont représentées par des éléments géométriques simples tels que points, triangles, tétraèdres ... etc. [5].

## 2.2. Format d'échange de données

C'est la convention qui permet d'échanger des données entre divers programmes ou logiciels par l'intermédiaire d'un fichier. Ils sont utilisés dans le cas où la CAO et la FAO ne font pas partie du même environnement pour assurer la transmission de l'information entre ces deux plateformes et garantir la compréhensibilité (Figure 1.2). Parmi ces formats, **STEP** (Standard d'Echange et Transfert), **IGES** (Initial Graphical Exchange System), **STL** (Standard Tassellation Language).

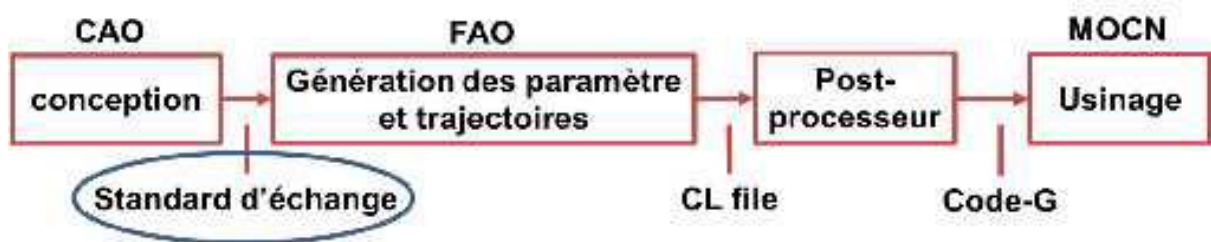


Figure 1.2. Echange de données entre deux systèmes.

➤ **Format de fichier STL (Standard Tessellation Language)** : le format de fichier STL est un format de fichier natif du logiciel CAO créé par 3D System. Il est pris en charge par de nombreux logiciels. Il est largement utilisé dans le prototypage rapide et pour la fabrication assistée par ordinateur. Le fichier STL ne décrit que la géométrie des surfaces d'un objet en 03 dimensions par un ensemble de triangles (Figure 1.3) et (Figure 1.4) [5].

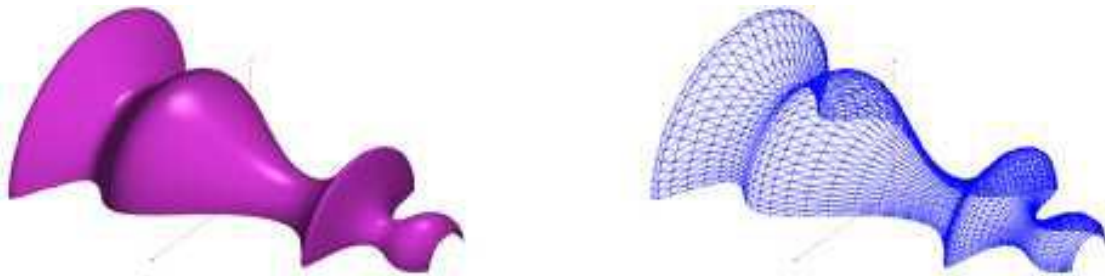


Figure 1.3. Modèle STL.

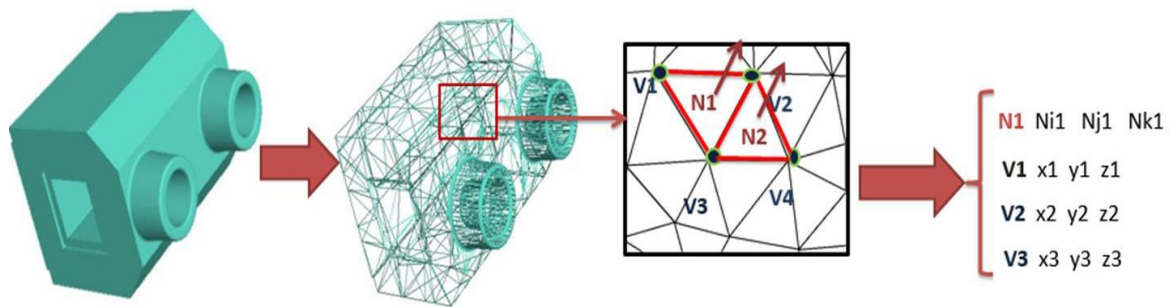


Figure 1.4. Représentation CAO représentation STL et paramètres d'un triangle.

➤ **Stockage des informations dans un fichier STL :** les informations des triangles sont stockées avec deux manières : codage ASCII et codage binaire. Pour les deux codages, les coordonnées des sommets et les composantes du vecteur normal unitaire au triangle sont stockées mais de deux manières différentes.

- **Format STL ASCII :** il est le plus descriptif et lisible mais de grande taille. Le fichier ASCII commence par la ligne « SolidName » où Name est une chaîne de caractère. Chaque triangle est défini par les coordonnées de ses sommets et les composantes de son vecteur normal unitaire. Il se termine par « endsolid Name » (Figure 1.5) [5].

- **Format STL BINAIRE :** il a moins de taille mais non lisible et trop condensé (Figure 1.5). Son entête se compose de 84 octets. Les premiers 80 octets sont utilisés pour obtenir des informations sur le fichier, et les 4 derniers octets représentent le nombre de facettes triangulaires. 50 octets sont utilisés pour représenter les composantes x, y et z de la normale pour chaque facette. 4 octets sont utilisés pour chaque coordonnée x, y et z de chaque sommet du triangle (48 octets par facette). Les deux derniers octets ne sont pas utilisés [5].

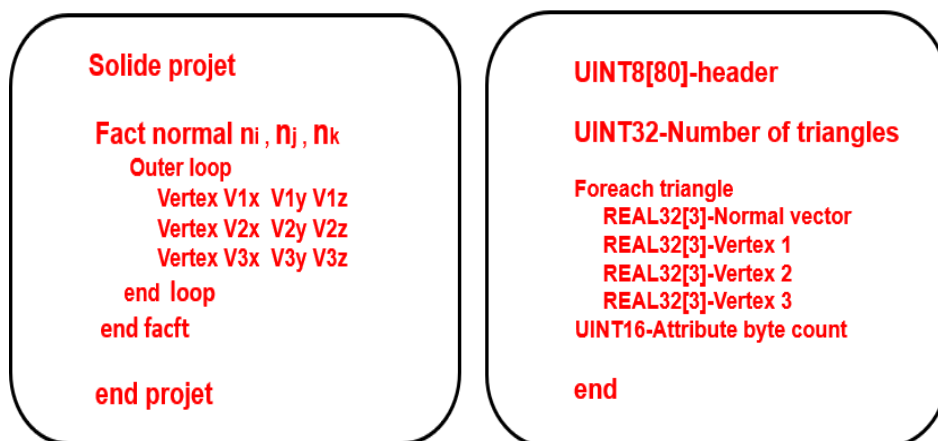


Figure 1.5. Formats STL ASCII et BINAIRE.

**Avantages et inconvénients du format STL :**

**Avantages :**

- Format de fichier compact et simple.

- Compatible avec de nombreux logiciels de modelage 3D.
- Très répandu dans l'industrie des prototypes [8].

**Inconvénients :**

- Supprime les informations de connectivité.
- Erreurs topologiques pour une géométrie complexe.
- Fichier volumineux [6].
- Le support des couleurs n'est pas standard et n'est souvent pas possible.
- Les faibles rayons de courbures exigent un grand nombre de triangles [5].

**2.3. Fabrication assistée par ordinateur (FAO) :**

Le but de la fabrication assistée par ordinateur « FAO » est de générer le fichier contenant le programme de pilotage d'une machine-outil à commande numérique. Ce fichier va décrire exactement les mouvements que doit exécuter la machine-outil pour réaliser la pièce demandée. Ce type de fichiers est appelé « G-Code ».

**2.3.1. Etapes de production d'une pièce :**

Afin de transformer une pièce brute de son état initial à son état final (pièce finie), il faut passer par trois opérations (Figure 1.6) :

- **Ebauchage** : elle permet d'enlever le maximum de matière en un minimum de temps [6].
- **Demi-finition** : préparer une finition précise en corrigeant les défauts géométriques en laissant une surépaisseur constante et faible pour la finition [6].
- **Finition** : obtention d'une bonne qualité de surface (dimensions, forme et rugosité) dans les tolérances permises par le dessin de définition [6].



a. Ebauchage

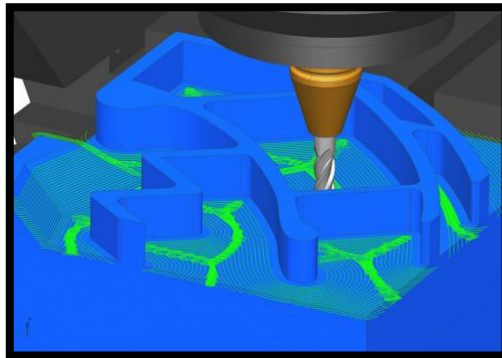
b. Demi finition

c. Finition.

**Figure 1.6.** Phases d'usinage.

### 2.3.2. Stratégie d'usinage :

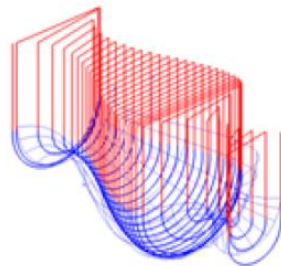
Une stratégie d'usinage est définie comme une technique pour usiner une forme, ou encore le mouvement de l'outil dans une pièce. Cette fonction dans le système FAO utilise deux outils : la MOCN et la forme de l'outil disponible. Avec ces deux informations, l'utilisateur sélectionne la forme à usiner et le système FAO utilise un algorithme de génération de trajectoire pour obtenir le résultat final (Figure 1.7) [9]. Les stratégies d'usinage permettent de calculer les trajectoires d'outil.



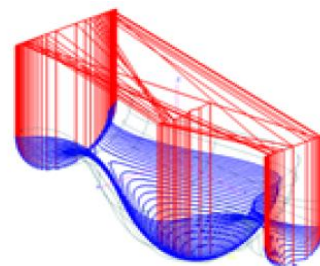
**Figure 1.7.** Stratégie d'usinage.

Les stratégies les plus utilisées pour la finition des surfaces gauches sont les suivantes :

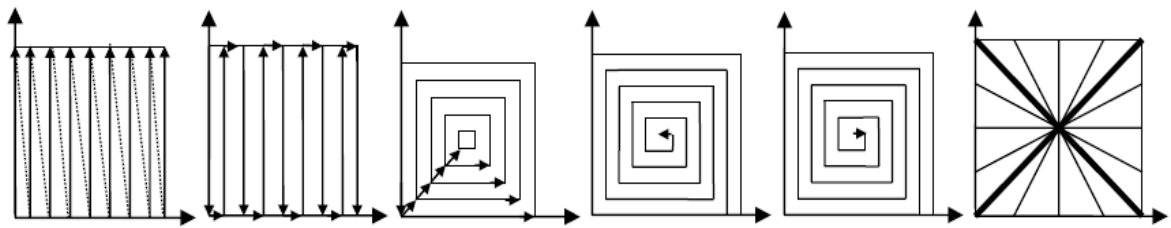
- **Plans parallèles** : elle s'appuie sur les trajectoires d'outil résultantes de l'intersection entre la surface à usiner et un ensemble de plans verticaux parallèles (Figure 1.8.a).
- **Z-Constant** : elle s'appuie sur les trajectoires d'outil résultantes de l'intersection entre la surface à usiner et un ensemble de plans horizontaux (Figure 1.8.b).
- **Isoparamétrique** : elle s'appuie sur le plan paramétrique de la surface pour générer le trajet d'outil avec plusieurs modes (Figure 1.8.c).



a. Plans verticaux parallèles



b. Z- Constant.



c. Mode de balayage Isoparamétrique.

**Figure 1.8.** Stratégies les plus utilisées pour la finition.

### 2.3.3. Outil de coupe :

L'outil de coupe (fraise) munie de dents est mis en rotation et taille la matière suite à sa rotation et au mouvement relatif de la fraise par rapport à la pièce. La forme de la fraise est variable. Elle peut être cylindrique, torique, hémisphérique ou quelquefois de forme plus complexe. Les caractéristiques physiques de la fraise (vitesse de rotation, avance) dépendent de la matière à usiner et de la profondeur de passe.

✓ **Type d'outils de coupe :** les outils de coupe sont définis par une enveloppe de type cylindrique, torique ou hémisphérique.

- **Fraise cylindrique :** elle convient aux travaux d'ébauche et de demi-finition grâce à leurs faces planes (Figure 1.9.a).
- **Fraise hémisphérique :** elle est très utilisée vu sa simplicité et sa géométrie et elle convient pour la finition de toutes les matières (Figure 1.9.b).
- **Fraise torique :** elle constitue un compromis entre la fraise cylindrique et la fraise hémisphérique (Figure 1.9.c) [11].



a. Fraise cylindrique

b. Fraise hémisphérique.

c. Fraise torique.

**Figure 1.9.** Formes de fraises.

✓ **Positionnement de l'outil :** le positionnement d'outil sur une surface gauche peut être décomposé en deux étapes : la définition du point de contact outil/pièce et ensuite l'orientation de l'axe de l'outil. Quel que soit les formes et les dimensions des outils utilisés, ils doivent être constamment tangents aux surfaces en chaque point de contact. Ce positionnement dépend du vecteur normal unitaire en ce point, de l'orientation de l'axe de l'outil ainsi que de la forme et des dimensions de l'outil (Figure 1.10) [12].

Les différentes positions d'outil hémisphérique sont données par :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n}. \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{n} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u}. \end{cases} \quad (4)$$

Les différentes positions d'outil cylindrique sont données par :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|}. \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} = \overrightarrow{OC_C} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|}. \end{cases} \quad (5)$$

Les différentes positions d'outil torique sont données par :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|}. \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|}. \end{cases} \quad (6)$$

Avec :

$$\vec{k} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|}. \quad (7)$$

$C_C$ : le point de contact entre l'outil et la surface

$C_E$ : le point centre de l'outil

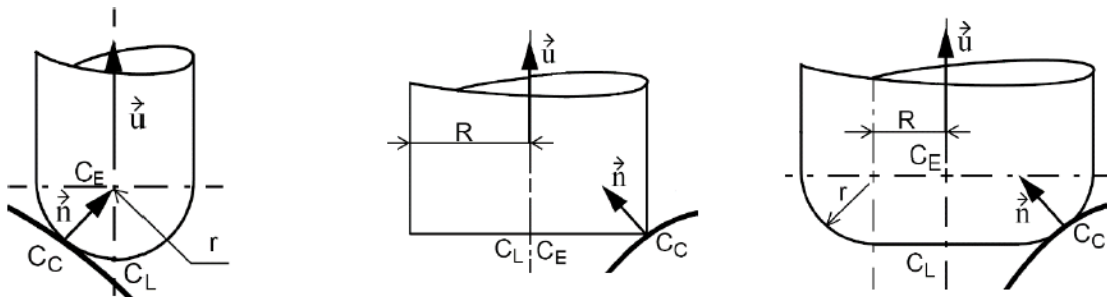
$C_L$ : le point extrémité de l'outil

$\vec{n}$ : Vecteur normal unitaire à la surface au point de contact

$\vec{u}$ : Vecteur orientation de l'axe de l'outil

$r$  : le rayon de l'outil hémisphérique et le petit rayon de l'outil torique

$R$  : le grand rayon de l'outil torique et le rayon de l'outil cylindrique [12].



a. Outil hémisphérique.

b. Outil cylindrique.

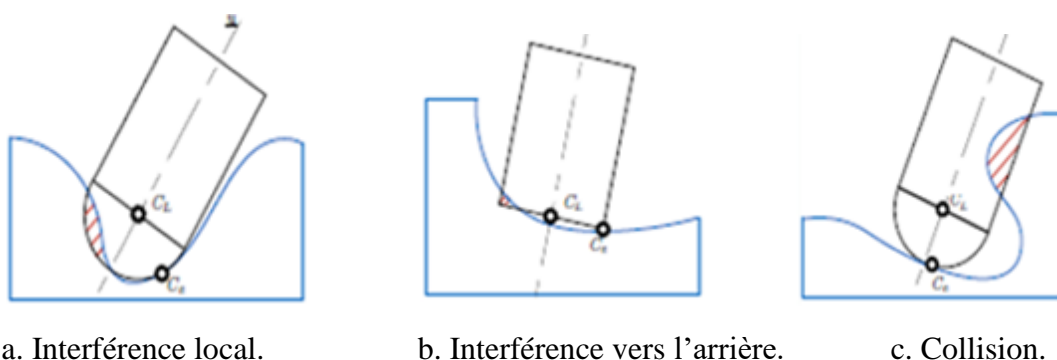
c. Outil torique.

**Figure 1.10.** Positionnement des différentes formes d'outil.

### 2.3.4. Problèmes de l'usinage

Les formes géométriques complexes des surfaces gauches peuvent générer des problèmes d'interférences et de collisions. Pour un point de contact outil pièce donné, toute la difficulté consiste à déterminer l'orientation efficace de l'axe de l'outil. Donc, la maîtrise des interférences outil/pièce lors de la génération de trajectoires d'outil est une condition nécessaire pour garantir la conformité de la pièce en termes de géométrie. Les interférences sont en général classées selon trois types.

- **Interférences locales** : elles traduisent un enlèvement de matière excessif par la partie active de l'outil sur la surface à usiner (Figure 1.11.a).
- **Interférences vers l'arrière** : elles sont des pénétrations intempestives de l'arrière de l'outil dans la surface à usiner (Figure 1.11.b).
- **Collisions** : les interférences globales (collisions) représentent des collisions entre l'ensemble {corps d'outil, porte outil, broche} et l'ensemble {pièce, porte Pièce, éléments de structure...} (Figure 1.11.c) [6].



a. Interférence local.

b. Interférence vers l'arrière.

c. Collision.

**Figure 1.11.** Problèmes de l'usinage.

### 2.3.5. Machine-Outil à Commande Numérique (MOCN)

Une MOCN est une machine-outil partiellement ou totalement automatisée. Les ordres de mouvements des différents organes sont donnés par programmation. Les positions successives de l'outil par rapport à la pièce sont exprimées sous forme numérique et sont définies dans un système de coordonnées normalisé.

### 2.3.6. Fraisage

C'est un procédé de fabrication par enlèvement de matière. En fraisage, l'enlèvement de Matière « copeaux » résulte de la combinaison de deux mouvements : rotation de l'outil de coupe d'une part (mouvement de coupe «  $M_c$  ») et avance rectiligne de la pièce à usiner (Mouvement d'avance «  $M_a$  »). Trois modes de fraisage peuvent être distingués fraisage en bout, fraisage en roulant et fraisage combiné. Le fraisage en bout est le plus répandu car il



permet de réaliser des surfaces complexes. Le fraisage en bout peut être réalisé sur des machines-outils à commande numérique (MOCN) 03-axes ou 05-axes [5].

### 2.3.6.1. Fraiseuse

Une fraiseuse est une machine-outil utilisée pour usiner des pièces mécaniques. Elle est équipée d'un outil coupant appelé une fraise. La fraiseuse est capable d'enlever de la matière sur une pièce initiale afin de la transformer en une autre pièce [5].

### 2.3.6.2. Différents types de fraiseuses

Nous pouvons classer les fraiseuses de la manière suivante : fraiseuses conventionnelles et fraiseuses à commande numérique. La fraiseuse à commande numérique dispose d'un organe de contrôle informatique (automate programmable ou base PC) où sont enregistrés les différents ordres nécessaires à la fabrication d'une pièce. Lorsque le programme est lancé, la machine effectue automatiquement toutes les opérations.

- **Fraiseuse à 3 axes** : une fraiseuse à 03-axes comporte une broche soit horizontale soit verticale ou universelle, et c'est à la table d'être positionnée en fonction du but recherché sur la pièce (Figure 1.12) [12]. Les axes de la machine sont définis de la manière suivante : l'axe Z est parallèle à la broche principale de la machine, l'axe X correspond au plus grand déplacement horizontal tandis que l'axe Y forme le trièdre direct avec les axes X et Z [5].



Figure 1.12. Fraiseuse horizontale.

## 3. Simulation d'usinage :

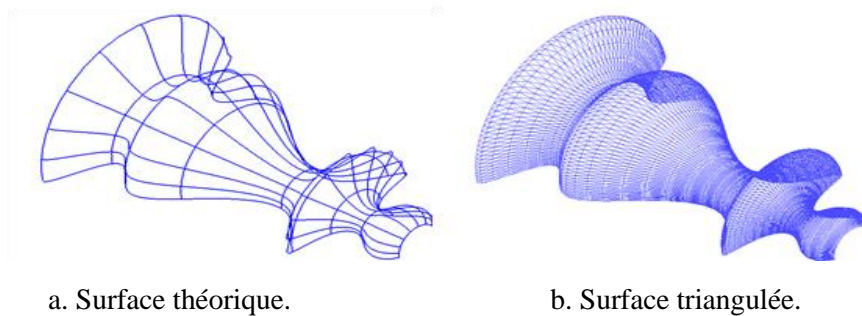
La simulation d'usinage est un moyen très important permettant de réduire le temps d'usinage réel et elle sert à vérifier les trajets d'usinage avant l'usinage réel. Sans simulation, les trajets d'outils sont examinés à plusieurs reprises sur des pièces modèles et les corrections sont effectuées au fur et à mesure. L'essai sur des machines réelles exige d'utiliser des machines chères avec des risques d'erreurs très importants ainsi que le temps d'usinage [1].

### 3.1. Méthodes d'approximation des surfaces

Pour réaliser la simulation d'usinage, nous avons besoin de trouver une méthode de représentation des objets pour simplifier les calculs [16]. Parmi ces méthodes on trouve :

#### 3.1.1. Triangulation :

C'est la plus simple car elle approxime la surface théorique par un ensemble de triangles avec une certaine précision (tolérance) spécifiée au départ. Plus la précision est importante, plus l'approximation de la surface est bonne (Figure 1.13). Pour cette méthode, les sommets des triangles générés appartiennent à la surface théorique.



a. Surface théorique.

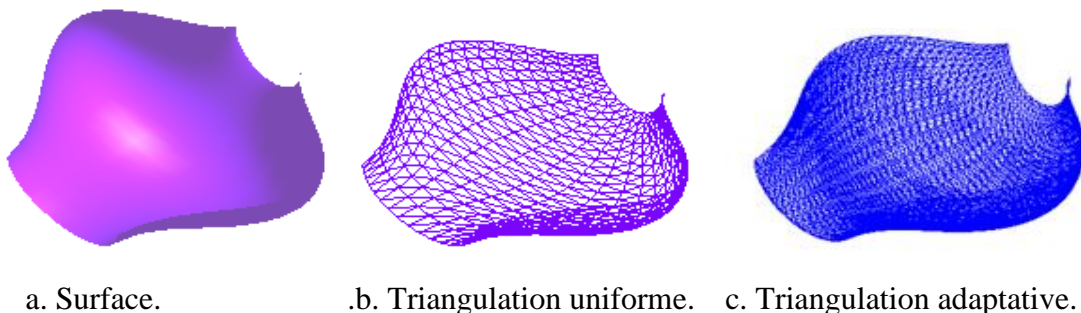
b. Surface triangulée.

**Figure 1.13.** Approximation d'une surface par des triangles.

#### 3.1.1.1. Types de triangulation :

**a. Triangulation uniforme :** le nombre de triangles est fixé suivant chaque direction dans le plan paramétrique et ensuite, les triangles sont générés dans l'espace (Figure 1.14.b).

**b. Triangulation adaptative :** les triangles sont créés d'une manière dynamique et adaptative en fonction de la précision exigée par l'utilisateur (Figure 1.14.c).



a. Surface.

.b. Triangulation uniforme.

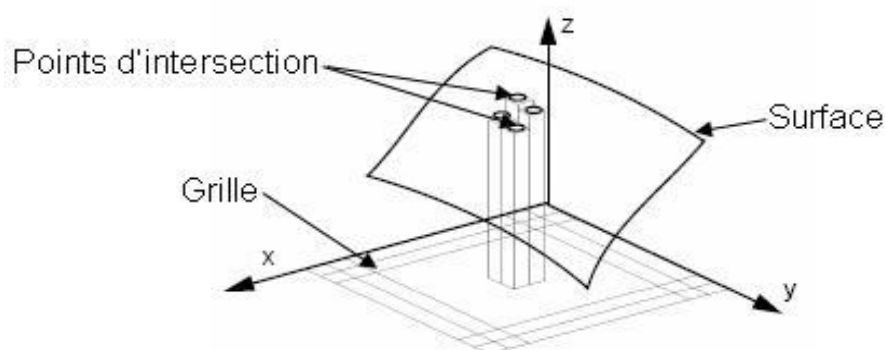
c. Triangulation adaptative.

**Figure 1.14.** Types de triangulations.

#### 3.1.2. Z-buffer :

Cette méthode est différente et performante. Elle consiste à projeter une grille selon une direction donnée sur la surface, suivant le point de vue choisi. La construction de la surface est obtenue par intersection entre un ensemble de droites parallèles à Z et la surface. Les

surfaces les plus adaptées ont une fonction qui réalise une bijection entre le plan de base XY et les points de la surface. Pour chaque droite, on cherche toutes les intersections avec l'ensemble des surfaces, et on retient l'intersection la plus haute appartenant à la peau de la pièce (Figure 1.15). Cette méthode est très simple à mettre en œuvre mais sa précision est en fonction du pas de la grille. Comme la grille est indépendante de l'orientation de la pièce, cette méthode est peu précise dans le cas d'approximation de parois ayant un angle de dépouille faible, ou verticale, c'est-à-dire parallèle à la direction de projection.



**Figure 1.15.** Méthode du Z-buffer.

### 3.2. Simulation et vérification et correction :

Pour la détection et l'élimination des erreurs, trois aspects distincts sont utilisés : simulation, vérification et correction.

Dans la phase de simulation, le modèle géométrique de la pièce est modifié et le modelage du volume balayé de chaque mouvement d'outil doit être fait. La vérification de l'exactitude de l'opération d'usinage exige une comparaison du modèle géométrique de la pièce obtenue avec un modèle géométrique de la pièce désirée. Les systèmes de simulation doivent contenir une information pour spécifier la tolérance désirée permettant la comparaison directe des dimensions des surfaces usinées et des surfaces théoriques avec la tolérance spécifiée. Finalement, les erreurs détectées doivent être corrigées.

#### 3.2.1. Simulation :

Le mot « simulation » vient du verbe simuler, qui veut dire faire apparaître comme réel. La simulation du fraisage consiste à reproduire toutes les actions de fraisage : le mouvement de l'outil, l'enlèvement de la matière et détection des erreurs qui peuvent être produites lors de l'usinage. Puisque l'opération d'usinage prend beaucoup de temps, surtout pour les grandes pièces, et même pour ne pas abuser de la matière, qui est très coûteuse, on simule l'opération d'usinage pour détecter les erreurs puis les corriger. Cette opération est effectuée en FAO avant de passer à l'usinage effectif (Figure 1.16). Donc, l'objectif de la simulation est de comparer la surface théorique avec la surface usinée afin de localiser les erreurs [19].

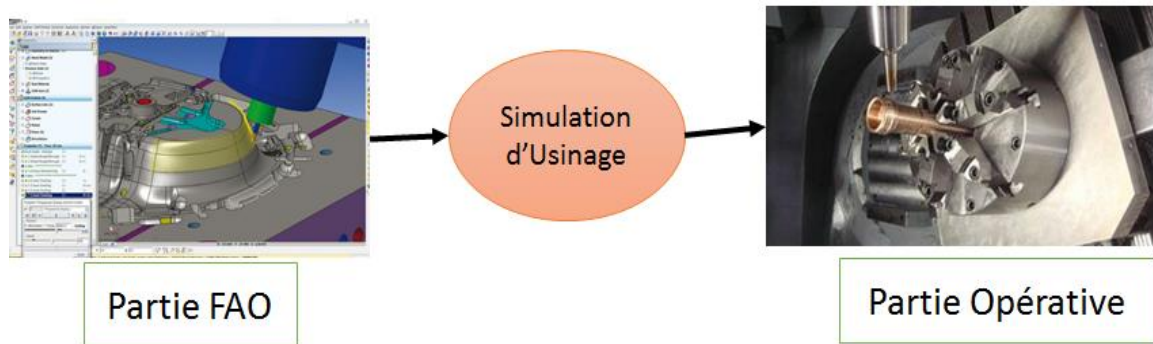


Figure 1.16. Simulation d'usinage entre FAO et machine.

### 3.3. Catégorie de simulation

La simulation de l'usinage est divisée en deux catégories : simulation géométrique et simulation physique (Figure 1.17) [18].

#### 3.3.1. Simulation géométrique

Elle est utilisée pour vérifier graphiquement les interférences et les collisions, le respect des tolérances exigées dans le cahier des charges par le constructeur.

#### 3.3.2. Simulation physique :

Comme son nom l'indique, la simulation physique d'un processus d'usinage vise à révéler les aspects physiques d'un processus d'usinage, comme l'effort de coupe, les vibrations, la rugosité des surfaces, la température d'usinage et l'usure de l'outil. Elle est basée sur la simulation géométrique et le choix du matériau de l'outil de coupe.

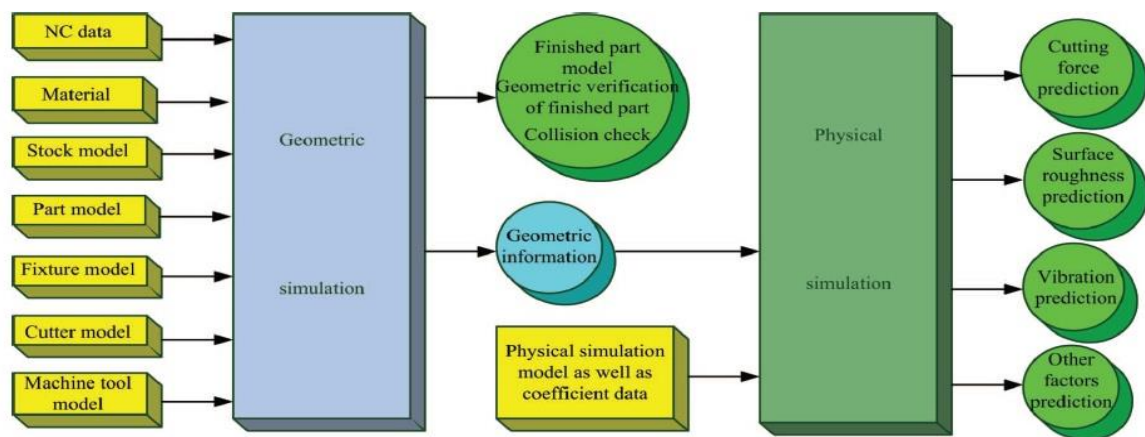


Figure 1.17. Architecture générale de la simulation de l'usinage.

### 3.4. Echelles de simulation :

L'approche multi-échelles aide à effectuer l'étude de l'usinage. Ceci permet de séparer les difficultés en limitant le nombre de phénomènes à prendre en compte et la taille du

modèle à une échelle donnée. On distingue trois échelles d'analyse : échelle humaine, échelle macroscopique et l'échelle microscopique.

#### **3.4.1. Echelle humaine :**

C'est une simulation globale de l'environnement de l'usinage. Afin de s'imprégner du comportement du moyen de production (mouvements des axes, position de la pièce sur la table et encombrement de la zone de travail). Donc elle permet de détecter d'éventuelles collisions pendant le processus. Cette méthode est indispensable quand le moyen de production est très complexe et induit des mouvements relatifs de la pièce par rapport à l'outil difficiles à anticiper. Généralement ce type de simulation est intégré dans les logiciels de FAO [19].

#### **3.4.2. Echelle macroscopique**

Elle permet de déterminer globalement le volume de matière enlevé pour chacune des phases d'usinage de la pièce. Les techniques de simulation à cette échelle permettent, entre autres, de visualiser et d'anticiper les défauts de surface liés purement à la stratégie programmée ou à la cinématique de la machine [19].

#### **3.4.3. Echelle microscopique :**

C'est l'étude des matériaux pour déduire certaines propriétés à partir de la structure fine du matériau. Ces propriétés sont, par exemple, les lois de comportement du matériau utilisé. A cette échelle, la description de la formation du copeau est étudiée [19].

### **3.5. Modèles de la simulation :**

Nous introduisons principalement, deux modèles à savoir : modèle dynamique et modèle géométrique.

#### **3.5.1. Modèle dynamique :**

Peut-être soit un modèle simple masse ressort ou un modèle complexe élément finis. Ce dernier permet une discrétisation spatiale beaucoup plus fine et plus souple. Elle permet aussi d'obtenir des modes de vibrations beaucoup plus réalistes et d'aborder le cas où la pièce et/ou l'outil sont déformables dans la zone de travail [19].

#### **3.5.2. Modèle géométrique :**

Trois modèles de représentation géométrique sont à considérer à savoir : modèles de la pièce, modèle de l'enveloppe de l'outil et modèle du volume balayé par l'outil. Les modèles géométriques peuvent aller du plus simple : une série de points, au plus complexe : description avec des facettes ou volumique. Nous distinguons quatre grandes familles de modèles : filaire, à base solide, à base objet et à base espace image. Dans ce dernier, se

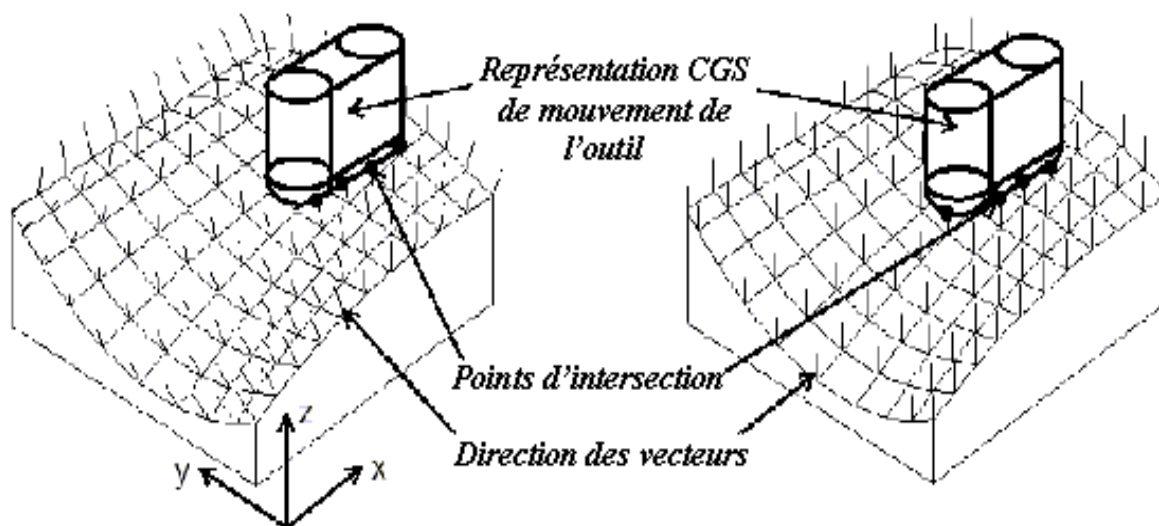
retrouve les techniques de représentation volumique telles que le Dixel, le Voxel et le Triple-Dixel [19].

### 3.6. Méthodes de simulation :

#### 3.6.1. Technique point-vecteur :

Elle consiste à découper la surface selon des méthodes spécifiques pour avoir un ensemble de points. Pour chaque point on associe un vecteur qui a comme limites la surface nominale et le brut. Ces vecteurs peuvent être orientés de deux manières (Figure 1.18).

- Suivant la normale à la surface (exacte) : dans ce cas, chaque vecteur est indépendant linéairement des autres vecteurs.
- Suivant l'axe Z de l'outil (simplifiée) : dans ce cas, tous les vecteurs sont parallèles à l'axe Z.



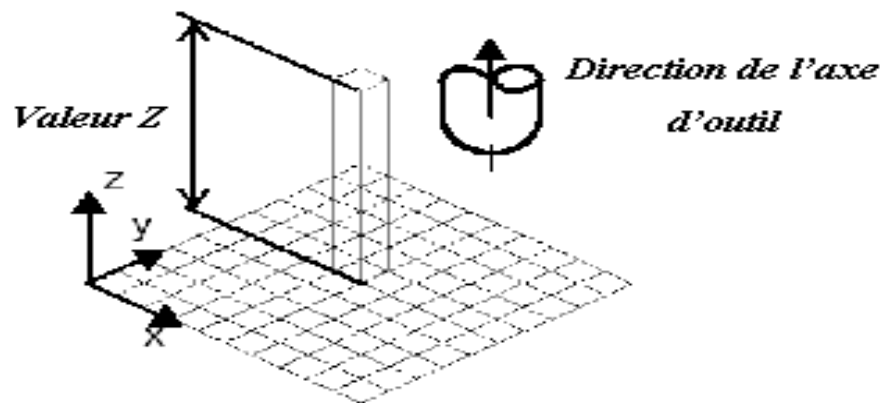
**Figure 1.18.** Orientation des vecteurs suivant la normale et l'axe de l'outil.

Cette technique a pour but de calculer la distance entre la surface usinée et la surface théorique. A partir d'un point de la surface, on construit une droite dans une direction donnée, et on cherche toutes les intersections de cette droite avec les trajets élémentaires de l'usinage. Ensuite, on calcule la distance entre chaque point d'intersection et la surface théorique. La plus petite distance est celle qui laisse le moins de matière avec une erreur minimale. Un ensemble « point-vecteur » est construit sur la surface, tel que les points appartiennent à la surface théorique et l'extrémité des vecteurs appartient à la surface brute de la pièce. Le calcul se résume ainsi au calcul de l'intersection entre un segment et l'enveloppe d'outil [17]. Pour détecter les erreurs comme les collisions, les interférences et les zones non usinées, il suffit de voir la direction des vecteurs (positive : zone non usinée, négative : usinage plus que voulu) et leurs longueurs (s'ils ne sont pas dans les tolérances d'usinage, il faut passer à la correction) [16].

### 3.6.2. Technique du Z-buffer :

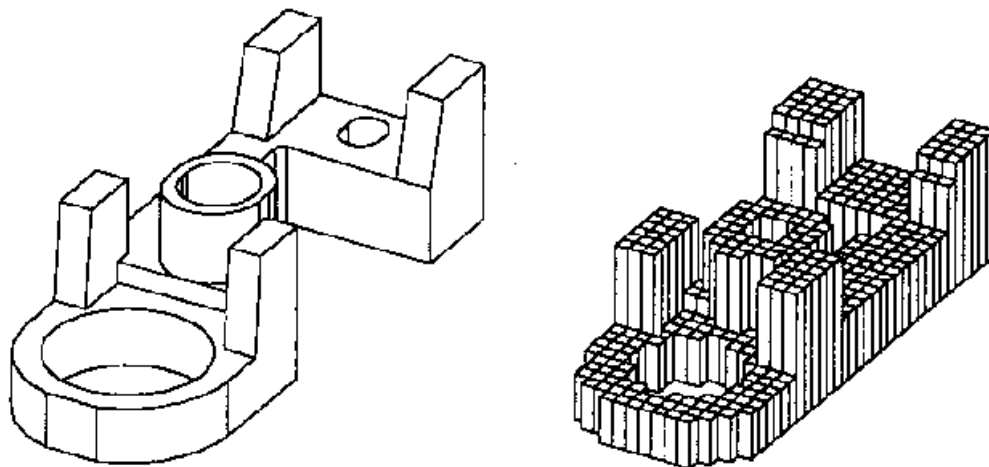
C'est une représentation de la surface sous forme d'une matrice ou d'une grille dont les lignes et les colonnes représentent un quadrillage géométrique dans le plan x y, puis associer un segment vertical d'une altitude z initiale (Figure 1.19). Lors de l'usinage, on calcule l'intersection des segments avec le trajet d'usinage. Pour chaque segment, on retient l'altitude d'intersection la plus basse. On obtient une représentation Z-buffer du trajet d'usinage. Les erreurs de simulation proviennent du pas de la grille et de la modélisation du trajet [16].

Pour mener les calculs demandés lors de la simulation, il suffit de réaliser un Z-buffer de la surface finale, du brut, et de la surface usinée. Enfin, soustraire les différents Z-buffer entre eux pour détecter les interférences, les collisions, les zones non usinées les volumes de matière enlevée [17].



**Figure 1.19.** Principe de la méthode du Z-Buffer.

Plus les carreaux de la grille sont plus petits on obtient des résultats plus exacts, donc l'opération devient coûteuse en temps à cause des opérations de calcul [16].



**Figure 1.20.** Représentation d'une pièce par le Z-Buffer.

### 3.6.3. Technique des Voxels :

Le Voxel (contraction de « volume » et « élément ») est un pixel en 3D (Figure 1.21). Il consiste à stocker une information colorimétrique avec ses coordonnées spatiales, voire temporelles, de positionnement ainsi que, facultativement, une taille relative à l'unité utilisée ou d'autres informations telles qu'une matière [4].

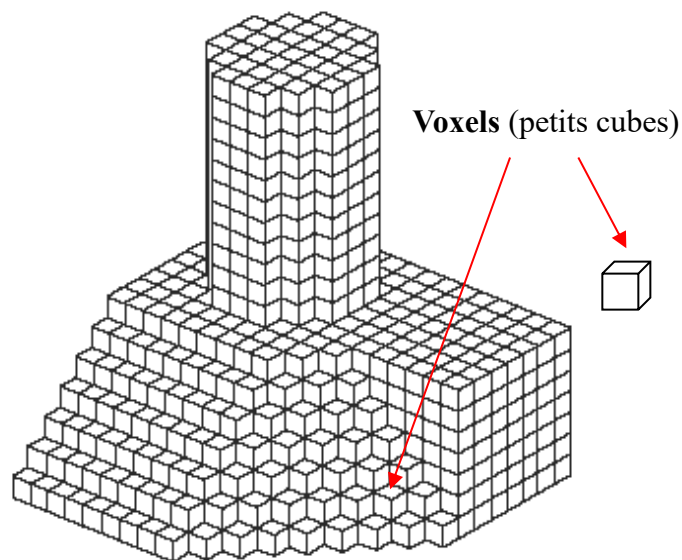


Figure 1.21. Représentation de volume par des Voxels.

### 3.6.4. Technique des Dexels

Un Dexel est un parallélépipède caractérisé par sa section représentée par la forme de la cellule, sa hauteur représentée par la longueur du segment de type « matière » (Figure 1.22). Ces derniers forment les Dexels de la pièce dans la direction de la droite de la cellule. A partir de la section et de la longueur d'un Dexel, le volume du Dexel est calculé. Le volume total de l'objet est la somme des volumes de tous les Dexels. Le modèle de l'objet en Triple-Dexels est obtenu en combinant les Dexels créés dans les trois directions [4].

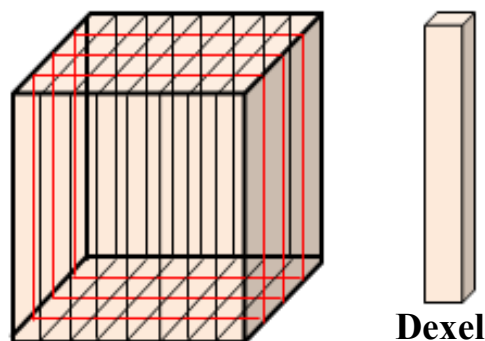


Figure 1.22. Représentation de solide par des Dexels.



## **Conclusion**

Nous avons considéré dans ce chapitre l'étude de l'état de l'art sur le processus de production des surfaces gauches en le divisant en deux parties. La première partie réservée aux généralités sur le système de CFAO ainsi que le format d'échange STL. La deuxième partie réservée à l'étude des différentes techniques de simulation de l'usinage des pièces complexes sur les fraiseuses 03 axes.

Dans notre projet, nous allons utiliser la méthode Dexcel pour réaliser la simulation d'enlèvement de matière. Dans le chapitre suivant, nous allons présenter notre conception.

# **Chapitre 2**

---

**Étude**

**conceptuelle**

## Introduction

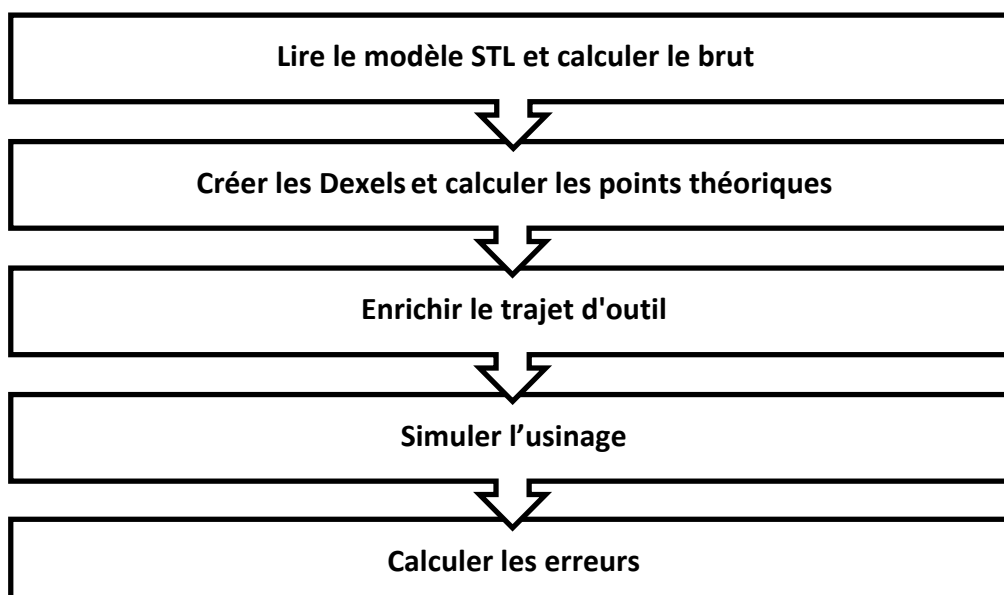
Avant de passer à l'usinage réel sur fraiseuse numérique, plusieurs types de simulations doivent être menées (dynamique, statique, mouvement, etc.). Parmi ces simulations, la simulation des différentes opérations d'enlèvement de matière pour évaluer la qualité des surfaces obtenues, le trajet d'outil, ...etc.

L'objectif de notre travail est la conception et le développement d'une application logicielle pour la simulation d'usinage des pièces complexes sur fraiseuses numériques à 03-axes. Pour cela, plusieurs problèmes doivent être résolus :

- Comment lire et structurer le fichier STL ?
- Comment lire et structurer le trajet d'outils ?
- Comment enrichir le trajet ?
- Comment approximer le modèle STL ?
- Comment mettre à jour le modèle de la pièce usinée ?
- Comment calculer l'intersection entre les différentes formes d'outils et le modèle approximé de la pièce ?
- Comment accélérer la simulation ?
- Comment vérifier la conformité des surfaces usinées ?

### 1. Solution proposée

En informatique, la résolution d'un problème passe par sa décomposition en une suite d'étapes successives pour pouvoir proposer une solution adéquate. Dans notre cas, les étapes suivies sont présentées dans l'organigramme général suivant (Figure 2.1).



**Figure 2.1.** Organigramme général de la démarche proposée.

### 1.1. Récupération des paramètres du fichier STL et création du brut

Cette étape consiste à vérifier la syntaxe du fichier STL qui décrit la pièce à usiner, à récupérer les informations des sommets et des triangles et les stocker dans un fichier texte.

Par la suite, on calcule les limites du brut qui a pour but d'englober tous les points du modèle STL dans une enveloppe parallélépipédique de faces parallèles aux axes X, Y et Z. Il s'agit de déterminer les coordonnées  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Z_{min}$  et  $Z_{max}$  et ensuite ses dimensions minimales (Longueur, Largeur et Hauteur) (Figure 2.2) :

$$\text{Longueur} = X_{max} - X_{min} \quad (1)$$

$$\text{Largeur} = Y_{max} - Y_{min} \quad (2)$$

$$\text{Hauteur} = Z_{max} - Z_{min} \quad (3)$$

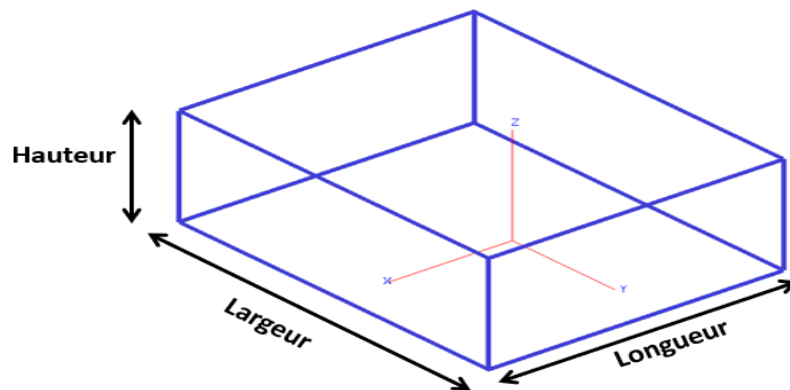


Figure 2.2. Dimensions minimales du brut.

### 1.2. Création des cellules

La création des cellules consiste en la subdivision du brut en cellules qui possèdent la même taille (blocs parallélépipédiques). Elle passe par la spécification du nombre de cellules suivant les trois axes X, Y et Z (Figure 2.3).

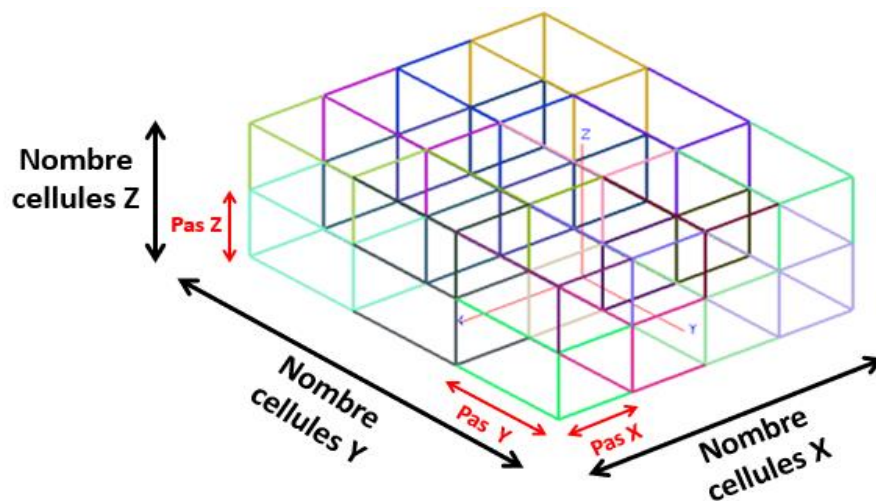


Figure 2.3. Création des Cellules.

A partir des limites du brut et du nombre de cellules, les pas et les limites de chaque cellule sont calculés par (Figure 2.4) :

$$pas_x = \text{Longueur} / \text{nombre\_cellule}_x \quad (4)$$

$$pas_y = \text{Largeur} / \text{nombre\_cellule}_y \quad (5)$$

$$pas_z = \text{Hauteur} / \text{nombre\_cellule}_z \quad (6)$$

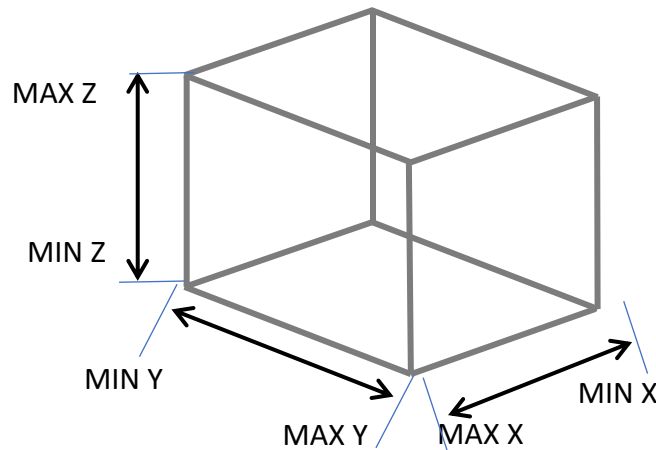


Figure 2.4. Dimensions minimales d'une cellule.

### 1.3. Affectation des sommets et des triangles aux cellules

L'affectation des sommets et des triangles aux cellules a pour but d'accélérer le calcul des points théoriques.

#### 1.3.1. Affectation des sommets

Après la création des cellules, chaque sommet du triangle doit être affecté à la cellule correspondante. Cette affectation consiste à déterminer la position de chaque sommet et puis affecter l'indice de ce dernier aux cellules (Figure 2.5).

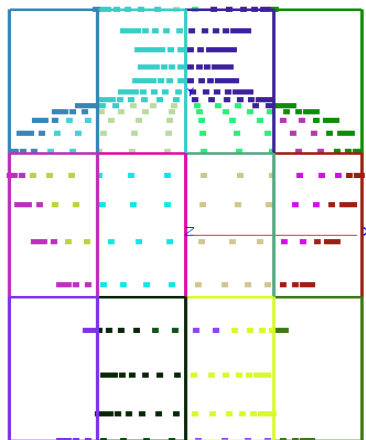


Figure 2.5. Affectation des sommets aux cellules.

Les indices de la cellule contenant un sommet de coordonnées **X**, **Y** et **Z** sont :

$$i = (x - xmin)/pasx \quad (7)$$

$$j = (y - ymin)/pasy \quad (8)$$

$$k = (z - zmin)/pasz \quad (9)$$

### 1.3.2. Affectation des triangles

Elle consiste à déterminer la cellule ou les cellules qui se chevauchent avec l'enveloppe du triangle. Ces cellules sont calculées en se basant sur les indices des cellules des trois sommets de chaque triangle (Figure 2.6).

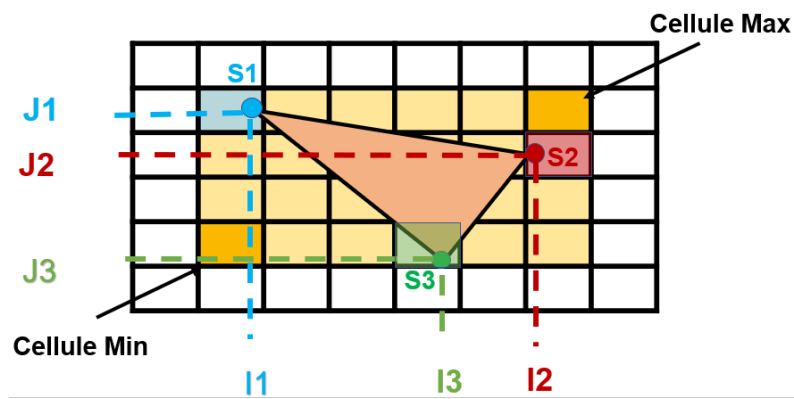


Figure 2.6. Affectation des triangles aux cellules.

## 1.4. Création des Dexels

### 1.4.1. Création des droites

Une droite est représentée par son point de début, point de fin, point ZSimulation et point ZThéorique avec (Figure 2.7) :

Point\_début : X, Y, ZMIN.

Point\_fin : X, Y, ZMAX.

ZSimulation : initialisé à ZMAX (intersection outil /droite).

ZThéorique : intersection droite/triangle.

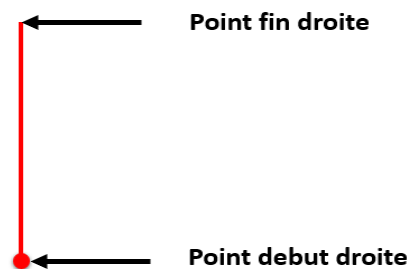


Figure 2.7. Points d'une droite.

### 1.4.2. Affectation des droites aux Dexels

A partir des pas des cellules suivant les axes X, Y et Z et les dimensions du brut (longueur, largeur et hauteur), les nombres de Dexels suivant les axes X, Y et Z sont calculés de la manière suivante :

$$\text{nombre\_dixel\_x} = \text{longueur}/\text{pas\_x} \quad (10)$$

$$\text{nombre\_dixel\_y} = \text{largeur}/\text{pas\_y} \quad (11)$$

$$\text{nombre\_dixel\_z} = \text{hauteur}/\text{pas\_z} \quad (12)$$

Ensuite, chaque Dixel est défini à partir des indices de quatre (04) droites (Figure 2.8).

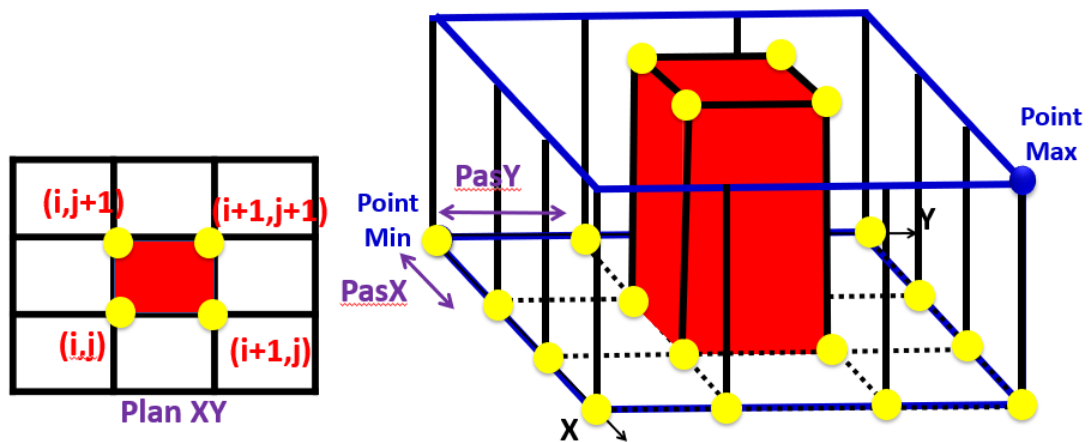


Figure 2.8. Création des Dexels.

### 1.4.3. Affectation des droites aux cellules

L'affectation des droites aux cellules a pour but d'accélérer les calculer des points théoriques et la simulation d'usinage. Après avoir créé les droites, on calcule leurs positions et on affecte l'indice de chaque droite aux cellules (Figure 2.9).

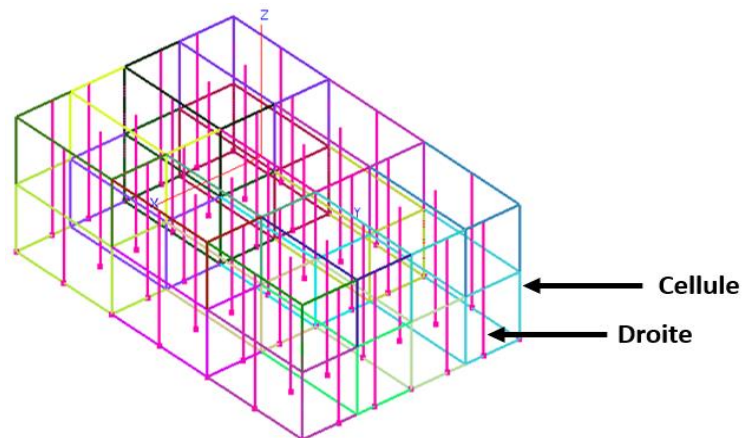
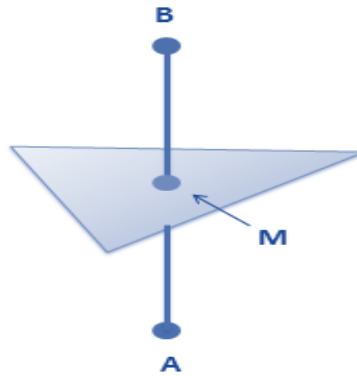


Figure 2.9. Affectation des droites aux cellules.

### 1.5. Calcul des points théoriques

Les points théoriques sont des points résultants de l'intersection des droites des Dexels avec les triangles des cellules auxquels ils appartiennent (Figure 2.10). Ces points servent à vérifier la conformité entre le modèle théorique de la pièce et les surfaces usinées résultantes.



**Figure 2.10.** Intersection droite triangle.

Pour déterminer les points théoriques il faut :

#### 1.5.1. Calculer l'enveloppe du triangle

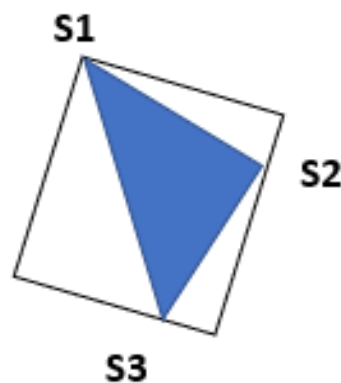
Les extrémités minimales et maximales de l'enveloppe du triangle sont données par (Figure 2.11) :

$$X_{\min} = \min (S1X, \min (S2X, S3X))$$

$$X_{\max} = \max (S1X, \max (S2X, S3X))$$

$$Y_{\min} = \min (S1Y, \min (S2Y, S3Y))$$

$$Y_{\max} = \max (S1Y, \max (S2Y, S3Y))$$



**Figure 2.11.** Enveloppe du triangle.

#### 1.5.2. Vérifier si la droite appartient à l'enveloppe

Donc pour que la droite appartienne à l'enveloppe il faut que (Figure 2.12) :



$$X_{\min} \leq X \text{ et } X_{\max} \geq X \text{ et } Y_{\min} \leq Y \text{ et } Y_{\max} \geq Y$$

Avec :

$X_{\min}$ ,  $X_{\max}$ ,  $Y_{\min}$ ,  $Y_{\max}$  sont les limites de l'enveloppe.

$X$ ,  $Y$  sont les coordonnées de la droite.

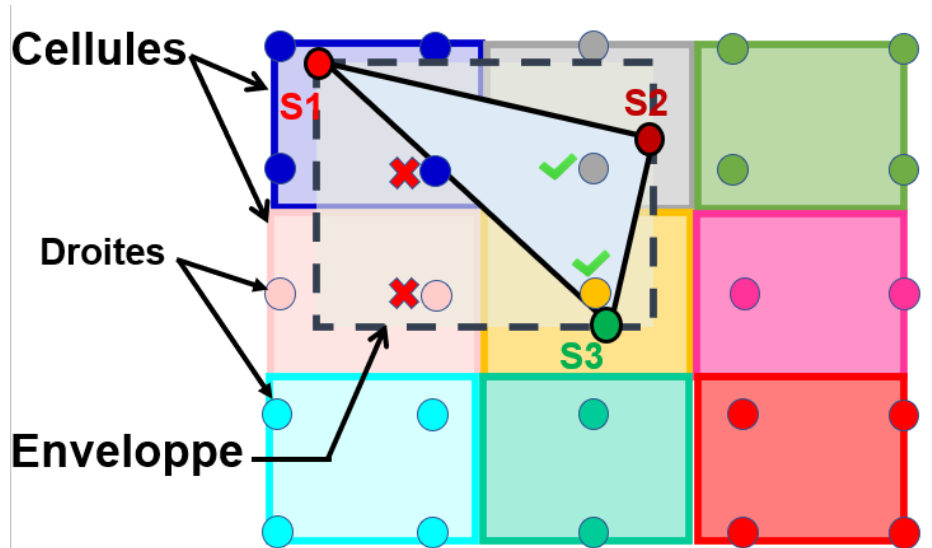


Figure 2.12. Cas d'appartenance et non appartenance d'une droite à l'enveloppe.

### 1.5.3. Calcul du point théorique

Par la suite, si la droite appartient à l'enveloppe, on calcule le point d'intersection. On a l'équation du plan qui est décrite par :

$$Ax + By + Cz + D = 0 \quad (13)$$

Où

$A$  : composante X de la normale du triangle.

$B$  : composante Y de la normale du triangle.

$C$  : composante Z de la normale du triangle.

$X$  : coordonnée X de la droite.

$Y$  : coordonnée Y de la droite.

$D$  : paramètre du plan.

Ainsi

$$Z = \frac{-D - AX - BY}{C} \quad (14)$$

Avec  $C \neq 0$

### 1.5.4. Test d'appartenance d'un point à un triangle

Le principe est de calculer la somme des aires des trois triangles partiels et la comparer à l'aire totale du triangle. L'aire totale  $S$  du triangle et les aires partielles  $S1$ ,  $S2$  et  $S3$  des trois triangles sont données par :

$$\begin{cases} P = \frac{1}{2}(AB + BC + AC) \\ P1 = \frac{1}{2}(BC + PC + BP) \\ P2 = \frac{1}{2}(AC + AP + PC) \\ P3 = \frac{1}{2}(AB + AP + BP) \end{cases} \quad (15)$$

$$\begin{cases} S = \sqrt{R * (R - AB) * (R - BC) * (R - AC)} \\ S1 = \sqrt{R1 * (R1 - BC) * (R1 - BP) * (R - PC)} \\ S2 = \sqrt{R2 * (R2 - AC) * (R2 - AP) * (R2 - PC)} \\ S3 = \sqrt{R3 * (R3 - AB) * (R3 - AP) * (R3 - BP)} \end{cases} \quad (16)$$

Avec :

$P$  : demi-périmètre.

Le point d'intersection  $P$  appartient au triangle si la condition suivante est vérifiée (Figure 2.13) :

$$S = S1 + S2 + S3 \quad (17)$$

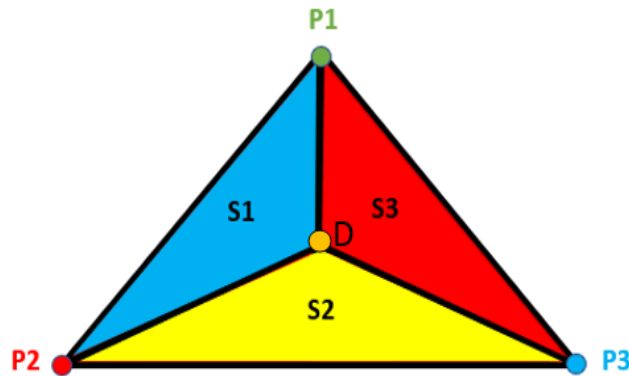


Figure 2.13. Tester l'appartenance d'un point au triangle.

## 1.6. Enrichissement du trajet d'outil

### 1.6.1. Lecture du trajet d'outil

Le fichier trajet outil (CL-File) contient des informations sur l'outil (forme, rayon, longueur), les coordonnées du point extrémité de l'outil CL et les composantes du vecteur d'orientation de l'axe de l'outil.

Cette étape consiste à vérifier le format du fichier et à créer un vecteur déplacement pour chaque deux positions successives de l'outil ayant le même outil.

### 1.6.2. Génération des positions intermédiaires

L'objectif principal de la création de positions intermédiaires pour chaque déplacement c'est pour pouvoir simuler correctement le mouvement continu de l'outil. En même temps, accélérer la simulation en évitant le calcul de l'enveloppe théorique pour chaque déplacement de l'outil. De plus, donner la possibilité de considérer toutes les formes d'outils possibles.

Pour cela, l'utilisateur doit spécifier un incrément (distance) entre chaque deux positions intermédiaires (Figure 2.14). Ces positions seront créées si la longueur du segment de déplacement est plus grande que l'incrément. La longueur du segment est donnée par :

$$distance = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2 + (Z2 - Z1)^2} \quad (18)$$

Où

**X1, Y1, Z1** sont les coordonnées du point CL de la position 1.

**X2, Y2, Z2** sont les coordonnées du point CL de la position 2.

### 1.6.3. Génération du nombre de segments

Le nombre de segments créés est donné par :

$$nombre\_segment = distance / increment \quad (19)$$

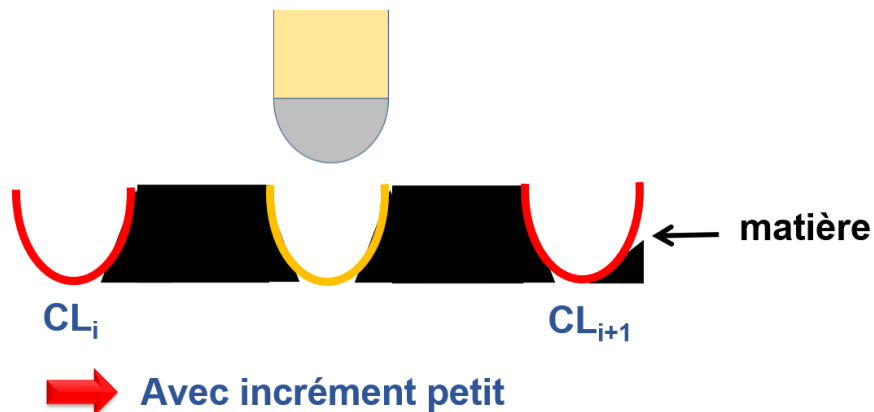


Figure 2.14. Création des positions intermédiaires.

## 1.7. Calcul des intersections

### 1.7.1. Calcul des enveloppes

C'est l'enveloppe qui englobe la sphère d'outil dans le plan XY (Figure 2.15).

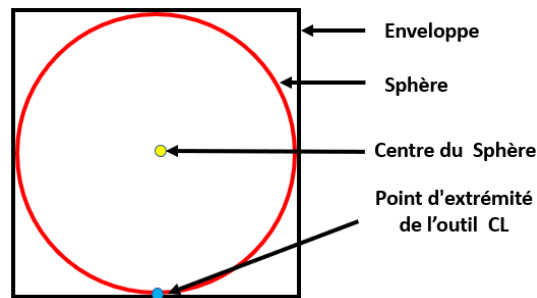


Figure 2.15. Enveloppe de la sphère d'outil.

### 1.7.2. Calcul du chevauchement de la cellule avec l'enveloppe

Après l'enveloppe d'outil, nous passons à la recherche des cellules qui se chevauchent avec l'enveloppe. Pour cela, on calcule les positions ( $i_{min}$ ,  $j_{min}$ ,  $i_{max}$ ,  $j_{max}$ ) des cellules seulement pour le premier et le troisième sommet de l'enveloppe (Figure 2.16).

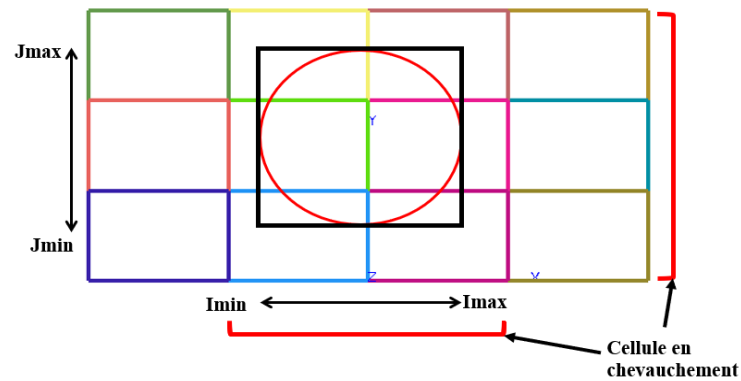
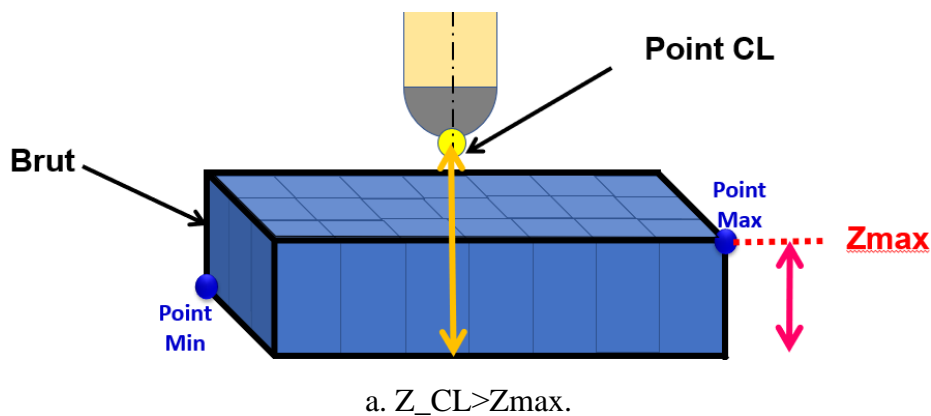


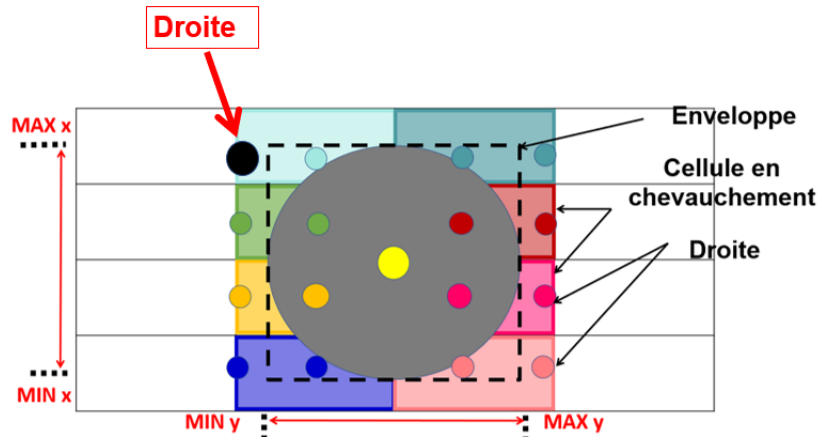
Figure 2.16. Cellules en chevauchement avec l'enveloppe.

### 1.7.3. Calcul de l'intersection

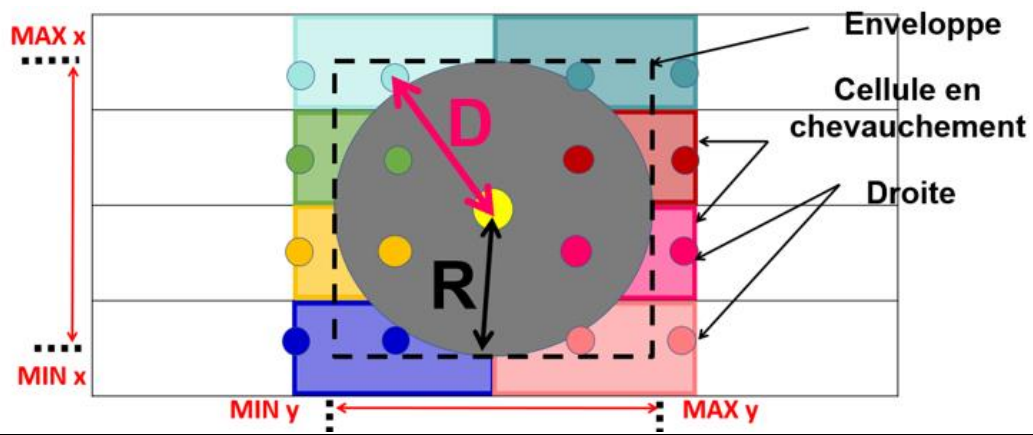
C'est l'intersection entre les droites en chevauchement avec l'enveloppe et l'outil.

**Cas où il n'y a pas d'intersections :** nous n'avons pas d'intersection pour deux cas. Le premier cas lorsque le point CL est au-dessus du brut de la pièce (Figure 2.17.a). Le deuxième cas lorsque la droite n'appartient pas à l'enveloppe de l'outil (Figure 2.17.b).





b. Droite n'appartient pas à l'enveloppe.



c. Distance entre la droite et le centre de la sphère est supérieur au rayon de la sphère.

**Figure 2.17.** Pas d'intersections entre droite et outil.

**Cas où il y a intersections :** dans ce cas, la droite est au-dessous de l'ombre de l'outil hémisphérique (Figure 2.18) ou de l'outil cylindrique (Figure 2.19).

➤ **Cas de l'outil hémisphérique :** la sphère est définie par un centre de coordonnées  $(CL_X, CL_Y, CL_Z+R)$  où  $R$  est son rayon. L'équation de cette sphère est donnée par :

$$(x - CL_X)^2 + (y - CL_Y)^2 + (z - CL_Z - R)^2 = R^2 \quad (20)$$

Pour la droite de coordonnées  $x$  et  $y$ , la coordonnée  $z$  (ZSimulation) du point d'intersection est donnée par (Figure 2.18) :

$$z = -\sqrt{-(x - CL_X)^2 - (y - CL_Y)^2 + (R)^2} + CL_Z + R \quad (21)$$

Si la distance entre le point d'intersection et le centre de la sphère est inférieure ou égale au rayon de l'outil, alors il y a intersection. Sinon, pas intersection.

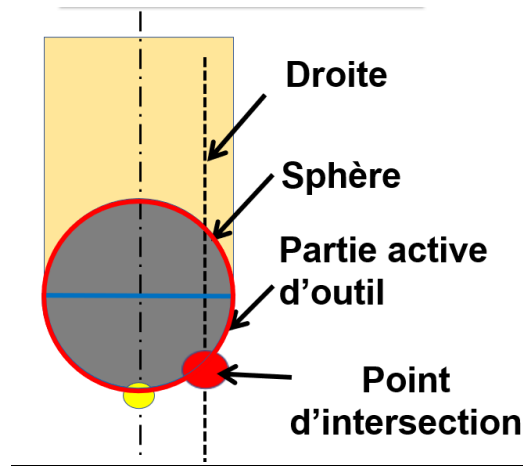


Figure 2.18. Intersection entre droite et outil hémisphérique.

➤ **Cas de l'outil cylindrique :** la coordonnée  $z$  (ZSimulation) du point d'intersection est égale à la coordonnée  $Z$  du point CL ( $Z_{Simulation} = CL_Z$ ) (Figure 2.19).

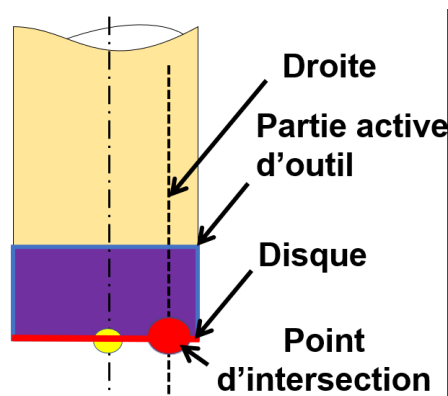
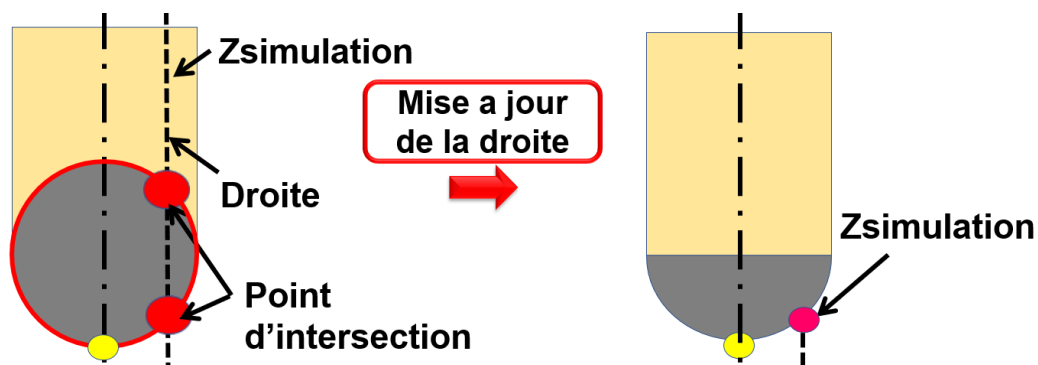


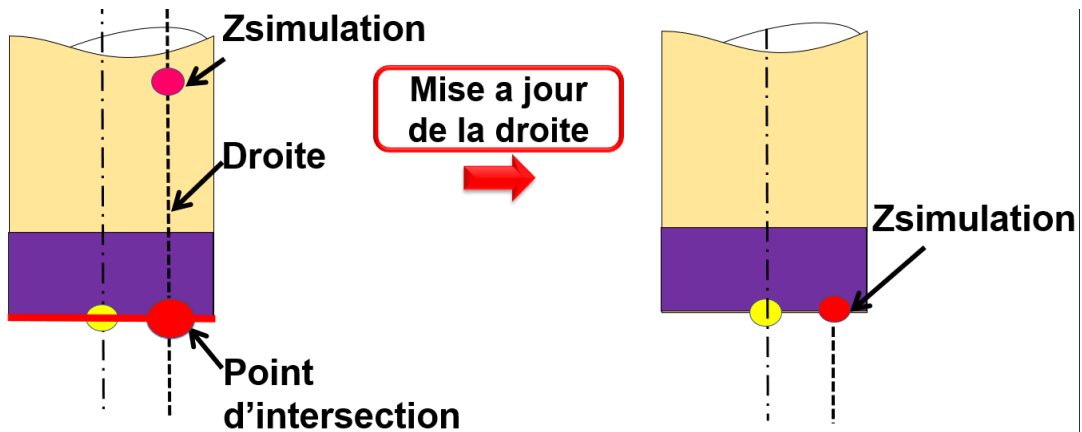
Figure 2.19. Intersection entre droite et outil cylindrique.

### 1.8. Mise à jour des droites

Après le calcul des intersections des droites avec les différents outils, il faut les mettre à jour en considérant le point d'intersection le plus bas suivant l'axe  $Z$  pour chaque droite (Figure 2.20).



a. Mise à jour pour l'outil hémisphérique.



b. Mise à jour pour l'outil cylindrique.

Figure 2.20. Mise à jour des droites.

### 1.9. Calcul des écarts

Cette étape permet de vérifier et de détecter les erreurs afin de contrôler la conformité des surfaces résultantes avec le modèle théorique. Le principe consiste à calculer les écarts supérieur et inférieur à partir des tolérances inférieure et supérieure spécifiées par l'utilisateur et le point théorique. En fonction de la position du point, nous avons trois cas (Figure 2.21) :

- Si les points de simulation sont dans l'intervalle [théorique - tolérance inférieure, théorique + tolérance supérieure], alors il n'y a pas d'erreurs.
- Si les points de simulation < [théorique + tolérance supérieure], alors il y a des erreurs et les points de simulation sont récupérables.
- Si les points de simulation > [théorique + tolérance supérieure], alors il y a des erreurs et les points de simulation sont irrécupérables.

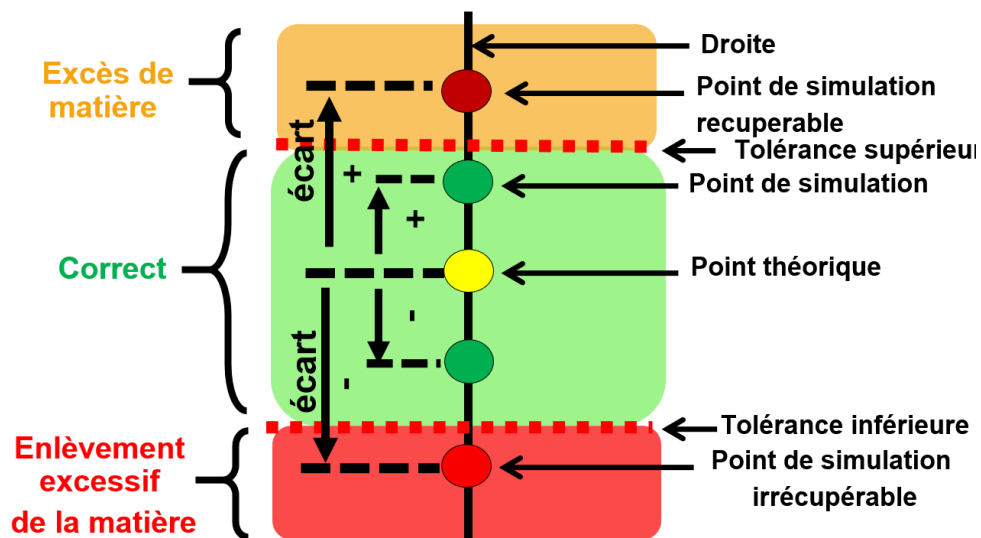


Figure 2.21. Calcul des écarts.

## 2. Modélisation UML (Unified Modeling Language).

UML est un langage visuel constitué d'un ensemble de schémas, appelés diagrammes, qui donnent chacun une vision différente du projet à traiter. Il nous fournit des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, ...etc. UML permet de :

- Donner une structuration cohérente des fonctionnalités et des données.
- Penser objets dès le départ.
- Faciliter l'analyse, la compréhension et la réduction de la complexité d'un système.

Dans ce projet, trois diagrammes sont utilisés à savoir diagramme de cas d'utilisation, diagramme d'activité et diagramme de classes.

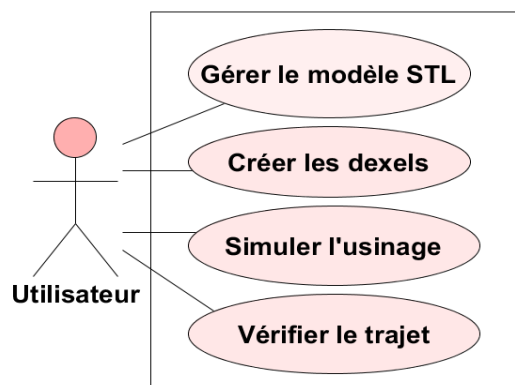
### 2.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de représenter les différentes façons dont un utilisateur peut interagir avec un système. Nous commençons par le diagramme de cas d'utilisation général qui donne une vue globale du fonctionnement du système.

### 2.2. Diagramme d'activité

Les diagrammes d'activités permettent de déterminer les traitements a priori séquentiels. Ils offrent un pouvoir d'expression très proche des langages de programmation objet : spécification des actions de base (déclaration de variables, affectation, etc.), structures de contrôle (conditionnelles, boucles) ainsi que les instructions particulières à la programmation orientée objet (appels d'opérations, exceptions, etc.). Ils sont bien adaptés à la spécification détaillée des traitements en phase de réalisation. On peut également les utiliser de façon plus informelle pour décrire des enchaînements d'actions de haut niveau, en particulier pour la description détaillée des cas d'utilisation.

**Diagramme de cas d'utilisation général** : le système est divisé en quatre grandes phases



**Figure 2.22.** Diagramme de cas d'utilisation général.



Diagramme d'activité général (Figure 2.23) :

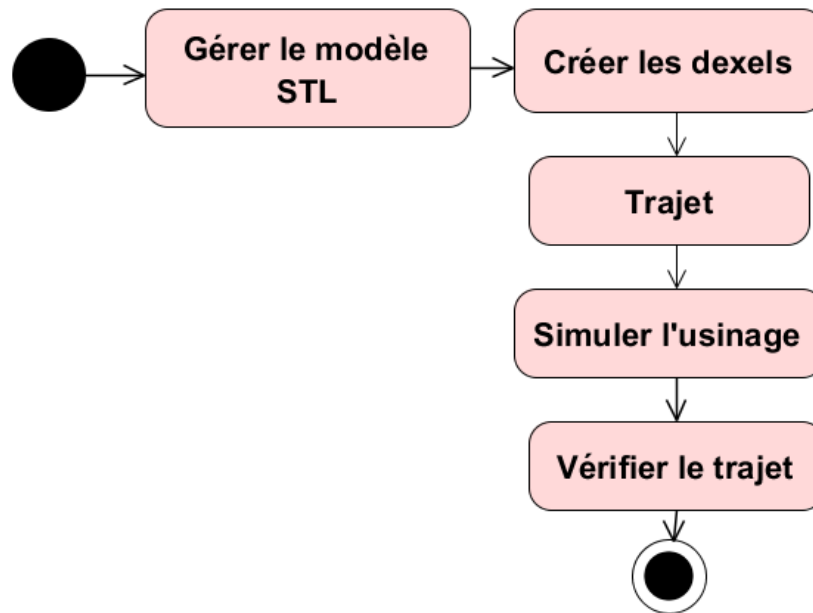


Figure 2.23. Diagramme d'activité général.

Diagramme de cas d'utilisation « Gérer le modèle STL » (Figure 2.24) :

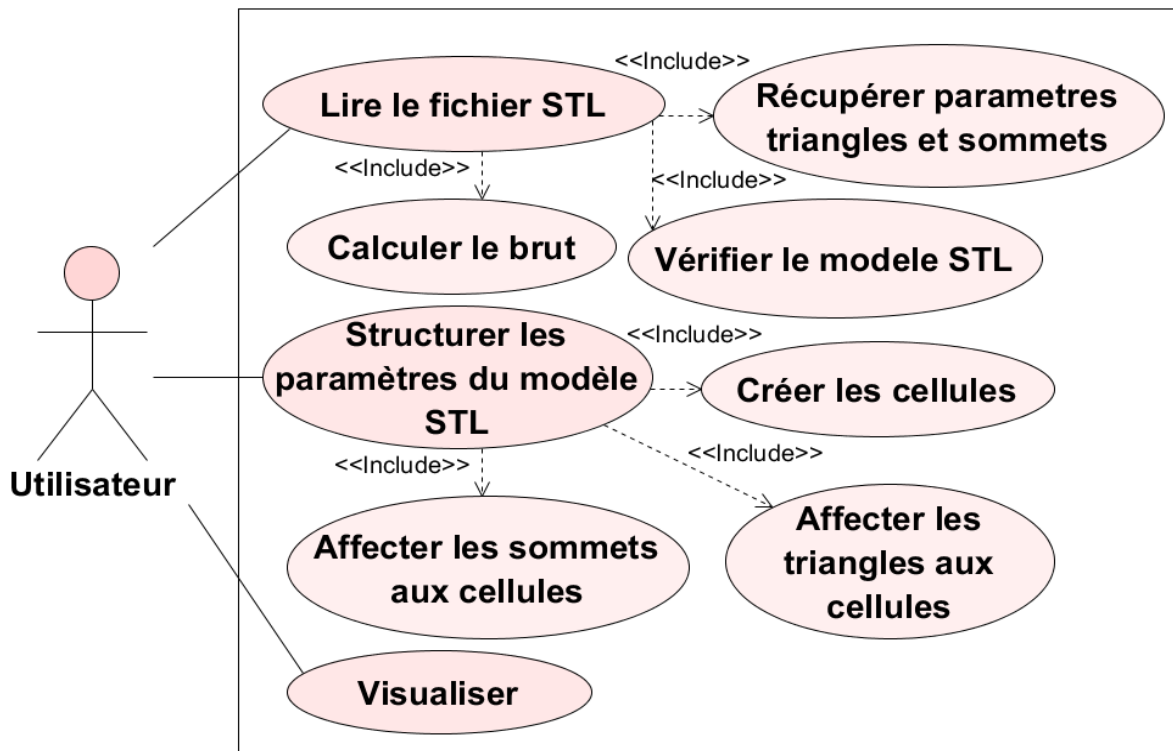


Figure 2.24. Diagramme de cas d'utilisation « Gérer le modèle STL ».

Diagramme d'activité « Gérer le modèle STL » (Figure 2.25) :

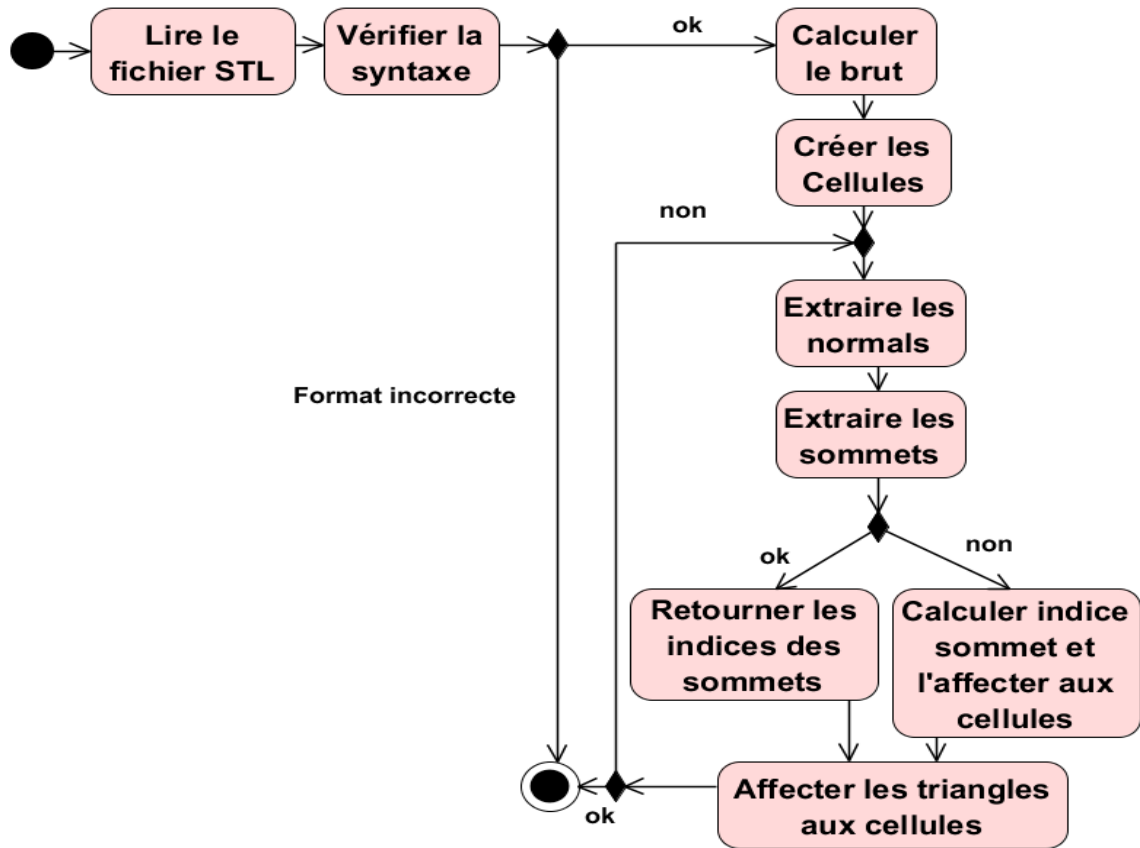


Figure 2.25. Diagramme d'activité « Gérer le modèle STL ».

Diagramme de cas d'utilisation « Créer les Dexels » (Figure 2.26) :

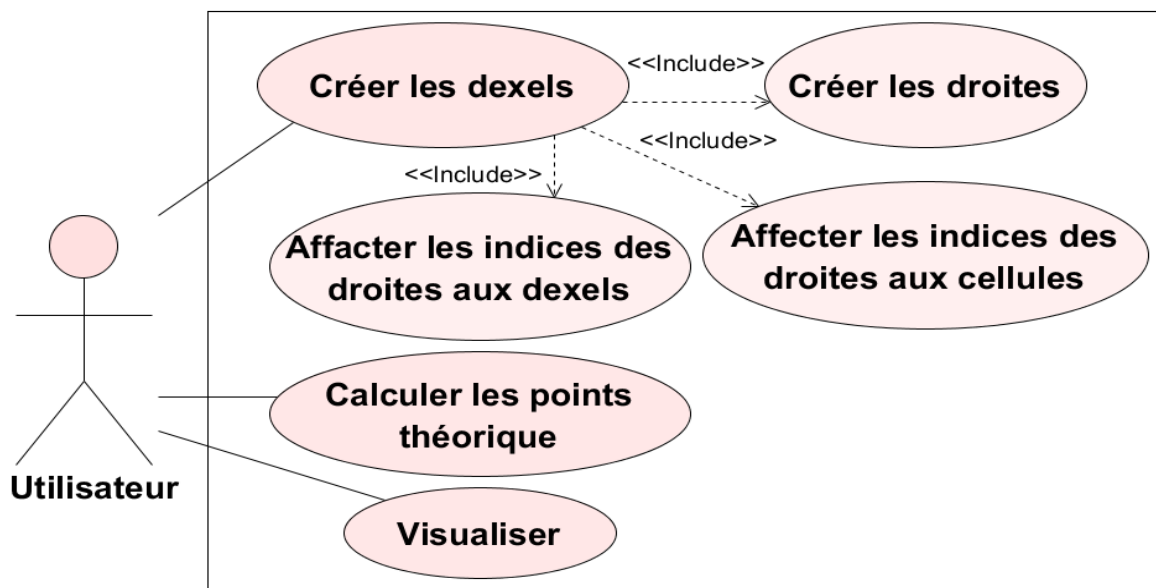


Figure 2.26. Diagramme de cas d'utilisation « Créer les Dexels ».

Diagramme d'activité « Créer les Dexels » (Figure 2.27) :

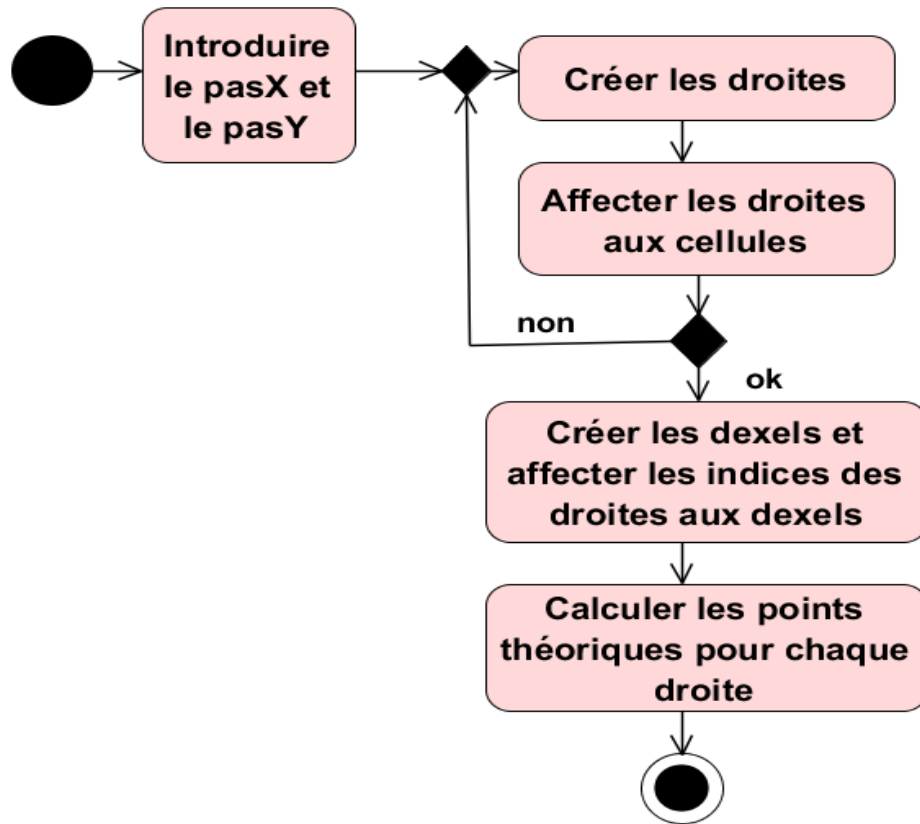


Figure 2.27. Diagramme d'activité « Créer les Dexels ».

Diagramme de cas d'utilisation « Simuler l'usinage » (Figure 2.28) :

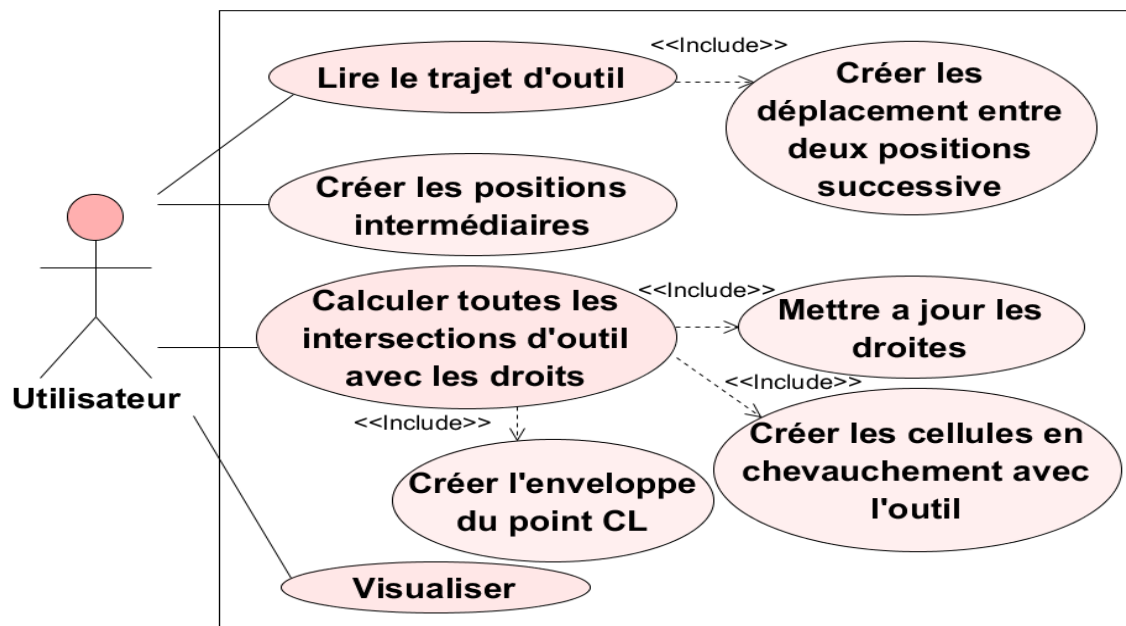


Figure 2.28. Diagramme de cas d'utilisation « Simuler l'usinage ».

Diagramme d'activité « Simuler l'usinage » (Figure 2.29) :

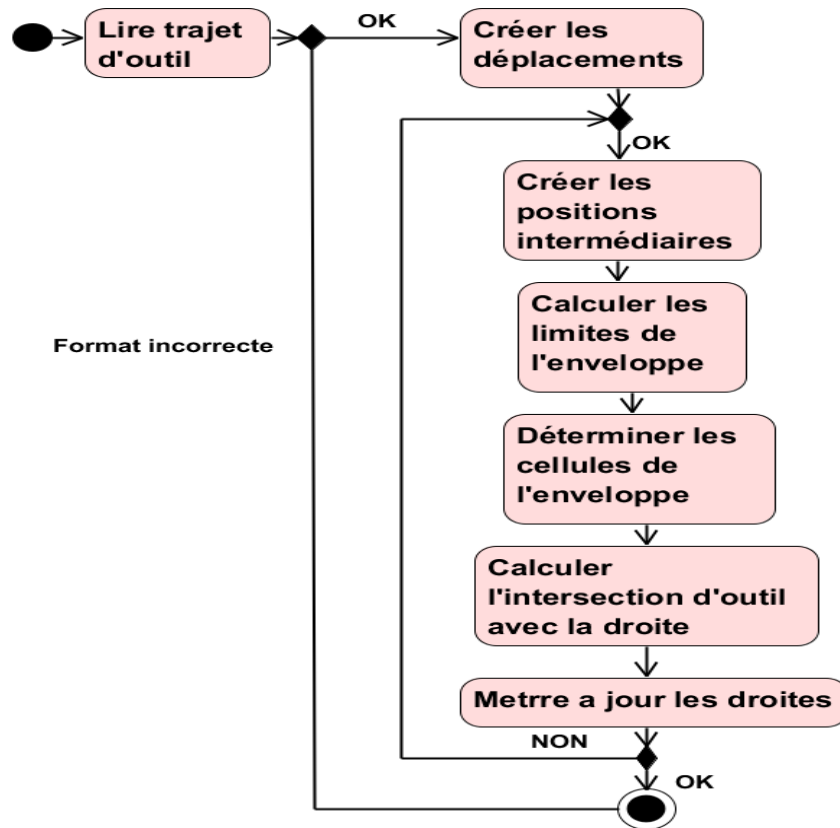


Figure 2.29. Diagramme d'activité « Simuler l'usinage ».

Diagramme de cas d'utilisation « Vérifier le trajet » (Figure 2.30) :

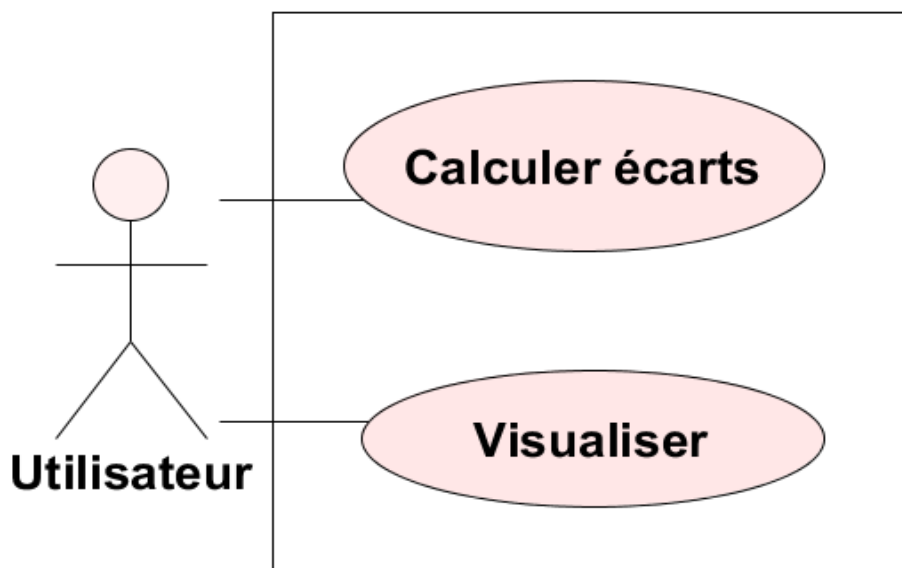


Figure 2.30. Diagramme de cas d'utilisation « Vérifier le trajet ».

Diagramme d'activité « Calculer les erreurs » (Figure 2.31) :

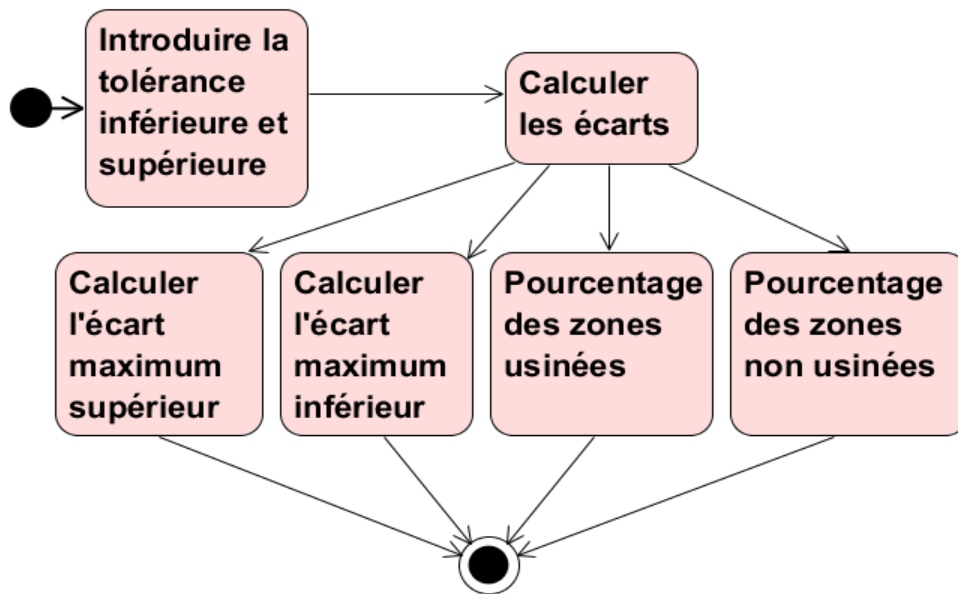


Figure 2.31. Diagramme d'activité « Calculer les erreurs ».

### 2.3. Diagramme de classes

Le diagramme de classes est une représentation statique des éléments qui composent un système et leurs relations. Il permet de spécifier qui intervient à l'intérieur du système (Figure 2.32).

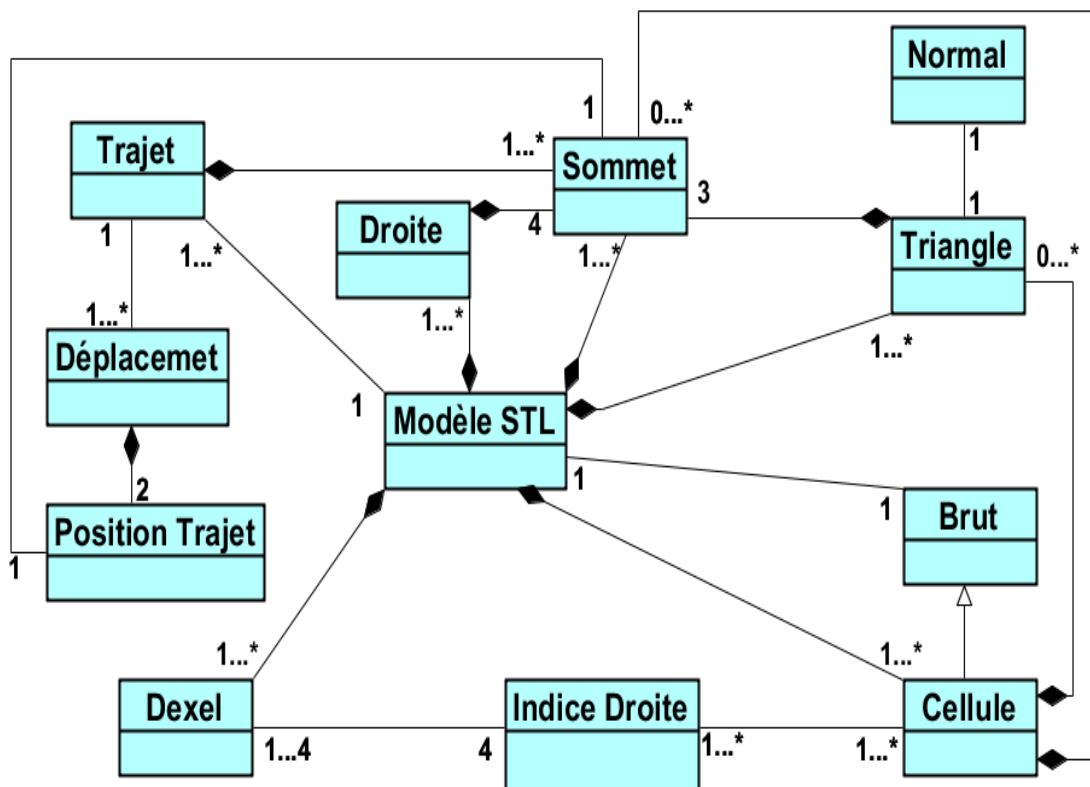


Figure 2.32. Diagramme de classes.

❖ **Classe « Modèle STL »**

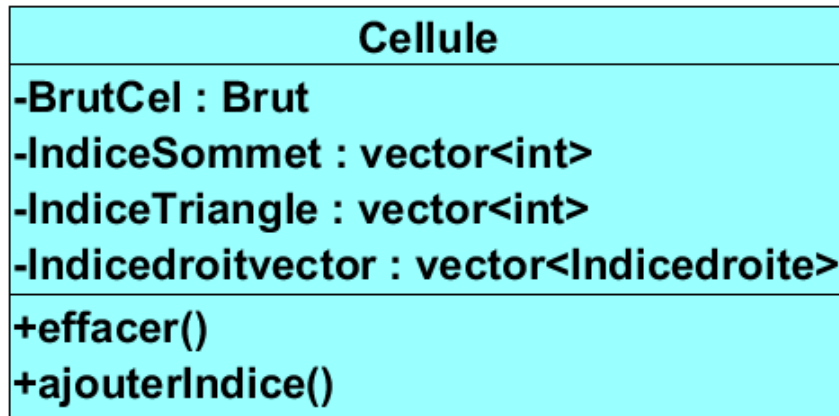
Elle regroupe tous les sommets et les triangles, les cellules créées, les droites, les informations sur l'outil et les points du trajet. Les méthodes de cette classe nous permettent de lire le fichier STL, de récupérer ces paramètres, de créer la grille, de lire le trajet d'outils, ... etc. (Figure 2.33).

<b>Modèle STL</b>
<p> <b>-Nombre_triangles : int</b>  <b>-Nombre_Sommet : int</b>  <b>-Largeur : double</b>  <b>-Longueur : double</b>  <b>-Hauteur : double</b>  <b>-Nombre_Cel_x : int</b>  <b>-Nombre_Cel_y : int</b>  <b>-Nombre_Cel_z : int</b>  <b>-Nbr_Dexel_pasX : int</b>  <b>-Nbr_Dexel_pasY : int</b>  <b>-Brut : Brut</b>  <b>-Tab_triangles : vector&lt;Triangle&gt;</b>  <b>-Tab_sommets : vector&lt;Sommet&gt;</b>  <b>-Tab_cellules : Cellule**</b>  <b>-Point_XY : Droit**</b>  <b>-Tab_dexel : Dexel**</b>  <b>-Deplacement : vector&lt;Deplacement&gt;</b>  <b>-Point_trajet : vector&lt;Position_Trajet&gt;</b>  <b>-point_trajet_globale : vector&lt;Position_Trajet&gt;</b>  <b>-ecart_max_inf : double</b>  <b>-ecart_max_sup : double</b>  <b>-PourcentageMatiereEnleve : double</b> </p>
<p> <b>+Lire_STL()</b>  <b>+creer_cellule()</b>  <b>+Parametre_STL()</b>  <b>+Creer_Grille()</b>  <b>+Lire_trajetdoutil()</b>  <b>+calculer_intersection_pointtrajet()</b>  <b>+Calculetheorique()</b>  <b>+Calculetolerance()</b>  <b>+calculerMatiereEnleve()</b> </p>

Figure 2.33. Classe « Modèle STL ».

❖ **Classe « Cellule » :**

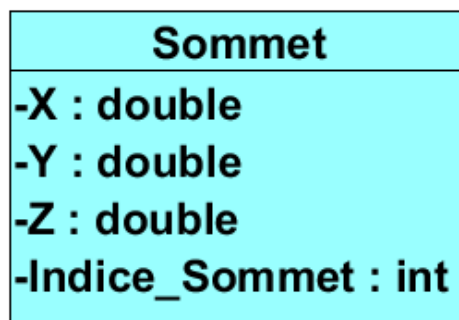
Elle regroupe les indices des sommets, des triangles et des droites. Les méthodes de cette classe nous permettent d'ajouter des indices aux vecteurs et les effacer. Elle hérite de la classe brute (Figure 2.34).



**Figure 2.34.** Classe « Cellule ».

❖ **Classe « Sommet » :**

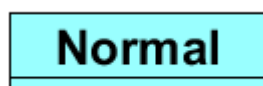
C'est la classe qui définit un sommet par ses différentes caractéristiques (Figure 2.35).



**Figure 2.35.** Classe « sommet ».

❖ **Classe « Normal » :**

C'est la classe qui englobe la normale d'un triangle (Figure 2.44).



**Figure 2.36.** Classe « Normal ».

## ❖ Classe « Triangle » :

C'est la classe qui englobe les différentes caractéristiques du triangle (Figure 2.36).

<b>Triangle</b>
-Indice_Sommet1 : int
-Indice_Sommet2 : int
-Indice_Sommet3 : int
-Normal : Normal
-D : double
-Aire_Triangle : double
-Xmin : double
-Ymin : double
-Xmax : double
-Ymax : double
<b>+calculerSurfaceTriangle()</b>

Figure 2.37. Classe « triangle ».

## ❖ Classe « Dixel » :

Elle regroupe les indices des quatre (04) droites (Figure 2.37).

<b>Dixel</b>
-IndDroit1 : Indicedroite
-IndDroit2 : Indicedroite
-IndDroit3 : Indicedroite
-IndDroit4 : Indicedroite
-R : double
-G : double
-B : double
<b>+affecter_couleur()</b>

Figure 2.38. Classe « Dixel ».



❖ **Classe « Droite » :**

Elle regroupe les paramètres d'une droite (Figure 2.38).

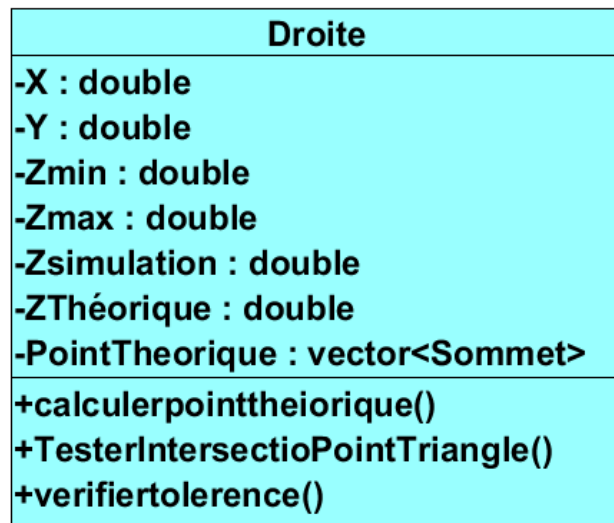


Figure 2.39. Classe « droite ».

❖ **Classe « Indice Droite » :**

Elle regroupe les indices d'une droite dans le plan XY (Figure 2.43).

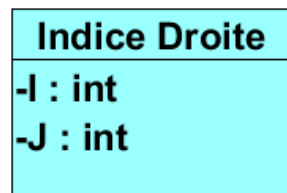


Figure 2.40. Classe « Indice Droite ».

❖ **Classe « Brut » :**

C'est la classe qui détermine les dimensions minimales d'une pièce (Figure 2.39).

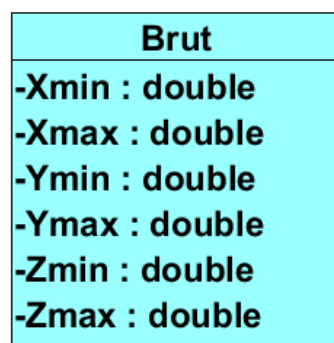


Figure 2.41. Classe « brute ».

❖ **Classe « Position Trajet » :**

Elle représente un outil. Les fonctions de cette classe sont implémentées afin de déterminer les différentes caractéristiques de l'outil (Figure 2.40).

Position Trajet
-Forme : string -Longueur : double -Rayon : double -CL : Sommet -s1 : Sommet -s2 : Sommet -s3 : Sommet -s4 : Sommet
+CreerEnveloppe() +CreerCelluleChevauchement() +IntersectionOutilDroit() +calculerIntersection()

Figure 2.42. Classe « Position Trajet ».

❖ **Classe « Déplacement » :**

Elle regroupe les indices de deux positions successives ainsi que les points intermédiaires (Figure 2.41).

Déplacement
-Ind_pos_1 : int -Ind_pos_2 : int -Points_intermediaires : vector<Position Trajet>
+cree_position_intermediaire()

Figure 2.43. Classe « Déplacement ».

❖ **Classe « Trajet » :**

Elle regroupe le vecteur des positions de passage des outils ainsi que les déplacements associés (Figure 2.42).

Trajet
-Nombre_Point_Trajet : int -PointCL : vector<Sommet> -VectorDéplacement : vector<Déplacement>

Figure 2.44. Classe « Trajet ».

## Conclusion

Dans ce chapitre, nous avons présenté l'architecture générale de notre application d'un point de vue conceptuel, où nous avons recensé nos besoins en spécifiant toutes les étapes suivies et toutes les solutions proposées.

Dans le chapitre suivant, l'implémentation et la mise en œuvre de notre application seront présentées et la validation de l'application sera prouvée à travers un exemple.

# **Chapitre 3**

---

**Implémentation**

**Informatique**

**et Validation**

## Introduction

Dans ce dernier chapitre, nous allons présenter l'implémentation informatique de la solution proposée et sa validation à travers plusieurs exemples. Nous commençons par une présentation des outils utilisés et de l'application développée et nous terminons par la phase de validation.

### 1. Présentation des outils utilisés

Ce travail s'insère dans le cadre de développement d'une plateforme logicielle pour la production des pièces de formes complexes initié par l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO » de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA ». Les outils utilisés pour le développement de la plateforme sont le langage C++ et la bibliothèque OpenGL.

#### 1.1. Présentation du langage C++

Créé initialement par Bjarne Stroustrup dans les années 1980, il est aujourd'hui normalisé par l'ISO. C++ est un langage de programmation compilé permettant la programmation sous multiples paradigmes. Ses bonnes performances et sa compatibilité avec le C en font l'un des langages de programmation les plus utilisés dans les applications où la performance est critique. Ils existent de nombreuses bibliothèques C++ en plus de la bibliothèque standard du C++ (C++ Standard Library) qui est incluse dans la norme. Par ailleurs, C++ permet l'utilisation de l'ensemble des bibliothèques C existantes. Parmi les environnements de développements : Anjuta DevStudio, C++ Builder, Code Blocks (open-source), Dev-C++, Eclipse (open-source), Microsoft Visual C++, etc.



#### 1.2. Présentation d'OpenGL (Open Graphics Library)

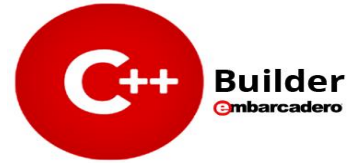
Est une API graphique multiplateformes open-source, de bas niveau, dédiée aux applications générant des images 2D ou 3D. OpenGL est sortie en 1992 par la Silicon Graphics Inc (SGI) et est actuellement gérée par Khronos Group. OpenGL est conçue pour développer des applications graphiques portables. Elle favorise l'innovation et la simplicité d'utilisation en offrant aux développeurs de nombreuses fonctionnalités de visualisation. Etant libre, ouverte et portable, la bibliothèque graphique a pu se vendre dans le milieu scientifique et industriel et être utilisée dans de nombreuses applications artistiques ou de traitement d'images. Du fait de ses performances, l'API OpenGL est aussi présente dans différents domaines tels que jeux-vidéo ludiques, les applications web, les systèmes embarqués et les téléphones portables. C'est une API très performante et simple d'utilisation qui permet de réaliser des rendus et des scènes complexes selon les besoins. Cette technologie est ouverte



et est implémentée par de nombreuses bibliothèques sous des langages de programmations différents dont des langages de script comme Python ou Ruby.

### 1.3. Présentation d'Embarcadero Builder C++ :

Builder C++ est un logiciel de développement rapide d'applications conçu par Borland. Tout d'abord C++ est un outil RAD, c'est-à-dire orienté vers le développement rapide d'applications (Rapid Application Development) sous Windows. Builder C++ permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows. Le but du Builder C++ est de créer des applications rapidement sous Windows grâce à ses bibliothèques variées ainsi qu'à une interface graphique avec son éditeur de ressources.



## 2. Présentation de la plateforme logicielle

### 2.1. Fenêtres principales

La plateforme logicielle est composée de deux parties (Figure 3.1) :

- **Partie visualisation** : elle permet de visualiser tous les objets géométriques en 3D (points, surfaces, outils, machine, simulation, trajet d'usinage, etc.) en utilisant la bibliothèque graphique « OpenGL ». Dans cette partie, les objets visualisés peuvent subir diverses transformations géométriques telles que translations, rotations, projections, zoom et changement d'échelles.
- **Partie manipulation** : cette partie est composée d'une barre de menu et d'un ensemble de boutons de manipulation des différents paramètres des modèles CAO. Dans cette partie sont lancés tous les modules logiciels intégrés à la plateforme logicielle.

### 2.2. Barre du menu principal

Elle est composée de trois rubriques :

- **Fichier** : comporte toutes les fonctionnalités de manipulation des fichiers.
- **Option** : permet la modification des différents paramètres des courbes et des surfaces ainsi que le lancement des différentes opérations d'usinage.
- **Aide** : pour afficher l'aide.

Pour lancer le module logiciel intégré à la plateforme logicielle, il faut passer par « Option », « Applications Industrielles » et « Simulation Usinage 3 Axes » (Figure 3.2).

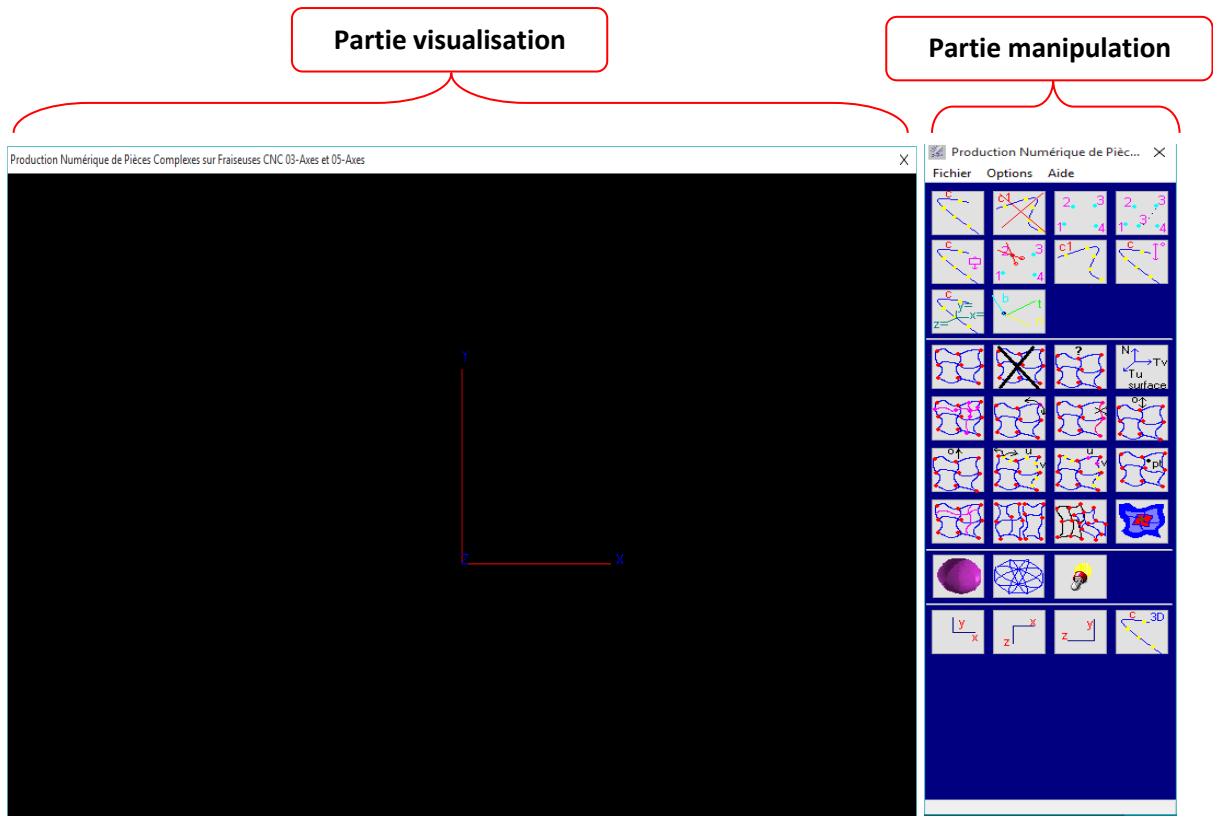


Figure 3.1. Fenêtres principales de la plateforme logicielle.

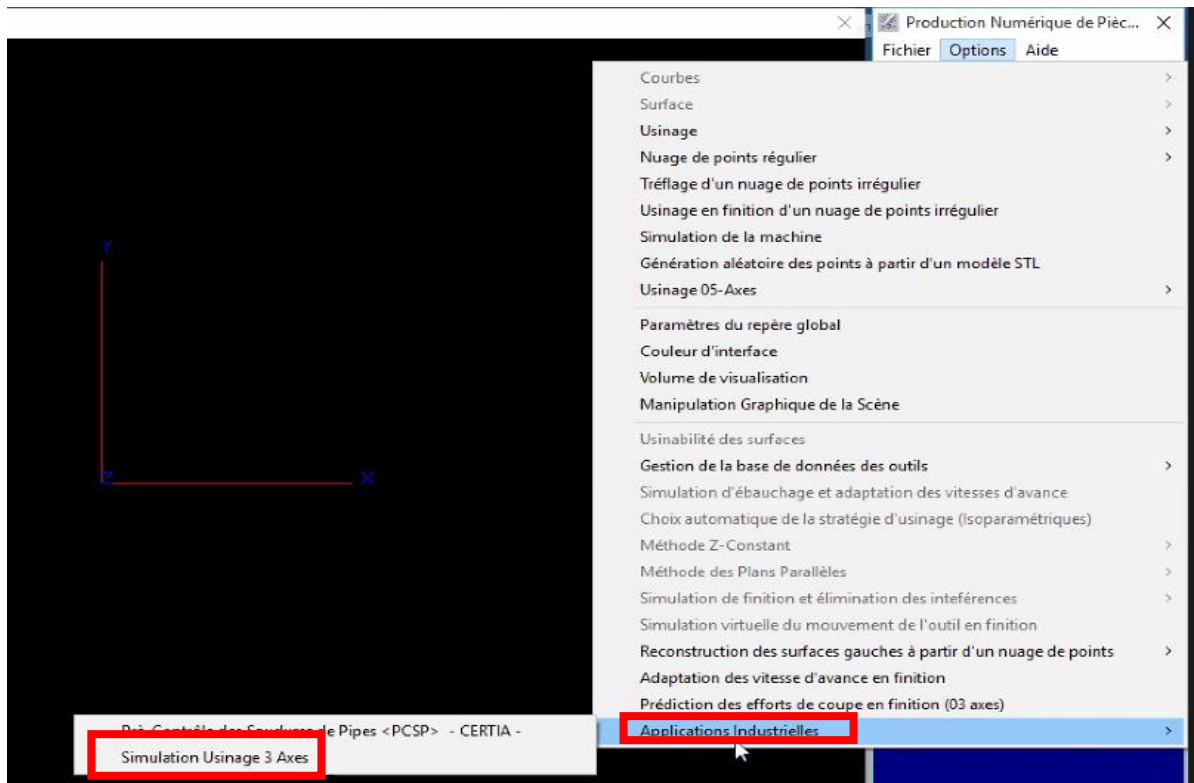


Figure 3.2. Lancement du logiciel développé « Simulation Usinage 3 Axes ».

### 2.3. Simulation 3 Axes

Pour atteindre l'objectif principal qui la simulation d'usinage et la vérification des surfaces résultantes, le module logiciel développé est composé de plusieurs fonctionnalités complémentaires. Il est composé de quatre (04) onglets (Figure 3.3). Les différents onglets seront détaillés dans les paragraphes suivants.



**Figure 3.3.** Onglets du module logiciel développé.

#### 2.3.1. Onglet « Gérer le modèle STL »

Le premier onglet « **Gérer le modèle STL** » permet de lire le fichier STL contenant des informations sur les sommets (coordonnées des sommets) et les triangles (la normale et les indices des sommets) (Figure 3.4).

Par un clic sur le bouton « **Lire le fichier STL** » on sélectionne le fichier désiré. La syntaxe du fichier est vérifiée et un fichier de sortie contenant les informations est généré.

Une fois le nombre de cellules suivant les axes X, Y et Z est introduit et par un clic sur le bouton « **Structurer paramètres modèle STL** », les paramètres des triangles et des sommets sont récupérés.

En plus de ces informations, il est possible de visualiser :

- Sommets du triangle.
- Triangle en deux modes rendu et filaire.
- Les normales des triangles.
- Sommets du brut.
- Brut en deux modes (rendu et filaire).
- Sommets affectés aux cellules.
- Cellules en deux modes (rendu et filaire).
- Triangles affectés aux cellules.



Simulation Usinage 3 Axes

Gérer le modèle STL    Créer les dexels

Lire le fichier STL

**Informations**

Xmin: -108.422    Ymin: 0.000    Zmin: -42.927

Xmax: 11.578    Ymax: 100.000    Zmax: -2.927

Longueur: 120.000    Largeur: 100.000    Hauteur: 40.000

**Paramètre des Cellules**

Nombre cellules X: 10    Nombre cellule Y: 10

Nombre cellule Z: 1

Structurer paramètre modèle STL

100%

**Resultat**

Nombre triangles: 28    Nombre sommets: 16

**Visualisation**

**Brut**

Sommets du brut: 1

Brut rendu

Brut filaire: 1

**Modele**

Sommets: 1

Triangles rendu

Triangles filaire: 1

Normales des triangles: 1

**Cellule**

Sommets des cellules: 1

Cellules rendu

Cellules filaire: 1

Triangle des cellules: 1

Activer l'interface  
Accédez à...

Figure 3.4. Onglet « Gérer le modèle STL ».

### 2.3.2. Onglet « Créer les Dexels »

Le deuxième onglet « **Créer les Dexels** » permet la création des Dexels suite à la spécification des pas suivant les axes X et Y et avec un simple clic sur le bouton « **créer les Dexels** » (Figure 3.5). Une fois les Dexels créés, un clic sur le bouton « **calculer le point théorique** » permet de calculer les points d'intersections entre les droites des Dexels et les triangles du modèle STL. A ce niveau, les options suivantes sont disponibles :

- Visualiser les droites des Dexels.
- Visualiser les Dexels en deux modes filaire et rendu.
- Visualiser les points théoriques les plus hauts.

- Visualiser tous les points théoriques.

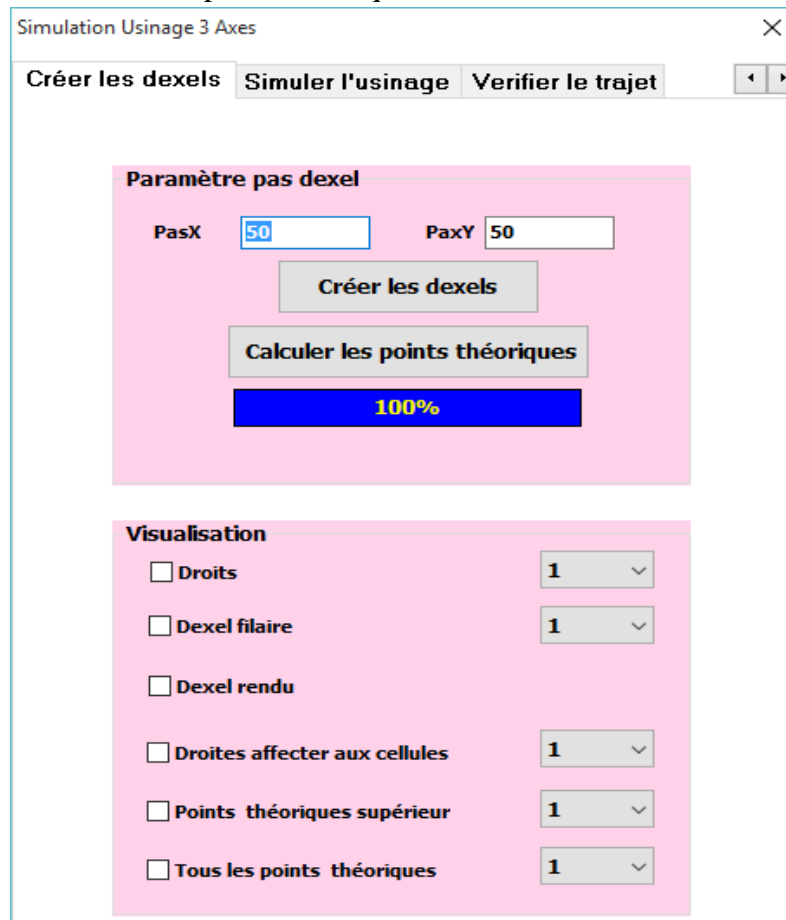


Figure 3.5. Onglet « Créer les Dexels ».

### 2.3.3. Onglet « Simuler l'usinage »

Dans cet onglet, l'utilisateur doit effectuer trois (03) opérations (Figure 3.6). La première opération consiste à lire le trajet d'outil et à récupérer les points d'extrémité de chaque outil par un clic sur le bouton « **ouvrir le trajet d'outil** ». Une fois le trajet est lu, le nombre de points du trajet apparu, la visualisation du trajet en points et en lignes sera possible. Ensuite, le bouton « **créer des positions intermédiaire** » permet de créer des positions supplémentaires en spécifiant un incrément. Une fois les positions sont créées, le nombre de points du trajet global apparu et la visualisation des points intermédiaires sera possible. A ce niveau, les options suivantes sont disponibles :

- Visualiser l'outil utilisé en deux modes filaire et rendu.
- Visualisation partielle du point CL (point d'extrémité de l'outil) :
  - ✓ Point CL.
  - ✓ Enveloppe.
  - ✓ Cellules de l'enveloppe.
  - ✓ Droites après l'intersection.

La dernière opération permet de calculer toutes les intersections par le bouton « **calculer toutes les intersections** ». La forme de l'outil, son rayon, sa longueur et les coordonnées du point CL sont affichés au fur et à mesure. Il est possible de visualiser :

- Point CL et les positions intermédiaires.
- Dexels filaire et rendu.
- Points de simulation.

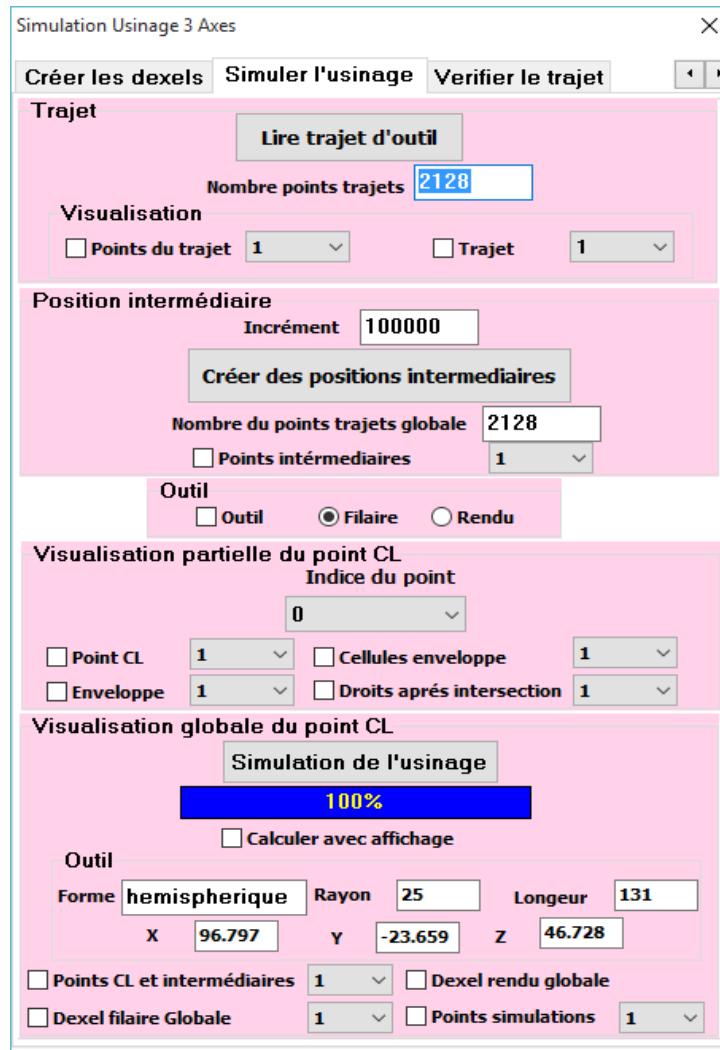


Figure 3.6. Onglet « Simuler l'usinage ».

### 2.3.4. Onglet « Vérifier le trajet »

Le dernier onglet permet de vérifier le trajet et de détecter les zones usinées et non usinées (Figure 3.7). L'utilisateur doit introduire la tolérance inférieure et supérieure puis avec le bouton « **calculer les écarts** », les écarts maximums inférieurs, les écarts maximums supérieurs, pourcentage des zones non usinées et la matière enlevée sont calculés et affichés. On peut visualiser :

- ZSimulation.
- Segments.
- Dexels.

Simulation Usinage 3 Axes

Créer les dexels Simuler l'usinage **Vérifier le trajet**

**calculer les écarts**

Tolérance inférieure  Tolérance supérieure

**Calculer les écarts**

**Informations**

Ecart max inférieure	<input type="text" value="-0.289"/>
Ecart max supérieure	<input type="text" value="1.686"/>
Pourcentage des zones non usinées	<input type="text" value="50.000"/>
Pourcentage de la matière Enlevée	<input type="text" value="26.119"/>

**Visualisation**

<input type="checkbox"/> Zsimulations	<input type="text" value="1"/>
<input type="checkbox"/> Segments	<input type="text" value="1"/>
<input type="checkbox"/> Dexels	<input type="text" value="1"/>

Figure 3.7. Onglet « Vérifier le trajet ».

### 3. Test et validation

Le module logiciel que nous avons intégré a été testé sur un PC dont les caractéristiques sont les suivantes :

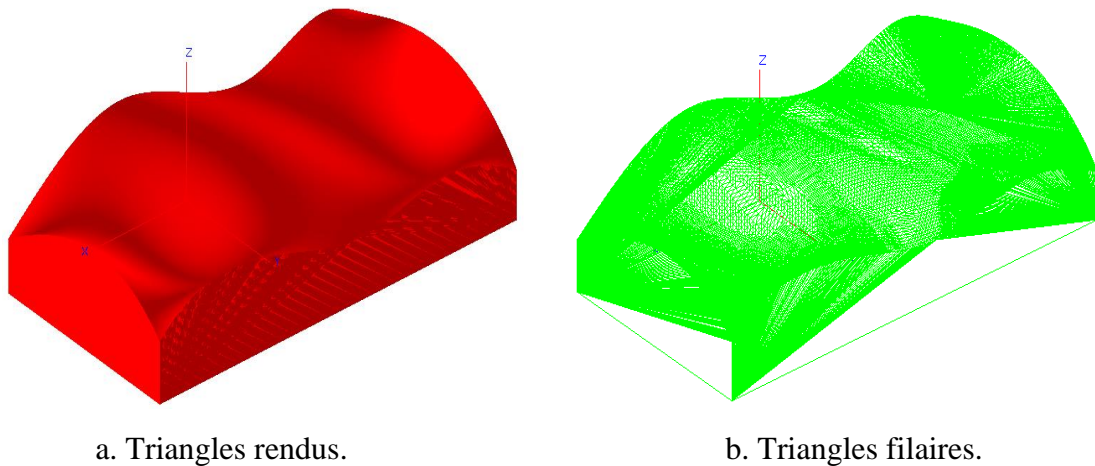
- Windows 7.
- Processeur Intel(R) Core (TM) i7-6600U CPU @ 2.60GHz.
- Mémoire installée (RAM) : 8.00 (7.88 Go utilisable).
- Type de système : système d'exploitation 64 bits.

Arrivé au terme de la présentation de l'application, la phase de validation du travail et les différents résultats auxquels nous avons abouti sont présentés dans la suite. Plusieurs tests ont été menés sur différents modèles de pièce pour valider l'approche proposée. Dans

ce qui suit, nous allons présenter les résultats pour deux modèles de pièces conçues dans un logiciel de CAO.

### 3.1. Premier modèle STL « modèle de finition » :

Le modèle STL de la première pièce est représenté par la Figure 3.8. Dans ce qui suit, nous allons montrer toutes les étapes permettant de simuler l'usinage pour l'opération de finition de cette pièce.



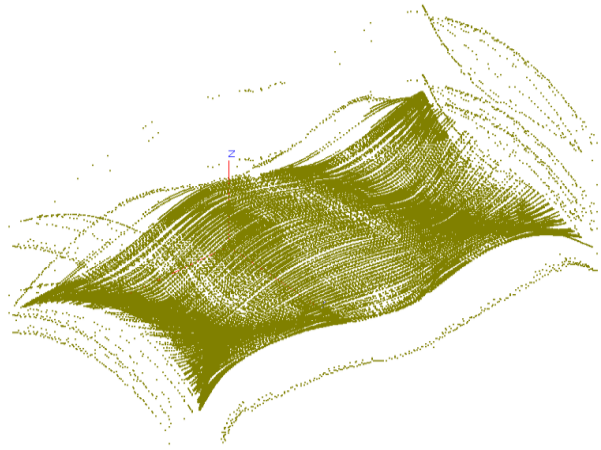
**Figure 3.8.** Premier modèle STL.

- **Etape 1 :** lecture du fichier texte contenant les paramètres STL de la pièce à usiner et les limites du brut. Les résultats obtenus sont :
 

Xmin = -89.000 mm	Ymin = 0.000 mm	Zmin = 0.000 mm
Xmax = 89.000 mm	Ymax = 90.000 mm	Zmax = 48.523 mm
Longueur = 178.000 mm	Largeur = 90.000 mm	Hauteur = 48.523 mm
- **Etape 2 :** subdivision du brut en 800 cellules suivant l'axes X et 800 cellules suivant l'axes Y et 1 cellule suivant l'axe Z et affectation des sommets et des triangles aux cellules. Les résultats obtenus sont :
 

Nombre total de triangles = 40079	Nombre total de sommets = 20134
-----------------------------------	---------------------------------
- **Etape 3 :** création des Dexels avec des pas suivant l'axe X et l'axe Y.
- **Etape 4 :** calcul des points théoriques.
- **Etape 5 :** lecture du fichier trajet et calcul des points du trajet donnent le résultat suivant : Nombre de points du trajet : 68751

Les résultats de la visualisation sont présentés par la Figure 3.9.

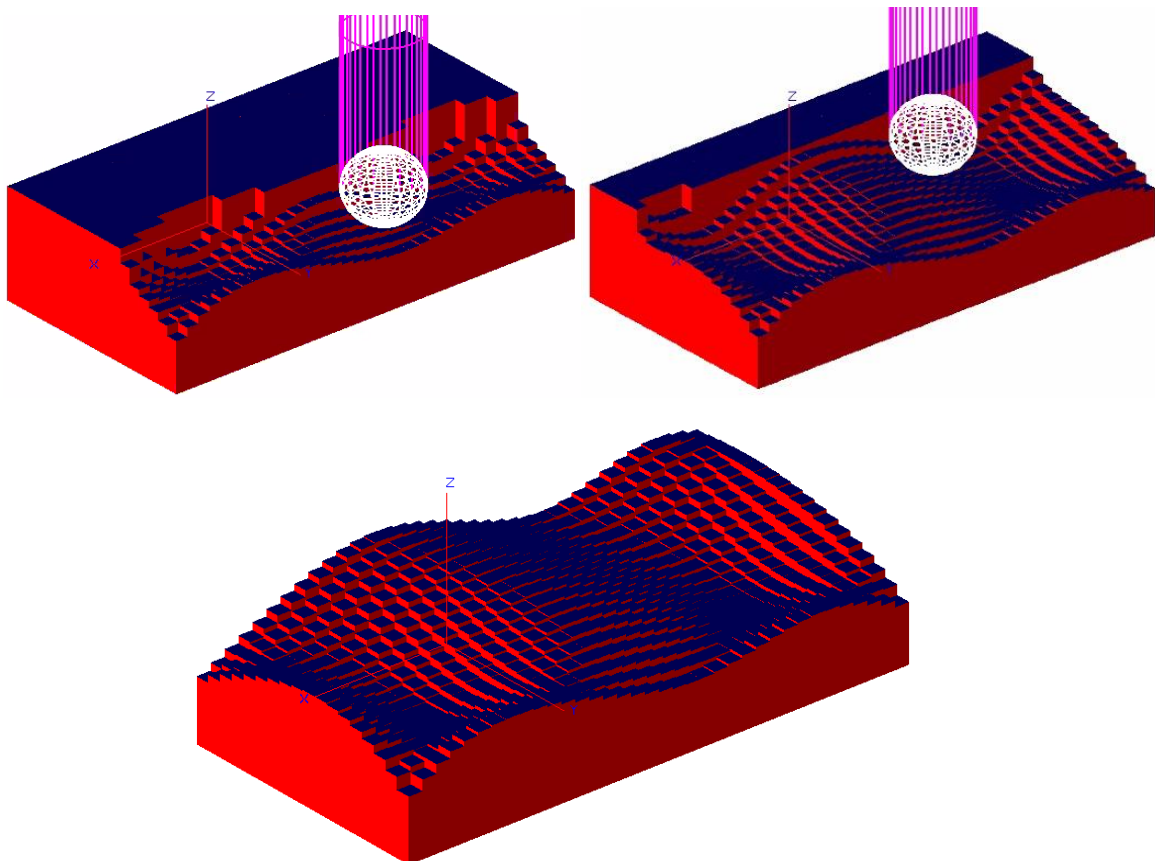


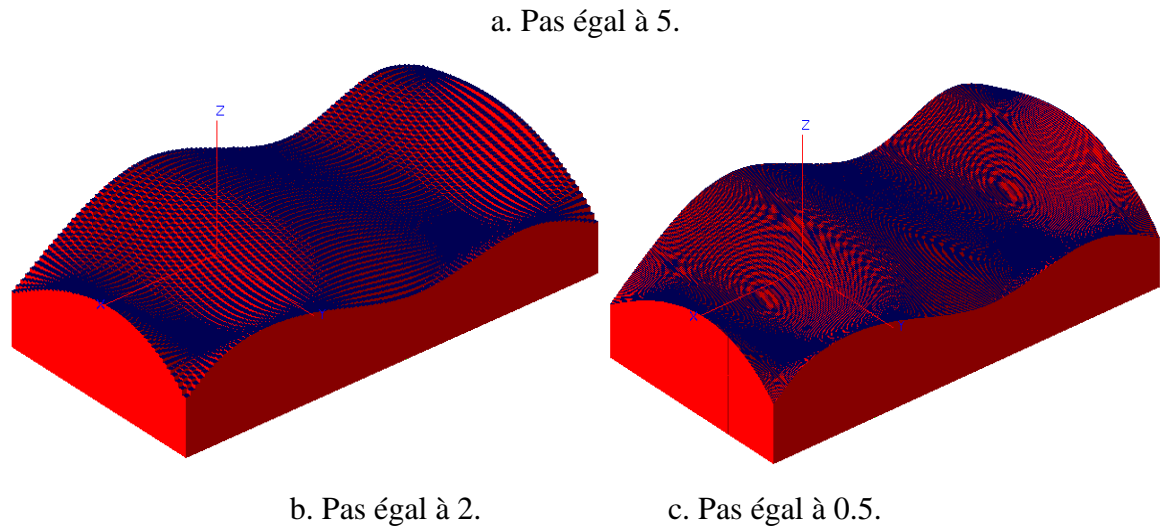
**Figure 3.9.** Points du trajet d'outils.

- **Etape 6 :** création des points intermédiaires après l'introduction d'un incrément égal à 100000 mm pour ne pas insérer de points intermédiaires. Donc, le nombre de points du trajet global ne change pas.

Pour chaque point du trajet, on crée son enveloppe et on calcule l'intersections entre les droites des cellules qui sont en chevauchement avec cette enveloppe et l'outil.

- **Etape 7 :** c'est l'étape du calcul des intersections des droites avec l'outil pour simuler l'usinage (Figure 3.10). Il ressort des résultats que plus les pas sont fins plus la pièce résultante se rapproche de la pièce réel.





**Figure 3.10.** Résultat de la simulation d'usinage pour différents pas.

- **Etape 8 :** avec une tolérance inférieure égale à 1 et une tolérance supérieure égale à 1, les résultats obtenus pour les différents pas sont :

- Pas=5mm :

Ecart inférieur maximum = -0.010mm  
 Ecart maximum supérieur =0.241mm  
 Pourcentage des zones non usinées =0%  
 Pourcentage de la matière enlevée =35.593%

- Pas=2mm :

Ecart inférieur maximum = -0.014mm  
 Ecart maximum supérieur =0.232mm  
 Pourcentage des zones non usinées =0%  
 Pourcentage de la matière enlevée =36.09%

Avec une tolérance inférieure ou égale à 0.5 et supérieure ou égale à 0.5, les résultats obtenus sont :

- Pas=0.5mm :

Ecart inférieur maximum = -0.020mm  
 Ecart maximum supérieur =0.453mm  
 Pourcentage des zones non usinées =0%  
 Pourcentage de la matière enlevée =36.326%

La Figure 3.11 montre que toute la surface est usinée et donc absence de zones non usinées.

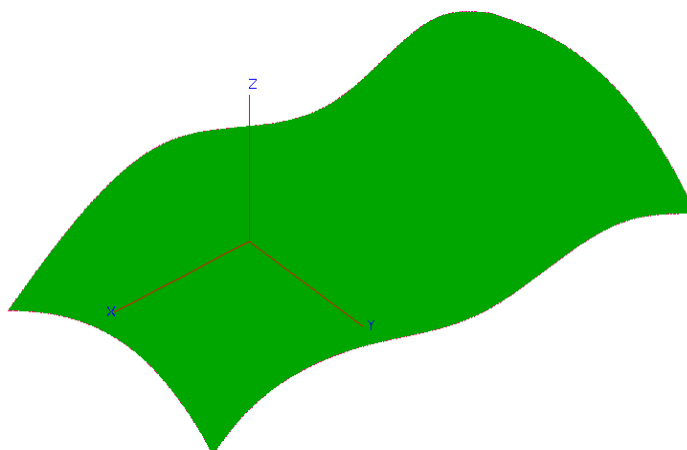
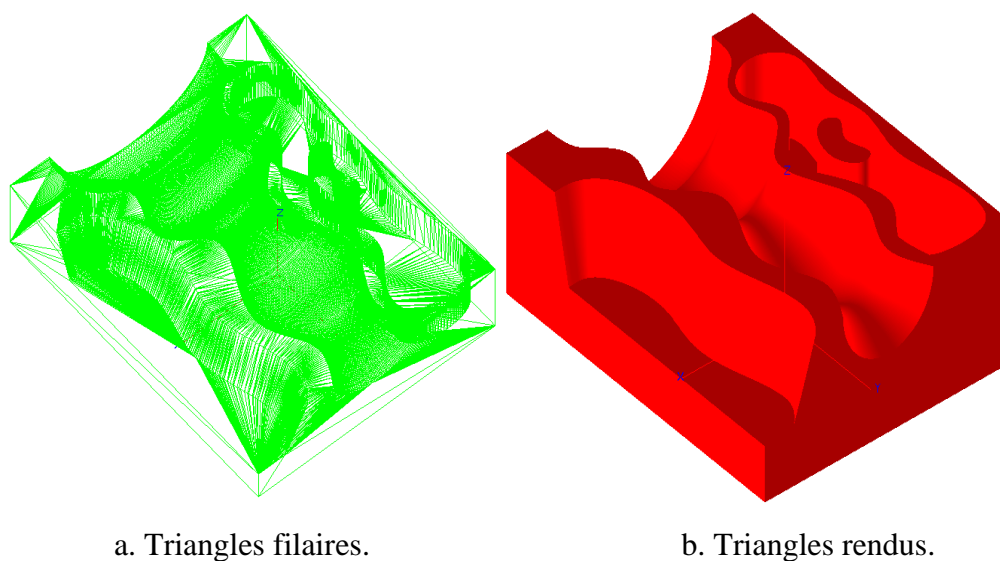


Figure 3.11. Zones usinées.

### 3.2. Deuxième modèle STL « modèle d'ébauchage »

Le modèle STL de la deuxième pièce est représenté par la Figure 3.12. Dans ce qui suit, nous allons montrer toutes les étapes permettant de simuler l'usinage pour l'opération d'ébauchage de cette pièce.



a. Triangles filaires.

b. Triangles rendus.

Figure 3.12. Deuxième modèle STL.

- **Etape 1 :** lecture du fichier texte contenant les paramètres STL de la pièce à usiner et les limites du brut. Les résultats obtenus lors la lecture sont :

Xmin = -49.871 mm

Ymin = -77.904 mm

Zmin = 0.000 mm

Xmax = 70.129mm

Ymax = 72.096 mm

Zmax = 50.000 mm



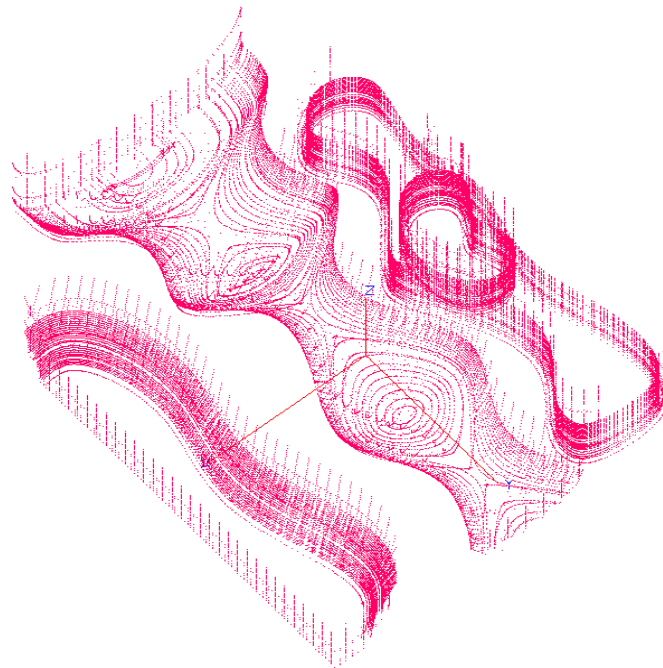
Longueur = 120.000 mm    Largeur = 150.000mm    Hauteur= 50.000 mm

- **Etape 2** : subdivision du brut en 500 cellules suivant l'axes X et 500 cellules suivant l'axes Y et 1 cellule suivant l'axe Z puis affectation des sommets et des triangles aux cellules. Les résultats obtenus lors la lecture sont :

Nombre total des triangles =47116                      Nombre total des sommets = 23558

- **Etape 3** : création des Dexels avec un pas suivant l'axe X égal à 1mm et un pas suivant l'axe Y égal à 1mm.
- **Etape 4** : calcul des points théoriques.
- **Etape 5** : lecture du fichier trajet et calcul des points du trajet donne 73806 positions d'outil.

Les résultats sont visualisés par la Figure 3.13.



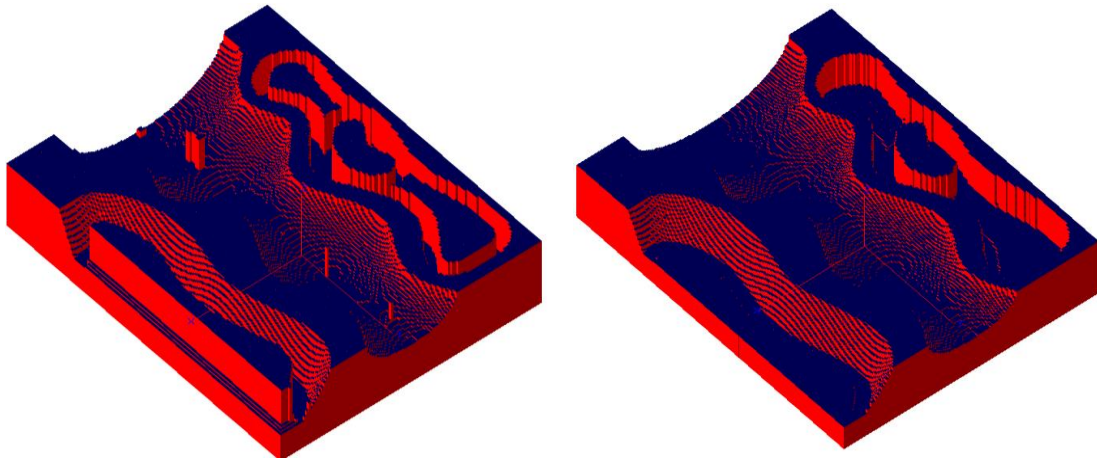
**Figure 3.13.** Points du trajet d'outil.

- **Etape 6** : création des points intermédiaires après introduction d'un incrément égale à 1000 mm (sans points intermédiaires). Le nombre de points du trajet d'outil reste inchangé.

Création des points intermédiaires après l'introduction d'un incrément égal à 0.5mm. Le nombre de points global du trajet devient 427871.

- **Etape 7** : simulation de l'usinage (Figure 3.14). Il ressort d'après les résultats que l'insertion de points intermédiaires a permis de simuler correctement les

mouvements continus des outils et par conséquent la forme finale de la pièce résultante.



a. Sans les points intermédiaires.

b. Avec des points intermédiaires.

**Figure 3.14** Résultat de la simulation de l'usinage.

- **Etape 8 :** calcul des écarts.

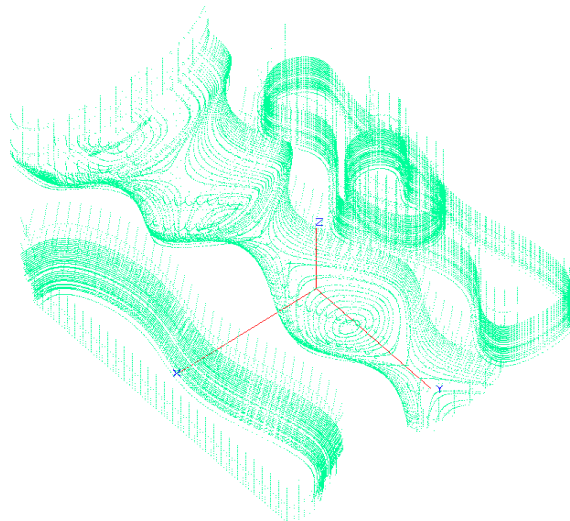
**Sans points intermédiaires :** la tolérance inférieure est égale à 0.8mm et la tolérance supérieure est égale à 0.8mm. Les résultats obtenus sont (Figure 3.15 et Figure 3.16) :

Ecart maximum inférieur = -0.354mm

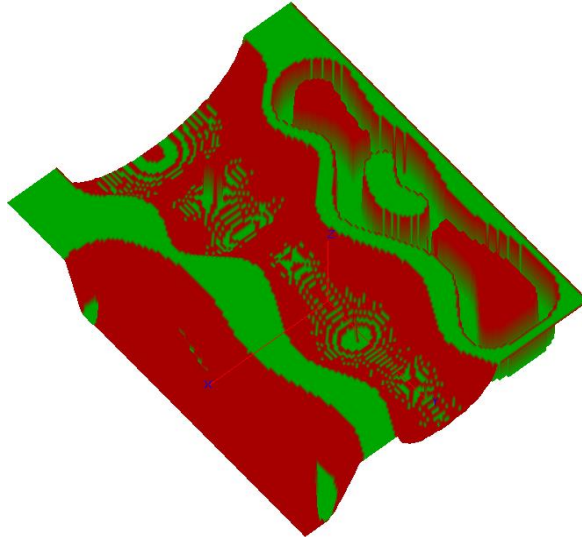
Ecart maximum supérieur = 0.531mm

Pourcentage des zones non usinées = 64.922%

Pourcentage de la matière enlevée = 26.049%



**Figure 3.15.** Points intermédiaires et trajets.



**Figure 3.16.** Zones usinées et zones non usinées.

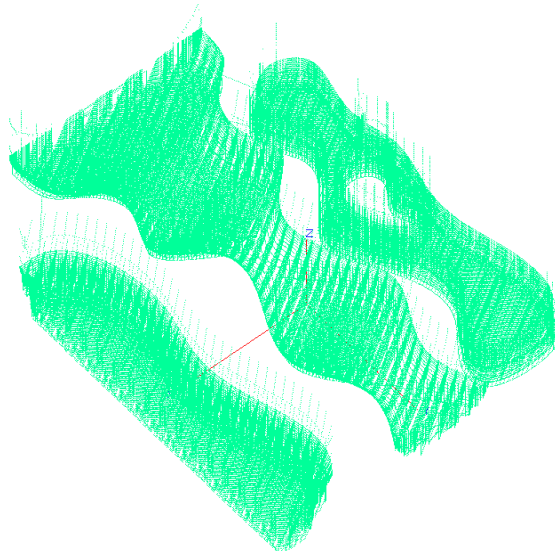
**Avec les points intermédiaires :** la tolérance inférieure est égale à 0.01mm et la tolérance supérieure est égale à 0.01. Les résultats obtenus sont (Figure 3.15 et Figure 3.16) :

Ecart maximum inférieur = -0.354mm

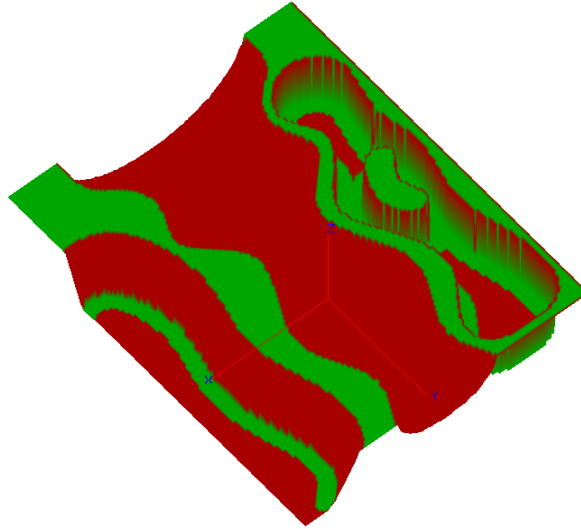
Ecart maximum supérieur = 0.532mm

Pourcentage des zones non usinées = 66.244%

Pourcentage de la matière enlevée = 31.892%



**Figure 3.17.** Points intermédiaires et trajet.



**Figure 3.18.** Zones non usinées

### **Conclusion :**

Dans ce chapitre, nous avons présenté application développée et les résultats de test obtenus. Au début, nous avons présenté l'environnement de développement, le langage de programmation utilisé ainsi qu'une brève définition de la bibliothèque graphique OpenGL. Ensuite, nous avons présenté les différentes interfaces et fonctionnalités de l'application tout en enrichissant cette présentation par des tests de validations sur deux exemples de modèle STL depuis sa lecture jusqu'à la simulation de l'usinage.

## CONCLUSION GENERALE

Le travail réalisé au sein de l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO » de la Division Productique et Robotique du « CDTA », a pour but, la conception et l'implémentation d'un module logiciel graphique et interactif permettant de simuler l'usinage des pièces de formes complexes, définies par leurs modèles STL, sur des fraiseuses numériques à 03-axes pour toutes les opérations d'usinage (ébauchage, demi-finition et finition) et les stratégies d'usinage associées en considérant les outils cylindriques et hémisphériques.

Lors de la réalisation de ce projet, nous avons mené une étude bibliographique sur le processus de production des pièces mécaniques de formes complexes, le format d'échange de données et les différentes techniques de simulation d'usinage. Par la suite, nous avons présenté la solution proposée et l'étude conceptuelle de l'application. Dans la dernière partie, nous avons montré les fonctionnalités du module logiciel développé avec deux exemples de validation.

Le résultat du module logiciel développé est l'intégration à la plateforme logicielle de l'équipe « CFAO » des fonctionnalités permettant la :

- Récupération des principaux paramètres du modèle STL de la pièce.
- Création des cellules et affectation des sommets et des triangles aux cellules.
- Création des Dexels et affectations des droites aux cellules.
- Calcul des points théoriques.
- Enrichissement du trajet d'outils.
- Simulation de l'opération d'enlèvement de matière.
- Calcul des écarts.

En perspective, nous recommandons le traitement des points suivants :

- 1 Considération de d'autres formes d'outils (toriques, coniques, etc.).
- 2 Adaptation des vitesses d'avance des outils en fonction du volume de matière enlevé pour chaque déplacement d'outil.
- 3 Calcul des efforts de coupe.

- 4 Simulation de l'usinage en 05-axes.
- 5 Programmation en utilisant des cartes GPU.
- 6 Simulation virtuelle des fraiseuses numériques à 05-axes.
- 7 Correction du trajet d'outils.

Le long de notre stage de fin d'étude au CDTA, nous avons pu mettre en pratique les connaissances acquises durant notre cursus à l'USDB et d'autre part, de préparer notre intégration à la vie professionnelle.

- [1]. P. BOURDET, CHRISTOPHE TOURNIER. Contribution à la conception des formes complexes : la surface d'usinage en fraisage 5 axes isocrête. Thèse de Doctorat. Ecole normale supérieure de Cachan. France. 2011.
- [2]. PR. PHILIPPE VERON , PR. ALI CHAABA ,DR. GEORGE MORARU, MERIEM HAYANI MECHKOURI. Intégration des comportements vibratoires dans les systèmes de conception et fabrication assistées par ordinateur (cfao). Thèse de Doctorat. Ecole nationale supérieure des arts et métiers. Meknès. Maroc. 2018.
- [3]. MOHAMED SLAMANI, HACHEMI FERHAT. Conception d'un montage d'usinage pour la fixation des pièces de formes complexes (directrice de turbine à gaz). Mémoire de Master, Université Mohamed Boudiaf - M'sila, Algérie. 2018.
- [4]. ZEMMORI MOHAMED ISLEM, KERKAR FARES. Prédiction de la topographie 3D des surfaces gauches lors de l'opération de finition en 05-axes. Mémoire de Master, Université Saad Dahleb, Blida. 2018.
- [5]. SAIDANE ASMAA, SERIR KAMILIA. Finition des pièces complexes par la stratégie « plans parallèles » sur des fraiseuses numériques à 03-axes. Mémoire de Master, Université Saad Dahleb, Blida. 2019.
- [6]. BELLEILI HAFSA CHARACHAL LATIFA. Combinaison des outils hémisphériques cylindriques et toriques pour la finition des surfaces complexes sur des fraiseuses à 05-axes. Mémoire de Master, Université Saad Dahleb, Blida. 2019
- [7]. DJEBBAR MERIEM, KOUDJI RADHIA. Fusion multidirectionnelle des triangulations de Delaunay en 3D. Mémoire de Master, Université Saad Dahleb, Blida. 2019.
- [8]. <https://all3dp.com/what-is-stl-file-format-extension-3d-printing> [consulté le 15 février 2020]
- [9]. DEROME, JEAN-PHILIPPE. L'association de stratégies d'usinage pour les pièces structurelles en aéronautique avec des caractéristiques de forme en utilisant un réseau de neurones. Mémoire de Maîtrise. Montréal, Canada. 2008.
- [10]. BENZAADA, S. Coupe des métaux. Université Mohamed Khider, Biskra, Algérie.
- [11]. GILLES, P. Positionnement d'outil torique pour l'usinage de surfaces gauches en fraisage 5-axes avec équilibrage de l'effort de coupe transversale. Thèse de Doctorat. Université de Toulouse 3 Paul Sabatier, France. 2008.
- [12]. H. BENDIFALLAH, M. BEY, W. OULD AMAR ET N. MAHROUG. Optimisation de l'opération de finition des surfaces gauches par la combinaison des formes et des dimensions d'outils à partir des modèles STL. Congrès algérien de mécanique, 14-17 Novembre 2011, Guelma, Algérie. .

- [13] OULMI TASSADIT ET MERZOUKI ZINA. Conception et étude d'une fraiseuse à commande numérique MOCN. Mémoire de Master, Université Mouloud Mammeri, Tizi Ouzou. 2015
- [14]. P. BOURDET,CHRISTOPHE TOURNIER. Contribution à la conception des formes complexes : la surface d'usinage en fraisage 5 axes isocrête. Thèse de Doctorat. Ecole normale supérieure de cachan. France. 2011.
- [15]. CIRP encyclopedia of production engineering. Publisher Springer publishing company. 2014.
- [16]. CHOURAR, ALI ET NEBBAT, HICHEM. Simulation et vérification des programmes d'usinage des surfaces gauche sur des fraiseuses à commande numérique à 3 axes. Mémoire de Master, Université Saad Dahleb, Blida. 2004.
- [17]. BRAHIM AGEUNINI ET MESSAOUDI HAMZA. Simulation et vérification de l'opération d'echauchage des surfaces gauches sur fraiseuses à commande numérique à 3 axes. Mémoire de Master, Université Saad Dahleb, Blida. 2006.
- [18]. KHALED SEBTI ET MOULAY ABDELHAMID. Modélisation volumique des pièces de formes complexes par triple dexels à partir de leurs modèles STL. Mémoire de Master, Université Saad Dahleb, Blida. 2015.
- [19]. BOUMENIR SIDAHMED ET DEBIEB MOHAMED CHEMS EDDINE. Conception et développement d'une approche optimale pour la génération du modèle triple-dexels de pièces de formes complexes. Mémoire de Master, Université Saad Dahleb, Blida. 2018.