

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE SAAD DAHLAB DE BLIDA  
FACULTE DES SCIENCES  
DEPARTEMENT D'INFORMATIQUE



## MEMOIRE DE FIN D'ETUDES

Pour l'obtention

d'un Diplôme de Master en Informatique

Option : Systèmes Informatiques et Réseaux

### THÈME :

**Mise en Place d'un Environnement Logiciel  
pour la Virtualisation de l'Usinage des Pièces  
Complexes sur Fraiseuses Numériques à 5-Axes**

**Réalisé par :**

M<sup>elle</sup>. BENMIMOUNA Amina

M<sup>elle</sup>. ELOUCHEFOUNE Imene

**Soutenu devant :**

Mr. BEY Mohamed	CDTA,	Encadreur
Mr. BENDIFALLAH Hassène	CDTA,	Encadreur
Mr. HAMOUDA Mohamed	USDB,	Promoteur
Mme. CHERIGUENE Soraya	USDB,	Présidente
Mr. OULD AISSA Ahmed	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 virtualisation de l'usinage des pièces complexes, définies par leurs modèles STL, sur des fraiseuses numériques à 05-axes. Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle, un module logiciel graphique et interactif permettant la simulation des différentes opérations d'usinage (ébauchage, demi-finition et finition) sur des pièces complexes discrètes en considérant les différentes formes d'outils (cylindriques et hémisphériques).

**Mots Clés :** Pièce Complexe, Modèle STL, Virtualisation de l'Usinage, Fichier CL-File, Trajet Outil, Fraiseuse 05-axes.

## **Abstract:**

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 virtualization of the machining of complex parts, defined by their STL models, on 05-axis CNC milling machines. This involves designing, developing and integrating into the software platform, a graphic and interactive software module allowing the simulation of the different machining operations (roughing, semi-finishing and finishing) on complex discrete parts by considering the different shapes of tools (cylindrical and hemispherical).

**Keywords:** Complex part, STL Model, Virtualization of the Machining, CL-File, Toolpath, 05-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 بإضفاء الطابع الافتراضي على محاكاة قطع الأجزاء المعقدة، المحددة بواسطة نماذج STL الخاصة بها، على آلات التفريز الرقمية ذات 05 محاور. يتضمن ذلك تصميم، تطوير ودمج في منصة البرامج، وحدة برمجية رسومية وتفاعلية تسمح بمحاكاة عمليات التصنيع المختلفة (التخشين، نصف التشطيب والتشطيب) على الأجزاء المنفصلة المعقدة من خلال مراعاة الأدوات ذات الأشكال المختلفة (أسطوانية و نصف كروية).

**الكلمات المفتاحية:** الجزء المعقد، نموذج STL، محاكاة القطع، CL-File، مسار الأداة، آلة التفريز الرقمية ذات 05 محاور.

## **REMERCIEMENTS**

*Nous remercions, tout d'abord, Dieu tout puissant de nous avoir donné le courage, la force et la patience d'achever ce modeste travail.*

*Les travaux présentés dans ce mémoire pour sanctionner la fin des études en Master 2 ont été effectués au CDTA « Centre de Développement des Technologies Avancées », sous la direction de Mr. Bey Mohamed, qu'il trouve ici tous nos remerciements et gratitude pour toute l'attention qu'il nous a porté, par sa présence tout au long du stage, ses orientations et surtout sa patience et son dévouement pour ce résultat.*

*Nous remercions Mr. Bendifallah Hassène pour sa collaboration et proposition.*

*Nous tenons à remercier vivement Mr. Hammouda Mohamed pour avoir accepté de diriger cette étude, la confiance qu'il nous a accordé et pour ses conseils, son orientation, les diverses corrections et encouragements qu'il nous a apportés pendant notre recherche.*

*Nous tenons à remercier les membres du jury qui nous ont fait l'honneur de juger ce travail et qui par leurs disponibilités, leurs observations et leurs rapports nous ont permis d'enrichir notre travail.*

*Enfin, nous tenons aussi à remercier toutes les personnes qui ont participé de manière directe ou indirecte à la concrétisation de ce travail.*

**Merci.**

## **Dédicaces**

### **À MES CHERS PARENTS**

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et pour mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.*

### **A MES CHERS ET ADORABLE SŒURS**

*Aya l'aimable, Arwa, la prunelle de mes yeux  
Je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.*

### **A MON FRERE YACINE**

*Les mots sont insuffisants pour exprimer ma profonde estime.  
Je vous dédie ce travail en témoignage de mon attachement indéfectible.  
Que Dieu te protège, t'accorde santé, succès et plein de bonheur dans ta vie*

### **A LA MEMOIRE DE MA GRANDE MERE**

*J'aurais tant aimé que vous soyez présente.  
Que Dieu ait vos âmes dans sa sainte miséricorde.*

### **À MES AMIES DE TOUJOURS :**

*Randa, Asmaa, Hadjer, Asmaa, Loubna, Samira ...  
Souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble.*

**AMINA**

## **Dédicaces**

*Au Nom de Dieu Clément et Miséricordieux,*

*Louange à Dieu pour sa Grâce et sa miséricorde,*

*Louange à Dieu pour la science qui nous a offert,*

*Du fond du cœur*

*Je dédie ce modeste travail à :*

*A mes très chers parents,*

*A mon frère Fayçal et à ma sœur Radjae,*

*A Youcef et sa famille,*

*A tous les membres de la famille,*

*A ma binôme Amina et sa famille,*

*A ma collègue Latifa et sa famille,*

*A toute personne ayant contribué de loin ou de près à accomplir ce travail.*

**Imene**

---

<b>Introduction générale</b> .....	1
<b>Chapitre 1 : Production des pièces de formes complexes</b> .....	3
Introduction.....	4
I. Processus de production des pièces de formes complexes .....	4
I.1. Apparition de l'idée .....	4
I.2. Conception « CAO » .....	5
I.3. Fabrication « FAO » .....	5
I.4. Usinage .....	5
II. Surfaces paramétriques et format d'échange de données.....	6
II.1. Surfaces paramétriques (modèles continus) .....	6
II.2. Format d'échange de données.....	6
III. Usinage des pièces complexes .....	9
III.1. Procédé d'usinage (fraisage) .....	9
III.2. Phases d'usinage .....	9
III.3. Outils de fraisage .....	9
III.4. Problèmes d'usinage .....	11
III.5. Stratégies d'usinage .....	12
III.6. Fraiseuses .....	14
IV. Généralités sur les techniques de simulation d'usinage .....	15
IV.1. Simulation d'usinage .....	15
Conclusion .....	19
<b>Chapitre 2 : Etude conceptuelle</b> .....	20
Introduction.....	21
I. Problématique et démarche proposée .....	21
I.1. Lecture et structuration du modèle STL .....	22
I.2. Génération des modèles de Triple-Dexel.....	23
I.3. Création des positions intermédiaires .....	26
I.4. Simulation d'enlèvement de matière .....	29
I.5. Vérification de la conformité des surfaces obtenues .....	33
II. Modélisation en UML .....	34
II.1. Diagramme de cas d'utilisation .....	35
II.2. Diagramme d'activité .....	39
II.3. Diagramme de classes .....	42
Conclusion.....	48

---

<b>Chapitre 3 : Implémentation informatique et validation</b> .....	49
Introduction .....	50
I. Présentation des langages utilisés .....	50
I.1. Présentation du langage C++ .....	50
I.2. Présentation d'OpenGL .....	50
I.3. Présentation d'Embarcadero C++ Builder 10 Seattle .....	51
II. Matériel utilisé .....	51
III. Présentation de l'environnement de CFAO .....	51
IV. Présentation de l'application .....	53
IV.1. Lecture du fichier STL .....	53
IV.2. Génération du modèle en Triple-Dexels .....	54
IV.3. Enrichissement du trajet d'outils .....	55
IV.4. Simulation d'enlèvement de matière .....	57
IV.5. Vérification de la conformité des surfaces résultantes .....	58
V. Test et validation .....	59
Conclusion .....	67
<b>Conclusion générale</b> .....	68
<b>Références bibliographiques</b> .....	70

## Chapitre 1 : Production des pièces de formes complexes

<b>Figure 1.</b> Processus de production des pièces complexes.....	4
<b>Figure 2.</b> Représentation d'un modèle STL.....	7
<b>Figure 3.</b> Fichier STL.....	8
<b>Figure 4.</b> Phases d'usinage.....	9
<b>Figure 5.</b> Formes d'outils.....	10
<b>Figure 6.</b> Positionnement des différentes formes d'outils.....	11
<b>Figure 7.</b> Types d'interférences.....	11
<b>Figure 8.</b> Stratégie d'usinage par « Plans Parallèles ».....	12
<b>Figure 9.</b> Types de parcours.....	12
<b>Figure 10.</b> Stratégie d'usinage par « Contours Décalés ».....	13
<b>Figure 11.</b> Stratégie d'usinage par « Plans Parallèles ».....	13
<b>Figure 12.</b> Stratégie d'usinage par surface guide.....	13
<b>Figure 13.</b> Stratégie isoparamétrique.....	14
<b>Figure 14.</b> Fraiseuse 05-axes.....	14
<b>Figure 15.</b> Représentation de volume par des Voxels.....	16
<b>Figure 16.</b> Modèle en Dexels.....	16
<b>Figure 17.</b> Paramètres d'un Dixel.....	17
<b>Figure 18.</b> Représentation en Dixel et en Triple-Dexels.....	17
<b>Figure 19.</b> Technique du Z-Buffer.....	18
<b>Figure 20.</b> Différents modes de balayage.....	18
<b>Figure 21.</b> Technique « point-vecteur ».....	19

## Chapitre 2 : Etude conceptuelle

<b>Figure 1.</b> Organigramme général de la démarche proposée.....	21
<b>Figure 2.</b> Lecture du fichier STL.....	22
<b>Figure 3.</b> Paramètres du brut.....	22
<b>Figure 4.</b> Génération des cellules.....	23
<b>Figure 5.</b> Génération des matrices.....	24
<b>Figure 6.</b> Génération des Dexels.....	24
<b>Figure 7.</b> Matrice des Dexels.....	25
<b>Figure 8.</b> Méthode des aires.....	26
<b>Figure 9.</b> Méthode barycentrique.....	26
<b>Figure 10.</b> Fichier du trajet d'outils.....	27
<b>Figure 11.</b> Repère associé à chaque forme d'outil.....	27
<b>Figure 12.</b> Enveloppes d'outils.....	27
<b>Figure 13.</b> Influence des positions intermédiaires sur la simulation.....	28
<b>Figure 14.</b> Intersection d'un segment avec l'outil hémisphérique.....	29
<b>Figure 15.</b> Intersection d'un segment avec l'outil cylindrique.....	31
<b>Figure 16.</b> Mise à jour des segments.....	33
<b>Figure 17.</b> Calcul des écarts.....	34
<b>Figure 18.</b> Diagramme de cas d'utilisation général.....	35

<b>Figure 19.</b> Diagramme de cas d'utilisation « Lire fichier STL » .....	36
<b>Figure 20.</b> Diagramme de cas d'utilisation « Génération du modèle en Triple-Dexel ».....	36
<b>Figure 21.</b> Diagramme de cas d'utilisation « Générer les matrices pour chaque plan » .....	37
<b>Figure 22.</b> Diagramme de cas d'utilisation « Calculer les points théoriques ».....	38
<b>Figure 23.</b> Diagramme de cas d'utilisation « Enrichir le trajet d'outil » .....	38
<b>Figure 24.</b> Diagramme d'activité général .....	39
<b>Figure 25.</b> Diagramme d'activité « Lire fichier STL ».....	39
<b>Figure 26.</b> Diagrammes d'activité « Calculer les points théoriques ».....	40
<b>Figure 27.</b> Diagrammes d'activité « Lire trajet d'outils » .....	40
<b>Figure 28.</b> Diagrammes d'activité « Générer les positions intermédiaires » .....	41
<b>Figure 29.</b> Diagrammes d'activité « Simuler l'enlèvement de matière » .....	41
<b>Figure 30.</b> Diagramme de classes général .....	42
<b>Figure 31.</b> Classe « Sommet ».....	43
<b>Figure 32.</b> Classe « Triangle » .....	43
<b>Figure 33.</b> Classe « Cellule » .....	44
<b>Figure 34.</b> Classe « model STL » .....	44
<b>Figure 35.</b> Classe « Segment » .....	45
<b>Figure 36.</b> Classe « SommetDexel » .....	45
<b>Figure 37.</b> Classe « Dexel » .....	46
<b>Figure 38.</b> Classe «TripleDexels » .....	46
<b>Figure 39.</b> Classe «Possition_Outil » .....	47
<b>Figure 40.</b> Classe «Déplacement » .....	47
<b>Figure 41.</b> Classe «Déplacement » .....	47

### Chapitre 3 : Implémentation informatique et validation

<b>Figure 1.</b> Environnement de « CFAO ».....	52
<b>Figure 2.</b> Lancement du module logiciel développé « Simulation usinage 5-axe ».....	52
<b>Figure 3.</b> Onglets de l'application développée .....	53
<b>Figure 4.</b> Onglet « Lecture du fichier STL ».....	54
<b>Figure 5.</b> Onglet « Générer le modèle en Triple-Dexels ».....	55
<b>Figure 6.</b> Onglet « Trajet d'outil ».....	56
<b>Figure 7.</b> Onglet « Simulation d'enlèvement de matière » .....	57
<b>Figure 8.</b> Onglet « Vérification » .....	58
<b>Figure 9.</b> Exemple 1.....	59
<b>Figure 10.</b> Lecture du fichier STL de la pièce .....	59
<b>Figure 11.</b> Paramètres visibles après la lecture du fichier .....	60
<b>Figure 12.</b> Création des cellules .....	60
<b>Figure 13.</b> Visualisation des sommets et des segments .....	61
<b>Figure 14.</b> Création des matrices .....	61
<b>Figure 15.</b> Visualisation des points théoriques .....	61
<b>Figure 16.</b> Lecture du fichier trajet d'outils .....	62
<b>Figure 17.</b> Visualisation du trajet d'outil et des points CL .....	62

<b>Figure 18.</b> Visualisation des déplacements et des positions intermédiaires .....	62
<b>Figure 19.</b> Visualisation de la pièce pendant la simulation .....	63
<b>Figure 20.</b> Simulation d'enlèvement de matière pour chaque plan .....	63
<b>Figure 21.</b> Extrémités des segments .....	64
<b>Figure 22.</b> Visualisation des écarts .....	64
<b>Figure 23.</b> Exemple 2 de test .....	65
<b>Figure 24.</b> Visualisation des paramètres .....	65
<b>Figure 25.</b> Visualisation des matrices et des points théoriques .....	65
<b>Figure 26.</b> Visualisation des déplacements et des positions intermédiaires .....	66
<b>Figure 27.</b> Visualisation des intersections dans le plan XY .....	66
<b>Figure 28.</b> Simulation d'enlèvement de matière globale .....	66
<b>Figure 29.</b> Visualisation des écarts .....	67

# INTRODUCTION GENERALE

## Contexte

Le présent travail s'inscrit dans le cadre du programme de recherche de l'équipe « CFAO » du Centre de Développement des Technologies Avancées « CDTA » qui traite les problèmes liés à la production automatisée des pièces complexes sur des fraiseuses numériques multi-axes. Il nous a été demandé de proposer et de concevoir une méthodologie pour la simulation de l'opération d'enlèvement de matière en 05-axes.

## 1. Problématique

Aujourd'hui, l'outil informatique est très utilisé dans la résolution des problèmes complexes car il offre l'efficacité, la fiabilité et la rapidité dans le traitement. Les industries automobile, aéronautique, mécanique, ...etc. sont parmi les domaines qui nécessitent son utilisation. Ces industries cherchent à réduire le temps et le coût de fabrication, tout en améliorant la quantité et la qualité des produits et les conditions de travail. 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. 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. Elles ne peuvent être usinées que sur des fraiseuses à commande numérique multi-axes (03-axes à 05-axes) en raison de leurs géométries très complexes.

La production des pièces mécaniques passe par deux importantes étapes qui sont : la conception et la fabrication. Dans la première étape, le modèle théorique (modèle CAO) de la pièce est généré en utilisant les différentes techniques de conception assistée par ordinateur. Dans la deuxième étape, la forme finale de la pièce est obtenue en trois opérations : 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. Ces opérations exigent la prise en compte simultanée de plusieurs paramètres (stratégies d'usinage, outils, vitesses d'avance, conditions de coupe, configuration de la machine, ...etc.). Avant d'aller directement à l'usinage réel sur machine, il est indispensable de passer par la simulation virtuelle de l'usinage afin de vérifier le trajet d'outils, de détecter les différents problèmes d'usinage (interférences et collisions), de prédire les efforts de coupe, la rugosité de la surface finie, ...etc.

Notre projet consiste à mettre en œuvre une approche pour valider virtuellement la conformité des surfaces résultantes après exécution des opérations d'usinage sur des pièces de formes complexes définies par leurs modèles « STL » sur des fraiseuses numériques à 05-axes.

### **2. 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 05-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 la simulation des différentes opérations d'usinage (ébauchage, demi-finition et finition) sur des pièces complexes discrètes en considérant les différentes formes d'outils (cylindriques et hémisphériques) tout en vérifiant la conformité des surfaces résultantes avec les modèles théoriques.

### **3. Structuration du mémoire**

Le présent mémoire est composé des chapitres suivants :

- Le premier chapitre est consacré à la présentation de l'état de l'art du processus de production des pièces de formes complexes. Une étude bibliographique des techniques d'approximation des modèles des pièces (Dxel, Triple-Dxel, etc.) et les méthodes de simulation virtuelle d'usinage en 05-axes sont également présentées dans ce chapitre.
- L'étude conceptuelle et les algorithmes utilisés sont présentés dans le deuxième chapitre.
- Le dernier chapitre présente l'application logicielle développée, les tests et la validation des résultats.

# CHAPITRE 1

---

## ETAT DE L'ART

Introduction.

1. Processus de production des pièces de formes complexes.
2. Surfaces paramétriques et format d'échange de données.
3. Usinage des pièces complexes.
4. Généralités sur les techniques de simulation d'usinage.

Conclusion.

## 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.

Dans ce chapitre, nous allons voir les stratégies d'usinage permettant de générer les trajectoires d'outils, les formes d'outils et leurs positionnements par rapport aux surfaces à usiner tout en mettant l'accent sur les problèmes d'usinage. Nous terminons par les techniques de simulation.

## I.Processus de production des pièces de formes complexes :

Le processus de production des pièces de formes complexes suit les phases suivantes :

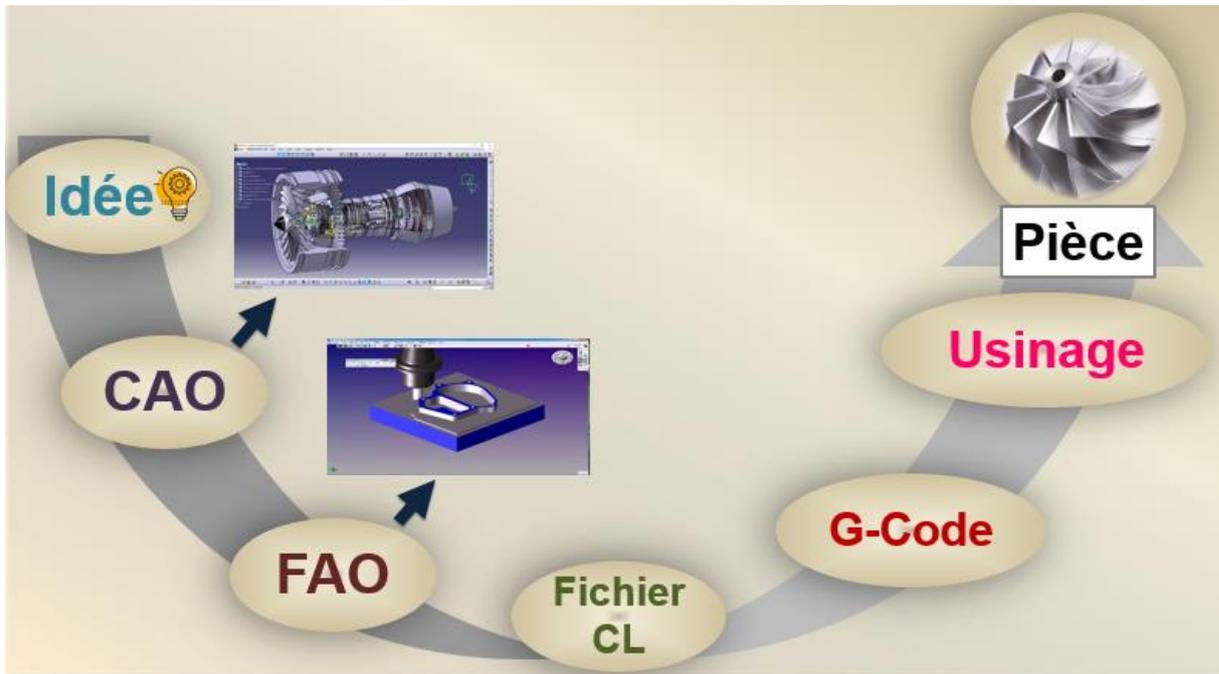


Figure 1. Processus de production des pièces complexes.

### I.1 Apparence d'idée :

Aujourd'hui, les industriels sont attachés à une matérialisation des besoins des clients. Pour qu'une entreprise réponde à ces besoins, elle doit proposer des produits et des

innovations pour se démarquer de la concurrence. Donc, l'apparence de l'idée vient suite à l'analyse des besoins. Elle donne naissance à une idée qui conduit à la définition des solutions.

## **I.2 Conception « CAO » :**

Pour réaliser cette idée, il faut faire une bonne conception. Aujourd'hui, la conception d'un objet nécessite à utiliser une solution numérique pour avoir des modèles tridimensionnels. Pour cela, des logiciels de CAO « Conception Assistée par Ordinateur » sont utilisés. La CAO est un ensemble d'outils et de programmes informatiques permettant d'assister l'ingénieur dans la conception d'un produit [1]. Les avantages liés à l'utilisation d'un logiciel de CAO sont :

- ✓ Conception plus réaliste et plus exacte.
- ✓ Amélioration de la qualité des produits.
- ✓ Modification et mise à jour très faciles.

On obtient à la fin un modèle géométrique (modèle CAO) qui est exporté par la suite vers le logiciel de FAO « Conception Assistée par Ordinateur » pour la fabrication.

## **I.3 Fabrication « FAO » :**

La FAO « Fabrication Assistée par Ordinateur » est un ensemble d'outils informatiques qui assiste l'ingénieur dans la mise sur pieds d'un processus de fabrication en créant un programme d'usinage, à partir du modèle CAO de la pièce, pour commander la machine d'usinage. En effet, un plan de fabrication correctement défini donne les résultats escomptés lors de la production [1]. Les principaux avantages de la FAO sont :

- ✓ Optimisation des équipements de production.
- ✓ Assistance dans la création, la vérification et l'optimisation des programmes d'usinage pour une productivité maximale.
- ✓ Génération d'un programme « G-Code » pour le pilotage d'une machine-outil à commande numérique. Ce fichier va décrire précisément les mouvements que doit exécuter la machine-outil pour réaliser la pièce demandée.

La combinaison entre CAO et FAO nous donne CFAO. Cette intégration permet de concevoir la forme d'une pièce mécanique puis de réaliser les diverses opérations d'usinage sur une machine-outil. Pour faire la fabrication réelle des produits, il faut utiliser des outils spéciaux afin d'obtenir des pièces complexes.

## **I.4 Usinage :**

C'est la phase de fabrication que l'on met en œuvre par de nombreuses techniques particulières qui varient en fonction de la forme que l'on souhaite donner à la matière, le type de la matière usinée, la taille des pièces à usiner et le nombre de pièces à réaliser (unitaire ou en série). Pour donner à une matière brute une forme précise, l'usinage fait appel à une technique complémentaire qui s'appelle le fraisage.

## II. Surfaces paramétriques et format d'échange de données :

### II.1 Surfaces paramétriques (modèle continue) :

Une surface paramétrique est une surface dans l'espace  $R^3$  définie par une fonction paramétrique avec deux paramètres ( $u$  et  $v$ ) qui appartiennent à l'intervalle  $[0,1]$  [2] :

$$\begin{cases} X = X(u, v) \\ Y = Y(u, v) \\ Z = Z(u, v) \end{cases} \quad (1)$$

$$P(u, v) = (X(u, v), Y(u, v), Z(u, v)) \quad (2)$$

### II.2 Format d'échange de données :

Deux types de formats d'échange de données existent.

#### II.2.1 Formats natifs (direct) :

Dans le cas où la CAO et la FAO sont complètement intégrées, c'est-à-dire environnement homogène, les informations passent d'une manière flexible (directement) et on a besoin d'aucune intermédiaire.

#### II.2.2 Formats neutres :

Dans l'autre cas où la CAO et la FAO ne sont pas intégrées, c'est-à-dire environnement hétérogène, le passage nécessite l'utilisation des intermédiaires qui s'appelle les translateurs (standards) pour éviter les différents problèmes tels que l'incompatibilité à différents niveaux : géométrie, erreurs de conversion où des pertes d'informations pendant l'échange de données entre les différents systèmes. Les formats neutres les plus utilisés sont décrits dans les paragraphes suivants.

➤ **Standard IGES (Initial Graphic Exchange Standard)** : il définit un format de données neutre qui permet l'échange de données, des fichiers graphiques et des informations entre les systèmes CAO. Ce format, malgré ses limitations, est l'un des plus répandus pour la visualisation ou l'export de modèles CAO 3D. Il représente des modèles filaires ou surfaciques. Si le modèle est solide, l'export vers un format IGES va permettre de ne sauvegarder que la « peau » extérieure du solide. Pour récupérer un solide, il faudra alors utiliser une fonction d'un modéleur permettant la génération d'un solide à partir de sa frontière surfacique [4].

- **Avantage** : le fichier IGES est l'un des formats les plus utilisés à la fois pour la visualisation et l'export des modèles CAO 3D. Il est directement importable dans les logiciels CAO. Le format IGES permet de procéder à l'export des modèles surfaciques ou filaires.
- **Inconvénient** : une interface de lecture et d'écriture difficile à supporter et elle ne constitue pas la solution optimale.

➤ **Standard STEP (Standard for the Exchange of Product model data) :** STEP est une norme standard intermédiaire qu'a été créée comme une amélioration d'IGES. Il permet la résolution de certains problèmes d'échange de données. L'objectif principal est de représenter sans aucune ambiguïté les données liées à l'ingénierie et aux pièces complexes dans un format neutre interprétable par tous les systèmes informatiques sans perte [3].

- **Avantages :** grâce au STEP, nous aurons la possibilité de récupérer la description complète et sans ambiguïté des produits industriels.
- **Inconvénients :** la norme STEP n'est pas toujours l'outil idéal. Son déploiement permet d'assurer une réelle communication des données par une interopérabilité entre les différents logiciels métiers.

➤ **Standard STL (Standard Tessellation Language) :** il permet de représenter la peau extérieure des objets tridimensionnels par des facettes triangulaires dont le nombre et la taille dépendent des tolérances d'approximation sans représentation de couleur, de texture ou d'autres attributs de modèles courants (Figure 2). L'idée de base était de paver la surface extérieure bidimensionnelle des modèles 3D à l'aide de minuscules triangles « facettes » et de stocker des informations sur les facettes dans un fichier. Ces fichiers sont générés en CAO. L'objectif du format de fichier STL est d'encoder la géométrie de surface d'un objet 3D. Il code cette information en utilisant un concept simple appelé « tessellation ». La tessellation est le processus de mosaïque d'une surface avec une ou plusieurs formes géométriques telles qu'il n'y ait pas de chevauchements ni de vides [5].

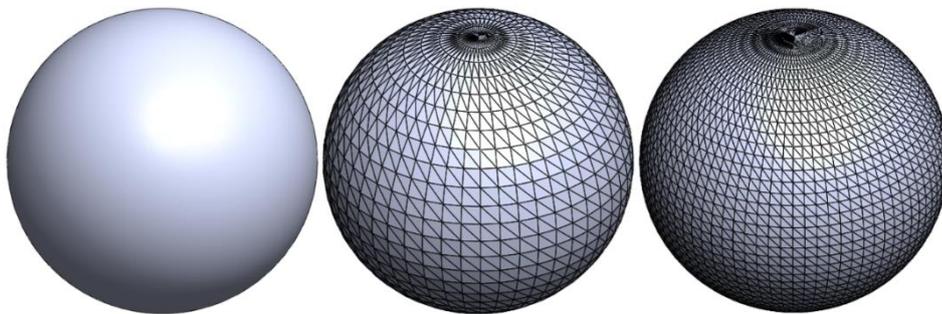


Figure 2. Représentation d'un modèle STL.

### Format d'un fichier STL :

Le format du fichier STL fournit deux codages pour stocker les informations des facettes triangulaires.

### Codage ASCII :

- ✓ Un fichier STL ASCII commence par la ligne : « **solid name** ».
- ✓ Les « **facet normal** » sont de simples triangles suivis par une boucle « **outer loop** ».
- ✓ Le fichier contient n'importe quel nombre de triangles.
- ✓ Le fichier se termine par « **endsolid name** » (Figure 3.a).

**Codage binaire :**

Un fichier STL binaire est architecturé de la façon suivante :

- ✓ Les 80 premiers octets sont un commentaire.
- ✓ Les 4 octets suivants forment un entier codé sur 32 bits représentant le nombre de triangles présents dans le fichier.
- ✓ Ensuite, chaque triangle est codé sur 50 octets, selon la décomposition suivante :
  - ❖ 3 fois 4 octets, chaque paquet de 4 octets représentent un nombre à virgule flottante correspondant aux composantes X, Y et Z de la normale du triangle.
  - ❖ 3 paquets de 3 fois 4 octets, chaque groupe de 4 octets représente un nombre à virgule flottante correspondant aux coordonnées X, Y et Z de chacun des sommets du triangle.
  - ❖ Deux octets représentant un mot de contrôle de 16 bits (Figure 3.b).

```

solid name
{
  facet normal  $n_i$   $n_j$   $n_k$ 
  outer loop
    vertex  $v_1$   $v_1$   $v_1$ 
    vertex  $v_2$   $v_2$   $v_2$ 
    vertex  $v_3$   $v_3$   $v_3$ 
  endloop
endfacet
}
endsolid name
    
```

a. Fichier STL ASCII.

Bytes	Data type	Description
80	ASCII	Header. No data significance.
4	unsigned long integer	Number of facets in file
4	float	$i$ for normal
4	float	$j$
4	float	$k$
4	float	$x$ for vertex 1
4	float	$y$
4	float	$z$
4	float	$x$ for vertex 2
4	float	$y$
4	float	$z$
4	float	$x$ for vertex 3
4	float	$y$
4	float	$z$
2	unsigned integer	Attribute byte count

b. Fichier STL binaire.

**Figure 3.** Fichier STL [8].

**Avantages :**

- ✓ Le format de fichier STL est universel.
- ✓ Les imprimantes 3D l'utilisent pour l'impression d'objets.

**Inconvénients :**

- ✓ Le nombre de triangles pour décrire les surfaces courbes génère des fichiers énormes.
- ✓ Le format de fichier STL ne peut pas stocker d'informations supplémentaires telles que la couleur, le matériau, etc.

### III. Usinage des pièces complexes :

#### III.1 Procédé d'usinage (Fraisage) :

Le fraisage est un procédé de fabrication où l'enlèvement de matière sous forme de copeaux résulte de la combinaison de deux mouvements : rotation de l'outil de coupe et avance de la pièce à usiner.

#### III.2 Phases d'usinage :

La forme finale d'une pièce est obtenue en trois phases d'usinage :

- ✓ **Ebauchage** : pour enlever le maximum de matière (Figure 4.a).
- ✓ **Demi-finition** : pour uniformiser la surépaisseur d'usinage.
- ✓ **Finition** : pour obtenir la forme finale avec un bon état de surface (Figure 4.b).



a. Ébauchage.



b. Finition.

Figure 4. Phases d'usinage.

#### III.3 Outils de fraisage :

##### III.3.1 Formes d'outils :

- **Hémisphériques** : sont des outils plus chers. Ils sont utilisés pour usiner les zones concaves (Figure 5.a).
- **Cylindriques** : sont des outils moins chers (Figure 5.b).
- **Toriques** : sont des outils intermédiaires entre les fraises cylindriques et les fraises hémisphériques (Figure 5.c).



Figure 5. Formes d'outils [6].

### III.3.2 Positionnement d'outil :

Au niveau de la phase de finition, l'outil quel que soit sa forme et ses dimensions, il doit être tangent à la surface en chaque point de contact. Cela dépend du vecteur normal unitaire au point de contact, de l'orientation de l'axe de l'outil et de la forme et des dimensions de l'outil.

Les différentes positions d'un outil hémisphérique sont données par (Figure 6.a) [7]:

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} \end{cases} \quad (3)$$

Les différentes positions d'un outil cylindrique sont données par (Figure 6.b) :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} = \overrightarrow{OC_C} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|} \end{cases} \quad (4)$$

Avec :  $\vec{k} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|}$

Les différentes positions d'un outil torique sont données par (Figure 6.c) :

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

Avec :  $\vec{k} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|}$

Avec :

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

$C_E$  : Le point centre de l'outil.

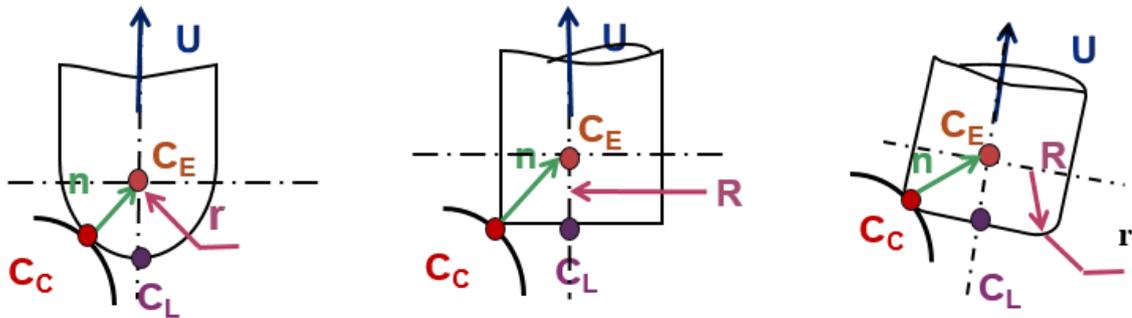
$C_L$  : Le point extrémité de l'outil.

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

$\vec{u}$ : 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.



a. Outil hémisphérique.

b. Outil cylindrique.

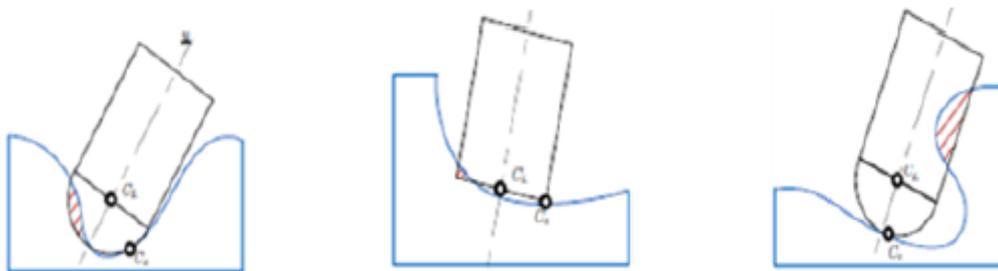
c. Outil torique.

Figure 6. Positionnement des différentes formes d'outils [7].

### III.4 Problème d'Usinage :

De nombreux problèmes peuvent survenir lors de l'usinage. Les problèmes les plus fréquents sont :

- ✓ **Interférence locale** : elles traduisent un enlèvement de matière en plus par la partie active de l'outil dans la surface à usiner (Figure 7.a) [8].
- ✓ **Interférence vers l'arrière** : elles sont des pénétrations indésirables de l'arrière de l'outil dans la surface à usiner (Figure 7.b) [8].
- ✓ **Interférence globale (collision)** : elles représentent des collisions entre l'ensemble (corps d'outil, porte outil, broche) et l'ensemble (pièce, porte pièce, etc...) (Figure 7.c) [8].



a. Interférence locale.

b. Interférence vers l'arrière.

c. Collision.

Figure 7. Types d'interférences [8].

### III.5 Stratégies d'usinage :

Une stratégie d'usinage est une méthodologie utilisée pour générer une série d'opérations pour réaliser une forme donnée. Elle permet d'associer un ensemble d'opérations comprenant la définition des outils, des conditions de coupe et des trajectoires d'usinage [9].

#### III.5.1 Stratégies d'ébauchage :

➤ **Plans parallèles :** elle s'appuie sur des trajectoires d'outil résultantes des intersections entre la surface à usiner et un ensemble de plans parallèles. [9].

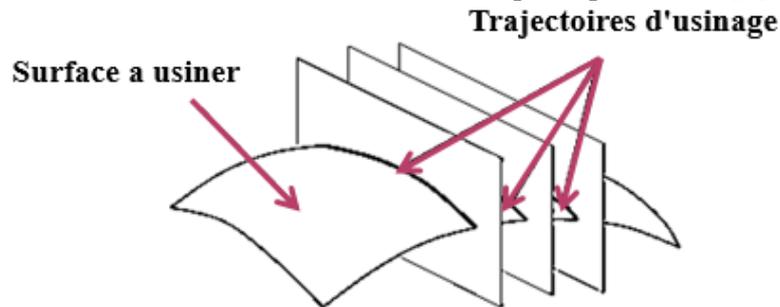
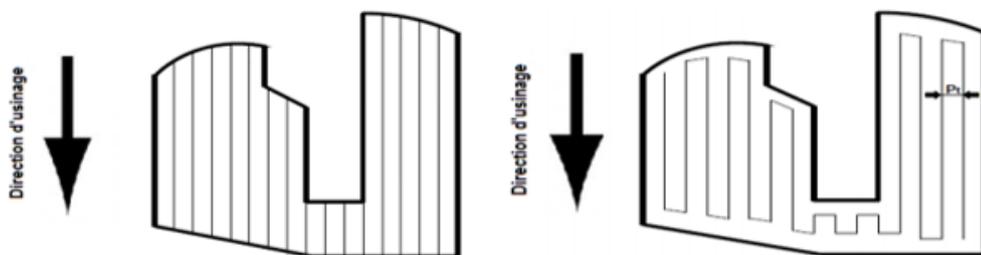


Figure 8. Stratégie d'usinage par plans parallèles [9].

✓ **Parcours de type One-Way :** l'outil est toujours sollicité de la même manière d'une passe à l'autre (Figure 9.a).

✓ **Parcours de type Zig-Zag :** l'avantage de cette méthode est la diminution du temps d'usinage en évitant les retraits hors matière (Figure 9.b) [9].



a. One-Way.

b. Zig-Zag.

Figure 9. Types de Parcours [9].

➤ **Conteur décalés :** elle consiste à générer les conteurs décalés à partir de la pièce (Figure 10) [10].

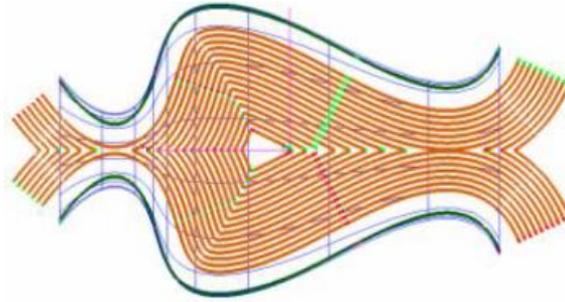


Figure 10. Stratégie d'usinage par « Contours décalés » [10].

### III.5.2 Stratégies de demi-finition et finition :

➤ **Plan parallèle** : elle s'appuie sur des trajectoires résultantes des intersections entre la surface à usiner et un ensemble de plans parallèles. Ces plans parallèles sont caractérisés par une normale commune dans une direction donnée. (Figure 11) [9].

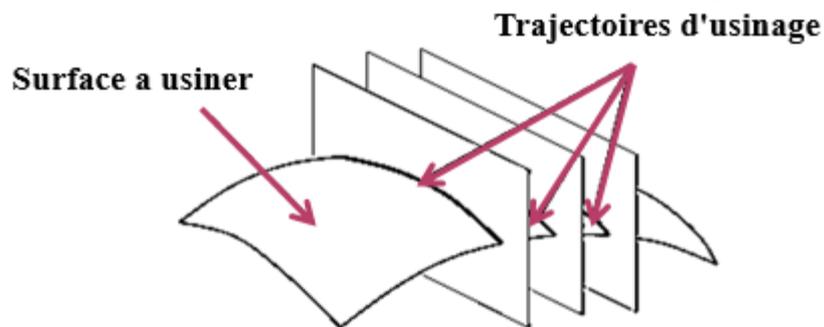


Figure 11. Stratégie d'usinage par plans parallèles [9].

➤ **Surface guide** : cette stratégie définit la trajectoire de l'outil à partir d'une surface intermédiaire appelée surface guide. L'outil se déplace suivant une direction donnée en restant en contact avec la surface à usiner et la surface guide (Figure 12) [9].

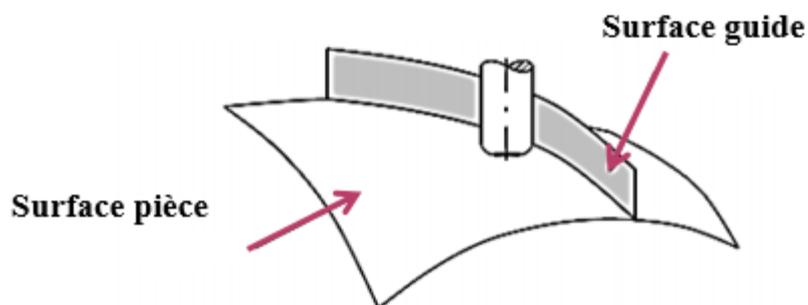


Figure 12. Stratégie d'usinage par surface guide [9].

➤ **Iso paramétrique** : elle s'appuie sur le plan paramétrique de la surface pour générer le trajet d'outil avec plusieurs modes (Figure 13) [9].

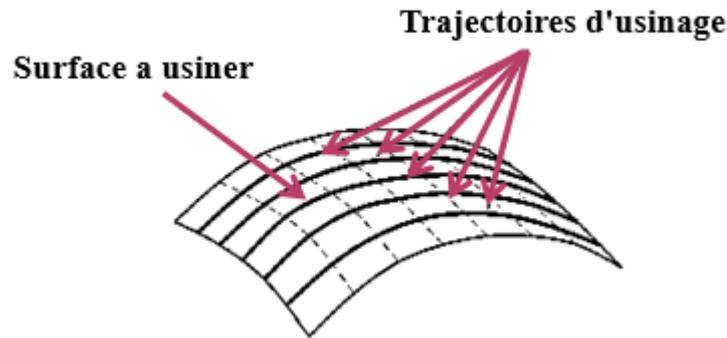


Figure 13. Stratégie iso paramétrique [9].

### III.6 Fraiseuses :

Une fraiseuse est une machine-outil utilisée pour usiner tous types de pièces mécaniques, à l'unité ou en série, par enlèvement de matière à l'aide d'un outil coupant nommé fraise. Les fraiseuses à commande numérique peuvent répéter les mouvements donnés une fois par l'opérateur (enregistrement des mouvements) pour usiner la pièce. Les fraiseuses numériques peuvent être de différentes configurations :

- ✓ Fraiseuse numérique à 03-axes : c'est fraiseuse pouvant effectuer trois (03) translations simultanées suivant les axes X, Y et Z. Elle contient une broche verticale ou une broche horizontale. Pour une fraiseuse verticale, la fraise est perpendiculaire à la table.
- ✓ Fraiseuse numérique à 05-axes : elle comporte toujours trois (03) axes linéaires (X, Y et Z) et deux (02) axes rotatifs à choisir parmi les axes (A, B et C) qui peuvent varier suivant la configuration de la machine (Figure 14). Tous les axes sont synchronisés.

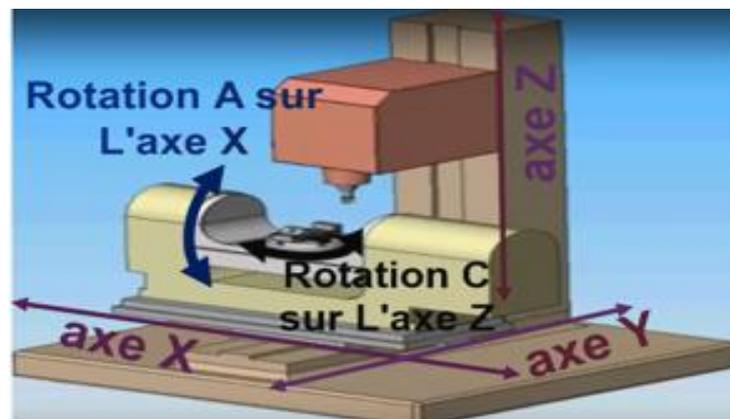


Figure 14. Fraiseuse 05-axes.

#### Avantage des fraiseuses 05-axes :

- ✓ Facilité de l'usinage des pièces à 05 faces.
- ✓ Usinage continu sur les grandes surfaces.
- ✓ Obtention d'une surface et des pièces de très bonne qualité.
- ✓ Economie de temps et d'argent.

**Inconvénient des fraiseuses 05-axes :**

- ✓ Difficulté du calcul de la position de l'outil.
- ✓ Problèmes d'interférence et de collision plus fréquents.
- ✓ Usinage 05 axes plus complexe.

**IV.Généralité sur les techniques de simulation d'usinage :**

**IV.1 Simulation d'usinage :**

C'est une opération virtuelle qui s'effectue après la phase FAO et avant de passer à l'usinage réel sur une machine-outil. Elle s'appelle aussi l'usinage virtuel. La simulation du fraisage est une opération très importante permettant de faire toutes les actions de fraisage de façon virtuelle (mouvement de l'outil, enlèvement de la matière et détection des erreurs) pour réduire le temps d'usinage effectif ainsi que la validation des trajets d'usinage en détectant les différentes erreurs et leurs emplacements afin de les corriger [9].

**IV.1.1 Catégories de simulation d'usinage :**

**Simulation du trajet d'outil :**

C'est un type de simulation qui permet :

- ✓ La vérification du trajet d'outil.
- ✓ La détection des collisions.

**Simulation d'enlèvement de matière :**

C'est un type de simulation qui permet de traiter le problème de prédiction de la topographie de la surface où la quantification de la matière enlevée est indispensable. Elle permet de prédire :

- ✓ La qualité des surfaces usinées.
- ✓ Les efforts de coupe.

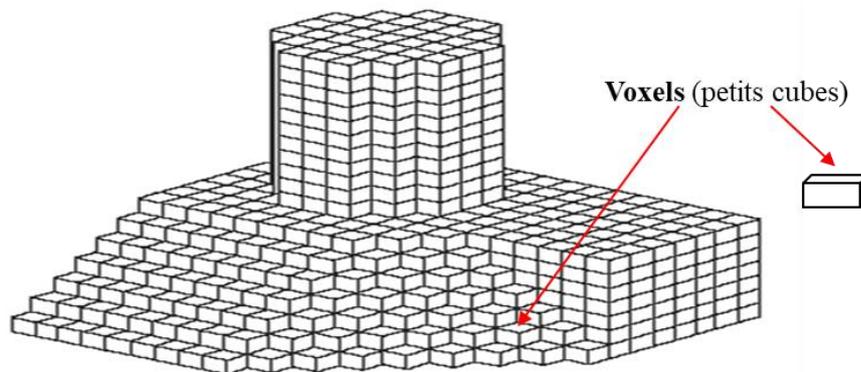
Dans notre travail, on s'intéresse à la simulation d'enlèvement de matière.

**IV.1.2 Modèles géométrique de la pièce :**

Les pièces sont modélisées par des volumes élémentaires en utilisant des méthodes de discrétisation telles que Dexels, Voxels et Triple-Dexels. Ces modèles peuvent être continus ou discrets. Ces derniers peuvent aller du plus simple, une série de points, au plus complexe, description facétisée de la surface ou volumique, représentation à base espace image, Dixel, Voxels et plus avancé le modèle Triple-Dexels.

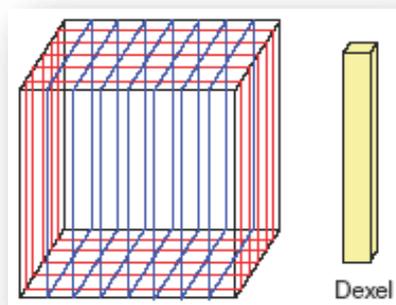
➤ **Voxel :** les solides sont représentés par une liste de Voxels (éléments de volume). Les Voxels sont des cubes de taille fixe répartis selon une grille. En général, le solide est défini par la liste des coordonnées des centres des cellules. Les Voxels permettent de faciliter la validation, de simplifier l'accès à un point donné et d'assurer l'unicité spatiale. Par contre,

ils nécessitent énormément de ressources pour le stockage que pour le rendu qui ne bénéficie pas d'accélération matérielle (Figure 15) [11].



**Figure 15.** Représentation de volume par des Voxels [11].

➤ **Dexels** : ils peuvent modéliser des pièces de forme complexe ayant des trous où des surfaces courbes et ondulées. Le modèle en Dexels permet d'approximer un solide par un ensemble de parallélépipèdes appelé Dexels orientés selon l'un des trois directions principales (X, Y et Z) dont le nombre et la taille dépendent des précisions imposées (Figure 16). Un Doxel est un pixel avec profondeur. Chaque Doxel est défini le long de l'axe Z qui est une colonne de matière parallèle à cet axe, par un centre (X0, Y0) et par une hauteur H (Z min et Z max) de base carrée ou rectangulaire ( $\Delta X$  et  $\Delta Y$ ) (Figure 17). Les valeurs de base  $\Delta X$ ,  $\Delta Y$  et  $\Delta Z$  sont les paramètres qui déterminent la précision du modèle. Selon la forme de l'objet à modéliser, plusieurs Dexels peuvent être créés sur la même ligne droite [11].



**Figure 16.** Modèle en Dexels [11].

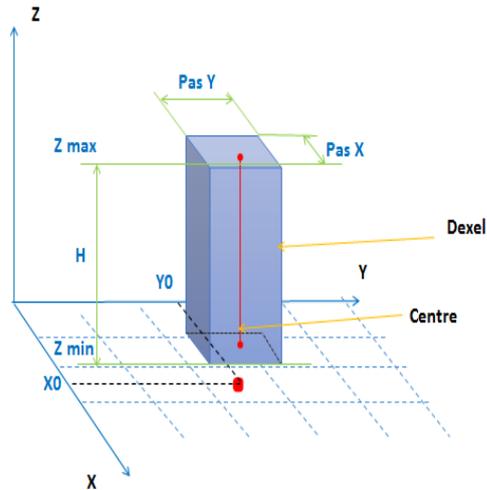


Figure 17. Paramètres d'un Dixel [11].

➤ **Triple-Dixels** : c'est une extension de la modélisation simple Dixel. Ce modèle est construit par des rayons orientés suivant les trois directions orthogonales X, Y et Z pour discrétiser le modèle (Figure 18). En raison de ses opérations booléennes rapides, de sa structure de données simple et de sa mise en œuvre facile, la modélisation Triple-Dixels est très adaptée aux applications de simulation graphiques en temps réel telles que la vérification de l'usinage numérique et la sculpture virtuelle [11].

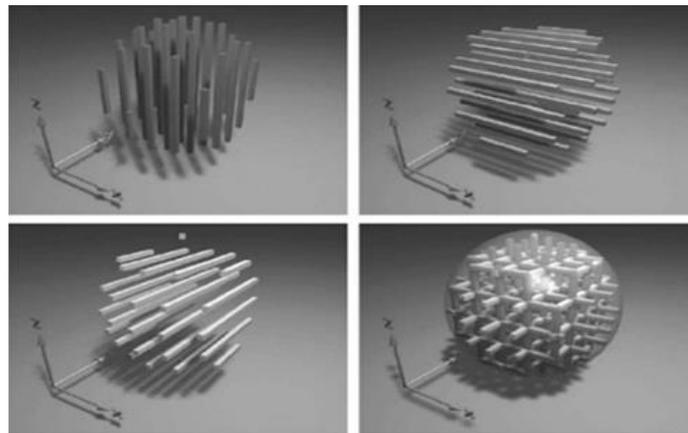


Figure 18. Représentation en Dixel et en Triple-Dixels [11].

### IV.1.3 Techniques de simulation d'usinage :

#### IV.1.3.1 Technique de Z-Buffer :

Cette technique passe par la construction d'une grille dans un plan XY. A chaque point de la grille, on associe un segment vertical d'une altitude z initiale. Lors de l'usinage, on calcule l'intersection des segments avec le trajet d'usinage (Figure 19). Pour chaque segment, on retient l'altitude d'intersection la plus basse. Cette méthode est très simple mais la précision dépend du pas de la grille. 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. On obtient

alors par soustraction des différents Z-Buffer entre eux, les erreurs d'usinage, les zones non usinées, les volumes de matière enlevée et les collisions [12].

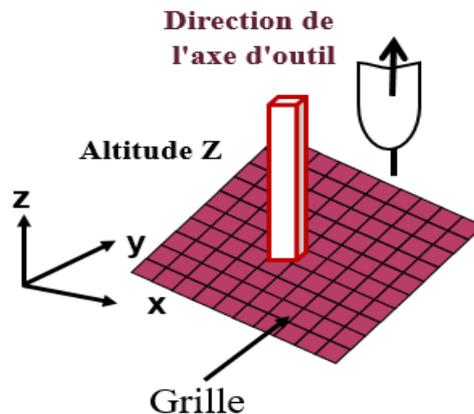


Figure 19. Technique du Z-Buffer [12].

#### IV.1.3.2 Technique de représentation Solide-CSG :

C'est une technique de simulation basée sur la représentation solide CSG (Constructive Solid Geometry) et s'appuie sur la modélisation solide de l'enveloppe du trajet de l'outil (Figure 20). La simulation est effectuée en créant les volumes générés par les mouvements de l'outil sous forme CSG, puis en contrôlant s'il y a intersection entre le modèle de la surface nominale et chaque volume balayé par l'outil [12].

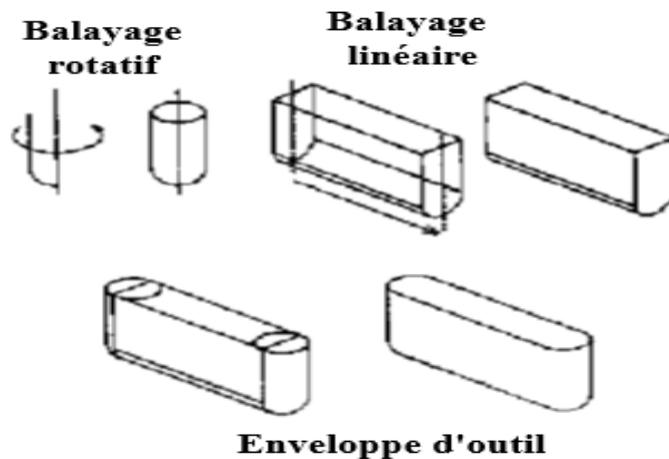


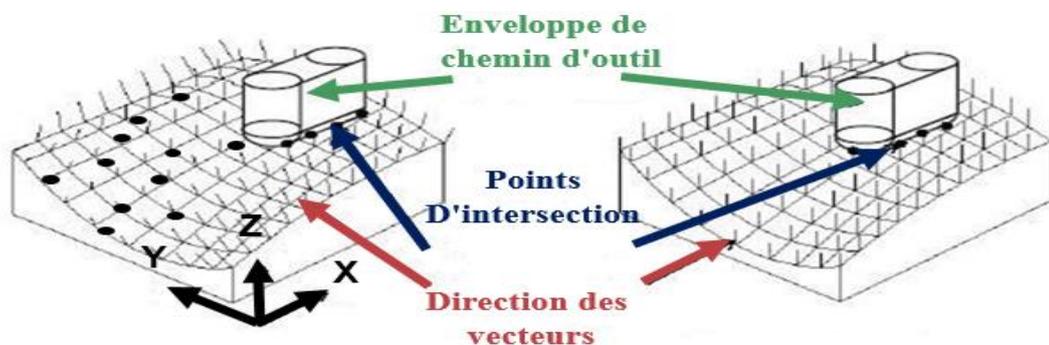
Figure 20. Représentation solide du volume balayé par l'outil [12].

#### IV.1.3.3 Technique du point-vecteur :

La technique « point-vecteur » 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, puis 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. Pour mettre en place cette technique il faut :

- ✓ Discrétiser la surface.
- ✓ Orienter les vecteurs.
- ✓ Modéliser le trajet d'usinage.

Un ensemble « point-vecteur » est construit sur la surface où 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. Le sens du vecteur indique la surépaisseur de matière laissée par rapport à la surface nominale après le passage de l'outil. A chaque point, on associe un vecteur. On peut donner à ce vecteur, une direction normale à la surface (Figure 21.a). Mais pour simplifier les calculs et ne considérer que la zone qui est sous l'outil, on préfère donner une direction parallèle à l'axe de l'outil (axe Z) (Figure 21.b) [13].



a. Orientation suivant les normales.

b. Orientation suivant l'axe Z.

**Figure 21.** Technique « point-vecteur » [12].

## Conclusion :

Dans ce chapitre, nous avons présenté d'une manière générale le processus de production des pièces complexes en mettant l'accent sur le modèle « STL » support de notre étude. Par la suite, nous avons explicité le processus d'usinage des pièces complexes sur des fraiseuses numériques à 05-axes et les différentes stratégies d'usinages. Nous avons terminé ce chapitre par la présentation des différentes techniques de modélisation des pièces et des techniques de simulation d'usinage.

Dans notre travail, on s'intéresse à la simulation d'enlèvement de matière en modélisant la pièce par des Triple-Dexels pour vérifier la conformité des pièces usinées.

# CHAPITRE 2

---

## ETUDE CONCEPTUELLE

Introduction.

1. Démarche proposée.
2. Modélisation de l'application avec UML.

Conclusion.

### Introduction :

En informatique, la résolution d'un problème exige sa décomposition en une suite d'étapes successives pour ensuite donner une solution adéquate.

L'objectif de notre travail est la conception d'une application logicielle graphique et interactive permettant la simulation des différentes opérations d'usinage (ébauchage, demi-finition et finition) sur des pièces complexes discrètes en considérant les différentes formes d'outils (cylindrique et hémisphériques). Dans ce qui suit, les différentes étapes de la solution proposée et la conception de l'application seront présentées.

### I. Problématique et démarche proposée :

L'usinage 5-axes permet l'obtention des pièces avec une bonne finition. Le passage direct à l'usinage réel génère une perte de la matière première et du temps d'usinage avec la possibilité de rencontrer des problèmes lors de l'usinage tels que les interférences et les collisions. Pour cela, il est primordial de virtualiser l'usinage réel avant de passer à l'usinage effectif sur machine afin de détecter toutes les erreurs possibles.

L'étape logique de résolution d'un problème d'informatique donné, est la décomposition de celui-ci en des parties cohérentes et simples à résoudre. Dans notre cas, les différentes étapes suivies sont données par l'organigramme général suivant (Figure 1).

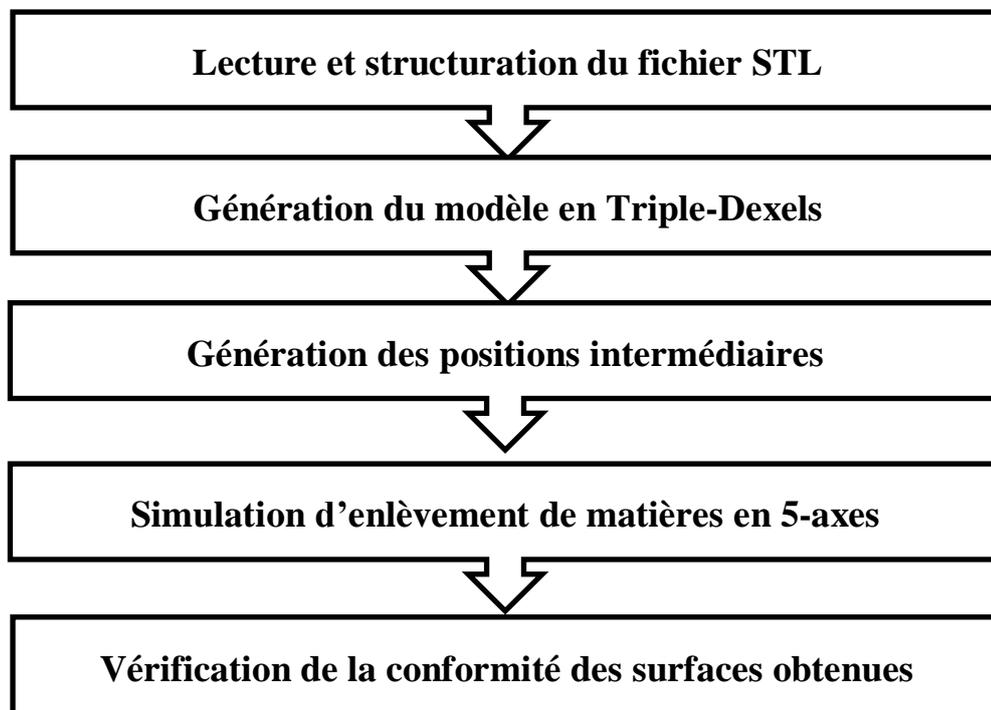


Figure 1. Organigramme général de la démarche proposée.

## I.1 Lecture et structuration du modèle STL :

Le fichier STL décrivant la pièce à usiner doit être analysé. Cette analyse consiste à vérifier son extension et sa syntaxe.

### I.1.1 Vérification du format STL et définition du brut :

Dans cette étape, le fichier STL est vérifié et son brut est défini (Figure 2). Pour cela, les informations concernant les triangles, leurs sommets leurs normales sont sauvegardés dans un nouveau fichier pour éviter de lire à chaque fois les mots inutile tels que « facet » et « outerloop », etc.

La définition du brut passe par la détermination des coordonnées Xmin, Xmax, Ymin, Ymax, Zmin et Zmax de ses points limites afin de calculer ses dimensions (Figure 3).

$$\begin{cases} \text{Longueur} = X_{\max} - X_{\min} \\ \text{Largeur} = Y_{\max} - Y_{\min} \\ \text{Hauteur} = Z_{\max} - Z_{\min} \end{cases} \quad (1)$$

```
facet normal -0.78092157 0.61690771 0.097910043
  outer loop
    vertex 26.06 84.144 -3.19132e-15
    vertex 13.143443 68.323504 -3.340128
    vertex 13.56222 68.323504 -1.6608369e-15
  endloop
endfacet
```

a. Fichier STL.

```
-0.78092157 0.61690771 0.097910043 26.06 84.144 -3.19132e-15 13.143443
```

b. Fichier texte.

Figure 2. Lecture du fichier STL.

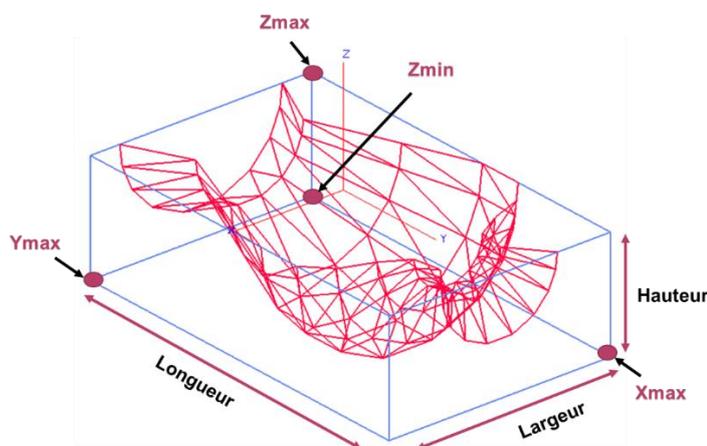


Figure 3. Paramètres du brut.

### I.1.2 Récupération des paramètres :

Les paramètres à récupérer à partir du nouveau fichier sont la liste des triangles et la liste des sommets. Le problème qui se pose est que la liste des sommets doit contenir les sommets sans répétition sinon la taille de la liste des sommets devient très volumineuse. Pour éviter ces répétitions, il faut toujours vérifier si le sommet récupéré est déjà sauvegardé ou non. Pour minimiser cette opération nous avons proposé la création de cellules par la subdivision du brut en blocs parallélépipédiques de mêmes dimensions (Figure 4). Ces cellules sont créées après la spécification du nombre de cellules suivant les axes X, Y et Z. Par la suite, à chaque cellule sont associés les indices des sommets et les indices triangles. De cette manière, la recherche de l'existence d'un sommet sera menée exclusivement dans une cellule donnée ce qui réduit considérablement les temps de calcul.

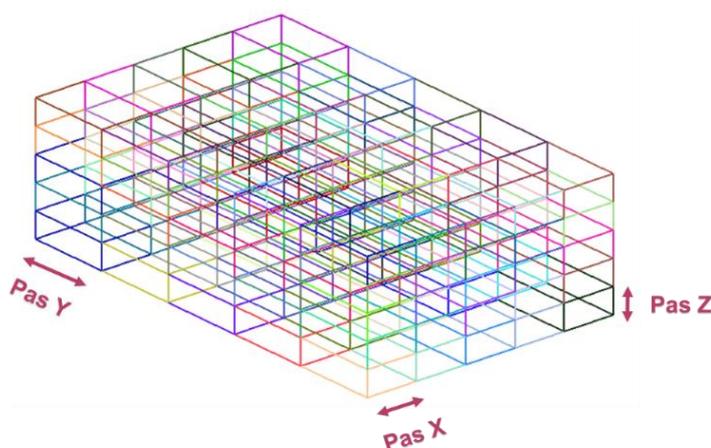


Figure 4. Génération des cellules.

## I.2 Génération des modèles de Triple-Dexel :

### I.2.1 Génération des matrices :

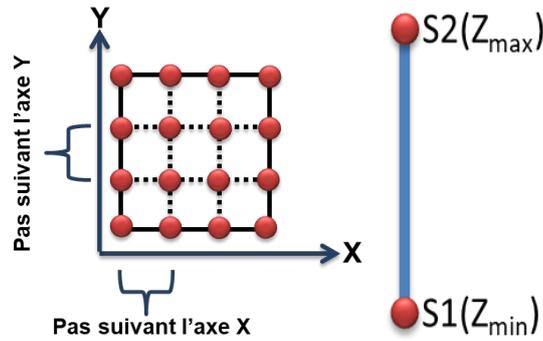
Pour approximer le modèle STL par des Triple-Dexels, il faut commencer par la création des matrices bidimensionnelles dans les plans XY, XZ et YZ. Ces matrices servent à stocker les sommets des Dexels et les segments. Dans ce qui suit, nous allons montrer la création de ces matrices dans le plan XY.

**Dans le plan XY :** pour créer cette matrice il faut introduire les pas (pas\_x et pas\_y) suivant les axes X et Y (Figure 5.a). Les coordonnées d'un sommet Dexel d'indices I et J suivant les axes X et Y sont données par :

$$\begin{cases} X = X_{\min} + I * \text{pas}_x \\ Y = Y_{\min} + J * \text{pas}_y \\ Z = Z_{\min} \end{cases} \quad (2)$$

A chaque sommet Dixel de la matrice est associé un segment défini par deux sommets (Figure 5.b) :

- Le premier est le sommet Dixel lui-même.
- Le deuxième sommet a les mêmes coordonnées X, Y que le sommet Dixel mais sa coordonnée z est égale à  $Z_{max}$  du brut.

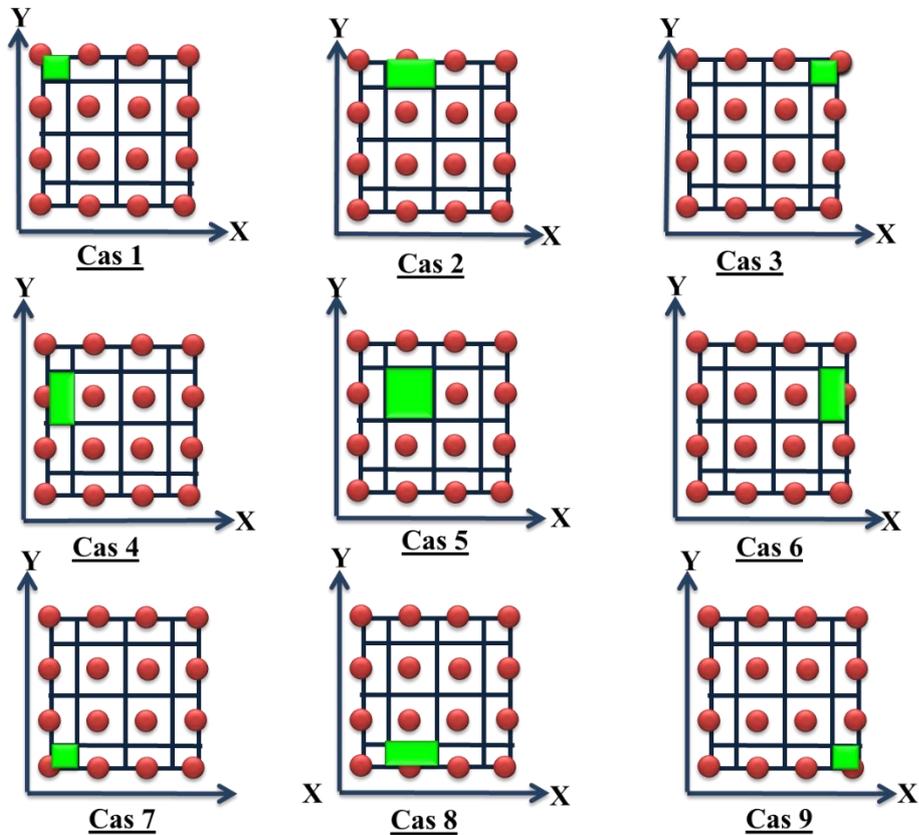


a. Matrice des sommets Dixel.

b. Segment du Dixel.

**Figure 5.** Génération des matrices.

Une fois la matrice est créée, les Dixels peuvent être créés. Pour cela, neuf (09) cas de figures sont à considérer en fonction de la position du sommet Dixel (Figure 6). La Figure 7 montre la matrice des Dixels initiale.



**Figure 6.** Génération des Dixels.

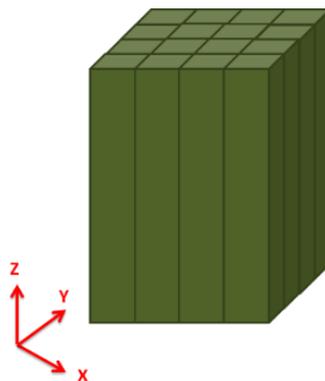


Figure 7. Matrice des Dexels.

Les mêmes étapes sont suivies pour la création des matrices dans les plans XZ et YZ. Dans le plan XZ, le segment est créé suivant l'axe Y tandis que pour le plan YZ le segment est créé suivant l'axe X.

### I.2.2 Calcul des points théoriques :

Le point théorique est le point d'intersection du segment associé au sommet Dixel avec un des triangles du modèle STL de la pièce. Les étapes de calcul sont :

- Détermination de la cellule qui contient le sommet Dixel.
- Identification de tous les triangles appartenant à cette cellule.
- ❖ Pour chaque triangle on va tester :
  - Si le sommet Dixel appartient à l'enveloppe du triangle.
    - Si oui, remplacer les coordonnées x, y du sommet Dixel dans l'équation du plan passant par le triangle pour calculer la coordonnée z du point d'intersection.
      - Si le point calculé appartient au triangle en utilisant soit la méthode des aires (Figure8) ou la méthode barycentrique (Figure9).
      - ✓ Si ce point appartient au triangle, il est inséré dans la liste des points théoriques associé au sommet Dixel.
- Pour chaque sommet Dixel, les points théoriques sont triés dans un ordre croissant en fonction du plan de calcul. Cette étape est nécessaire pour faciliter les calculs de vérification de la conformité des surfaces résultantes.

**Méthode des aires :** un point M appartient au triangle  $P_1P_2P_3$  si la condition suivante est vérifiée (Figure 8) :

$$\text{Aire}(P_1P_2P_3) = \text{Aire}(P_1MP_3) + \text{Aire}(P_1MP_2) + \text{Aire}(P_2MP_3) \quad (3)$$

**Méthode barycentrique :** un point M appartient au triangle  $P_1P_2P_3$  si les conditions suivantes sont vérifiées (Figure 9) :

$$\overrightarrow{P_1M} = \alpha * \vec{U} + \beta * \vec{V} \quad (4)$$

$$\begin{cases} \vec{U} = \overrightarrow{P_1P_2} \\ \vec{V} = \overrightarrow{P_1P_3} \\ 0 \leq \alpha \leq 1 \\ 0 \leq \beta \leq 1 \\ 0 \leq \alpha + \beta \leq 1 \end{cases} \quad (5)$$

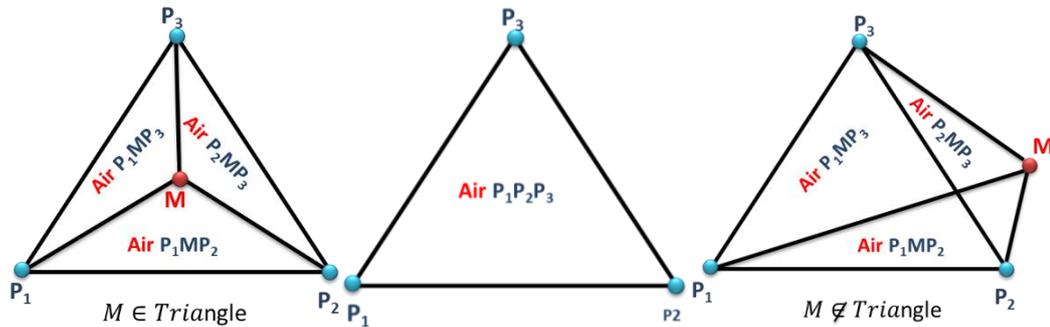


Figure 8. Méthode des aires.

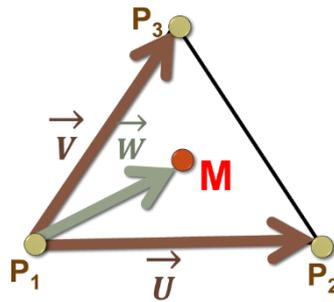


Figure 9. Méthode barycentrique.

### I.3 Création des positions intermédiaires :

Avant de passer à la simulation d'enlèvement de matière, il faut lire le fichier contenant le trajet d'outils (Figure 10). Dans ce fichier, la première ligne indique le nombre total des positions d'outils (points CL). Par la suite, sur chaque ligne sont donnés les paramètres associés à chaque position CL. Pour chaque position d'outil CL, est associé un repère local dont l'origine est le point CL et d'axes unitaires les vecteurs tangents ( $T_1$  et  $T_2$ ) et le vecteur axe d'outil (Figure 11). De plus, les enveloppes d'outil sont calculées pour chaque plan (Figure 12).

100228

Forme d'outil	Nombre des position			Point de contact « CL »			L'axe d'outil		
	R	r	L	Coordonne X	Coordonne Y	Coordonne Z	$U_x$	$U_y$	$U_z$
cylindrique	5	0	150	286.9849	225.05435	394.06352	0.21623683	0.062004914	0.97437006
cylindrique	5	0	150	240.43077	211.70516	184.28913	0.21623683	0.062004914	0.97437006
cylindrique	5	0	150	239.96348	211.57117	182.18353	0.21623683	0.062004914	0.97437006

Figure 10. Fichier du trajet d'outils.

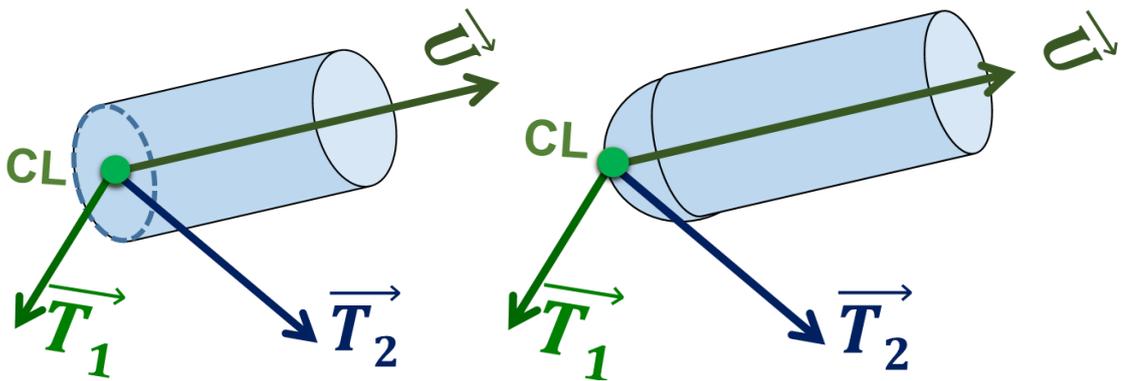
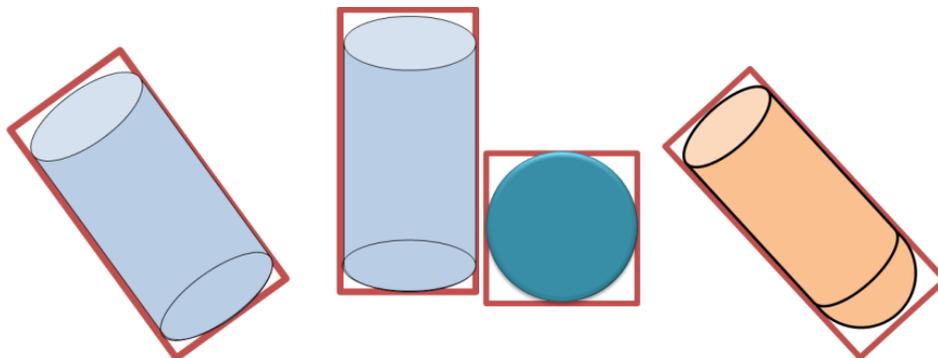


Figure 11. Repère associé à chaque forme d'outil.



a. Enveloppe d'outil cylindrique.

b. Enveloppe d'outil hémisphérique.

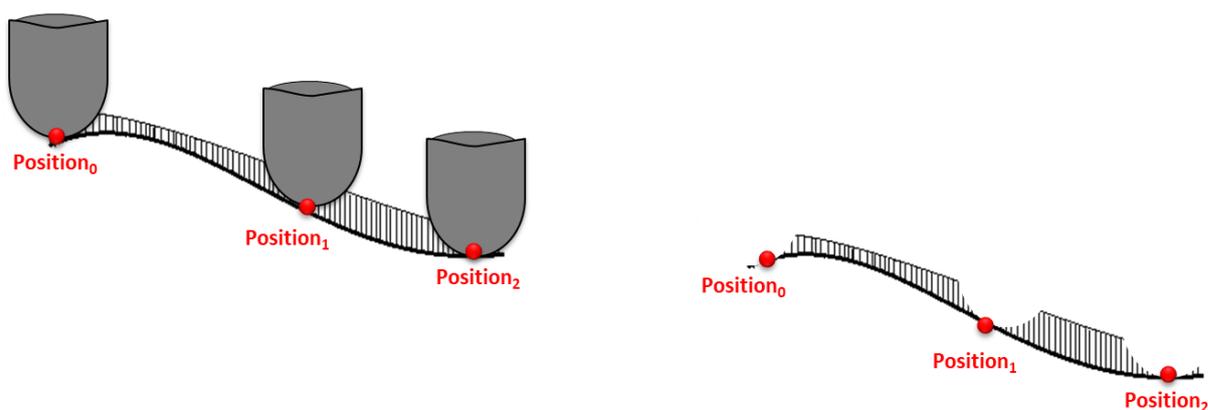
Figure 12. Enveloppes d'outils.

Si dans la simulation d'enlèvement de matière sont considérés uniquement les points CL, le résultat peut ne pas être correct si la distance entre deux points CL consécutifs est grande. Donc, pour éviter ce problème et pour s'approcher du mouvement continu de l'outil, il est nécessaire d'insérer des positions CL intermédiaires pour chaque déplacement de l'outil (Figure 13). Pour cela, il faut créer la liste des déplacements. Un déplacement est créé pour deux points CL successifs qui ont les mêmes paramètres d'outil. La création des positions intermédiaires requiert la spécification d'un incrément (distance maximale) entre chaque deux positions intermédiaires.

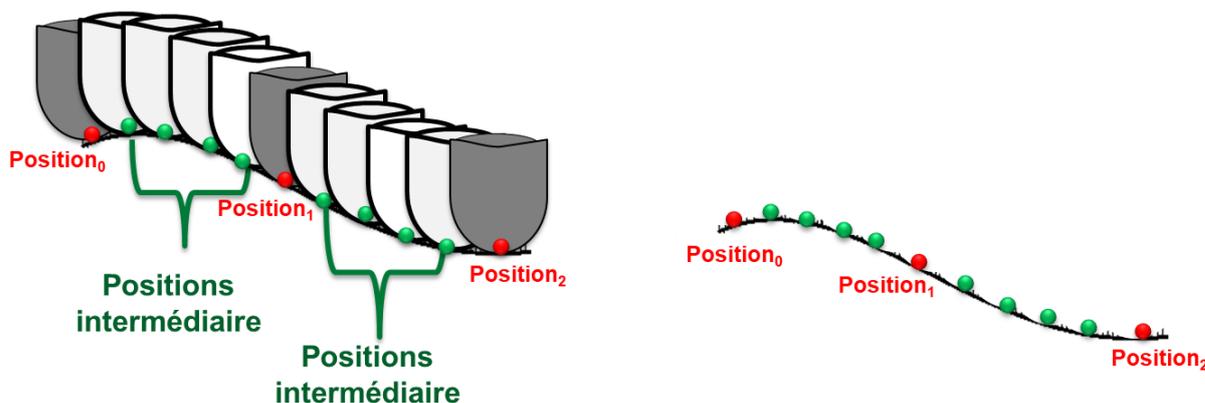
➤ Pour chaque déplacement :

- Si l'incrément est supérieur à la distance de déplacement, alors insérer les points extrémités du déplacement comme des positions intermédiaires.
- Sinon :
  - ❖ Insérer le premier point extrémité du déplacement s'il n'existe pas.
  - ❖ Créer les positions intermédiaires et les insérer.
  - ❖ Insérer le deuxième point extrémité du déplacement s'il n'existe pas.

Notons qu'en usinage 05-axes nous avons deux rotations. Donc, pour chaque nouvelle position intermédiaire il faut calculer les coordonnées du point CL, les vecteurs tangents ( $T_1$  et  $T_2$ ), le vecteur axe d'outil et les enveloppes d'outils dans les trois (03) plans.



a. Sans positions intermédiaires.



b. Avec positions intermédiaires.

**Figure 13.** Influence des positions intermédiaires sur la simulation.

## I.4 Simulation d'enlèvement de matière :

L'objectif de notre travail est de simuler l'enlèvement de matière sur des fraiseuses 05-axes. Cette simulation nécessite le calcul, pour chaque position intermédiaire, les intersections entre les segments issus des sommets Dexels avec les outils. Dans ce projet, les formes d'outils considérées sont les outils hémisphériques et les outils cylindriques.

### I.4.1 Intersection avec l'outil hémisphérique :

Le calcul du point d'intersection de l'outil hémisphérique avec un segment revient à calculer l'intersection entre une sphère et une droite (Figure 14).

Dans le plan XY : les coordonnées x et y du segment sont fixés. Donc, la coordonnée z du point d'intersection est obtenue suite en remplaçant les coordonnées x et y dans l'équation de la sphère. Cette dernière est donnée par :

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \quad (6)$$

Avec :

R : rayon de la sphère.

$(x_0, y_0, z_0)$  : origine de la sphère.

Pour l'outil hémisphérique, les coordonnées du centre de la sphère sont données par :

$$\begin{cases} x_0 = CL_x + R * U_x \\ y_0 = CL_y + R * U_y \\ z_0 = CL_z + R * U_z \end{cases} \quad (7)$$

La résolution de l'équation (6) donne deux valeurs pour la coordonnée z des points d'intersection :

$$z_0 = z_0 + \sqrt{R^2 - (x - x_0)^2 - (y - y_0)^2} \quad (8)$$

$$z_1 = z_0 - \sqrt{R^2 - (x - x_0)^2 - (y - y_0)^2} \quad (9)$$

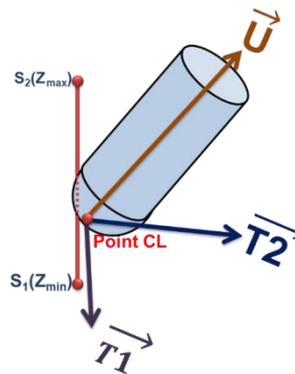


Figure 14. Intersection d'un segment avec l'outil hémisphérique.

### I.4.2 Intersection avec l'outil cylindrique :

Le calcul du point d'intersection de l'outil cylindrique avec un segment revient à calculer l'intersection entre un disque et un cylindre avec une droite (Figure 15).

**Dans le plan XY :** les coordonnées x et y du segment sont fixes.

➤ Calcul de l'intersection avec le disque : le calcul du point d'intersection entre le disque et le segment revient à remplacer l'équation de la droite dans l'équation du plan passant par le disque qui est donnée par :

$$U_x * x + U_y * y + U_z * z + D = 0 \quad (10)$$

$$D = -(U_x * x + U_y * y + U_z * z) \quad (11)$$

Avec :

$U_x, U_y, U_z$  : les composantes du vecteur axe d'outil U.

x, y, z : les coordonnées du point extrémité d'outil CL.

En remplaçant dans l'équation (10) les coordonnées x, y du sommet Dixel, la coordonnée  $z_0$  du point d'intersection théorique du segment avec le disque est donnée par :

$$z_0 = -(U_x * x + U_y * y + D) / U_z \quad (12)$$

Le point calculé n'est considéré un point d'intersection valide que si sa distance au point extrémité d'outil CL est inférieure ou égale au rayon du disque (rayon d'outil).

➤ Calcul d'intersection avec la partie de cylindre : le calcul du point d'intersection entre le cylindre et le segment revient à remplacer l'équation de la droite dans l'équation du cylindre incliné qui est donnée par :

$$x^2 + y^2 + z^2 - \frac{(U_x * x + U_y * y + U_z * z)^2}{U_x^2 + U_y^2 + U_z^2} = R^2 \quad (13)$$

En remplaçant dans l'équation (13) les coordonnées x, y du sommet Dixel, les coordonnées des points d'intersection théoriques du segment avec le cylindre sont les solutions de l'équation suivante :

$$G * z^2 + F * z + J = 0 \quad (14)$$

Avec :

$$G = (1 - U_z^2) \quad (15)$$

$$F = -2 * E * U_z \quad (16)$$

$$J = A - E^2 - R^2 \quad (17)$$

$$A = x^2 + y^2 \quad (18)$$

$$U_x^2 + U_y^2 + U_z^2 = 1 \quad (19)$$

$$E = U_x * x + U_y * y \quad (20)$$

Les solutions de l'équation (11) passe par le calcul du déterminant donné par :

$$\Delta = F^2 - 4 * G * J \quad (21)$$

Nous avons des intersections lorsque  $\Delta \geq 0$ . Dans ce cas, les coordonnées z des points d'intersection sont données par :

$$z_0 = \frac{-F + \sqrt{\Delta}}{2 * G} \quad (22)$$

$$z_1 = \frac{-F - \sqrt{\Delta}}{2 * G} \quad (23)$$

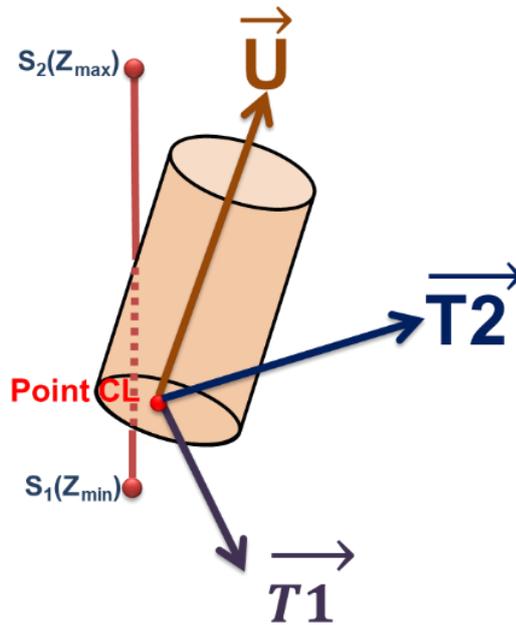


Figure 15. Intersection d'un segment avec outil cylindrique.

### I.4.3 Mise à jour des segments et des Dexels :

Avant de mettre à jour les segments, il faut toujours se souvenir que nous avons pour chaque segment et pour chaque forme d'outil deux points d'intersection et qui sont triés suivant la direction de calcul. Cette mise à jour nécessite le calcul des deux valeurs suivantes pour chaque segment appartenant à la liste des segments du sommet Dixel :

$$\alpha_0 = \frac{z_0 - z_{S1}}{z_{S2} - z_{S1}} \quad (24)$$

$$\alpha_1 = \frac{z_1 - z_{S1}}{z_{S2} - z_{S1}} \quad (25)$$

Par la suite, chaque segment est mis à jour en fonction des valeurs de  $\alpha_0$  et  $\alpha_1$ . L'analyse a montré que nous devons considérer quatre (04) cas (Figure 16) :

✓ **Cas 1** : ( $\alpha_0 \geq 0$  et  $\alpha_0 \leq 1$ ) et ( $\alpha_1 < 0$ ) : le segment devient :

- $S_1$  prend les coordonnées du point d'intersection  $I_0$ .
- $S_2$  ne change pas.

✓ **Cas 2** : ( $\alpha_0 \geq 0$  et  $\alpha_0 \leq 1$ ) et ( $\alpha_1 > 1$ ) : le segment devient :

- $S_2$  prend les coordonnées du point d'intersection  $I_0$ .
- $S_1$  ne change pas.

✓ **Cas 3** : ( $\alpha_0 < 0$  et  $\alpha_1 > 1$ ) : le segment est supprimé.

✓ **Cas 4** : ( $\alpha_0 \geq 0$  et  $\alpha_0 \leq 1$ ) et ( $\alpha_1 > 0$  et  $\alpha_1 \leq 1$ ) : la partie centrale du segment est supprimé et création de deux segments définis par :

➤ **Segment 1** :

- $S_2$  prend les coordonnées du point d'intersection  $I_0$ .
- $S_1$  ne change pas.

➤ **Segment 2** :

- $S_1$  prend les coordonnées du point d'intersection  $I_1$ .
- $S_2$  ne change pas.

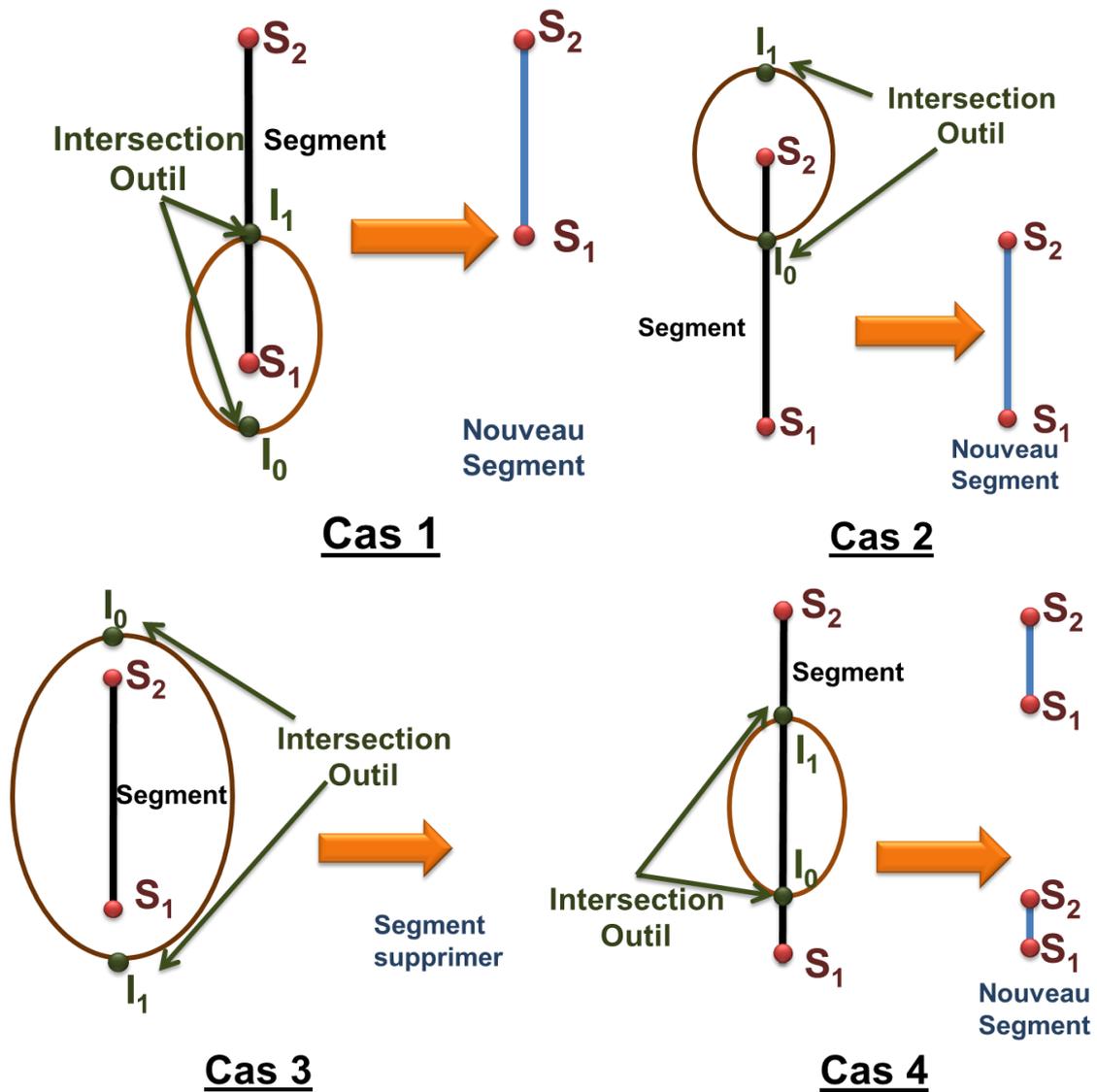


Figure 16. Mise à jour des segments.

## I.5 Vérification de la conformité des surfaces obtenues :

Après avoir simulé l'enlèvement de matière, il faut calculer les écarts entre les Dexels résultants et les points théoriques dans le but de vérifier la conformité des formes obtenues avec les formes théoriques. Pour cela, nous avons proposé deux méthodes. La première permet de calculer les écarts et la deuxième pour avoir une idée détaillée sur la répartition réelle des écarts.

### I.5.1 Calcul de l'écart :

Dans cette méthode, l'écart est calculé entre un sommet des segments associé au sommet Dexel avec le point théorique le plus proche. Pour cela, les tolérances supérieures et inférieures doivent être spécifiées (Figure 17). Il est possible de distinguer trois zones :

- **Zone 1 :** les points au-dessous du point acceptable minimum présentent des problèmes d'inférences. Donc, ces erreurs ne peuvent être réglées.
- **Zone 2 :** les points compris entre le point acceptable minimum et le point acceptable maximum vérifient la contraintes imposées.
- **Zone 3 :** les points au-dessus du point acceptable maximum présentent un excès de matière. Donc, ces erreurs peuvent être réglées en procédant à un usinage supplémentaire.

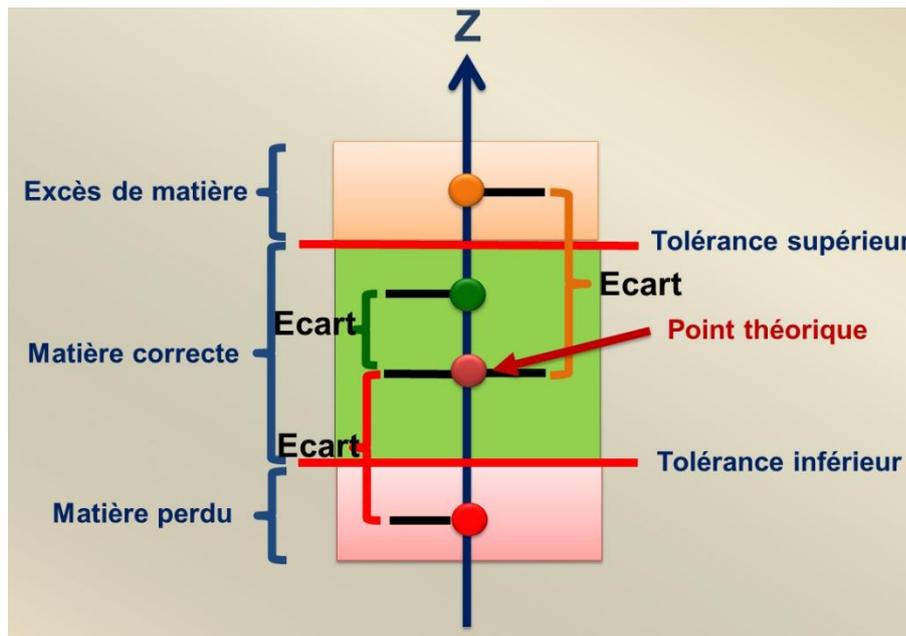


Figure 17. Calcul des écarts.

### I.5.2 Spectre des écarts :

Afin de donner une visualisation tridimensionnelle de la répartition des écarts, nous avons pensé au spectre pour voir en dégradation des couleurs les écarts. Ce spectre permet de localiser rapidement les différentes zones.

## II. Modélisation en UML :

Le langage UML (Unified Modeling Language) est un langage de modélisation graphique et textuel conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en objet. Pour présenter les différentes fonctionnalités de notre application, nous avons utilisé quelques diagrammes du langage UML. Dans notre projet, nous avons utilisé les diagrammes suivants :

- Diagramme de cas d'utilisation.
- Diagramme d'activité.
- Diagramme de classes.

## II.1 Diagramme de cas d'utilisation :

Bien souvent, les utilisateurs ne sont pas des informaticiens. Il leur faut donc un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation qui permettent de recueillir, d'analyser et d'organiser les besoins et de recenser les grandes fonctionnalités d'un système. Nous commençons par le diagramme de cas d'utilisation général qui donne une vue globale du fonctionnement du système.

### II.1.1 Diagramme de cas d'utilisation général :

Le système est divisé en cinq (05) grandes phases comme illustré par la Figure 18.

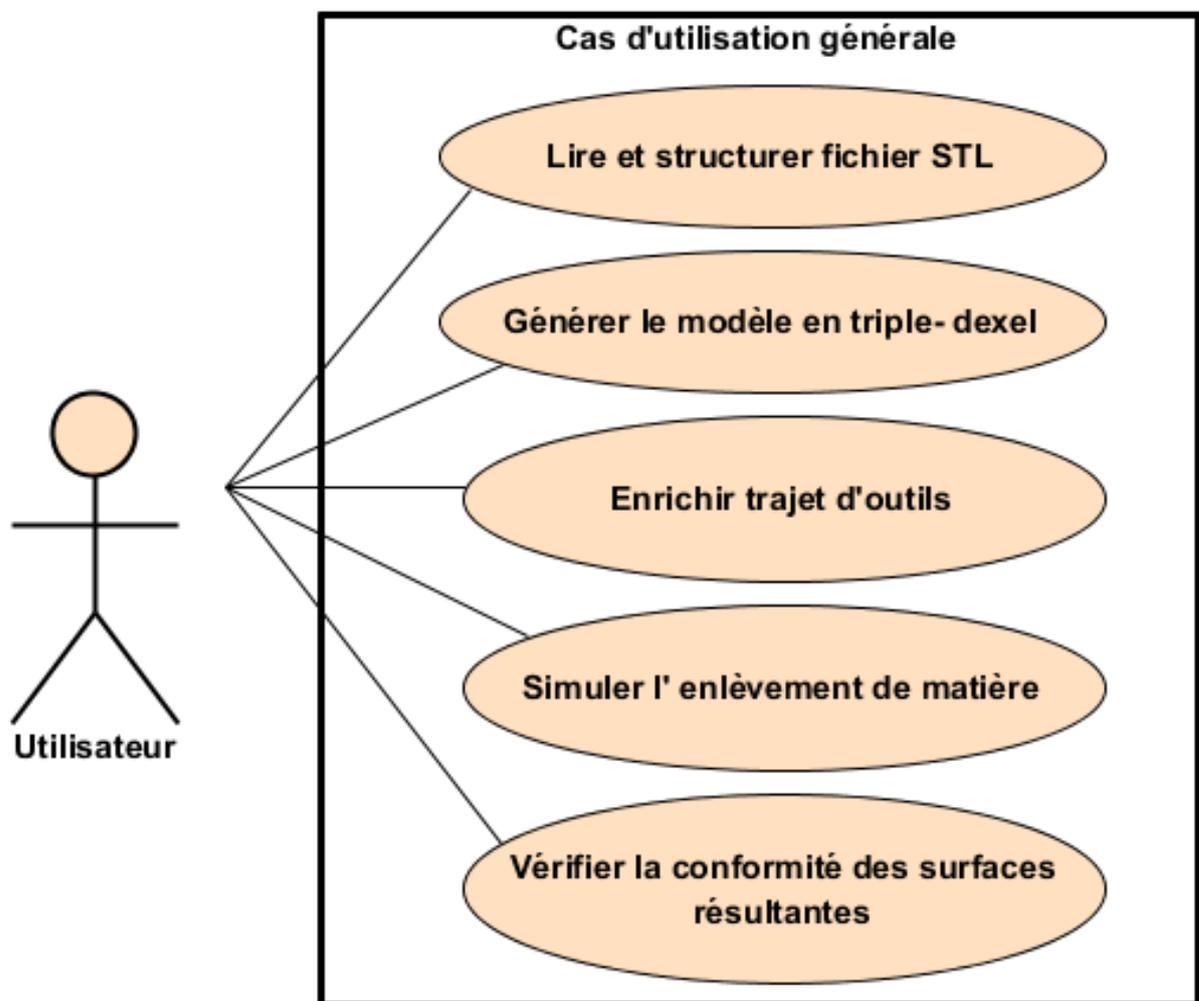


Figure 18. Diagramme de cas d'utilisation général.

II.1.2 Diagramme de cas d'utilisation « Lire fichier STL » :

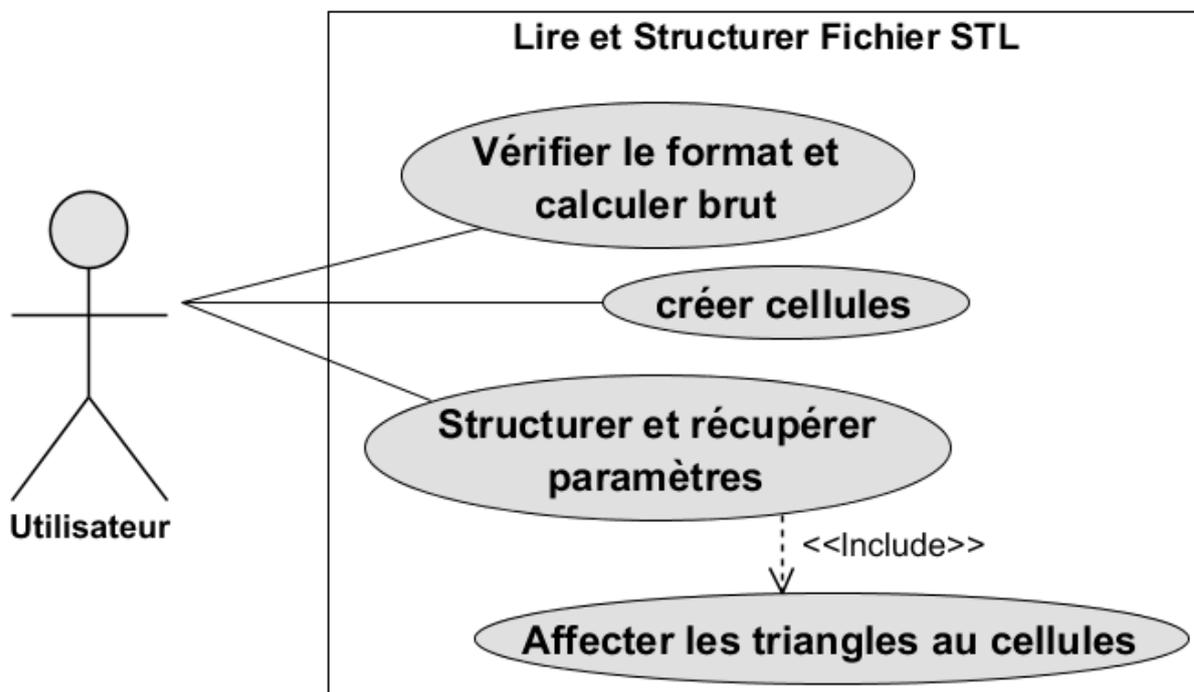


Figure 19. Diagramme de cas d'utilisation « Lire fichier STL ».

II.1.3 Diagramme de cas d'utilisation « Générer le modèle en Triple-Dexel » :

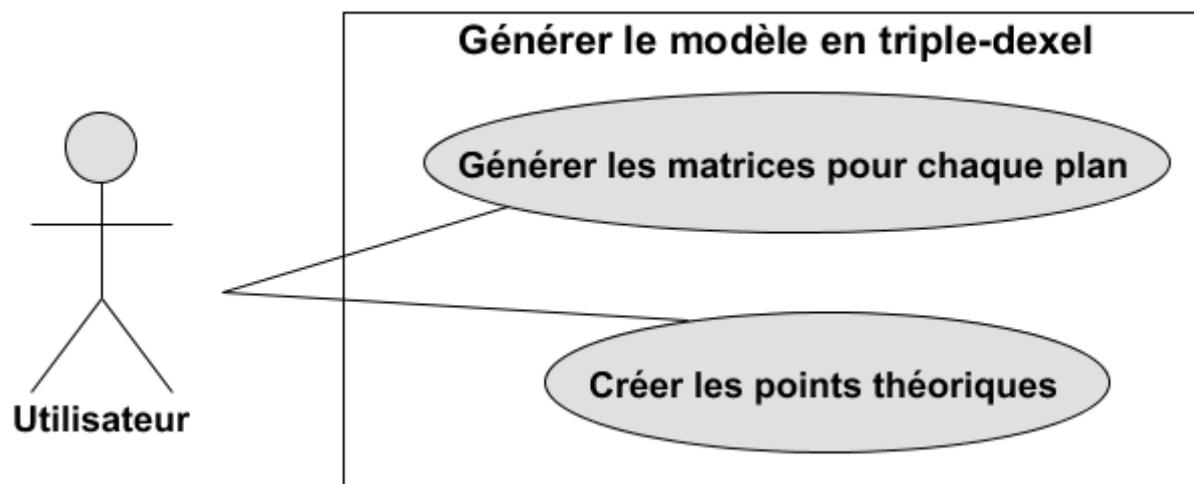


Figure 20. Diagramme de cas d'utilisation « Génération du modèle en Triple-Dexel ».

II.1.4 Diagramme de cas d'utilisation « Générer les matrices pour chaque plan » :

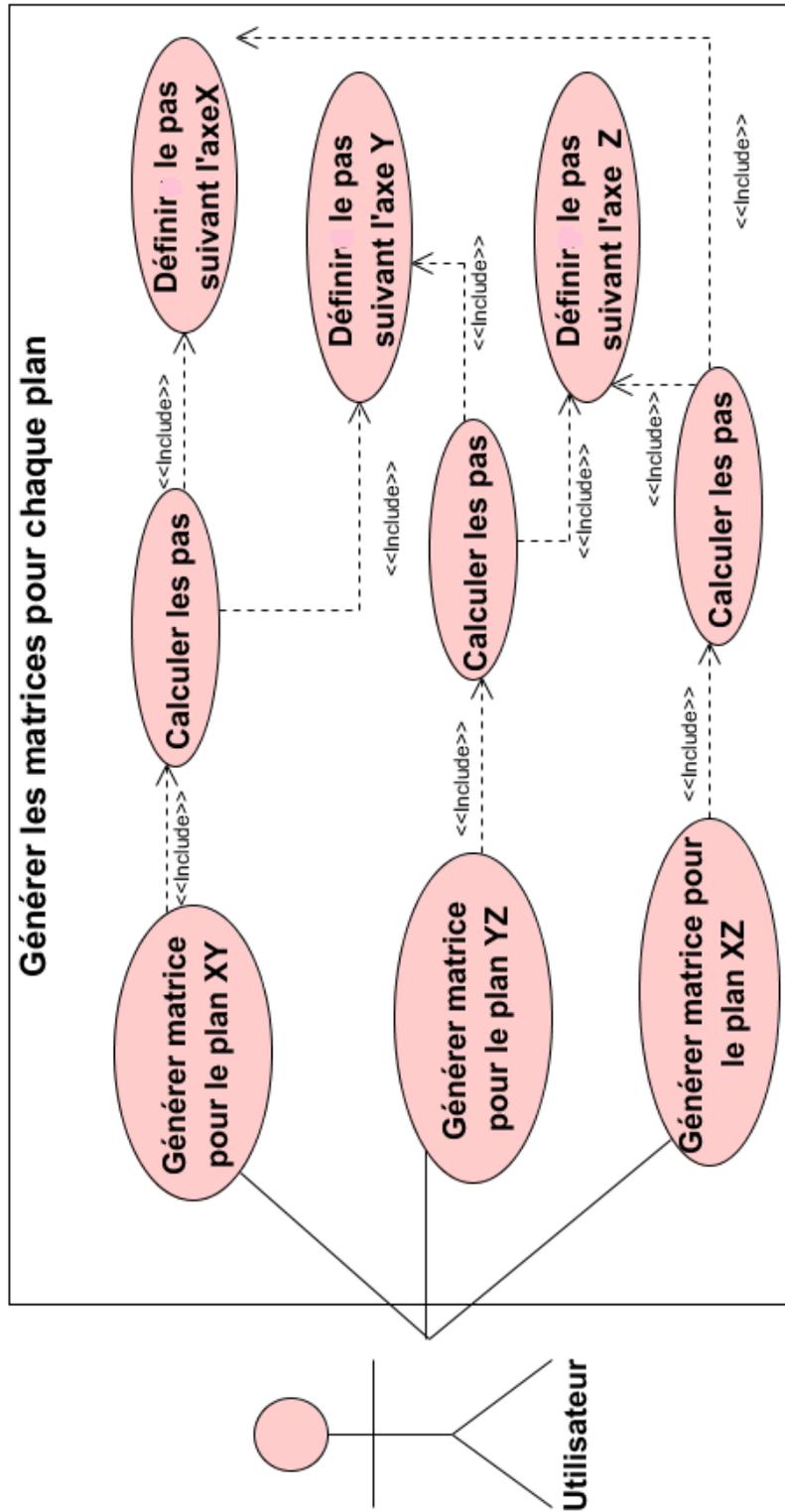


Figure21. Diagramme de cas d'utilisation « Générer les matrices pour chaque plan ».

II.1.5 Diagramme de cas d'utilisation « Calculer les points théoriques » :

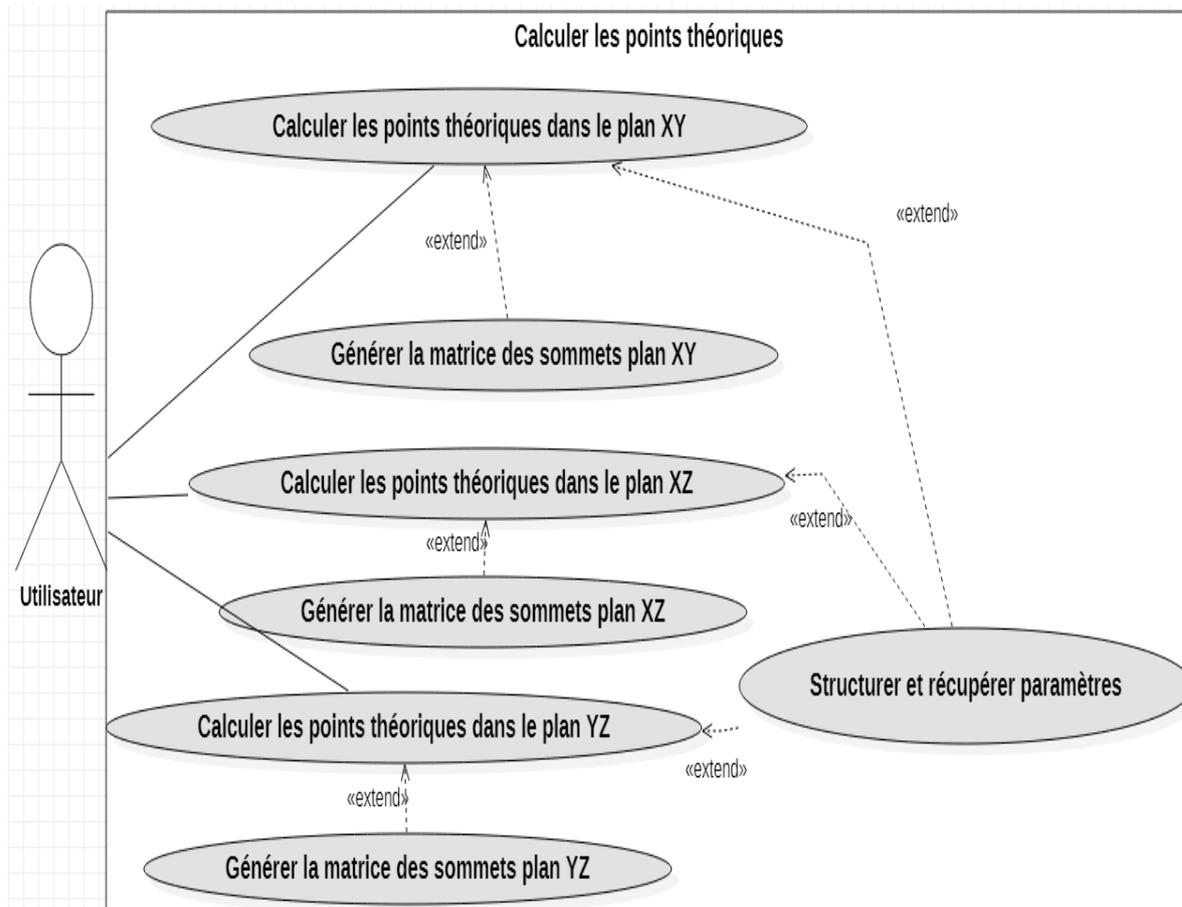


Figure 22. Diagramme de cas d'utilisation « Calculer les points théoriques ».

II.1.6 Diagramme de cas d'utilisation « Enrichir le trajet d'outil » :

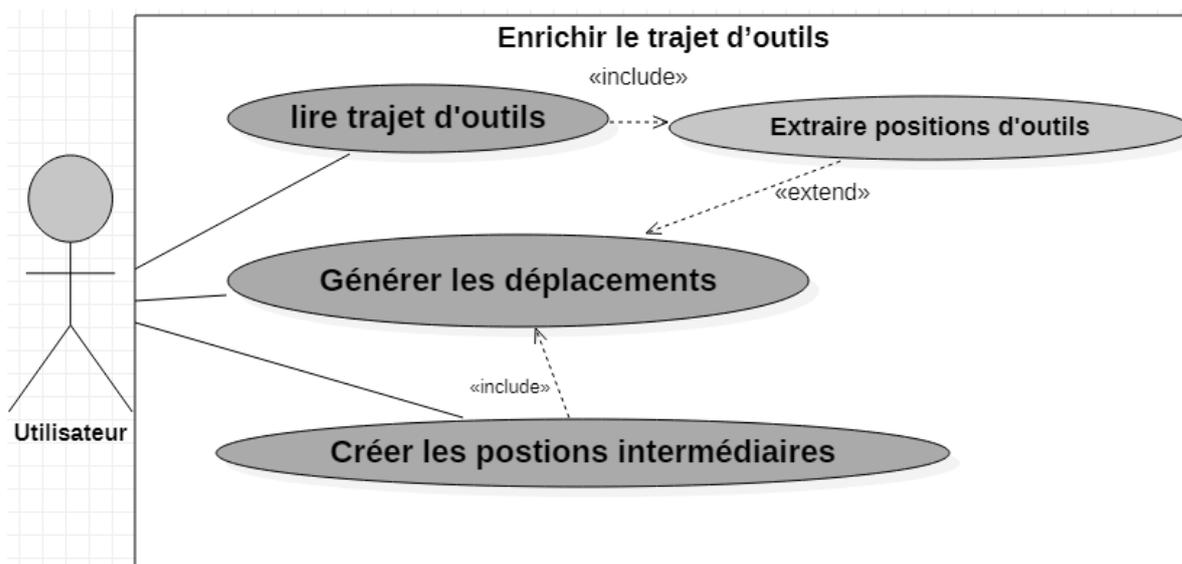


Figure 23. Diagramme de cas d'utilisation « Enrichir le trajet d'outil ».

## II.2 Diagramme d'activité :

Le diagramme d'activité permet de mettre l'accent sur les traitements. Il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

### II.2.1 Diagramme d'activité général :

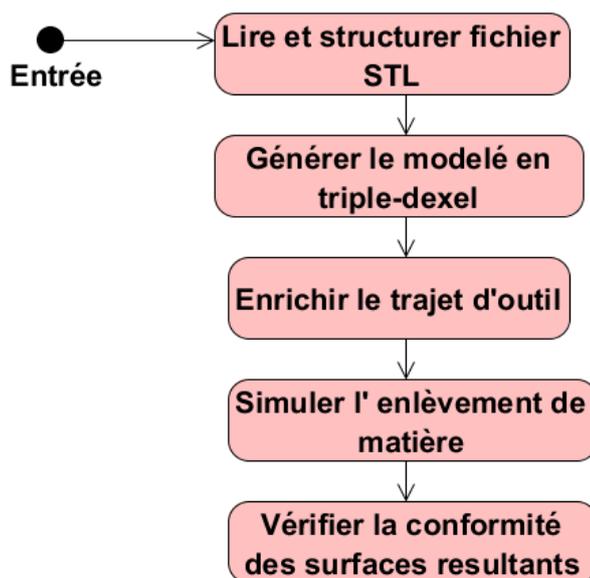


Figure 24. Diagramme d'activité général.

### II.2.2 Diagramme d'activité « Lire fichier STL » :

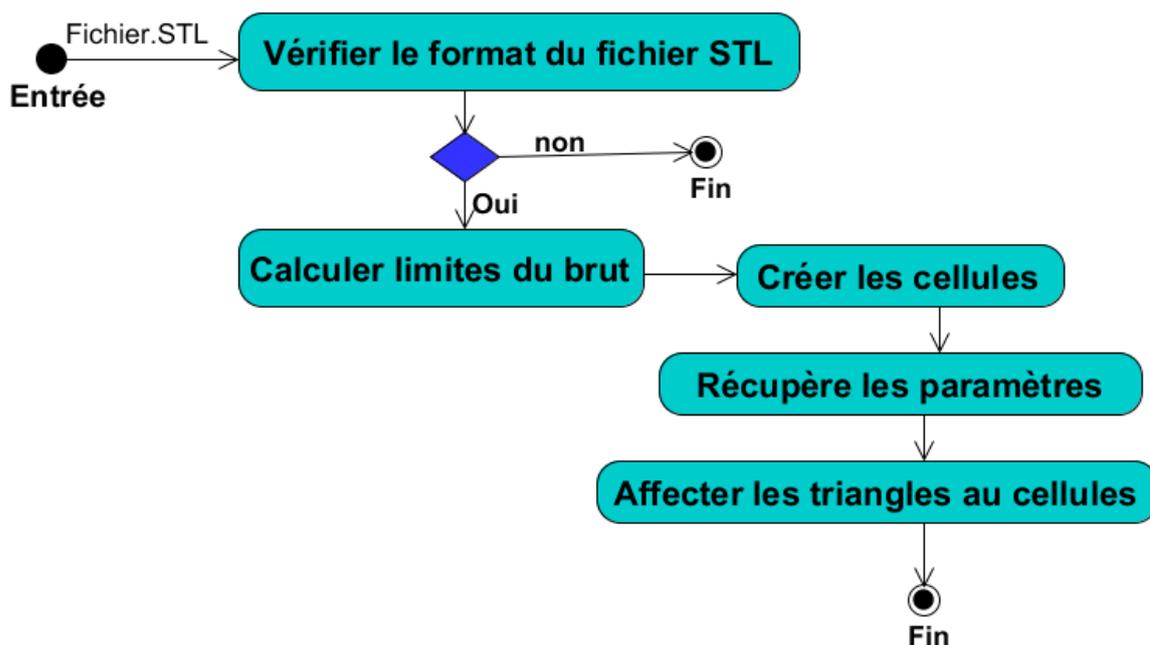


Figure 25. Diagramme d'activité « Lire fichier STL ».

II.2.3 Diagrammes d'activité « Calculer les points théoriques » :

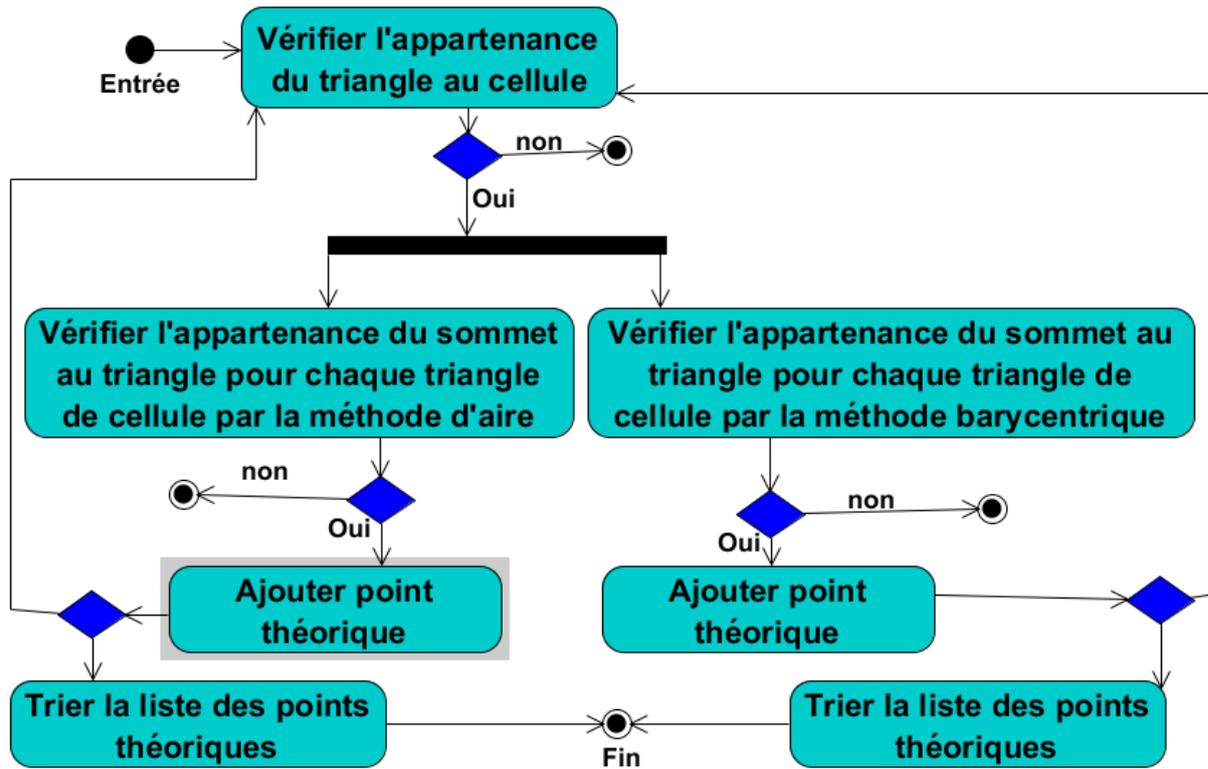


Figure 26. Diagrammes d'activité « Calculer les points théoriques ».

II.2.4 Diagrammes d'activité « Lire trajet d'outil » :

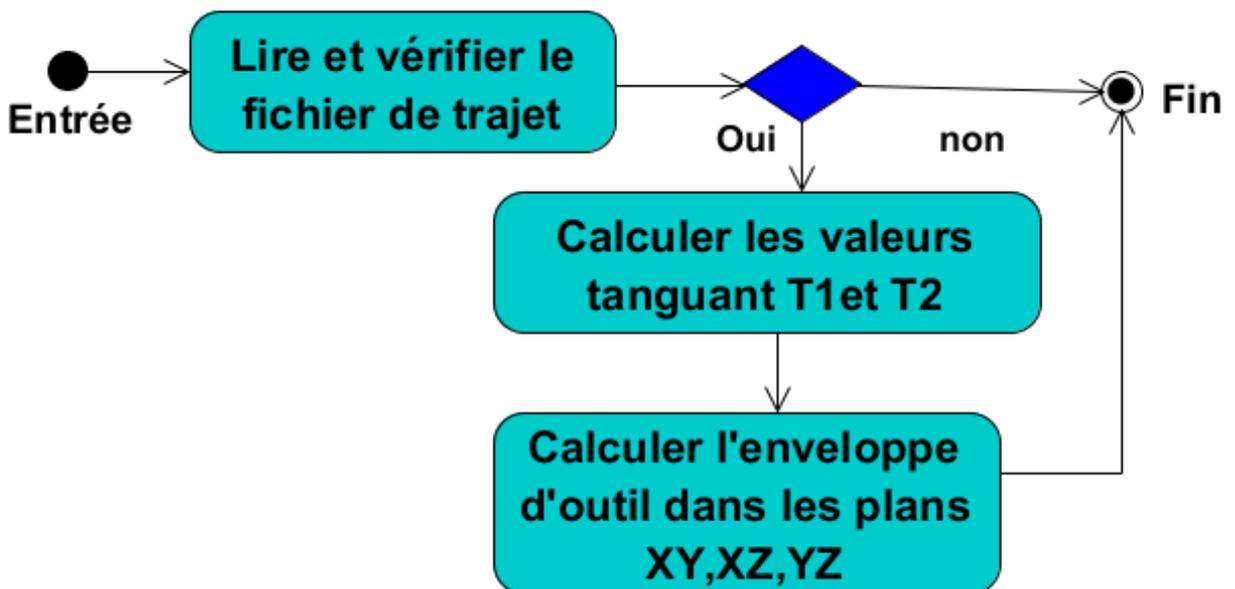


Figure 27. Diagrammes d'activité « Lire trajet d'outils ».

II.2.5 Diagrammes d'activité « Générer les positions intermédiaires » :

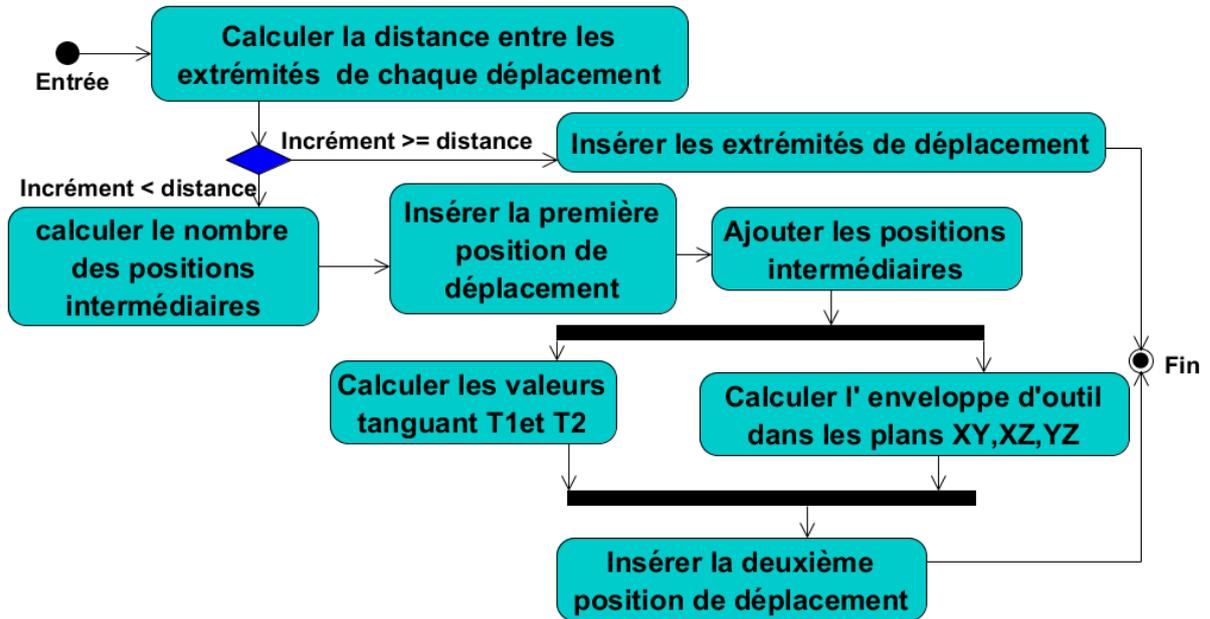


Figure 28. Diagrammes d'activité « Générer les positions intermédiaires ».

II.2.6 Diagramme d'activité « Simuler l'enlèvement de matière » :

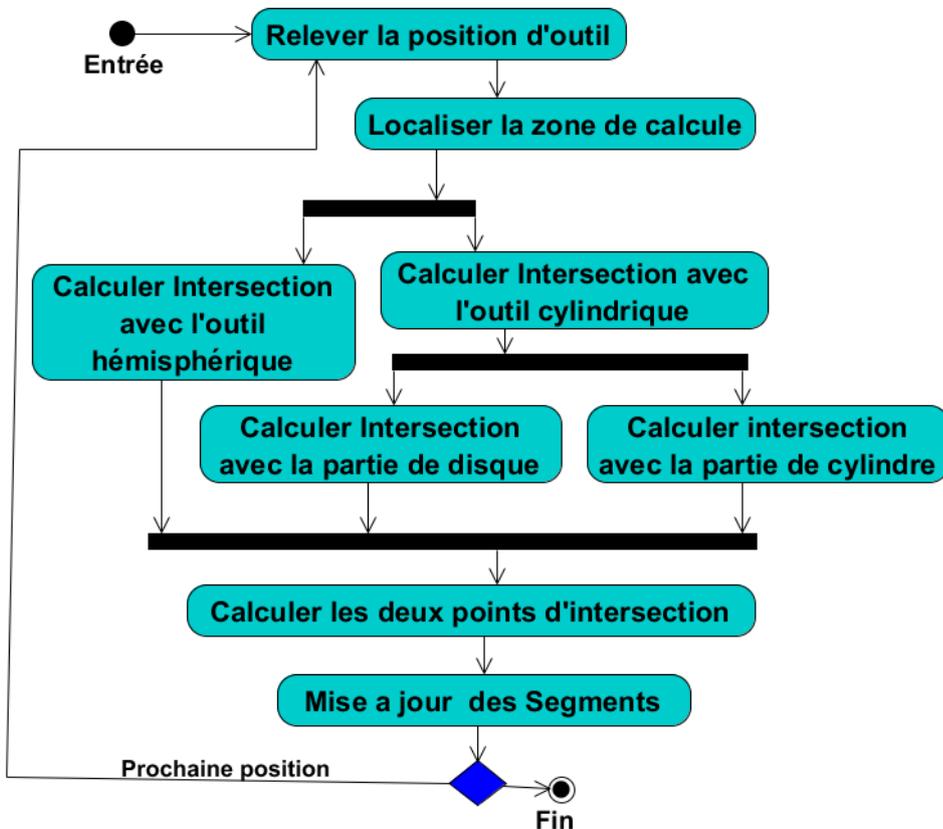


Figure 29. Diagramme d'activité « Simuler l'enlèvement de matière ».

### II.3 Diagramme de classes :

Le diagramme de classe montre la structure interne du logiciel. Il fournit une représentation abstraite des objets du système ainsi que les différentes relations entre eux. Il s'agit d'une vue statique car il fait abstraction des aspects temporels et dynamiques. La Figure 30 montre notre diagramme de classe.

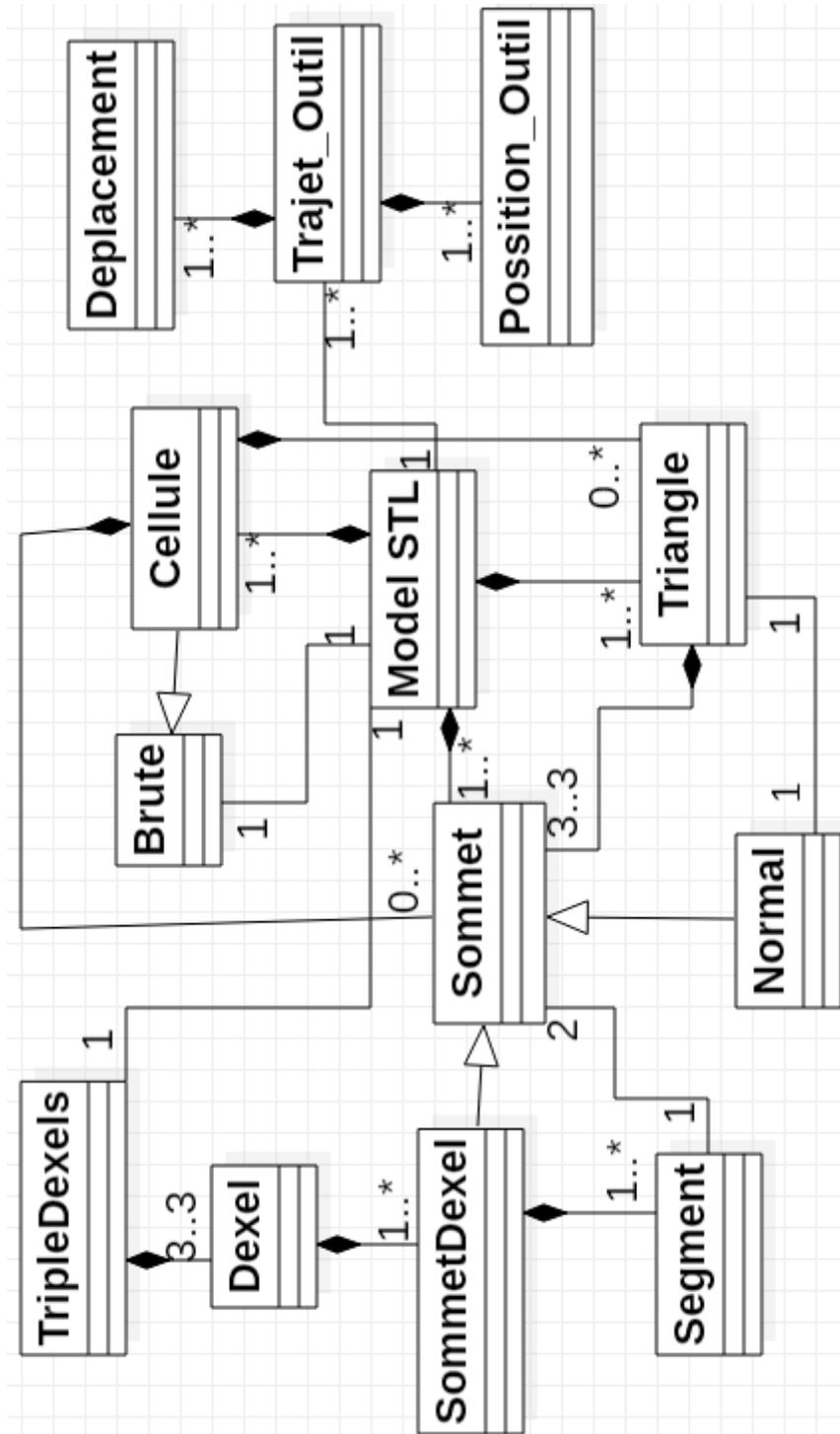


Figure 30. Diagramme de classes général.

- **Classe « Sommet » :**

Regroupe les coordonnées  $(x, y, z)$  d'un point.



Figure 31. Classe « Sommet ».

- **Classe « Triangle » :**

Regroupe les paramètres d'un triangle.

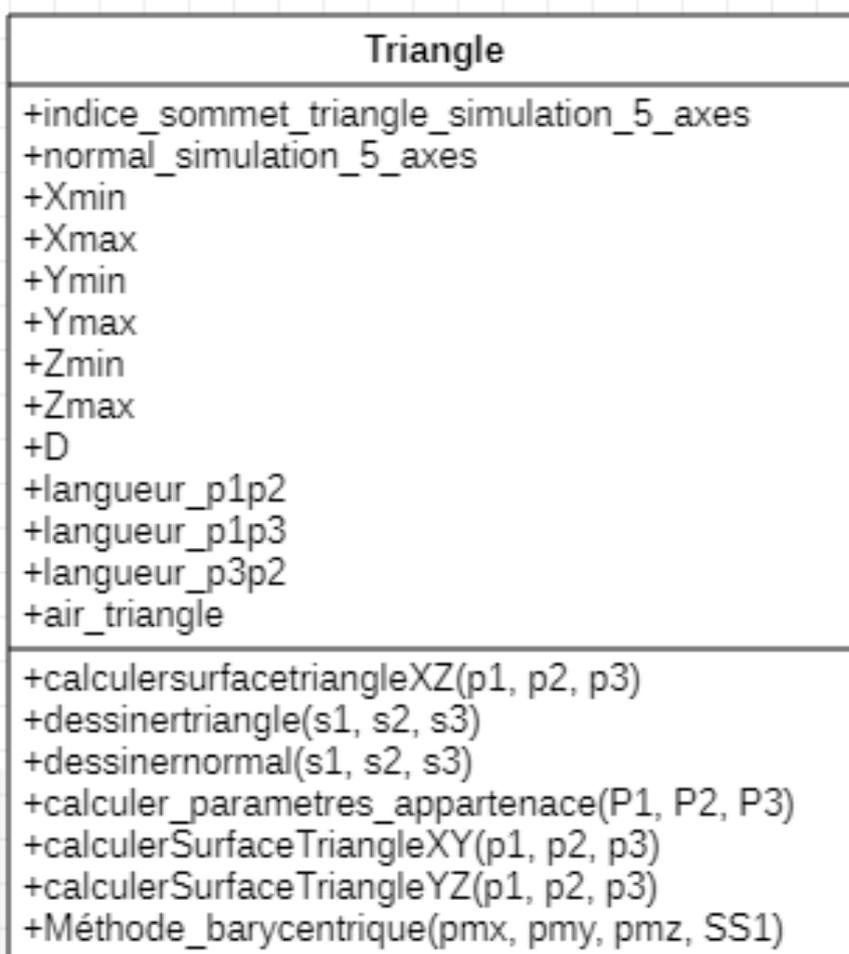


Figure 32. Classe « Triangle ».

- **Classe «Cellule» :**

Regroupe les paramètres d'une cellule.

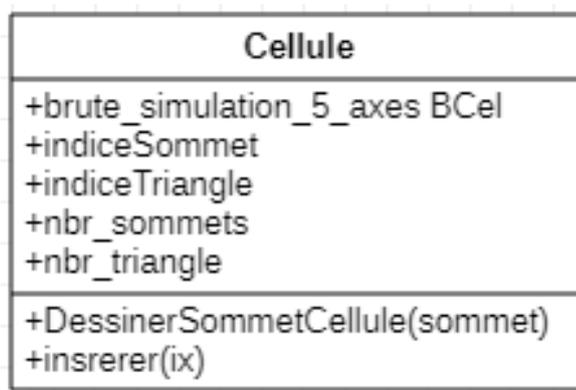


Figure 33. Classe « Cellule».

- **Classe « model STL» :**

C'est la classe principale qui contient la liste des sommets et la liste des triangles du fichier STL ainsi que les cellules créées. Les méthodes de cette classe nous permettent de récupérer les principales informations du fichier STL.

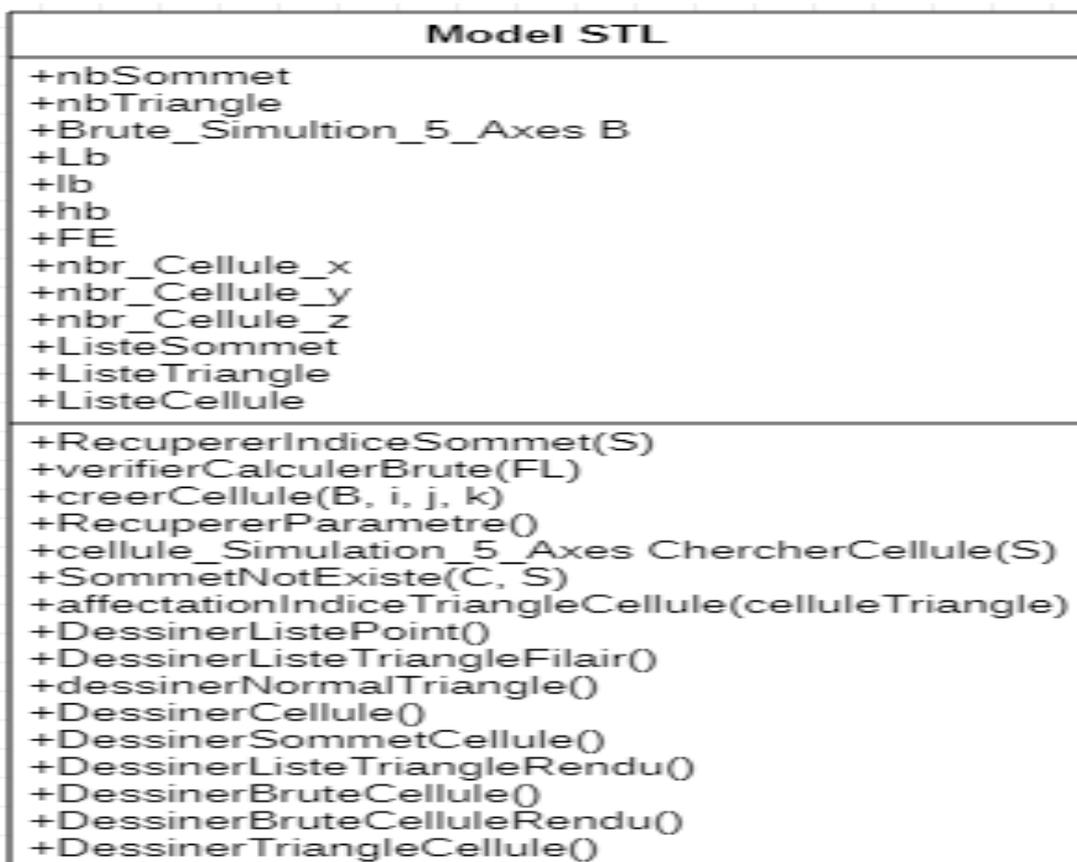


Figure 34. Classe « model STL».

- **Classe « Segment » :**

Regroupe les paramètres d'un segment.

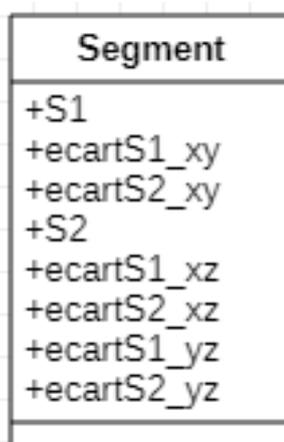


Figure 35. Classe « Segment ».

- **Classe « SommetDexel » :**

C'est la classe qui contient pour chaque sommet Dexel la liste des segments et la liste des points théoriques.

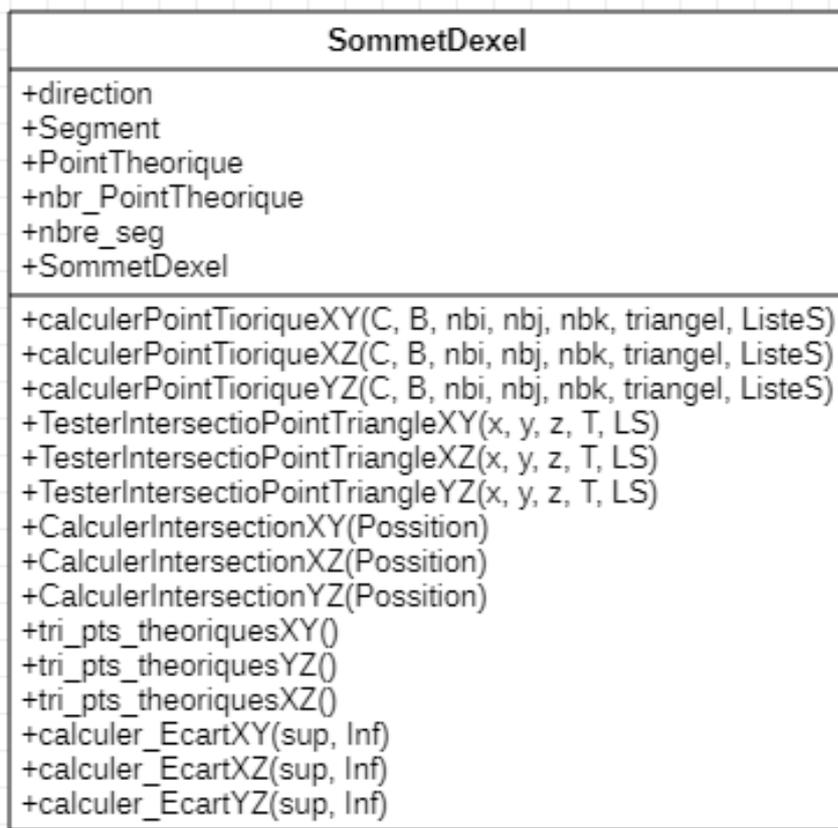


Figure 36. Classe « SommetDexel ».

- **Classe « Dixel » :**

C'est la classe qui permet de créer les matrices des sommets Dixel.

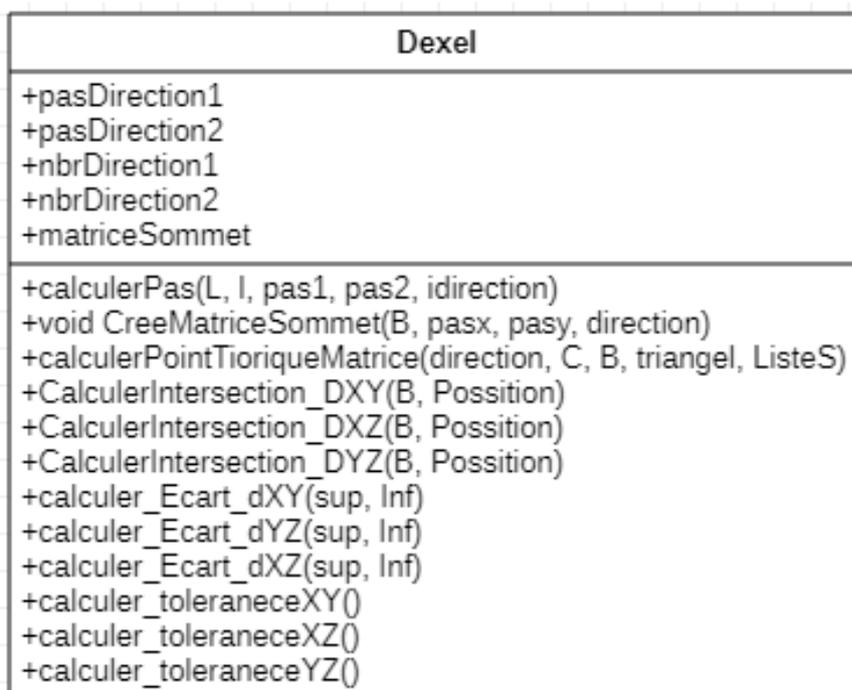


Figure 37. Classe « Dixel».

- **Classe « TripleDexels » :**

C'est la classe qui contient les paramètres des trois plans.

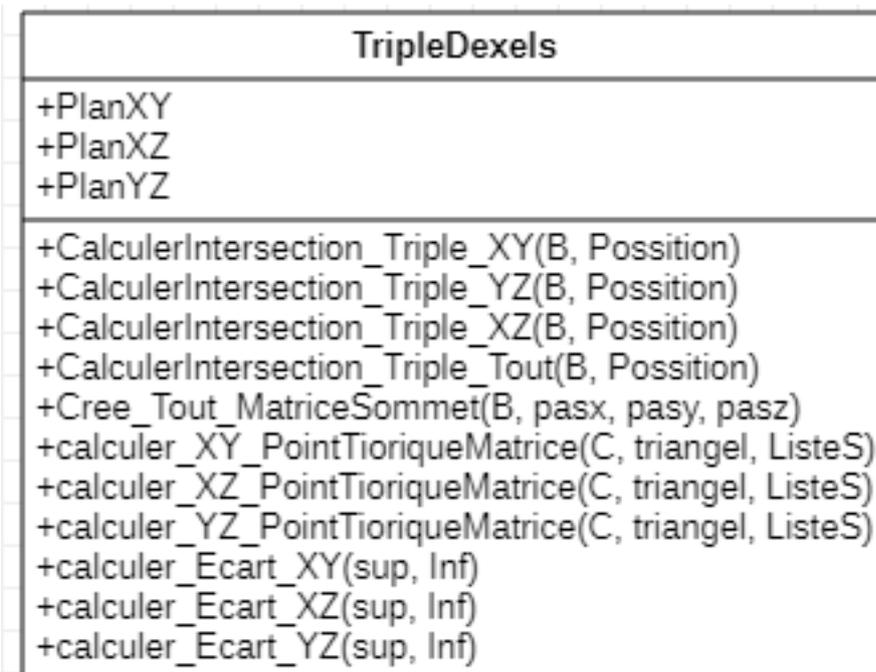


Figure 38. Classe « TripleDexels».

- **Classe « Possition\_Outil » :**

C'est la classe qui contient les paramètres de l'outil pour chaque position.

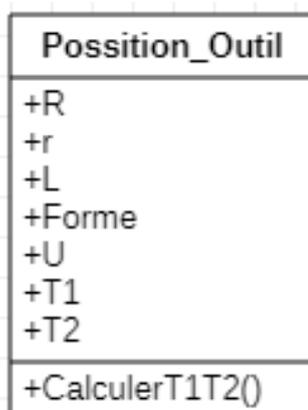


Figure 39. Classe « Possition\_Outil ».

- **Classe « Déplacement » :**

C'est la classe qui contient la liste des positions intermédiaires.

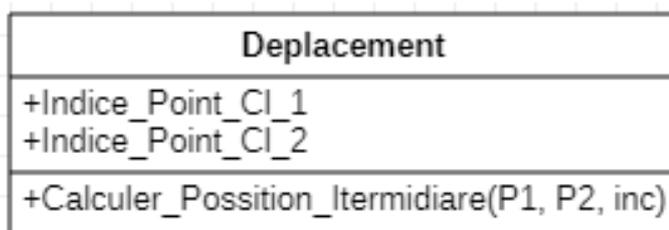


Figure 40. Classe « Déplacement ».

- **Classe « Trajet\_Outil » :**

C'est la classe principale qui permet de lire le trajet d'outils et qui contient la liste des positions d'outils ainsi que la liste des déplacements.

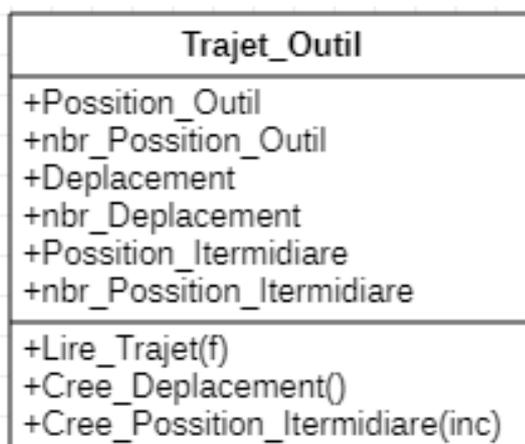


Figure 41. Classe « Trajet\_Outil ».

### **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 traitées à travers des exemples de validation.

# CHAPITRE 3

---

## IMPLEMENTATION ET VALIDATION

Introduction.

1. Présentation des langages utilisés.
2. Présentation de l'environnement de CFAO.
3. Présentation de l'application.
4. Test et validation.

Conclusion.

### Introduction :

Après avoir présenté la conception de notre système dans le deuxième chapitre, l'objectif de ce chapitre est la présentation des différentes fonctionnalités de notre système. Nous commençons par définir les outils de développement de notre application ainsi qu'une vue générale en utilisant des captures d'écrans. Par la suite, une validation est réalisée par un ensemble de tests qui ont été préparés pour chaque fonctionnalité de l'application.

### I. Présentation des langages utilisés :

Comme notre système doit être intégré à la plateforme logicielle de production des pièces de formes complexes développée par l'équipe « CFAO » du « CDTA », les outils de développement utilisés lors de la mise en œuvre de notre application (C++, OpenGL) sont les mêmes utilisés par l'équipe « CFAO » pour se conformer à la tendance qui veut que la majorité des systèmes de « CFAO » sont développés en C++ et en OpenGL.

#### I.1 Présentation du langage C++ :

C++ est un langage de programmation compilé créé initialement par Bjarne Stroustrup dans les années 1980. Aujourd'hui normalisé par l'ISO. Le langage C++ c'est l'amélioration du langage C et il est un des langages de programmation les plus utilisés par les développeurs et dans les applications graphiques (création des programmes en 3D, les jeux vidéo). Il permet la programmation sous de multiples paradigmes (programmation procédurale, orientée objet ou générique).



#### I.2 Présentation d'OpenGL :

OpenGL (Open Graphics Library) est un ensemble normalisé de fonctions de calcul d'images 2D ou 3D lancé par Silicon Graphics en 1992. Cette interface de programmation est disponible sur de nombreuses plateformes où elle est utilisée pour des applications qui vont du jeu vidéo jusqu'à la CAO en passant par la modélisation. OpenGL permet à un programme de déclarer la géométrie d'objets sous forme de points, de vecteurs, de polygones, de bitmaps et de textures. OpenGL effectue ensuite des calculs de projection pour déterminer l'image à l'écran, en tenant compte de la distance, de l'orientation, des ombres, de la transparence et du cadrage. L'interface regroupe environ 250 fonctions différentes qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes à partir de simples primitives géométriques. Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes les plateformes, elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D. Cette bibliothèque est également utilisée dans l'industrie du jeu vidéo où elle est souvent en rivalité avec la bibliothèque de Microsoft « Direct3D » [13].



### I.3 Présentation d'Embarcadero C++ Builder 10 Seattle :

#### I.3.1 C++ Builder :

C++ Builder est un logiciel de développement rapide d'applications « RAD » conçu par Borland qui reprend les mêmes concepts, la même interface et la même bibliothèque que Delphi en utilisant le langage C++. Il permet de créer rapidement des applications Win32, Win64, MacOS, iOS et Android, ainsi qu'une interface graphique avec son éditeur de ressources. Il est compatible avec la version de norme ISO C++17.

#### I.3.2 Embarcadero C++ Builder 10 Seattle :

Embarcadero C++ Builder 10 Seattle est le moyen le plus efficace et rapide pour les développeurs afin de construire et de mettre à jour des applications hyper-connectées, intégrant des jeux étendus de données et graphiquement pour Windows 10, OS X, iOS, Android et IoT [14].



### II. Matériel utilisé :

Le matériel utilisé dans la phase de développement et de validation de l'application se résume en un PC portable sous Windows 10, Intel(R) Core(TM) i3-5005U CPU @2.00 GHz 2.00 GHz et 4.00Go de RAM.

### III. Présentation de l'environnement de CFAO :

C'est une plateforme logicielle dédiée à la production numérique des pièces de formes complexes sur fraiseuses numérique 03-axes et 05-axes. Cette plateforme est composée de deux parties (Figure 1) :

- Lancement des fonctionnalités.
- Visualisation.

Le module logiciel que nous avons développé est intégré à cette plateforme qui s'appelle « **Simulation usinage 5-axe** » (Figure 2).

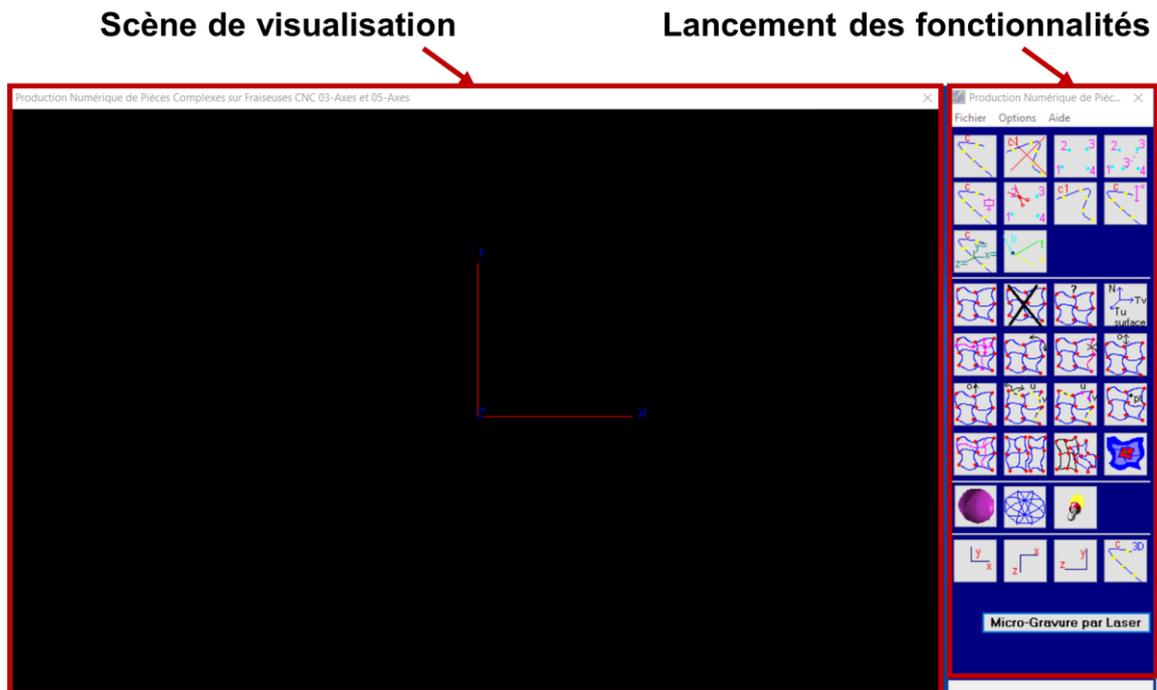


Figure 1. Environnement de « CFAO ».

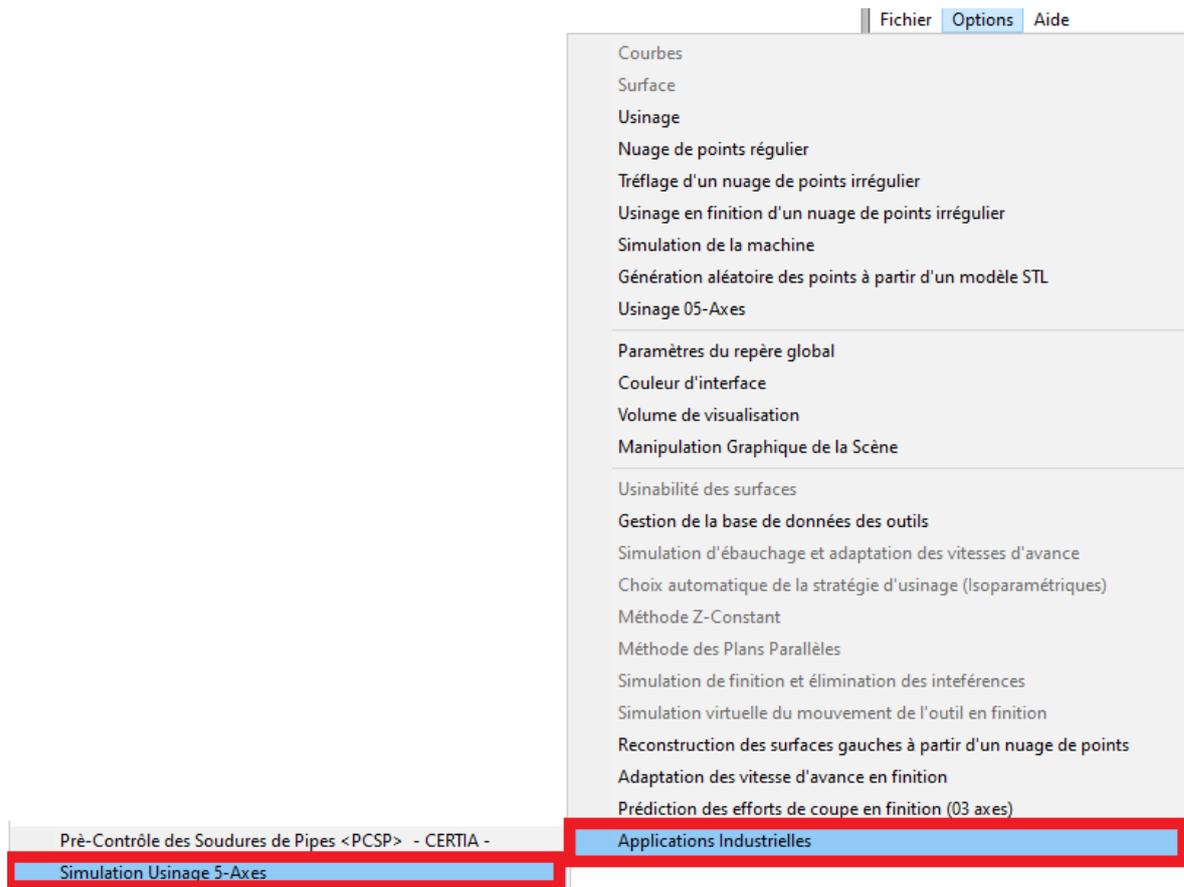


Figure 2. Lancement du module logiciel développé « Simulation usinage 5-axe ».

## IV. Présentation de l'application :

L'application développée est composée de plusieurs fonctionnalités complémentaires. Elle est composée de cinq (05) onglets (Figure 3).



Figure 3. Onglets de l'application développée.

La description détaillée de chaque onglet est réalisée dans les paragraphes suivants.

### IV.1 Lecture du fichier STL :

Le premier onglet « **Lecture du fichier STL** » permet la lecture du fichier texte contenant l'ensemble des paramètres des triangles (coordonnées des sommets et normales des triangles) (Figure 4).

Par un simple clic sur le bouton « **Ouvrir fichier STL** » et sélection du fichier à ouvrir, les limites du brut sont calculées.

Une fois le fichier est lu, les dimensions du brut (Longueur, Largeur et Hauteur) peuvent être récupérées.

Par la suite, en spécifiant le nombre de cellules suivant les trois axes (X, Y et Z) et Par un simple clic sur le bouton « **Structurer paramètre** », le nombre total des sommets et le nombre total des triangles sont calculées.

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

- Brut du modèle en deux modes filaire et rendu.
- Triangles du modèle en deux modes filaire et rendu.
- Sommets du modèle.
- Normales des sommets du modèle.
- Normales des triangles du modèle.
- Brut des cellules en deux modes filaire et rendu.
- Sommets des cellules.
- Triangles des cellules.

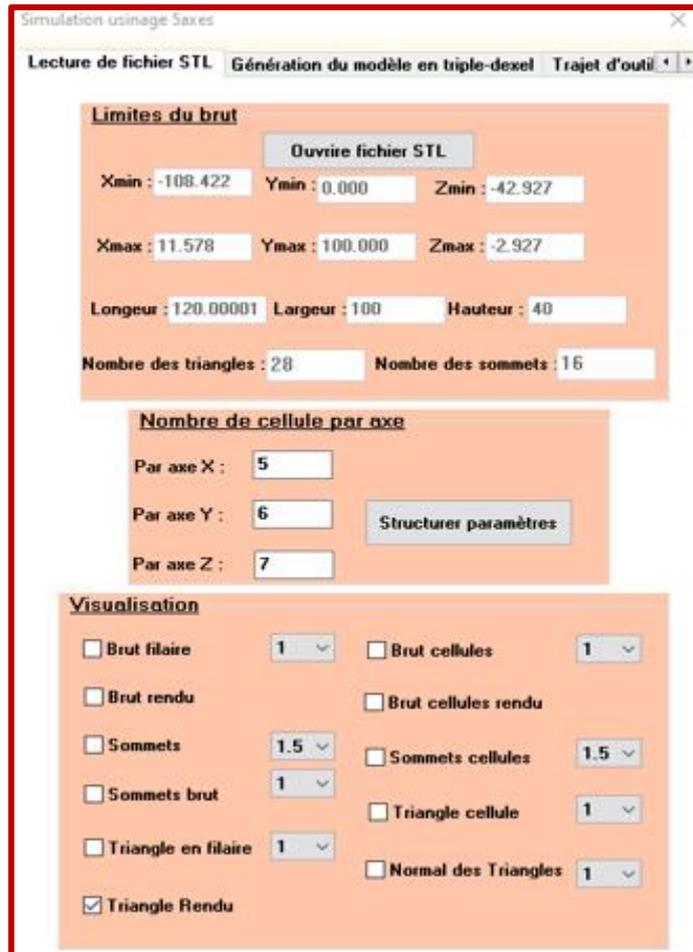


Figure 4. Onglet « Lecture du fichier STL ».

## IV.2 Génération du modèle en Triple-Dexels :

Le deuxième onglet « **Génération du modèle en Triple-Dexels** » permet la génération du modèle en Triple-Dexels (Figure 5). En spécifiant les pas suivant les trois axes (X, Y et Z) et suite à un simple clic sur le bouton « **Créer les matrices** », les matrices des droites sont générées pour les trois plans XY, XZ et YZ.

Ensuite, l'utilisateur peut visualiser pour chaque plan :

- Les sommets.
- Les segments.

Après avoir créé les matrices, nous devons choisir une des deux méthodes « Méthode barycentrique » ou « Méthode d'aire ». Ensuite, en cliquant sur le bouton « **Calculer les points théoriques** », les points théoriques pour les trois plans XY, XZ et YZ sont calculées. Par la suite, il est possible de visualiser les points théoriques de chaque plan.

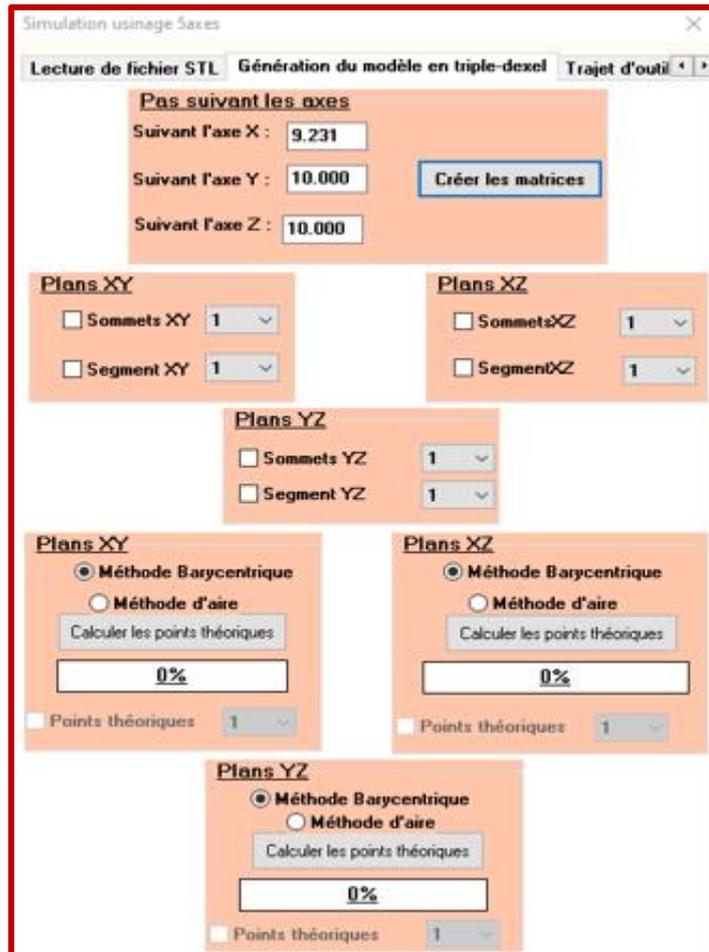


Figure 5. Onglet « Générer le modèle en Triple-Dexels ».

Une fois la lecture de fichier STL est terminée et le modèle en Triple-Dexels est généré, l'utilisateur passe au troisième onglet pour l'enrichissement du trajet d'outils.

### IV.3 Enrichissement du trajet d'outils :

Le troisième onglet « **Trajet d'outil** » permet la lecture du trajet d'outils (Figure 6). Il contient les paramètres d'outils (Forme d'outil, R « Rayon d'outil cylindrique », r « Rayon d'outil hémisphérique », longueur « L », coordonnées du point extrémité d'outil « CL », composantes du vecteur axe d'outil).

Par un simple clic sur le bouton « **Lire le trajet** » et sélection du fichier à ouvrir, l'utilisateur peut afficher les informations suivantes :

- Les points CL.
- Le trajet d'outils.

Après avoir lu le trajet, l'utilisateur doit spécifier la valeur de l'incrément et il clique sur le bouton « **Crée déplacement** » pour créer les déplacements pour les trois plans XY, XZ et YZ ainsi que la génération des positions intermédiaires. Il est possible d'afficher :

- Les déplacements.
- Les positions intermédiaires.

Ensuite, l'utilisateur clique sur le bouton « **Calculer intersection** » afin de calculer les intersections pour les trois plans XY, XZ et YZ. Il peut visualiser selon les indices des positions, les intersections entre l'outil et les Dexels.

De même, l'utilisateur peut visualiser les informations suivantes :

- Le point CL.
- Les axes  $T_1$  et  $T_2$ .
- Axe d'outil.
- L'outil en deux modes filaire et rendu.
- L'intervalle des points d'intersection.
- L'enveloppe d'outil pour chaque plan.

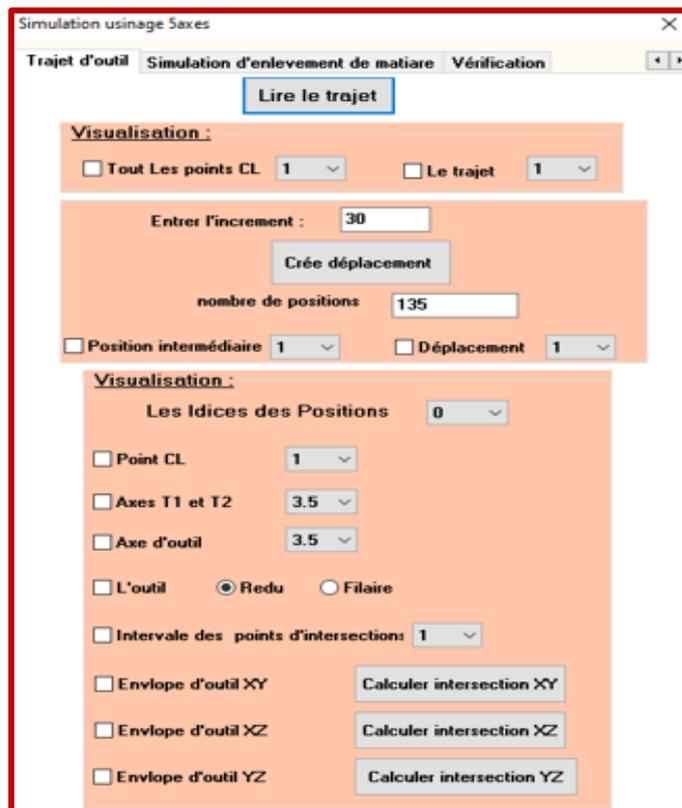


Figure 6. Onglet « Trajet d'outil ».

Une fois le trajet d'outil est généré, l'utilisateur passe à l'onglet suivant pour la simulation d'enlèvement de matière.

### IV.4 Simulation d'enlèvement de matière :

Il permet de simuler virtuellement d'enlèvement de matière (Figure 7). Pour cela, l'utilisateur a la possibilité de choisir entre deux modes de simulation :

- Simulation pour chaque plan XY, XZ et YZ.
- Simulation globale.

Avant de lancer la simulation, nous devons spécifier l'incrément d'affichage. Ensuite, un simple clic sur le bouton « **Calculer intersection** », les intersections sont calculées pour chaque plan XY, XZ et YZ ou globale.

L'utilisateur peut afficher :

- Les intersections pendant les calculs.
- Les points d'intersection.
- Les extrémités de segments.

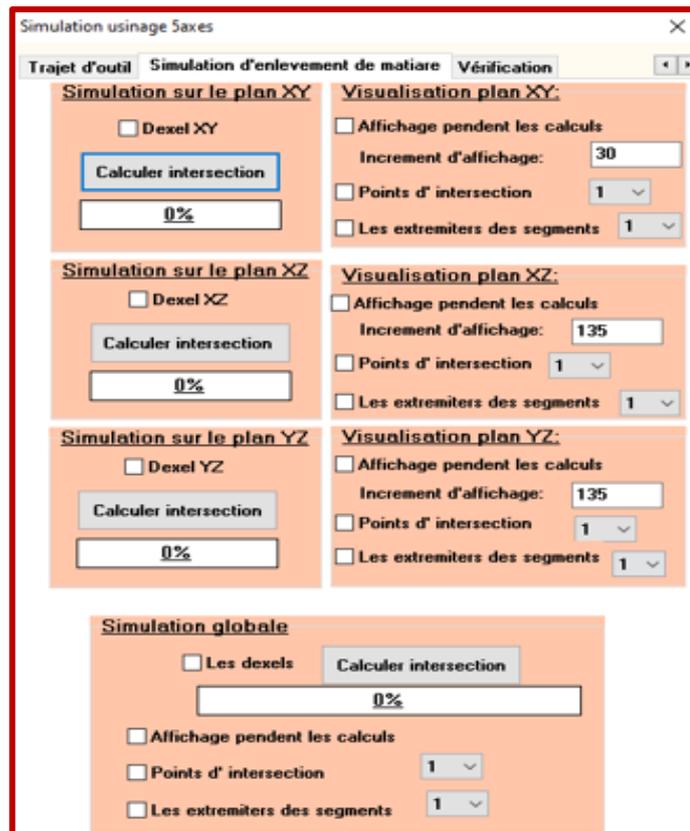


Figure 7. Onglet « Simulation d'enlèvement de matière ».

#### IV.5 Vérification de la conformité des surfaces résultantes :

Le dernier onglet « **Vérification** » permet la vérification de la conformité des surfaces résultantes avec le modèle « STL » (Figure 8). Pour cela, l'utilisateur doit spécifier :

- La valeur inférieure de l'intervalle de tolérance.
- La valeur supérieure de l'intervalle de tolérance.

Ensuite, par un simple clic sur le bouton « **Calculer l'écart** », les écarts sont calculés pour chaque plan XY, XZ et YZ. Par la suite, les informations suivantes sont affichées :

- L'écart minimum global.
- L'écart maximum global.

Pour avoir une idée très claire sur la répartition des écarts, l'utilisateur peut afficher le spectre correspondant.

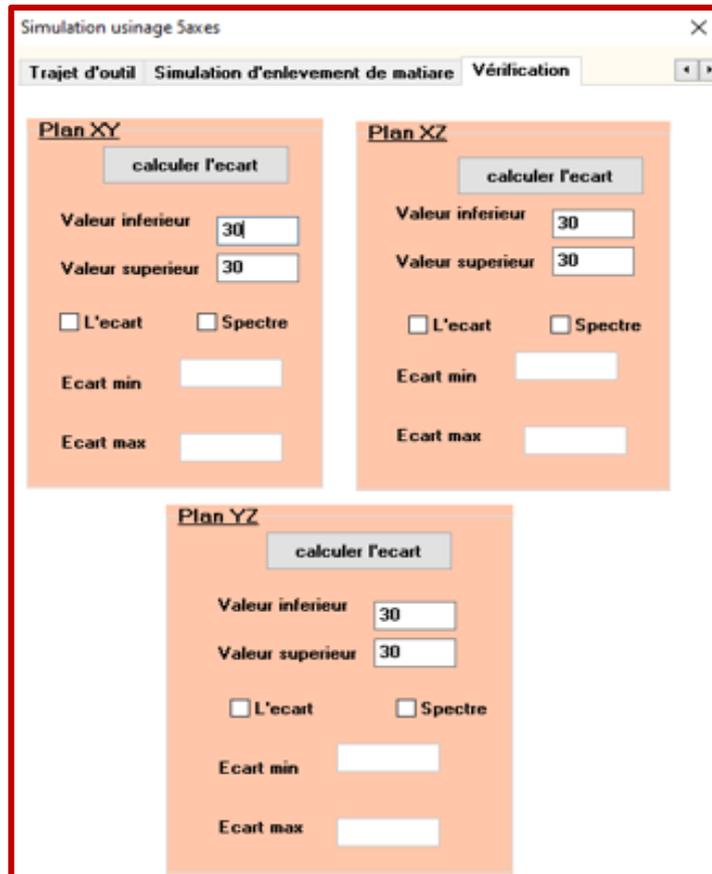


Figure 8. Onglet « Vérification ».

### V. Test et validation :

Arrivée 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. Le test de validation consiste à utiliser un modèle STL d'une pièce issue d'une conception générée dans un logiciel de CAO. Tout le processus de simulation d'enlèvement de matière d'usinage 5-axes est appliqué sur deux exemples. Le premier représente une pièce simple tandis que le second correspond à une pièce de géométrie très complexe.

Nous allons commencer par l'exemple 1 pour mettre en évidence les fonctionnalités de l'application (Figure 9).

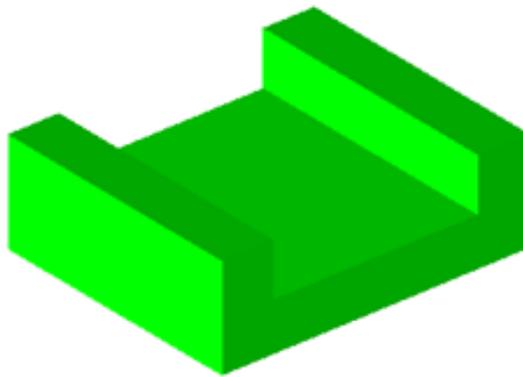


Figure 9. Exemple 1.

- **Etape 1** : lecture du fichier STL de la pièce à usiner (Figure 10).

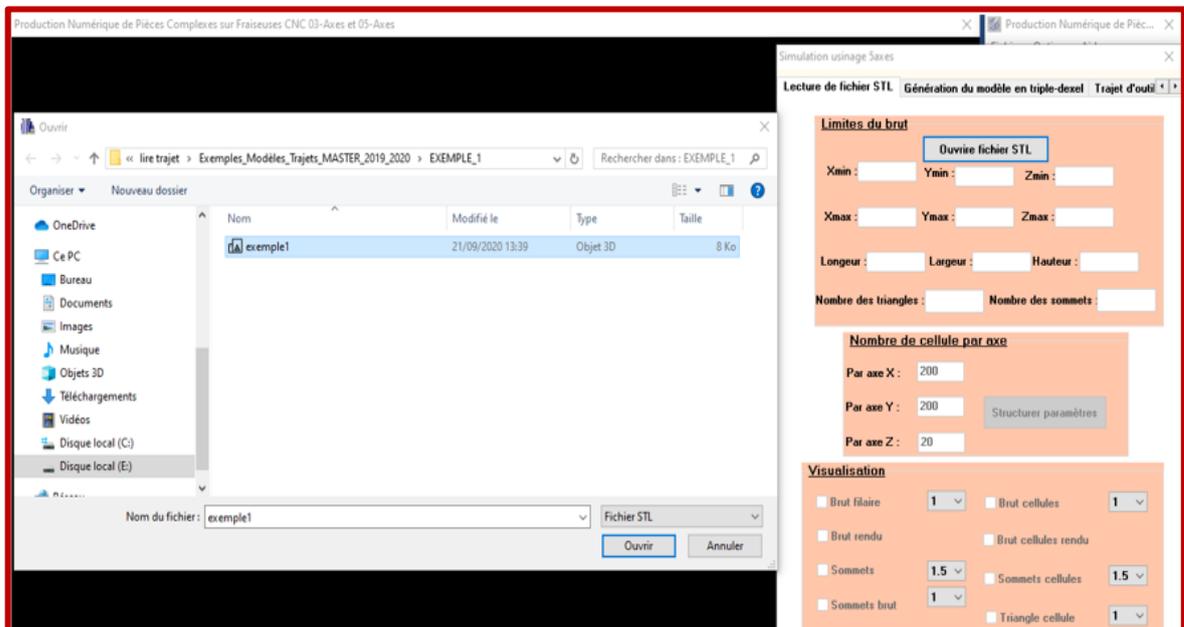


Figure 10. Lecture du fichier STL de la pièce.

Les résultats obtenus lors de la lecture sont :

Tableau 1. Résultats obtenus.

<b>Xmin = -108.422 mm</b>	<b>Ymin = 0.000 mm</b>	<b>Zmin = -42.927 mm</b>
<b>Xmax = 11.578 mm</b>	<b>Ymax = 100.000 mm</b>	<b>Zmax = -2.927 mm</b>
<b>Longueur = 120.00001 mm</b>	<b>Largeur = 100 mm</b>	<b>Hauteur = 40 mm</b>
<b>Nombre de triangles = 28</b>		<b>Nombre de sommets =16</b>

Les autres paramètres sont visualisés par la Figure 11.

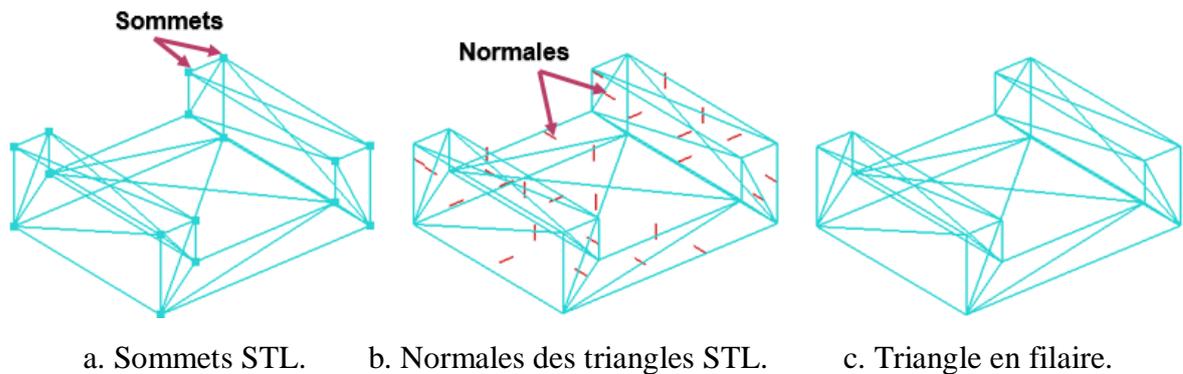


Figure 11. Paramètres visibles après la lecture du fichier.

Pour la création des cellules, les nombres de cellules suivant les axes X, Y et Z sont 5, 6 et 7 respectivement

Les cellules créées sont représentées par la Figure 12.

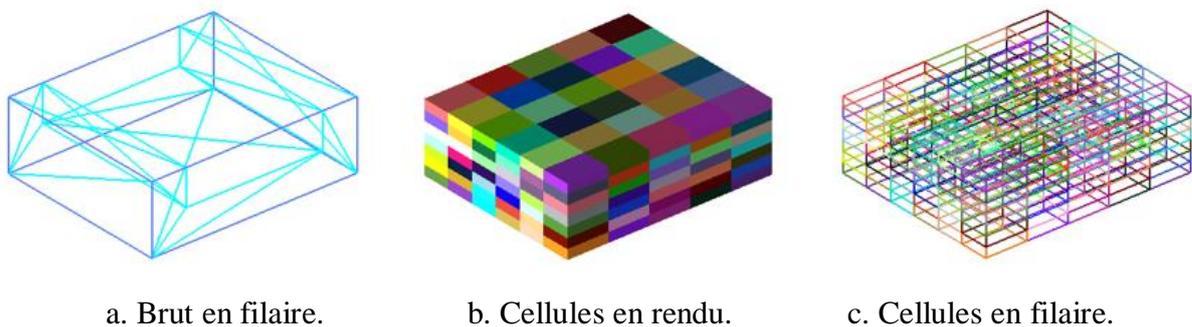
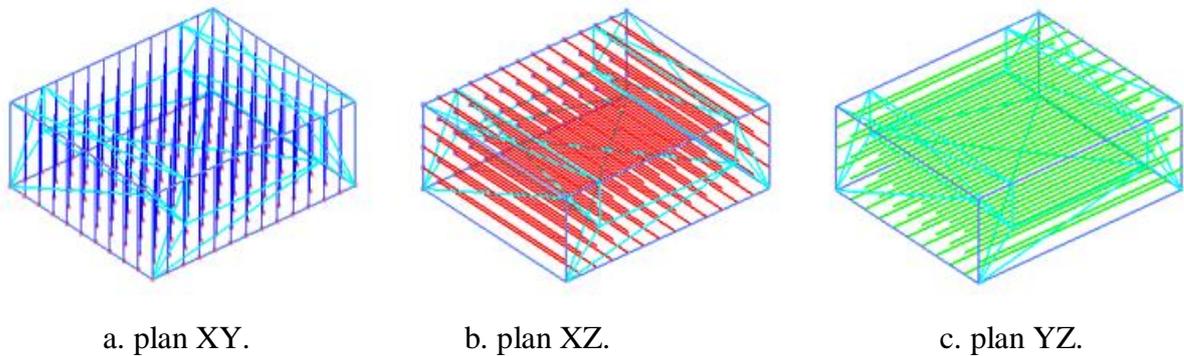


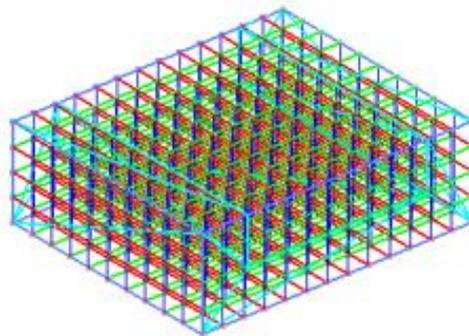
Figure 12. Création des cellules.

- **Etape 2 :** création des matrices en deux dimensions pour chaque plan XY, XZ et YZ en spécifiant le pas suivants les trois axes X, Y et Z. Pour l'exemple de validation, la valeur du pas est égale à 10 pour chaque axe. Afin de visualiser les matrices dans les trois plans,

nous avons afficher les sommets Dexel et les segments par plan (Figure 13) et suivant les trois plans simultanément (Figure 14).

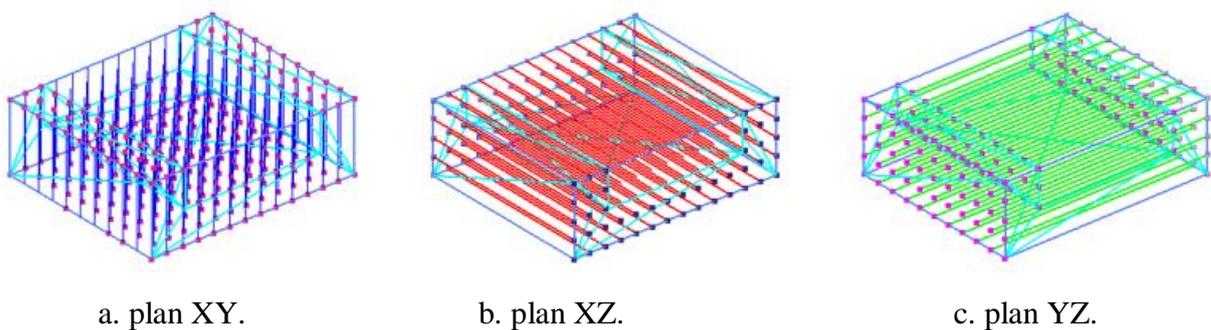


**Figure 13.** Visualisation des sommets et des segments.



**Figure 14.** Création des matrices.

- **Etape 3 :** afin de calculer les points théoriques, nous avons choisi la méthode « barycentrique » parce qu'elle permet de réduire considérablement les temps de calcul comparée à la méthode « d'aire » (Figure 15).



**Figure 15.** Visualisation des points théoriques.

- **Etape 4 :** lecture du fichier trajet d'outils (Figure 16). La Figure 17 montre ce trajet.

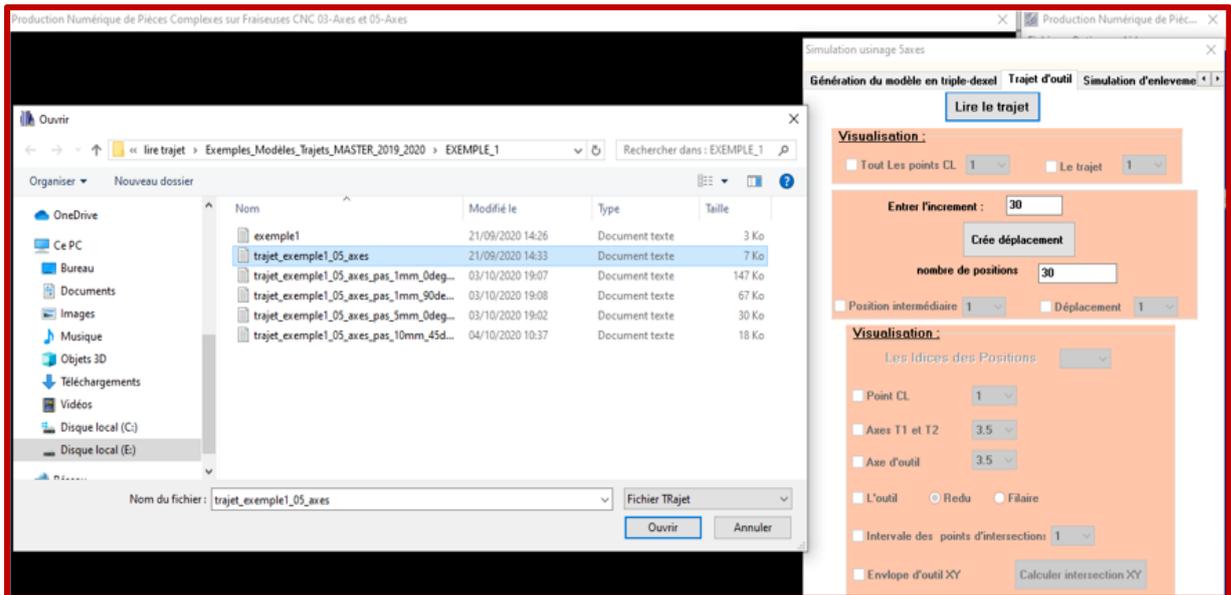


Figure 16. Lecture du fichier trajet d'outils.

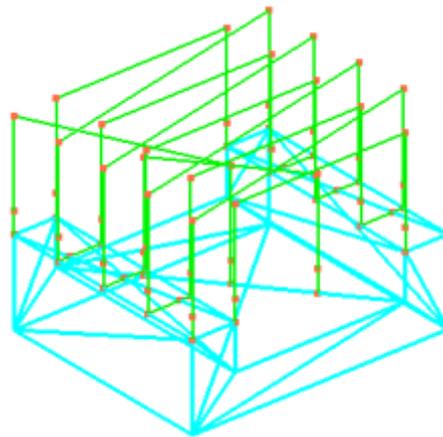


Figure 17. Visualisation du trajet d'outil et des points CL.

- **Etape 5 :** création des déplacements pour chaque plan XY, XZ et YZ en spécifiant la valeur de l'incrément égale à 30mm. La Figure 18 montre les déplacements entre deux points successifs du trajet ainsi que les positions intermédiaires.

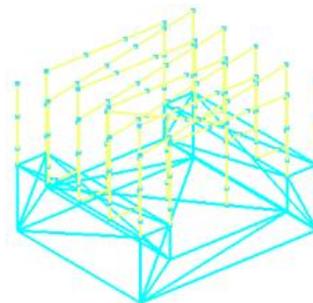
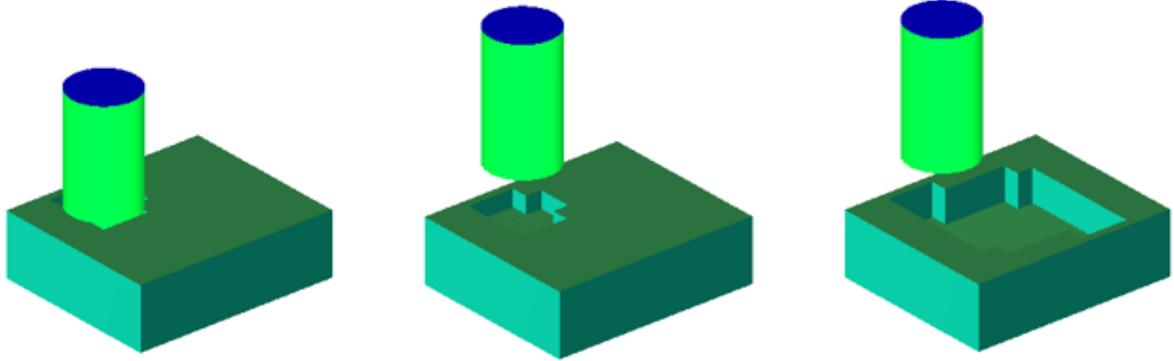


Figure 18. Visualisation des déplacements et positions intermédiaires.

• **Etape 6 :** simulation de l'usinage en chaque position du passage de l'outil. La Figure 19 montre l'aspect visuel de la pièce pour trois positions différentes du trajet d'outils.

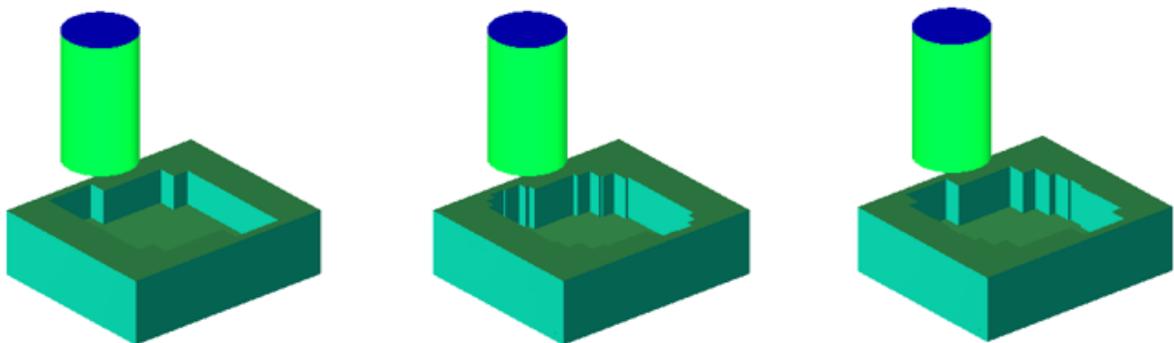


**Figure 19.** Visualisation de la pièce pendant la simulation.

• **Etape 7 :** simulation d'enlèvement de matière pour chaque plan XY, XZ et YZ. En spécifiant la valeur de l'incrément d'affichage :

- Plan XY incrément d'affichage égal à 10.
- Plan XZ incrément d'affichage égal à 135.
- Plan YZ incrément d'affichage égal à 135.

Ensuite, nous avons calculé les intersections pour chaque plan XY, XZ et YZ. Lors de la simulation, les intersections sont données par la Figure 20.



a. Plan XY.

b. Plan XZ.

c. Plan YZ.

**Figure 20.** Simulation d'enlèvement de matière pour chaque plan.

Les extrémités des segments sont visualisées par la Figure 21.

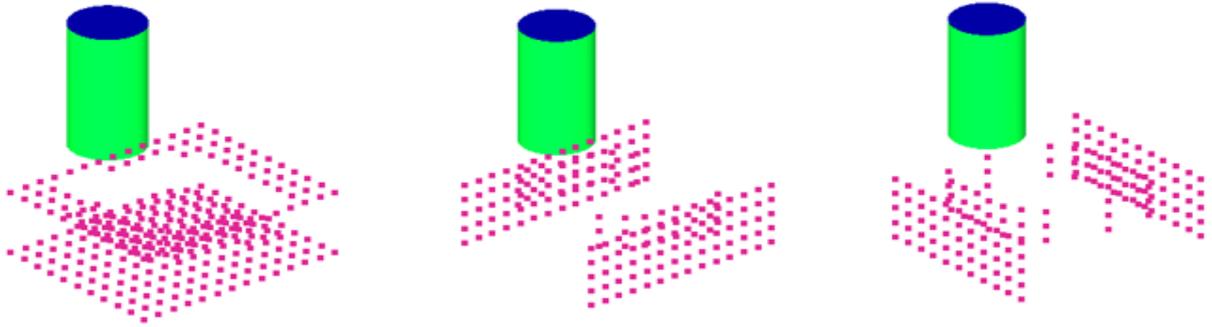
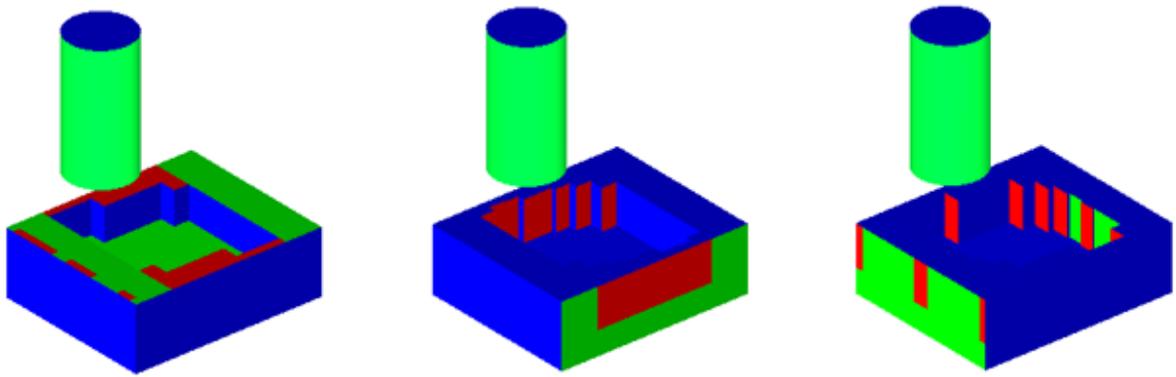


Figure 21. Extrémités des segments.

• **Etape 8 :** vérification de la conformité des surfaces résultantes pour chaque plan XY, XZ et YZ. En spécifiant les valeurs inférieur et supérieur de l'intervalle de tolérance :

- Valeur inférieure est égale à 1mm.
- Valeur supérieure est égale à 1mm.

Les résultats sont donnés par la Figure 22.



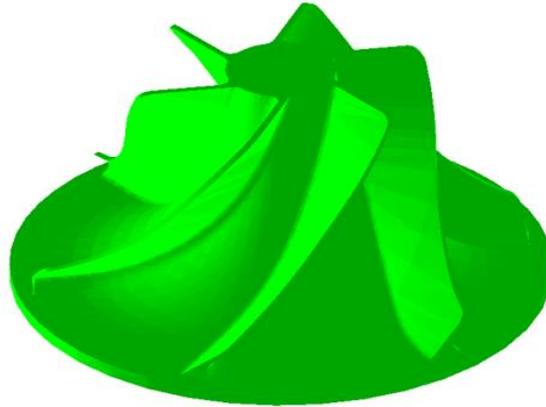
c. Ecart plan XY.

d. Ecart plan XZ.

e. Ecart plan YZ.

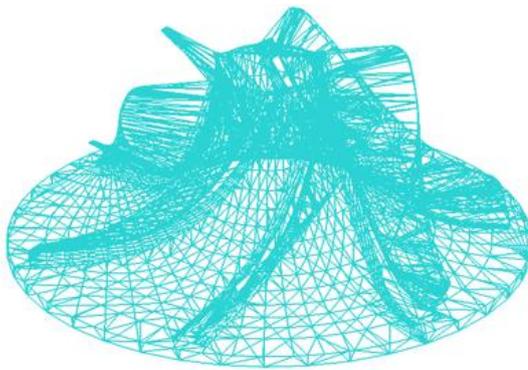
Figure 22. Visualisation des écarts.

Le deuxième exemple de validation est donné par la Figure 23.

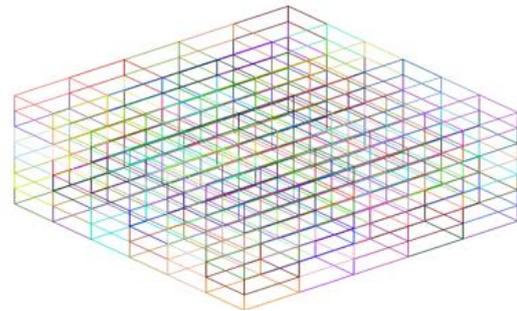


**Figure 23.** Exemple 2 de test.

Après la lecture du fichier STL, la Figure 24 montre les triangles du modèle et les cellules générés.



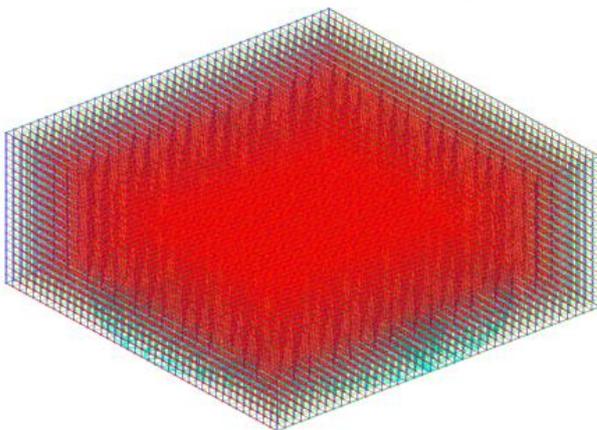
a. Triangles en filaire



b. Cellules en filaire.

**Figure 24.** Visualisation des paramètres.

Pour la création des matrices, nous avons pris un pas égal à 0.5mm (Figure 25).



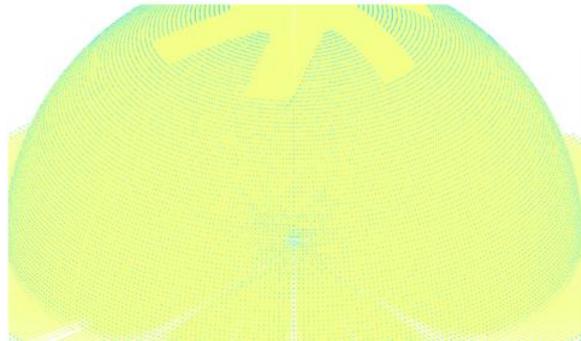
a. Création des matrices



b. Points théoriques du plan XY.

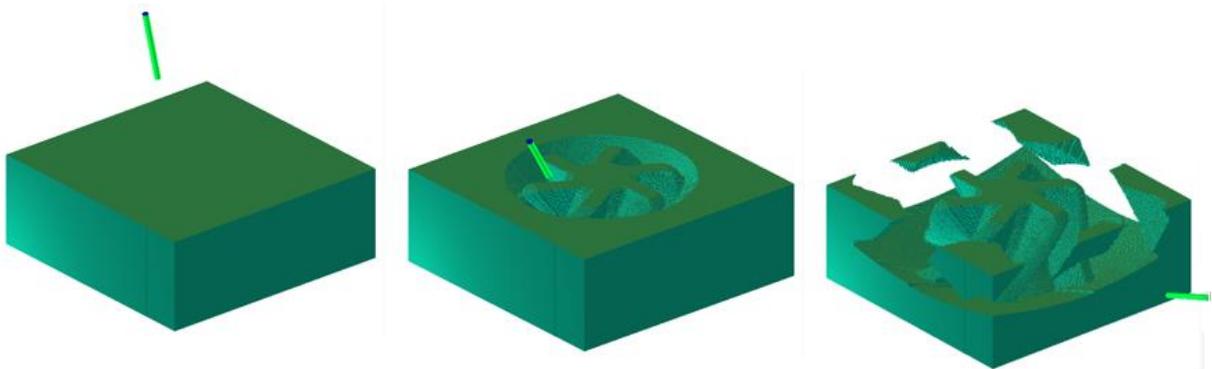
**Figure 25.** Visualisation des matrices et des points théoriques.

Après la lecture du trajet d'outils, nous avons créé les déplacements en spécifiant la valeur de l'incrément égale à 0.5mm. la Figure 26 montre les déplacements et les positions intermédiaires.

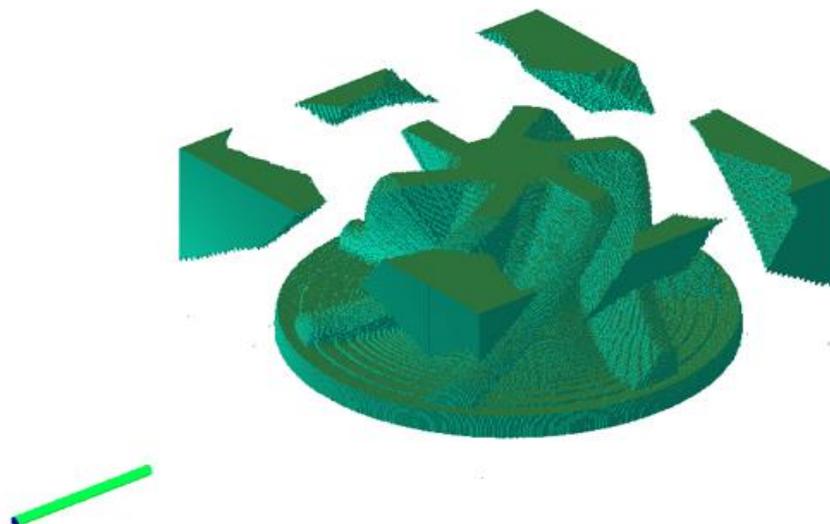


**Figure 26.** Visualisation des déplacements et positions intermédiaires.

Nous avons calculé les intersections en chaque position de passage de l'outil. Les paramètres de visualisation sont montrés dans la Figure 27. La Figure 28 montre le résultat final de la simulation d'enlèvement de matière.



**Figure 27.** Visualisation des intersections dans le plan XY.

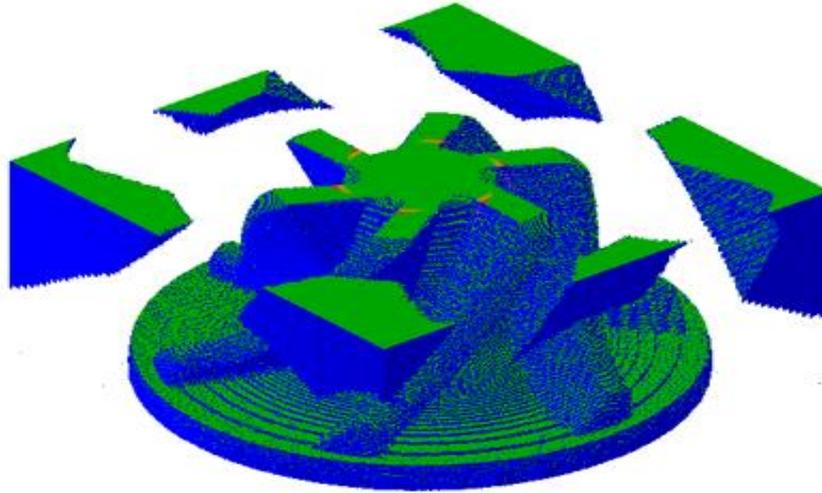


**Figure 28.** Simulation d'enlèvement de matière globale.

En spécifiant les valeurs inférieure et supérieure de l'intervalle de tolérance :

- Valeur inférieure égale à 1mm.
- Valeur supérieure égale à 1mm.

Les écarts sont calculés et les paramètres de visualisation sont montrés par la Figure 29.



**Figure 29.** Visualisation des écarts.

### **Conclusion :**

Au cours de ce chapitre, nous avons présenté notre application afin de valider les résultats obtenus. Dans un premier temps, nous avons présenté l'environnement de développement de notre application ainsi que les interfaces pour décrire les différentes fonctionnalités. Dans un second temps, nous avons testé les différentes fonctionnalités de notre application à travers des exemples depuis la lecture du fichier STL jusqu'à la vérification de la conformité des surfaces résultantes.

## CONCLUSION GENERALE

Le travail réalisé au sein de l'équipe « Conception et Fabrication Assistées par Ordinateur » «CFAO » a pour but, la conception et l'implémentation d'un module logiciel graphique et interactif sous Windows permettant la simulation virtuelle des différentes opérations d'usinage (ébauchage, demi-finition et finition) des pièces complexes discrètes, définies par leurs modèles « STL », sur des fraiseuses numériques à 05-axes, en considérant les outils cylindriques et hémisphériques tout en vérifiant la conformité des surfaces résultantes.

Lors de la réalisation de ce projet, une étude bibliographique relative aux méthodes de conception et d'usinage des pièces complexes en 05-axes a été menée. Par la suite, nous avons proposé une approche pour réaliser la simulation d'enlèvement de matière et pour vérifier la conformité des surfaces résultantes avec les modèles théoriques. L'achèvement du travail réalisé est l'implémentation informatique de l'approche proposée sous Windows en utilisant le langage C++, l'environnement de développement Embarcadero et la bibliothèque graphique OpenGL. Dans la dernière partie, nous avons mis en évidence les fonctionnalités du module logiciel développé à travers des exemples de validation.

Le résultat de l'application développée est l'intégration à l'application logicielle graphique de l'équipe « CFAO » des fonctions qui permettent de :

- Lire et structurer le modèle STL.
- Créer les matrices et calculer les points théoriques.
- Générer le modèle en Triple-Dexel.
- Créer les déplacements et générer les positionnements intermédiaires.
- Calculer les intersections.
- Simuler l'enlèvement de matière en 5-axes.
- Calculer les écarts et dessiner le spectre.
- Vérifier la conformité des surfaces résultantes.

En perspective, nous recommandons de traiter les thématiques suivantes :

- 1 Prise en compte des autres formes d'outils (torique, conique, etc.).

- 2 Proposition d'une nouvelle approche pour la simulation d'enlèvement de matière.
- 3 Calcul des points théoriques avec des méthodes plus optimisées.
- 4 Quantification des volumes de matières enlevées.
- 5 Calcul des efforts de coupe.
- 6 Adaptation des vitesses d'avance.
- 7 Simulation géométrique des fraiseuses 05-axes.
- 8 Génération du « G-Code ».
- 9 Correction du trajet d'outils.

## Références bibliographiques

- [1]. L. Kheris, W. Hedroug, « Conception et Développement d'une Application de Simulation d'Enlèvement de Matière Lors de la Finition des Surfaces Complexes sur Fraiseuses 05-axes », Mémoire de Master, Université Saad Dahlab, Blida, 2016.
- [2]. S. Dubuisson, « Courbes et Surfaces - Algorithmique Graphique et Modélisation », Laboratoire d'Informatique de Paris 6, Sorbonne Université, France, 2007.
- [3]. M. Mahiddini Brahim, « Echange des Données entre Systèmes de CFAO : Application à la Conversion par Approximation des Courbes et Surfaces de Bézier de Degré Elevé », Magister en Génie Mécanique-Productique, Université M'hamed Bougara, Boumerdes, 2006.
- [4]. El Hadj Mimoune Mourad, « Contribution à la modélisation explicite et à la représentation des données des composants industriels : application au modèle PLIB », Thèse de Doctorat, Faculté des sciences fondamentale et appliquées, Université de Poitiers, France, Avril 2012.
- [5]. P. Dubois, A. Aoussat, R. Duchamp, « Prototypage Rapide Généralités », Laboratoire Conception de Produits Nouveaux et Innovation à l'Ecole Nationale Supérieure d'Arts et Métiers de Paris, France, 2000.
- [6]. Eric Cloutier, «Génie de Conception», <https://www.genieconception.ca/une-fraise-cylindrique-pour-composites-894>, 26 août 2013.
- [7]. M. Bey, H. Bendifallah, W. 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, CAM2011, 14-17 novembre 2011, Guelma, Algérie, 2011.
- [8]. H. Belleili et L. Charchal, « Combinaison des Outils Hémisphériques, Cylindriques et Toriques pour la Finition des Surfaces Complexes sur des Fraiseuses à 05-Axes », Mémoire de Master, Université Saad Dahlab, Blida, 2019.
- [9]. S. Djebali, « Optimisation Globale du Processus d'Usinage des Surfaces Gauches », Thèse de Doctorat, Université Toulouse III - Paul Sabatier, France, 2014.
- [10]. M. Bey, T. Zahida, L. Kheidri, et N. Benhenda, « Automatisation de l'Opération de Tréflage des Surfaces Gauches à Partir des Modèles STL », Journées d'Etudes Nationale de Mécanique, JENM'2011, 07-08 Mars, Ouargla, Algérie, 2011.
- [11]. S. Boumenir, M. Debieb, « Conception et Développement d'une Approche Optimale pour la Génération du Modèle Triple-Dexels de Pièces de Formes Complexes », Mémoire de Master, Université Saad Dahlab, Blida, 2018.

- [12]. A. Chourar, H. Nebbat, « Simulation et Vérification des Programmes d'Usinage des Surfaces Gauches sur des Fraiseuses à Commande Numérique à 3-Axes », Mémoire de Master, Université Saad Dahlab, Blida, 2004.
- [13]. Margaret Rouse, «WhatIs.com», <https://whatis.techtarget.com/definition/OpenGL-Open-Graphics-Library>, Mars 2011.
- [14]. Embarcadero Technologies, Embarcadero, <https://www.embarcadero.com/>, 2020.