



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur

et de la Recherche Scientifique

CDTA<sup>7</sup>

UNIVERSITE SAAD DAHLAB DE BLIDA

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

**MEMOIRE DE FIN D'ETUDE**

Pour l'obtention

d'un diplôme de Master en Informatique

Option : Systèmes Informatiques et Réseaux

**Thème :**

**Ebauchage des Pièces Complexes par la  
Stratégie « Plans Parallèles » sur Fraiseuses  
Numériques à 03-Axes**

**Réalisé par :**

M<sup>elle</sup>. SARAOUI Loubna

M<sup>elle</sup>. MESLI Manel

**Soutenu devant :**

Mr. BEY Mohamed	CDTA	Encadreur
Mr. BENDIFALLAH Hassène	CDTA	Encadreur
Mme. TOUBALINE Nesrine	USDB	Promotrice
Mr. HAMOUDA Mohamed	USDB	Président
Mr. KAMECHE Abdallah Hicham	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 l'ébauchage des pièces de formes complexes, définies par leurs modèles STL sur des fraiseuses numériques à 03-axes, en considérant la stratégie d'usinage « Plans Parallèles ». Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle, un module logiciel graphique et interactif permettant de générer automatiquement le trajet d'outils «G-Code» représentant la succession des points de positionnement de l'outil par rapport à la surface tout en évitant les problèmes d'interférence et de collision. Ce travail permettra de réduire les coûts et les temps de génération des programmes.

**Mots Clés :** Pièce Complexe, Modèle STL, Plans Parallèles, G-Code, Ebauchage, Interférence, Collision, Fraiseuse 03-axes.

## 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 ، نهتم بتخشين الأجزاء المعقدة ، المعرفة بواسطة نماذج STL الخاصة بها، على آلات التفرير الرقمية ذات 03 محاور، وذلك باستخدام استراتيجية التصنيع "المستويات المتوازية". يتضمن ذلك تصميم وتطوير ودمج وحدة برمجية رسومية وتفاعلية في منصة البرنامج، لإنشاء مسار الأداة تلقائياً "G-Code" الذي يمثل تعاقب نقاط تموضع الأداة بالنسبة للسطح مع تجنب مشاكل التداخل والاصطدام. يسمح هذا العمل بتحسين جودة المعالجة وتقليل أوقات إنشاء البرامج.

**الكلمات المفتاحية :** الجزء المعقد، نموذج STL، المستويات المتوازية، G-Code، التخشين، التداخل، الاصطدام، آلة التفرير ثلاثية المحور.

## 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 roughing complex parts, defined by their STL models on 03-axis CNC milling machines, considering the "Parallel Planes" machining strategy. This involves designing, developing and integrating into the software platform a graphic and an interactive software module in order to generate automatically toolpaths «G-Code» while avoiding interferences and collisions problems. This work reduces costs and program generation times.

**Keywords :** Complex part, STL model, Parallel planes, G-Code, Roughing, Interference, Collision, 03-axis CNC milling machine.

# Remerciements

*Nous tenons tout d'abord à remercier le bon Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce modeste travail.*

*En second lieu, nous adressons nos sincères remerciements à Mr. Bey Mohamed qui nous a confié ce sujet et qui nous a assuré l'encadrement de notre projet et l'intérêt qu'il nous a porté pour notre travail, sa bienveillance, sa rigueur scientifique ses discussions fructueuses, ses hautes qualités humaines, qui ont constitué une aide précieuse et qui nous a permis de mener à terme ce travail.*

*Nous remercions Mr. Bendifallah Hassène pour ses efforts et sa bonne coordination durant le stage.*

*Par la même occasion, nous adressons nos remerciements à notre promotrice Mme Toubaline Nesrine pour l'orientation et la confiance qui ont constitué un apport considérable sans laquelle ce travail n'aurait pas pu être mené au bon port.*

*Nos remerciements aussi à toute l'équipe du CDTA.*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Enfin, nous tenons également à remercier nos familles et nos amis et toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.*

# *Dédicace*

*À ma très chère mère,*

*Quoi je fasse ou je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.*

*À mon très cher père*

*Tu as toujours été à mes côtés pour me soutenir et m'encourager.*

*Que ce travail traduit ma gratitude et mon affection.*

*À mes très chères sœurs «Lamia», «Nesrine» et «Nada» pour leurs soutiens moraux et leurs conseils précieux tout au long de mes études.*

*À mes chers grands-parents.*

*À mes meilleurs amies «Samira», «Dehbia», «Zineb» et «Amina».*

*À toute ma famille et mes proches, À ma chère «Asma».*

*À ceux qui me donnent de l'amour et de la vivacité et à ceux qui m'ont aidé et supporté dans les moments difficiles.*

*À toute personne utilisant ce document pour un bon usage.*

*Je dédie ce mémoire*

*Saraoui Loubna*

# Dédicace

## **Je dédie ce mémoire 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 mon bien être.

Je vous remercie pour tout le soutien et l'amour que vous me porter 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 sorte que jamais je ne vous déçoive.

## **A Mes Chers Adorables Frères**

Abderrahmane l'aimable et le généreux des frères, Walid et Amine mes petits frères que j'aime profondément.

En témoignage de mon affectation fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès que Dieu, le tout puissant, vous protège et vous garde.

## **A Ma Grand-mère CHERIE**

Qui m'a accompagné par ses prières, sa douceur, puisse Dieu lui prêter longue vie et beaucoup de santé et de bonheur dans les deux vies.

## **A mes chers Oncles, Tantes A Mes chers Cousins Cousines**

Veillez trouver dans ce travail l'expression de mon respect les plus profonds et mon affectation la plus sincères.

**Une spéciale Dédicaces a ces deux personnes qui comptent de ses énormément pour moi, et pour qui je porte beaucoup de tendresse et de respect.**

## **A vous AFFAF ET KAMI**

**Mesli Manel**

---

<b>Résumé</b> .....	1
<b>Remerciements</b> .....	2
<b>Table des matières</b> .....	5
<b>Liste des figures</b> .....	7
<b>Introduction générale</b> .....	10
<b>I . Etude Bibliographique</b> .....	13
Introduction.....	14
I.1. Généralité sur le processus de production de pièce mécanique.....	14
I.1.1. Définition de pièces de formes complexes .....	15
I.1.2. Définition de CFAO.....	15
I.1.3. Conception assistée par ordinateur (CAO) .....	15
I.1.4. Fabrication assistée par ordinateur (FAO).....	16
I.2. Etude du format d'échange de données « STL ».....	17
I.2.1. Surface paramétrique .....	17
I.2.2. Format d'échange de données.....	18
I.3. Généralités sur l'opération d'ébauchage des pièces complexes sur des fraiseuses numériques à 03-axes.....	21
I.3.1. Fraiseuse .....	21
I.3.2. Stratégies d'usinage .....	22
I.3.3. Stratégies d'ébauchage .....	24
I.4. Etude de la Stratégie d'ébauchage « plans parallèles » .....	25
I.4.1. Principe de l'approche plan parallèle.....	26
Conclusion .....	27
<b>II . Conception de l'application</b> .....	28
Introduction.....	29
II.1. Problématique.....	29
II.2. Solution Proposée .....	29
II.2.1. Lecture du modèle STL .....	31
II.2.2. Création des plans et des contours initiaux .....	33
II.2.3. Spécification de la surépaisseur d'usinage et décalage des contours initiaux .....	37
II.2.4. Détermination d'outil optimum pour chaque plan .....	38
II.2.5. Limitation du nombre d'outils .....	42
II.2.6. Spécification de la direction d'usinage et création des passes .....	42
II.2.7. Création des segments d'usinage .....	43

---

II.2.8. Génération du trajet d'usinage suivant les passes .....	46
II.2.9. Génération du trajet d'usinage suivant les contours offset .....	47
II.2.10. Simulation du trajet .....	47
II.3. Modélisation de l'application avec UML .....	47
II.3.1. Diagramme de cas d'utilisation.....	47
II.3.2. Diagramme d'activité.....	51
II.3.3. Diagramme de classes .....	55
Conclusion .....	62
<b>III . Implémentation et Validation .....</b>	<b>63</b>
Introduction .....	64
III.1. Présentation des langages utilisés .....	64
III.1.1. Présentation du langage C++ .....	64
III.1.2. Présentation d'embarcadero C++ Builder 10 Seattle.....	64
III.1.3. Présentation d'OpenGL .....	64
III.2. Matériel utilisé .....	65
III.3. Présentation de l'application.....	65
III.3.1. Fenêtre principale de la plateforme de «CFAO» .....	65
III.3.2. Rubrique «Ebauchage 03-Axes des modèles STL» .....	66
III.4. Présentation de l'application développées.....	66
III.4.1. Lecture du fichier STL .....	67
III.4.2. Génération des contours .....	68
III.4.3. Détermination des outils .....	69
III.4.4. Création des passes .....	70
III.4.5. Génération du Trajet d'Ebauchage.....	71
III.4.6. Simulation de trajet .....	72
III.5. Test et validation.....	73
III.6. Validation du trajet par simulation .....	77
Conclusion .....	78
<b>Conclusion générale .....</b>	<b>79</b>
<b>Références bibliographiques.....</b>	<b>81</b>

## Chapitre I : Étude bibliographie

<b>Figure I.1.</b> Pièces de formes complexes .....	15
<b>Figure I.2.</b> Entités géométriques .....	16
<b>Figure I.3.</b> Polyèdres de subdivision .....	16
<b>Figure I.4.</b> Modes de fraisage .....	16
<b>Figure I.5.</b> Localisation d'un point sur une surface paramétrique.....	18
<b>Figure I.6.</b> Modes de triangulation d'une surface.....	18
<b>Figure I.7.</b> Modèle STL d'un objet .....	19
<b>Figure I.8.</b> Syntaxe d'un fichier STL ASCII .....	20
<b>Figure I.9.</b> Syntaxe d'un fichier STL binaire.....	20
<b>Figure I.10.</b> Types de fraiseuses .....	22
<b>Figure I.11.</b> Modes de fraisage .....	22
<b>Figure I.12.</b> Opérations d'usinage.....	23
<b>Figure I.13.</b> Formes de fraise.....	23
<b>Figure I.14.</b> Problèmes d'usinage .....	24
<b>Figure I.15.</b> Ebauchage avec un outil cylindrique .....	24
<b>Figure I.16.</b> Stratégie des « Contours Décalés ».....	25
<b>Figure I.17.</b> Zones non usinées avec la stratégie « Contours Décalés » .....	25
<b>Figure I.18.</b> Stratégie des « Plans Parallèles » .....	26
<b>Figure I.19.</b> Différents modes de balayage .....	27

## Chapitre II : Conception de l'application

<b>Figure II.1.</b> Organigramme de la démarche proposée .....	30
<b>Figure II.2.</b> Limites du brut .....	31
<b>Figure II.3.</b> Création des cellules.....	32
<b>Figure II.4.</b> Sommets des cellules .....	32
<b>Figure II.5.</b> Affectation d'un triangle aux cellules .....	33
<b>Figure II.6.</b> Plans horizontaux .....	33
<b>Figure II.7.</b> Différents cas d'intersection.....	34
<b>Figure II.8.</b> Intersection entre un plan et un segment .....	35
<b>Figure II.9.</b> Types des segments .....	35
<b>Figure II.10.</b> Création des contours initiaux .....	36
<b>Figure II.11.</b> Création des contours décalés .....	37
<b>Figure II.12.</b> Détermination de la zone d'interférence .....	39
<b>Figure II.13.</b> Insertion de points intermédiaires dans un segment .....	40
<b>Figure II.14.</b> Appartenance d'un point à un cercle .....	41
<b>Figure II.15.</b> Appartenance d'un point à un rectangle .....	42
<b>Figure II.16.</b> Création des passes d'usinage.....	43
<b>Figure II.17.</b> Création des Contours offset .....	43
<b>Figure II.18.</b> Position des différents points par rapport à $AB$ .....	44
<b>Figure II.19.</b> Segments d'usinage.....	45
<b>Figure II.20.</b> Appartenance d'un point à un triangle .....	46
<b>Figure II.21.</b> Trajet d'usinage.....	46

<b>Figure II.22.</b> Lissage des contours.....	47
<b>Figure II.23.</b> Diagramme de cas d'utilisations générale.....	48
<b>Figure II.24.</b> Cas d'utilisations «Lire Fichier STL».....	48
<b>Figure II.25.</b> Cas d'utilisations «Générer les contours».....	49
<b>Figure II.26.</b> Cas d'utilisations «Déterminer les outils optimums» .....	49
<b>Figure II.27.</b> Cas d'utilisations «Créer les passes» .....	50
<b>Figure II.28.</b> Cas d'utilisations «Générer le trajet d'usinage» .....	50
<b>Figure II.29.</b> Cas d'utilisations «Simuler le trajet» .....	51
<b>Figure II.30.</b> Diagramme d'activité général .....	51
<b>Figure II.31.</b> Diagramme d'activité «Lire Fichier STL» .....	52
<b>Figure II.32.</b> Diagramme d'activité «Générer les contours» .....	52
<b>Figure II.33.</b> Diagramme d'activité «Déterminer les outils optimums» .....	53
<b>Figure II.34.</b> Diagramme d'activité «Créer les passes» .....	53
<b>Figure II.35.</b> Diagramme d'activité «Générer le trajet d'usinage» .....	54
<b>Figure II.36.</b> Diagramme d'activité «Simuler le trajet» .....	54
<b>Figure II.37.</b> Diagramme de classes .....	56
<b>Figure II.38.</b> Classe Sommet_Ebauchage.....	57
<b>Figure II.39.</b> Classe Triangle_Ebauchage.....	57
<b>Figure II.40.</b> Classe Brute_Ebauchage .....	57
<b>Figure II.41.</b> Classe Cellule_Ebauchage.....	58
<b>Figure II.42.</b> Classe ModelSTL_Ebauchage.....	58
<b>Figure II.43.</b> Classe Segment_Ebauchage .....	59
<b>Figure II.44.</b> Classe Contour_Ebauchage .....	59
<b>Figure II.45.</b> Classe Outil_Ebauchage .....	60
<b>Figure II.46.</b> Classe Droite_Ebauchage .....	60
<b>Figure II.47.</b> Classe PlanCoup _Ebauchage .....	61
<b>Figure II.48.</b> Classe Trajet_Ebauchage.....	62

## Chapitre III : Implémentation et Validation

<b>Figure III.1.</b> Environnement de « CFAO » .....	66
<b>Figure III.2.</b> Rubrique « Ebauchage 03-Axes des modèles STL ».....	66
<b>Figure III.3.</b> Onglets de l'application développée.....	67
<b>Figure III.4.</b> Onglet « Lecture du fichier STL ».....	68
<b>Figure III.5.</b> Onglet « Génération des contours ».....	69
<b>Figure III.6.</b> Onglet « Détermination des outils » .....	70
<b>Figure III.7.</b> Onglet « Création des passes » .....	71
<b>Figure III.8.</b> Onglet « Génération du Trajet d'Ebauchage » .....	72
<b>Figure III.9.</b> Onglet « Simulation de trajet » .....	73
<b>Figure III.10.</b> Modèle STL de la pièce de test.....	73
<b>Figure III.11.</b> Création des cellules et affectations des points.....	74
<b>Figure III.12.</b> Génération des contours.....	74
<b>Figure III.13.</b> Positionnement de l'outil pour le premier plan de coupe .....	75
<b>Figure III.14.</b> Zones d'usinage .....	76
<b>Figure III.15.</b> Trajet d'ébauchage avec les deux modes de balayage.....	77

<b>Figure III.16.</b> Simulation du trajet en 03-Axes .....	77
<b>Figure III.17.</b> Validation du trajet par simulation virtuelle de l'usinage. ....	78

## Présentation du sujet :

L'informatique industrielle désigne la branche de l'informatique regroupant l'ensemble des techniques de conception, d'analyse et de programmation des systèmes informatiques à vocation industrielle. Constituant un domaine très vaste, elle trouve de multiples applications dans l'industrie.

C'est donc grâce à l'informatique industrielle que l'on programme les automates affectés à la production, mais également que l'on détecte leurs pannes et que l'on planifie leur maintenance. Bien souvent, c'est même la chaîne de fabrication, voire l'usine entière qui est pilotée par l'outil informatique, que ce soit un seul ordinateur central ou un réseau d'ordinateurs reliés ensemble.

Cet outil a permis aux entreprises d'augmenter la productivité et la qualité des produits réalisés. Pour réduire les temps de fabrication, minimiser les coûts et pour réduire le cycle de développement d'un produit qui passe forcément par la réduction des délais de conception et de fabrication.

Parmi les domaines issus de l'essor informatique, le domaine de la conception et la fabrication mécanique. Les grandes entreprises utilisent des logiciels de CFAO « Conception et Fabrication Assistées par Ordinateur » pour assister les ingénieurs dans les phases de conception, d'analyse et de planification de l'usinage sur des machines-outils à commande numérique (MOCN).

Les surfaces gauches (moules, matrices, ...etc.) sont très rencontrées dans notre vie quotidienne. Elles sont utilisées actuellement dans divers domaines d'activité tels que l'aéronautique et l'automobile. Ces pièces sont conçues dans le but d'assurer des fonctions inscrites dans le cahier des charges. En raison de leurs complexités géométriques, elles sont usinées sur des fraiseuses à commande numériques à 03-axes, à 04 ou à 05-axes par la prise en compte de plusieurs aspects tels que les modèles des surfaces, les stratégies d'usinage, les formes et les dimensions des outils, les paramètres d'usinage, ...etc.

L'usinage de ces surfaces passe généralement par trois étapes : ébauchage, demi-finition et finition. Après la spécification d'une surépaisseur d'usinage, l'opération ébauchage consiste à utiliser des outils cylindriques pour enlever le maximum de matière dans des plans de coupe horizontaux. Pour cette opération, deux stratégies d'usinage peuvent être utilisées : «Plans Parallèles» et «Contours Décalés». Pour toutes ces étapes, il est nécessaire de choisir judicieusement les formes et les dimensions des outils ainsi que les stratégies d'usinage minimisant les temps d'usinage et les coûts de fabrication. A la fin, il faut générer un ensemble d'instructions écrites dans un langage propre à la machine appelé «G-Code » qui sera transmis à la MOCN pour réaliser l'usinage.

## **Problématique :**

L'adéquation entre la forme usinée et la forme attendue par le concepteur, dépend de l'aptitude de chacune des activités du processus de conception et de fabrication à modéliser ou à produire la géométrie attendue. Il est nécessaire d'intégrer les contraintes inhérentes à chaque activité, et notamment celles de fabrication afin que le modèle de référence généré par le concepteur soit en adéquation avec les procédés d'obtention retenus. Cette intégration n'est cependant pas suffisante pour atteindre le niveau de qualité recherché car de nouvelles erreurs sont introduites lors de la génération des trajectoires. En effet, l'extraction de la géométrie nominale et sa remodelisation sous forme de trajets d'usinage introduit de nombreuses approximations.

Pour l'opération d'ébauchage plusieurs stratégies peuvent être utilisées. La stratégie «Plans Parallèles » est plus robuste et très utilisée dans les logiciels de fabrication. Elle est souhaitable pour la génération du trajet pour l'ensemble de la surface. La trajectoire d'outil résulte de l'intersection de la surface avec des plans parallèles dans l'espace cartésien. La distance entre les plans est déterminée en se basant sur certaines contraintes technologiques. Cette méthode peut être utilisée pour les surfaces composées et pour les surfaces définies par le format STL.

La géométrie complexe de ces surfaces rend cette tâche très difficile. Le but recherché est une méthodologie pour optimiser l'opération d'ébauchage des pièces de formes complexes à partir de leurs modèles STL sur des fraiseuses numériques à 03-axes par la stratégie «Plans Parallèles» et le choix des outils optimums pour assurer un minimum de temps, un usinage de qualité, continu et sans interférences.

## **Objectifs et besoins du travail :**

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).

Les travaux menés dans le cadre de ce mémoire portent sur « Ebauchage des Pièces Complexes par la Stratégie "Plan Parallèles" sur Fraiseuses Numériques à 03-axes ». Il s'agit de mettre en place un module logiciel graphique et interactif permettant la génération des contours initiaux qui délimitent la matière à enlever, le choix automatique des dimensions des outils cylindriques pour chaque plan de coupe permettant d'éviter les interférences, la génération des passes et en fin la génération du trajet d'outils représentant la succession des points de positionnement de l'outil par rapport à la surface. Ce travail permettra de réduire les coûts, d'améliorer la qualité d'usinage et de réduire les temps de génération des programmes.

## **Structuration du mémoire :**

Le présent mémoire est composé de trois chapitres :

➤ Le premier chapitre est consacré à l'étude bibliographique du processus de production des pièces mécaniques, le format d'échange de données et des généralités sur l'opération d'ébauchage des pièces complexes sur des fraiseuses numériques à 03-axes ainsi que la stratégie d'ébauchage « Plans Parallèles ».

➤ Le deuxième chapitre est réservé à la présentation de la solution proposée et à l'étude conceptuelle de l'application.

➤ Le dernier chapitre présente l'implémentation informatique de l'application et les tests de validation des résultats du travail effectué.

Cette mémoire se termine par une conclusion générale et des perspectives.



# Chapitre I

Etude

bibliographique

## Introduction

Tous les secteurs industriels sont aujourd'hui soumis à une pression économique importante et à la concurrence imposée par le phénomène de la globalisation des marchés. Pour répondre à cet état de fait, les entreprises sont contraintes d'innover et d'améliorer le cycle d'élaboration des produits et/ou du processus depuis la mise sur le marché des produits. Les demandes des clients et/ou du marché correspondent à de profonds chargements et imposent aux entreprises de fournir un produit ou/et des services respectant les exigences de « prix toujours plus faible, en proposant un choix varié, disponible et de bonne qualité ». Les stratégies mises en place par les entreprises sont diverses et conduisent à des choix industriels répondant partiellement à ces exigences (délocalisation, fabrication de produits intermédiaires, transfert de technologie, automatisation de la production, nouveaux systèmes manufacturiers, développement de la logistique, etc.) [1].

En mécanique industrielle, la fabrication d'une pièce se fait par l'usinage. Il s'agit d'un enlèvement de la matière par un outil coupant. La qualité d'usinage est déterminée par les erreurs de forme. Une surface usinée est le résultat de la reproduction géométrique et cinématique de la forme et de la trajectoire de la pointe de l'outil.

Les surfaces de formes complexes dites « surface gauches » sont très répandues dans le domaine de l'industrie manufacturière. Le processus de réalisation de ces surfaces a largement été modifié par l'utilisation de nouvelles techniques comme l'usinage à grande vitesse et la maquette numérique, c'est-à-dire la modélisation complète du produit dans un système de «CFAO» (Conception et Fabrication Assistées par Ordinateur). Par conséquent, la forme finale de la pièce est directement associée à la forme usinée par le mouvement de l'outil [2].

Ce chapitre est organisé en quatre parties. Tout d'abord une analyse du processus de conception et de fabrication des pièces mécaniques est présentée, ainsi que les techniques de modélisation des surfaces et la génération de la trajectoire. Dans la deuxième partie, les formats d'échange de données sont étudiés. Par la suite, des généralités sur l'opération d'ébauchage des pièces de formes complexes sur des fraiseuses numériques à 03-axes et les différentes stratégies d'usinage sont présentées. Dans la dernière partie, l'opération d'ébauchage avec la stratégie des « Plans Parallèles » sont détaillés.

### I.1. Généralité sur le processus de production de pièces mécaniques

La production née à partir d'une idée, puis vient la conception, pour aboutir enfin à la fabrication. Pour cela, le processus de production de pièces mécaniques se décompose en deux activités principales; la première est la conception « CAO » (Conception Assistée par Ordinateur) ensuite la fabrication « FAO » (Fabrication Assistée par Ordinateur).

### I.1.1. Définition des pièces de formes complexes

Les pièces de formes complexes sont des pièces qui ont des formes libres et difficiles à réaliser. Ils ont des surfaces qui ne peuvent pas être décomposées en éléments géométriques simples (point, cercle, sphère, droite, cylindre, cône, plan) (Figure I.1) [3].

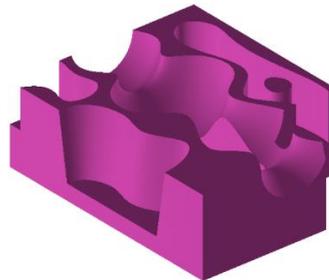


Figure I.1. Pièces de formes complexes.

### I.1.2. Définition de « CFAO »

La « CFAO » signifie **C**onception et **F**abrication **A**ssistées par **O**rdinateur). La «CFAO» a pour champs d'applications l'ensemble du processus de conception «CAO» et de fabrication « FAO » des entreprises quel que soit le produit final. Elle s'intègre dans l'ensemble des outils informatiques permettant à un opérateur d'utiliser et de transférer des données [3].

### I.1.3. Conception assistée par ordinateur « CAO »

La conception d'un produit consiste à proposer ou à reconcevoir des solutions existantes pour répondre aux contraintes du cahier des charges. L'obtention d'une solution n'est pas directe sauf pour des problèmes extrêmement simples. Le processus est plutôt itératif au cours duquel un objet est conçu et modifié itérativement jusqu'à satisfaction des contraintes imposées. La conception d'un produit est basée sur la modélisation géométrique [4].

#### I.1.3.1. Modélisation géométrique

Au cours de la conception d'une pièce ou d'un objet, trois catégories de modèles géométriques peuvent être utilisées pour décrire les surfaces frontières [4] :

- **Modèle polyédrique** : des entités géométriques simples (points, segments, etc.) sont utilisées dans la modélisation d'un objet (Figure I.2). Il permet la représentation des objets en 3D et il est généré par construction de surfaces à partir de nuages de points numérisés ou bien par facettisation de modèles « CAO ».

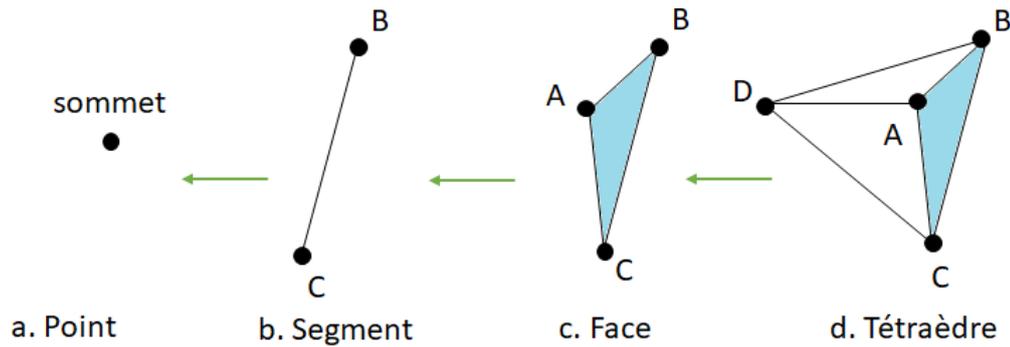


Figure I.2. Entités géométriques.

- **Modèle paramétrique** : il permet d'utiliser des propriétés mathématiques pour étudier les courbes et les surfaces paramétriques, ce modèle est utilisé pour le design ou la conception en mécanique.

- **Modèle de subdivision** : il s'appuie sur les modèles polyédriques et permet de modéliser des surfaces complexes et lisses par des étapes de subdivision itératives (Figure I.3).

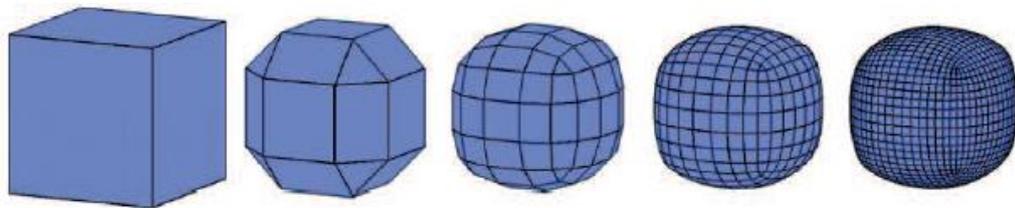


Figure I.3. Polyèdres de subdivision.

## I.1.4. Fabrication assistée par ordinateur « FAO »

La fabrication assistée par ordinateur « FAO » désigne les logiciels d'assistance à la programmation des machines d'usinage pour la production automatisée des pièces mécaniques.

### I.1.4.1. Fabrication

Les pièces de formes complexes « surfaces gauches » sont obtenues par usinage sur des machines d'usinage « fraiseuses » suivant deux modes : usinage en bout par balayage d'un outil (Figure I.4.a) et usinage par le flanc d'un outil cylindrique ou conique (Figure I.4.b) [2].

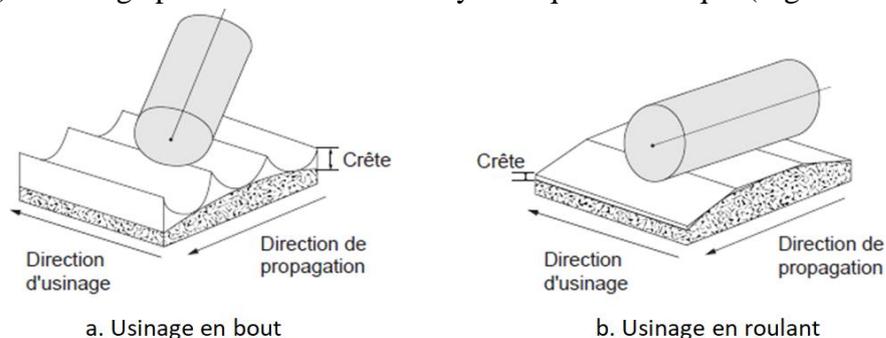


Figure I.4. Modes de fraisage.

### **I.1.4.2. Programmation de commande numérique (CN)**

Les fraiseuses utilisées dans l'usinage des pièces de formes complexes sont commandées numériquement, c'est-à-dire elles ont besoin d'un programme pour piloter la machine numériquement « commande numérique ». La programmation d'une machine passe par les étapes suivantes [5] :

- Définir la géométrie de la pièce à usiner.
- Définir la cinématique de la machine et sa commande.
- Définir les outils.
- Définir le brut de la pièce.
- Définir les parcours d'outils.
- Générer le programme de commande numérique.
- Simuler et vérifier le programme de commande numérique.

## **I.2. Etude du format d'échange de données «STL»**

Il est essentiel que les transferts de données entre les différents logiciels soient correctement effectués afin d'éviter toute perte d'information. Les outils de numérisation ont pour objectif de recueillir des informations sur la topologie de l'objet à reproduire. Celles-ci se présentent sous la forme de nuages de points de coordonnées X, Y et Z. Ces nuages de points, traités par les logiciels de reconstruction de surfaces, permettent d'aboutir à des modèles surfaciques de différents types (Béziers, NURBS, ...etc.) exportables aux formats natifs ou neutres (IGES, VDA, SET, STL, ...etc.) [6].

### **I.2.1. Surface paramétrique**

Pour la modélisation géométriques, deux approches peuvent être utilisées : modélisation géométrique continue (Bézier, B-Splines ou NURBS) et modélisation géométrique discrète (maillages ou surfaces de subdivision).

#### **I.2.1.1. Définition d'une surface paramétrique**

C'est une représentation continue de la surface d'un objet défini mathématiquement. Dans ce modèle, à chaque point **P** de coordonnées  $(u, v)$  dans le plan paramétrique est associé un point dans l'espace de coordonnées **X**, **Y** et **Z** (Figure I.5).

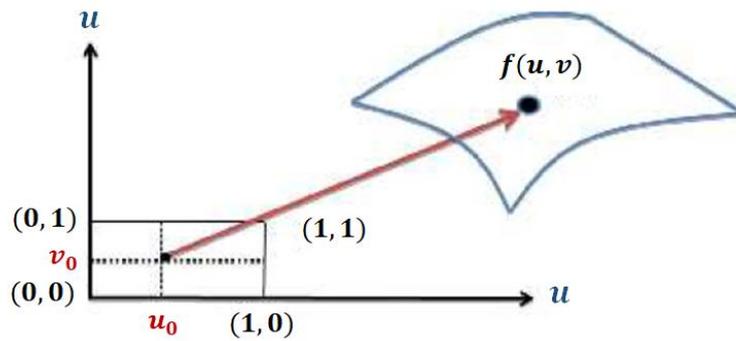
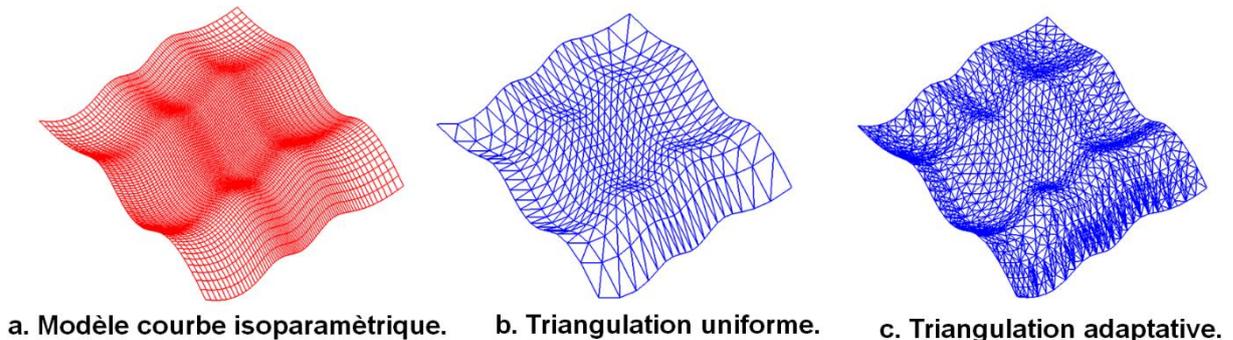


Figure I.5. Localisation d'un point sur une surface paramétrique.

### I.2.1.2. Triangulation

Les surfaces gauches sont conçues par la jonction d'un grand nombre de surfaces complexes. Pour générer la trajectoire d'usinage, il est nécessaire de passer par une étape d'approximation de ces surfaces compliquées par des entités géométriques plus simples. Donc la première étape consiste à approximer chaque surface par des triangles, c'est la triangulation. Deux modes de triangulations peuvent être utilisés [18] :

- **Triangulation uniforme** : le nombre de triangles est défini uniformément, c'est-à-dire les nombres de points suivant les deux directions du plan paramétrique  $UV$  sont fixés (Figure I.6.b).
- **Triangulation adaptative** : les triangles sont créés suite à des subdivisions itératives et adaptatives à partir des critères de précision imposés (Figure I.6.c).



a. Modèle courbe isoparamétrique. b. Triangulation uniforme. c. Triangulation adaptative.

Figure I.6. Modes de triangulation d'une surface.

### I.2.2. Format d'échange de données

L'échange de données entre la CAO et la FAO consiste à échanger des fichiers contenant des données sur les modèles 3D, codées selon différents formats, mais il pose des problèmes de compatibilité. Ils existent deux méthodes de transfert des données [7] :

- **Méthode d'interface directe** : dédiées à un couple de logiciels seulement. Elle permet une meilleure adaptation possible entre les deux logiciels, mais nécessitent une mise à jour chaque fois qu'un des deux logiciels est mis à jour.

- **Méthode de formats neutres** : c'est-à-dire des formats indépendants des logiciels de conception. Les formats d'échange standard les plus utilisés IGES, STEP, STL.

### I.2.2.1. Format IGES

IGES (**I**nitial **G**raphics **E**xchange **S**pecification) est un standard d'échange de données de dessin 2D et 3D entre différents systèmes de CAO. Le principal objectif est l'échange des dessins techniques. Il est utilisé généralement en mécanique et en aéronautique [7].

### I.2.2.2. Format STEP

STEP (**S**Tandard for **E**xchange of **P**roduct model data) est un standard d'échange de données dont l'objectif était de définir une représentation non ambiguë des données du produit, interprétable par tout système informatique, et couvrant tout le cycle de vie des produits [8].

### I.2.2.3. Format STL

STL (**S**Téréo **L**ithographie) est un format d'échange de données entre la « CAO » et la « FAO », développé par la société 3D Systems. Il permet de représenter des modèles par facettes triangulaires, c'est-à-dire que chaque facette est un triangle [9]. Les triangles ne peuvent se croiser qu'aux arêtes et sommets communs. Il fournit une meilleure approximation de la surface de l'objet. Il est généré par l'ensemble des logiciels de « CAO ».

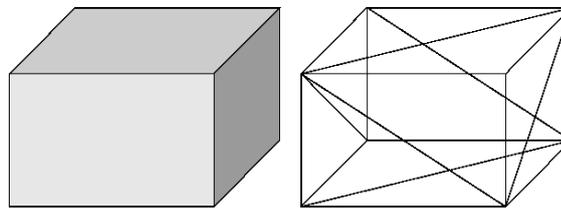


Figure I.7. Modèle STL d'un objet.

#### I.2.2.3.1. Structure d'un fichier STL

Le format de fichier STL fournit deux manières différentes pour stocker des informations sur les facettes triangulaires : codage ASCII et codage binaire.

- **Format du fichier ASCII** : il est lisible et peut être modifié dans un éditeur de texte (Figure I.8). Il commence par une ligne contenant le mot obligatoire « **solid** » suivi du nom du modèle 3D. Il contient des informations sur les triangles de couverture, les sommets et le vecteur normal de chaque triangle. Il se termine par la ligne obligatoire « **endsolid** » [10].

```

solid name
  {
    facet normal  $n_i n_j n_k$ 
      outer loop
        vertex  $v1_x v1_y v1_z$ 
        vertex  $v1_x v1_y v1_z$ 
        vertex  $v1_x v1_y v1_z$ 
      endloop
    endfacet
  }
endsolid name
  
```

Figure I.8. Syntaxe d'un fichier STL ASCII.

- **Format du fichier STL binaire :** les fichiers binaires STL sont organisés comme un en-tête de 84 octets (les premiers 80 octets sont utilisés pour obtenir des informations sur le fichier, et les 4 derniers octets représente le nombre de facettes triangulaires), suivi d'enregistrements de 50 octets (sont utilisé pour représenter les composantes x, y et z du vecteur normal pour chaque facette, chaque composant utilise 4 octets), les deux derniers octets ne sont pas utilisés (Figure I.9) [11].

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

Figure I.9. Syntaxe d'un fichier STL binaire.

### I.2.2.3.2. Avantages et inconvénients du modèle STL [10]

- **Avantages :**
  - Il est universel et pris en charge par presque tous les logiciels « CAO ».
  - Il est plus simple et universel.

- **Inconvénients :**

- Il ne peut pas stocker des informations telles que la couleur, le matériau, etc.
- Approximation de la frontière par des facettes triangulaires.
- Le fichier généré est souvent très volumineux.
- Les rayons de courbures très faibles exigent un grand nombre de triangles.
- Redondance de l'information car les sommets de chaque triangle sont partagés par plusieurs triangles ce qui augmente considérablement la taille de fichier.

### **I.3. Généralités sur l'opération d'ébauchage des pièces complexes sur des fraiseuses numériques à 03-axes**

Les pièces complexes utilisées dans la conception et la fabrication sont usinées sur des fraiseuses à commande numérique multiaxes. A partir de leurs modèles, la forme finale de ces pièces est obtenue en trois étapes : ébauchage, demi-finition et finition. Pour l'ébauchage, différentes stratégies peuvent être utilisées (« Plans parallèles », « Contours décalés », etc.).

#### **I.3.1. Fraiseuse**

Une fraiseuse est une machine-outil utilisée pour usiner des pièces mécaniques, à l'aide d'un outil « fraise » qui est fixé à une broche qui le fait tourner (mouvement de coupe), et peut se déplacer en translation par rapport à la pièce, suivant trois directions (mouvement d'avance). Elle est capable d'enlever de la matière sur une pièce initiale afin de la transformer en une autre pièce. Ils existent des fraiseuses de différents types selon l'orientation de l'outil (verticales, horizontales et universelles). Les fraiseuses selon le nombre des axes peuvent être de trois (03), quatre (04), et cinq (05) axes où un axe représente un degré de liberté (mouvement) [12, 13].

##### **I.3.1.1. Fraiseuse à 03-axes**

Une fraiseuse à trois axes peut être à broche verticale ou à broche horizontale (Figure I.10) [13]. Les axes de la machine sont définis comme suit :

- Axe Z est parallèle à la broche de la machine.
- Axe X revient au plus grand déplacement horizontal.
- Axe Y forme le trièdre direct avec les axes X et Z.

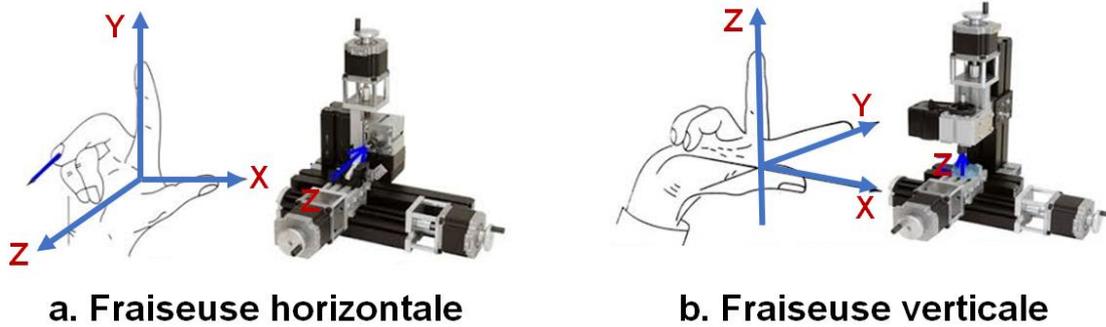


Figure I.10. Types de fraiseuses.

### I.3.1.2. Types de fraisage

Ils existent deux modes de fraisage : fraisage en bout et fraisage en roulant [13]. Pour chaque mode, l'usinage peut être :

- **En avalant** : l'outil avance dans le sens de sa rotation (Figure I.11.a).
- **En opposition** : l'outil avance dans le sens inverse de sa rotation (Figure I.11.b).

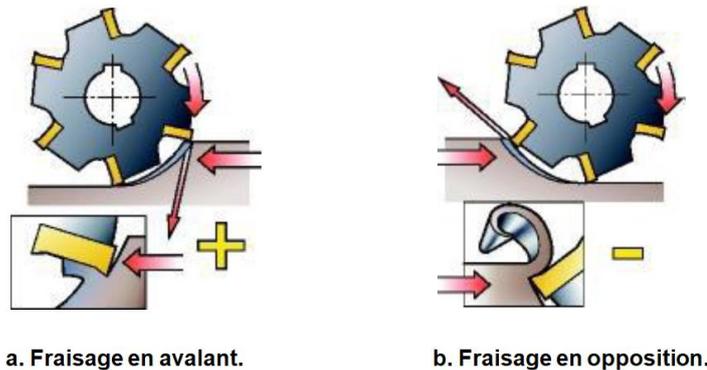


Figure I.11. Modes de fraisage.

### I.3.2. 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 processus d'usinage à une entité géométrique et la définition d'une opération d'usinage [14].

#### I.3.2.1. Processus d'usinage

Le processus d'usinage peut se traduire par les séquences d'opérations nécessaires pour l'enlèvement de la matière dans chaque plan de coupe. Pour la fabrication d'une pièce, trois opérations d'usinage sont nécessaires [15] :

- **Ebauchage** : c'est l'opération qui consiste à enlever le maximum de copeaux en minimum de temps sans s'intéresser à l'état de surface (Figure I.12.a).
- **Demi-finition** : c'est l'opération qui permet d'obtenir une finition partielle de la pièce en supprimant les épaulements laissés sur la surface après la phase d'ébauche.

- **Finition** : c'est l'opération qui consiste à finir la pièce aux côtés prescrites (Figure I.12.b).

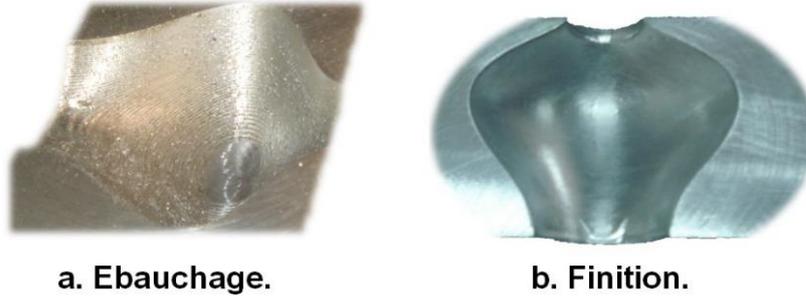


Figure I.12. Opérations d'usinage.

### I.3.2.2. Outils d'usinage

#### I.3.2.2.1. Définition

La fraise est l'outil de fraisage, elle comporte plusieurs arêtes de coupe dont chacune enlève une certaine quantité de métal sous forme de copeaux. Les principales formes de fraises utilisées dans l'usinage des surfaces gauches sur des fraiseuses multiaxes sont : cylindrique (Figure I.13.a), hémisphérique (Figure I.13.b) et torique (Figure I.13.c) [12, 16].

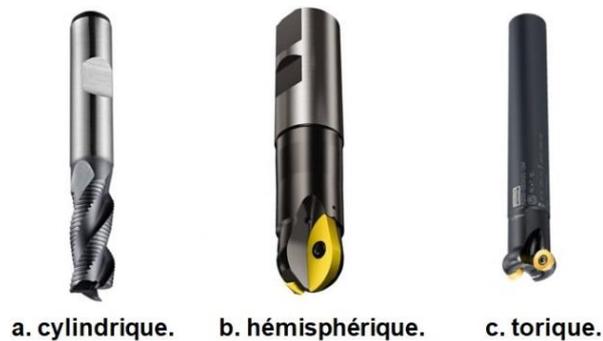


Figure I.13. Formes de fraise.

#### I.3.2.2.2. Positionnement de l'outil

Le positionnement de l'outil sur une surface gauche se fait par le calcul de la trajectoire d'usinage. Il est nécessaire de connaître à chaque instant la position de l'arête de coupe dans le repère de la pièce et de calculer l'intersection de cette courbe avec l'ensemble des plans orthogonaux au déplacement, en recevant des informations sur la forme géométrique de la surface, la géométrie de l'outil et ses dimensions ainsi que la stratégie d'usinage qui définit la direction d'usinage et les tolérances d'usinage [17].

#### I.3.2.2. Problèmes d'usinage

Lors du calcul de la trajectoire d'usinage pour le positionnement d'outil, des problèmes d'interférences et de collisions peuvent se produire.

- **Interférences actives :**

- **Interférences locales :** elles se génèrent quand la partie active de l'outil enlève la matière excessivement sur la surface à usiner.
- **Interférences vers l'arrière :** elles sont des pénétrations de l'arrière de l'outil dans la surface à usiner.

- **Collision :**

Les interférences globales se produisent quand le corps de l'outil entre en collision avec la surface à usiner ou avec le système de fixation de la pièce.

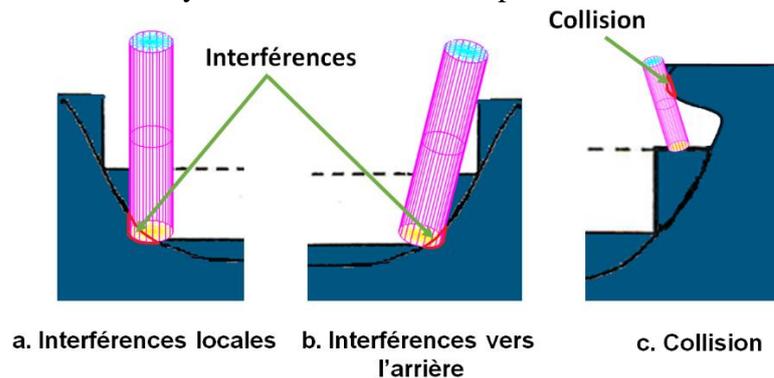


Figure I.14. Problèmes d'usinage.

### I.3.3. Stratégies d'ébauchage [12, 17]

L'opération d'ébauchage est la première étape du processus d'usinage. Elle peut être réalisée par un outil cylindrique, un outil hémisphérique ou un outil torique en fonction de la géométrie de la pièce, de la profondeur et de l'étendue de la forme à usiner. L'outil le plus utilisé est l'outil cylindrique (Figure I.15). Dans ce cas, l'ébauchage est une série d'évidement de poches dans les différents plans de coupe (plan d'usinage) où la profondeur de passe "d" est fixée par l'opérateur. Cette opération laisse à la fin des épaulements.

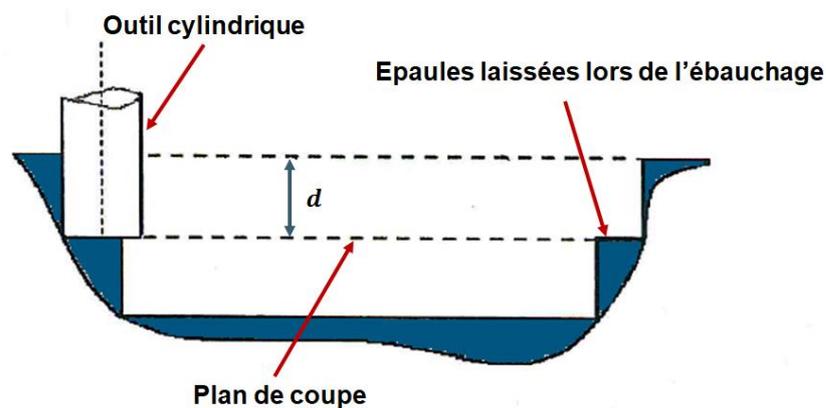


Figure I.15. Ebauchage avec un outil cylindrique.

Le trajet d'ébauchage peut être généré par plusieurs stratégies. Les deux principales stratégies sont « Plans Parallèles » et « Contours Décalés ».

### I.3.3.1. Stratégie des « Contours Décalés »

L'usinage de la surface avec cette stratégie est composé de deux étapes (Figure I.16) :

- Calcul des contours d'intersection entre des plans de coupe et la surface à usiner.
- Calcul des nouveaux contours décalés à partir des contours décalés créés précédemment d'une certaine distance «  $\omega$  » fixée par l'utilisateur qui doit être inférieur au rayon de l'outil.

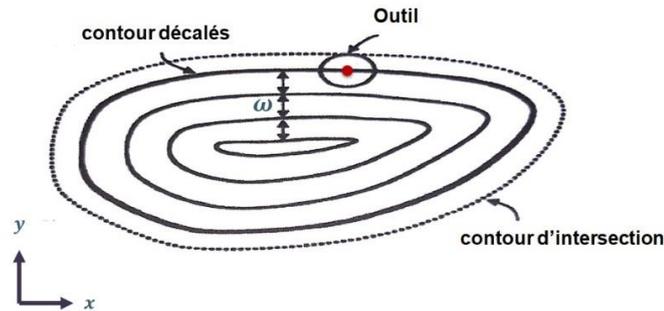


Figure I.16. Stratégie des « Contours Décalés ».

- **Avantages :**

- Usinage continu.
- Un bon comportement dynamique de l'outil.
- Réduction du nombre d'engagements et de dégagements de l'outil.

- **Inconvénient :**

- Possibilité d'existence de zones non usinées si la distance de décalage est plus grande que le rayon de l'outil (Figure I.17).

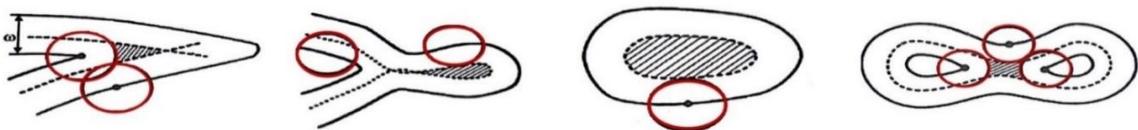


Figure I.17. Zones non usinées avec la stratégie « Contours Décalés ».

### I.4. Etude de la Stratégie d'ébauchage « Plans Parallèles »

L'opération d'ébauchage d'une pièce avec des poches complexes sur des fraiseuses numériques à 03-axes nécessite de générer la trajectoire d'usinage à partir de son modèle géométrique obtenu dans la phase de conception, pour enlever le maximum de la matière et laisser une faible surépaisseur pour l'opération de finition. Ce trajet peut être traduit par un ensemble de points issus de l'intersection des plans de coupe avec la surface à usiner. Pour cela, la stratégie des plans parallèles est utilisée.

### I.4.1. Principe de la stratégie « Plans Parallèles »

La stratégie « Plans Parallèles » permet de déterminer les segments d'intersection. Elle est composée de trois étapes (Figure I.18) [12, 17] :

- **1<sup>ère</sup> étape :** consiste à délimiter les régions concernées par l'enlèvement de matière en calculant les contours d'intersection entre les plans horizontaux (plan de coupe) et la surface à usiner.
- **2<sup>ème</sup> étape :** décalage des contours avec une distance offset égale à la somme de la surépaisseur d'usinage et du rayon d'outil.
- **3<sup>ème</sup> étape :** calcul des intersections entre ces contours et des plans parallèles et verticaux ayant une certaine orientation par rapport au plan XZ. La distance entre ces plans égale au pas transversal.

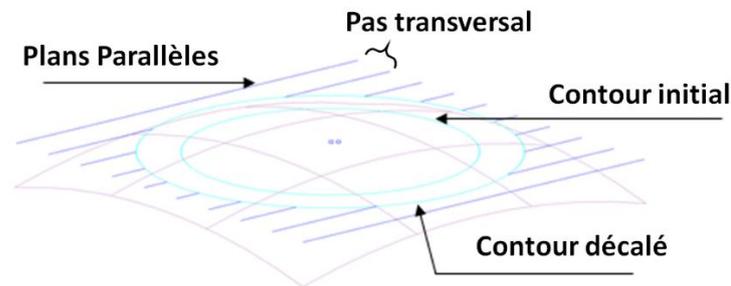


Figure I.18. Stratégie des « Plans Parallèles ».

Après le calcul des segments d'intersection et la génération des contours, le trajet d'usinage peut être généré par deux modes de balayage d'outil :

- **« One-Way » :** l'usinage de la surface pour ce mode est réalisé avec le trajet d'outil sur un ensemble de droites parallèles. Ce dernier suit une droite jusqu'à sa fin, puis il quitte la surface et passe au début de la droite suivante pour continuer l'usinage. Ce processus sera répété jusqu'à l'usinage de la surface entière (Figure I.19.a). Ce mode de balayage donne des flexions de l'outil dans la même direction. Généralement, il est utilisé pour l'usinage des matériaux durs.
- **« Zig-Zag » :** l'usinage de la surface pour ce mode est réalisé avec le trajet d'outil sur un ensemble de droites parallèles dans les deux sens (aller et retour). C'est-à-dire que l'outil suit une droite jusqu'à sa fin, et sans quitter la surface, il passe à la fin de la droite suivante pour continuer l'usinage (Figure I.19.b). Ce mode de balayage donne des flexions de l'outil dans des directions alternées. Pratiquement, il est utilisé pour l'usinage des matériaux tendres.

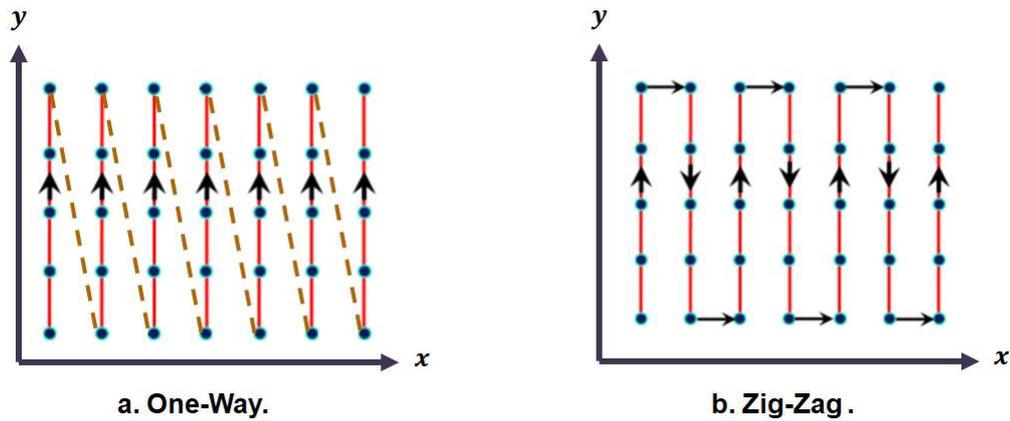


Figure I.19. Différents modes de balayage.

## Conclusion

Dans ce chapitre est donnée une vue générale sur les différents processus suivis dans l'utilisation de la « CAO » et de la « FAO ». Pour transférer les données entre ces deux derniers, l'utilisation du format STL comme format d'échange de données a été choisi. Par la suite, des généralités sur le processus d'usinage sont présentées (outils utilisés, leurs positionnements, problèmes d'usinage). Une étude détaillée est menée sur l'opération d'ébauchage.

Le prochain chapitre fera l'objet de la présentation de la solution proposée et de la conception de notre application.



# Chapitre II

## Conception de l'application

## Introduction

La résolution d'un problème passe par une succession d'étapes pour arriver à une solution convenable.

L'objectif de ce travail, est la conception d'une application logicielle permettant la génération des trajets d'ébauchage des pièces de formes complexes par la stratégie « Plans Parallèles ». Dans ce qui suit, les différentes étapes de la solution proposée et la conception de l'application seront présentées.

### II.1. Problématique

Les pièces de formes complexes sont utilisées dans la conception et la réalisation des pièces mécaniques, ...etc. L'ébauchage de ces pièces, définies par le format STL, nécessite la détermination des outils adéquats évitant les problèmes d'usinage et la génération de leurs trajectoires. Pour atteindre cet objectif, plusieurs problèmes sont rencontrés :

- La géométrie complexe de ces pièces rend cette tâche très difficile et coûteuse en temps à cause de la taille volumineuse du fichier STL.
- La délimitation des parties à usiner.
- L'évitement du choc de l'outil et le lissage de la surface.
- Risque d'interférences et de collisions.
- Le nombre des outils dans le magasin d'outils diffère d'une machine à une autre.
- L'évidement de poches sans laisser des zones non usinées.
- Création des segments définissant la trajectoire.
- Le choix du mode de balayage en fonction des matériaux (durs ou tendres).
- L'élimination des résidus de la matière.

### II.2. Solution proposée

Pour résoudre les différents problèmes énumérés, la logique de la solution proposée est donnée par la Figure II.1.

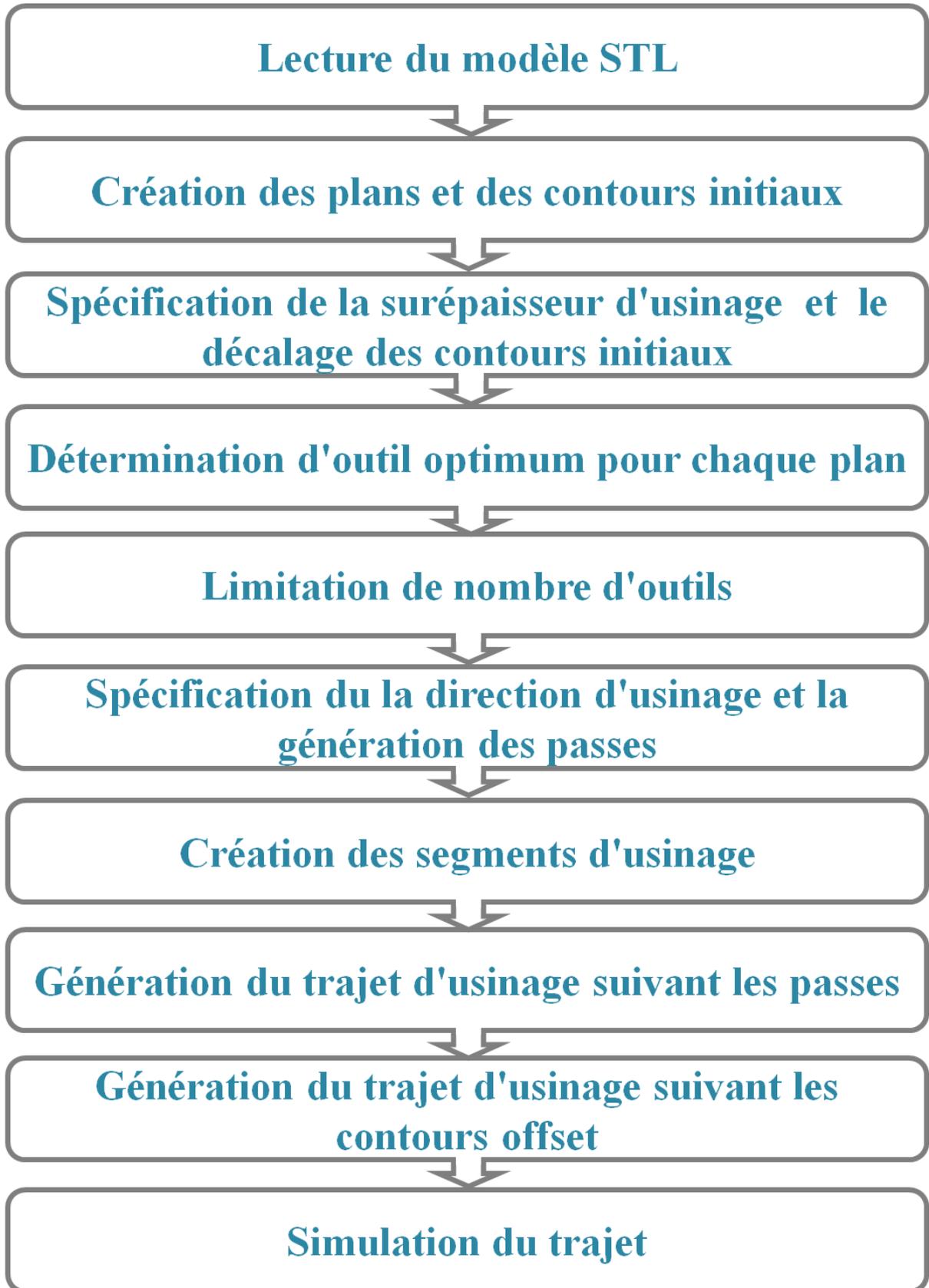


Figure II.1. Organigramme de la démarche proposée.

## II.2.1. Lecture du modèle STL

### II.2.1.1. Vérification du format STL et calcul du brut

Pour récupérer les paramètres de la pièce à usiner, il faut d'abord analyser le format du fichier STL qui décrit cette pièce. En même temps, lire les informations des facettes et les stocker dans un fichier texte, en calculant les limites du brut du modèle qui sont définies par les coordonnées  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Z_{min}$ ,  $Z_{max}$  et les dimensions (Longueur, Largeur et Hauteur) (Figure II.2).

$$\left\{ \begin{array}{l} \text{Longueur} = X_{max} - X_{min} \\ \text{Largeur} = Y_{max} - Y_{min} \\ \text{Hauteur} = Z_{max} - Z_{min} \end{array} \right. \quad (1)$$

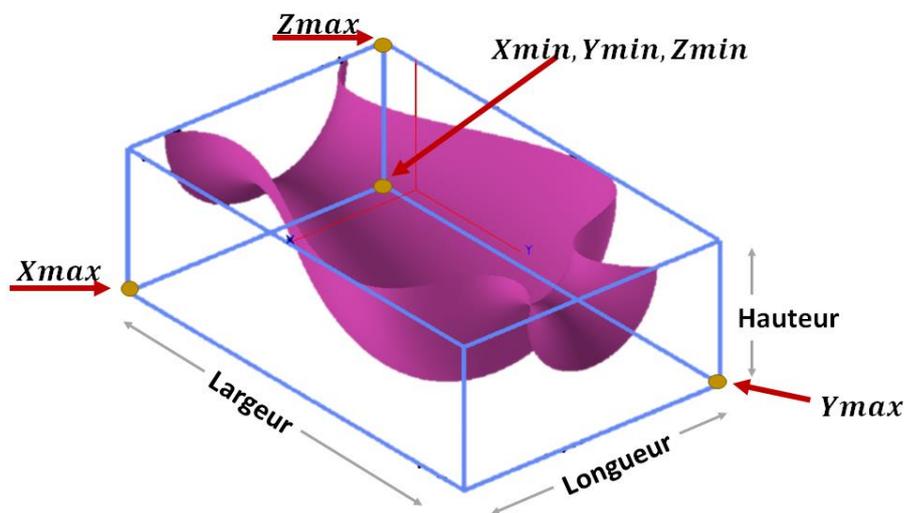


Figure II.2. Limites du brut.

### II.2.1.2. Création des cellules

Afin d'accélérer la localisation des sommets et des triangles, le brut sera subdivisé en blocs parallélépipédiques de tailles identiques (Figure II.3). Il est nécessaire d'introduire le nombre de cellules  $nbi$ ,  $n bj$  et  $n bk$  suivant les axes  $X$ ,  $Y$  et  $Z$ . Les dimensions des cellules sont calculées comme suit :

$$\left\{ \begin{array}{l} \text{Longueur\_Cellule} = \text{Longueur} / nbi \\ \text{Largeur\_Cellule} = \text{Largeur} / n bj \\ \text{Hauteur\_Cellule} = \text{Hauteur} / n bk \end{array} \right. \quad (2)$$

Les limites de chaque cellule sont calculées comme suit :

$$\left\{ \begin{array}{l}
 \text{Pour } i=0 \text{ à } nbi : X_{min} = X_{min\_Brut} + i * \text{Longueur\_Cellule} ; \\
 \quad X_{max} = X_{min} + \text{Longueur\_Cellule} ; \\
 \text{Pour } j=0 \text{ à } nbi : Y_{min} = Y_{min\_Brut} + j * \text{Largeur\_Cellule} ; \\
 \quad Y_{max} = Y_{min} + \text{Longueur\_Cellule} ; \\
 \text{Pour } k=0 \text{ à } nbi : Z_{min} = Z_{min\_Brut} + k * \text{Hauteur\_Cellule} ; \\
 \quad Z_{max} = Z_{min} + \text{Hauteur\_Cellule} ;
 \end{array} \right. \quad (3)$$

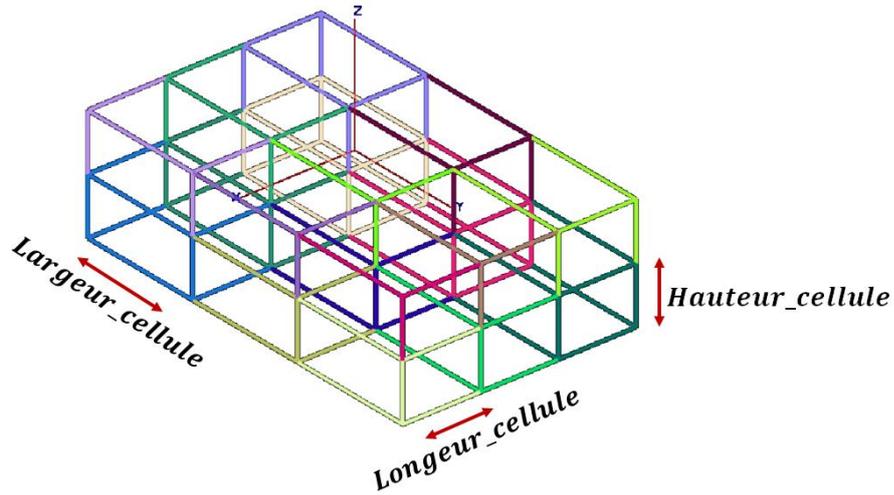


Figure II.3. Création des cellules.

### II.2.1.3. Récupération des paramètres

Les paramètres du modèle sont structurés sous forme de listes (liste des sommets et liste des triangles). Les sommets et les triangles sont affectés aux cellules créées (Figure II.4). La cellule d'un sommet est obtenue par :

$$\left\{ \begin{array}{l}
 i = (X_{sommets} - X_{min\_Brut}) / \text{Longueur\_Cellule}; \\
 j = (Y_{sommets} - Y_{min\_Brut}) / \text{Largeur\_Cellule}; \\
 k = (Z_{sommets} - Z_{min\_Brut}) / \text{Hauteur\_Cellule};
 \end{array} \right. \quad (4)$$

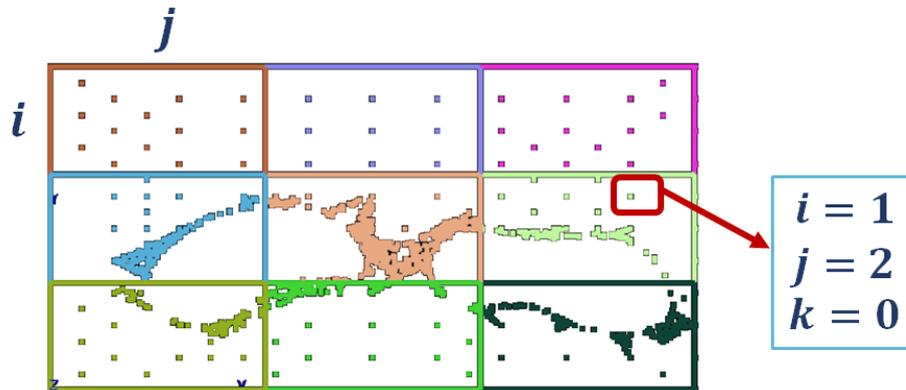


Figure II.4. Sommets des cellules.

Les cellules contenant un triangle sont obtenues à partir des indices des cellules de ses sommets (Figure II.5).

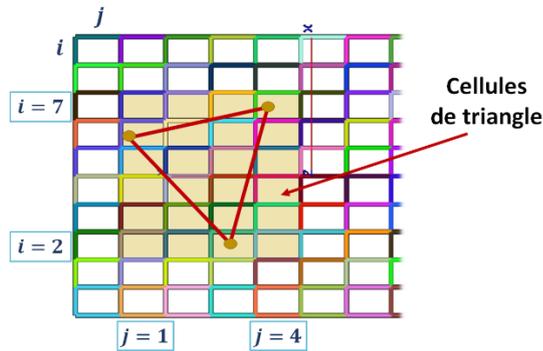


Figure II.5. Affection d'un triangle aux cellules.

## II.2.2. Création des plans et des contours initiaux

### II.2.2.1. Création des plans de coupe

La génération des plans de coupe nécessite la spécification de la profondeur de passe  $d$  (Figure II.6). Par la suite, les triangles et les cellules se chevauchant avec chaque plan de coupe sont identifiés.

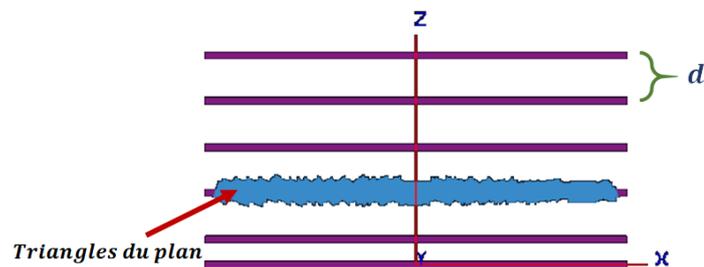


Figure II.6. Plans horizontaux.

### II.2.2.2. Calcul des segments d'intersection

Pour créer les contours initiaux, il faut d'abord calculer l'intersection entre les triangles et les plans de coupe.

- **Intersections entre un plan et un triangle :**

Soit  $ABC$  un triangle de sommets  $A(x_a, y_a, z_a)$ ,  $B(x_b, y_b, z_b)$  et  $C(x_c, y_c, z_c)$  et un plan horizontal  $(\pi)$ . Pour qu'un plan  $(\pi)$  intersecte un triangle, il faut et il suffit qu'il intersecte au moins un segment du triangle. Pour que l'intersection entre un plan et un triangle soit un ensemble vide, il faut qu'il ait deux segments de triangle tels que l'intersection entre chacun d'eux et le plan soit un ensemble vide.

Pour chaque triangle, cinq cas d'intersections peuvent être rencontrés :

**1<sup>er</sup> cas :** pas d'intersection . Le plan ne passe pas par le triangle (Figure II.7.a).

**2<sup>ème</sup> cas :** les trois segments sont dans le plan . C'est-à-dire que :

$$z_a = z_b = z_c = z \quad (5)$$

Dans ce cas, les segments sont ignorés (Figure II.7.b).

**3<sup>ème</sup> cas :** un sommet est dans le plan. Comme ce cas ne génère pas un segment , il sera ignoré (Figure II.7.c).

**4<sup>ème</sup> cas :** une arête est dans le plan. Dans ce cas l'un des segments du triangle est dans le plan (Figure II.7.d).

$$\begin{cases} [AB] \cap (\pi) = [AB] \\ \text{ou } [AC] \cap (\pi) = [AC] \\ \text{ou } [BC] \cap (\pi) = [BC] \end{cases} \quad (6)$$

**5<sup>ème</sup> cas :** un segment est dans le plan. Le plan passe par deux arêtes (Figure II.7.e).

$$\begin{cases} [AB] \cap (\pi) = \emptyset \\ [BC] \cap (\pi) = \{m_0\} \\ [CA] \cap (\pi) = \{m_1\} \end{cases} \quad (7)$$

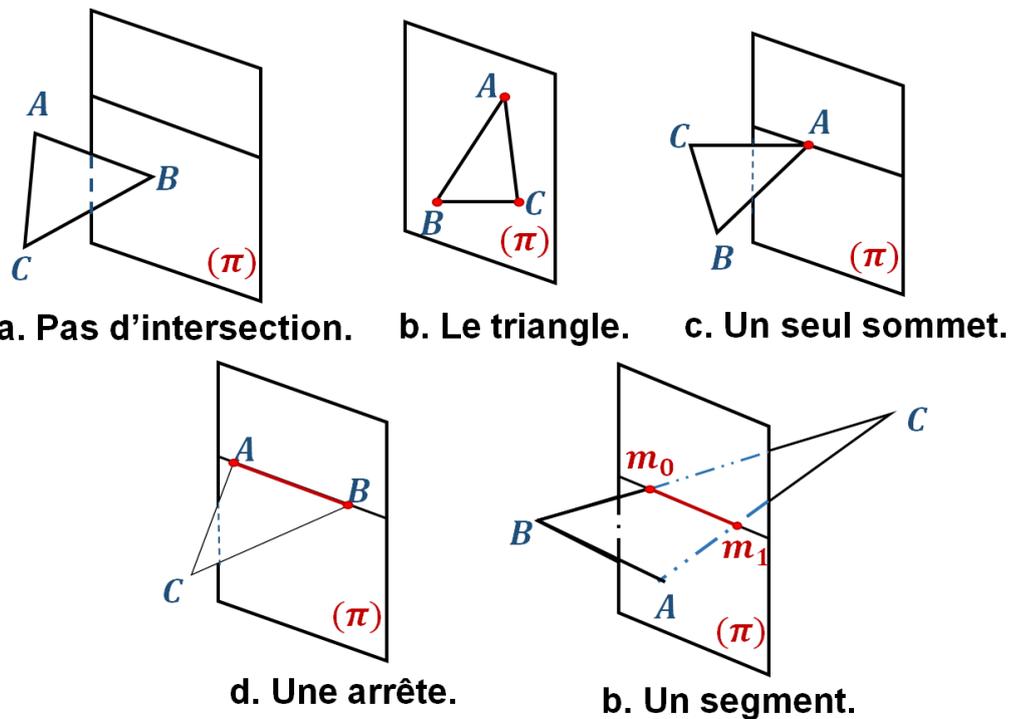


Figure II.7. Différents cas d'intersection.

• **Intersection entre un plan et un segment :**

Soit  $[AB]$  un segment dans l'espace,  $(\pi)$  un plan dont  $z = \text{constant}$  et  $M(x, y, z)$  un point dont  $M \in [AB]$  et  $M \in (\pi)$  (Figure II.8).

Nous avons :

$$\begin{pmatrix} x - x_a \\ y - y_a \\ z - z_a \end{pmatrix} = \alpha \cdot \begin{pmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \end{pmatrix} \quad (8)$$

Alors :

$$\alpha = \frac{z - z_a}{z_b - z_a} = \frac{y - y_a}{y_b - y_a} = \frac{x - x_a}{x_b - x_a} \quad (9)$$

Nous savons que  $M \in (\pi)$  et  $z = \text{constant}$ , alors à l'aide de  $z$  nous pouvons calculer  $\alpha$ . D'où :

$$\begin{cases} x = x_a + \alpha * (x_b - x_a) \\ y = y_a + \alpha * (y_b - y_a) \end{cases} \quad (10)$$

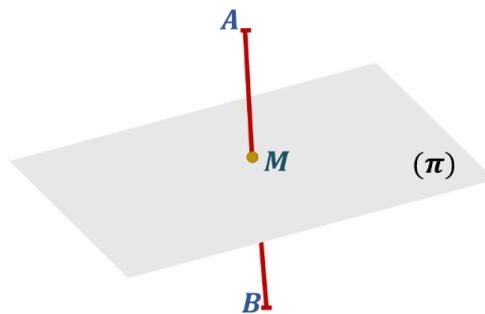


Figure II.8. Intersection entre un plan et un segment.

• **Normale d'un segment :**

Ils existent trois types des segments : horizontal (Figure II.9.a), vertical (Figure II.9.b) et incliné (Figure II.9.c).

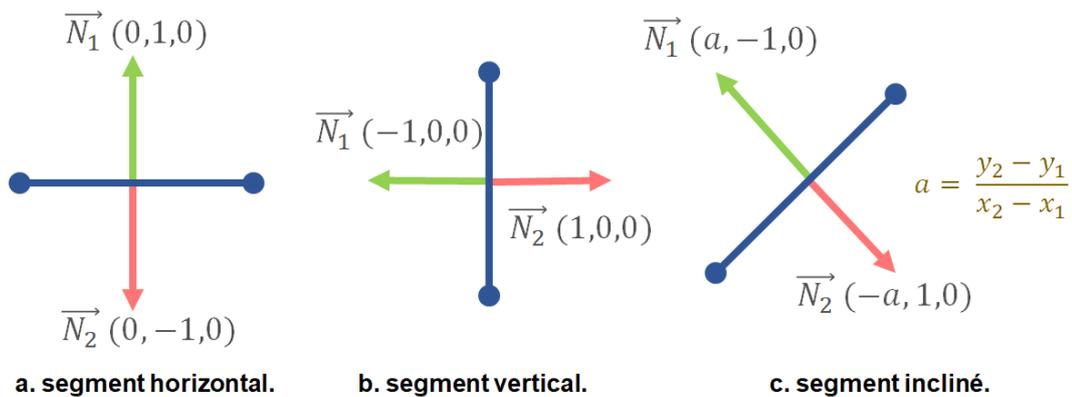


Figure II.9. Types de segments.

Le choix du bon vecteur normal est basé sur cette condition :

$$\text{Si } \vec{N}_1 \cdot \vec{N} \geq 0 \quad \text{alors } \vec{N}_1 \quad \text{Sinon } \vec{N}_2.$$

### II.2.2.3. Génération des contours initiaux

Le résultat des intersections entre les plans de coupe et les triangles est une liste des segments non ordonnés. A l'aide de ces segments, les contours de chaque plan seront construits. Ces segments sont reliés entre eux pour générer une chaîne fermée ou ouverte appelée contour (Figure II.10). Un plan peut contenir plusieurs contours. La génération des contours suit les étapes suivantes :

Tant que ( $i < \text{nombre de segments}$ ) faire :

- Ajouter le premier segment du contour :
  - Chercher le premier segment non inséré.
  - Marquer ses sommets comme début et fin du contour.
- Remplissage du contour :
  - Parcourir la liste des segments plusieurs fois en cherchant le segment qui suit le dernier segment inséré sur la liste et, l'ajouter à la fin, ou bien le segment qui vient exactement avant le premier segment inséré sur la liste et l'ajouter au début. Jusqu'à ce que le début et la fin soient égaux (le cas d'un contour fermé) ou bien jusqu'à ce qu'on puisse plus trouver un segment qui vérifie les conditions (le cas d'un contour ouvert).

Fin Tant que;

- Affecter pour chaque cellule ses segments : pour limiter les calculs d'intersection sur une partie précise du plan, l'enveloppe de chaque segment sera calculée pour affecter ces segments aux cellules qu'ils appartiennent.

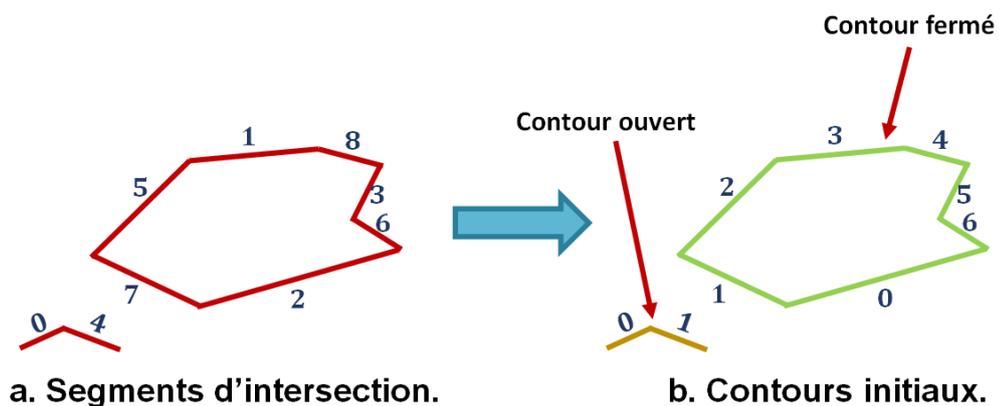


Figure II.10. Création des contours initiaux.

### II.2.3. Spécification de la surépaisseur d'usinage et décalage des contours initiaux

Les contours initiaux sont tangents avec la surface, il faut alors les décaler avec une surépaisseur spécifiée par l'utilisateur. La génération des contours décalés suit les étapes suivantes :

- Elimination des segments colinéaires : les segments ayant la même normale seront fusionnés en un seul segment (Figure II.11).
- Décalage des segments : après l'élimination des segments colinéaires, les segments seront décalés par la surépaisseur spécifiée (Figure II.11). Le décalage se fait comme suit :

Soit  $[AB]$  un segment dans le plan, le décalage de  $[AB]$  est  $[A'B']$  dont :

$$\begin{cases} x_{a'} = x_a + \text{surépaisseur} * N_x; \\ y_{a'} = y_a + \text{surépaisseur} * N_y; \end{cases} \quad (11)$$

$$\begin{cases} x_{b'} = x_b + \text{surépaisseur} * N_x; \\ y_{b'} = y_b + \text{surépaisseur} * N_y; \end{cases} \quad (12)$$

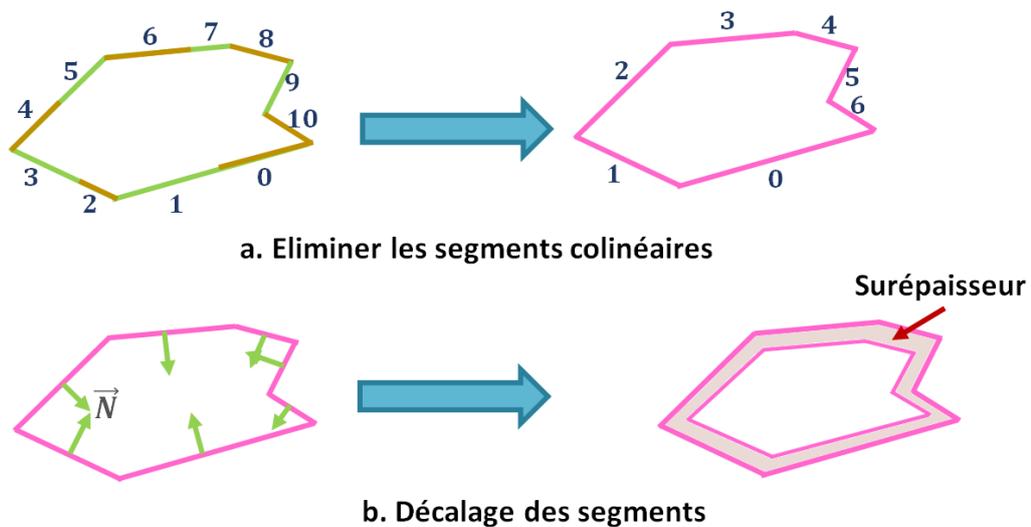


Figure II.11. Création des contours décalés.

- Calcul des points d'intersection et ajustement des contours décalés :

➤ L'intersection entre deux droites :

Soit  $[AB]$  et  $[A'B']$  deux segments dans le plan :

$$\begin{cases} \overrightarrow{AM} = \alpha \cdot \overrightarrow{AB} \\ \overrightarrow{A'M} = \beta \cdot \overrightarrow{A'B'} \end{cases} \quad (13)$$

Donc :

$$\begin{pmatrix} x - x_a \\ y - y_a \end{pmatrix} = \alpha \cdot \begin{pmatrix} x_b - x_a \\ y_b - y_a \end{pmatrix} \quad (14)$$

$$\begin{pmatrix} x - x_{a'} \\ y - y_{a'} \end{pmatrix} = \beta \cdot \begin{pmatrix} x_{b'} - x_{a'} \\ y_{b'} - y_{a'} \end{pmatrix} \quad (15)$$

D'où :

$$\begin{cases} x_a + \alpha \cdot (x_b - x_a) = x_{a'} + \beta \cdot (x_{b'} - x_{a'}) \\ y_a + \alpha \cdot (y_b - y_a) = y_{a'} + \beta \cdot (y_{b'} - y_{a'}) \end{cases} \quad (16)$$

On pose :

$$\begin{cases} A = (x_b - x_a); B = -(x_{b'} - x_{a'}); C = (x_{a'} - x_a); \\ A' = (y_b - y_a); B' = -(y_{b'} - y_{a'}); C' = (y_{a'} - y_a); \end{cases} \quad (17)$$

Alors :

$$\begin{cases} A \cdot \alpha + B \cdot \beta = C \\ A' \cdot \alpha + B' \cdot \beta = C' \end{cases} \quad (18)$$

On calcule le déterminant :

$$\begin{cases} \Delta = \begin{vmatrix} A & B \\ A' & B' \end{vmatrix} = A \cdot B' - B \cdot A'; \\ \Delta_\alpha = \begin{vmatrix} C & B \\ C' & B' \end{vmatrix}; \\ \Delta_\beta = \begin{vmatrix} C & A \\ C' & A' \end{vmatrix}; \end{cases} \quad (19)$$

Si  $\Delta \neq 0$  alors  $\alpha = \frac{\Delta_\alpha}{\Delta}; \quad \beta = \frac{\Delta_\beta}{\Delta};$

On remplace par  $\alpha$  :

$$\begin{cases} x = \alpha * A + x_a \\ y = \alpha * B + y_a \end{cases} \quad (20)$$



Le point d'intersection.

#### II.1.4. Détermination de l'outil optimum pour chaque plan

L'outil optimum d'un plan est l'outil avec le rayon le plus grand qui ne produit pas une interférence avec la surface.

Chaque outil est spécifié par son rayon  $R$ , son centre d'outil  $O(x, y, z)$ , sa vitesse, sa forme géométrique (cylindrique) et son mode (lent ou rapide) (Figure II.12.a). Le choix de l'outil commence par le plus grand outil dans la base de données des outils. Si l'outil produit une interférence on va tester avec un outil plus petit jusqu'à trouver l'outil correspondant. Pour déterminer l'outil optimum, il faut positionner l'outil sur les segments des contours décalés. Pour chaque segment, l'outil sera positionné sur les deux extrémités, puis vérifier l'existence des interférences. La détermination des zones d'interférence consiste à identifier pour chaque point de contact, les points des cellules qui se chevauchent avec l'enveloppe du déplacement de l'outil (Figure II.12.b). Ces points sont l'ensemble des points qui forment les contours initiaux en rajoutant des points intermédiaires. Cette façon permet d'éviter le test avec l'ensemble des points des contours initiaux entiers.

La vérification consiste à examiner l'appartenance d'un point du contour dans le disque d'outils sur les deux extrémités du segment ou dans son enveloppe (Figure II.12.c).

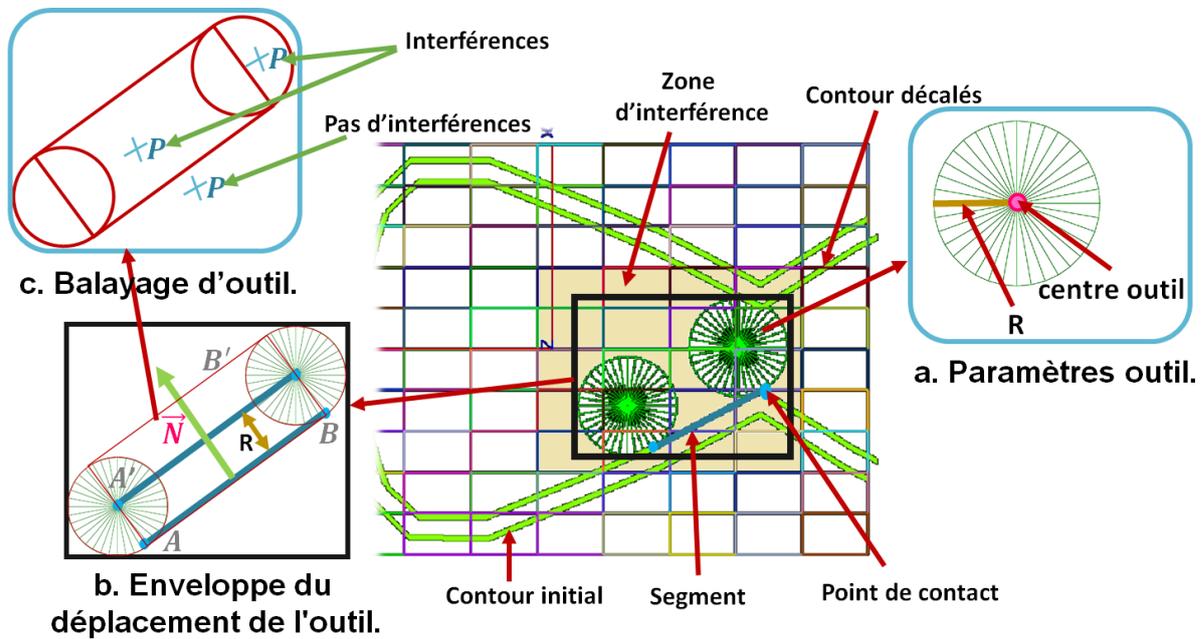


Figure II.12. Détermination de la zone d'interférence.

- **Coordonnées deux points centre outil :**

Soit  $[AB]$  un segment dans le plan, le positionnement de l'outil sera  $[A'B']$  (Figure II.12.b) :

$$\begin{cases} A' (x_a + R * N_x , y_a + R * N_y); \\ B' (x_b + R * N_x , y_b + R * N_y); \end{cases} \quad (21)$$

- **Enveloppe du déplacement de l'outil :**

Le positionnement de l'outil sur les deux extrémités du segment forme un rectangle, c'est l'enveloppe de déplacement de l'outil sur le segment (Figure II.12.b).

Soit  $x_{min}, y_{min}, x_{max}, y_{max}$  l'enveloppe du segment  $[A'B']$ , les coordonnées du rectangle sont :

$$\begin{cases} (x_{min} - R, y_{min} - R) \\ (x_{max} + R, y_{min} - R) \\ (x_{max} + R, y_{max} + R) \\ (x_{min} - R, y_{max} + R) \end{cases} \quad (22)$$

- **Ajouter des points intermédiaires dans un segment :**

Soit  $[AB]$  un segment dans le plan, et l'ensemble des points  $M(x, y, z) \in [AB]$ .

On a :

$$\overrightarrow{OM} = (1 - \alpha) \cdot \overrightarrow{OA} + \alpha \cdot \overrightarrow{OB} \quad (23)$$

Avec :

$$\alpha = \frac{\text{Nombre Segments}}{\| \overrightarrow{AB} \|} \quad (24)$$

Calcul du Nombre Segments :

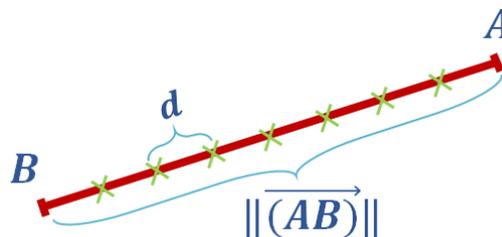
$$\text{Nombre Segments} = \frac{\| \overrightarrow{AB} \|}{d}, \quad (25)$$

Si (Nombre Segments est un réel) alors Nombre Segments= Nombre Segments+1;

Donc, les points intermédiaires sont :

Pour  $i = 0$  à Nombre Segments :

$$\begin{cases} x = (1 - \alpha * i) \cdot x_a + (\alpha * i) \cdot x_b \\ y = (1 - \alpha * i) \cdot y_a + (\alpha * i) \cdot y_b \end{cases} \quad (26)$$



**Figure II.13.** Insertion de points intermédiaires dans un segment.

- **Appartenance d'un point à un cercle :**

Soit un cercle  $(C)$  du centre  $O(x, y)$  et du rayon  $R$  et  $M(x, y)$  un point sur le plan.

$$M \in (C) \Rightarrow \| \overrightarrow{OM} \| \leq R$$

$$\|\overrightarrow{OM}\| = \sqrt{(x_m - x_o)^2 + (y_m - y_o)^2} \quad (27)$$

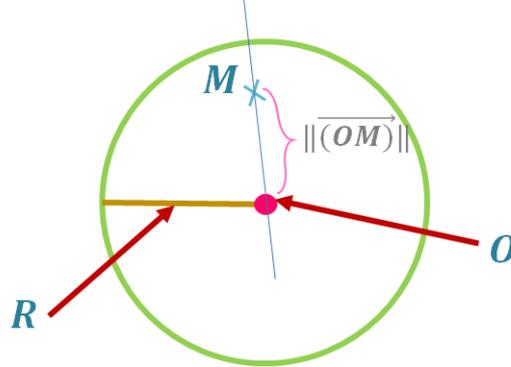


Figure II.14. Appartenance d'un point à un cercle.

• **Appartenance d'un point à un rectangle :**

Soit  $(x_a, y_a)$ ,  $B(x_b, y_b)$  et  $C(x_c, y_c)$  trois des quatre points formant un coin d'un rectangle dans un plan. On dispose donc des deux vecteurs :  $\overrightarrow{AB} = (x_b - x_a, y_b - y_a)$  et  $\overrightarrow{AC} = (x_c - x_a, y_c - y_a)$ . Ils nous permettent de définir deux vecteurs unitaires :

$$\begin{cases} \vec{u} = \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \\ \vec{v} = \frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} \end{cases} \quad (28)$$

Soit maintenant  $P(x_p, y_p)$  appartenant au plan défini par  $A, B$  et  $C$ .

On a :

$$\overrightarrow{AP} = (x_p - x_a, y_p - y_a) \quad (29)$$

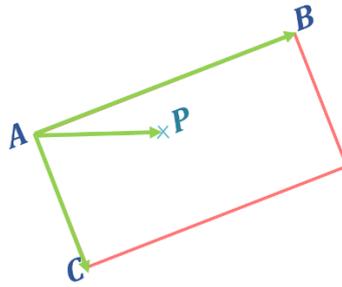
En notant  $\langle \vec{i}, \vec{j} \rangle$  le produit scalaire entre  $\vec{i}$  et  $\vec{j}$ , on peut écrire :

$$\overrightarrow{AP} = \langle \overrightarrow{AP}, \vec{u} \rangle \vec{u} + \langle \overrightarrow{AP}, \vec{v} \rangle \vec{v} \quad (30)$$

Ce qu'on peut également réécrire en :

$$\overrightarrow{AP} = \frac{\langle \overrightarrow{AP}, \overrightarrow{AB} \rangle}{\|\overrightarrow{AB}\|^2} \overrightarrow{AB} + \frac{\langle \overrightarrow{AP}, \overrightarrow{AC} \rangle}{\|\overrightarrow{AC}\|^2} \overrightarrow{AC} = \alpha \cdot \overrightarrow{AB} + \beta \cdot \overrightarrow{AC} \quad (31)$$

Le point  $P$  appartient alors au rectangle ssi  $0 \leq \alpha \leq 1$  et  $0 \leq \beta \leq 1$ .



**Figure II.15.** Appartenance d'un point à un rectangle.

### II.2.5. Limitation du nombre d'outils

Après l'affectation des outils pour chaque plan, il faut définir l'outil réel pour chaque plan en limitant le nombre des outils à utiliser pour tout le trajet en fonction du nombre d'outils disponibles dans le magasin d'outils.

Le nombre d'outils  $N$  est fixé par l'utilisateur. La sélection des outils réels suit les étapes suivantes :

- Calculer le nombre d'utilisation pour chaque outil.
- Chercher le plus petit outil optimum utilisé et le stocker.
- Choisir les  $N - 1$  premiers outils optimums ayant le nombre d'utilisation le plus grand sans compter le plus petit outil optimum.
- Trier la liste des outils réels du plus petit au plus grand.
- Affecter pour chaque plan son outil réel.
- Calculer les positions des nouveaux outils réels.

### II.1.6. Spécification de la direction d'usinage et création des passes

L'approche d'usinage utilisée est basée sur la création des passes. La distance entre ces passes est constante. Elle est égale au pas transversal qui est calculé à l'aide du pourcentage de recouvrement fixé par l'utilisateur avec la formule suivante :

$$Pas\ transversal = \frac{\text{pourcentage de recouvrement}}{100} * 2 * R \quad (32)$$



$R$  est le rayon de l'outil réel.

La direction d'usinage dépend de l'angle d'orientation (Figure II.16). Le nombre des passes dépend du pas transversal, des dimensions du brut et de l'angle d'orientation.

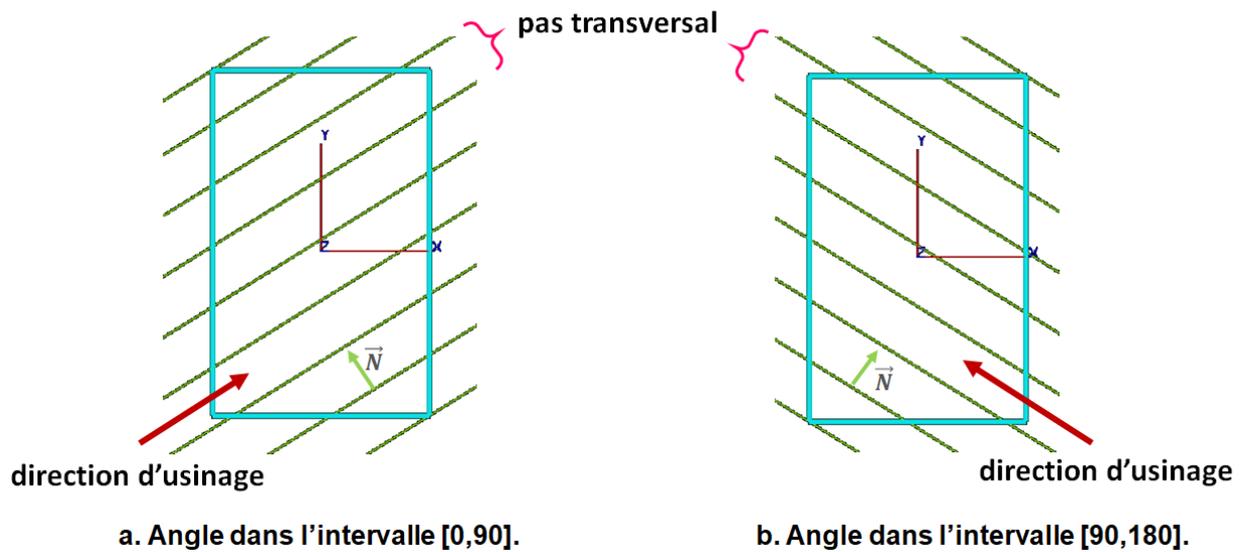


Figure II.16. Création des passes d'usinage.

## II.2.7. Création des segments d'usinage

### II.2.7.1. Calcul des intersections entre les passes et les contours offset

Les contours offset sont générés de la même façon que les contours décalés avec un décalage (Figure II.17) :

$$\begin{cases} x = x + (\text{surépaisseur} + R) * N_x \\ y = y + (\text{surépaisseur} + R) * N_y \end{cases} \quad (33)$$

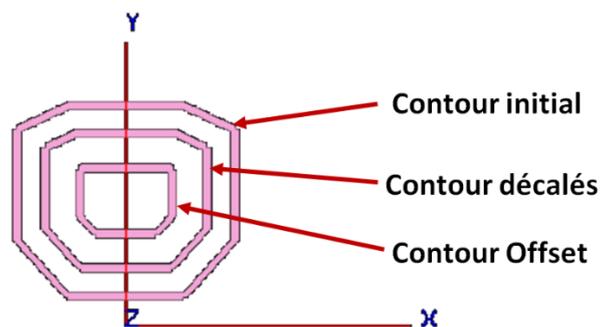


Figure II.17. Création des Contours offset.

Chaque segment du contour est affecté aux cellules qu'il appartient.

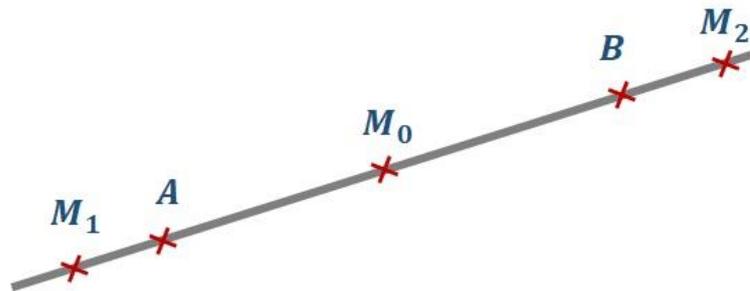
Le calcul des intersections entre les passes et les contours offset consiste à calculer le point d'intersection entre deux droites et à vérifier si le point appartient au segment. Les points d'intersection seront stockés dans une liste des sommets pour chaque droite et triés en fonction de la coordonnée  $x$  (d plus petit au plus grand) pour créer les segments de la droite.

• **Appartenance d'un point à un segment de droite :**

Soit  $[AB]$  un segment de droite dans l'espace (Figure II.18), on dit que :

$$M \in [AB] \Leftrightarrow \exists \alpha / 0 \leq \alpha \leq 1, \overrightarrow{AM} = \alpha \cdot \overrightarrow{AB}$$

$$\begin{cases} M_0 \in [AB] \text{ et } 0 \leq \alpha \leq 1 \\ M_1 \notin [AB] \text{ car } \alpha \geq 1 \\ M_2 \notin [AB] \text{ car } \alpha < 0 \end{cases} \quad (34)$$



**Figure II.18.** Position des différents points par rapport à  $AB$ .

De la relation précédente on peut déduire la relation suivante :

$$\overrightarrow{AM} = \alpha \cdot \overrightarrow{AB}, \quad 0 \leq \alpha \leq 1 \quad (35)$$

On pose :

$$A(x_A, y_A, z_A), B(x_B, y_B, z_B) \text{ et } M(x, y, z)$$

Donc :

$$\begin{pmatrix} x - x_a \\ y - y_a \\ z - z_a \end{pmatrix} = \alpha \cdot \begin{pmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \end{pmatrix} \quad (36)$$

$$\Downarrow \quad 0 \leq \alpha < 1$$

Alors :

$$\alpha = \frac{x - x_a}{x_b - x_a} = \frac{y - y_a}{y_b - y_a} = \frac{z - z_a}{z_b - z_a} \quad (37)$$

D'où, pour qu'un point appartienne à un segment de droite, il suffit que :

$$\left\{ \begin{array}{l} 0 \leq \frac{x-x_A}{x_B-x_A} \leq 1 \\ \text{ou} \quad 0 \leq \frac{y-y_A}{y_B-y_A} \leq 1 \\ \text{ou} \quad 0 \leq \frac{z-z_A}{z-z_A} \leq 1 \end{array} \right. \quad (38)$$

### II.2.7.2. Génération des segments d'usinage

Pour qu'un segment soit usiné, il faut vérifier qu'il n'y a pas de matière au-dessus (Figure II.19). Pour cela, nous procédons de la manière suivante :

- Calcul de l'intersection entre le milieu de ce segment et le modèle STL.
- Si ce point n'appartient pas à l'un des triangles du modèle, dans ce cas-là on procède à une deuxième vérification avec les contours initiaux.
- Si le segment n'intersecte pas l'un des segments des contours initiaux du plan, à ce moment-là, on fera un dernier test pour s'assurer que l'usinage de ce segment ne produit pas d'interférences.

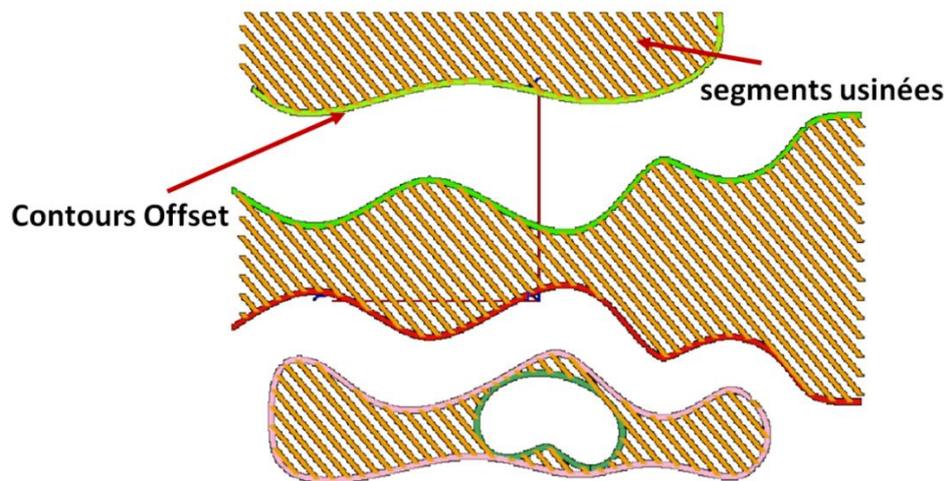


Figure II.19. Segments d'usinage.

- **Appartenance d'un point à un triangle :**

Soit  $(ABC)$  un triangle et  $M(x, y, z)$  un point dans l'espace,  $M \in (ABC) \Rightarrow$

$$Surface_{ABC} = Surface_{ABM} + Surface_{ACM} + Surface_{BCM} \quad (39)$$

Calculer la surface d'un triangle :

$$Surface_{ABC} = \sqrt{r * (r - \|\vec{AB}\|) * (r - \|\vec{AC}\|) * (r - \|\vec{BC}\|)} \quad (40)$$

$$\hookrightarrow r = \frac{\|\vec{AB}\| + \|\vec{AC}\| + \|\vec{BC}\|}{2}$$

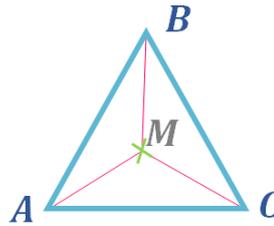


Figure II.20. Appartenance d'un point à un triangle.

### II.2.8. Génération du trajet d'usinage suivant les passes

Le trajet d'usinage représente les mouvements des outils lors de l'opération d'ébauchage. La génération du trajet suit les étapes suivantes :

- Fixer les distances de sécurité et d'engagement par l'utilisateur.
- Insertion des deux points d'engagement et de sécurité pour chaque segment usiné au début et à la fin.
- Stocker les mouvements d'outil pour chaque droite sous forme d'un point CL (point d'extrémité d'outil).
- Construire le trajet d'usinage :
  - One-Way : la construction du trajet en aller-simple consiste à regrouper les points CL de chaque droite tels qu'ils sont, de façon que l'usinage de la surface suit une droite jusqu'à sa fin, puis il passe au début de la droite suivante (Figure II.21.a).
  - Zig-Zag : la construction du trajet en aller-retour consiste à suivre une droite jusqu'à sa fin en ajoutant les points CL de cette droite, et sans quitter le plan, vérifier si le déplacement du dernier point CL de cette droite au dernier point CL de la droite suivante, ne produit pas des interférences, dans ce cas on passe directement à la fin de la droite suivante, sans passer par les points d'engagement et de dégagement. Dans le cas contraire, on les rajoute puis on passe à la fin de la droite suivante et on continue l'usinage (Figure II.21.b).

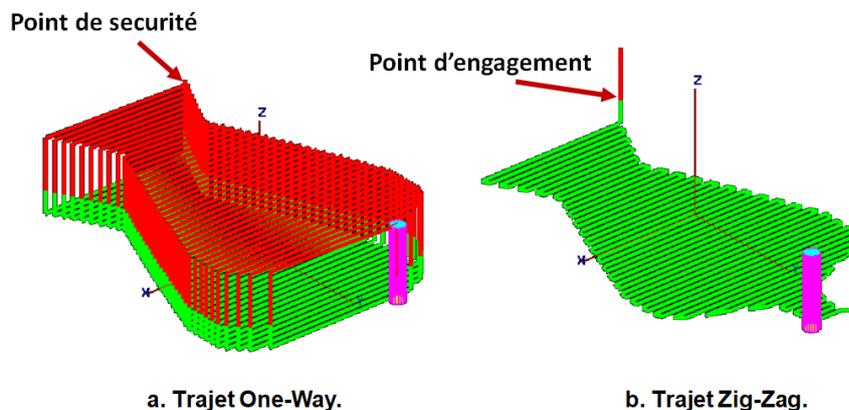


Figure II.21. Trajet d'usinage.

### II.2.9. Génération du trajet d'usinage suivant les contours offset

L'usinage des passes laisse à la fin des résidus de la matière. Pour les éliminer, on passe l'outil sur les contours d'usinage en ajoutant les points d'engagement et de dégagement au début et la fin du contour (Figure II.22).

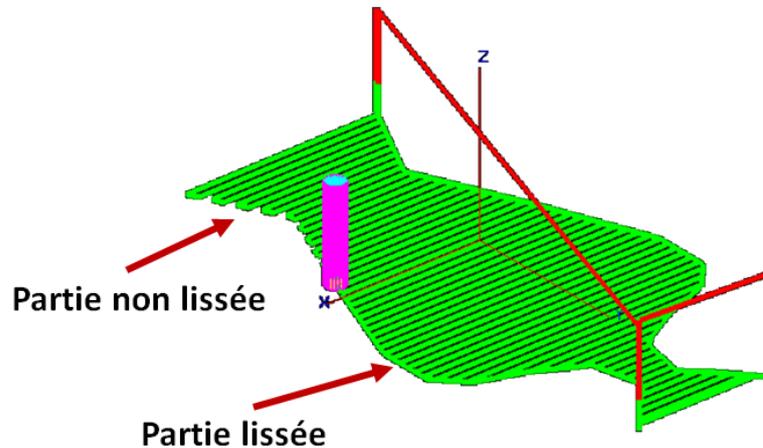


Figure II.22. Lissage des contours.

### II.2.10. Simulation du trajet

La simulation du trajet d'outils consiste à visualiser graphiquement l'outil dans toutes les positions de passage de l'outil. Le trajet sera enregistré sur un fichier.

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

UML «Unified Modeling Language» ou «Langage de modélisation unifié», est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde de génie logiciel, dans le cadre de la conception orientée objet. Il est couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes, ne se limitant pas au domaine informatique. Il permet donc de modéliser les objets et ainsi représenter l'application sous forme de diagrammes.

### II.3.1. Diagramme de cas d'utilisation

Les cas d'utilisation permettant de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Les principaux concepts de ce diagramme sont les acteurs qu'ils interagissent avec les cas d'utilisations. Ces derniers ils décrivent le comportement d'un système de point de vue d'utilisateur sous la forme des actions et des réactions.

### II.3.1.1. Diagramme de cas d'utilisation général

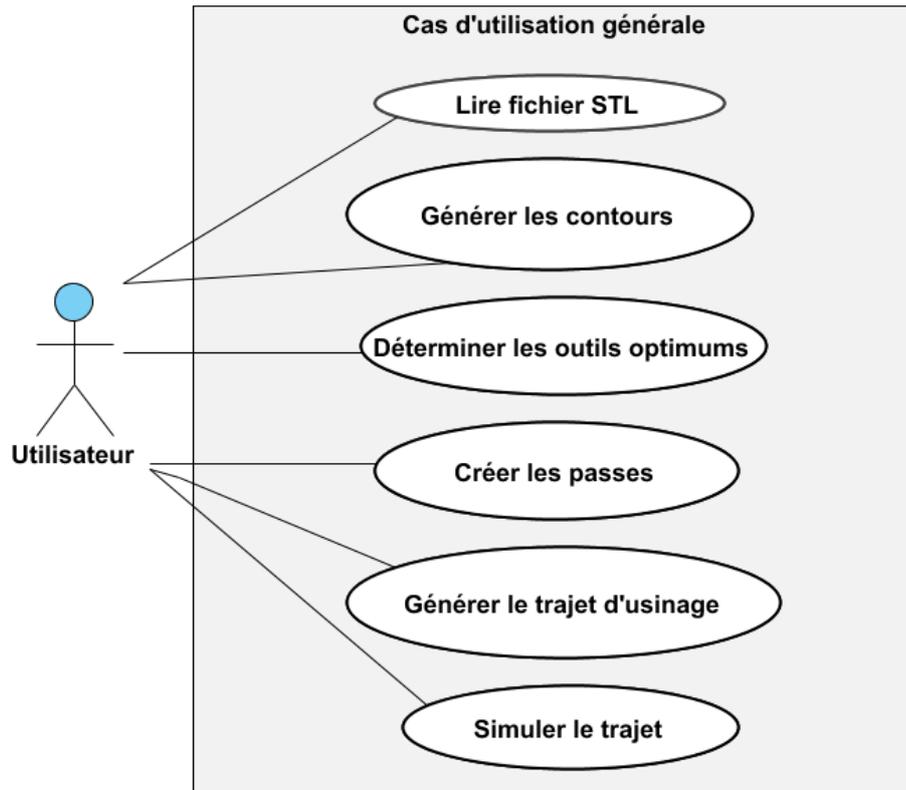


Figure II.23. Diagramme de cas d'utilisation général.

### II.3.1.2. Diagramme de cas d'utilisation « Lire Fichier STL »

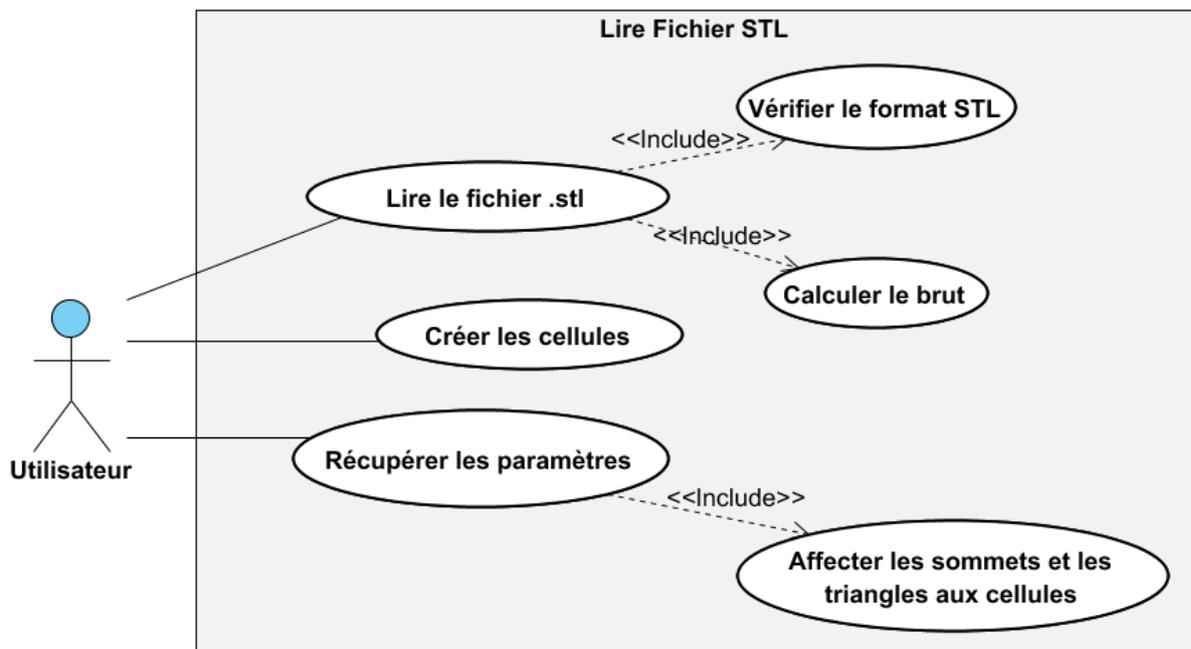


Figure II.24. Cas d'utilisations « Lire Fichier STL ».

**II.3.1.3. Diagramme de cas d'utilisation «Générer les contours»**

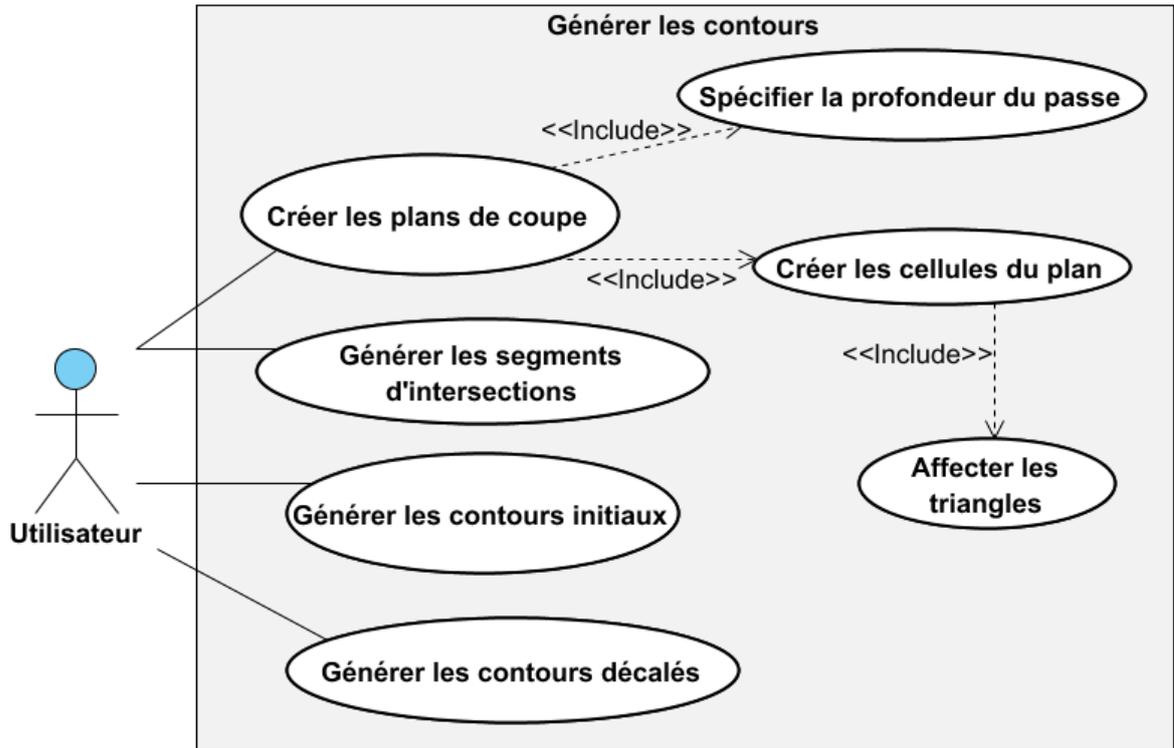


Figure II.25. Cas d'utilisations « Générer les contours ».

**II.3.1.4. Diagramme de cas d'utilisation « Déterminer les outils optimums »**

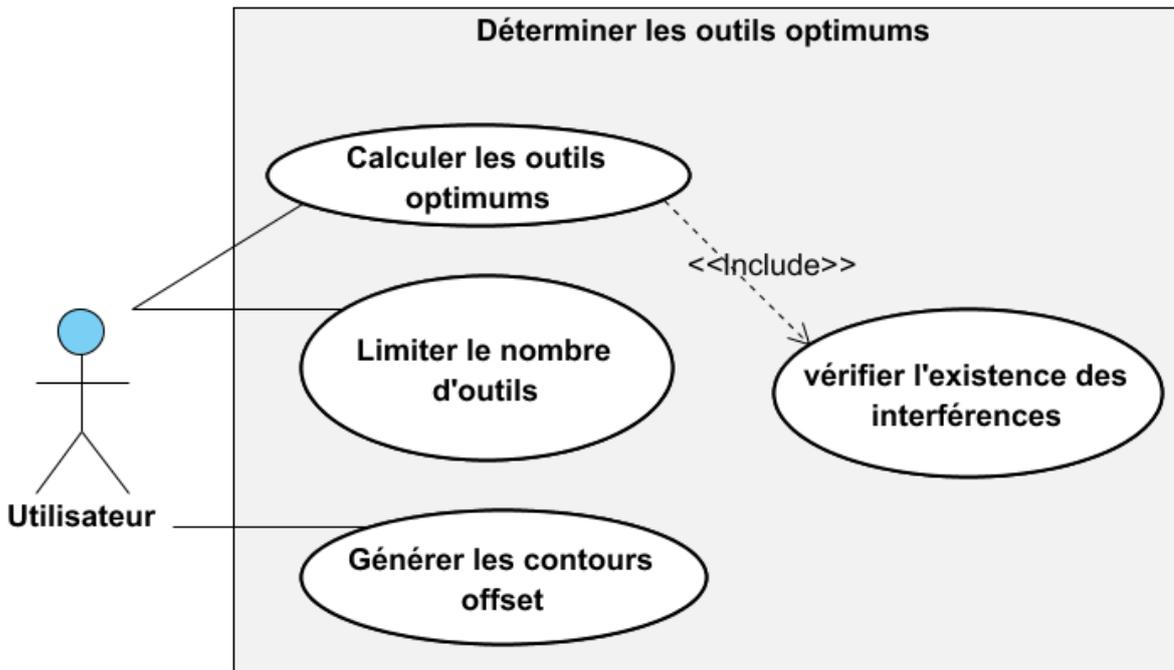


Figure II.26. Cas d'utilisations « Déterminer les outils optimums ».

### II.3.1.5. Diagramme de cas d'utilisation «Créer les passes»

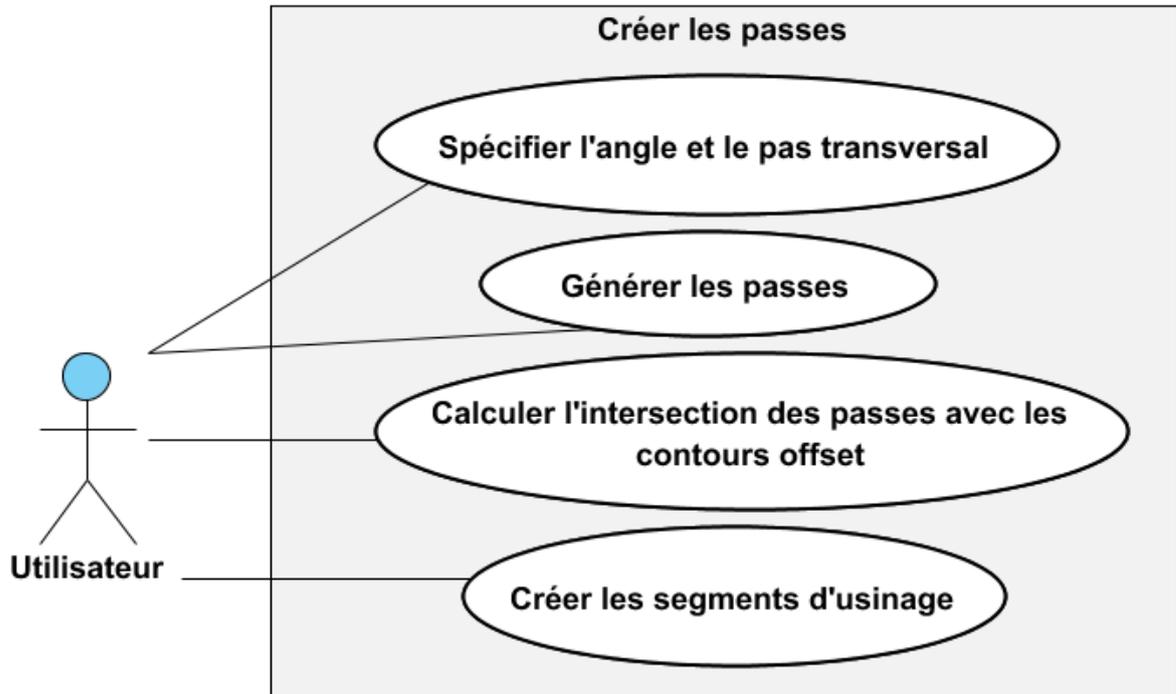


Figure II.27. Cas d'utilisations « Créer les passes ».

### II.3.1.6. Diagramme de cas d'utilisation «Générer le trajet d'usinage»

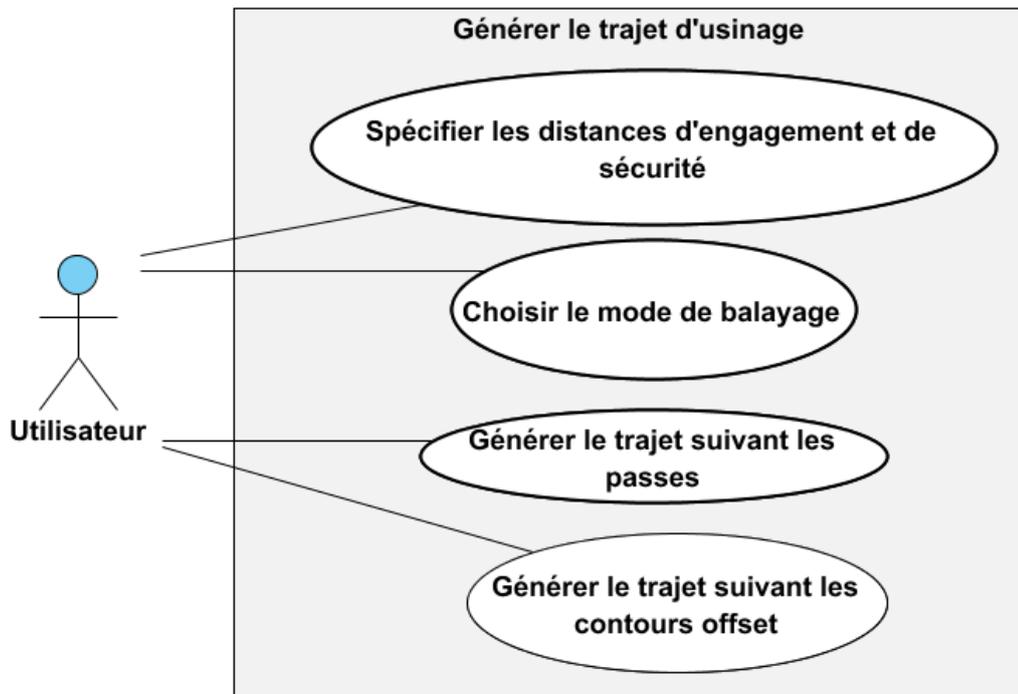


Figure II.28. Cas d'utilisations « Générer le trajet d'usinage ».

### II.3.1.7. Diagramme de cas d'utilisation « Simuler le trajet »

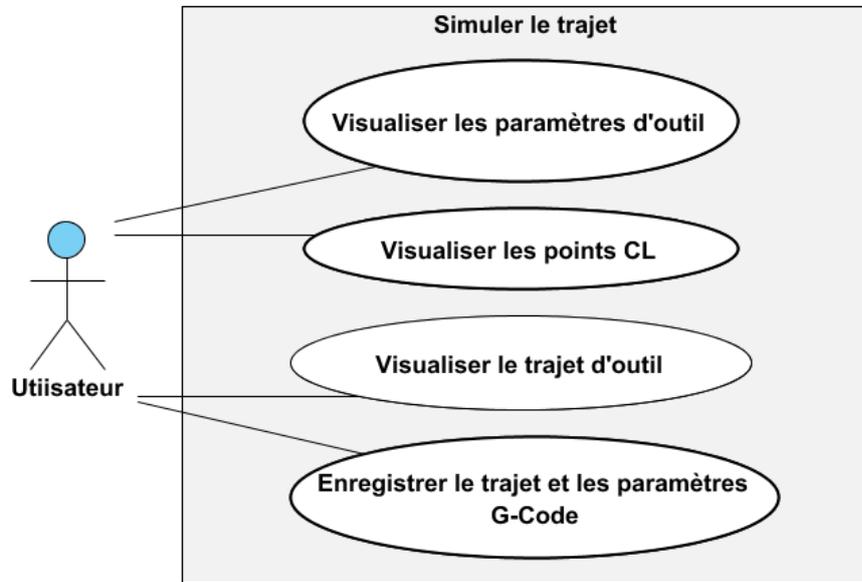


Figure II.29. Cas d'utilisations « Simuler le trajet ».

### II.3.2. Diagramme d'activité

Un diagramme d'activité permet de mettre l'accent sur les traitements. Il est donc particulièrement adapté à la modélisation du cheminement de flots de données. Il permet ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

#### II.3.2.1. Diagramme d'activité général

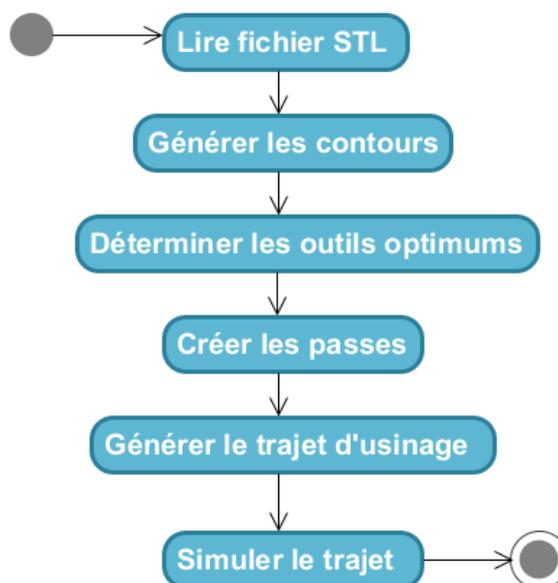


Figure II.30. Diagramme d'activité général.

**II.3.2.2. Diagramme d'activité « Lire Fichier STL »**

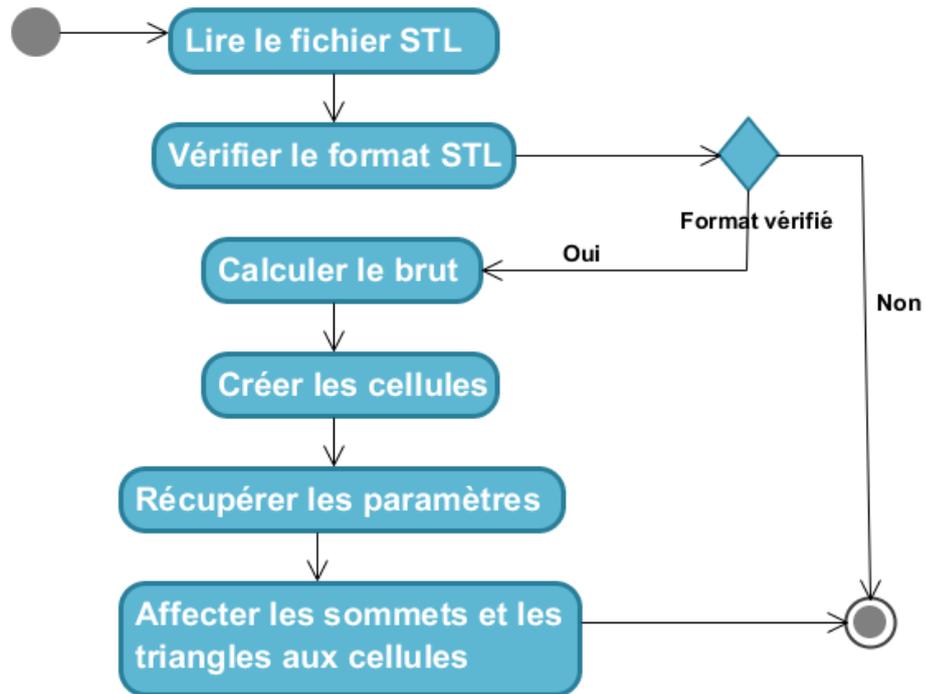


Figure II.31. Diagramme d'activité « Lire Fichier STL ».

**II.3.2.3. Diagramme d'activité « Générer les contours »**



Figure II.32. Diagramme d'activité « Générer les contours ».

II.3.2.4. Diagramme d'activité « Déterminer les outils optimums »

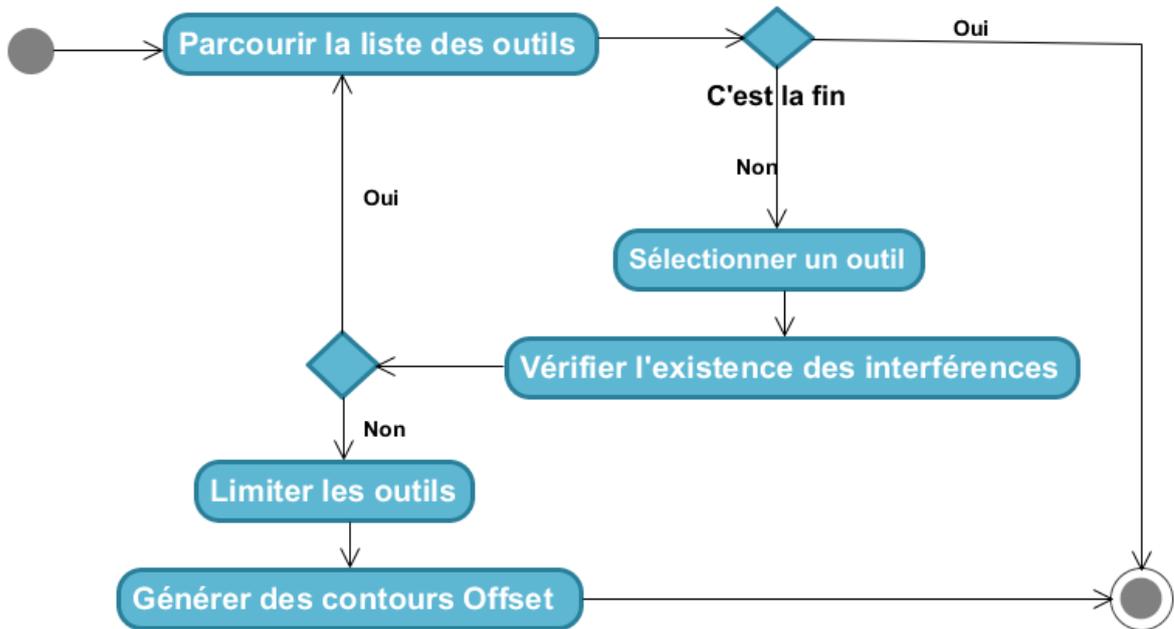


Figure II.33. Diagramme d'activité « Déterminer les outils optimums ».

II.3.2.5. Diagramme d'activité « Créer les passes »

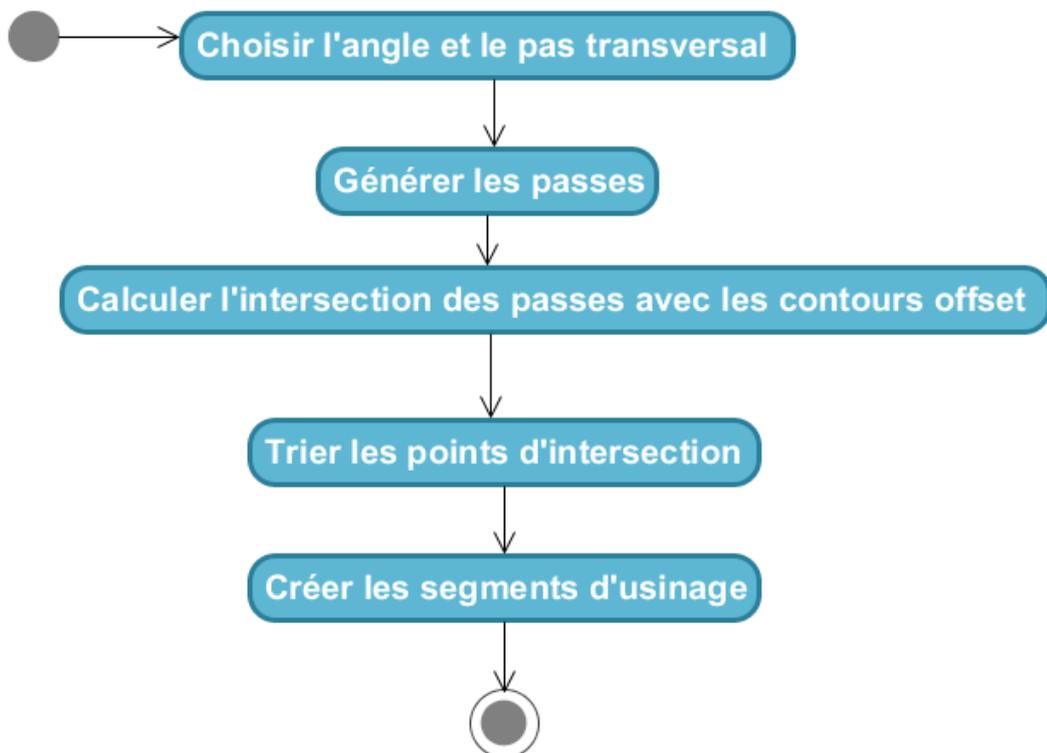


Figure II.34. Diagramme d'activité « Créer les passes ».

### II.3.2.6. Diagramme d'activité « Générer le trajet d'usinage »

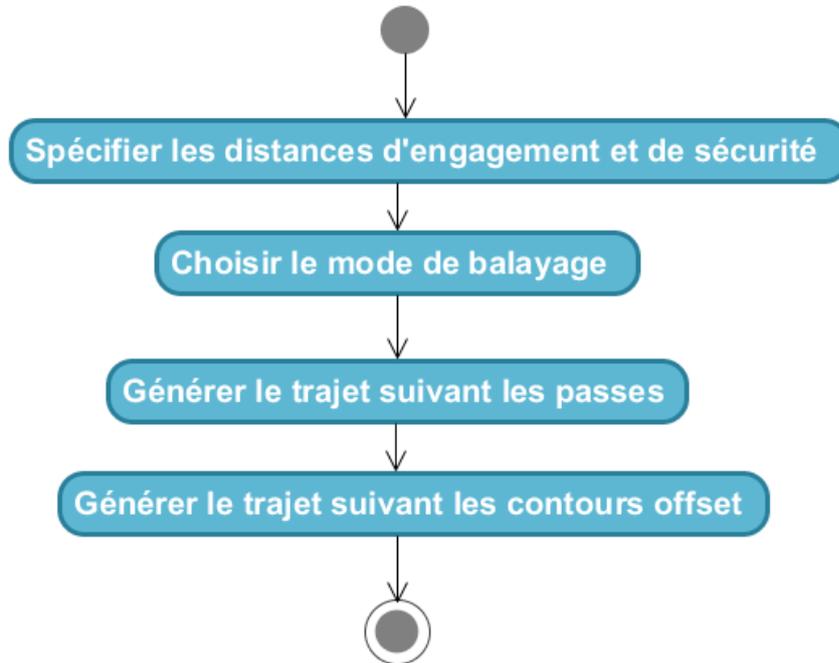


Figure II.35. Diagramme d'activité « Générer le trajet d'usinage ».

### II.3.2.6. Diagramme d'activité « Simuler le trajet »

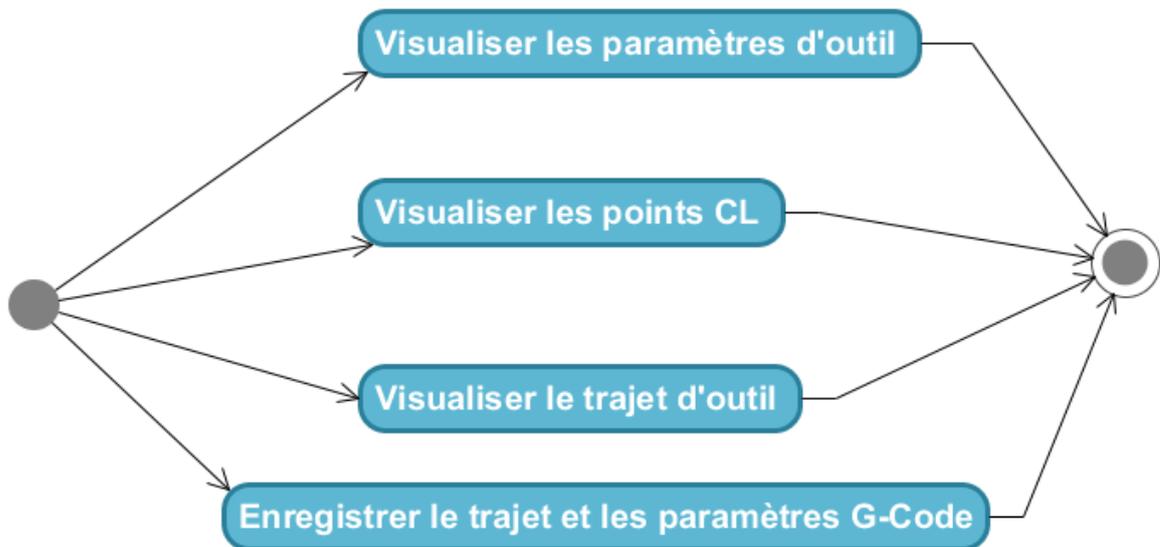


Figure II.36. Diagramme d'activité « Simuler le trajet ».

### II.3.3. Diagramme de classes

Le diagramme de classe montre la structure interne du logiciel. Il fournit une représentation 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 II.37 montre notre diagramme de classe.

Dans ce qui suit, nous présentons le diagramme de classe de notre système et nous détaillons les principales classes utilisées. Les classes implémentées dans ce travail sont :

- Classe Sommet\_Ebauchage
- Classe Normal\_Ebauchage
- Classe Triangle\_Ebauchage
- Classe Brute\_Ebauchage
- Classe Cellule\_Ebauchage
- Classe ModelSTL\_Ebauchage
- Classe Segment\_Ebauchage
- Classe Contour\_Ebauchage
- Classe CellulePlan\_Ebauchage
- Classe Outil\_Ebauchage
- Classe Droite\_Ebauchage
- Classe PlanCoup\_Ebauchage
- Classe Trajet\_Ebauchage

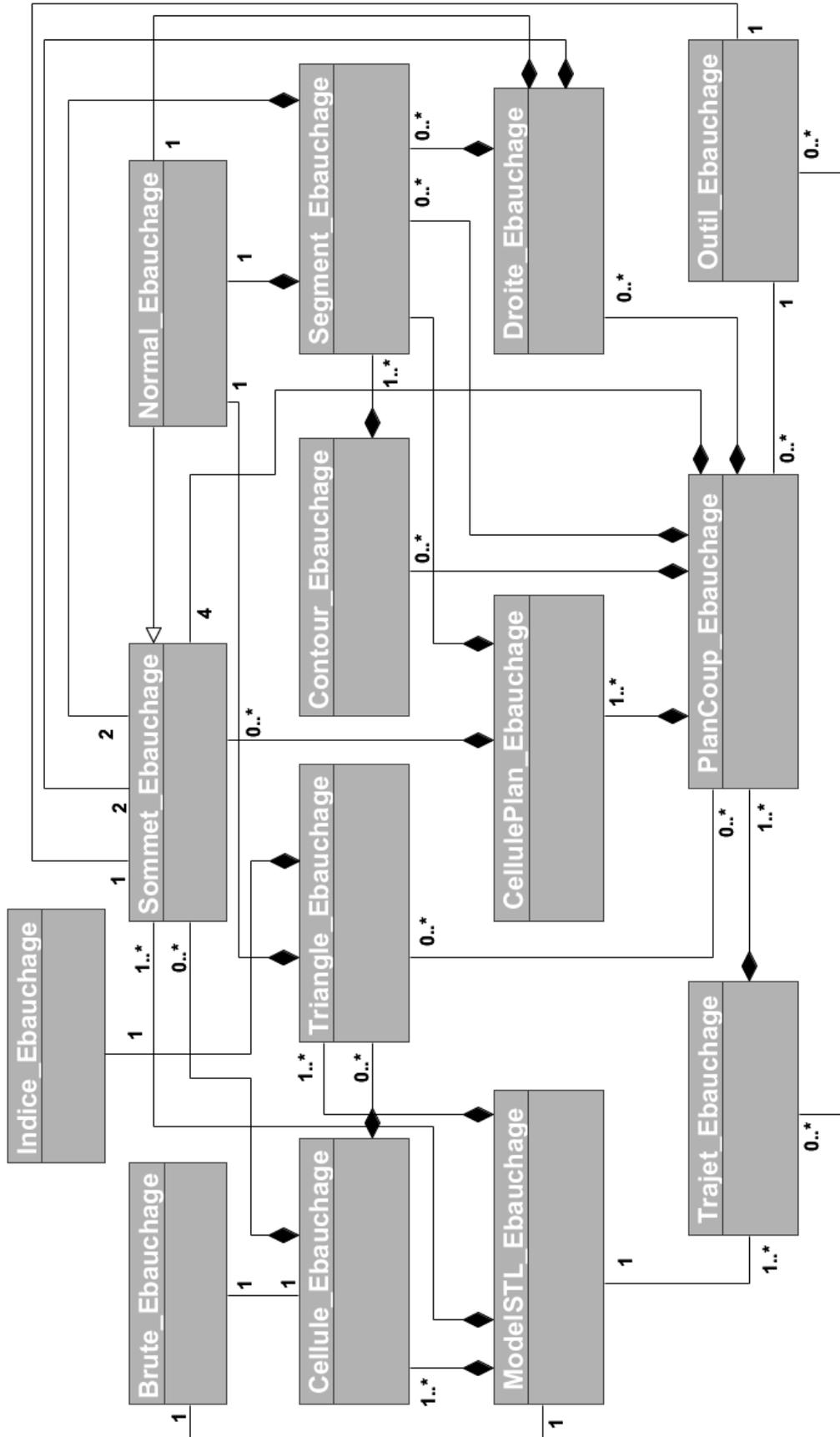


Figure II.37. Diagramme de classes.

La description détaillée (attribut et fonctions) des classes :

### II.3.3.1. Classe Sommet\_Ebauchage

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

Sommet_Ebauchage
#x : double
#y : double
#z : double
+DessinerSommet() : void

Figure II.38. Classe Sommet\_Ebauchage.

### II.3.3.2. Classe Triangle\_Ebauchage

Se compose des indices des sommets qui forment le triangle et sa normale.

Triangle_Ebauchage
-S1 : Sommet_Ebauchage
-S2 : Sommet_Ebauchage
-S3 : Sommet_Ebauchage
-N : Normal_Ebauchage
+DessinerTriangleRendu() : void
+DessinerTriangleFilaire() : void
+DessinerNormal() : void

Figure II.39. Classe Triangle\_Ebauchage.

### II.3.3. Classe Brute\_Ebauchage

C'est la classe qui contient les limites du brute.

Brute_Ebauchage
-Xmin : double
-Xmax : double
-Ymin : double
-Ymax : double
-Zmin : double
-Zmax : double
+DessinerBrute() : void

Figure II.40. Classe Brute\_Ebauchage.

### II.3.3.4. Classe Cellule\_Ebauchage

Cette classe est définie par son brute. Elle contient les indices des sommets et des triangles.

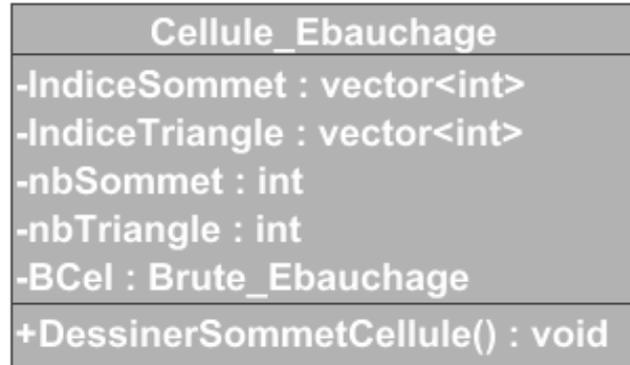


Figure II.41. Classe Cellule\_Ebauchage.

### II.3.5. Classe ModelSTL\_Ebauchage

C'est la classe qui regroupe toutes les informations concernons le modèle STL (l'ensemble des sommets et des triangles). Elle permet aussi de créer les cellules et structurer les paramètres.

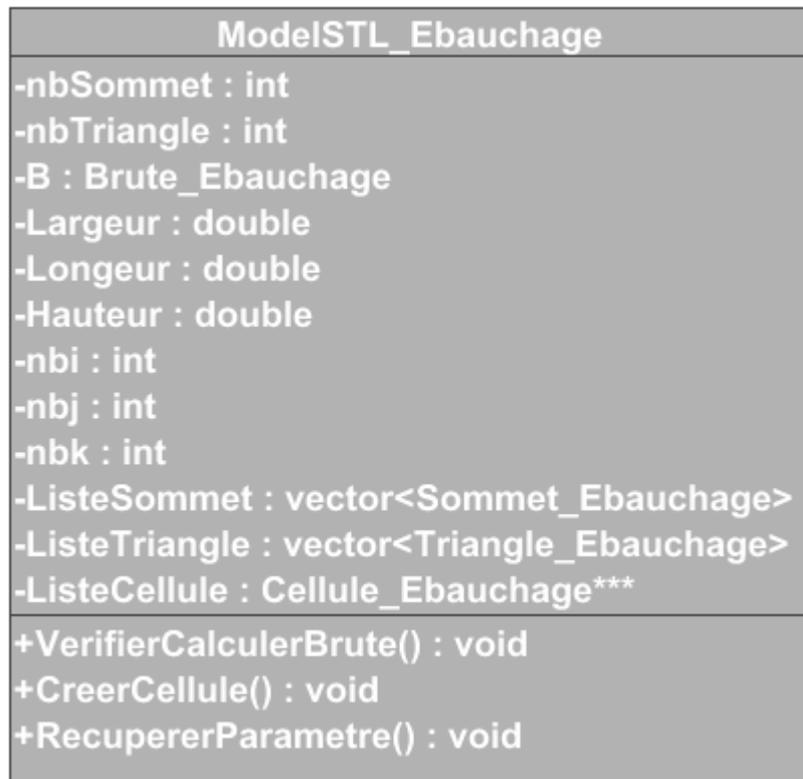


Figure II.42. Classe ModelSTL\_Ebauchage.

### II.3.3.6. Classe Segment\_Ebauchage

Se compose de deux sommets qui forment le segment et la normale.

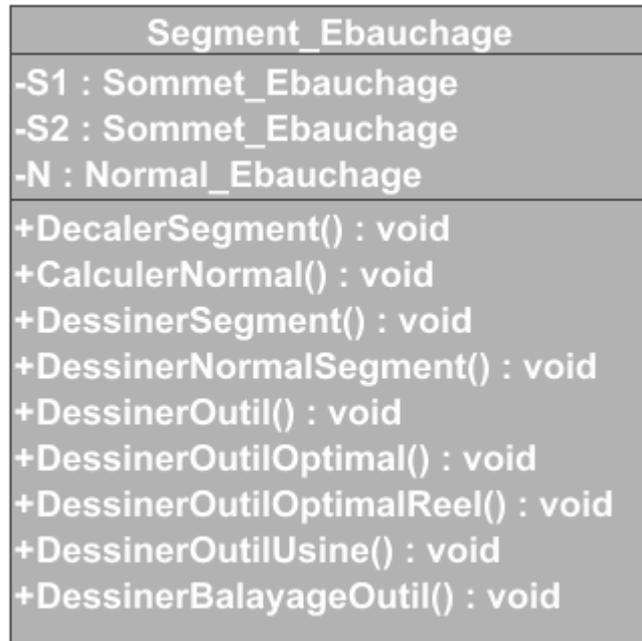


Figure II.43. Classe Segment\_Ebauchage.

### II.3.3.7. Classe Contour\_Ebauchage

Cette classe se compose d'un contour initial, contour décalé et contour offset.

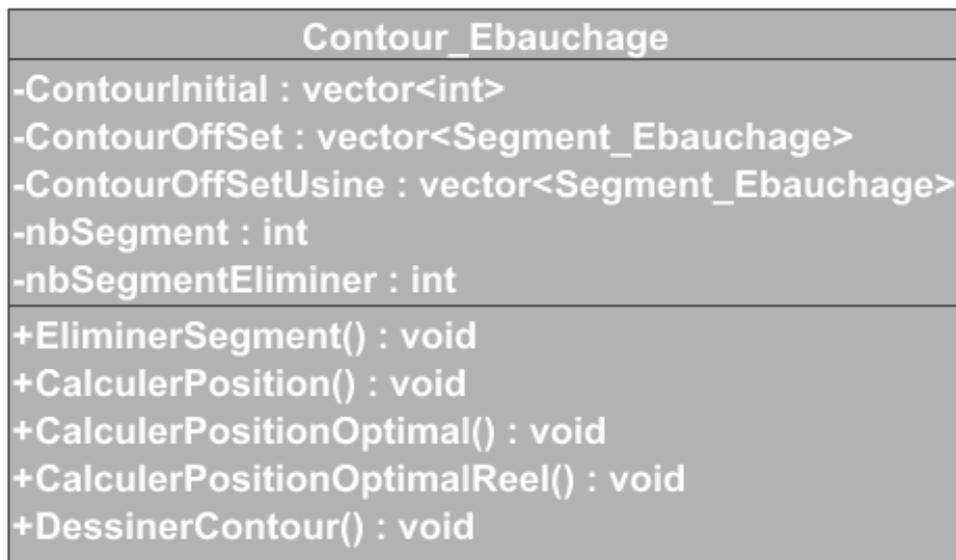


Figure II.44. Classe Contour\_Ebauchage.

### II.3.3.8. Classe Outil\_Ebauchage

Elle contient les informations de l'outil (rayon, longueur, vitesse, centre d'outil, forme et mode de déplacement).

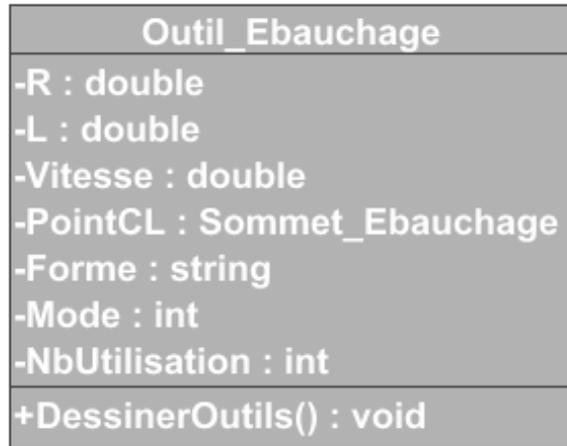


Figure II.45. Classe Outil\_Ebauchage.

### II.3.3.9. Classe Droite\_Ebauchage

Elle contient les segments d'usinage qui génèrent le trajet.

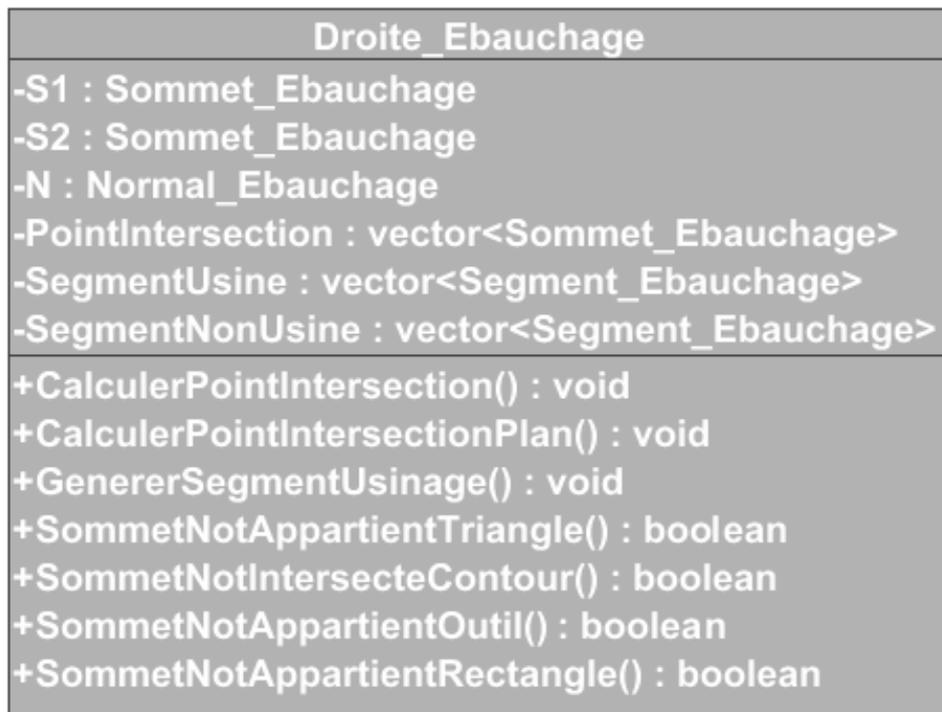


Figure II.46. Classe Droite\_Ebauchage.

### II.3.3.10. Classe PlanCoup\_Ebauchage

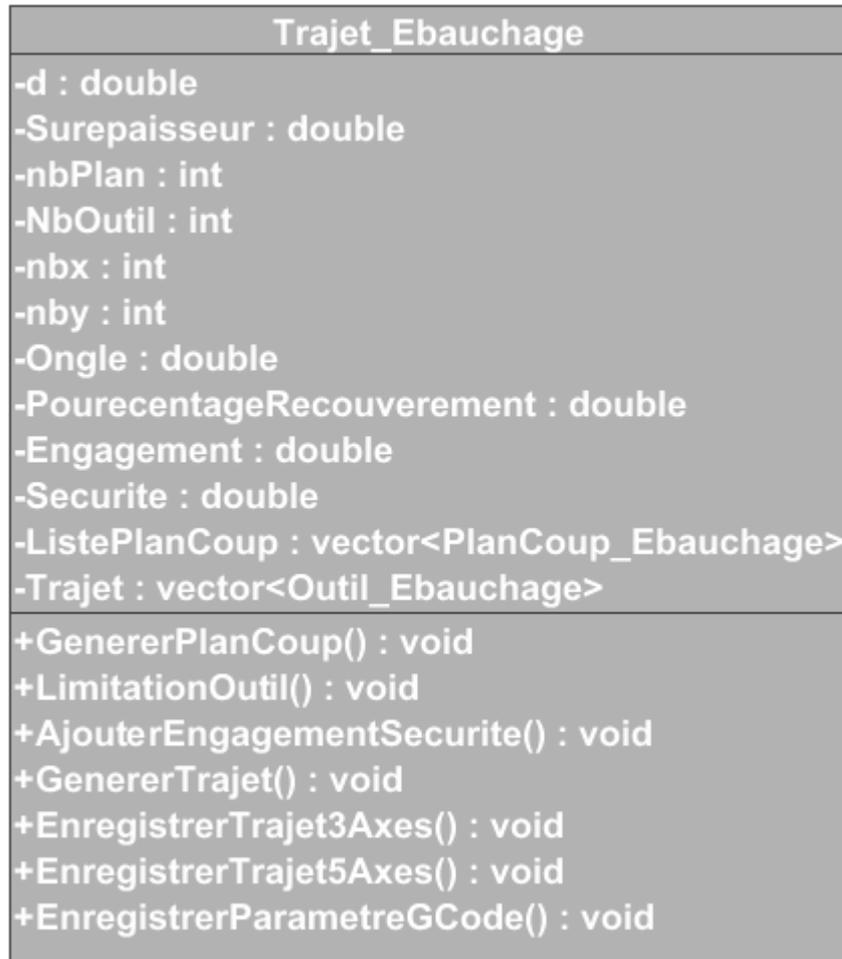
Elle contient les sommets qui forment le plan. Elle permet de générer les segments d'intersection, les contours initiaux, les contours décalés et de calculer l'outil optimum.

PlanCoup_Ebauchage
-A : Sommet_Ebauchage
-B : Sommet_Ebauchage
-C : Sommet_Ebauchage
-D : Sommet_Ebauchage
-nbSegment : int
-nbContour : int
-nbTriangle : int
-nbDroite : int
-outil : int
-outilReel : int
-PasTransversal : double
-ListeTriangle : vector<int>
-ListeSegment : vector<Segment_Ebauchage>
-ListeContour : vector<Contour_Ebauchage>
-ListePointsIntersection : vector<Sommet_Ebauchage>
-ListeDroite : vector<Droite_Ebauchage>
-ListeCellule : CellulePlan_Ebauchage **
-TrajetOneWay : vector<Outil_Ebauchage>
-TrajetZigZag : vector<Outil_Ebauchage>
+CreerCellule() : void
+GenererSegment() : void
+GenererContourInitial() : void
+GenererContourOffSet() : void
+GenererContourOffSetUsine() : void
+CalculerPosition() : void
+CalculerPositionOptimal() : void
+CalculerOutilOptimal() : void
+CalculerPositionOptimalReel() : void
+GenererPasse() : void
+AjouterEngagementSecurite() : void
+GenererTrajetOneWay() : void
+GenererTrajetZigZag() : void
+DessinerPlan() : void
+DessinerSommetTrajet() : void
+DessinerSegmentTrajet() : void

**Figure II.47.** Classe PlanCoup\_Ebauchage.

### II.3.3.11. Classe Trajet\_Ebauchage

C'est la classe principale qui génère les plans de coupe et le trajet d'usinage.



**Figure II.48.** Classe Trajet\_Ebauchage.

## Conclusion

Dans ce chapitre, la conception de l'application développée a été présentée. Cette présentation a été entamée par la définition de la problématique du projet, ainsi que les différents objectifs visés à atteindre. Ensuite la présentation des solutions adoptées, et les diagrammes de conception, qui ont été détaillés en focalisant sur le concept de la modélisation orientée objet et le langage de modélisation UML.

Dans le chapitre suivant, nous allons présenter le module logiciel développé et un test de validation.

# Chapitre III

## Implémentation et validation

## Introduction

Dans le chapitre précédent nous avons présenté l'étude conceptuelle du système, en mettant en évidence les différents objectifs de l'application à développer ainsi que les solutions proposées.

Ce dernier chapitre est consacré à la présentation des résultats obtenus après l'implémentation à travers des exemples de validation.

### III.1. Présentation des langages utilisés

Comme le système développé est réalisé pour être intégré dans la plateforme logicielle de production des pièces complexes développée par l'équipe «CFAO» du «CDTA», les outils de développement utilisés lors de sa mise en œuvre (C++, OpenGL) sont les mêmes utilisés par l'équipe «CFAO» pour se conformer à la tendance qui veut que la majorité des systèmes «CFAO» sont développés en utilisant le C++ et la bibliothèque graphique OpenGL.

#### III.1.1. Présentation du langage C++

C++ est un langage de programmation à usage général qui prend en charge divers modèles de programmation informatique tels que la programmation orientée objet et la programmation générique. Il a été créé par Bjarne Stroustrup et «son objectif principal était de rendre l'écriture de bons programmes plus facile et plus agréable pour le programmeur individuel» [19]. Parmi les environnements de développement : Visual Studio, Eclipse, NetBeans, Delphi/C++Builder, Code::Blocks, etc.



#### III.1.2. Présentation d'Embarcadero C++ Builder 10 Seattle

C++ Builder est un logiciel de développement rapide d'applications conçu par Borland qui reprend les mêmes concepts, la même interface et la même bibliothèque que Delphi en utilisant le langage C++. Embarcadero C++ Builder 10 Seattle est le moyen le plus rapide de créer et de mettre à jours des applications riches en données, hyper connectées et visuellement engageantes pour Windows 10, Mac, Mobile, IoT et plus encore en utilisant le standard C++ et aussi mettre à jour les applications VCL et FMX vers Windows [20].



#### III.1.3. Présentation d'OpenGL

OpenGL (Open Graphics Library) est une API très performante et simple d'utilisation, qui permet de réaliser des rendus et des scènes complexes selon nos besoins. Cette technologie est ouverte et est implémentée par de nombreuses bibliothèques sous des langages de programmation différents dont les langages de script comme Python ou Ruby.



Par ailleurs, OpenGL est portable et multiplateforme. Théoriquement, un même code OpenGL fonctionne sur tous les systèmes d'exploitation et sur toutes les plateformes [21].

## **III.2. 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) i7-7500U CPU @2.70 GHz 2.90 GHz et 8.00Go de RAM.

## **III.3. Présentation de l'application**

### **III.3.1. Fenêtre principale de la plateforme de «CFAO»**

C'est la fenêtre principale qui apparaît après le lancement de la plateforme logicielle. Elle est composée de deux parties principales : la première est consacrée à la visualisation et à la manipulation graphique des différents objets (points, courbes, surfaces, etc.) tandis que la seconde est réservée au lancement de ses différents modules logiciels (Figure III.1).

#### **III.3.1.1. Scène de visualisation**

C'est une fenêtre d'affichage réservée à la visualisation des objets géométriques en 3D en utilisant la bibliothèque « OpenGL ». Dans cette partie sont visualisés les résultats.

#### **III.3.1.2. Barre du Menu principal**

Elle est composée de trois rubriques qui sont :

- Rubrique Fichier : qui comporte toutes les fonctionnalités de manipulation des fichiers (ouverture, création, sauvegarde, etc.).
- Rubrique Option : permet la modification des différents paramètres des courbes ainsi que les paramètres des surfaces et les paramètres d'usinage. Le module logiciel mis au point dans ce travail est lancé à partir de cette rubrique.
- Rubrique Aide : pour l'affichage de certaines informations concernant la réalisation et l'utilisation de l'application.

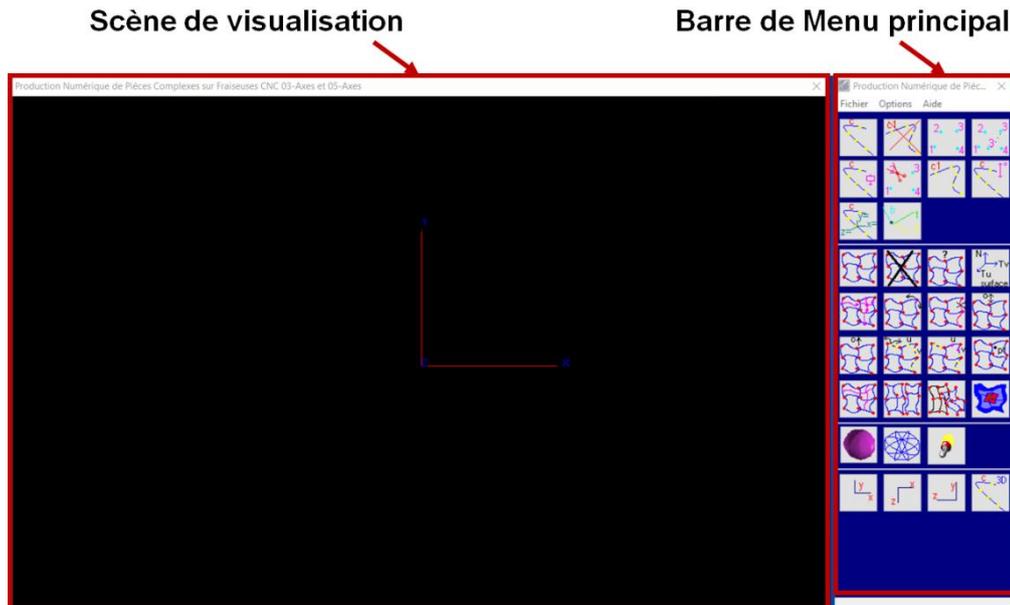


Figure III.1. Environnement de « CFAO ».

### III.3.2. Rubrique « Ebauchage 03-Axes des modèles STL »

La partie contenant le module logiciel développé dans le cadre de ce travail, est aussi lancée à partir du menu au niveau de la rubrique « Option, Applications Industrielles, Ebauchage 03-Axes des modèles STL » (Figure III.2).

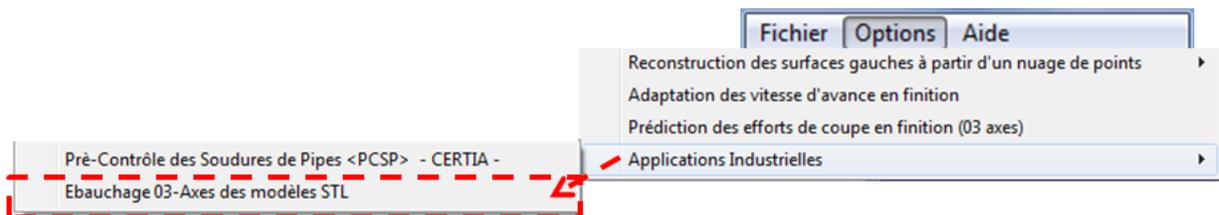


Figure III.2. Rubrique « Ebauchage 03-Axes des modèles STL ».

### III.4. Présentation de l'application développée

L'application développée est composée de plusieurs onglets où chaque onglet est dédié à des phases du processus de la stratégie d'ébauchage. La fenêtre principale de notre projet est composée de six (06) onglets (Figure III.3) :

- Onglet 1 : Lecture du fichier STL.
- Onglet 2 : Génération des contours.
- Onglet 3 : Déterminations des outils.
- Onglet 4 : Création des passes.
- Onglet 5 : Trajet d'Ebauchage.
- Onglet 6 : Simulation du trajet.



Figure III.3. Onglets de l'application développée.

### III.4.1. Lecture du fichier STL

Le premier onglet permet la lecture du fichier texte qui regroupe toutes les informations contenues dans la définition des triangles et leurs sommets récupérés à partir du modèle STL. Il est lancé par un simple clic sur le bouton « Ouvrir Fichier STL » suivi par la sélection du fichier à ouvrir. Une fois le fichier est lu, le brut est divisé en cellules ou le nombre des cellules suivant les trois axes **X**, **Y** et **Z** est spécifié. Donc, il est possible de visualiser les informations suivantes (Figure III.4) :

- Les limites de la pièce brute :  $X_{min}$ ,  $Y_{min}$ ,  $Z_{min}$ ,  $X_{max}$ ,  $Y_{max}$ ,  $Z_{max}$ , longueur, largeur et la hauteur.
- Le nombre de sommets et le nombre de triangles.
- Le brut du modèle en deux modes filaire et rendu.
- Les sommets du brut.
- Les triangles du modèle en deux modes filaire et rendu.
- Les cellules du brut en deux modes filaire et rendu.
- Les sommets du modèle de chaque cellule.

Ebauchage des Pièces de Formes Complexes en 03-Axes

**Lecture du fichier STL** | Génération des contours

Ouvrir Fichier STL

**Limites du brut**

Xmin : -49.871 Ymin : -77.904 Zmin : 0.000  
 Xmax : 70.129 Ymax : 72.096 Zmax : 50.000  
 Longueur : 120.000 Largeur : 150.000 Hauteur : 50.000  
 Nombre de sommets : 23558 Nombre de triangles : 47116

**Nombre de cellules par axe**

Par axe X : 30  
 Par axe Y : 30  
 Par axe Z : 30

Structurer les Paramètres

**Visualisation**

Brut filaire 5  
 Brut Rendu  
 Sommets Brut 5  
 Sommets 5  
 Normales des triangles 3 10  
 Triangles en filaire 5  
 Triangles Rendu  
 Brut Cellule 5  
 Brute Cellule Rendu  
 Sommet Cellule 5

Figure III.4. Onglet « Lecture du fichier STL ».

### III.4.2. Génération des contours

Dans cet onglet, l'utilisateur doit passer par quatre étapes :

- **Etape 1** : introduction de la profondeur de passe et génération des plans de coupe par un simple clic sur le bouton « Générer les plans de coupe ».
- **Etape 2** : génération des segments d'intersection avec chaque plan de coupe.
- **Etape 3** : création des contours à partir des segments d'intersection par un clic sur le bouton « Générer les contours initiaux ».
- **Etape 4** : introduction de la surépaisseur d'usinage et génération des contours décalés (Figure III.5).

A ce niveau, il est possible de visualiser globalement les informations suivantes :

- Plans de coupe, contours initiaux et contours décalés.
- Segments d'intersection et leurs normales.

De même, il est possible de visualiser pour chaque plan de coupe :

- Plan de coupe.
- Triangles se chevauchant avec le plan de coupe.
- Brutes des cellules.
- Segments d'intersection, leurs numéros et leurs sommets.
- Normales des segments.
- Contours initiaux et contours décalés.
- Segments de chaque cellule.

Figure III.5. Onglet « Génération des contours ».

### III.4.3. Détermination des outils

Ce troisième onglet est réservé à la partie de détermination des outils. L'intérêt principal de la première partie est le test manuel des différents outils. L'utilisateur peut choisir

l'indice de l'outil souhaité qui est récupéré à partir d'une base de données d'outils. Ensuite, par un simple clic sur le bouton « Calculer position » (Figure III.6). Par la suite, le bouton « Calculer Outil Optimal » sert à déterminer l'outil optimal en parcourant la base des données d'outils du plus grand au plus petit jusqu'à trouver l'outil hors interférence. Enfin, le bouton « Limiter les outils » sert à limiter le nombre d'outils utilisés en calculant le nombre d'utilisation pour chaque outil optimal puis en affectant pour chaque plan son outil réel. Une fois la détermination des outils est terminée, l'utilisateur peut visualiser la position de l'outil.

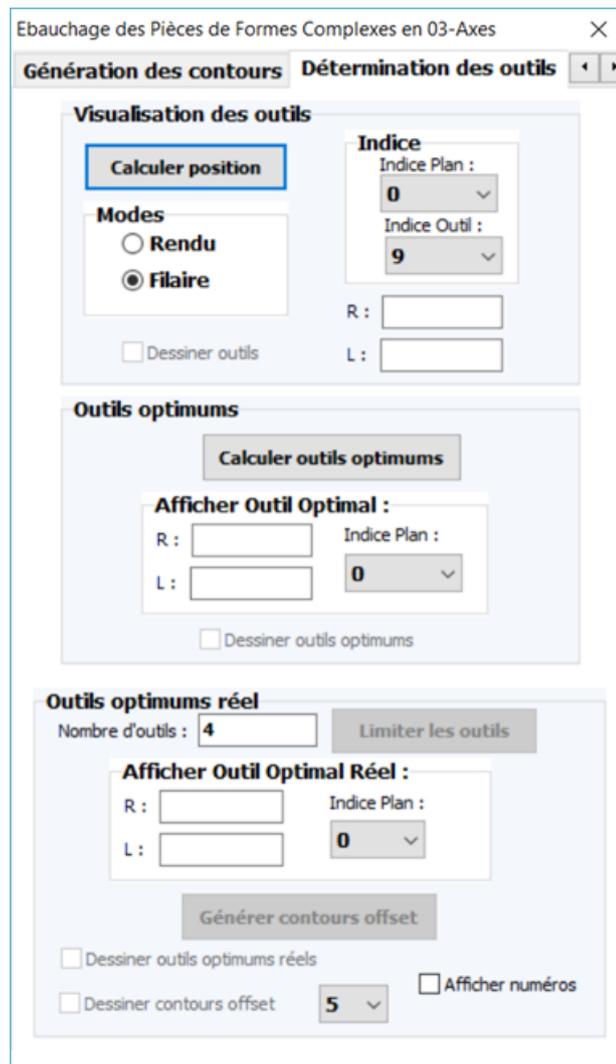


Figure III.6. Onglet « Détermination des outils ».

#### III.4.4. Création des passes

Cet onglet contient les étapes nécessaires à la direction d'usinage. Il est divisé en trois parties (Figure III.7) :

- Fixation de l'angle d'inclinaison : cette partie a pour but d'introduire l'angle d'inclinaison des droites d'usinage par rapport à l'axe des X.

- Génération des passes : introduction du pourcentage de recouvrement suivi par un clic sur le bouton « Générer les passes ». A ce niveau, l'utilisateur peut visualiser les passes (droites d'usinage).
- Création des segments d'usinage : un clic sur le bouton « Calculer point d'intersection » permet de calculer l'intersection entre les passes et les contours offset. Par la suite, un simple clic sur le bouton « Créer segment d'usinage » permet d'identifier les segments usinés des segments non usinés.

L'utilisateur peut visualiser les points d'intersection, les segments usinés et non usinés, ainsi que l'outil et le volume balayé par l'outil.

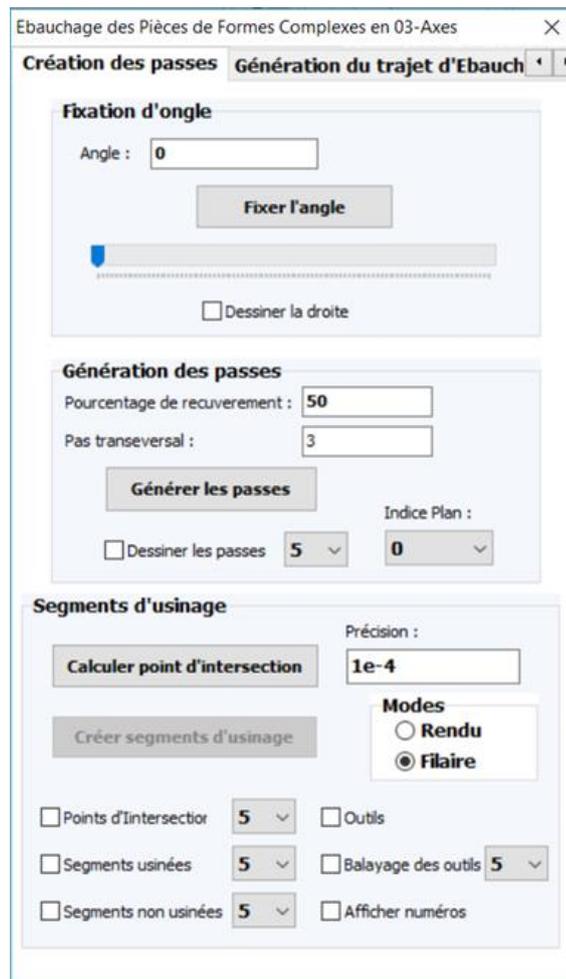


Figure III.7. Onglet « Création des passes ».

### III.4.5. Génération du Trajet d'Ebauchage

Cet onglet permet la génération de la trajectoire d'usinage. En spécifiant la distance d'engagement et la distance de sécurité et le mode de balayage d'outil (« One-Way » ou « Zig-Zag », le trajet d'outil est généré par un simple clic sur le bouton « Générer le trajet ». A ce niveau, l'utilisateur a la possibilité de visualiser les sommets et les segments du trajet d'une façon globale ou partielle (Figure III.8).

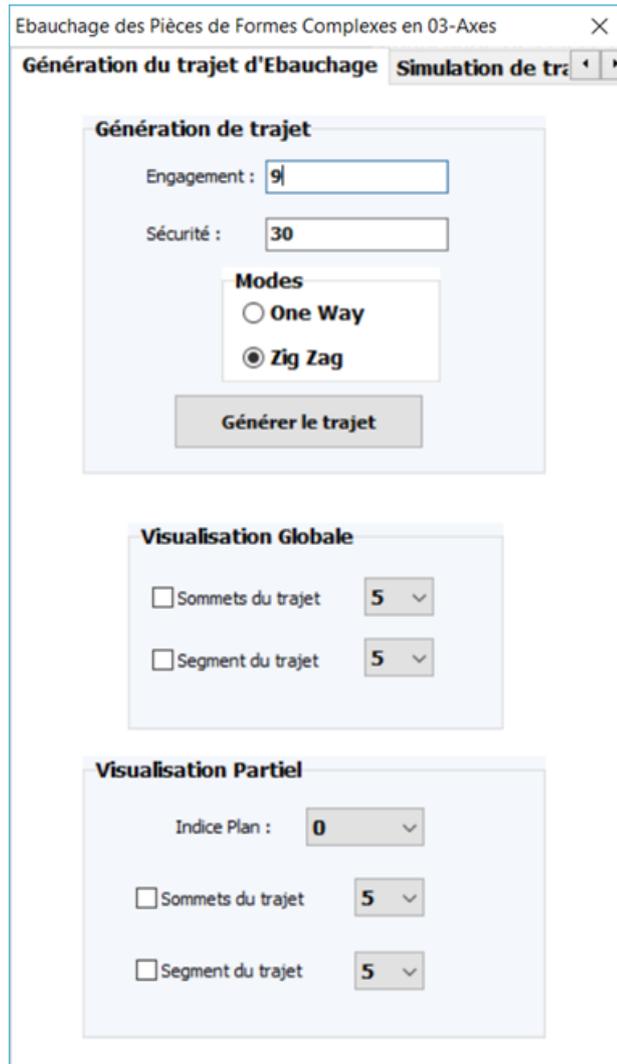


Figure III.8. Onglet « Génération du Trajet d'Ebauchage ».

### III.4.6. Simulation du trajet d'outils

Le dernier onglet permet de simuler virtuellement les mouvements des outils par rapport à la pièce à usiner. Des manipulations sont intégrées à cet onglet (Figure III.9). Il est possible de visualiser :

- Les paramètres d'outils et les coordonnées des points CL.
- Le déplacement de l'outil.
- La trajectoire d'usinage.
- Le pourcentage d'avancement.

Enfin, l'enregistrement du trajet d'outils et les paramètres du fichier « G-Code » dans un fichier texte.

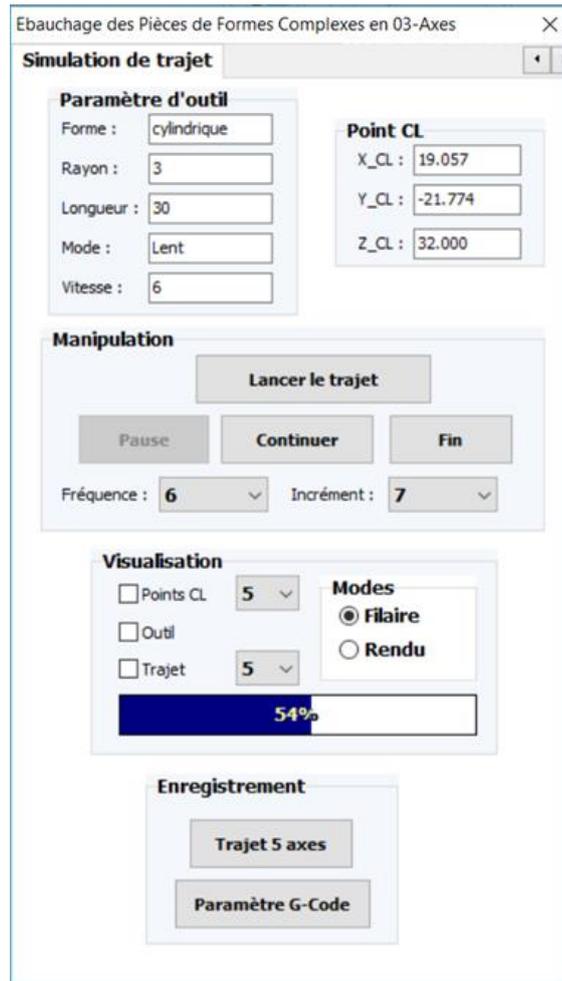
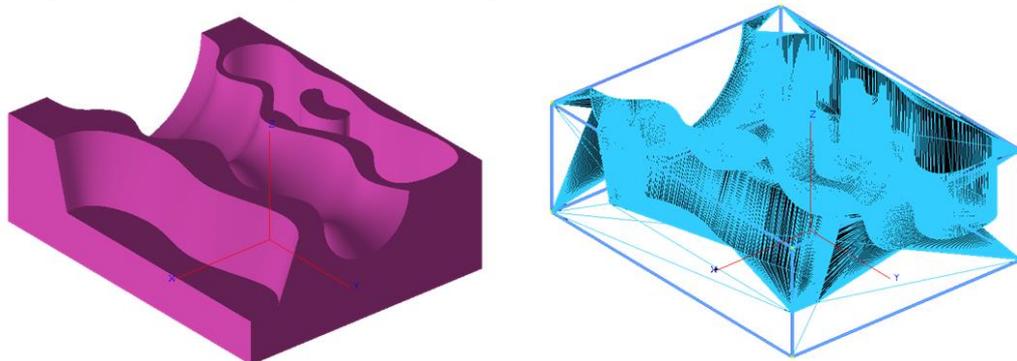


Figure III.9. Onglet « Simulation de trajet ».

### III.5. Test et validation

La phase de validation consiste à tester les approches proposées lors de la phase de conception. Pour cela, la pièce de la (Figure III.10.a) est prise comme support de validation. Le traitement est effectué sur le modèle STL (Figure III.10.b) depuis la lecture du fichier jusqu'à la génération de la trajectoire d'usinage en ébauche.



a. Triangle en rendu.

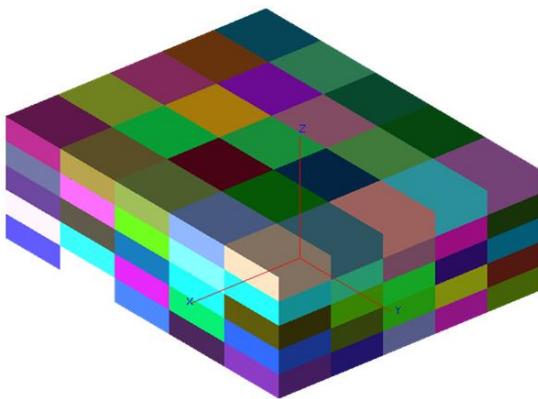
b. Triangle en filaire et brut de la pièce.

Figure III.10. Modèle STL de la pièce de test.

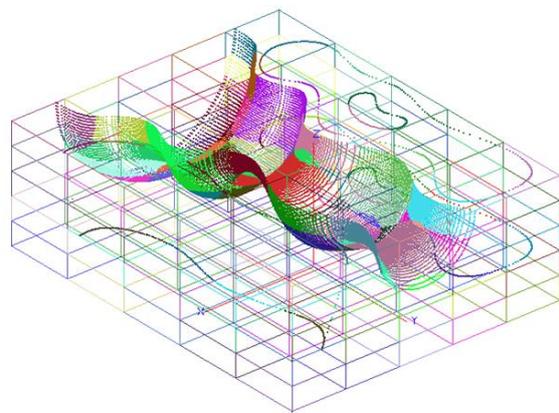
➤ **Etape 1 :** lecture du fichier STL, calcul des limites du brut (Tableau 1), subdivision du brut de la pièce en cellules et affectation des sommets aux cellules (Figure III.11). Le nombre de cellules suivant les trois axes X, Y et Z est fixé égal à 5.

**Tableau 1.** Paramètres du modèle STL.

<b>Xmin : -49.871 mm</b>	<b>Ymin : -77.904 mm</b>	<b>Zmin : 0.000 mm</b>
<b>Xmax : 70.129 mm</b>	<b>Ymax : 72.096 mm</b>	<b>Zmax : 50.000 mm</b>
<b>Longueur : 120.000</b>	<b>Largeur : 150.000</b>	<b>Hauteur : 50.000 mm</b>
<b>Nombre de sommets : 23558 mm</b>		<b>Nombre de triangles : 47116 mm</b>



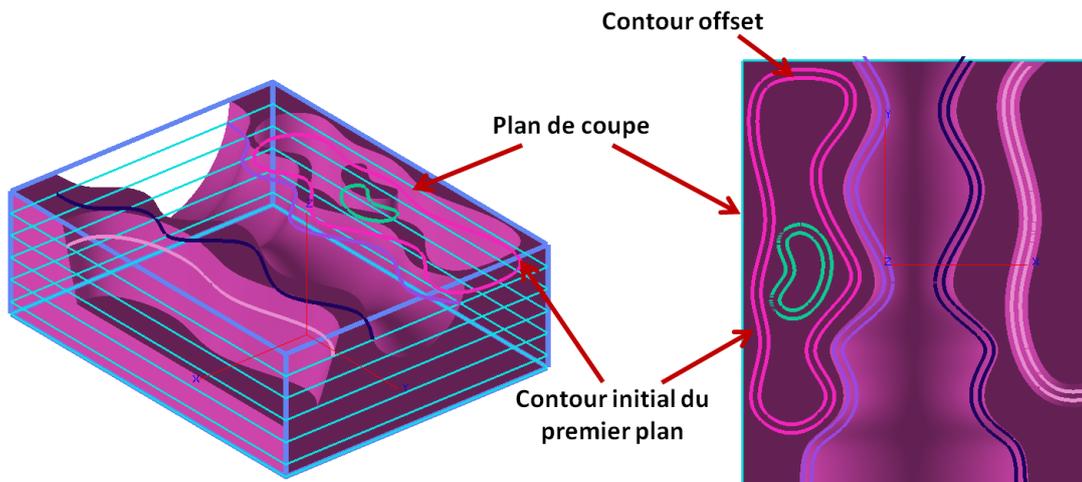
a. Cellules en rendu



a. Cellules en filaire avec les sommets affectés

**Figure III.11.** Création des cellules et affectations des points.

➤ **Etape 2 :** génération des plans de coupe pour une profondeur de passe égale à 9 mm (Figure III.12.a), génération des contours initiaux et les contours décalés pour une surépaisseur de 3 mm (Figure III.12.b).



a. Plans de coupe

b. Contours initiaux et contours décalés

**Figure III.12.** Génération des contours.

- **Etape 3 :** visualisation des différents outils.

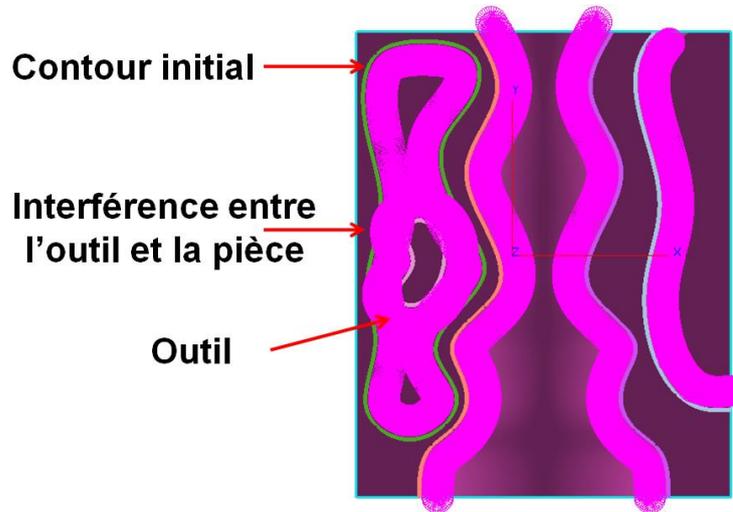


Figure III.13. Positionnement de l'outil pour le premier plan de coupe.

- **Etape 4 :** calcul des outils optimums et détermination des outils réels à utiliser et génération des contours offset. En variant le nombre d'outils disponible, une comparaison concernant l'outil optimum choisi dans les plans 31 (Tableau 2) et 34 (Tableau 3) a été faite.

Tableau 2. Tableau comparatif du plan 31.

2 outils		3 outils	
Outil théorique	Outil réel	Outil théorique	Outil réel
<p>R = 12 L = 120</p> <p>outil</p>	<p>R = 3 L = 30</p> <p>outil</p>	<p>R = 12 L = 120</p> <p>outil</p>	<p>R = 12 L = 120</p> <p>outil</p>

Tableau 3. Tableau comparatif du plan 34.

2 outils		3 outils	
Outil théorique	Outil réel	Outil théorique	Outil réel
<p>R = 10 L = 100</p> <p>outil</p>	<p>R = 3 L = 30</p> <p>outil</p>	<p>R = 10 L = 100</p> <p>outil</p>	<p>R = 3 L = 30</p> <p>outil</p>

➤ **Etape 5 :** pour des droites d'usinage inclinés de  $0^\circ$  et de  $60^\circ$  par rapport à l'axe des X et un pourcentage de recouvrement de 50%, les segments usinés et les segments non usinés sont identifiés (Figure III.14).

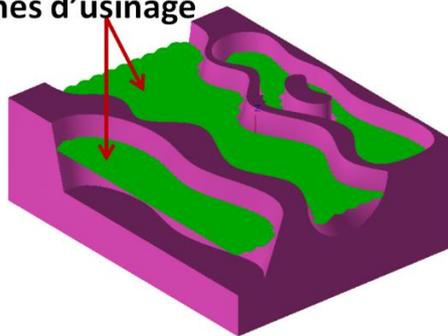


a. Segments d'usinage « angle  $0^\circ$  ».



b. Segments d'usinage « angle  $60^\circ$  ».

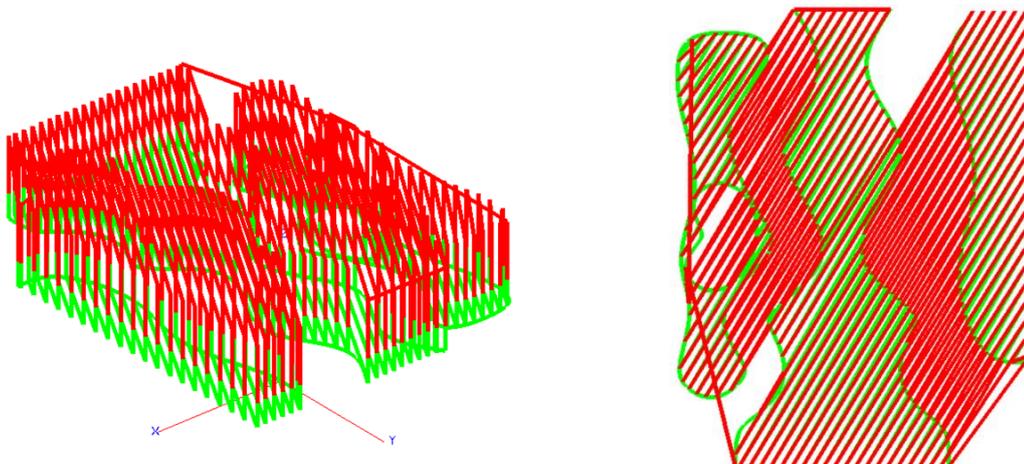
Zones d'usinage



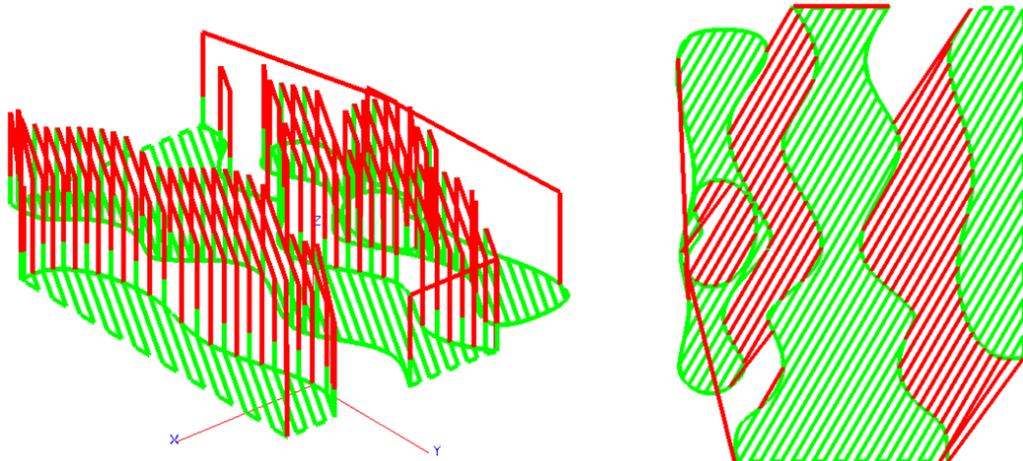
c. Zones d'usinage.

Figure III.14. Zones d'usinage.

➤ **Etape 6 :** la Figure III.15 montre les trajets d'outils pour les modes « One-Way » et « Zig-Zag » pour une distance de sécurité 30mm et une distance d'engagement de 9mm.



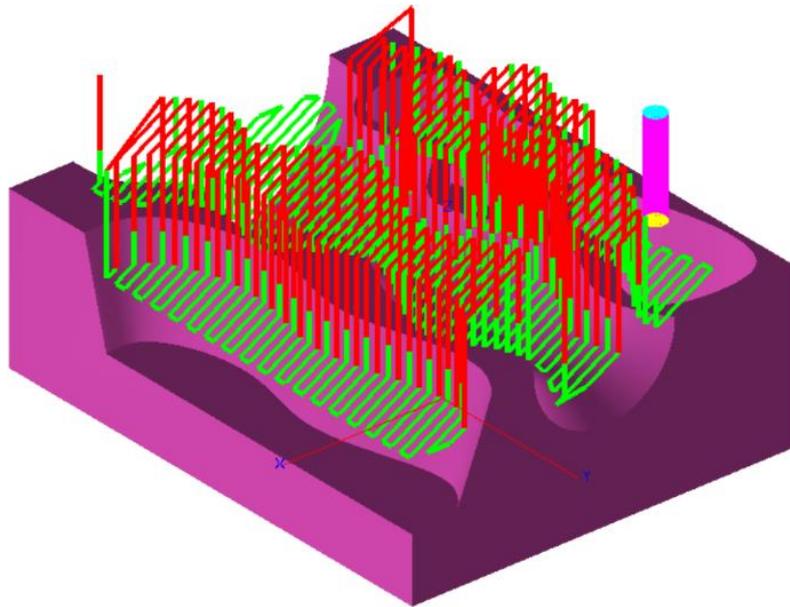
a. Trajet d'outils en « One-Way ».



b. Trajet d'outils en « Zig-Zag ».

**Figure III.15.** Trajet d'ébauchage avec les deux modes de balayage.

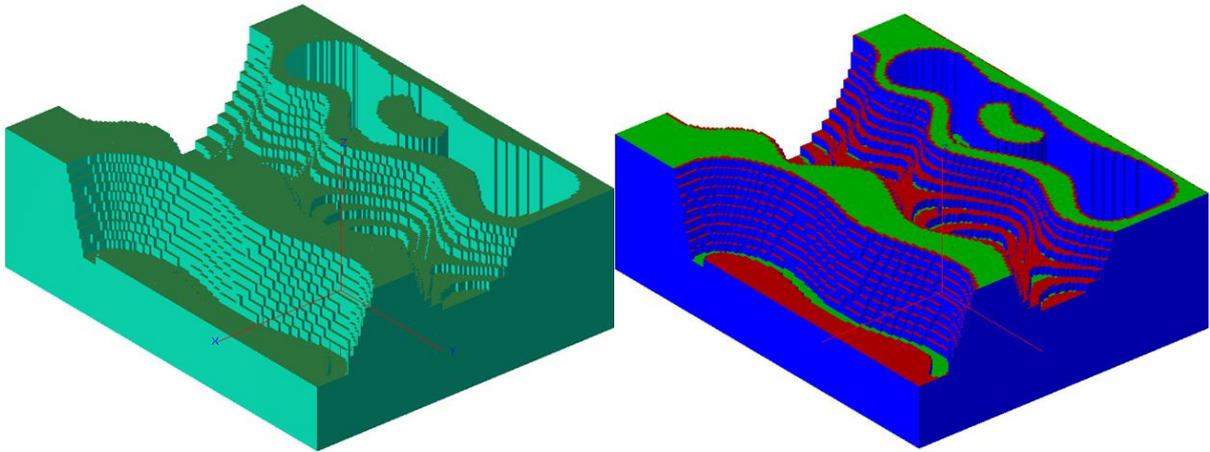
➤ **Etape 7 :** à n'importe quel moment, l'outil est visualisé et ses informations sont récupérées ainsi que les informations du point « CL » en cours sont récupérées.



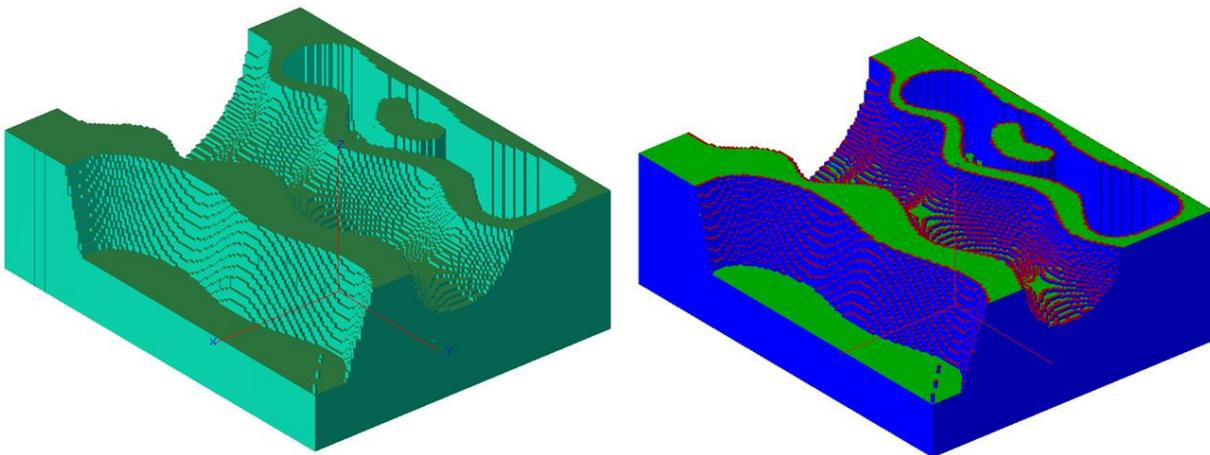
**Figure III.16.** Simulation du trajet en 03-Axes.

### III.6. Validation du trajet par simulation

La validation du trajet généré est réalisée à travers la simulation virtuelle de l'opération d'enlèvement de matière. Pour cela, le trajet d'outils est généré pour deux profondeurs de passe à savoir 3mm et 1mm. Pour chaque profondeur sont visualisés l'aspect général de la pièce après l'opération d'ébauchage et les écarts d'usinage par rapport au modèle STL de la pièce (Figure III.17). Il ressort d'après ces résultats que le trajet généré par notre application logicielle est fiable et que la forme finale ne présente pas des problèmes d'interférences.



a. Usinage avec profondeur égale à 3mm.



b. Usinage avec profondeur égale à 1mm.

**Figure III.17.** Validation du trajet par simulation virtuelle de l'usinage.

## Conclusion

Ce chapitre, a été consacré à l'implémentation de l'application développée et à la présentation des résultats des tests menés. Nous avons commencé par la présentation de l'environnement de développement, les langages de programmation utilisés, ainsi que le matériel utilisé et les différentes interfaces et leurs fonctionnalités. Afin de valider l'approche proposée, un exemple de modèle STL a été traité depuis la lecture du fichier STL jusqu'à la simulation du trajet.

## Conclusion générale :

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 permettant la sélection des outils cylindriques optimums, évitant les problèmes d'interférences, et la génération du trajet d'outils pour l'ébauchage des pièces de formes complexes, définies par leurs modèles STL, sur des fraiseuses numériques à 03-axes en utilisant la stratégie « Plans Parallèles » pour les deux modes de balayage « One-Way » et « Zig-Zag ».

Lors de la réalisation de ce projet, nous avons mené une étude bibliographique sur le processus de production des pièces mécaniques de formes complexes, le format d'échange de données et l'opération d'ébauchage sur des fraiseuses numériques à 03-axes ainsi que la stratégie d'ébauchage « Plans Parallèles ». Par la suite, nous avons présenté la solution proposée et l'étude conceptuelle de l'application. Dans la dernière partie, nous avons montré les fonctionnalités du module logiciel développé avec un exemple de validation.

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

- Lecture du fichier STL et structuration des paramètres.
- Génération des contours pour délimiter les zones à usiner.
- Détermination des outils cylindriques optimums évitant les interférences.
- Création des passes d'usinage.
- Génération du trajet d'usinage suivant les passes et les contours offset.
- Simulation virtuelle des mouvements des outils.

En perspective, nous recommandons de traiter les thématiques suivantes :

1. Intégration de la notion du parallélisme dans le calcul des différents paramètres associés à chaque plan de coupe (contours, contours décalés, outil optimums, trajet, etc.).
2. Combinaison de plusieurs outils dans le même plan de coupe pour minimiser davantage le temps d'usinage et par conséquent les coûts.
3. Combinaison des deux formes d'outils hémisphériques et cylindriques pour ébaucher les parties à fortes courbures de la pièce.

4. Développement et implémentation des stratégies de demi-finition des pièces de formes complexes.
5. Génération du programme d'usinage « G-Code » en fonction de différentes configurations des fraiseuses à 03-axes.
6. Optimisation du trajet d'outil pour chaque plan de coupe.
7. Adaptation des vitesses d'avance en fonction du volume de la matière enlevée.
8. Développement et implémentation de stratégies d'ébauchage des pièces de formes complexes sur des fraiseuses numériques à 05-axes.
9. Simulation virtuelle des fraiseuses numériques à 05-axes.
10. Programmation en utilisant des cartes GPU.

- [1]. Hasen Allabad, « Conception du système de fabrication de pièces mécaniques en grande série : formalisation de la configuration géométrique (enveloppe) et cinématique de Machine-Outil Reconfigurable (MOR) », Science de l'ingénieur, Art et Métiers ParisTech, 2009.
- [2]. Christophe Tournier, « Contribution à la conception des formes complexes : la surface d'usinage en fraisage à 5 axes isocrête », Thèse de Doctorat, LURPA, France, 2001.
- [3]. Colombié Michel, « Fabrication par usinage », Dunod, Paris, 2003.
- [4]. N. Aifaoui, « Intégration CAO/Calcul, une approche par les features de calcul », Thèse de Doctorat, Université de Valenciennes, France, Juillet 2003.
- [5]. A. Cheikh, « Cours Programmation DNC et FAO », Université de Tlemcen, 2012.
- [6]. Dubois Patrice, Aoussat Améziane, Duchamp Robert, « Prototypage Rapid Généralités », Laboratoire Conception de Produits Nouveaux et Innovation à l'École Nationale Supérieure d'Arts et Métiers de Paris, France, 2000.
- [7]. 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.
- [8]. Pierra Guy, « Représentation et Echange de données techniques », Laboratoire d'Informatique Scientifique et Industrielle E.N.S.M.A, France, 2000.
- [9]. Dolenc A, « An Overview of Rapid Prototyping Technologie in Manufacturing », Université de technologie d'Helsinki : Service national d'information technique, Juillet 24, 1994.
- [10]. Sebti Khaled, Moulay Abdelhamid, « Modélisation volumique des pièces de formes complexes par triple dexels à partir de leurs modèles STL », Mémoire de Master, Faculté des sciences, département d'informatique, Université de Blida, 2015.
- [11]. Valentan B, Brajlil T, Drsvtensek I, Balic J, « Basic Solutions on Shape Complexity Evaluation of STL Data », Faculté d'ingénierie mécanique, Université de Maribor, Janvier 2008.
- [12]. Aguenini B, Messaoudi H, « Simulation et vérification de l'opération d'ébauchage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes », Mémoire de Master, Faculté des sciences, département de génie informatique, Université de Blida, 2006.

- [13]. Tahraoui M, « Contribution à l'amélioration de la qualité d'usinage en fraisage à sec », Mémoire de Master, Faculté des sciences de l'ingénieur, département de génie mécanique, Université de Annaba, 2018.
- [14]. Yann Quinsat, « Contribution au choix de stratégies d'usinage : application à la finition des pièces de formes complexes », Thèse de Doctorat, Université Blaise Pascal, France, 2004.
- [15]. Djadda L, Saadi Y, « Conception et réalisation d'un distributeur à axe rotatif pour mini pelleuse », Mémoire de Master, Faculté de technologie, département de génie mécanique, Université de Blida, 2018.
- [16]. Sandvik, Coromant, « Fraisage – Principes », Technique de l'ingénieur, BM7082, 2001.
- [17]. Boukercha B, Kaouane S, « Vers l'approche contours décalés dans le processus d'usinage des surfaces de formes libres en ébauche », Mémoire de Master, Faculté des sciences, département d'informatique, Université de Blida, 2009.
- [18]. Bey M, Tchantchane Z, Billal B, Kaouane S, « Optimisation de l'opération d'ébauchage des surfaces gauches par l'intégration des outils optimums et des vitesses d'avance », First International Conference on Industrial Engineering & Manufacturing, ICIEM 2010, May 9-10, 2010, Batna University, Batna, Algeria.
- [19]. Bjarne Stroustrup, « Le langage de programmation C ++ », troisième édition. Reading, MA: Addison-Wesley, 1997.
- [20]. Embarcadero Technologies, Embarcadero, <https://www.embarcadero.com/>, 2020.
- [21]. Margaret Rouse, «WhatIs.com», <https://whatis.techtarget.com/definition/OpenGL-Open-Graphics-Library>, Mars 2011.