

République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur **CDTA**

Et de la Recherche Scientifique
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

**Génération des Trajectoires d'Outils des Opérations
de Demi-Finition et de Finition des Pièces Complexes
par la Stratégie « Z-Constant » sur Fraiseuses
Numériques à 03-Axes**

Réalisé par :

M^{elle}. EL-GHOUL Ikram

M^{elle}. SERRADJ Fairouz

Soutenu devant :

Mr. BEY Mohamed	CDTA	Encadreur
Mr. BENDIFALLAH Hassène	CDTA	Encadreur
Mr. HAMOUDA Mohamed	USDB	Promoteur
Mme. ABED Hafida	USDB	Présidente
M ^{elle} . BACHA Sihem	USDB	Examinatrice
Mr. CHERIF ZAHAR Amine	USDB	Examineur

2020/2021

Résumé :

Ce travail s'insère dans le cadre de développement d'une plateforme logicielle pour la production des pièces de formes complexes, sur des fraiseuses numériques multi-axes, 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 aux opérations de demi-finition et de finition des pièces complexes, définies par leurs modèles STL sur des fraiseuses numériques à 03-axes, en considérant la stratégie d'usinage « Z-Constant ». Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle de l'équipe CFAO un module logiciel graphique et interactif permettant de déterminer les outils optimaux évitant les interférences et les collisions, leurs combinaisons et le trajet d'outils global. Ce travail permettra de réduire le cycle de développement de nouveaux produits, d'augmenter la productivité et de réduire les temps d'usinage et par conséquent les coûts.

Mots Clés : Pièce Complexe, Modèle STL, Z-Constant, G-Code, CL-File, Trajet d'outils, Demi-Finition, Finition, Interférence, Collision, Fraiseuse 03-axes.

Abstract:

This work is part of a software platform for the production of parts of complex shapes, on multi-axis CNC milling machines, initiated by the Computer Aided Design and Manufacturing (CFAO) team of the Production and Robotics Division (DPR) of the Centre for the Development of Advanced Technologies (CDTA).

In this project, we are interested in the semi-finishing and finishing operations of complex parts, defined by their STL models on 03-axis CNC milling machines, considering the « Z-Constant » machining strategy. The aim is to design, develop and integrate into the software platform a graphical and interactive software module to determine the optimal tools avoiding interferences and collisions, their combinations and the tool path. This work reduces the product development cycle, increases productivity, reduces machining times, and therefore costs.

Keywords: Complex Part, STL Model, Z-Constant, G-Code, CL-File, Tool Path, Semi-Finishing, Finishing, Interferences, Collisions, 03-axis CNC Milling Machine.

ملخص:

هذا العمل هو جزء من تطوير منصة برمجية لإنتاج قطع ميكانيكية ذات الأشكال المعقدة، على آلات التفريز الرقمية متعددة المحاور، والتي بدأها فريق التصميم والتصنيع بمساعدة الكمبيوتر التابع لشعبة الإنتاج والروبوتات (DPR) في مركز تطوير التقنيات المتقدمة (CDTA).

نحن مهتمون في هذا المشروع بعمليات نصف تشطيب وتشطيب القطع الميكانيكية ذات الأشكال المعقدة، المحددة بنماذج STL على آلات التفريز الرقمية ذات 03 محاور، بواسطة استراتيجية التصنيع "Z-Constant" والهدف من ذلك هو تصميم وحدة برمجيات رسومية وتفاعلية وتطويرها وإدماجها في المنصة البرمجية لفريق CFAO من أجل تحديد أدوات القطع المثلى التي تتجنب التداخلات والاصطدامات وتوليفاتها والمسار الشامل للأدوات. سيقلل هذا العمل من دورة تطوير المنتجات الجديدة، ويزيد الإنتاجية، وسيققل من أوقات استخدام الآلات وبالتالي من التكاليف.

الكلمات المفتاحية: قطع معقدة، نموذج STL ، "Z-Constant" ، G-Code ، CL-File ، مسار الأداة، نصف تشطيب، تشطيب، التداخل، التصادم، آلة تفريز ذات 03 محاور.

Remerciements

*Par les premières lignes de ce document, j'éprouve une immense gratitude et une reconnaissance infinie à l'égard d'ALLAH le tout puissant qui ne cesse de nous protéger, qui nous a donné la foi, la connaissance, la volonté, la force, la patience et le courage d'accomplir ce modeste travail, qui nous a permis de voir ce jour tant attendu,
ALHAMDOLIELLAH.*

Nous adressons nos profondes gratitude et respectueuses reconnaissances à notre encadreur professionnel Mr. BEY MOHAMED. Il nous a beaucoup aidé à orienter ce travail, et également nous a encouragées pendant les périodes difficiles. Nous avons eu le privilège de travailler parmi votre équipe et d'admirer vos qualités et vos valeurs, votre sérieux, votre compétence et votre sens du devoir nous ont énormément marqués, on a eu l'honneur d'être parmi vos étudiants et de bénéficier de votre riche enseignement. Vos qualités pédagogiques et humaines sont pour nous un modèle. Votre gentillesse, et votre disponibilité permanente ont toujours suscité notre admiration. Veuillez bien monsieur recevoir nos remerciements pour votre bonne volonté d'accepter de nous encadrer.

Nous remercions également Mr. HASSENE BENDIFALLAH qui nous a guidé, encouragé, conseillé, et surtout qui il nous a donné beaucoup de son expérience, pour nous avoir accueilli, fait partager son bonne humeur et avoir fait preuve de patience et d'attention afin de nous donner les indications nécessaires au bon déroulement de notre stage de fin d'études.

Nous remercions monsieur Mr. HAMOUDA MOHAMED pour avoir accepté de diriger cette étude, la confiance qu'il nous a accordée ainsi que pour son assistance, ses précieux conseils et ses diverses corrections.

Nos remerciements vont particulièrement à Messieurs les membres du jury pour avoir accepté d'évaluer et de juger notre modeste travail. Je remercie enfin toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail

Dédicace

Je dédie ce modeste travail

A mon très cher père

Tu as toujours été pour moi un exemple du père respectueux, honnête, je tiens à honorer l'homme que tu es.

Je voudrais te remercier pour ton amour, ta générosité, ta compréhension... Aucune dédicace ne saura exprimer l'amour l'estime et le respect que j'ai toujours eu pour toi. Ce modeste travail est le fruit de tous les sacrifices que tu as déployé pour mon éducation et ma formation. Je t'aime papa et j'implore le tout-puissant pour qu'il t'accorde une bonne santé et une vie longue et heureuse.

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai pas te remercier comme il se doit, tu m'as donné la vie, la tendresse et le courage pour réussir. Tout ce que je peux t'offrir ne pourra pas exprimer l'amour et la reconnaissance que je te porte. Je t'aime mama et je t'offre ce modeste travail pour te remercier pour tes sacrifices et pour l'affection dont tu m'as toujours entouré.

A mon très cher frère Rachid et ma petite sœur adorée Sabrina

Puisse Dieu vous donne santé, bonheur, courage et surtout réussite.

A la mémoire de ma grand-mère

Qui a toujours souhaité que je réussisse dans mes études. J'aurais tant aimé que vous soyez présente, Je prie le Tout Puissant ALLAH pour le repos de son âme.

À notre encadreur Mr. Bey Mohamed pour les encouragements qu'il n'a cessé de nous apporter tout au long de notre stage.

A toutes ma famille, mes chères amies.

Sans oublier ma chère binôme Ikram pour son entente et sympathie, qui ma aidé et supporté dans les moments difficiles.

Enfin toute personne qui m'aime et que j'aime ...

Des fois, les mots ne suffisent pas pour exprimer tout le bien qu'on ressent ! Juste
MERCİ à vous !!!

Fairouz ♥

Dédicace

Je dédie ce travail à la mémoire de mon père ALLAH yarhmou, qui m'a toujours encouragé et motivé dans mes études.

A mon adorable maman, qui m'a soutenu et encouragé durant ces années d'études, qu'elle trouve ici le témoignage de ma profonde reconnaissance, MERCI MAMAN de m'avoir éclairé le chemin, je t'aime beaucoup, qu'ALLAH te garde pour moi, et pour nous.

A MERIEM mon âme sœur, merci d'être toujours avec moi, tu es ma compagne, ma moitié, mon soutien. JE T'AIME.

A ma chère sœur Karima pour son soutien et encouragements, tu es ma deuxième maman.

A mes chers sœurs et frères (Rachida, Raouf, Samir, Ilhem, Dani, Rahim et Imène) pour leurs amours, soutien et surtout pour tout l'encouragement durant les durs moments. A mes jolis neveux que j'aime, vous représenter la joie pour moi. A ma merveilleuse binôme Fairouz merci de m'avoir partagé les idées, de m'encouragé et d'avoir partagé tous ce parcours avec moi.

A mes chères copines (Yasmine, Fella, Samira), mes sœurs et amis, vous êtes le bonheur, l'espoir, la joie dans ma vie, je vous aime.

Ikram

Sommaire

Introduction générale.....	1
-----------------------------------	----------

Chapitre 1 : Etude Bibliographique

1.	Introduction.....	4
2.	Processus de production de pièces mécaniques.....	4
2.1	Définition des formes complexes.....	4
2.2	Processus de fabrication des pièces de formes complexes.....	5
2.2.1	CFAO	5
2.2.2	Naissance de l'idée.....	6
2.2.3	CAO	6
2.2.4	FAO.....	7
2.3	Généralités sur les fraiseuses numériques.....	7
2.3.1	Fraiseuses.....	7
2.4	Etude du format d'échange de données STL.....	8
2.4.1	Format IGES	9
2.4.2	Format STEP.....	9
2.4.3	Format VRML.....	9
2.4.4	Format STL.....	9
2.5	Post-processeur.....	12
2.5.1	CL-File.....	13
2.5.2	Définition du « G-Code »	13
2.6	Usinage.....	13
2.6.1	Outils d'usinage	14
2.6.2	Problèmes d'usinage.....	16
2.6.3	Stratégies d'usinage.....	16
3.	Conclusion.....	20

Chapitre 2 : Etude Conceptuelle

1.	Introduction.....	22
2.	Problématique et démarche de solution proposée	22
2.1	Lecture du modèle STL et calcul du brut.....	23
2.2	Création des cellules et enrichissement du modèle STL.....	24
2.3	Enrichissement des triangles du modèle STL.....	25
2.4	Affectation des points d'enrichissement aux cellules.....	26

2.5	Génération des plans et des contours de coupe	27
2.6	Gestion des bases de données d'outils.....	28
2.7	Positionnement et identification des outils optimums aux points de contact.....	29
2.7.1	Positionnement des outils.....	29
2.7.2	Création des enveloppes	31
2.7.3	Détermination des cellules en chevauchement	33
2.7.4	Vérification de l'intersection et de la collision	34
2.8	Association de la forme d'outils aux contours	40
2.9	Fixation du nombre d'outils par contour.....	42
2.10	Choix du nombre d'outils à utiliser pour le trajet.....	42
2.11	Génération de la trajectoire d'outils.....	43
2.12	Simulation du trajet d'outils.....	44
3.	Modélisation avec UML.....	44
3.1	Diagramme de cas d'utilisation.....	45
3.2	Diagramme d'activités.....	48
3.3	Diagramme de classes.....	51
4.	Conclusion.....	57

Chapitre 3 : Implémentation et validation

Introduction.....		59
1.	Présentation des outils utilisés.....	59
1.1	Présentation du langage C++.....	59
1.2	Présentation d'OpenGL	59
1.3	Embarcadero C++ Builder.....	60
2.	Présentation de l'environnement CFAO.....	60
3.	Présentation du module logiciel développé.....	61
3.1	Lecture du Modèle STL.....	62
3.2	Enrichissement du Modèle STL.....	63
3.3	Génération des Contours d'Usinage.....	64
3.4	Gestion des Bases des Données des Outils.....	65
3.5	Positionnement des Outils.....	66
3.6	Outils Optimums et Trajets d'Outils.....	68
3.7	Simulation des Trajets d'Outils.....	70

4.	Test et validation.....	72
4.1	Premier modèle STL.....	72
4.2	Deuxième modèle STL.....	79
5.	Conclusion.....	83
	Conclusion générale.....	84
	Références bibliographiques	86

CHAPITRE 1

Figure 1.1. Pièce mécanique complexe.....	5
Figure 1.2. Différentes formes locales des surfaces.	5
Figure 1.3. Processus de fabrication d'une pièce mécanique.	6
Figure 1.4. Différents types de fraiseuses.	8
Figure 1.5. Fraiseuses multiaxes.....	8
Figure 1.6. Modèle STL d'une pièce mécanique.....	9
Figure 1.7. Différentes entités géométriques	10
Figure 1.8. Paramètres d'un triangle.....	10
Figure 1.9. Structuration du modèle STL.	11
Figure 1.10. Syntaxe d'un fichier STL au format ASCII.....	11
Figure 1.11. Syntaxe d'un fichier STL au format binaire.....	12
Figure 1.12. Exemple d'un programme « G-Code ».	13
Figure 1.13. Phases d'usinage.	13
Figure 1.14. Différents types de fraises	14
Figure 1.15. Parties de l'outil.	14
Figure 1.16. Positionnement des différents outils.	15
Figure 1.17. Types d'interférences.....	16
Figure 1.18. Stratégie « Plans Parallèles ».....	17
Figure 1.19. Stratégie « Isoparamétrique ».	17
Figure 1.20. Stratégie « Z-Constant ».	18
Figure 1.21. Découpage d'un maillage triangulaire	18
Figure 1.22. Intersection d'un plan avec un triangle.	19
Figure 1.23. Création des contours.....	19
Figure 1.24. Mode One-Way.....	19
Figure 1.25. Mode Zig-Zag.....	20

CHAPITRE 2

Figure 2.1. Modèle STL et limites du brut.....	24
Figure 2.2. Création des Cellules.....	25
Figure 2.3. Enrichissement d'un triangle.....	26
Figure 2.4. Affectation d'un point à la cellule.	26
Figure 2.5. Affectation des points aux cellules	27
Figure 2.6. Plans de coupe	27
Figure 2.7. Types de contours.....	28
Figure 2.8. Positionnement des différentes formes d'outils.....	30
Figure 2.9. Enveloppes de la partie active des outils.....	32
Figure 2.10. Enveloppes de la partie corps des outils.....	33
Figure 2.11. Indices des cellules en chevauchement avec l'outil.....	34
Figure 2.12. Cellules de chevauchement	34
Figure 2.13. Cellules de chevauchement	34
Figure 2.14. Interférence avec partie active d'outil.....	35
Figure 2.15. Interférence avec le cylindre.....	35
Figure 2.16. Interférence avec la demi-sphère.	37
Figure 2.17. Interférence avec le tore	37
Figure 2.18. Partie active de l'outil torique.....	38
Figure 2.19. Absence d'interférence.....	38
Figure 2.20. Existence d'interférence	38
Figure 2.21. Existence d'interférence.....	39
Figure 2.22. Absence de collision.....	39
Figure 2.23. Existence de collision.....	40

Figure 2.24. Réaffectation des indices d'outils aux points de contact.....	42
Figure 2.25. Trajets d'outils	43
Figure 2.26. Points d'engagement et de dégagement	44
Figure 2.27. Diagramme de cas d'utilisation général	45
Figure 2.28. Diagramme de cas d'utilisation « Lire le modèle STL ».....	46
Figure 2.29. Diagramme de cas d'utilisation « Enrichir le modèle STL»	46
Figure 2.30. Diagramme de cas d'utilisation « Générer les plans et les contours».....	46
Figure 2.31. Diagramme de cas d'utilisation « Manipuler les bases de données des outils».....	47
Figure 2.32. Diagramme de cas d'utilisation « Positionner des outils».....	47
Figure 2.33. Diagramme de cas d'utilisation « Déterminer les outils optimaux et les trajets d'outils»	48
Figure 2.34. Diagramme de cas d'utilisation « Simuler les trajets d'outils».....	48
Figure 2.35. Diagramme d'activité général.....	49
Figure 2.36. Diagramme d'activité « Enrichir le modèle STL ».....	49
Figure 2.37. Diagramme d'activité «Déterminer outil optimum sommet ».....	50
Figure 2.38. Diagramme d'activité «Associer la forme aux contours».	50
Figure 2.39. Diagramme d'activité «Générer le trajet».	51
Figure 2.40. Diagramme d'activité « Simuler trajet ».	51
Figure 2.41. Diagramme de classes.	52
Figure 2.42. Classe « Objet Simulation Finale »	53
Figure 2.43. Classe « Trajet ».....	53
Figure 2.44. Classe « Plan de coupe»	53
Figure 2.45. Classe « Usinage trajet »	54
Figure 2.46. Classe « Outil»	54
Figure 2.47. Classe « Contour»	54

Figure 2.48. Classe « Sommer d’intersection»	55
Figure 2.49. Diagramme d’activité « Simuler trajet »	55
Figure 2.50. Classe « Coordonnées point».....	55
Figure 2.51. Classe « Triangle»	56
Figure 2.52. Classe « Modèle STL»	56
Figure 2.43. Classe « Brut»	56
Figure 2.54. Classe « Cellule élémentaire»	57
Figure 2.55. Classe « Couleur»	57

CHAPITRE 3

Figure 3.1. Logos des différents outils utilisés.....	59
Figure 3.2. Fenêtres principales de la plateforme logicielle.....	61
Figure 3.3. Lancement du module logiciel développé.....	61
Figure 3.4. Onglets du module logiciel développé.....	62
Figure 3.5. Onglet « Lecture du Modèle STL »	63
Figure 3.6. Onglet « Enrichissement du Modèle STL »	64
Figure 3.7. Onglet « Génération des Contours d’Usinage ».....	65
Figure 3.8. Onglet « Gestion des bases des Données des Outils ».....	66
Figure 3.9. Onglet « Positionnement des Outils »	68
Figure 3.10. Onglet « Outils Optimums et Trajets d’Outils ».....	70
Figure 3.11. Onglet « Simulation des Trajets d’Outils »	71
Figure 3.12. Premier modèle STL.....	72
Figure 3.13. Lecture du modèle STL.....	73
Figure 3.14. Brut du modèle STL.....	73
Figure 3.15. Création des cellules.....	74

Figure 3.16. Affectation des points aux cellules.....	74
Figure 3.17. Triangles enrichis et non enrichis.....	74
Figure 3.18. Plans et contours de la pièce.	75
Figure 3.19. Normales des points de contact d'un contour.	75
Figure 3.20. Cellules en chevauchement.....	75
Figure 3.21. Association de la forme d'outil aux contours.	76
Figure 3.22. Visualisation des outils pour le premier plan.....	77
Figure 3.23. Insertion des points d'engagement et de dégagement.....	78
Figure 3.24. Simulation des trajets d'outils.....	79
Figure 3.25. Enregistrement du fichier « CL-FILE ».	79
Figure 3.26. Deuxième modèle STL.....	79
Figure 3.27. Cellules en rendu.....	80
Figure 3.28. Contours du deuxième modèle.....	81
Figure 3.29. Temps d'usinage avec 3 outils sans optimisation.	82
Figure 3.30. Temps d'usinage avec 3 outils avec optimisation.....	82
Figure 3.31. Temps d'usinage avec 11 outils sans optimisation.	82
Figure 3.32. Temps d'usinage avec 11 outils avec optimisation.....	82
Figure 3.33. Simulation des mouvements d'outils.	83

CHAPITRE 2

Tableau 2.1. Extrémités des cellules.....24

Tableau 2.2. Description des paramètres.....28

CHAPITRE 3

Tableau 3.1. Paramètres du modèle STL.....73

Tableau 3.2. Positionnement des trois formes d'outils.76

Tableau 3.3. Limitation du nombre d'outils.....77

Tableau 3.4. Paramètres du modèle STL.....80

CHAPITRE 2

Organigramme 2.1. Organigramme de la démarche proposée.....23

Organigramme 2.2. Vérification de l'interférence.....36

Organigramme 2.3. Association de la forme d'outil aux contours.....41

Introduction générale

Présentation du sujet :

Durant la deuxième moitié du siècle passé, le monde a eu à faire face à de multiples transformations et grandes réalisations dans tous les domaines. Parmi elles, la demande de plus en plus accrue de nouveaux produits de bonnes qualités. La compétitivité de mise pousse les industriels à développer sans cesse de nouveaux produits et dans des temps de plus en plus courts. Le développement et la mise sur le marché d'un produit exigent à ce que les prix soient les plus bas possibles. De nos jours la réalisation des produits nouveaux exige des moyens informatiques car ils offrent l'efficacité, la fiabilité et la rapidité dans le traitement. Parmi les domaines qui exigent l'utilisation de ces moyens, on trouve le domaine de l'industrie, tels que l'industrie automobile, l'industrie aéronautique, l'industrie mécanique, ...etc.

Plusieurs industries ont été contraintes à trouver des solutions pour améliorer la productivité et la qualité des pièces fabriquées, faire le plus de pièces possibles, le plus rapidement possible, en réduisant au mieux les coûts et les défauts de production, particulièrement, l'industrie mécanique. Pour cela, il s'appuie sur la puissance de la CFAO « Conception et Fabrication Assistées par Ordinateur », qui apporte la flexibilité et la souplesse dans la conception et la fabrication des pièces de diverses complexités.

Les pièces de formes complexes (surfaces gauches) telles que les moules, les formes aérodynamiques, les carrosseries de voitures, les formes esthétiques, ...etc. sont devenues, par l'évolution du style et des techniques d'usinage, des pièces courantes de notre vie quotidienne. Comme toute pièce utilisée en mécanique, les pièces de formes complexes sont conçues dans le but d'assurer des fonctions inscrites dans le cahier des charges. Par conséquent, ces pièces doivent répondre à des exigences fonctionnelles et/ou de style. Ces pièces ne peuvent être usinées que sur des fraiseuses à commande numérique multiaxes (de 03-axes à 05-axes) en raison de leurs géométries très complexes par la prise en compte de plusieurs aspects tels que les modèles des surfaces, les stratégies d'usinages, les formes et les dimensions des outils, les paramètres d'usinages, ... etc.

Problématique :

Le but d'une opération d'usinage est d'enlever la matière sous forme de copeaux pour produire les pièces désirées. Les procédés d'usinage par enlèvement de matière sont nombreux, (fraisage, perçage, tournage, ...etc.). L'obtention de la forme finale d'une pièce de forme complexe requiert le passage par trois phases d'usinage : ébauchage, demi-finition et finition.

La demi-finition et la finition des pièces complexes passent par la détermination des formes et des dimensions des outils, des conditions de coupe et des stratégies d'usinage pour avoir un bon état de surface en un temps d'usinage minimum.

Les outils hémisphériques, toriques et cylindriques sont utilisés pour ces opérations. Le choix des outils est basé sur les propriétés géométriques des surfaces, les caractéristiques et les capacités des outils.

La stratégie la plus adaptée en demi-finition et en finition pour l'usinage des surfaces verticales ou inclinées et de cavités profondes est la stratégie « Z-Constant ». Cette stratégie est fiable et permet de garantir la continuité de l'usinage ce qui permet d'obtenir un bon état de surface et d'augmenter la durée de vie de l'outil.

La complexité géométrique des pièces complexes complique cette tâche, Le but recherché est une méthodologie pour optimiser les opérations de demi-finition et de finition des pièces complexes à partir de leurs modèles « STL » sur des fraiseuses numériques à 03-axes par la stratégie « Z-Constant ». Autrement dit, étant donné pièce complexe représentée par son modèle « STL », quelles sont les trajectoires d'outils, les formes et les dimensions des outils adéquats qui assurent un usinage de qualité, en un minimum de temps, continu sans interférences et sans collisions.

Objectif :

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

Dans ce projet, nous nous intéressons aux trois formes d'outils « Cylindriques, Hémisphériques et Toriques » lors de la demi-finition et de la finition des pièces complexes, définies par leurs modèles « STL », sur des fraiseuses numériques à 03-axes en considérant la stratégie d'usinage « Z-Constant ». Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle de production des pièces complexes développées par l'équipe « CFAO » du « CDTA », un module logiciel graphique et interactif permettant de choisir automatiquement et de combiner les formes et les dimensions des outils permettent d'éviter les interférences et les collisions, et de générer le trajet d'usinage qui représente la succession des points de positionnement des outils par rapport aux surfaces de la surface. Ce travail permettra d'améliorer la qualité d'usinage, d'augmenter la productivité et de réduire les coûts et les temps de génération des programmes d'usinage.

Structuration du mémoire :

Le présent mémoire est composé des chapitres suivants :

- Le premier chapitre est consacré à l'étude bibliographique du processus de « CFAO », du format d'échange de données « STL » et de la stratégie d'usinage « Z-Constant ».
- Le deuxième chapitre est réservé à la présentation de la solution proposée et à l'étude conceptuelle de notre application logicielle.
- Le dernier chapitre présente l'application logicielle développée, les tests et la validation des résultats.

Ce mémoire sera clôturé par une conclusion générale.

Chapitre 1 : Etude bibliographique

1 Introduction

L'usinage des pièces de haute précision devient de plus en plus concurrentiel grâce non seulement à la mondialisation qui entraîne l'ouverture d'autres marchés comme la Chine, l'Inde, etc., mais aussi aux besoins grandissants des clients. Les entreprises doivent fournir des pièces de haute qualité à moindre coût [1].

En mécanique industrielle, les pièces mécaniques de formes complexes sont largement utilisées dans divers industries (emballage, moules, automobile, aéronautique,...etc.). La fabrication de ces pièces se fait par enlèvement de matière (coupeaux), sur des machines-outils. Deux grandes familles de machines sont utilisées : les machines-outils conventionnelles et les machines-outils à commande numérique « MOCN ». L'usinage sur des machines conventionnelles consomme beaucoup de temps et demande une grande expérience des opérateurs pour arriver à une qualité d'une pièce juste moyenne en raison de la forme géométrique très complexe des pièces usinées [2]. Par contre, l'usinage sur des machines-outils à commande numérique (fraiseuses multiaxes de 03-axes à 05-axes), est plus précis et il ne demande pas beaucoup de temps mais plutôt une maîtrise de l'exploitation des capacités de la machine. Les pièces produites doivent répondre à des exigences fonctionnelles, ce qui pousse les entreprises à améliorer la qualité du produit toute en diminuant les temps de production et en réduisant les coûts. Donc, les entreprises doivent actualiser en permanence leurs méthodes et moyens de production.

Ce chapitre met en évidence cette problématique. Il est organisé en trois parties. Le processus de production de pièces mécaniques est décrit dans la première partie. La deuxième partie est consacrée à l'étude du format d'échange de données « *STL* ». La dernière partie est réservée aux fraiseuses numériques à 03-axes.

2 Processus de production de pièces mécaniques

La production des pièces mécaniques concerne de manière générale toutes les formes allant des formes élémentaires (parallélépipédiques) aux formes complexes (gauches). Dans la suite, les formes complexes sont détaillées.

2.1 Définition des formes complexes

Les pièces complexes dites aussi surfaces gauches sont des pièces qui ont des formes libres et difficiles à réaliser. Elles ont des surfaces qui ne peuvent pas être décomposées en éléments géométriques simples (point, cercle, sphère, droite, cylindre, plan ...) (Figure 1) [3]. Elles sont utilisées dans plusieurs domaines tels que les domaines aéronautiques, énergétiques et automobiles.



Figure 1. Pièce mécanique complexe [4].

Les pièces de forme complexe sont composées de plusieurs formes locales (régions) où chaque région peut prendre une des formes suivantes (Figure 2) :

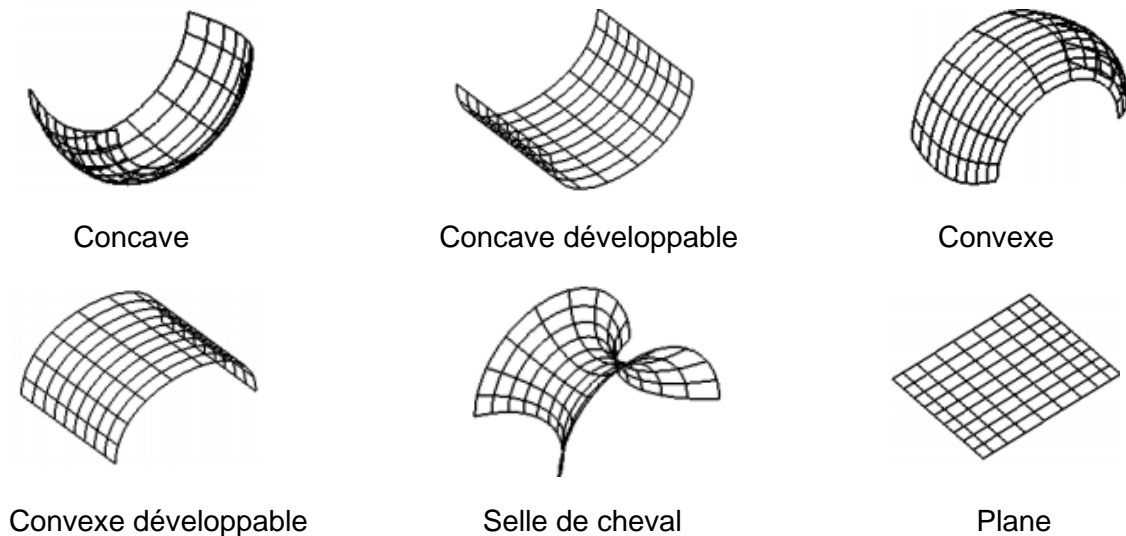


Figure 2. Différentes formes locales des surfaces [5].

2.2 Processus de fabrication des pièces de formes complexes

Le processus de fabrication est un ensemble d'activités (techniques) qui interagissent entre elles pour obtenir en sortie une pièce finie. L'obtention de la forme désirée nécessite parfois différents procédés de fabrication.

2.2.1 CFAO

La « **CFAO** », Conception et Fabrication Assistées par Ordinateur, est une activité qui regroupe deux processus : la conception assistée par ordinateur « **CAO** » et la fabrication assistée par ordinateur « **FAO** ». La « **CFAO** » permet de concevoir et de produire soit des pièces destinées à être utilisées individuellement soit à faire partie d'un assemblage pour former un système. L'idée générale de ce procédé est d'utiliser les capacités de l'ordinateur pour concevoir la pièce en trois (03) dimensions (**3D**). Ensuite, récupérer les informations contenues dans le modèle CAO pour les traiter dans la phase de fabrication afin de générer un fichier exploitable par les machines-outils à commande numérique. Les objectifs de la « **CFAO** » sont d'obtenir une extrême précision, de réaliser un gain de temps et de minimiser l'intervention

humaine [5]. Le processus de fabrication des pièces de formes complexes suit les étapes suivantes (Figure 3) :

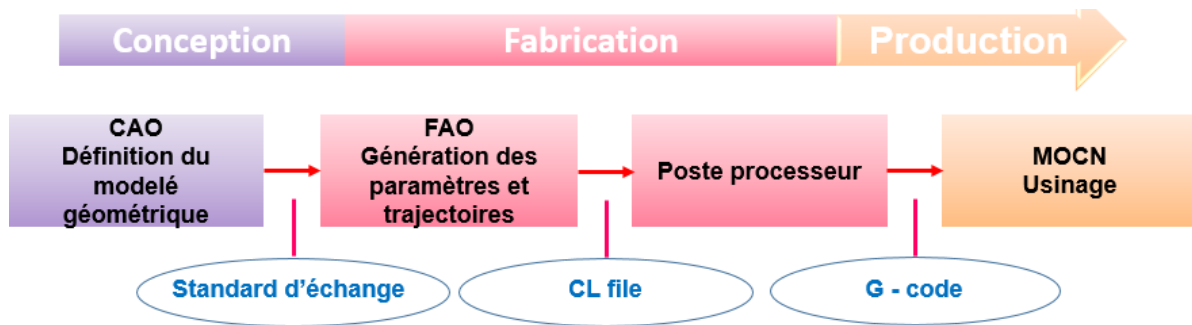


Figure 3. Processus de fabrication d'une pièce mécanique [6].

Ce processus est composé d'une activité de conception et d'une activité de fabrication. Il commence par une idée initiale relative à la forme générale de la pièce. Puis, la conception assistée par ordinateur « CAO » pour obtenir un modèle géométrique contenant des informations purement géométriques et l'envoyer au module de fabrication assistée par ordinateur « FAO ». Ce dernier génère un fichier exécutable par la machine d'usinage. Ce fichier contient les points de la trajectoire d'outils, les tâches d'usinage et les différents changements d'outils nécessaires pour l'obtention de la pièce conçue. Finalement, la pièce est usinée sur des machines-outils à commande numérique « MOCN ». Dans un système homogène où les modules de « CAO » et de « FAO » sont complètement intégrés dans un seul environnement, les informations circulent facilement et directement entre les deux modules. Dans le cas où ces deux modules ne sont pas intégrés dans le même environnement, la circulation des données nécessite l'utilisation des translateurs (formats d'échange de données).

2.2.2 Naissance de l'idée

L'être humain a tout le temps besoin de progression dans sa vie, ce qui le pousse à faire une analyse des besoins, une étude du marché, etc. Cette logique mène à une naissance d'idée et qui conduit à chercher des solutions.

2.2.3 CAO

Une fois l'idée murie, sa concrétisation nécessite le recours à des outils numériques puissants qui génèrent des modèles tridimensionnels riches en informations. Pour cela, des logiciels de « CAO » sont utilisés. La « CAO » est la contraction de Conception Assistée par Ordinateur, c'est le fait de dessiner sur un ordinateur une pièce ou une machine afin de la faire passer d'état d'idée à celui de quelque chose d'industrialisable [7]. Parmi ses avantages :

- Accroître la productivité et améliorer la qualité de la conception.
- Faciliter les modifications et les mises à jour.
- Valider par simulation des solutions adoptées avant leurs réalisations.

Le modèle CAO peut être ensuite exporté vers des logiciels de « **FAO** » afin de créer les programmes de pilotage des machines-outils à commande numérique.

2.2.4 FAO

La « **FAO** » qui signifie Fabrication Assistée par Ordinateur, désigne généralement l'utilisation d'applications logicielles de commande numérique « **CN** » pour générer un fichier contenant des instructions détaillées « **G-code** » exploitables par la commande numérique des machines-outils spécifiques équipées de calculateur appelées aussi machines « **CNC** ». Ce fichier décrit avec grande précision les mouvements que doivent exécuter les organes mobiles de la machine-outil en particulier les axes pour usiner une pièce donnée [7]. Les principaux avantages de la « **FAO** » sont :

- Assistance dans la création, la vérification et l'optimisation des programmes d'usinage pour une productivité maximale.
- Plus de précision ce qui permet aux ingénieurs de corriger les problèmes techniques avant la mise en œuvre.
- Une productivité accrue [8].

La combinaison entre « **CAO** » et « **FAO** » nous donne « **CFAO** ». Cette intégration permet de concevoir la forme d'une pièce mécanique puis de réaliser les diverses opérations d'usinage sur une machine-outil. La fabrication effective des produits, nécessite l'utilisation des outils spéciaux en fonction de la géométrie et des contraintes technologiques de la pièce à usiner.

2.3. Généralités sur les fraiseuses numériques :

2.3.1 Fraiseuses

Le fraisage « **CNC** », ou fraisage à commande numérique par ordinateur, est un processus d'usinage qui utilise des commandes informatisées et des outils de coupe multipoints rotatifs pour enlever progressivement la matière de la pièce et produire une pièce ou un produit conçu sur mesure. C'est un processus d'usinage mécanique où la matière est enlevée de la pièce par des moyens mécaniques, tels que les actions des outils de coupe de la fraiseuse. Elle est capable d'enlever de la matière d'une pièce initiale afin de la transformer en une autre pièce. Ils existent des fraiseuses de différents types selon l'orientation de la broche (verticales, horizontales et universelles) (Figure 4).



Fraiseuse universelle



Fraiseuse verticale



Fraiseuse horizontale

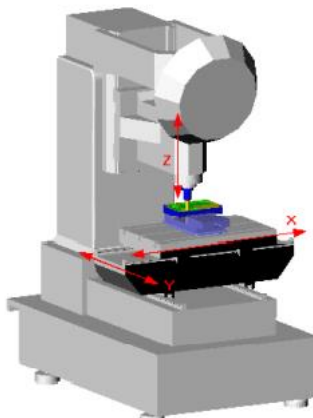
Figure 4. Différents types de fraiseuses [9].

Le fraisage présente plusieurs avantages tels qu'un rendement élevé, un bon fini de surface, une haute précision et une grande souplesse au niveau de la génération de différentes formes. La plupart des machines fonctionnent sur trois à cinq axes, où un axe représente un degré de liberté (mouvement), ce qui crée beaucoup plus de précision et de détails.

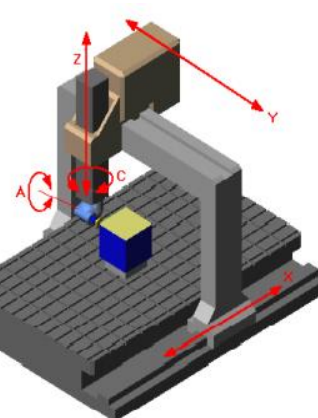
2.3.1.1 Fraiseuse à 03-axes

Une fraiseuse à trois axes peut être à broche verticale ou à broche horizontale. Les axes de la machine sont définis comme suit (Figure 5) :

- 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 .



Fraiseuse 03-axes



Fraiseuse 05-axes

Figure 5. Fraiseuses multiaxes. [10]

2.4. Etude du format d'échange de données STL

Les formats d'échange de données ou les standards d'échange de données sont des passerelles permettant de transmettre des données d'un système à un autre, tout en assurant la

fiabilité des informations contenues dans les données [6]. Plusieurs formats d'échange de données sont utilisés tels que **IGES**, **STEP**, **STL**, **DWG** etc. Le choix d'un format dépend du besoin, des contraintes de l'application et du domaine d'utilisation.

2.4.1 Format IGES

« **IGES** » (Initial Graphics Exchange Spécification) est un format de fichiers neutres permettant l'échange numérique d'informations à partir d'un logiciel de « **CAO** ». Il a été développé pour être compatible avec différents logiciels [11]. L'export vers un format « **IGES** » ne permet de sauvegarder que la « peau » extérieure du solide. Pour récupérer un solide, il faudra alors utiliser une fonction du modèleur permettant la génération d'un solide à partir de sa frontière surfacique.

2.4.2 Format STEP

Les fichiers « **STEP** » (Standard for the Exchange of Product data) représentent des objets 3D dans un logiciel de « **CAO** » et peuvent contenir des informations connexes, informations supplémentaires telles que des informations sur les matériaux et des tolérances [12]. Les fichiers « **STEP** » peuvent contenir des données sur l'ensemble du cycle de vie de la conception d'un produit, ainsi que les mêmes informations de définition du produit que « **IGES** », en plus des informations sur la topologie, les tolérances, les propriétés matérielles et d'autres données produits complexes. Le fichier « **STEP** » encapsule l'ensemble des modèles pièces et produits dans un unique fichier, ce qui facilite grandement les échanges. Son interprétation peut varier d'un éditeur à l'autre, ce qui entraîne parfois erreurs ou pertes d'informations sur les modèles.

2.4.3. Format VRML

Le fichier « **VRML** » (Virtual Reality Modeling Language) très répandu sur le web pour visualiser des scènes 3D dans un navigateur. Ce format peut aussi être utilisé pour diffuser largement un modèle solide CAO.

2.4.4. Format STL

Le fichier « **STL** » (Standard Tessellation Language) est un format de fichier natif de stéréolithographie créé par la société Systèmes **3D**. Un fichier « **STL** » décrit uniquement la géométrie des surfaces d'un objet tridimensionnel sans aucune représentation de couleur, de texture ou d'autres attributs du modèle CAO. Il stocke les informations du modèle **3D** sous la forme de triangles (Figure 6) [13].

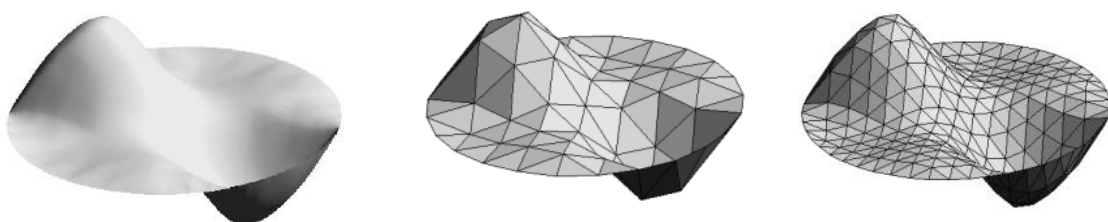


Figure 6. Modèle STL d'une pièce mécanique [14].

La description des surfaces est basée sur une représentation topologique. Les entités élémentaires, dans cette description sont les sommets, les arêtes et les facettes (Figure 7).

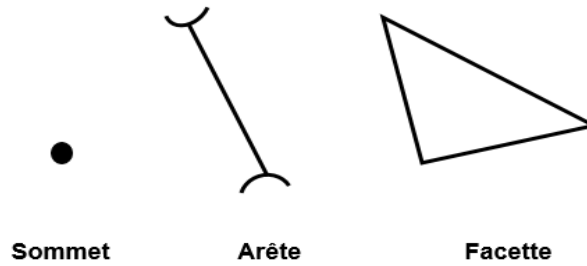


Figure 7. Différentes entités géométriques [15].

2.4.4.1 Lecture et Structuration du modèle STL

Le modèle « **STL** » représente la peau extérieure des objets qui se présentent sous la forme de nuage de points dont on connaît les coordonnées **x**, **y** et **z** aux formats « **ASCII** » ou « **Binaire** ». Les facettes sont des triangles. Donc, ce modèle est composé d'une liste de triangles. Chaque triangle est défini par les composantes de son vecteur normal unitaire **N** et par les coordonnées **X**, **Y** et **Z** de ses trois sommets **P1**, **P2** et **P3** [16] (Figure 8).

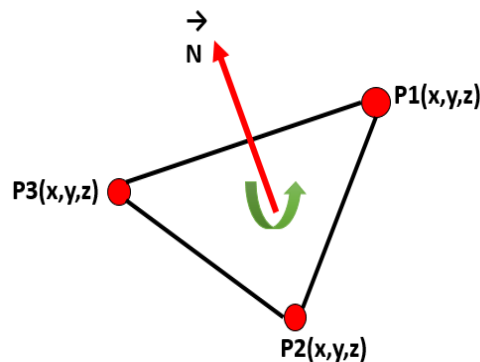


Figure 8. Paramètres d'un triangle. [15]

Pour éviter le problème de redondance et pour minimiser l'espace mémoire pour le stockage des données, deux listes sont utilisées. La première liste contient les sommets sans répétition tandis que la deuxième liste contient les triangles (Figure 9).

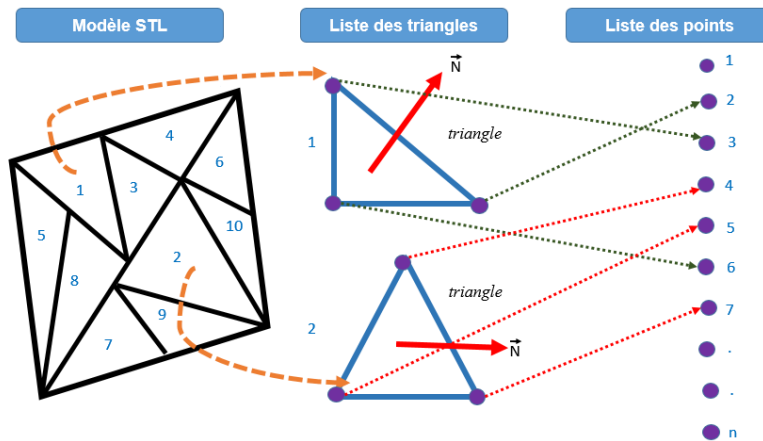


Figure 9. Structuration du modèle STL [15]

2.4.4.2 Stockage des informations dans un fichier STL

Deux formats sont utilisés pour le stockage des informations dans un fichier « **STL** » ; le format **ASCII** et le format binaire. Ces deux formats contiennent les mêmes informations sur le modèle. C'est-à-dire qu'il est possible de stocker le même modèle en tant que « **STL** » binaire ou en tant que « **STL** » **ASCII**.

➤ **Format de fichier ASCII STL :** le fichier **STL ASCII** est écrit comme suit (Figure 10) [6] :

- ✓ Solid au début du fichier.
- ✓ Le nom du modèle **3D**.
- ✓ Le fichier continu avec des informations sur les triangles :
 - Facette normale de composantes (**nx**, **ny**, **nz**) et sommet 1 de coordonnées (**v1x**, **v1y**, **v1z**) sommet 2 de coordonnées (**v2x**, **v2y**, **v2z**) et sommet 3 de coordonnées (**v3x**, **v3y**, **v3z**).
 - Endloop endfacet pour la fin des informations d'un triangle.
- ✓ End solid : le fichier se termine obligatoirement par cette ligne.

```

Solid name
  Facetnormal  $n_i$   $n_j$   $n_k$ 
    Outer loop
      Vertex v1 v1 v1
      Vertex v2 v2 v2
      Vertex v3 v3 v3
    Endloop
  Endfacet
End solid name
    
```

Figure 10. Syntaxe d'un fichier STL au format ASCII.

➤ **Format du fichier binaire :** le fichier « **STL** » binaire commence par un en-tête de 80 caractères, suivie par les informations sur les triangles. Chaque triangle est représenté par douze nombres à virgule flottante de 32 bits. Après chaque triangle, il y a une séquence de 2 octets appelée le « nombre d'octets d'attribut ». Ils sont utilisés pour coder des informations supplémentaires sur le triangle, ou ils sont mis à zéro et constitue un espace entre deux triangles (Figure 11) [6].

```

UINT8[80] – Header - 80 bytes
UINT32 – Number of triangles - 4 bytes
For each triangle - 50 bytes :
  REAL32[3] – Normal vector - 12 bytes
  REAL32[3] – Vertex 1 - 12 bytes
  REAL32[3] – Vertex 2 - 12 bytes
  REAL32[3] – Vertex 3 - 12 bytes
  UINT16 – Attribute byte count - 2 bytes
end

```

Figure 11. Syntaxe d'un fichier STL au format binaire.

2.4.4.3 Avantages et inconvénients du modèle « STL »

➤ **Avantage :**

- Il est universel, simple, très répandu dans l'industrie et pris en charge par presque tous les logiciels « CAO ».
- Facile mathématiquement.

➤ **Inconvénients :**

- Il ne peut pas stocker des informations telles que la couleur, le texte, etc.
- Les fichiers sont souvent très volumineux. De plus, chaque sommet est partagé par plusieurs triangles, ce qui augmente la taille du fichier.

2.5 Post-processeur

Les post-processeurs sont utilisés dans chaque système de « **FAO** » qui génère des données pour les machines-outils « **CNC** ». La fonction principale du post-processeur est de convertir les informations contenues dans le fichier CL-File tel que le format et la syntaxe en « G-Code » pour une machine spécifique. Un post-processeur est défini en fonction du format du « G-Code » qu'utilise la machine-outil « **MOCN** » pour laquelle il a été créé. Afin de pouvoir traiter correctement les données reçues sous forme de CL-File, l'information dont doit disposer le post-processeur inclut le type de « **MOCN** », sa configuration ainsi que les particularités du contrôleur de cette machine. Le type de « **MOCN** » indique au post-processeur sur quelle machine sera effectué l'usinage.

2.5.1. CL-File

Les grandes plateformes « **FAO** » et « **CFAO** » utilisent le CL-File comme format de représentation des trajectoires d'outils. Ce format contient les déplacements d'outils, nécessaires à l'usinage de la pièce. Toutefois, le langage d'entrée utilisé par la majorité des « **MOCN** » est différent du CL-File. Il convient alors d'utiliser un post-processeur, qui va traiter à nouveau cette information et créer un fichier de données appelé « **G-Code** » [17].

2.5.2 Définition du « G-Code »

C'est un langage qui se compose d'un ensemble d'instructions décrivant les différents mouvements de la machine nécessaire à la fabrication d'une pièce tout en contrôlant sa vitesse et sa position. Le « G-Code » est constitué d'instructions alphanumériques (chaînes de lettres et de chiffres) (Figure 12) [18].

```
N148 G1 Y1.12
N150 G3 X2.245 Y1.37 R.25
N152 G1 X.755
N154 G3 X.505 Y1.12 R.25
N156 G1 Y.63
N158 G3 X.755 Y.38 R.25
N160 G1 X2.245
```

Figure 12. Exemple d'un programme « G-Code » [18].

2.6 Usinage

L'usinage entre dans la gamme de fabrication des pièces mécaniques. Il consiste à réaliser des pièces par enlèvement de matière en respectant, l'état de surface, les tolérances de formes et la géométrie spécifiée par le dessin et le cahier de charges, pour donner à une matière brute une forme précise. Le processus d'usinage de ces pièces passe par trois opérations (Figure 13) :

- Ebauchage : pour enlever le maximum de matière.
- Demi-finition : pour s'approcher de la forme finale.
- Finition : pour obtenir la forme finale avec un bon état de surface [19].



Figure 13. Phases d'usinage [20].

2.6.1 Outils d'usinage

La fraise ou bien l'outil de fraisage, comporte plusieurs arrêtes de coupe dont chacune enlève une certaine quantité de métal sous forme de copeaux [21]. Les principaux types de fraises utilisées dans l'usinage des pièces avec des surfaces de formes libres (surfaces gauches) sont les suivants (Figure 14) [10, 22] :

- **Fraise cylindrique** : elle convient aux travaux d'ébauche et de demi-finition grâce à leurs robustesses et leurs faces planes.
- **Fraise hémisphérique** : est utilisée en général pour usiner les zones concaves. Elle est très utilisée vu sa simplicité et sa géométrie et elle convient pour la finition de toutes les matières.
- **Fraise torique** : sont des outils intermédiaires entre les fraises cylindriques et les fraises hémisphériques.



Figure 14. Différents types de fraises [22].

2.6.1.1 Différentes parties de l'outil

Les principales parties d'un outil de coupe sont les suivantes (Figure 15) :

- **Corps** : il assure la liaison de l'outil avec le porte-outil. Sa forme peut être prismatique ou cylindrique.
- **Queue** : élément qui permet la liaison de l'outil avec le porte-outil. Sa forme peut être cylindrique ou conique.
- **Partie active** : la partie qui agit directement sur la pièce pour provoquer un enlèvement de matière. Elle est caractérisée par sa forme [23].

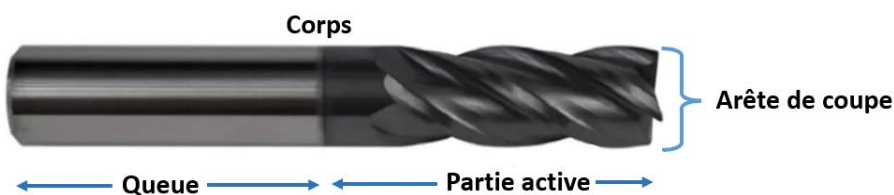


Figure 15. Parties de l'outil [23].

2.6.1.2 Positionnement de l'outil

L'usinage des pièces avec des surfaces gauches, nécessite le positionnement de l'outil par rapport aux surfaces à usiner tout le long du trajet d'usinage. Pour l'opération de finition, l'outil, quelles que soient sa forme et ses dimensions, doit être constamment tangent à la surface en chaque point de contact (Figure 16) [15].

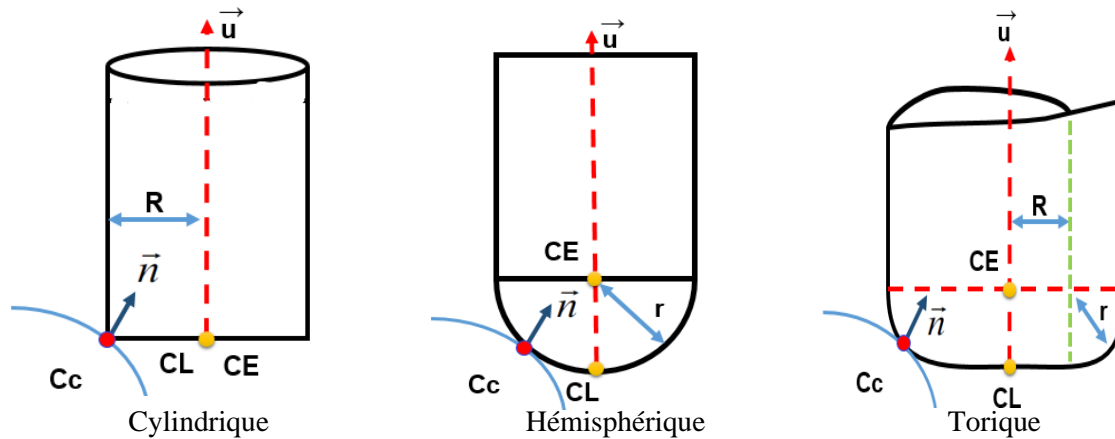


Figure 16. Positionnement des différents outils. [20]

Où

- C_c : point de contact entre l'outil et la surface,
- C_E : point centre de l'outil,
- C_L : point extrémité de l'outil,
- \vec{n} : vecteur normal unitaire à la surface au point de contact,
- \vec{u} : vecteur orientation de l'axe de l'outil,
- r : rayon de l'outil hémisphérique et le petit rayon de l'outil torique,
- R : grand rayon de l'outil torique et le rayon de l'outil cylindrique.

Les équations de positionnement d'un outil hémisphérique sont [15] :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} \end{cases} \quad (1.1)$$

Les équations de positionnement d'un outil cylindrique sont [15] :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} = \overrightarrow{OC_C} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|} \end{cases} \quad (1.2)$$

Les équations de positionnement d'un outil torique sont [15] :

$$\begin{cases} \overline{OC_E} = \overline{OC_C} + r\vec{n} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|} \\ \overline{OC_L} = \overline{OC_E} - r\vec{u} = \overline{OC_C} + r\vec{n} - r\vec{u} + R \frac{\vec{k} \wedge \vec{u}}{\|\vec{k} \wedge \vec{u}\|} \end{cases} \quad (1.3)$$

Avec :

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

2.6.2 Problèmes d'usinage

Les formes géométriques complexes des surfaces gauches peuvent générer des problèmes d'interférences et de collisions. Pour un point de contact outil/pièce donné, toute la difficulté consiste à déterminer l'orientation efficace de l'axe de l'outil. Donc, la maîtrise des interférences outil/pièce lors de la génération de trajectoires d'outils est une condition nécessaire pour garantir la conformité de la pièce en termes de géométrie et de contraintes fonctionnelles. Les problèmes d'interférences sont en général classés selon trois types (Figure 17) :

- **Interférences locales** : elles traduisent un enlèvement de matière excessif par la partie active de l'outil.
- **Interférences vers l'arrière** : elles sont des pénétrations de l'arrière de l'outil dans la surface à usiner.
- **Collisions** : les interférences globales (collisions) représentent des collisions entre l'ensemble {corps d'outil, porte outil, broche} et l'ensemble {pièce, porte Pièce, éléments de structure...} [10].

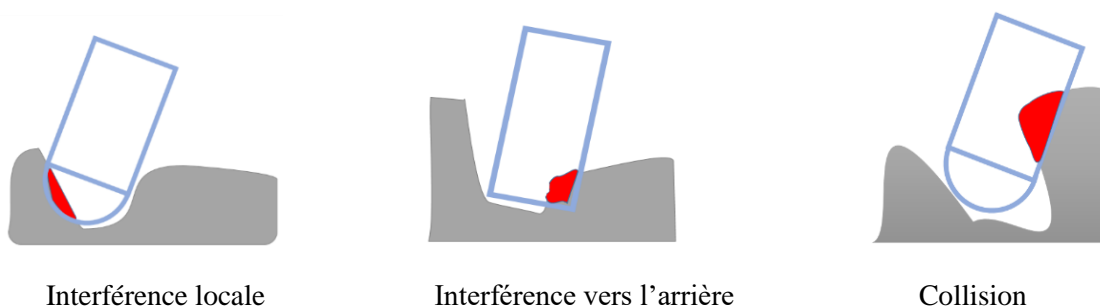


Figure 17. Types d'interférences. [20]

2.6.3 Stratégies d'usinage

Ils existent de nombreuses façons pour générer un parcours d'outils suivant des différentes stratégies d'usinage. Une stratégie d'usinage est définie comme étant une technique pour usiner

une forme, ou encore le mouvement de l'outil dans une pièce afin d'obtenir une série d'opérations pour réaliser une forme donnée [24]. L'usinage des pièces de formes gauches est réalisé en fraisage par une succession de passes juxtaposées selon des critères de tolérance de flèche et de hauteur de crête. L'objectif est de minimiser le temps d'usinage tout en garantissant un niveau de qualité attendu sur les pièces. Donc, il est essentiel de choisir les stratégies d'usinage optimales vis à vis des performances et du comportement du moyen de fabrication. Les stratégies les plus utilisées sont les suivantes :

➤ **Plans Parallèles** : elle s'appuie sur des trajectoires d'outil résultantes de l'intersection entre les surfaces à usiner et des plans verticaux parallèles entre eux (Figure 18). Pour cette stratégie, la planification des trajectoires d'outils est faite sur l'espace 3D et le plan parallèle est bon pour l'usinage des zones plates.

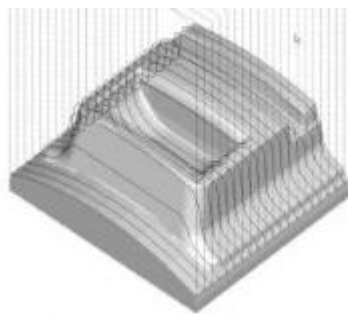


Figure 18. Stratégie « Plans Parallèles » [25]

➤ **Isoparamétrique** : elle s'appuie sur le plan paramétrique de la surface pour générer le trajet d'outil avec plusieurs modes (Figure 19).

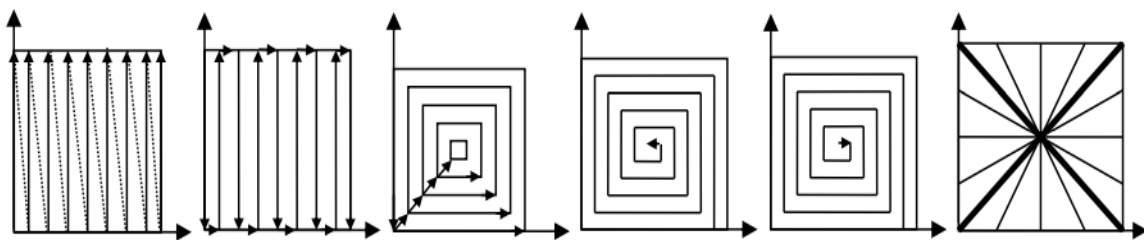


Figure 19. Stratégie « Isoparamétrique » [10].

➤ **Z-Constant** : elle s'appuie sur des trajectoires d'outil résultantes de l'intersection entre les surfaces à usiner et un ensemble de plans horizontaux parallèles entre eux. L'outil commence en haut du brut et enlève la matière en gardant sa position Z constante et se déplacent uniquement dans le plan XY. Une fois l'usinage est terminé sur ce plan, l'outil se positionne sur la seconde position selon l'axe Z et se déplace sur le plan XY pour balayer la surface. L'enlèvement de matière est répété jusqu'à ce que le niveau Z le plus bas soit atteint [26]. En industrie manufacturière, la stratégie « Z-Constant » est la plus adaptée pour la demi-finition et la finition [25]. Elle est plus appropriée pour l'usinage de surfaces verticales ou proche de la

verticale « **inclinées** » et de cavités profondes. Cette stratégie permet de réduire les longs engagements et dégagements d'outils.

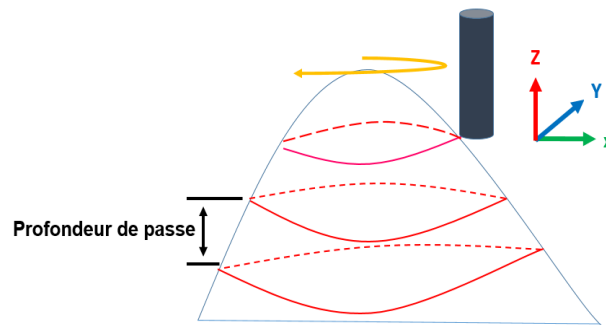


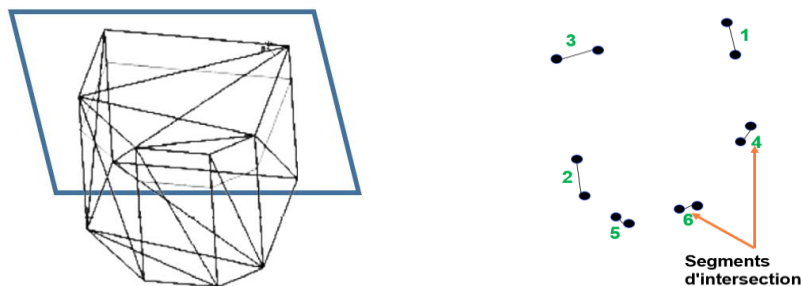
Figure 20. Stratégie « Z-Constant » [6].

Cette stratégie comprend deux parties :

Partie 1 :

Calcul des contours (éléments de trajectoire d'outils) en découpant la surface avec des plans horizontaux. Habituellement, le tranchage est effectué par un certain nombre de plans horizontaux également espacés, et la distance entre deux plans de tranchage consécutifs est souvent appelée « pas de plan » ou « profondeur de passe » [25]. La procédure de tranchage peut être divisée en deux parties :

- 1- Trouver des segments d'intersection des triangles et des plans horizontaux (Figure 21).



Maillage triangulaire et plan horizontal Segments d'intersection sans relation

Figure 21. Découpage d'un maillage triangulaire [25].

La Figure 22 illustre les différents cas d'intersection d'un triangle avec un plan.

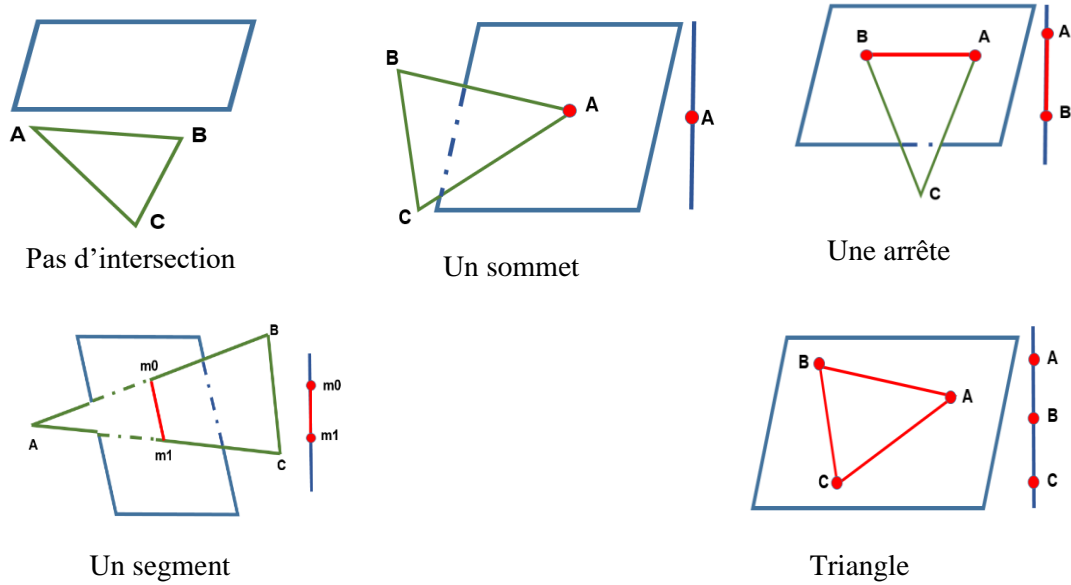


Figure 22. Intersection d'un plan avec un triangle [21].

2- Création des contours à partir des segments d'intersection.

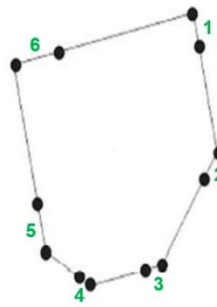


Figure 23. Création des contours [25].

Partie 2 :

Générer une trajectoire d'outils en reliant les contours d'usinage. La génération des trajectoires peut être menée selon deux modes :

❖ **One-Way** : pour ce mode, l'usinage de la surface est réalisé avec le trajet d'outils sur un ensemble de contours. C'est-à-dire l'outil usine le premier contour qui a le plus grand Z, puis il quitte la surface (dégagement d'outil) et passe au début du contour suivant (engagement d'outil) pour continuer l'usinage. ce processus est répété jusqu'à l'usinage de la pièce entière (Figure 24).

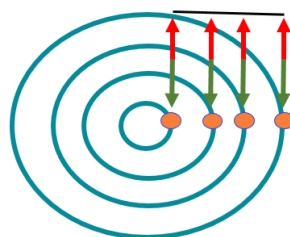


Figure 24. Mode One-Way [27]

❖ **Zig-Zag** : l'outil dans ce mode usine le premier contour, ensuite sans quitter la surface il passe au contour suivant et reprend l'usinage. L'avantage de ce mode est la réduction du temps d'usinage en évitant les retraits hors matière (Figure 25).

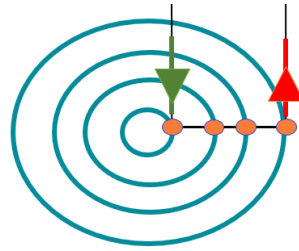


Figure 25. Mode Zig-Zag [27].

Dans notre projet, c'est la stratégie « Z-Constant » qui est adoptée pour la demi-finition et la finition des pièces de formes complexes.

3. Conclusion

Nous avons présenté dans cette recherche bibliographique le processus de production des pièces avec des surfaces gauches. Nous avons montré que ce processus se divise en deux activités, la première est consacrée à la conception et la deuxième à la fabrication. Nous avons mis en évidence l'hétérogénéité des informations manipulées dans chaque activité, ce qui nécessite l'utilisation des translateurs pour le transfert de données tels que format « STL ». En fin, nous avons terminé ce chapitre par des généralités sur le processus d'usinage pièces avec des surfaces gauches.

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

Chapitre 2 : Etude Conceptuelle

1. Introduction :

Le recours à la modélisation est une étape incontournable dans le processus de développement de logiciels. Elle a pour rôle de cerner les problèmes : les identifier, trouver leurs solutions, schématiser ces dernières, puis enfin préparer le terrain d'action [1].

L'objectif de notre travail est la conception et le développement d'une application logicielle permettant la génération des trajectoires d'outils pour l'usinage en demi-finition et en finition des pièces complexes par la stratégie « Z –Constant », sur des fraiseuses numériques à 03-axes, par la combinaison des différentes formes d'outils (cylindriques, toriques et hémisphériques) évitant les problèmes d'usinage (interférences et collisions) tout en assurant une bonne qualité des surfaces produites et un minimum de temps d'usinage.

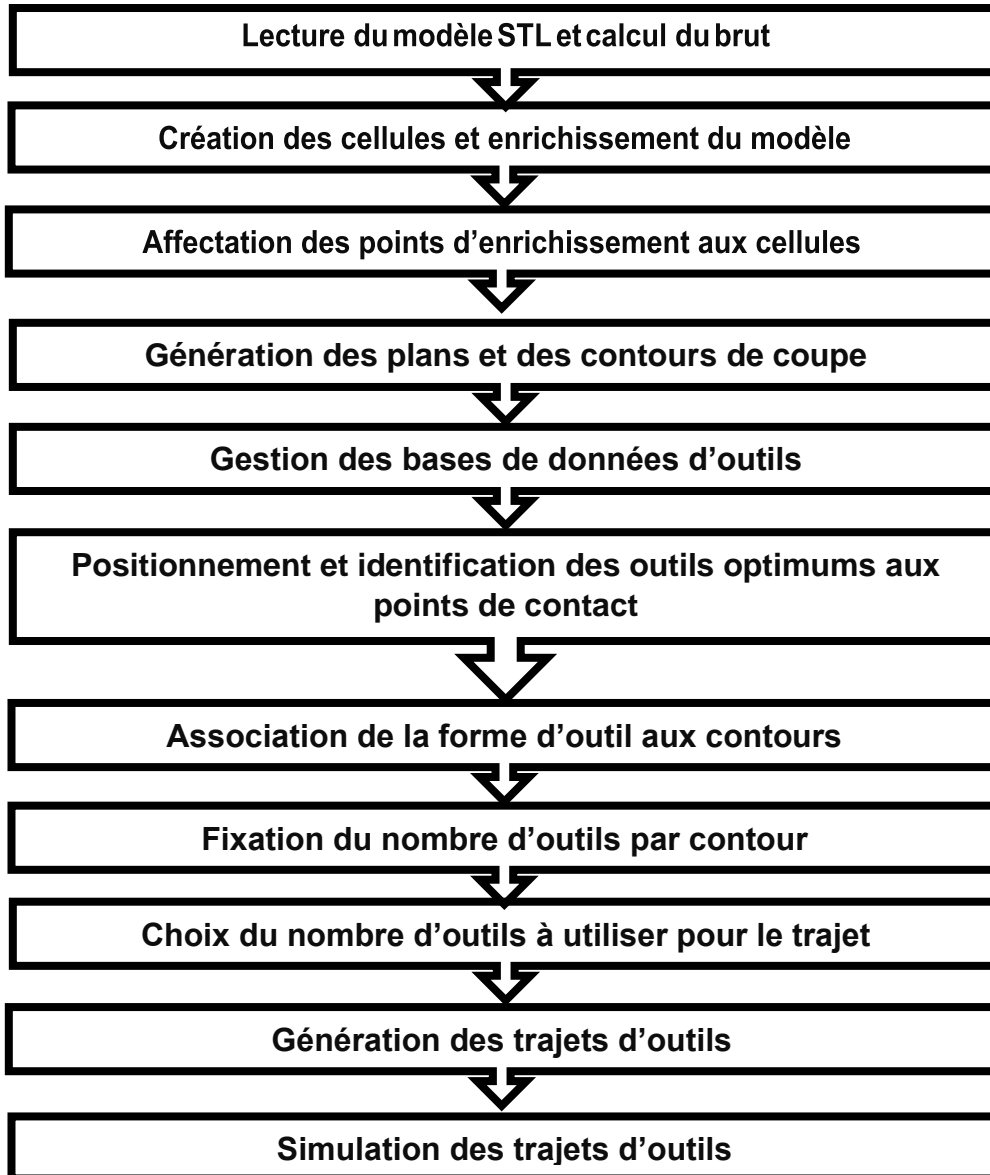
Dans ce qui suit, les différentes étapes de la solution proposée et la conception de l'application seront présentées.

2. Problématique et démarche de solution proposée :

La décomposition d'un grand problème en parties cohérentes et simples à résoudre est une étape logique de résolution d'un problème informatique donné. Pour atteindre l'objectif visé, plusieurs problèmes doivent être résolus :

- Comment lire et structurer le modèle STL ?
- Comment enrichir le modèle STL ?
- Comment lire et structurer les trajets d'outils ?
- Comment positionner les différentes formes d'outils ?
- Comment sélectionner les outils optimums ?
- Comment procéder pour accélérer les calculs ?
- Comment combiner ces outils ?
- Comment limiter le nombre d'outils à utiliser ?
- Comment chaîner les trajets d'outils ?
- Comment accélérer la simulation ?
- Comment vérifier virtuellement les trajets d'outils ?

La solution à proposer doit répondre à toutes ses questions. Donc, le problème à résoudre est décomposé en une suite d'étapes successives pour ensuite donner une solution adéquate. Dans notre cas, les différentes étapes de la solution sont présentées dans l'organigramme général donné par l'**Organigramme 2.1**.



Organigramme 2.1. Organigramme de la démarche proposée.

2.1. Lecture du modèle STL et calcul du brut

Cette étape consiste à analyser le fichier STL qui décrit la pièce à usiner. Cette analyse consiste à vérifier son extension et sa syntaxe. Par la suite, calcul des points limites et des dimensions du brut qui englobe le modèle **STL** dans une enveloppe parallélépipédique de faces parallèles aux axes **X**, **Y** et **Z** (Figure 2.1).

Les dimensions du brut sont calculées comme suit :

$$\begin{cases} \textit{Longueur} = X_{max} - X_{min} \\ \textit{Largeur} = Y_{max} - Y_{min} \\ \textit{Hauteur} = Z_{max} - Z_{min} \end{cases} \quad (2.1)$$

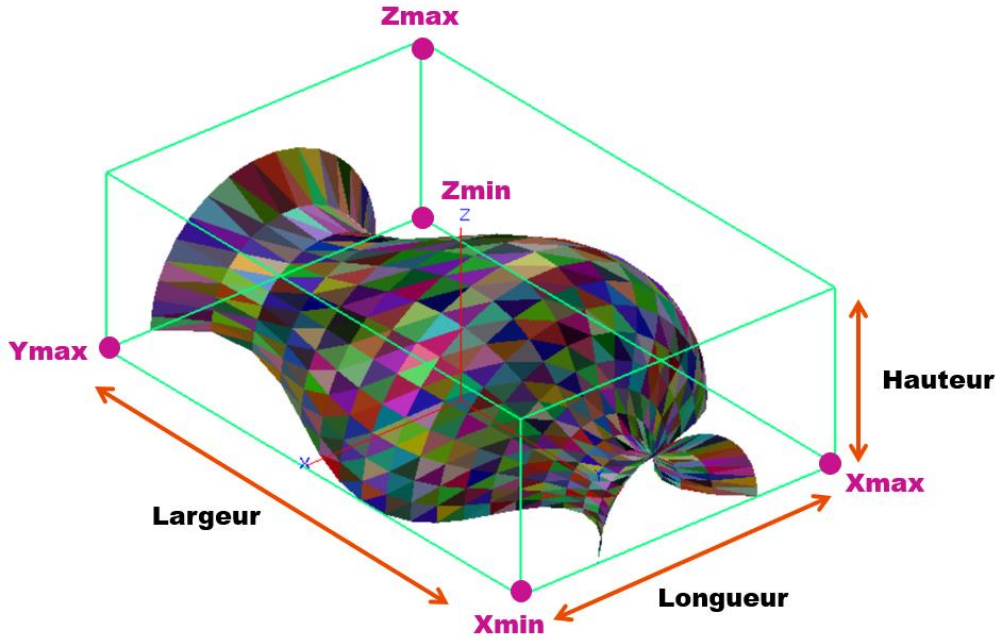


Figure 2.1. Modèle STL et limites du brut.

2.2. Création des cellules et enrichissement du modèle STL

La détection des problèmes d’usinage (interférences et collisions) de l’outil avec le modèle STL nécessite la vérification d’intersection de l’outil avec les triangles du modèle STL en chaque point de positionnement de l’outil le long du trajet d’usinage. En raison du grand nombre de triangles, cette étape est laborieuse et requiert un important temps de calcul. Afin de le minimiser, l’approche proposée consiste à subdiviser le brut en blocs parallélépipédiques (cellules) de mêmes dimensions. La création de ces cellules nécessite la spécification du nombre de cellules suivant les axes X, Y et Z. Après la création des cellules, les dimensions de chaque cellule sont calculées par :

$$\begin{cases} \text{Longueur_Cellule} = (\text{Xmax_Brut} - \text{Xmin_Brut}) / \text{Nombre_Cellule_x} \\ \text{Largeur_Cellule} = (\text{Ymax_Brut} - \text{Ymin_Brut}) / \text{Nombre_Cellule_y} \\ \text{Hauteur_Cellule} = (\text{Zmax_Brut} - \text{Zmin_Brut}) / \text{Nombre_Cellule_z} \end{cases} \quad (2.2)$$

Les calculs des extrémités des cellules sont résumés dans le Tableau suivant :

Tableau 2.1. Extrémités des cellules.

	Extrémités minimales	Extrémités maximales
Pour $i = 0$ à Nombre_Cellule_x	$X_{Min} = X_{min_{Brut}} + i * \text{Longueur}_{Cellule}$	$X_{max} = X_{max_{Brut}} + \text{Longueur}_{Cellule}$
Pour $j = 0$ à Nombre_Cellule_y	$Y_{Min} = Y_{min_{Brut}} + i * \text{Largeur}_{Cellule}$	$Y_{max} = Y_{max_{Brut}} + \text{Largeur}_{Cellule}$
Pour $k = 0$ à Nombre_Cellule_z	$Z_{Min} = Z_{min_{Brut}} + i * \text{Hauteur}_{Cellule}$	$Z_{max} = Z_{max_{Brut}} + \text{Hauteur}_{Cellule}$

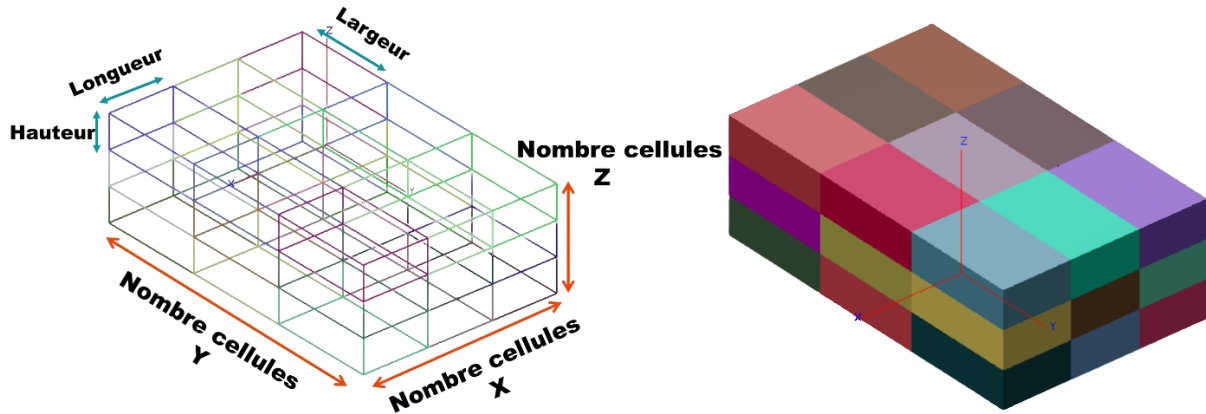


Figure 2.2. Création des cellules.

2.3. Enrichissement des triangles du modèle STL

Dans le cas où le modèle STL n'est pas très fin (le nombre de sommets est faible et/ou la taille des triangles est grande), les problèmes de collisions et d'interférences peuvent ne pas être détectés. Afin d'éviter ces problèmes et de pouvoir les détecter efficacement, il est indispensable d'augmenter le nombre de points pour couvrir tout le modèle STL. Notre approche consiste à insérer, pour chaque triangle, de nouveaux points d'une façon aléatoire. Le nombre de points à insérer est déterminé en fonction de l'aire du triangle et de la densité des points spécifiée par l'utilisateur.

Avant l'enrichissement, il faut calculer l'aire de chaque triangle. Pour un triangle défini par ses sommets $A (A_x, A_y, A_z)$, $B (B_x, B_y, B_z)$ et $C (C_x, C_y, C_z)$, son aire est calculée par la formule suivante (Figure 2.3) :

$$Aire = \frac{1}{2} \sqrt{\left(\det \begin{pmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{pmatrix} \right)^2 + \left(\det \begin{pmatrix} A_y & B_y & C_y \\ A_z & B_z & C_z \\ 1 & 1 & 1 \end{pmatrix} \right)^2 + \left(\det \begin{pmatrix} A_z & B_z & C_z \\ A_x & B_x & C_x \\ 1 & 1 & 1 \end{pmatrix} \right)^2} \quad (2.3)$$

L'insertion des points ne concerne que les triangles dont leurs aires sont supérieures à une aire (seuil) spécifiée par l'utilisateur. Cette valeur doit être comprise entre l'aire minimum et l'aire maximum des triangles.

Les coordonnées du point $P(P_x, P_y, P_z)$ généré aléatoirement à l'intérieur du triangle ABC sont données par les relations suivantes :

$$\begin{cases} P_x = ((1 - \sqrt{r_1}) * A_x + (1 - \sqrt{r_2}) * B_x + (\sqrt{r_1} * \sqrt{r_2}) * C_x \\ P_y = ((1 - \sqrt{r_1}) * A_y + (1 - \sqrt{r_2}) * B_y + (\sqrt{r_1} * \sqrt{r_2}) * C_y \\ P_z = ((1 - \sqrt{r_1}) * A_z + (1 - \sqrt{r_2}) * B_z + (\sqrt{r_1} * \sqrt{r_2}) * C_z \end{cases} \quad (2.4)$$

Où :

r_1, r_2 : nombres réels aléatoires dans l'intervalle $[0, 1]$.

Une fois le traitement terminé, chaque triangle aura sa liste des points générés aléatoirement. Par la suite, chaque point ajouté est affecté à la cellule qui lui correspond.

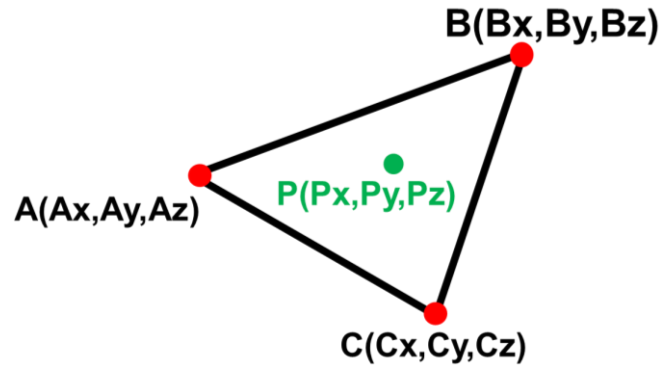


Figure 2.3. Enrichissement d'un triangle.

2.4. Affectation des points d'enrichissement aux cellules

Après la création des cellules et l'enrichissement des triangles par des points, chaque point du triangle doit être affecté à la cellule correspondante. Les indices de cette dernière sont calculés en se basant sur les coordonnées du point P et les coordonnées minimales du brut de la manière suivante (Figure 2.4 et Figure 2.5) :

$$\begin{cases} \text{Indice}_x = \frac{(P_x - X_{\min_{\text{Brut}}})}{\text{Longueur}_{\text{Cellule}}} \\ \text{Indice}_y = \frac{(P_y - Y_{\min_{\text{Brut}}})}{\text{Largeur}_{\text{Cellule}}} \\ \text{Indice}_z = \frac{(P_z - Z_{\min_{\text{Brut}}})}{\text{Hauteur}_{\text{Cellule}}} \end{cases} \quad (2.5)$$

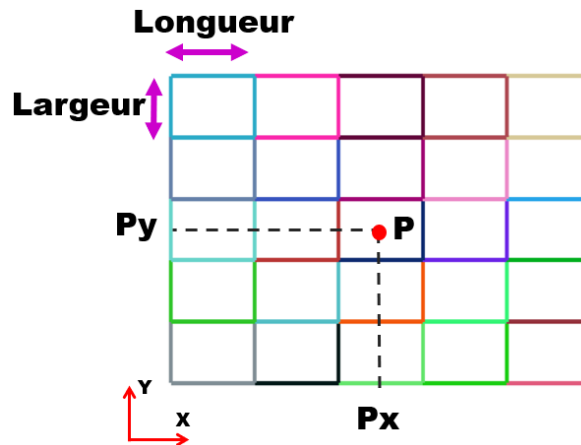


Figure 2.4. Affectation d'un point à la cellule.

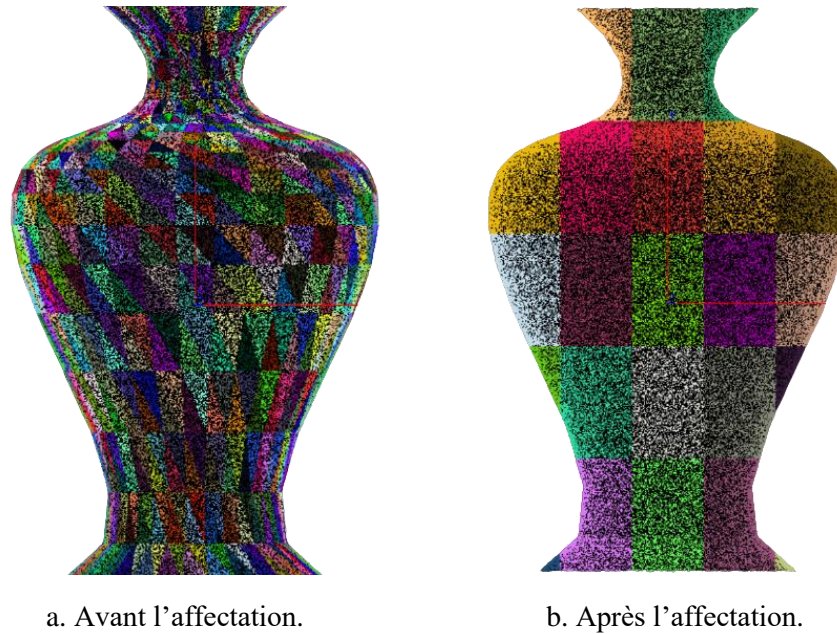


Figure 2.5. Affection des points aux cellules.

2.5. Génération des plans et des contours de coupe

La stratégie d'usinage adoptée « Z-Constant » est basée sur la création des plans horizontaux de coupe où la distance entre deux plans consécutifs est appelé « pas de plan » ou « profondeur de passe » fixée par l'utilisateur en fonction de la précision désirée (Figure 2.6). Le nombre de plans dépend de cette dernière et des dimensions du brut de la pièce.

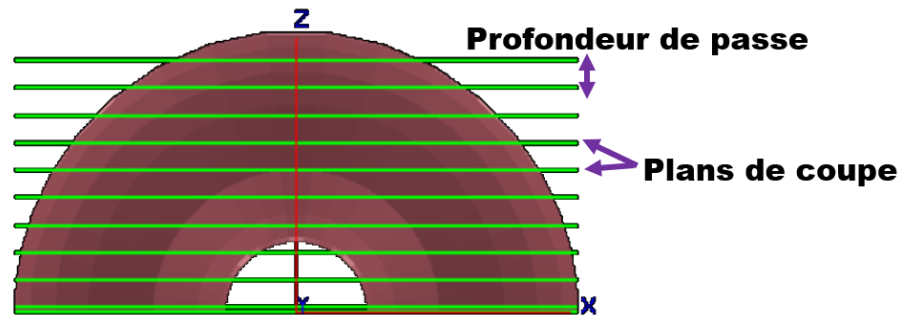


Figure 2.6. Plans de coupe.

L'intersection entre ces plans de coupe et les triangles du modèle STL génère des segments non ordonnés qui sont reliés entre eux selon un ordre bien déterminé pour former une liste de segments raccordée. Cette liste est appelée « Contour ». En fonction du modèle STL de la pièce à usiner, un plan de coupe peut contenir un ou plusieurs contours pouvant être fermés ou ouverts (Figure 2.7).

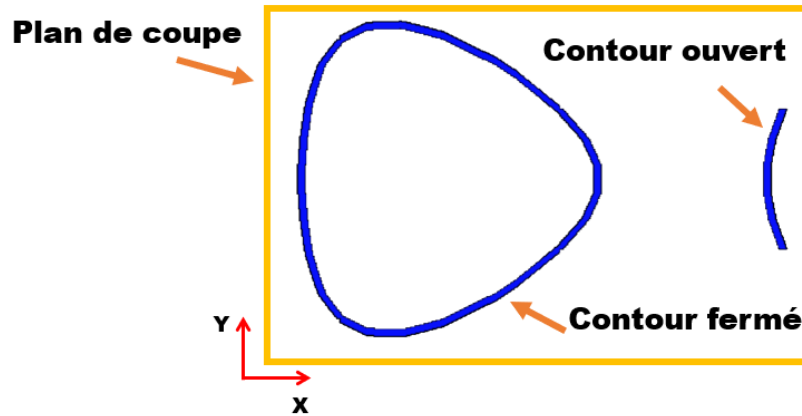


Figure 2.7. Types de contours.

Il est à noter que la génération des plans et des contours de coupe a été réalisée lors d'un projet mené auparavant. Nous avons seulement récupéré les plans et les contours.

2.6. Gestion des bases de données d'outils

L'usinage des pièces complexes nécessite l'utilisation de plusieurs formes d'outils. Dans notre cas, trois formes sont utilisées à savoir : cylindrique, torique et hémisphérique. Pour cela, nous avons créé pour chaque forme une base de données composée d'une seule table qui contient toutes les spécifications (géométriques et technologiques) qui sont présentées par le tableau suivant :

Tableau 2.2. Description des paramètres.

	Paramètre	Description
Paramètres géométriques	Rayon	Segment qui relie le centre avec sa circonférence.
	Longueur	Longueur de l'outil (corps d'outil + partie active).
	Longueur partie active	Longueur de la partie concernée par l'enlèvement de matière.
	Rayon de tore	Le rayon du tore concerne seulement l'outil torique.
Paramètres technologiques	Vitesse de coupe (V_c)	Exprimée en mm/min est la vitesse instantanée du mouvement de coupe du point considéré de l'arête par rapport à la pièce.
	Z	Nombre de dents
	Avance par dent (F_z)	Exprimée en mm/dent est la distance linéaire parcourue par une dent de l'outil. Elle représente aussi la distance couverte par la pénétration de deux dents successives dans la pièce.

A partir de ces données, il est possible de calculer les paramètres technologiques à spécifier dans le programme d'usinage « G-Code » à savoir la vitesse de broche N et la vitesse d'avance V_f .

Vitesse de broche N :

Elle est exprimée en tr/min, c'est-à-dire le nombre de tours que la fraise effectue par minute sur la broche. Elle est calculée par la formule suivante :

$$N = \frac{1000 * V_C}{2R * \pi} \tag{2.6}$$

Vitesse d'avance V_f :

Elle est exprimée en mm/min, c'est-à-dire la vitesse instantanée du mouvement d'avance du point considéré de l'arête par rapport à la pièce. Elle est calculée avec la formule suivante :

$$V_f = N * F_z * Z \tag{2.7}$$

2.7. Positionnement et identification des outils optimums aux points de contact

L'objectif principal de notre projet consiste à déterminer les outils optimums permettant de générer une trajectoire d'outils pour l'usinage des pièces complexes en demi-finition et en finition en un minimum de temps. Pour atteindre cet objectif, il faut d'abord déterminer les outils optimums pour tous les points de contact des contours.

Cet objectif exige le passage par les étapes suivantes :

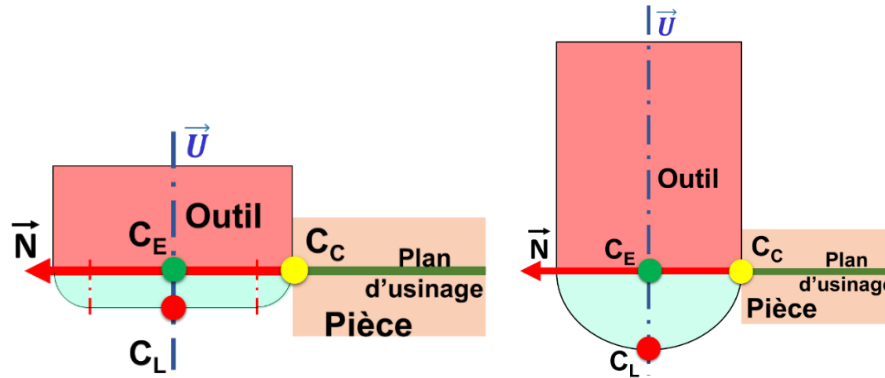
- Calculer les positions des différentes formes d'outils en chaque point de contact.
- Créer les enveloppes des différentes parties des outils.
- Déterminer les cellules en chevauchement avec ces enveloppes.
- Vérifier l'existence des interférences et des collisions.

2.7.1. Positionnement des outils

Dans notre projet, sont considérées les opérations d'usinage en demi-finition et en finition. Le calcul de la position (point centre et point extrémité) de l'outil par rapport à la surface à usiner dépend des paramètres suivants (Figure 2.8) :

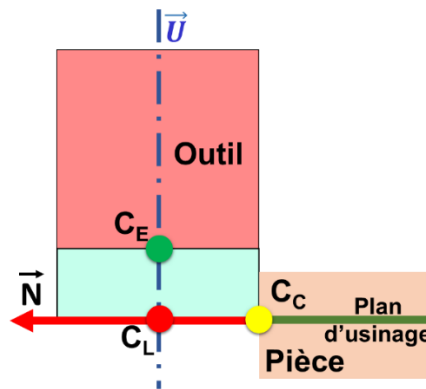
- Forme géométrique de l'outil (partie active).
- Point de contact outil-surface C_C de coordonnées (X_{CC}, Y_{CC}, Z_{CC}) et la normale au point de contact $\vec{N}(X_N, Y_N, Z_N)$.
- Rayon d'outil R et rayon du tore pour l'outil torique R_T .

- Orientation de l'axe d'outil \vec{U} . Lors de l'usinage sur une fraiseuse numérique à 03-axes à broche verticale, les composantes du vecteur \vec{U} sont $(0, 0, 1)$.
- Surépaisseur spécifiée par l'utilisateur. Si sa valeur est nulle, alors c'est la finition, sinon c'est la demi-finition.



a. Outil torique.

b. Outil hémisphérique.



c. Outil cylindrique.

Figure 2.8. Positionnement des différentes formes d'outils.

Pour l'opération de demi-finition, l'outil doit être éloigné du point de contact suivant la normale au point de contact d'une distance égale à la surépaisseur d'usinage. Pour l'opération de finition, l'outil doit être tangent au point de contact.

Le positionnement consiste à déterminer à chaque fois les nouvelles coordonnées du point $C_L (X_{CL}, Y_{CL}, Z_{CL})$ aux différents points de contact. Les coordonnées du point C_L sont calculées comme suit :

Outil hémisphérique :

$$\begin{cases} X_{CL} = X_{cc} + (R + \text{Surépaisseur}) * X_N \\ Y_{CL} = Y_{cc} + (R + \text{Surépaisseur}) * Y_N \\ Z_{CL} = Z + (R + \text{Surépaisseur}) * Z_N - R \end{cases} \quad (2.8)$$

Outil cylindrique :

$$\begin{cases} X_{CL} = X_{cc} + R * A + \text{Surépaisseur} * X_n \\ Y_{CL} = Y_{cc} + R * B + \text{Surépaisseur} * Y_n \\ Z_{CL} = Z_{cc} + R * C + \text{Surépaisseur} * Z_n \end{cases} \quad (2.9)$$

Outil torique :

$$\begin{cases} X_{CL} = X_{cc} + (R_T + \text{Surépaisseur}) * X_N + (R - R_T) * A \\ Y_{CL} = Y_{cc} + (R_T + \text{Surépaisseur}) * Y_N + (R - R_T) * B \\ Z_{CL} = Z_{cc} + (R_T + \text{Surépaisseur}) * Z_N - R_T + (R - R_T) * C \end{cases} \quad (2.10)$$

Avec les valeurs des réels A , B et C sont les résultats du produit vectoriel suivant : $\vec{K}(X_K, Y_K, Z_K) \wedge \vec{U}(0, 0, 1)$. Elles sont données par :

$$\begin{cases} A = Y_k / \sqrt{Y_k^2 + X_k^2} \\ B = -X_k / \sqrt{Y_k^2 + X_k^2} \\ C = 0 \end{cases} \quad (2.11)$$

Le vecteur \vec{K} est calculé comme suit :

$$\vec{K} = \frac{\vec{U} \wedge \vec{N}}{\|\vec{U} \wedge \vec{N}\|}$$

Les composantes du vecteur \vec{K} sont données par :

$$\begin{cases} X_K = -\frac{Y_N}{\sqrt{Y_N^2 + X_N^2}} \\ Y_K = \frac{X_N}{\sqrt{Y_N^2 + X_N^2}} \\ Z_K = 0 \end{cases} \quad (2.12)$$

2.7.2. Création des enveloppes

Après avoir positionné l'outil au point de contact, il est possible maintenant de passer à la vérification des interférences et des collisions en chaque point de contact. Afin d'accélérer cette vérification, une approche basée sur les enveloppes est proposée. La création de l'enveloppe de la partie corps et de la partie active des trois formes d'outils est une étape importante pour la détermination des cellules qui se chevauchent avec ces enveloppes (Figure 2.9).

➤ **Calcul des extrémités de l'enveloppe de la partie active (Figure 2.9) :**

- Les extrémités minimales sont données par :

$$\begin{cases} X_{\min_{\text{EnveloppePActive}}} = X_{CL} - R \\ Y_{\min_{\text{EnveloppePActive}}} = Y_{CL} - R \\ Z_{\min_{\text{EnveloppePActive}}} = Z_{CL} \end{cases} \quad (2.13)$$

- Les extrémités maximales sont données par :

$$\left\{ \begin{array}{l} X_{\max}^{\text{EnveloppePActive}} = X_{CL} + R \\ Y_{\max}^{\text{EnveloppePActive}} = Y_{CL} + R \\ Z_{\max}^{\text{EnveloppePActive}} = Z_{CL} + \text{Longueur}_{\text{PartieActive}} \end{array} \right. \quad (2.14)$$

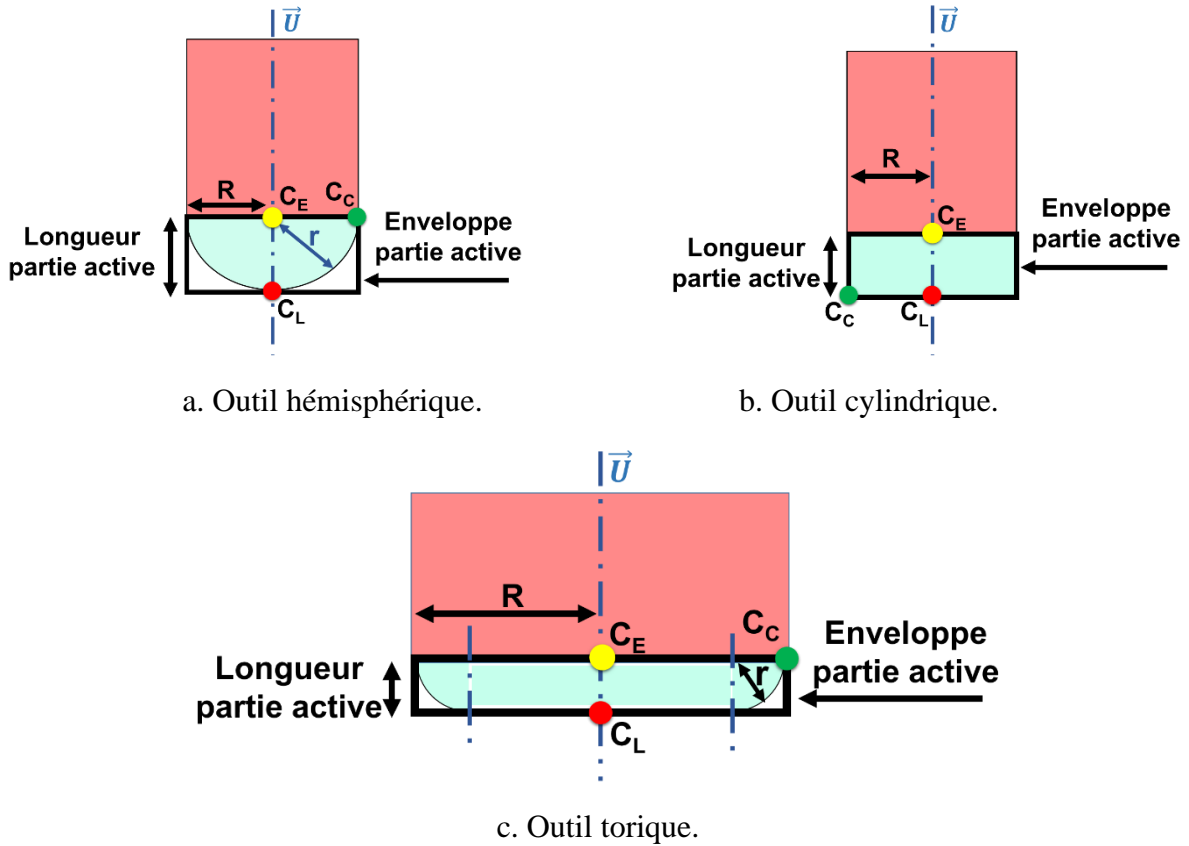


Figure 2.9. Enveloppes de la partie active des outils.

➤ Calcul des extrémités de l'enveloppe de la partie corps (Figure 2.10) :

- Les extrémités minimales sont données par :

$$\left\{ \begin{array}{l} X_{\min}^{\text{EnveloppePCorps}} = X_{CL} - R \\ Y_{\min}^{\text{EnveloppePCorps}} = Y_{CL} - R \\ Z_{\min}^{\text{EnveloppePCorps}} = Z_{CL} + \text{Longueur}_{\text{PartieActive}} \end{array} \right. \quad (2.15)$$

- Les extrémités maximales sont données par :

$$\left\{ \begin{array}{l} X_{\max}^{\text{EnveloppePCorps}} = X_{CL} + R \\ Y_{\max}^{\text{EnveloppePCorps}} = Y_{CL} + R \\ Z_{\max}^{\text{EnveloppePCorps}} = Z_{CL} + \text{Longueur} \end{array} \right. \quad (2.16)$$

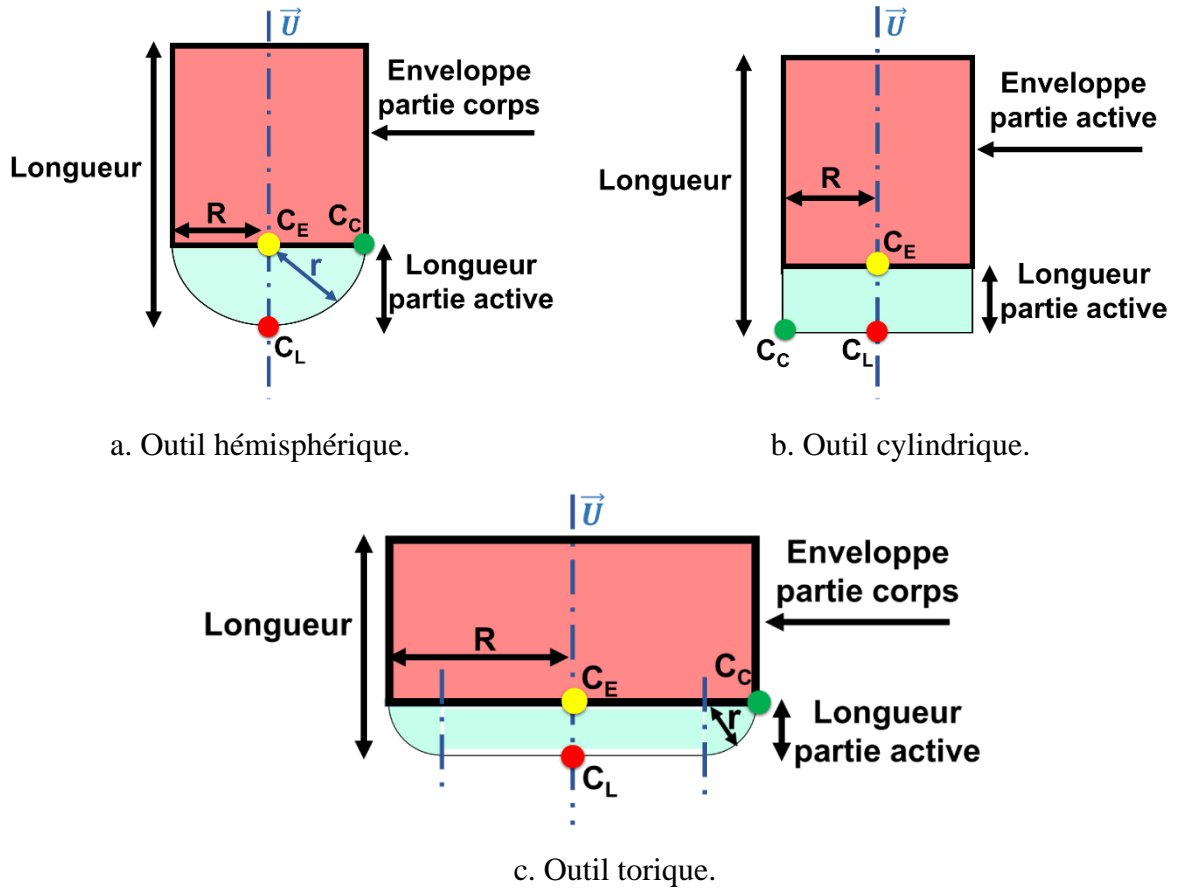


Figure 2.10. Enveloppes de la partie corps des outils.

2.7.3. Détermination des cellules en chevauchement

Cette étape consiste à localiser pour chaque point de contact, les cellules en chevauchement avec l'enveloppe de l'outil pour éviter de tester tous les points du modèle de la pièce à usiner lors de la détection des zones d'interférences et de collisions (Figures 2.11, Figure 2.12 et Figure 2.13).

Les indices des cellules qui se chevauchent avec l'outil sont données par :

$$\begin{cases} \text{Indice}_{i_{Min}} = (X_{min_{EnveloppePartieActive}} - X_{min_{Brut}}) / \text{Longueur}_{Cellule} \\ \text{Indice}_{j_{Min}} = (Y_{min_{EnveloppePartieActive}} - Y_{min_{Brut}}) / \text{Largeur}_{Cellule} \\ \text{Indice}_{k_{Min}} = (Z_{min_{EnveloppePartieActive}} - Z_{min_{Brut}}) / \text{Hauteur}_{Cellule} \end{cases} \quad (2.17)$$

$$\begin{cases} \text{Indice}_{i_{Max}} = (X_{max_{EnveloppePartieActive}} - X_{max_{Brut}}) / \text{Longueur}_{Cellule} \\ \text{Indice}_{j_{Max}} = (Y_{max_{EnveloppePartieActive}} - Y_{max_{Brut}}) / \text{Largeur}_{Cellule} \\ \text{Indice}_{k_{Max}} = (Z_{max_{EnveloppePartieActive}} - Z_{max_{Brut}}) / \text{Hauteur}_{Cellule} \end{cases} \quad (2.18)$$

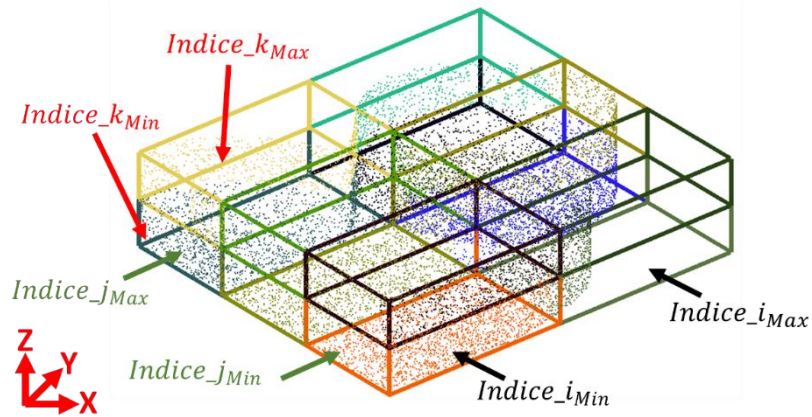


Figure 2.11. Indices des cellules en chevauchement avec l'outil.

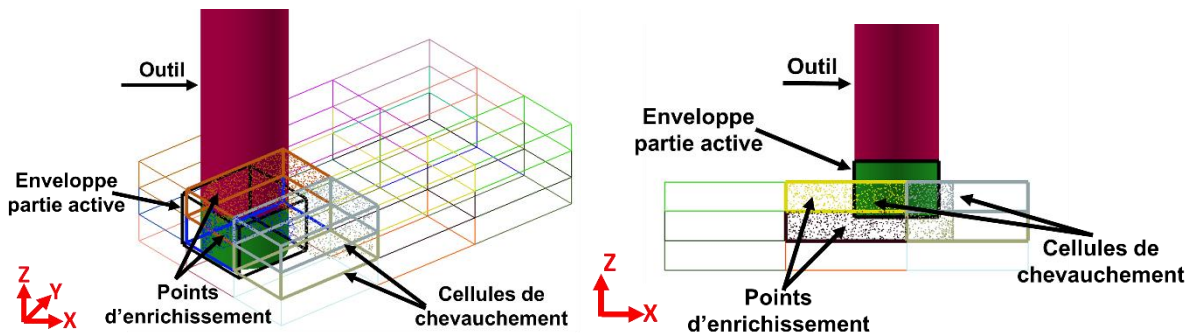


Figure 2.12. Cellules de chevauchement.

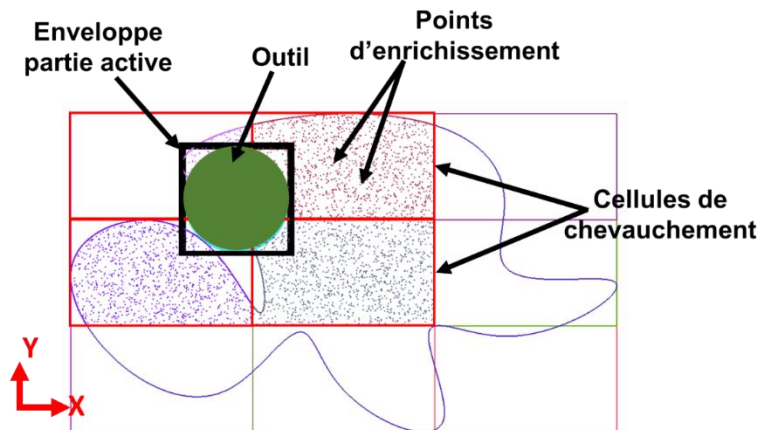


Figure 2.13. Cellules de chevauchement.

2.7.4. Vérification de l'intersection et de la collision

Après la détermination des zones susceptibles de créer les problèmes d'interférences et de collisions, les points d'enrichissement dans ces zones sont utilisés pour tester l'existence des collisions et des interférences.

✚ Problème d'interférences :

❖ Outil cylindrique :

Pour tester l'interférence on doit vérifier si les points appartiennent à l'enveloppe de la partie active. Tout d'abord on compare ces points avec les extrémités de l'enveloppe de la partie active (Figure 2.14).

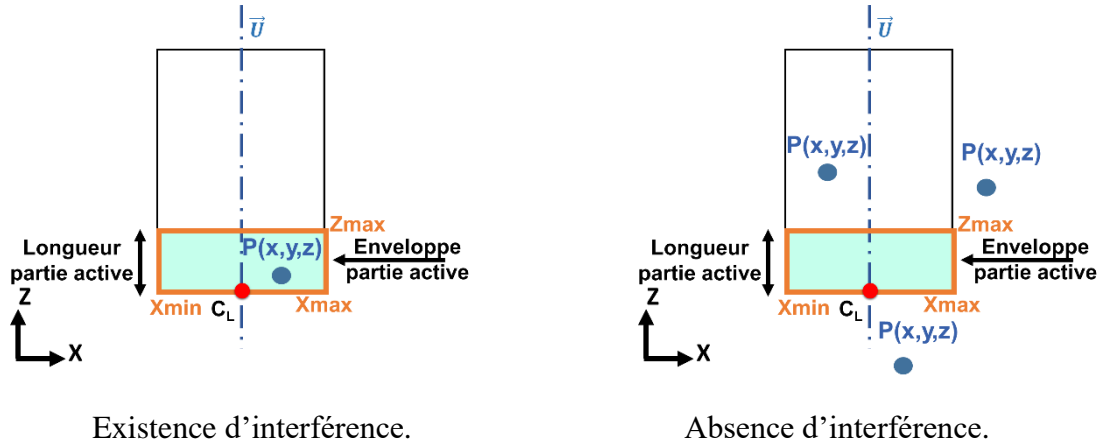


Figure 2.14. Interférence avec partie active d'outil.

➤ Les points sont en dehors de l'enveloppe si les conditions suivantes sont satisfaites :

- $X \leq X_{min}$ ou $X \geq X_{max}$
- $Y \leq Y_{min}$ ou $Y \geq Y_{max}$
- $Z \leq Z_{CL}$ ou $Z \geq Z_{CL} + Longueur_{PartieActive}$

➤ Si les points appartiennent à l'enveloppe, il y a une possibilité d'interférence. Donc, on doit vérifier pour chaque point s'il appartient au cylindre en utilisant la distance entre le point P et le point C_L (centre du cylindre) et on la compare avec le rayon R (Figure 2.15). La distance est calculée comme suit :

$$Distance = \sqrt{(X - X_{CL})^2 + (Y - Y_{CL})^2} \quad (2.19)$$

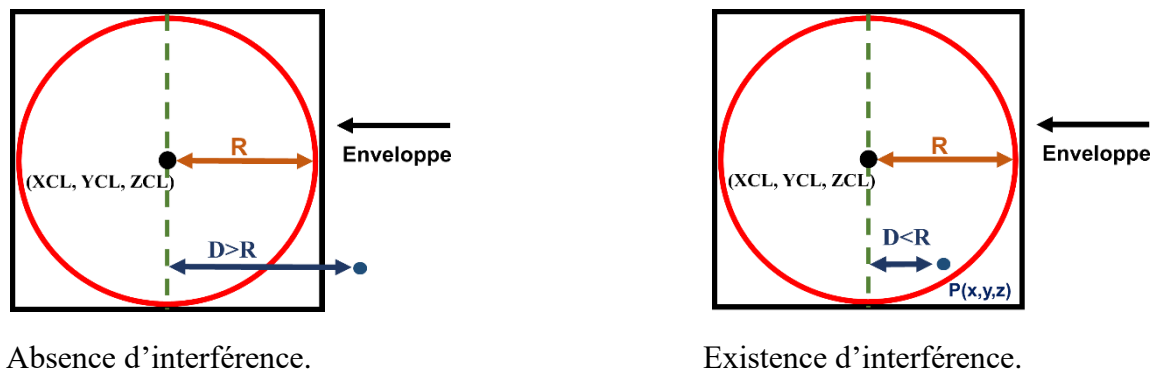
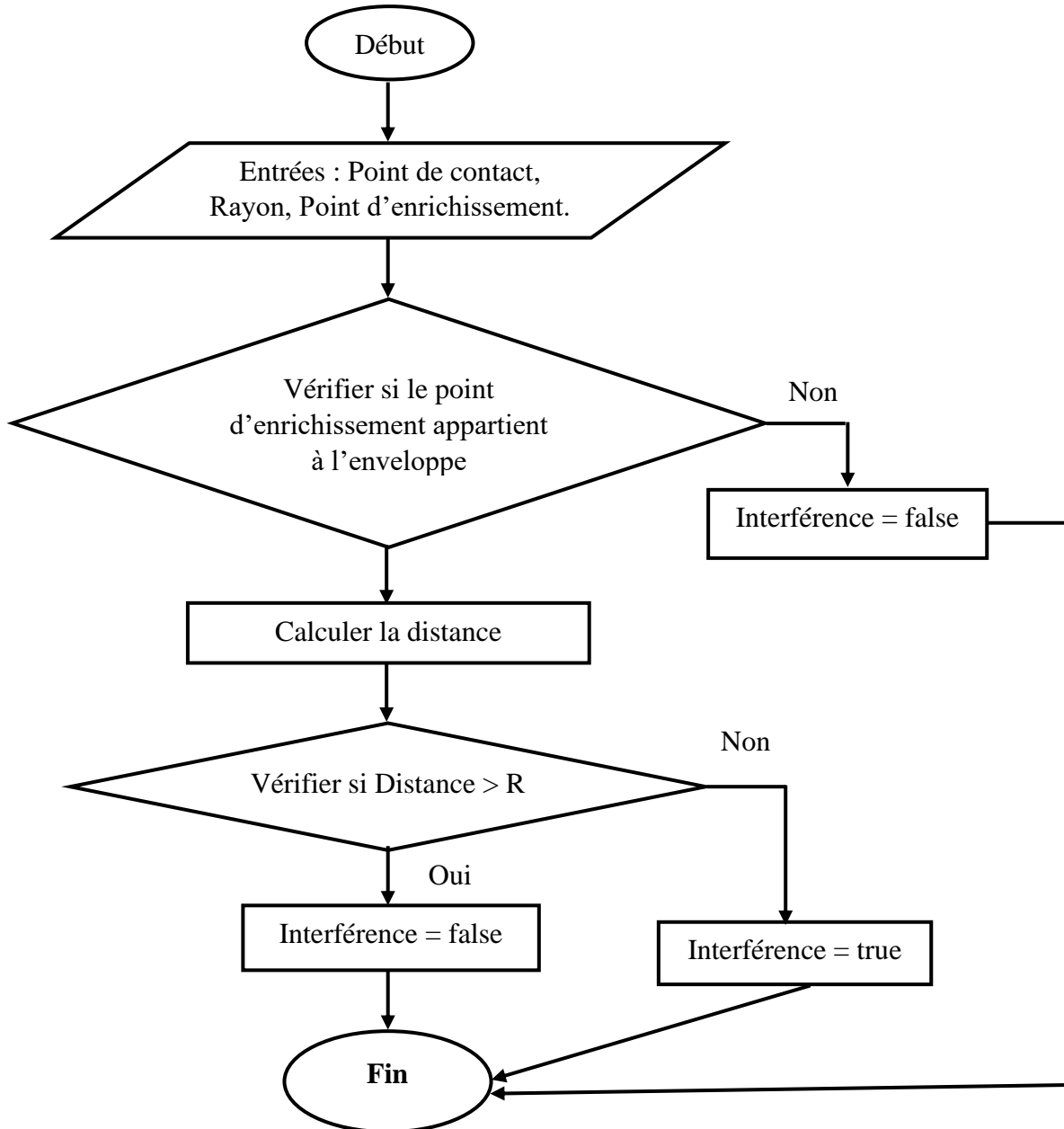


Figure 2.15 Interférence avec le cylindre.

L'organigramme suivant montre les étapes de vérification :



Organigramme 2.2. Vérification de l'interférence.

❖ **Outil hémisphérique :**

La vérification d'appartenance des points d'enrichissement à la partie active se fait de la même manière comme l'outil cylindrique (Figure 2.16).

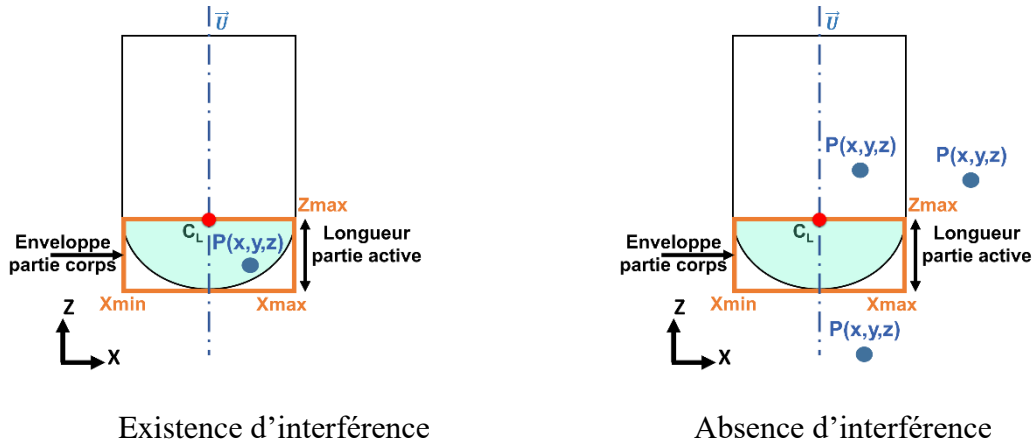


Figure 2.16. Interférence avec la demi-sphère.

- Les points sont en dehors de l'enveloppe si les conditions suivantes sont satisfaites :
 - $X \leq X_{min}$ Ou $X \geq X_{max}$
 - $Y \leq Y_{min}$ Ou $Y \geq Y_{max}$
 - $Z \leq Z_{CL}$ Ou $Z \geq Z_{CL} + \text{Longueur}_{PartieActive}$

- Dans le cas où le point appartient à l'enveloppe de la partie active, on doit vérifier l'appartenance du point à la demi-sphère. Pour cela, on utilise l'équation de la sphère :

$$Distance = \sqrt{(X - X_{CL})^2 + (Y - Y_{CL})^2 + (Z - Z_{CL} - R)^2} \quad (2.20)$$

Si $Distance < R$, alors il existe une interférence (Figure 2.16).

❖ **Outil torique :**

De la même manière, il faut tester l'appartenance du point P à l'enveloppe de la partie active de l'outil (Figure 2.17).

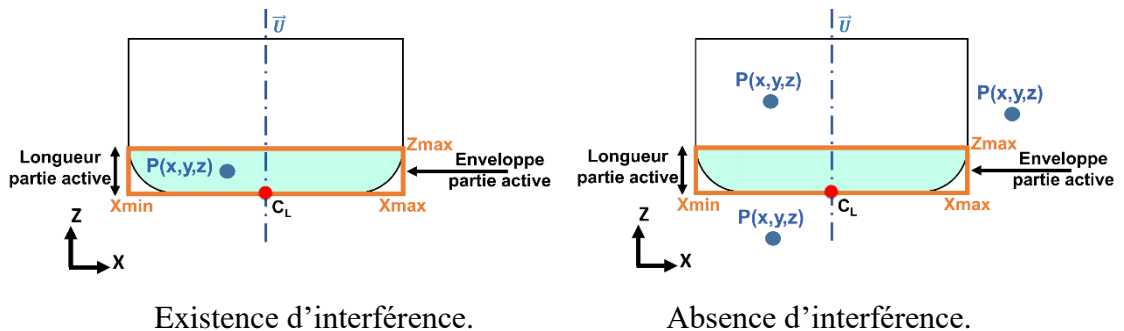


Figure 2.17. Interférence avec le tore.

Si le point appartient à la partie active, il y'a une possibilité d'interférence. Donc, il faut tester s'il appartient à la partie cylindrique ou à la partie torique (Figure 2.18).

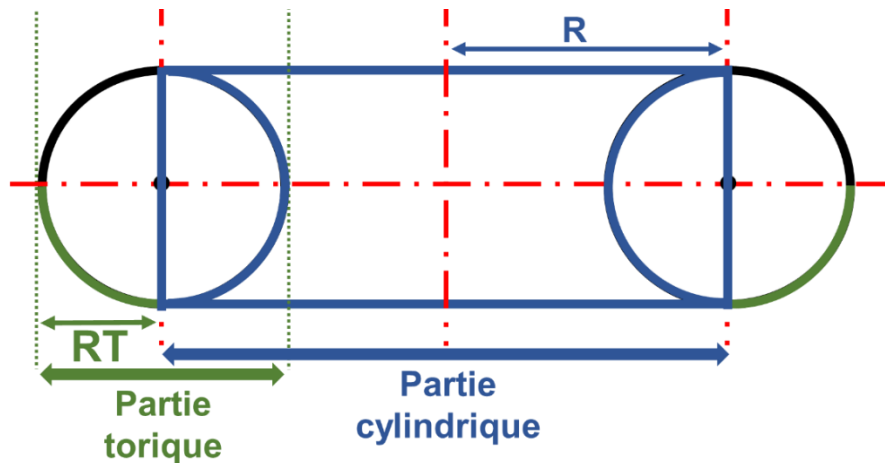


Figure 2.18. Partie active de l'outil torique.

On doit calculer d'abord la distance entre le point P et le point CL comme suit :

$$Distance = \sqrt{(X - X_{CL})^2 + (Y - Y_{CL})^2} \quad (2.21)$$

Les différents cas possibles sont :

- ✓ Si $Distance > R + R_T$: pas d'interférence (Figure 2.19).

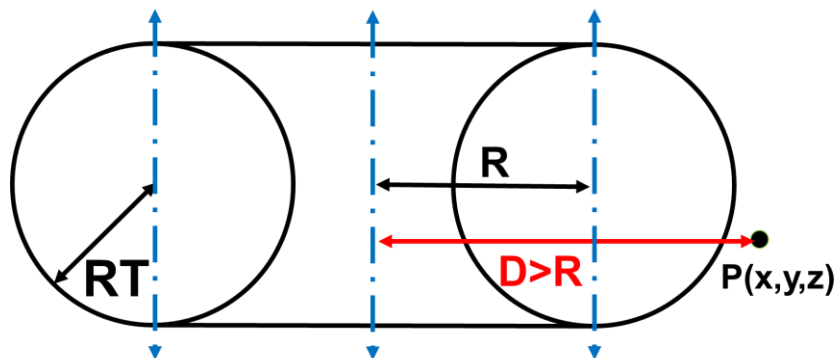


Figure 2.19 Absence d'interférence.

- ✓ Si $Distance < R$: interférence au niveau de la partie cylindrique (Figure 2.20).

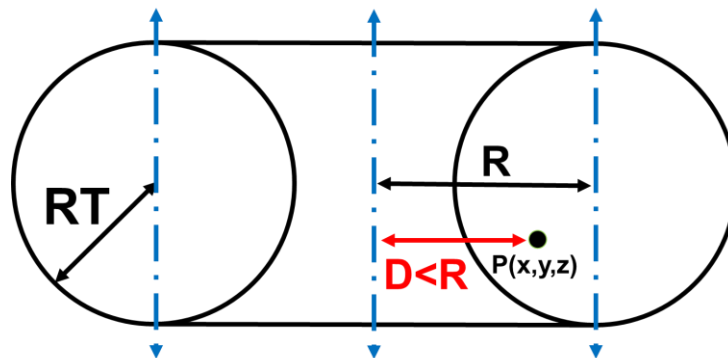


Figure 2.20. Existence d'interférence.

- ✓ Si $Distance > R$ et $Distance < R+R_T$: possibilité d'interférence au niveau de la partie torique (Figure 2.21).

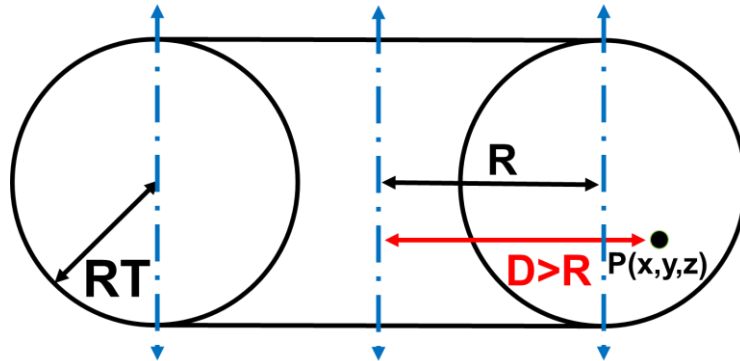


Figure 2.21 Existence d'interférence.

L'interférence dans cette partie existe si la valeur de Z vérifie les deux conditions suivantes :

$$\begin{cases} Z > Z_{Moins} \\ Z < Z_{Plus} \end{cases}$$

Avec :

$$Z_{Plus} = Z_{CL} + R_T + \sqrt{D} \tag{2.22}$$

$$Z_{Moins} = Z_{CL} + R_T - \sqrt{D} \tag{2.23}$$

$$D = R_T^2 - \left(\sqrt{(\sqrt{X - X_{CL}})^2 + (\sqrt{Y - Y_{CL}})^2} - R + R_T \right)^2 \tag{2.24}$$

✚ Problème de collisions :

La détection des collisions exige d'appliquer le même principe utilisé pour détecter l'interférence mais cette fois-ci avec la partie corps de l'outil.

- ❖ Absence de collision (Figure 2.22).

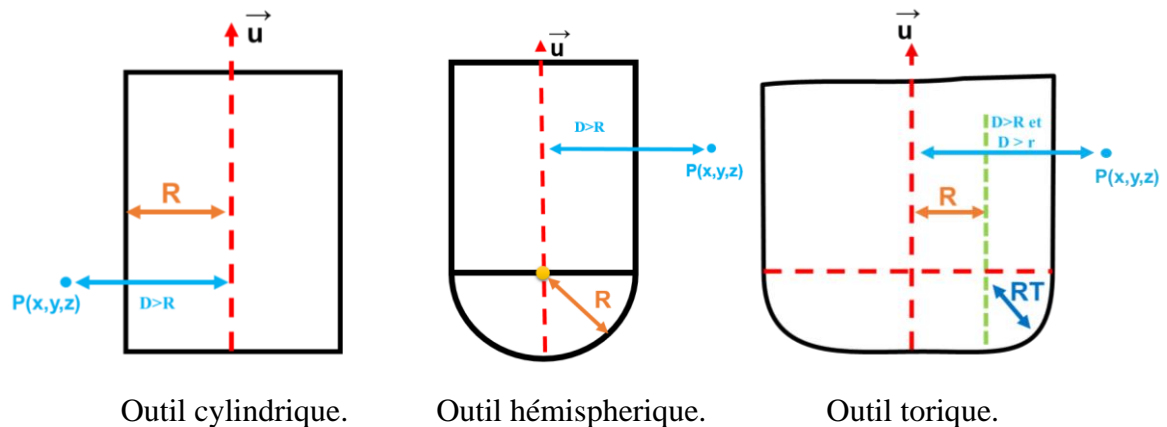


Figure 2.22. Absence de collision.

- ❖ Existence de collision (Figure 2.23).

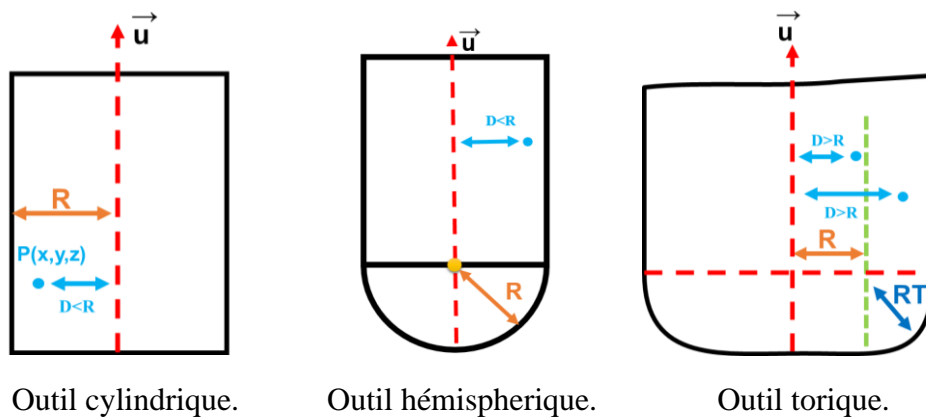


Figure 2.23. Existence de collision.

2.8. Association de la forme d'outils aux contours :

L'association de la forme d'outil à chaque contour est une étape primordiale pour atteindre l'objectif principal de ce projet. Afin d'identifier une forme d'outil pour un contour donné, la démarche suivante est suivie :

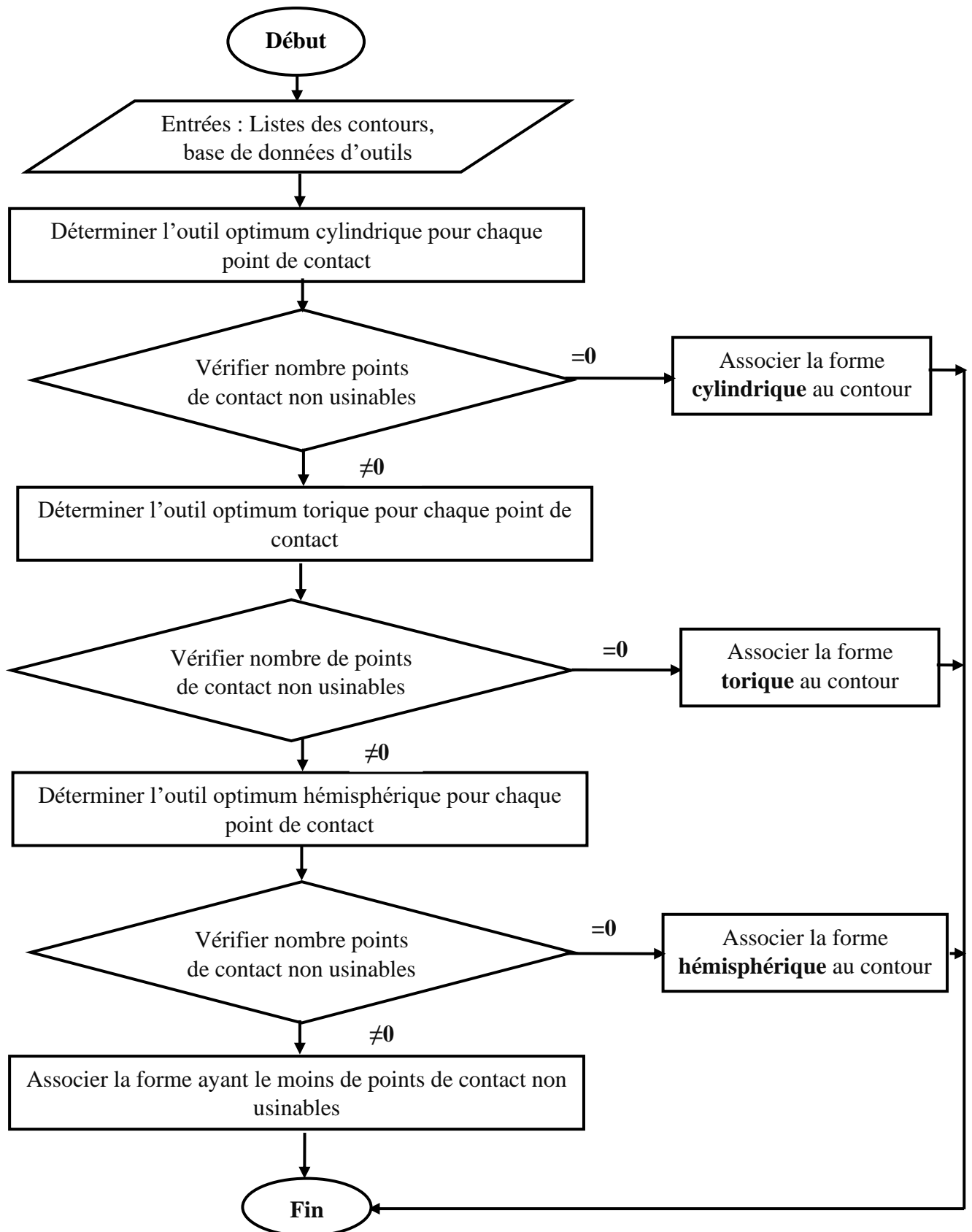
- Déterminer l'outil optimum cylindrique pour chaque point de contact. Ce point peut être :

Usinable : dans ce cas, il existe un outil optimum qui ne génère pas d'interférences ou de collisions.

Non usinable : aucun des outils de la table des bases de données ne peut être positionné hors collisions et interférences.

- Si tous les points de contact sont usinables, la forme « cylindrique » sera associée au contour. Si ce n'est pas le cas, on doit répéter le même parcours avec l'outil torique et puis hémisphérique. Si on arrive à un contour qui possède des points de contact non usinables avec les trois formes d'outils, la forme associée est la forme qui a le moins de points de contact non usinables.

L'organigramme suivant résume toute cette démarche :



Organigramme 2.3. Association de la forme d'outil aux contours.

2.9. Fixation du nombre d'outils par contour

Après avoir associé à chaque contour une forme, l'utilisateur a le droit de limiter le nombre d'outils N par contour. Dans ce projet, il est possible d'utiliser un maximum de trois (03) outils.

- ✓ Si $N=1$, affectation du plus petit outil avec l'indice I_0 à tous les points de contact du contour.
- ✓ Si $N=2$, affectation du plus petit outil I_0 et l'outil le plus utilisé I_1 .
- ✓ Si $N = 3$, affectation du plus petit outil I_0 , le premier outil le plus utilisé I_1 et le deuxième outil le plus utilisé I_2 .

Avant d'affecter les outils aux points de contact, il faut comparer les indices des outils I_0 , I_1 et I_2 avec les indices initiaux affectés précédemment (Figure 2.24).

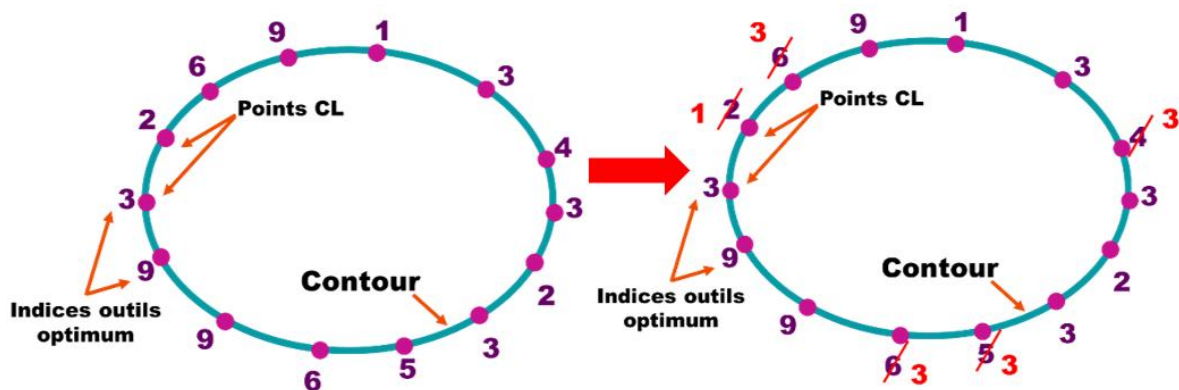


Figure 2.24. Réaffectation des indices d'outils aux points de contact.

2.10. Choix du nombre d'outils à utiliser pour le trajet

Après avoir identifié pour chaque contour un, deux ou trois outils, on doit limiter le nombre d'outils à utiliser pour tout le trajet en fonction du nombre d'outils disponibles dans le magasin d'outils de la fraiseuse. Pour cela, on possède comme suit :

- Déterminer les indices finaux du trajet, c'est-à-dire on parcourt tous les contours du trajet en calculant le nombre d'utilisation de chaque outil et on les met dans un tableau sans répétition. On aura en final, trois tableaux, chacun avec une forme précise.
- Trier les trois (03) tableaux en fonction du nombre d'utilisation en utilisant l'algorithme « trie par fusion » du plus grand au plus petit.
- Fixée le nombre total N pour chaque forme des trois outils à utiliser.

La sélection des N outils réels choisis suit les étapes suivantes :

- Chercher le plus petit outil optimum de chaque forme et le stocker dans une liste.
- Choisir les N-1 premiers outils optimums pour chaque forme ayant le nombre d'utilisation le plus grand sans compter le plus petit outil optimum et les stocker dans la liste précédente.
- Trier la liste des indices d'outils réels choisis du plus petit au plus grand.
- Réaffectation finale des outils réels :
 1. Réaffecter pour chaque contour les indices des outils réels selon sa forme.
 2. Réaffecter tous les points de contact avec les nouveaux indices du contour et calculer les positions des nouveaux outils réels.

2.11. Génération de la trajectoire d'outils

Cette étape consiste à générer tous les déplacements des outils du trajet lors des opérations de demi-finition et de finition de toute la pièce à usiner. La génération du trajet suit les étapes suivantes :

- a) Séparer les segments usinés avec le même outil et les stocker dans une liste, c'est-à-dire pour chaque contour, sauvegarder tous les points CL (points d'extrémités des outils) qui ont le même outil (Figure 2.25).

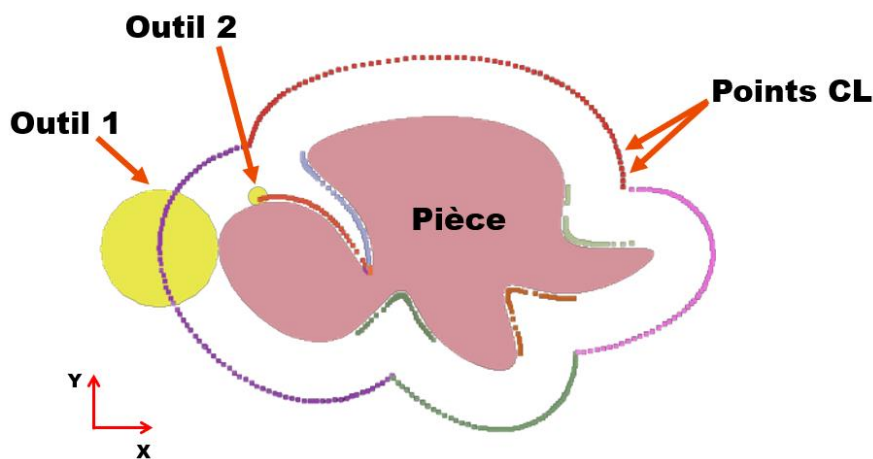


Figure 2.25. Trajets d'outils

- b) Fixer les distances de sécurité, d'engagement et de dégagement par l'utilisateur.
- c) Ajouter à chaque liste des points quatre (04) points, deux points au début (P1, P2) appelés points d'engagement et deux points à la fin (P3, P4) appelés points de dégagement (Figure 2.26).

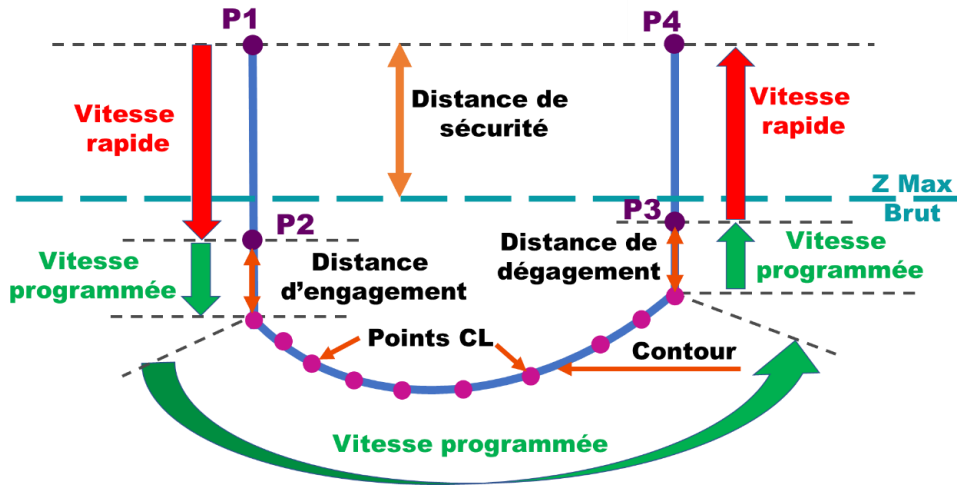


Figure 2.26. Points d'engagement et de dégagement.

La vitesse d'outil entre le point P1 et P2 est rapide puis lente (vitesse programmée) jusqu'au point P3 pour éviter les chocs et les vibrations, et par conséquent avoir un bon état de surface. Puis elle redevient rapide.

A la fin de cette étape, un tableau contenant tous les points de passage avec leurs outils est généré. La génération du trajet comprend deux modes possibles :

- **Sans optimisation** : le premier outil fait un engagement et enlève de la matière en une portion, puis il fait un dégagement. La fraiseuse fait un changement d'outil et ainsi de suite jusqu'à la fin du trajet. Le séquençement d'outils se fait par rapport aux outils affectés aux contours.
- **Avec optimisation** : le premier outil fait un engagement et enlève de la matière en une portion, puis il fait un dégagement, mais cette fois-ci la fraiseuse ne fait pas un changement d'outil jusqu'à terminer toutes les portions qui peuvent être usinées, puis il fait un dégagement, et de la même façon pour les autres outils qui restent. Dans ce scénario, le changement d'outils n'arrive pas souvent, donc le temps de changement d'outil va diminuer, par conséquent le temps d'usinage va diminuer aussi. Le séquençement d'outils se fait par rapport aux outils affectés aux contours triés du plus grand rayon jusqu'au petit rayon.

2.12. Simulation du trajet d'outils

La simulation du trajet d'outils implique la virtualisation des déplacements des outils sur tous les points de contact par rapport à la pièce à usiner. Son objectif est la vérification visuelle du trajet d'outils avant la génération du fichier « CL-File ».

3. Modélisation avec UML

UML « Unified Modeling Language » ou « Langage de Modélisation Unifier », est un langage graphique qui permet de représenter, de communiquer les divers aspects d'un système d'information, où chaque symbole graphique possède une sémantique, de manière précise, complète et sans ambiguïté. Il est basé sur des pictogrammes conçus pour fournir des modèles normalisés pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en programmation orienté objet (POO).

Dans ce projet, trois (03) diagrammes sont utilisés : diagramme de cas d'utilisation, diagramme d'activités et diagramme de classe.

3.1. Diagramme de cas d'utilisation

Le diagramme de cas représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est une séquence d'activités ou d'actions organisées en étapes distinctes, et qu'un système effectue en réponse à une sollicitation extérieure. C'est le premier diagramme du modèle UML, celui qui assure la relation entre l'utilisateur et les objets que le système met en œuvre. Parmi ces objectifs :

- Définir les besoins fonctionnels du système
- Définir le dialogue entre l'utilisateur et le système.
- Etablir les scénarios fonctionnels qui seront utilisés pour la recette du système.
- Servir de support tout au long des phrases de développement du système.

❖ **Diagramme de cas d'utilisation général** : le système est divisé en sept grandes phases (Figure 2.27).

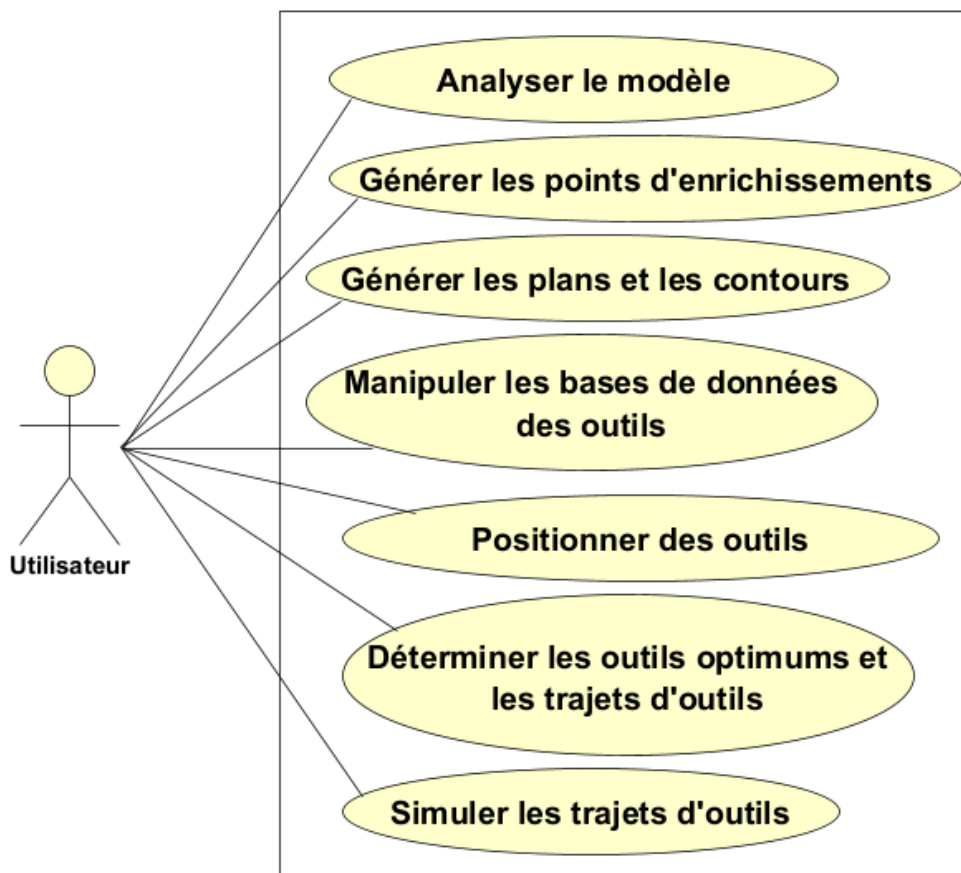


Figure 2.27. Diagramme de cas d'utilisation général.

❖ Diagramme de cas d'utilisation « Analyser le modèle »

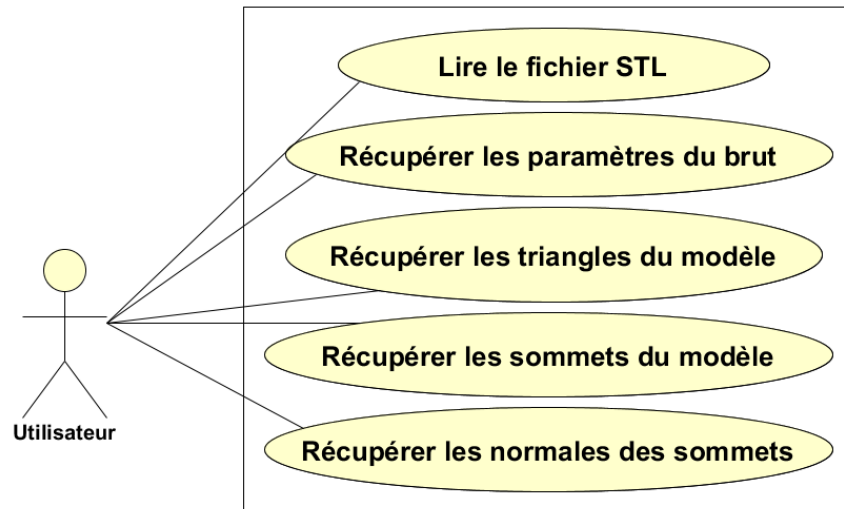


Figure 2.28. Diagramme de cas d'utilisation « Analyser le modèle ».

❖ Diagramme de cas d'utilisation « Générer les points d'enrichissements »

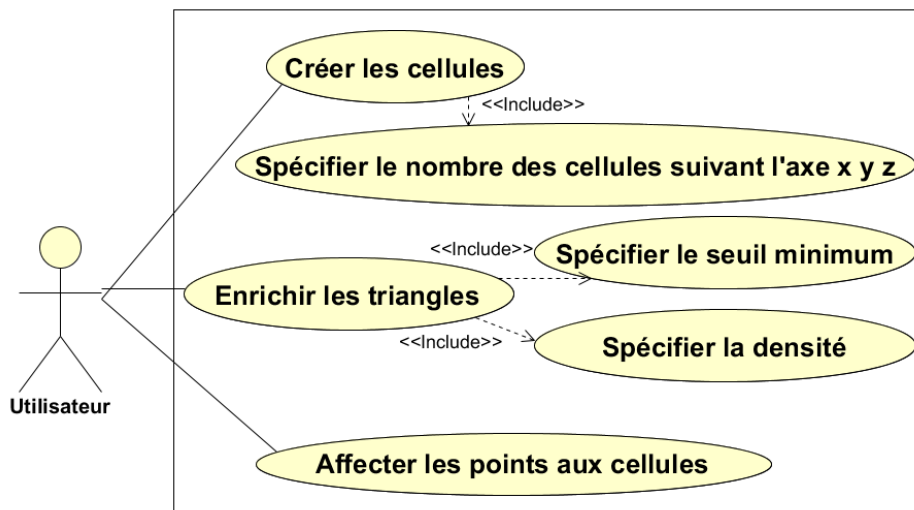


Figure 2.29. Diagramme de cas d'utilisation « Générer les points d'enrichissements ».

❖ Diagramme de cas d'utilisation « Générer les plans et les contours »

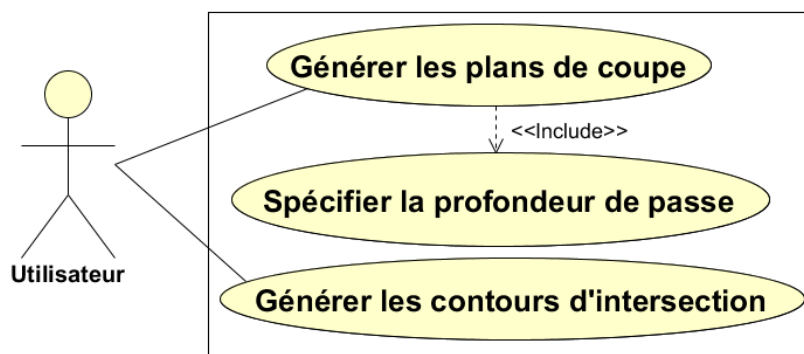


Figure 2.30. Diagramme de cas d'utilisation « Générer les plans et les contours ».

❖ Diagramme de cas d'utilisation « Manipuler les bases de données des outils »

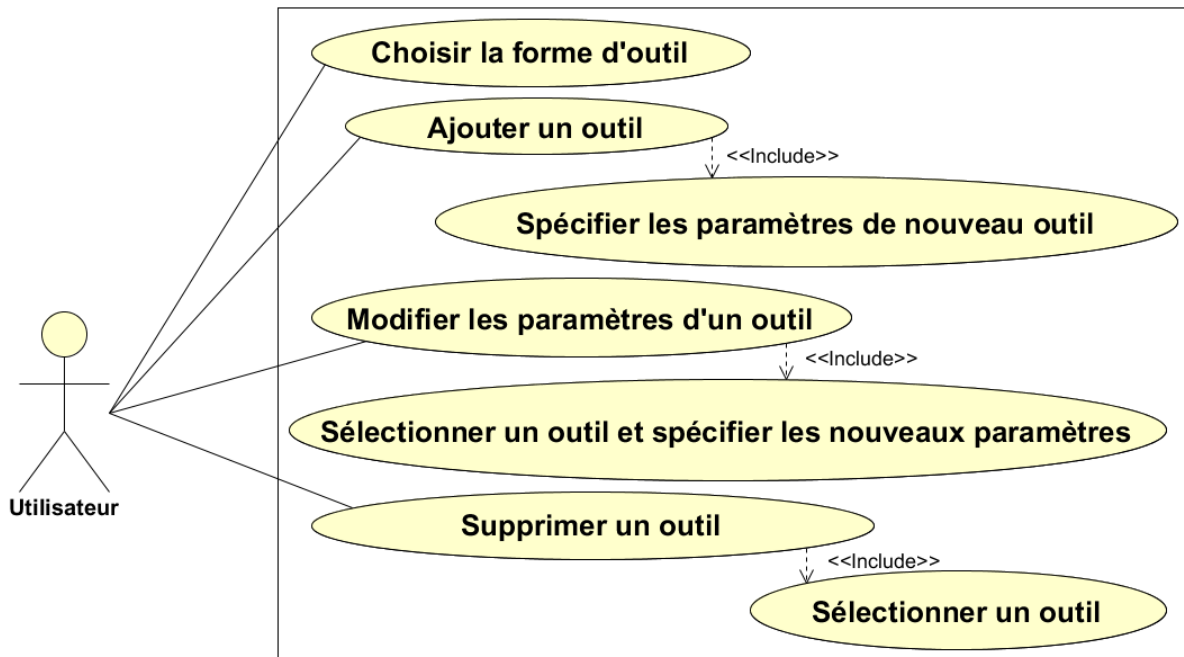


Figure 2.31. Diagramme de cas d'utilisation « Manipuler les bases de données des outils ».

❖ Diagramme de cas d'utilisation « Positionner des outils »

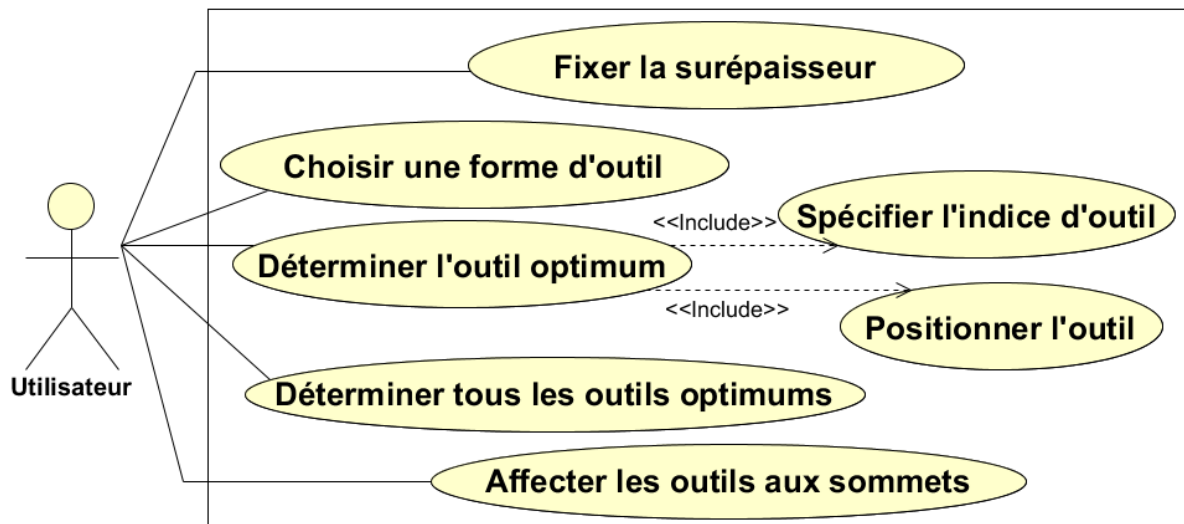


Figure 2.32. Diagramme de cas d'utilisation « Positionner des outils ».

❖ Diagramme de cas d'utilisation « Déterminer les outils optimums et les trajets d'outils »

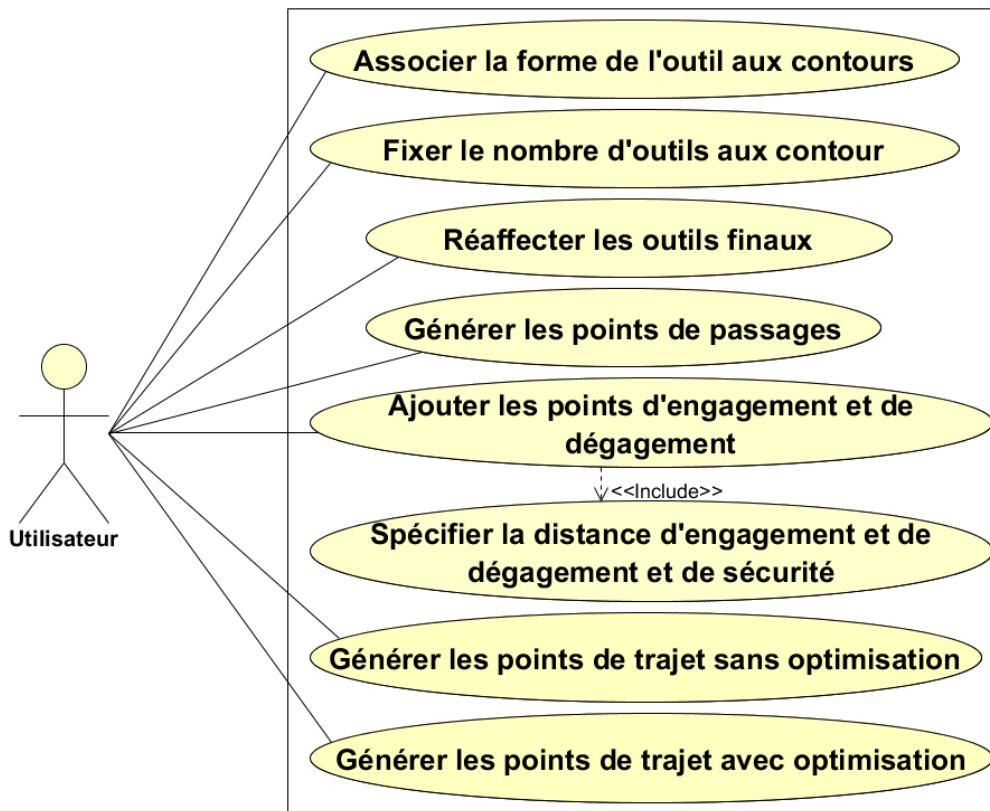


Figure 2.33. Diagramme de cas d'utilisation « Déterminer les outils optimums et les trajets d'outils ».

❖ Diagramme de cas d'utilisation « Simuler les trajets d'outils »

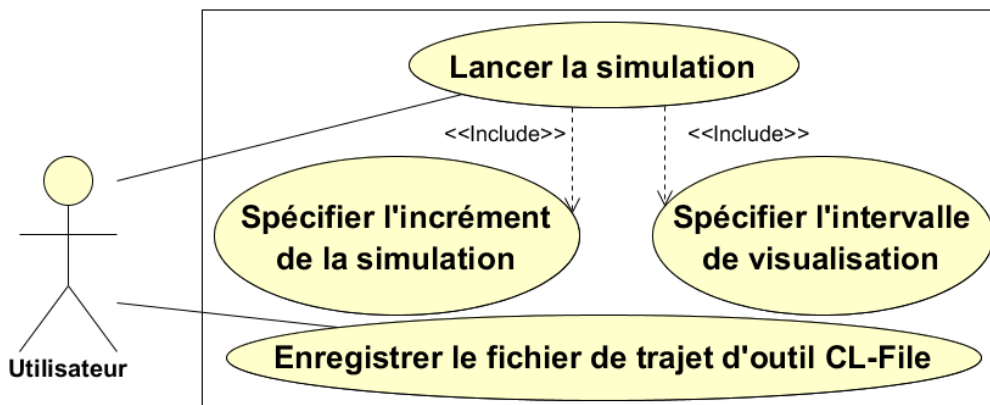


Figure 2.34. Diagramme de cas d'utilisation « Simuler les trajets d'outils ».

3.2. Diagramme d'activités

Le diagramme d'activités expose les activités séquentielles et parallèles d'un processus. Il peut être utilisé pour décrire la logique procédurale d'une activité, il permet de modéliser des processus métier et des enchaînements d'activités.

❖ Diagramme d'activité général

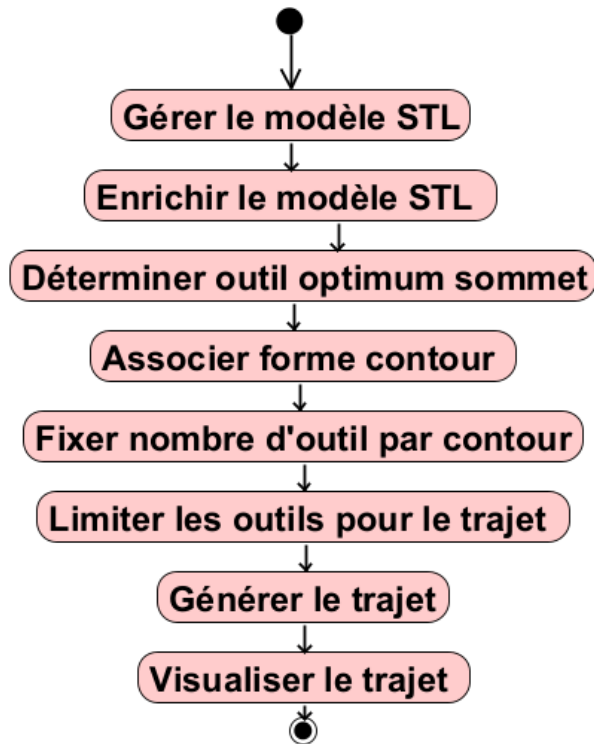


Figure 2.35. Diagramme d'activité général.

❖ Diagramme d'activité « Enrichir le modèle STL »

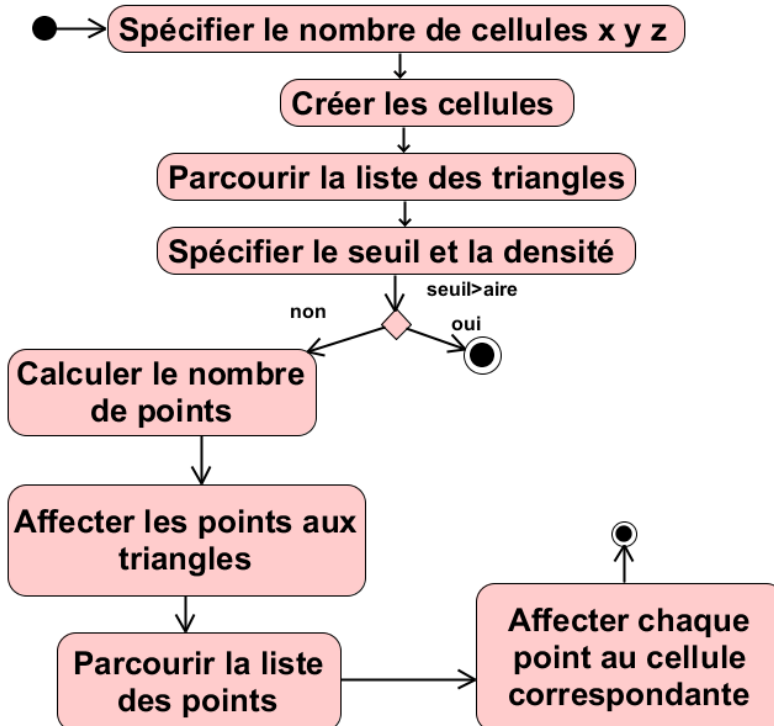


Figure 2.36. Diagramme d'activité « Enrichir le modèle STL ».

❖ Diagramme d'activité « Déterminer outil optimum sommet »

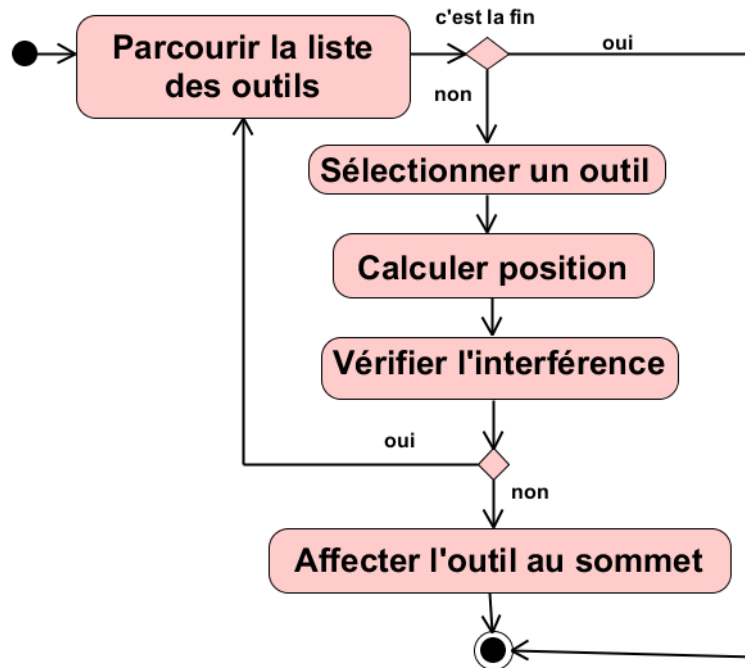


Figure 2.37. Diagramme d'activité « Déterminer outil optimum sommet ».

❖ Diagramme d'activité « Associer la forme aux contours »

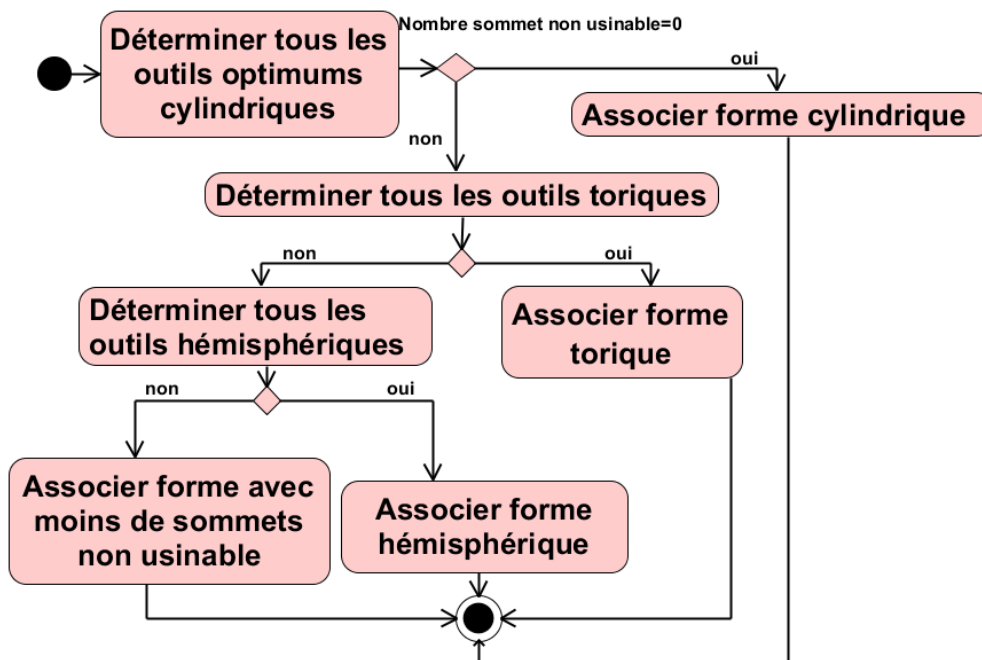


Figure 2.38. Diagramme d'activité « Associer la forme aux contours ».

❖ Diagramme d'activité « Générer le trajet »

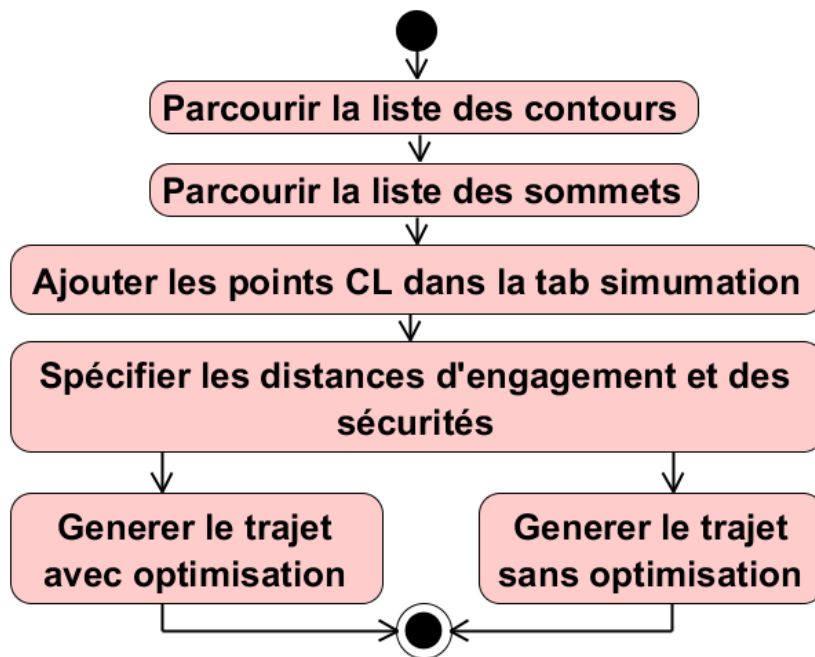


Figure 2.39. Diagramme d'activité « Générer le trajet ».

❖ Diagramme d'activité « Visualiser le trajet »

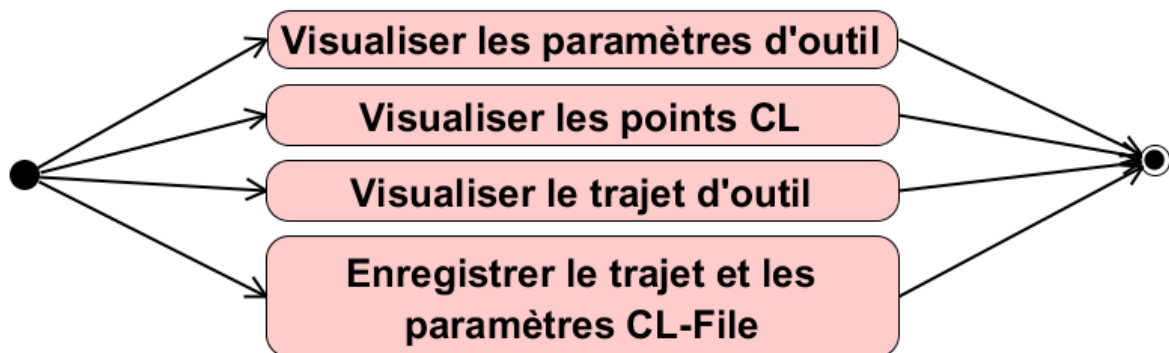


Figure 2.40. Diagramme d'activité « Visualiser le trajet ».

3.3. Diagramme de classes

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système. Il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation). Il permet de décrire la structure interne des classes en termes d'attributs et d'opérations (Figure 2.41).

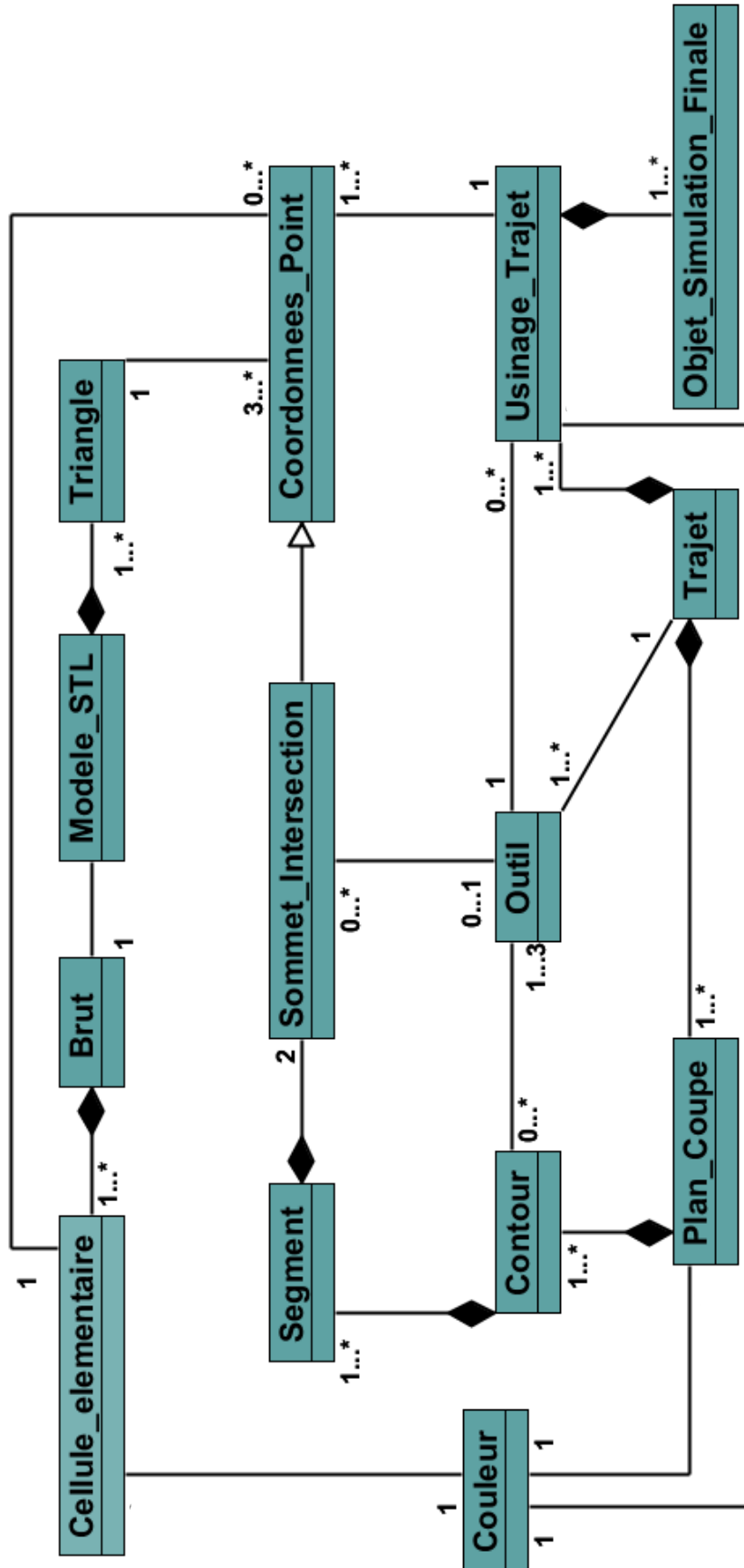


Figure 2.41. Diagramme de classes.

- ✚ **Classe « Objet Simulation Finale »** : c'est la classe qui contient une portion du trajet, ainsi que le mode de vitesse de simulation (Figure 2.42).

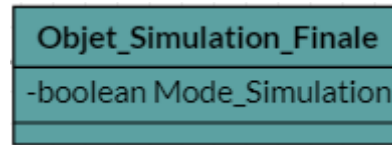


Figure 2.42. Classe « Objet Simulation Finale ».

- ✚ **Classe « Trajet »** : c'est la classe qui regroupe les plans de tous les trajets, ainsi que les listes des trois formes d'outils (Figure 2.43).

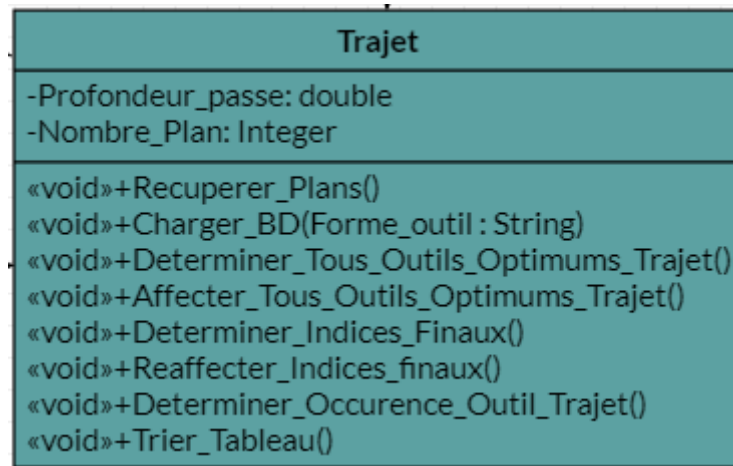


Figure 2.43. Classe « Trajet ».

- ✚ **Classe « Plan de coupe »** : contient les contours du plan. Elle permet de déterminer les outils optimums des contours et réaffecter les indices des outils finaux (Figure 2.44).

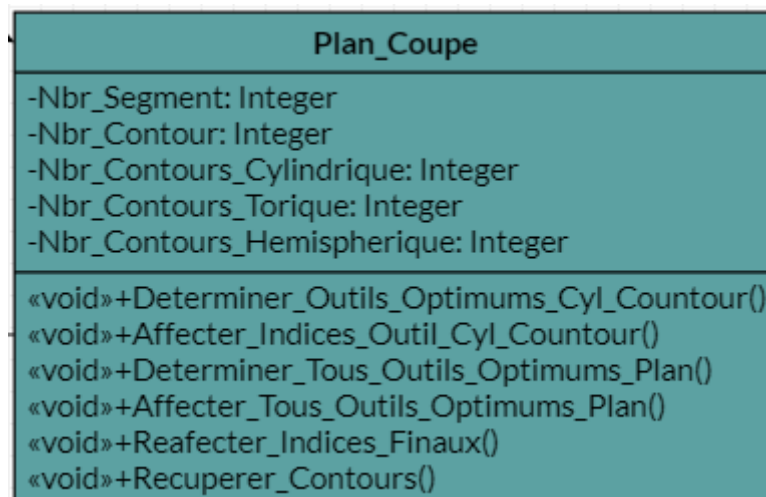


Figure 2.44. Classe « Plan de coupe ».

- ✚ **Classe « Usinage trajet »** : cette classe contient tous les points de passage. Elle permet d'ajouter les points d'engagement et de dégagement (Figure 2.45).

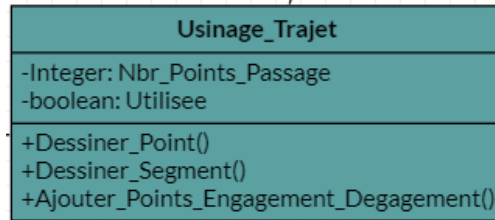


Figure 2.45. Classe « Usinage trajet ».

- + **Classe « Outil »** : cette classe regroupe les différents paramètres géométriques et technologiques d'un outil. Elle permet de visualiser l'enveloppe de l'outil, de déterminer l'interférence (Figure 2.46).

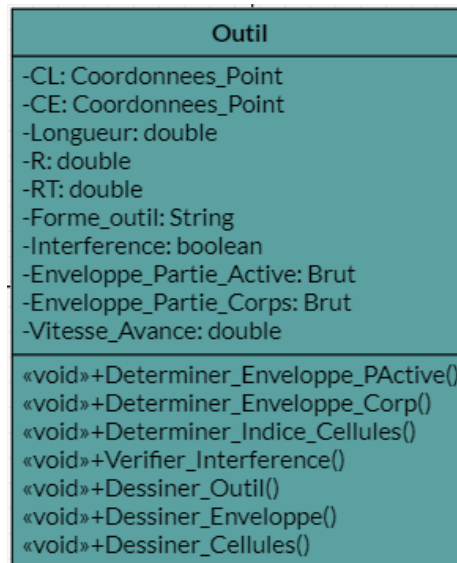


Figure 2.46. Classe « Outil ».

- + **Classe « Contour »** : cette classe contient les segments d'un contour, les indices des outils d'un contour, et d'autres informations (Figure 2.47). Elle permet de déterminer les outils optimums des différentes formes d'outils et d'affecter à chaque sommet d'un contour un indice d'outil.

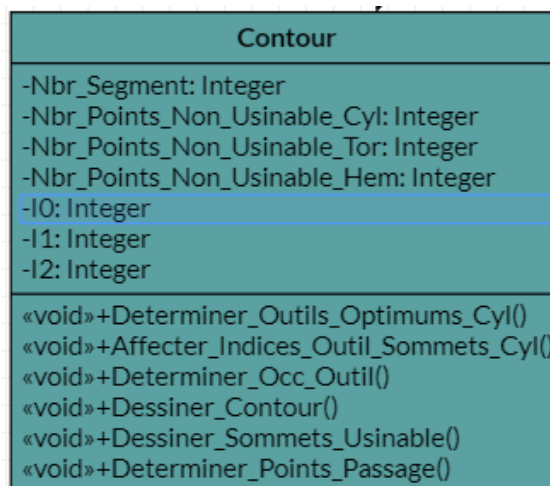


Figure 2.47. Classe « Contour ».

- ✚ **Classe « Segment »** : cette classe regroupe les deux sommets d'intersection. Elle permet de dessiner un segment (Figure 2.48).

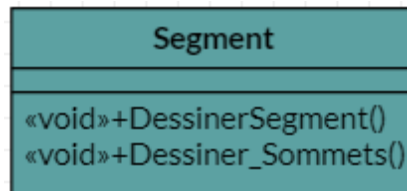


Figure 2.48. Classe « Segment »

- ✚ **Classe « Sommet d'intersection »** : c'est la classe qui définit un sommet par ses différentes caractéristiques (coordonnées x, y, z, vecteur normal, etc.). Les méthodes de cette classe permettent de calculer la position des outils, déterminer l'outil optimum (Figure 2.49).

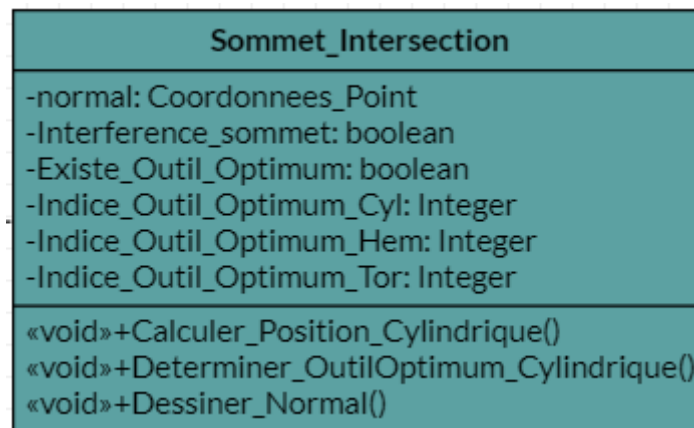


Figure 2.49. Classe « Sommet d'intersection ».

- ✚ **Classe « Coordonnées point »** : cette classe regroupe les coordonnées d'un point x, y, z (Figure 2.50).

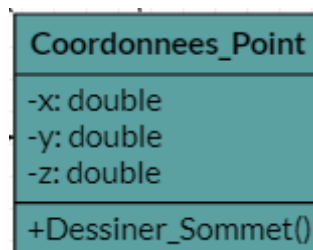


Figure 2.50. Classe « Coordonnées point ».

- ✚ **Classe « Triangle »** : c'est une classe qui englobe les différentes caractéristiques des triangles (sommets, aire, normale, etc.). Elle permet de calculer les paramètres du triangle (Figure 2.51).

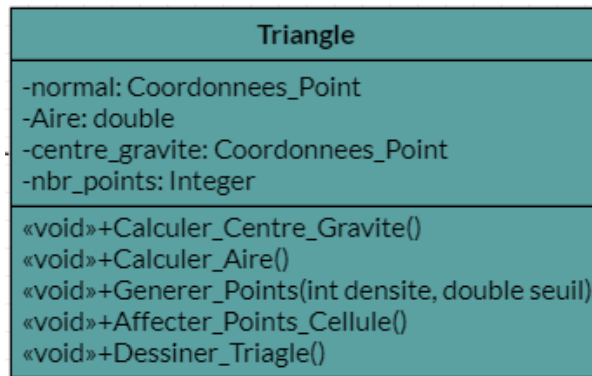


Figure 2.51. Classe « Triangle ».

- + **Classe « Modèle STL »** : c'est la classe qui regroupe toutes les informations concernant le modèle STL telles que les sommets, les triangles, les informations sur le brut. Les méthodes de cette classe permettent de récupérer le fichier STL, avec ces paramètres, créer les cellules, etc. (Figure 2.52).

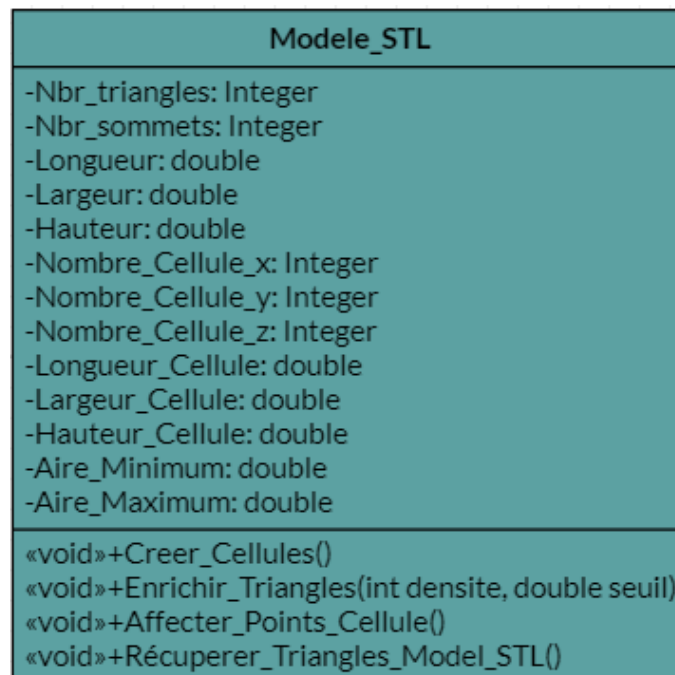


Figure 2.52. Classe « Modèle STL».

- + **Classe « Brut »** : c'est la classe qui contient les limites du modèle (Figure 2.53).

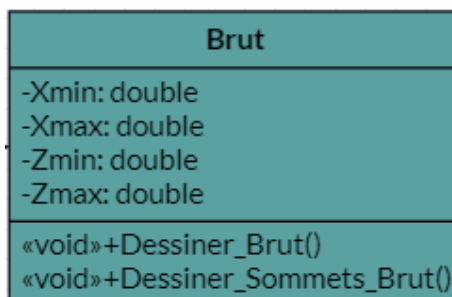


Figure 2.53. Classe « Brut ».

- ✚ **Classe « Cellule élémentaire »** : cette classe est définie par son brut. Elle contient les points d'enrichissement. Elle permet de vérifier l'interférence des trois formes d'outils (Figure 2.54).

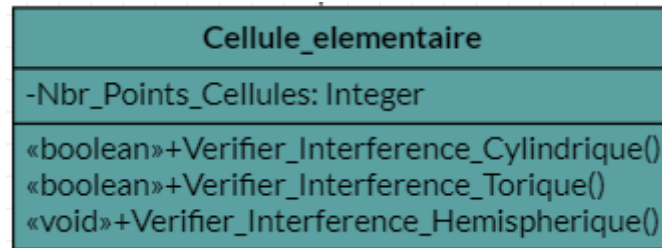


Figure 2.54. Classe « Cellule élémentaire ».

- ✚ **Classe « Couleur »** : cette classe regroupe les couleurs primaires. Elle permet d'affecter une couleur aux différents objets géométriques (sommets, triangles, brut, cellules, etc.) (Figure 2.55).

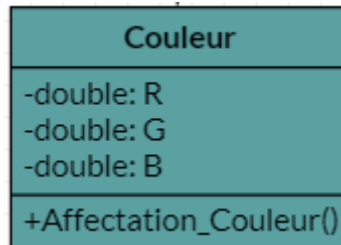


Figure 2.55. Classe « Couleur ».

4. Conclusion

Dans ce chapitre, nous avons présenté la conception de l'application à développer, ou nous avons défini la problématique du projet, ainsi que l'objectif principal à atteindre. Par la suite, nous avons présenté toutes les étapes suivies et toutes les solutions proposées. 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, l'implémentation et la mise en œuvre de notre application seront traitées à travers deux exemples de validation.

Chapitre 3 : Implémentation et validation

Introduction

Ce chapitre représente la dernière partie de ce rapport. Il traite la phase qui a pour objectif l'implémentation de notre application. Nous débutons, tout d'abord, par définir les outils de développement de notre application ainsi qu'une vue générale détaillée en utilisant des captures d'écrans. Ensuite, une validation est réalisée par un ensemble de tests qui ont été préparés pour chaque fonctionnalité de l'application.

1. Présentation des outils utilisés

Notre projet s'intègre dans le cadre de développement d'une plateforme logicielle pour la production des pièces de formes complexes initié par l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO » de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA ». Les outils utilisés pour le développement de la plateforme sont le langage C++ et la bibliothèque graphique OpenGL (Figure 3.1).



Figure 3.1. Logos des différents outils utilisés.

1.1. Présentation du langage C++

C++ est un langage de programmation compilé créé initialement par Bjarne Stroustrup en 1983. Il est un langage sous de multiples paradigmes, dont la programmation procédurale et la programmation générique. L'une des plus célèbres est la Programmation Orientée Objet (POO) qui donne une structure claire aux programmes et permet la réutilisation du code, réduisant ainsi les coûts de développement. C'est un langage dit « de bas niveau », très puissant et particulièrement rapide. C'est un langage multiplateforme qui peut être utilisé pour créer des applications hautes performances de plus le langage C++ donne aux programmeurs un haut niveau de contrôle sur les ressources système et la mémoire. C++ est actuellement le 3^{ème} langage le plus utilisé au monde. Il est portable : un même code source peut théoriquement être transformé sans problème en exécutable sous Windows, Mac OS et Linux, Ils existent de nombreuses bibliothèques pour le C++, en les combinant il est possible de créer des programmes 3D, réseaux, audio, fenêtrés, etc [29].

1.2. Présentation d'OpenGL

OpenGL (Open Graphics Library) est une interface logicielle graphique multiplateformes open-source. L'interface se compose de plus de 250 appels de fonctions différentes qui peuvent être utilisées pour visualiser et rendre des scènes complexes en 2D et

en 3D à partir de primitives géométriques simples telles que des points, des lignes et des polygones. Ils existent également des routines pour le rendu des scènes avec un contrôle sur l'éclairage, les propriétés de la surface de l'objet, la transparence, l'anticrénelage et le mappage de texture. Du fait de ses performances, l'API OpenGL est aussi présente dans différents domaines tels que jeux-vidéo ludiques, systèmes embarqués et téléphones portables. OpenGL est conçue comme une interface rationalisée et indépendante du matériel à implémenter sur de nombreuses plateformes matérielles graphiques différentes [30].

1.3. Embarcadero C++ Builder

Embarcadero C++Builder est un IDE intégré avancé pour le langage C++ moderne avec des fonctionnalités robustes permettant de créer des applications natives, incroyablement rapides et ultra-réactives pour les plus importantes plateformes. Il est le moyen le plus rapide pour construire et mettre à jour des applications hyper connectées, intégrant des jeux étendus de données et graphiquement attractives pour Windows 10, Mac, systèmes mobiles, l'Internet des objets et bien d'autres environnements [31].

2. Présentation de l'environnement CFAO

C'est une plateforme logicielle dédiée à la production numérique des pièces complexes sur fraiseuses numériques 03-axes et 05-axes. Cette plateforme est composée de deux parties (Figure 3.2) :

➤ **Partie visualisation** : elle permet de visualiser tous les objets géométriques en 3D (points, surfaces, outils, machine, simulation, trajet d'usinage, etc.) en utilisant la bibliothèque graphique « OpenGL ». Dans cette partie, les objets visualisés peuvent subir diverses transformations géométriques telles que translations, rotations, projections, zoom et changement d'échelles.

➤ **Partie manipulation** : cette partie est composée d'une barre de menu et d'un ensemble de boutons de manipulation des différents paramètres des modèles CAO. Dans cette partie, sont lancés tous les modules logiciels intégrés à la plateforme logicielle.

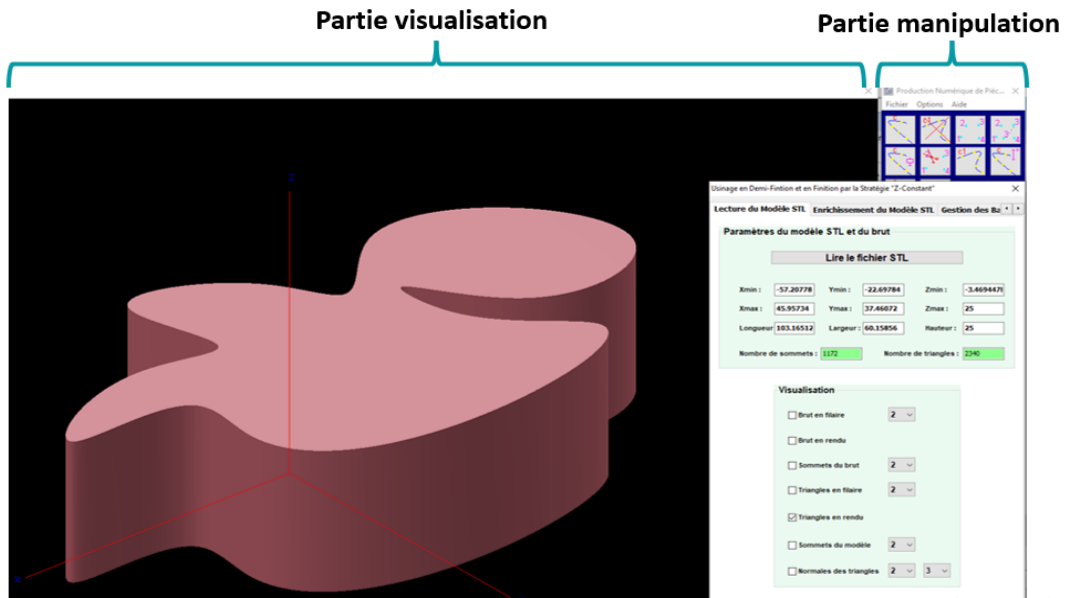


Figure 3.2. Fenêtres principales de la plateforme logicielle.

3. Présentation du module logiciel développé

Pour lancer le module logiciel développé et intégré à la plateforme logicielle, il faut passer par « Option », « Projet Socio-Economique (Production de Pièces Complexes en 03-Axes) » et « 1/2 Finition et Finition en 03-Axes par « Z-Constant » » (Figure 3.3).

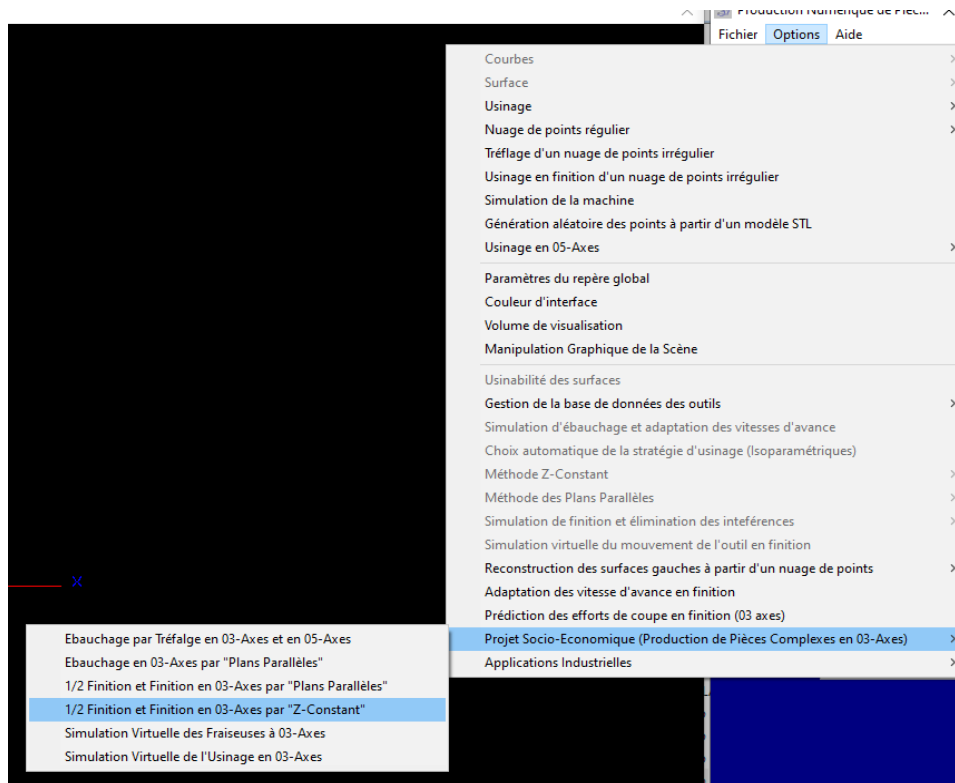


Figure 3.3. Lancement du module logiciel développé.

Pour atteindre l'objectif principal qui est la génération de la trajectoire d'outil, le module logiciel développé est composé de sept (07) onglets (Figure 3.4) pour l'activation des fonctionnalités intégrées à la plateforme logicielle. Les différents onglets seront détaillés dans les paragraphes suivants :

- Onglet 1 : Lecture du Modèle STL.
- Onglet 2 : Enrichissement du Modèle STL.
- Onglet 3 : Génération des Contours d'Usinage.
- Onglet 4 : Gestion des Bases des Données des Outils.
- Onglet 5 : Positionnement des Outils.
- Onglet 6 : Outils Optimums et Trajets d'Outils.
- Onglet 7 : Simulation des Trajets d'Outils

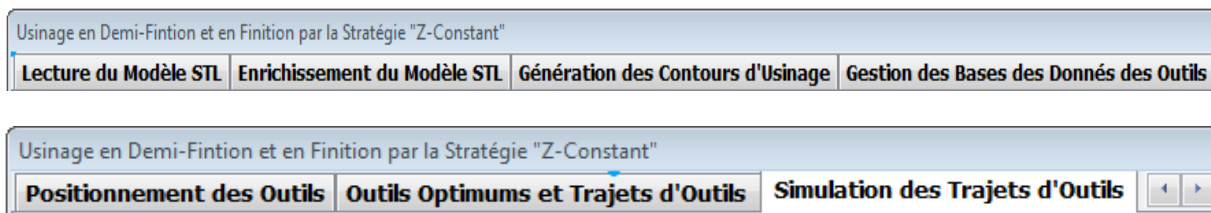


Figure 3.4. Onglets du module logiciel développé.

3.1. Lecture du Modèle STL

Le premier onglet « Lecture du Modèle STL » permet de lire le fichier STL contenant l'ensemble des paramètres des triangles (coordonnées des sommets et normale des triangles) (Figure 3.5). Par un clic sur le bouton « Lire le fichier STL », on sélectionne le fichier désiré. La syntaxe du fichier est vérifiée. Une fois le fichier est lu, il est possible de visualiser et d'afficher :

- ❖ Les paramètres de la pièce brute : Xmin, Ymin, Zmin, Xmax, Ymax, Zmax, longueur, largeur et 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 filaires et rendu.
- ❖ Les sommets des triangles du modèle.
- ❖ Les normales des triangles du modèle.

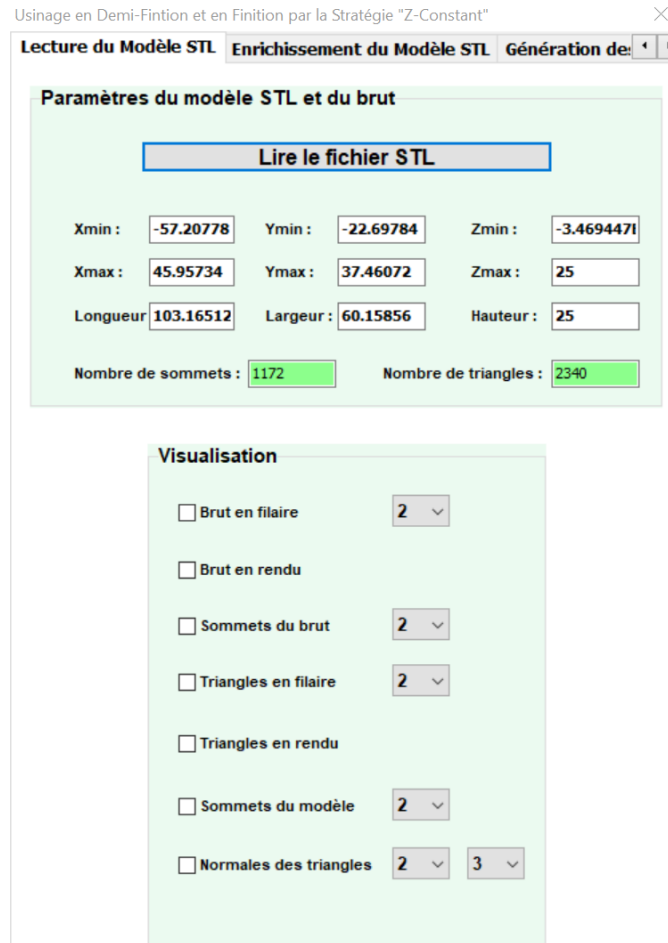


Figure 3.5. Onglet « Lecture du Modèle STL ».

3.2. Enrichissement du Modèle STL

Dans cet onglet, l'utilisateur doit effectuer trois étapes (Figure 3.6). La première consiste à créer les cellules en introduisant le nombre de cellules suivant les axes X, Y et Z et en cliquant sur le bouton « Créer Les Cellules ». La deuxième étape consiste à enrichir les triangles du modèle STL par l'insertion d'un ensemble de points pour chaque triangle. Pour cela, l'aire minimum et l'aire maximum sont affichées par le système pour que l'utilisateur fixe le seuil de l'aire des triangles puis il fixe aussi la densité et il clic sur le bouton « Enrichir les triangles ». La dernière étape consiste à affecter les nouveaux points insérés aux cellules par un simple clic sur le bouton « Affecter les points aux cellules ».

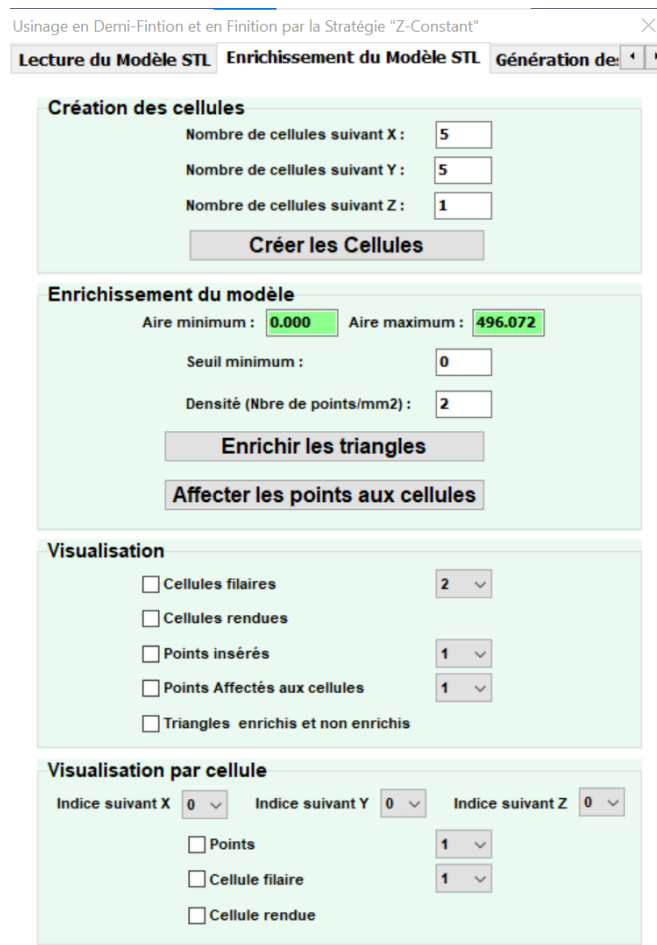


Figure 3.6. Onglet « Enrichissement du Modèle STL ».

A ce niveau, Il est possible de visualiser :

- ❖ Cellules en deux modes (rendu et filaire).
- ❖ Points insérés.
- ❖ Points affectés aux cellules.
- ❖ Triangles enrichis et non enrichis.

De même il est possible de visualiser pour chaque cellule :

- ❖ Points affectés.
- ❖ Cellule en deux modes (rendu et filaire).

3.3. Génération des Contours d'Usinage

Dans cet onglet, l'utilisateur doit effectuer deux opérations (Figure 3.7). La première opération consiste à générer les plans de coupe après la spécification de la profondeur de passe suite à un clic sur le bouton « Générer les plans de coupe ». Une fois générés, le nombre de

plans sera affiché et la visualisation des plans sera possible. Ensuite le bouton « Générer les contours d'usinages » permet de générer les contours pour chaque plan. Par la suite, le nombre de contours et le nombre de segments sont affichés et il sera possible de visualiser les contours.

Usinage en Demi-Finition et en Finition par la Stratégie "Z-Constant"

Enrichissement du Modèle STL Génération des Contours d'Usinage

Plan de coupe

Profondeur de passe (mm) : 1

Générer les plans de coupe

Nombre de plans : 26

Contours d'usinage

Générer les contours d'usinage

Nombre de contours : 26 Nombre de segments : 29298

Visualisation globale

Plans 2

Contours 2

Visualisation partielle

Indice du plan : 0

Plan de coupe 2

Contour 2

Normales des sommets 2 2

Figure 3.7. Onglet « Génération des Contours d'Usinage ».

De même l'utilisateur peut visualiser selon l'indice du plan :

- ❖ Plan de coupe.
- ❖ Contour.
- ❖ Normales des points de contact.

3.4. Gestion des Bases des Données des Outils

Ce troisième onglet est réservé à la gestion des bases des données d'outils (Figure 3.8). L'utilisateur peut manipuler facilement les tables, il peut ajouter de nouveaux outils, en choisissant au début la forme d'outil puis en spécifiant tous les paramètres géométriques et technologiques par un simple clic sur « Ajouter outil ». Pour la suppression, il suffit de sélectionner directement l'outil sur la table puis un clic sur « Supprimer l'outil ».

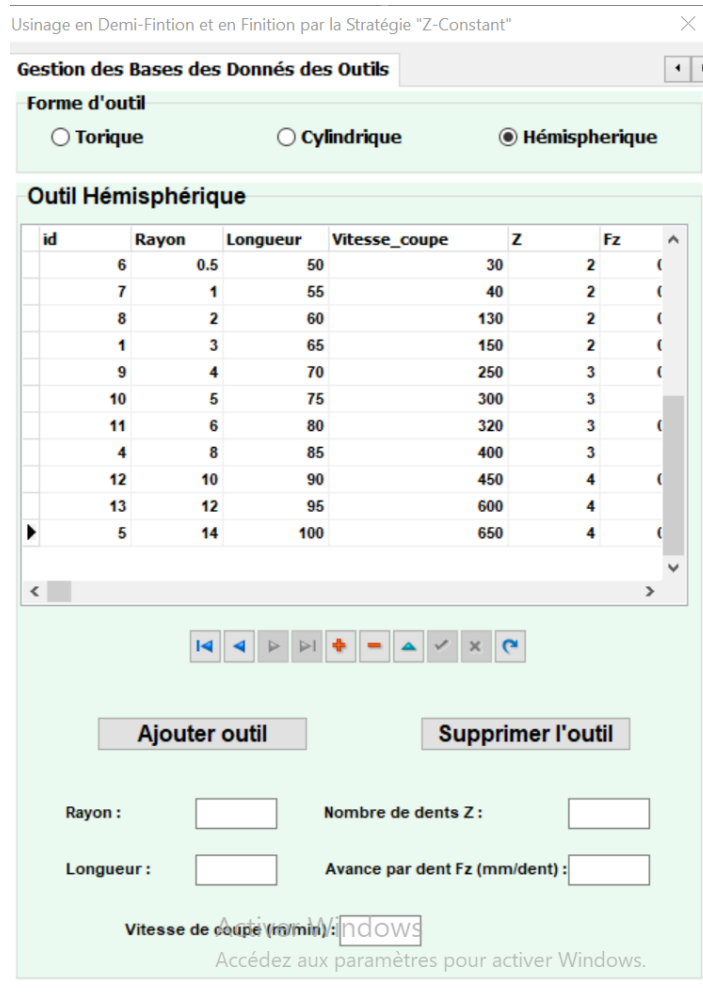


Figure 3.8. Onglet « Gestion des bases des Données des Outils ».

3.5. Positionnement des Outils

L'intérêt principal de cet onglet est le test manuel du positionnement et la détermination des outils optimums des différentes formes d'outils (Figure 3.9). Tout d'abord, l'utilisateur doit choisir la stratégie d'usinage, si c'est la demi-finition est sélectionnée, la surépaisseur doit être spécifiée. Puis, un simple clic sur le bouton « Fixer la surépaisseur ». Si c'est la finition est sélectionnée, alors la surépaisseur est fixée automatiquement égale à 0. L'utilisateur peut choisir la forme et l'indice d'outil souhaités qui sont récupérés à partir d'une base de données d'outils, et le positionner en chaque point de contact sélectionné. À ce niveau, le rayon et la longueur de l'outil seront visualisées ainsi que :

- ❖ Plan selon son indice sélectionné.
- ❖ Contour selon son indice sélectionné.
- ❖ Point de contact selon son indice sélectionné.
- ❖ Outil en deux modes filaire et rendu.

- ❖ Enveloppes de l'outil (corps et partie active).
- ❖ Cellules en chevauchement avec l'outil.
- ❖ Points enrichis des cellules.
- ❖ Normales des sommets.
- ❖ Existence ou absence de l'interférence.

Le bouton « Déterminer l'outil optimum » permet de déterminer l'outil optimum pour un point de contact, contour, plan choisi en parcourant la base des données d'outils du plus grand au plus petit jusqu'à trouver l'outil hors interférence et son indice sera affiché. On peut aussi déterminer tous les outils optimaux pour tous les points de contact du contour sélectionné avec le bouton « Déterminer tous les outils optimaux ». Ensuite, il est possible de visualiser :

- ❖ Tous les outils optimaux pour tous les points de contact du contour.
- ❖ Outil optimum pour chaque point de contact sélectionné.
- ❖ Points de contact usinables.
- ❖ Points de contact non usinables.

La dernière partie de cet onglet consiste à choisir le nombre d'outils à utiliser pour chaque contour en sélectionnant le nombre (1, 2 ou 3). Puis, par un simple clic sur le bouton « Affecter les outils aux sommets », les nouveaux outils sont affectés à tous les points de contact. Par la suite, il est possible de visualiser les outils.

Figure 3.9. Onglet « Positionnement des Outils ».

3.6. Outils Optimums et Trajets d'Outils

Le but de cet onglet est de calculer les outils optimums de toute la surface puis limiter le nombre d'outils à utiliser en intégrant la contrainte du nombre maximum d'outils à utiliser (capacité du magasin d'outil de la machine), ensuite de générer les trajets des outils avant de passer à la simulation (Figure 3.10).

- Bouton « Associer la forme de l'outil aux contours » sert à associer pour chaque contour une forme d'outils (cylindrique, torique ou hémisphérique), le nombre de contours usinés avec chaque forme sera affiché. On peut visualiser les contours associés.
- Bouton « Fixer le nombre d'outils par contour » sert à limiter une première fois le nombre d'outils en sélectionnons (1, 2 ou 3) outils pour chaque contour puis par un clic sur ce bouton.

A ce niveau il est possible de visualiser :

- ❖ Tous les outils optimums du contour sélectionné.
 - ❖ Outil optimum de chaque point de contact sélectionné.
 - ❖ Paramètres géométriques de l'outil optimum.
 - ❖ Nombre total des outils.
 - ❖ Nombre d'outils associé à chaque forme.
-
- Bouton « Réaffecter les outils aux points de contact » sert à limiter le nombre d'utilisation de chaque outil pour tout le trajet et à réaffecter les nouveaux outils en chaque point de contact.
 - Bouton « Générer les points de passage CL » sert à générer tous les points de passage des outils. L'utilisateur peut visualiser les points extrémités outils CL « Cutter Location ».
 - Bouton « Ajouter les points d'engagement et de dégagement » permet d'ajouter à chaque portion usinable deux points d'engagement et deux points de dégagement en spécifiant les distances par l'utilisateur.

A cette étape, avant la génération du trajet, le temps de changement d'outils doit être spécifié.

- Bouton « Générer trajet sans optimisation » sert à déterminer le trajet d'outils sans tenir compte du séquençement des outils.
- Bouton « Générer trajet avec optimisation » sert à créer le trajet d'outils par la prise en compte du séquençement d'outils. L'usinage est réalisé du plus grand outil au plus petit outil.

Le temps total d'usinage sera affiché en minutes.

Usinage en Demi-Finition et en Finition par la Stratégie "Z-Constant" ☒

Positionnement des Outils | **Outils Optimums et Trajets d'Outils** ◀ ▶

Association des formes d'outils aux contours

Associer la forme de l'outil aux contours

Nombre des contours avec forme cylindrique : Contours associés

Nombre des contours avec forme torique : Contours associés

Nombre des contours avec forme hémisphérique : Contours associés

Sélection du nombre d'outils par contour

Nombre d'outils par contour

Fixer le nombre d'outils par contour

Plans Contours Sommets

Tous les outils Un outil

Paramètres géométriques de l'outil optimum

Forme : Indice : Rayon : Longueur :

Limitation du nombre d'outils pour le trajet

Nombre total des outils :

Nombre d'outils cylindriques : Nombre à utiliser :

Nombre d'outils hémisphériques : Nombre à utiliser :

Nombre d'outils toriques : Nombre à utiliser :

Réaffecter les outils aux points de contact

Trajets d'outils

Générer les points de passage CL Points CL

Distance d'engagement (mm) : Distance de dégagement (mm) :

Distance de sécurité (mm) :

Ajouter les points d'engagement et de dégagement

Temps de changement d'outils (s) : **Générer trajet sans optimisation** temps d'usinage (min)

Générer trajet avec optimisation

Figure 3.10. Onglet « Outils Optimums et Trajets d'Outils ».

3.7. Simulation des Trajets d'Outils

Le dernier onglet permet de simuler virtuellement les mouvements des outils par rapport à la pièce à usiner (Figure 3.11). Des manipulations sont intégrées à cet onglet tel que le lancement, la pause, et la fin qui permet d'afficher l'état finale de l'outil. L'utilisateur peut aussi manipuler la vitesse d'affichage et l'incrément d'avance des outils.

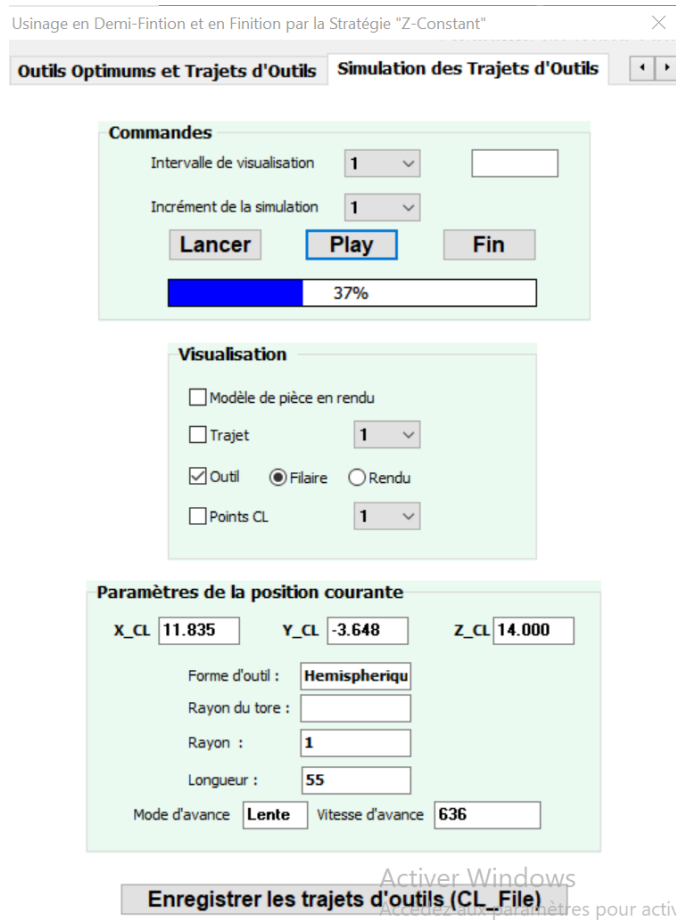


Figure 3.11 Onglet « Simulation des Trajets d’Outils ».

On peut visualiser au fur et à mesure :

- ❖ Le pourcentage d’avancement.
- ❖ Les triangles du modèle en rendu.
- ❖ La trajectoire d’usinage.
- ❖ Le déplacement de l’outil.
- ❖ Les coordonnées des points CL.
- ❖ Les paramètres d’outils.

Finalement, le bouton « Enregistrer les trajets d’outils (CL_File) » permet l’enregistrement des trajets d’outils dans un fichier texte.

4. Test et validation

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

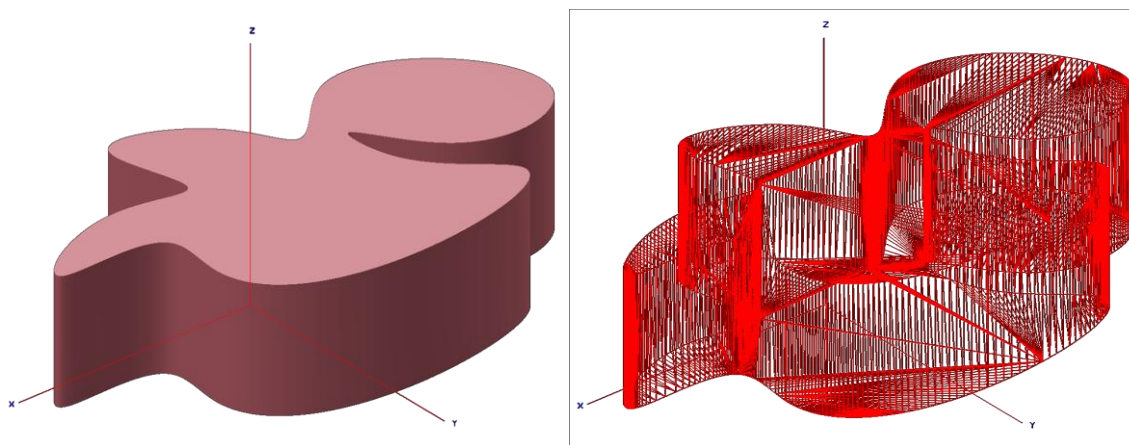
- ❖ ▪ Windows 10.
- ❖ ▪ Processeur Intel(R) Core (TM) i9-10900K CPU @ 3,70GHz.
- ❖ ▪ Mémoire installée (RAM) : 32.00 Go.
- ❖ ▪ Type de système : système d'exploitation 64 bits.

La validation du travail consiste à tester les approches proposées lors de la phase de conception en utilisant un modèle STL d'une pièce complexe issue d'une conception générée dans un logiciel de CAO spécialisé dans la conception des pièces de formes complexes.

Dans ce qui suit, nous allons présenter les résultats pour deux modèles de pièces, Le premier modèle vise à montrer les différentes étapes de génération du trajet d'outils avec des données simples, Par contre, pour le deuxième modèle on a pris des paramètres plus réalistes pour avoir des résultats plus précis.

4.1 Premier modèle STL

L'exemple considéré est représenté par la Figure 3.12. Dans ce qui suit, nous allons montrer toutes les étapes permettant de générer le trajet d'usinage pour l'opération de demi-finition et de finition de cette pièce.



a. Triangle en rendu

b. Triangles en filaire.

Figure 3.12. Premier modèle STL.

➤ **Étape 1** : lecture du fichier STL et calcul des limites du brut (Figure 3.13).

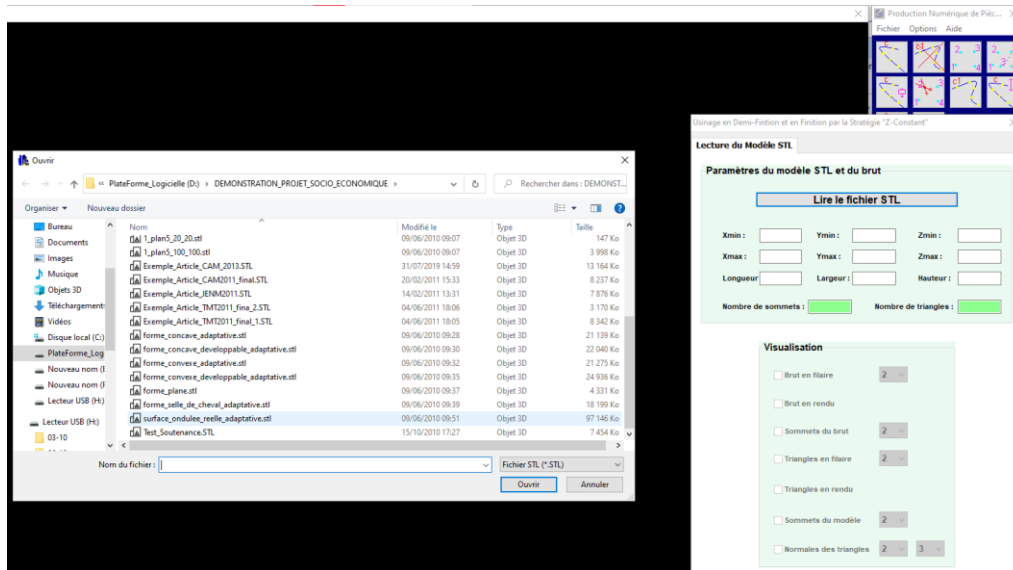


Figure 3.13. Lecture du modèle STL.

Les résultats obtenus lors de la lecture sont donnés dans le Tableau 3.1 :

Tableau 3.1. Paramètres du modèle STL.

Xmin : -57,207	Ymin : -22,697	Zmin : -3,469
Xmax : 45,957	Ymax : 37,460	Zmax : 25
Longueur : 103,165	Largeur : 60,158	Hauteur : 25
Nombre de sommets : 1172		Nombre de triangles : 2340

La Figure 3.14 montre le modèle STL de la pièce considérée et son brut.

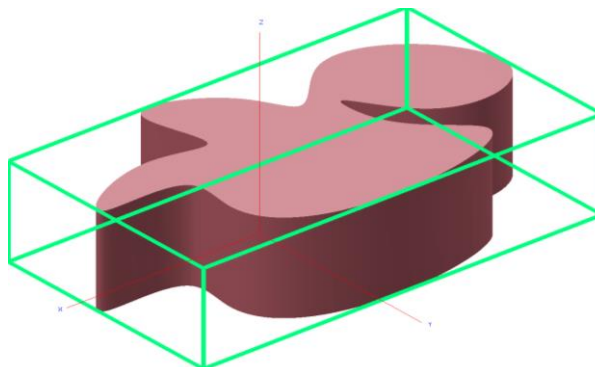


Figure 3.14. Brut du modèle STL.

- **Etape 2 :** création des cellules en subdivisant le brut en 03 cellules suivant l'axe X, 03 cellules suivant l'axe Y et 03 cellules suivant l'axe Z (Figure 3.15).

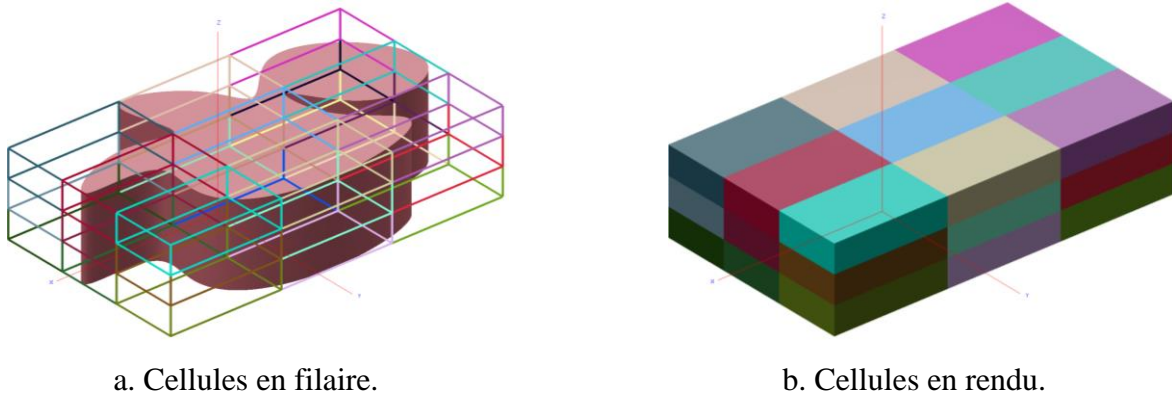


Figure 3.15. Création des cellules.

Après la création des cellules, on va enrichir le modèle STL en spécifiant la densité des points pour une aire de 1mm^2 et le seuil minimum de l'aire des triangles qui doit être comprise entre l'aire minimum (0.000mm^2) et l'aire maximum ($496,072\text{mm}^2$) des triangles à enrichir. Pour cet exemple, la densité est prise égale à 30point/mm^2 . La Figure 3.16 montre les points insérés aux triangles et leurs affectations aux cellules.

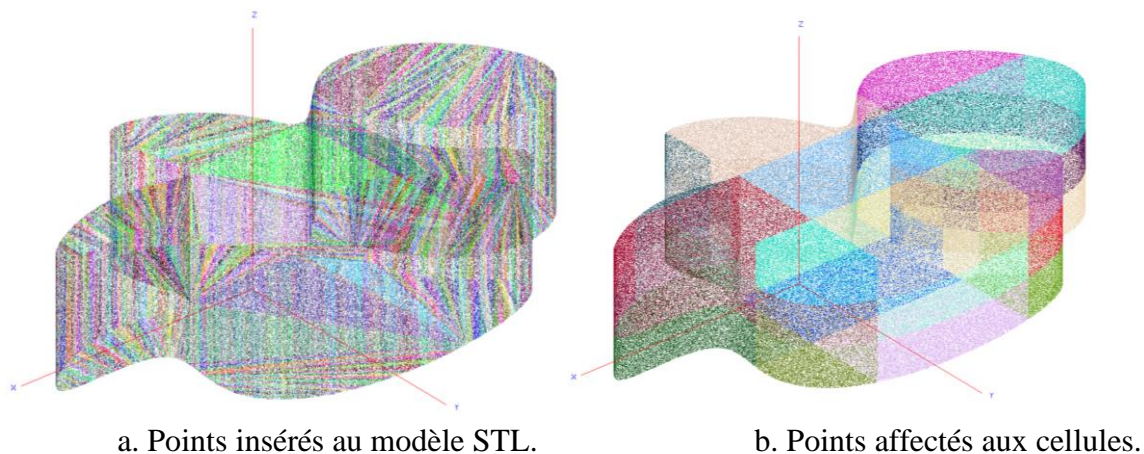
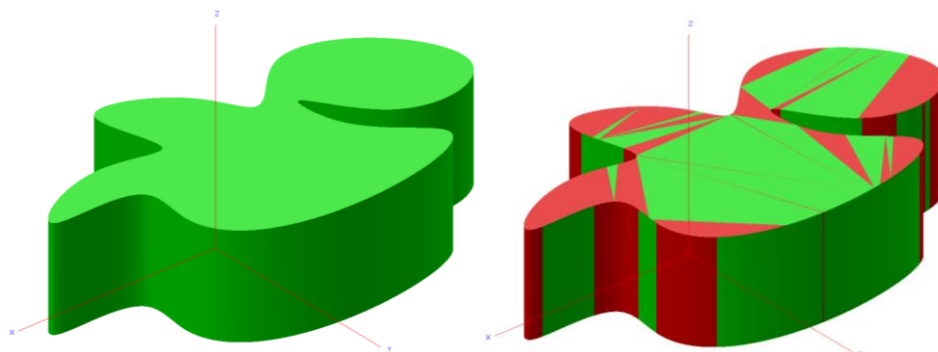


Figure 3.16. Affectation des points aux cellules.

Si la valeur du seuil est égale à 0, alors tous les triangles du modèle STL seront enrichis. Sinon, les triangles avec une aire inférieure au seuil ne seront pas enrichis. La Figure 3.17 montre deux couleurs : vert pour les triangles enrichis et rouge pour les triangles non enrichis.



Seuil=0 mm²

Seuil=20 mm²

Figure 3.17. Triangles enrichis et non enrichis.

- **Etape 3 :** génération des plans et des contours avec une profondeur de passe égale à 5mm. Les résultats obtenus sont (Figure 3.18 et Figure 3.19) :

Nombre de plans : 6

Nombre de contours : 6

Nombre de segments : 5860

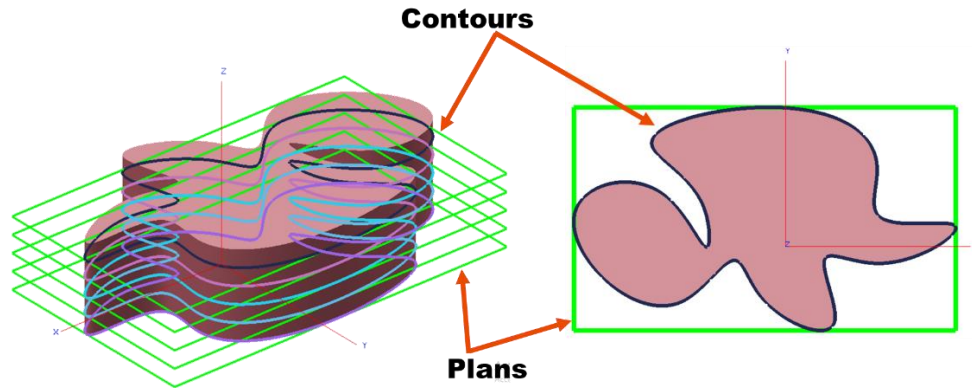


Figure 3.18. Plans et contours de la pièce.

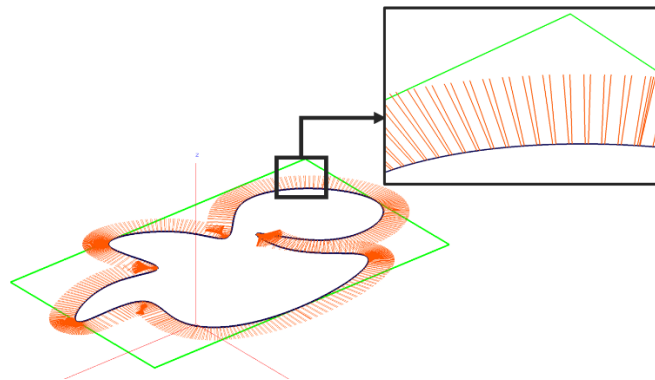


Figure 3.19. Normales des points de contact d'un contour.

- **Etape 4 :** détermination de l'opération d'usinage en spécifiant la surépaisseur. Puis, les paramètres et le positionnement d'outil pour un point de contact donné en fonction du choix de l'utilisateur. Par la suite, détermination de l'outil optimum en chaque point de contact du contour sélectionné. Le Tableau 3.2 montre le positionnement des trois formes d'outils pour le cas de l'usinage en demi-finition et en finition. Par la suite, en chaque point de contact, les cellules de chevauchement sont calculées (Figure 3.20).

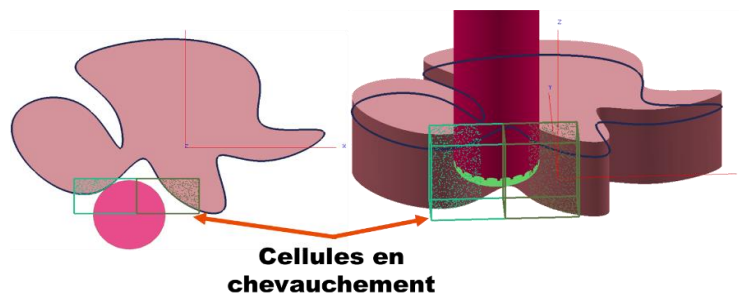
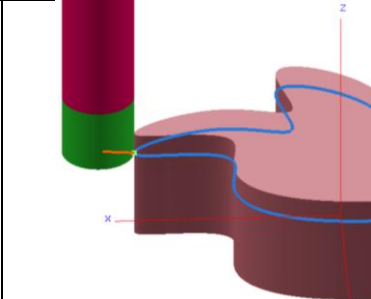
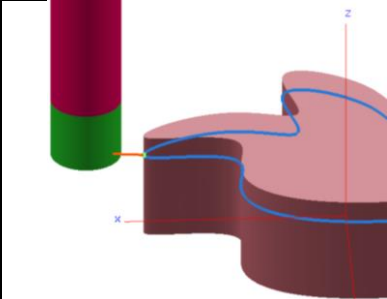
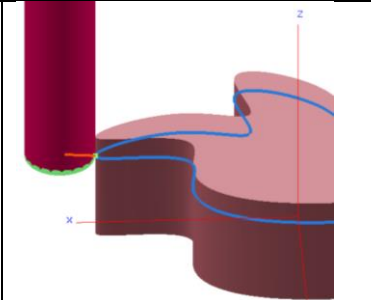
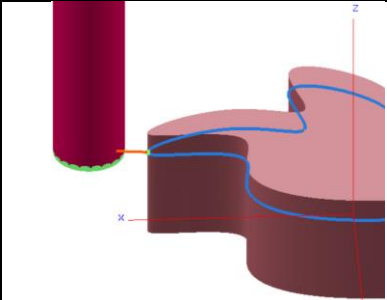
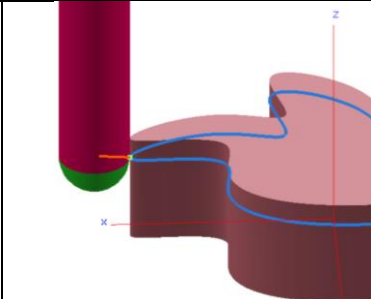
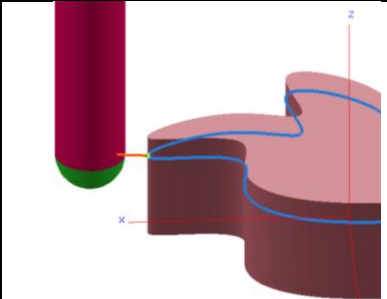


Figure 3.20. Cellules en chevauchement.

Tableau 3.2. Positionnement des trois formes d'outils.

Forme d'outil	Surépaisseur = 0 mm	Surépaisseur = 5 mm
Cylindrique		
Torique		
Hémisphérique		

- **Etape 5:** association de la forme d'outil aux contours. Pour ce modèle, les résultats sont : 0 contours avec forme cylindrique, 0 contours avec forme torique et 06 contours avec forme hémisphérique.

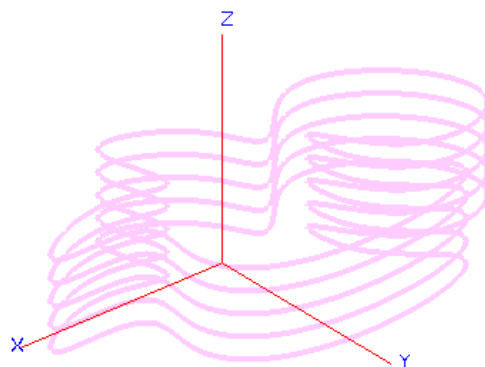


Figure 3.21. Association de la forme d'outil aux contours.

- **Etape 6:** sélection du nombre d’outils par contour. La Figure 3.22 montre la sélection d’un, de deux et de trois outils.

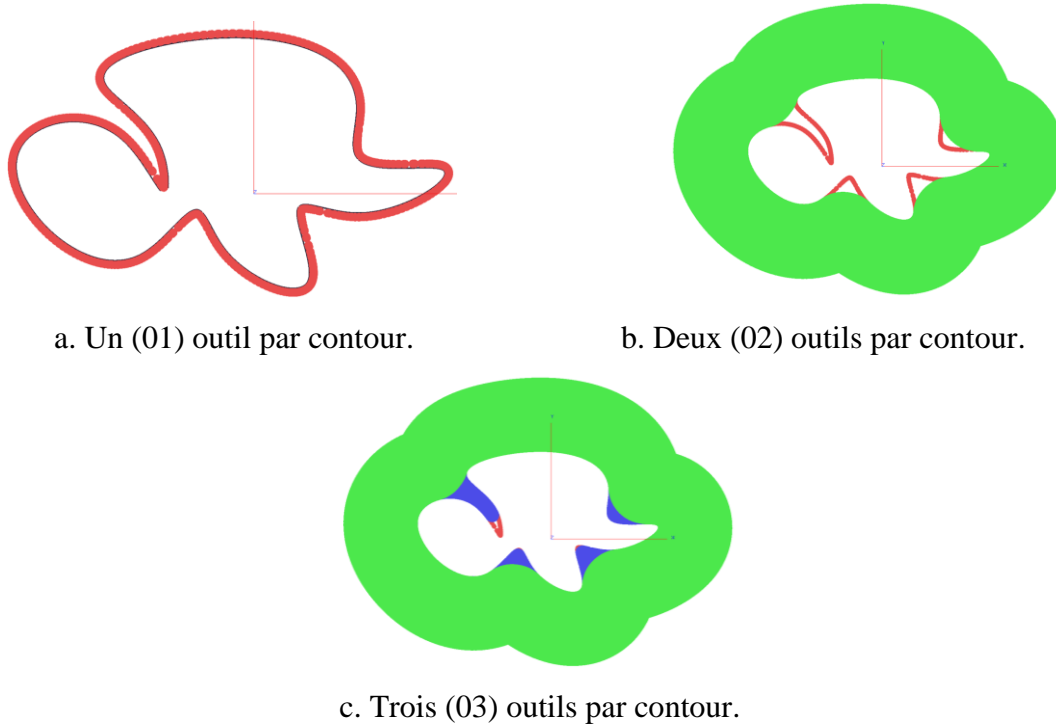
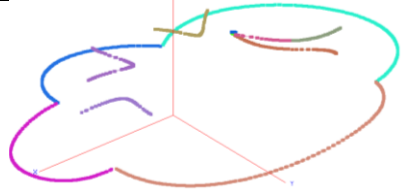


Figure 3.22. Visualisation des outils pour le premier plan.

- **Etape 7:** limitation du nombre outils à utiliser en fixant le nombre total d’outils qui sera placé dans le magasin d’outils de la fraiseuse. Les différents cas possibles pour notre modèle de validation sont présentés par le tableau suivant :

Tableau 3.3. Limitation du nombre d’outils.

Nombre d’outils par contour	Nombre d’outils utilisés			Nombre d’outils à utiliser			Visualisation des Points CL
	Cyl	Tor	Hem	Cyl	Tor	Hem	
1	0	0	1	0	0	1	
2	0	0	2	0	0	2	

3	0	0	3	0	0	3	
---	---	---	---	---	---	---	--

- **Etape 8:** insertion des points d'engagement et de dégagement pour chaque contour usinable en introduisant la distance d'engagement, la distance de dégagement et la distance de sécurité. Ces distances sont fixées égales à 10 mm, 10 mm et 20 mm.

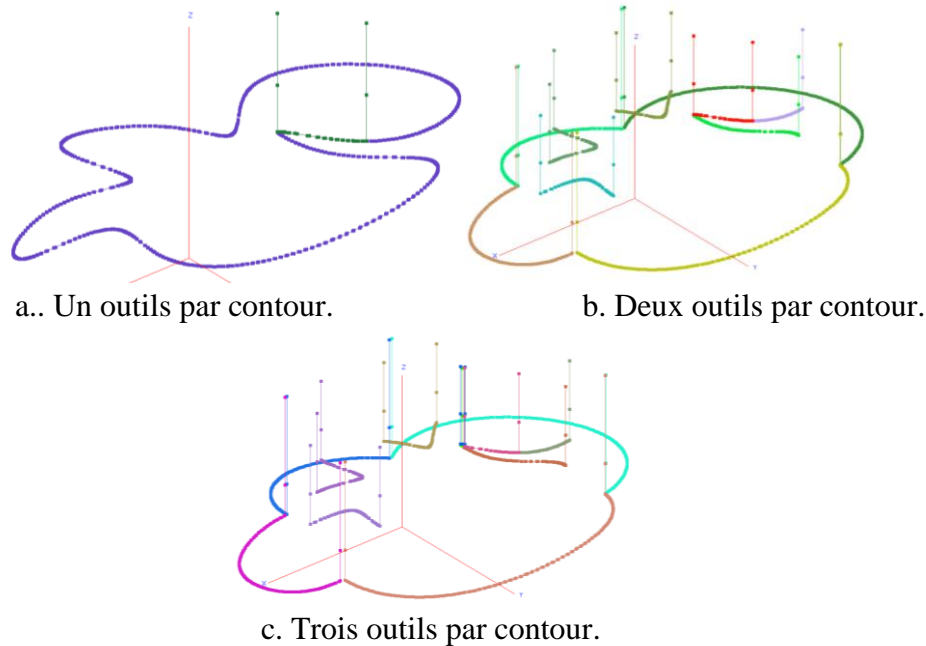


Figure 3.23. Insertion des points d'engagement et de dégagement.

- **Etape 9 :** génération du trajet d'outils. Pour cela, deux modes sont possibles :
 - **Sans optimisation** : le séquençement d'outils se fait par rapport aux outils affectés aux contours.
 - **Avec optimisation** : le séquençement d'outils se fait par ordre décroissant en commençant du plus grand outil au plus petit outil.
- **Etape 10 :** simulation des mouvements des outils. Lors de la simulation, en chaque position de passage de l'outil, les paramètres géométriques de l'outil, les coordonnées du point extrémité « CL », l'outil et les trajets d'outils sont affichés (Figure 3.24). La couleur verte représente le trajet où la vitesse est programmée, et la couleur rouge représente le déplacement hors matière où la vitesse est rapide.

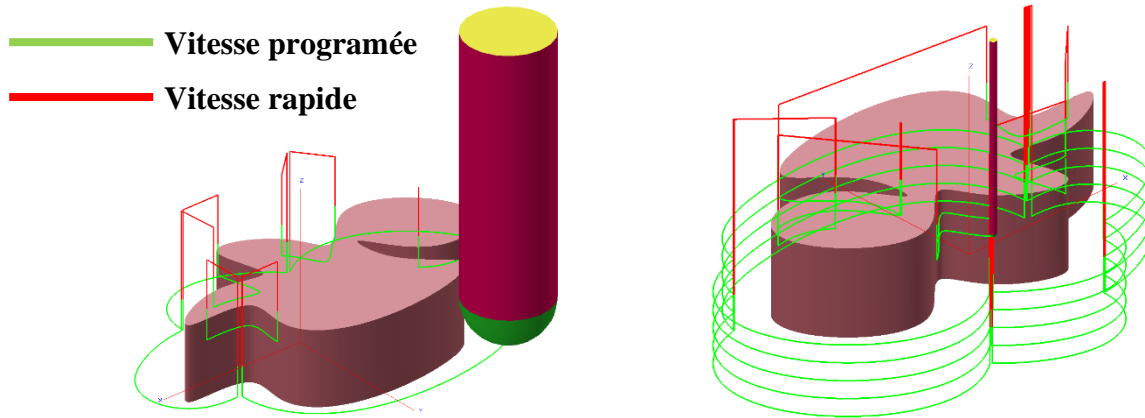


Figure 3.24. Simulation des trajets d'outils.

- **Etape 11 :** enregistrement du trajet d'outils « CL-FILE » dans un fichier texte (Figure 3.25).

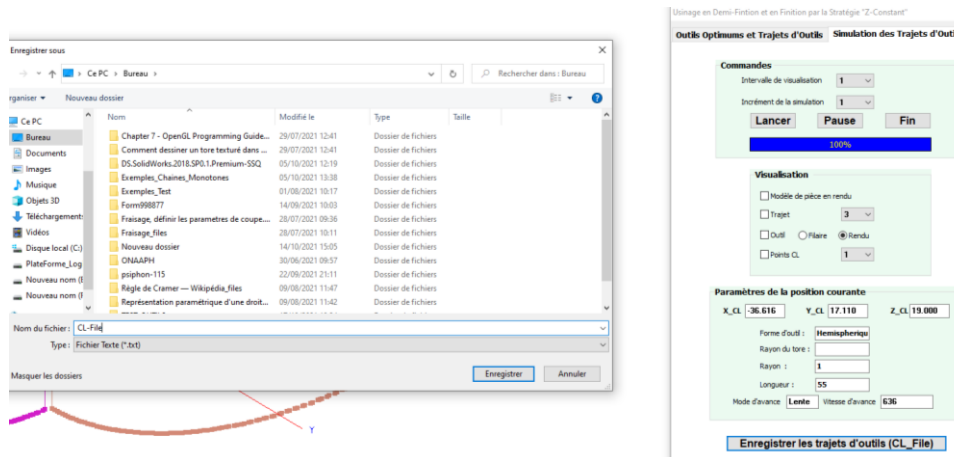


Figure 3.25. Enregistrement du fichier « CL-FILE ».

4.2. Deuxième modèle STL

Le modèle STL de la deuxième pièce est représenté par la Figure 3.26. Dans ce qui suit, nous allons montrer toutes les étapes permettant de générer le trajet d'usinage pour l'opération de demi-finition et de finition de cette pièce.

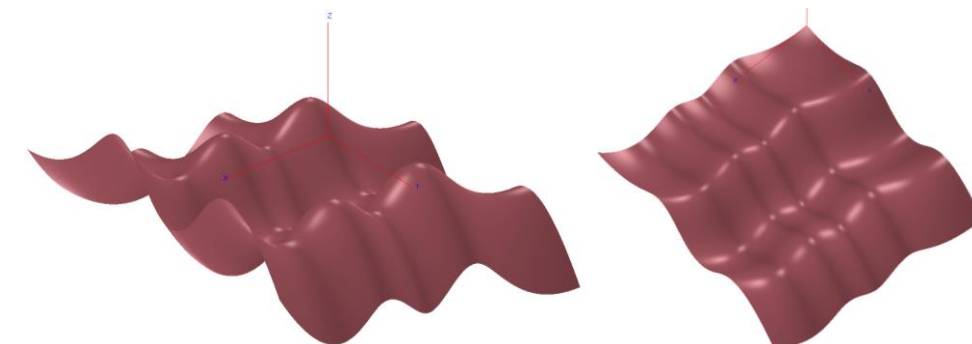


Figure 3.26. Deuxième modèle STL.

- **Etape 1 :** lecture du fichier STL et calcul des limites du brut donne les résultats suivants :

Tableau 3.4. Paramètres du modèle STL.

Xmin : 0	Ymin : 0	Zmin : 0
Xmax : 150	Ymax : 150	Zmax : 79,903
Longueur : 150	Largeur : 150	Hauteur : 79,903
Nombre de sommets : 191291		Nombre de triangles : 381462

- **Etape 2 :** subdivision du brut en cellules avec un nombre de 50 cellules suivant les trois axes X, Y et Z (Figure 3.27). Enrichissement du modèle STL avec une densité de 20 points/mm².

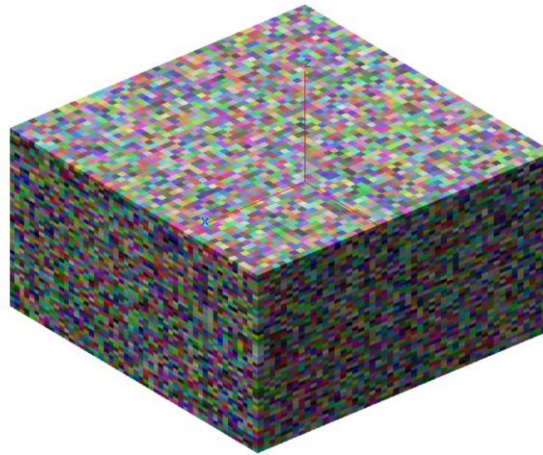


Figure 3.27. Cellules en rendu.

- **Etape 3 :** génération des plans et des contours de coupe avec une profondeur de passe égale à 1mm. Les résultats obtenus sont (Figure 3.28) :

Nombre de plans : 80 Nombre de contours : 1197 Nombre de segments : 180336

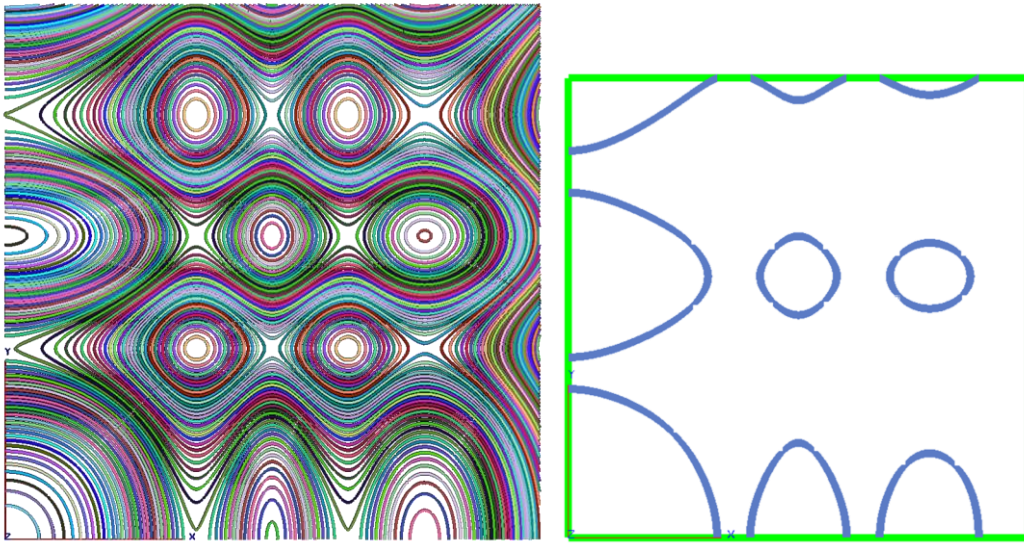


Figure 3.28. Contours du deuxième modèle.

- **Etape 4 :** usinage en finition.
- **Etape 5 :** association de la forme d'outil aux contours. Les résultats sont : 1197 contours avec forme cylindrique, 0 contours avec forme torique et 0 contours avec forme hémisphérique.
- **Etape 6 :** choix de 03 outils par contour.
- **Etape 7 :** limitation des outils à utiliser, Les résultats obtenus sont :

Nombre total des outils utilisés : 11. Avec 11 outils cylindriques, 0 outil torique et 0 outil hémisphérique.

On a effectué cette étape avec deux tests :

1^{er} test : nombre d'outils à utiliser : 3

2^{ème} test : nombre d'outils à utiliser : 11

- **Etape 8 :** insertion des points d'engagement et de dégagement. Les distances choisies dans le test de validation sont :

Distance d'engagement = 10mm

Distance de dégagement = 10mm.

Distance sécurité = 50mm.

- **Etape 09 :** génération des trajets obtenus dans l'étape 7. Les résultats obtenus pour les différents outils choisis sont :

❖ **Nombre d'outils = 3 :**

Sans optimisation : le temps d'usinage est égal à 80.35 minutes (Figure 3.29).

Temps de changement d'outils (s) :	<input type="text" value="2"/>	Générer trajet sans optimisation	temps d'usinage (min)
		Générer trajet avec optimisation	80.354918

Figure 3.29. Temps d'usinage avec 3 outils sans optimisation.

Avec optimisation : le temps d'usinage est égal à 18.15 minutes (Figure 3.30).

Temps de changement d'outils (s) :	<input type="text" value="2"/>	Générer trajet sans optimisation	temps d'usinage (min)
		Générer trajet avec optimisation	18.154918

Figure 3.30. Temps d'usinage avec 3 outils avec optimisation.

❖ **Nombre d'outils = 11 :**

Sans optimisation : le temps d'usinage est égal à 203.69 minutes (Figure 3.31).

Temps de changement d'outils (s) :	<input type="text" value="2"/>	Générer trajet sans optimisation	temps d'usinage (min)
		Générer trajet avec optimisation	203.69029

Figure 3.31. Temps d'usinage avec 11 outils sans optimisation.

Avec optimisation : le temps d'usinage est égal à 23.89 minutes (Figure 3.32).

Temps de changement d'outils (s) :	<input type="text" value="2"/>	Générer trajet sans optimisation	temps d'usinage (min)
		Générer trajet avec optimisation	23.890297

Figure 3.32. Temps d'usinage avec 11 outils avec optimisation.

Avec ces résultats, nous avons une très importante réduction du temps d'usinage. Nous avons une réduction de **88,27%** lors de l'usinage avec 11 outils et une réduction de **77.41%** lors de l'usinage avec 03 outils. Ces résultats impliquent une productivité maximale avec des coûts minimums. Donc, l'objectif principal de notre projet est atteint.

- **Etape 10 :** simulation des mouvements de l'outil pour la finition 03-axes (Figure 3.33).

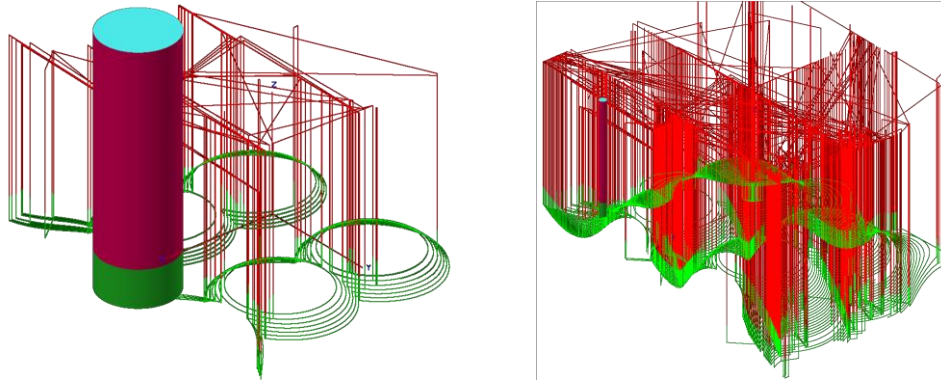


Figure 3.33. Simulation des mouvements d'outils.

- **Etape 11 :** enregistrement du trajet d'outils « CL-FILE ».

Conclusion

Au cours de ce chapitre, nous avons présenté notre module logiciel développé et les résultats des différents tests. Dans un premier temps, nous avons présenté les outils utilisés dans le développement de notre application ainsi que les interfaces pour décrire les différentes fonctionnalités. Dans un second temps, nous avons testé les différentes fonctionnalités de notre application à travers deux exemples de modèles STL.

Dans ce projet, nous avons mis en évidence l'importance de la combinaison des différentes formes et dimensions d'outils dans le processus de production des pièces de formes complexes sur des fraiseuses numériques à 03-axes et son impact sur la productivité et par conséquent les coûts.

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 génération de la trajectoire d'outils pour les opérations de demi-finition et de finition sur des fraiseuses numériques à 03-axes, des pièces complexes définies par leurs modèles « STL » en utilisant la stratégie d'usinage « Z-Constant ». L'objectif principal est la détermination de la combinaison optimale des outils cylindriques, toriques et hémisphériques optimums évitant les problèmes d'usinage (interférences et collisions) et réduisant les temps d'usinage et les coûts.

Dans un premier temps, nous avons mené une étude bibliographique sur le processus de production des pièces mécaniques de formes complexes, les formats d'échange de données, les fraiseuses numériques, la stratégie d'usinage « Z-Constant », les différentes formes d'outils et les problèmes d'usinage. Par la suite, nous avons présenté la solution proposée, les différentes approches utilisées et l'étude conceptuelle de l'application. Dans le dernier chapitre, nous avons présenté le module logiciel développé et ses fonctionnalités. A la fin, une validation est menée sur différents exemples de modèles de pièces.

Le résultat de notre travail est le développement d'un module logiciel graphique et interactif sous Windows et son l'intégration à la plateforme logicielle de l'équipe « CFAO ». Ses principales fonctionnalités sont :

- ✓ Récupération des principaux paramètres du modèle STL de la pièce.
- ✓ Récupération des plans et des contours de coupe.
- ✓ Création des cellules et insertion des points d'enrichissement au modèle et leurs affectations aux cellules.
- ✓ Positionnement des différentes formes d'outils hors interférences et hors collisions.
- ✓ Détermination de l'outil optimum en chaque point de contact.
- ✓ Détermination de la forme d'outil pour chaque contour.
- ✓ Sélection d'un maximum de trois outils par contour.
- ✓ Limitation du nombre d'outils pour tout le trajet.
- ✓ Détermination des combinaisons des différentes formes d'outils.
- ✓ Génération du trajet d'usinage sans optimisation.
- ✓ Optimisation du trajet d'usinage par la minimisation du nombre de changement d'outils.
- ✓ Simulation virtuelle des mouvements des outils et estimation du temps d'usinage.
- ✓ Création du fichier « CL-FILE » utilisé dans la phase de génération du programme d'usinage « G-Code ».

Le projet que nous avons mené au niveau de l'équipe « CFAO » nous a permis de voir réellement l'importance de l'informatique dans le domaine de la fabrication mécanique et la nécessité de travailler dans une équipe pluridisciplinaire regroupant des compétences de différentes spécialités.

En perspective de notre travail, nous recommandons le traitement des points suivants :

1. Génération de la trajectoire d'outils en mode de balayage « Zig-Zag ».
2. Combinaison de différentes formes d'outils pour le même contour.
3. Utilisation de plus de trois outils par contour.
4. Génération des trajectoires d'outils avec « Z-Constant » en 05-axes.
5. Intégration du parallélisme en utilisant des cartes graphiques GPU pour accélérer les calculs.
6. Adaptation des vitesses d'avance en fonction du volume de la matière enlevée.
7. Génération du programme d'usinage « G-Code » en fonction des différentes configurations des fraiseuses numériques à 03-axes.

Références

- [1] : Jean Christian Bakena Mboua, « Etude et amélioration du processus d'usinage : cas du tournage chez APN Inc. », le 04 mars 2016. Maîtrise avec mémoire, Université du Québec, Canada.
- [2] : A. Benbekhti, « Etude de réalisation d'un support pour affutage des forêts sur une machine à commande numérique », Octobre 2013. Master, Université Abou Bekr Belkaid, Tlemcen, Algérie.
- [3] : M. Colombier, « Fabrication par usinage », Dunod, Paris, 2003.
- [4] : En ligne : http://www.madeitunisia.net/vitrine/art_detail.php?pc1=la6Wp6E%3D&tc1=IKyTlq2S&fbclid=IwAR1OI8b_xzj0RvOF4fqg8glxi_ySBv24EnAmvkN4-Gw9dnf4Tl9hjpgdH0-o [consulté le 24 octobre 2021].
- [5] : H. Bendifallah, M. Bey, W. Ould Amar et N. Mahroug, « Optimisation de l'opération de finition des surfaces gauches par la combinaison des formes et des dimensions d'outils à partir des modèles STL ». Congrès Algérien de Mécanique, CAM2011, 14-17 novembre 2011, Guelma, Algérie.
- [6] : A. SaidanE, K. Serir, « Finition des Pièces Complexes par la Stratégie « Plans Parallèles » sur des Fraiseuses Numériques à 03-Axes », 2018. Master, Université Saad Dahlab, Blida, Algérie.
- [7] : En ligne : <https://www.futura-sciences.com/tech/definitions/informatique-cao-4453/>. [consulté le 02 mars 2021].
- [8] : En ligne : <https://www.organisation-industrielle.fr/quest-ce-que-la-fabrication-assistee-par-ordinateur/>. [consulté le 02 septembre 2021].
- [9] : En ligne : <https://www.mts-equipements.com/>. [Consulté le 02 septembre 2021].
- [10] : Belleili Hafsa, Charchal Latifa, « Combinaison des Outils Hémisphériques, Cylindriques et Toriques pour la Finition des Surfaces Complexes sur des Fraiseuses à 05-Axes », 2018. Master en Informatique, Université Saad Dahlab, Blida, Algérie.
- [11] : En ligne : <https://www.core77.com/posts/67499/Understanding-the-Different-Types-of-3D>. [Consulté le 02 septembre 2021].
- [12] : En ligne : <https://www.core77.com/posts/67499/Understanding-the-Different-Types-of-3D-Files>. [Consulté le 02 septembre 2021].
- [13] : En ligne : <https://cults3d.com/fr/blog/articles/les-differents-formats-de-fichiers-3d-pour-impression-3d>. [Consulté le 02 septembre 2021].
- [14] : D. Rypl, Z. Bittnar, « Triangulation of 3D Surfaces Recovered from STL Grids », 2004, Paper in Computer Science, Technical Université, Thèque.
- [15] : M. Bey, H. Bendifallah†, M. Dermache, N. Abdelhamid « Finition des Surfaces Gauches par Combinaison de Différentes Formes et Dimensions d'Outils » Centre de Développement des Technologies Avancées (CDTA), Baba Hassen, Alger, Algérie.

- [16] : M. Bey, K. Bouhadja, k. Sebti, M. Bouaziz, « Modélisation Volumique des Pièces de Formes Complexes par Triple-Dexels », 10ème Journées de Mécanique de l'EMP 12-13 Avril 2016, Alger, Algeria.
- [17] : Yan Boutin, « Intégration des cycles d'usinage avancés des machines-outils à commande numérique aux logiciels de FAO », Maîtrise en génie mécanique, le 22 janvier 2002. Université du Québec, Canada.
- [18] : En ligne : https://www.mekanika.io/fr_BE/blog/apprentissage-1/notre-guide-pour-tout-savoir-sur-le-g-code-16. [Consulté le 02 septembre 2021].
- [19] : A. Benmimouna, I. Elouchefoune, « Mise en place d'un environnement logiciel pour la virtualisation et l'usinage des pièces complexe sur des fraiseuse numérique à 5-axes », Master en Informatique, 2019. Université Saad Dahlab, Blida, Algérie.
- [20] : S. Lezami, N. Madi, « Conception et Développement d'une Application Logicielle Pour la Simulation d'Usinage des Pièces Complexes sur Fraiseseuses Numériques à 03-Axes », Master en Informatique, 2019. Université Saad Dahlab, Blida, Algérie.
- [21] : L. Saraoui, M. Mesli, « Ebauchage des pièces complexes par la stratégie Plan parallèle su des fraiseuses numériques à 3-axes », Mastère en Informatique, 2019, Université Saad Dahlab, Blida, Algérie.
- [22] : Sandvik, « Outils rotative fraisage, perçage, Alésage» ,2020.
- [23] : En ligne : https://mekanika1.odoo.com/fr_BE/blog/apprentissage-1/guide-de-selection-pour-fraises-cnc-4. [Consulté le 02 septembre 2021].
- [24] : Jean Philippe, « L'association de de stratégie d'usinage pour les pièces structurelles en aéronautique avec des caractéristiques de forme en utilisant un réseau de neurones », le 18 Février 2008. Maitrise en mémoire, Université du Québec Monreale, Canada.
- [25]: Sang C. Park, « Tool-path generation for Z-Constnt contour machining », Cubic Technology Research Center, 18 avril 2001, Seoul, South, korea.
- [26] : En ligne : https://mecsoft.com/VisualCAMcData/Help/vcc_3axis_zlevel_finishing.htm. [Consulté le 02 septembre 2021].
- [27]: M. Bey, A. Cherfi, « Finishing of freeform surfaces with an optimized Z-Constant machining strategy », 2018. 8th CIRP Conférence on High Performance.
- [28] : F. Boudjadi, L. Cheklat, « Conception et réalisation d'une application mobile sensible au contexte pour un musée», Master en Informatique, 2013. Université Abderrahmane Mira, Béjaïa, Algérie.
- [29] : En ligne : <https://openclassrooms.com/fr/courses/1894236-programmez-avec-le-langage-c/1894377-quest-ce-que-le-c>. [Consulté le 02 septembre 2021].

[30] : En ligne : <https://whatis.techtarget.com/definition/OpenGL-Open-Graphics-Library>. [Consulté le 02 septembre 2021].

[31] : En ligne : <https://www.embarcadero.com/fr/>. [Consulté le 02 septembre 2021].