



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

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

CDTA

MEMOIRE DE FIN D'ETUDES

Pour l'obtention

D'un Diplôme de Master en Informatique

Option : Ingénierie Logiciel

THÈME :

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

Réalisé par :

M^{elle}. SAIDANE Asmaà.

M^{elle}. SERIR Kamilia.

Soutenu devant :

Mr. BEY Mohamed	CDTA,	Encadreur
Mr. BENDIFALLAH Hassène	CDTA,	Encadreur
Mme. TCHANTCHANE Zahida	CDTA,	Encadreur
Mr. HAMOUDA Mohamed	USDB,	Promoteur
	USDB,	Président
	USDB,	Examineur

2018/2019

Résumé :

Ce travail s'insère dans le cadre de développement de modules logiciels 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 du « Centre de Développement des Technologies Avancées » « CDTA ».

Dans ce projet, nous nous intéressons à la finition des pièces complexes, définies par leurs modèles « STL », sur des fraiseuses numériques à 03-axes en considérant la stratégie d'usinage « Plans Parallèles ». Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle de production des pièces complexes développée par l'équipe « CFAO » du « CDTA », un module logiciel graphique et interactif permettant la combinaison optimale des deux formes d'outils « Cylindriques » et « Hémisphériques » lors de la finition des surfaces complexes tout en évitant les problèmes d'interférences et de collisions. Ce travail permettra d'améliorer la qualité d'usinage et de réduire les temps d'écriture des programmes.

Mots Clés : Finition, Plans Parallèles, Pièce Complexe, Interférence, Collision, Fraiseuse 03-Axes, Modèle STL.

Abstract:

This work is part of the development of software modules for the production of complex shaped parts initiated by the "Computer-Aided Design and Manufacturing" "CAD/CAM" team of the Products and Robotics Division of the "Advanced Technologies Development Center" "CDTA".

In this project, we are interested in finishing complex parts, defined by their "STL" models, on 03-axis digital milling machines by considering the "Parallel Plans" machining strategy. The aim is to design, develop and integrate into the production software platform complex parts developed by the "CAD/CAM" team of the "CDTA", a graphic and interactive software module allowing the optimal combination of the two forms of tools "Cylindrical" and "Hemispheric" when finishing complex surfaces while avoiding interference and collision problems. This work will improve machining quality and reduce program writing times.

Keywords: Finishing, Parallel Plans, Free Form Surfaces, Interference, Collision, 03-axis machines, STL model.

:

هذا العمل جزء من تطوير وحدات برمجية لإنتاج أجزاء الأشكال المعقدة التي بدأها فريق " التصميم والتصنيع الكمبيوتر " "CFAO" التابع لقسم الإنتاجية والروبوتات في " مركز تطوير التقنيات المتقدمة " "CDTA".

في هذا نحن مهتمون بإنهاء الأجزاء طرازاتها "STL" على آلات رقمية ذات 3 النظر في استراتيجية المعالجة "خطوط موازية". وهو يتضمن تصميم وتطوير وإدماج في منصة برمجيات إنتاج الأجزاء المعقدة التي طورها الفريق "CFAO" "CDTA". وهي وحدة برمجيات رسومية وتفاعلية تتيح الجمع الأمثل بين شكلين " " " " " عند الانتهاء من الأسطح العمل إلى تحسين جودة الآلات وتقليل وقت كتابة . سيؤدي هذا

REMERCIEMENT

Nous remercions en premier lieu le bon Dieu le tout puissant et miséricordieux, qui nous a donné la force, la patience et le courage d'accomplir ce modeste travail.

Nous voudrions présenter nos remerciements vivement à :

Nos encadreurs monsieur Bey Mohammed et monsieur Bendifallah Mohamed El-Hassène pour leur inestimable aide et soutien, leurs orientations et conseils précieux, durant toute cette période d'encadrement.

Nous remercions également notre promoteur monsieur Hamouda Mohamed pour avoir accepté de diriger ce travail, pour son aide, son encouragement et ses conseils tout au long de ce projet.

Nous tenons à remercier aussi toute l'équipe du CDTA en particulier Mme. Bouhadja Khadidja et Mme. Tchantchane Zahida pour leurs encouragements.

Nous remercions les membres du jury de nous honorer par leur présence afin d'évaluer ce travail et pour toutes leurs remarques et critiques.

Plus généralement tout le personnel enseignant du département d'informatique de l'université BLIDA – 1 – qui ont assuré notre formation durant ces 5 années d'étude ainsi qu'à tous les étudiants qui nous ont aidés de près ou de loin.

Un grand merci également à nos familles pour leurs soutiens aussi bien moral que financier et pour leurs sacrifices.

MERCI A VOUS TOUS

Dédicaces

À mon Dieu :

À l'éternel, le tout puissant de m'avoir aidé à arriver au bout de mes études, lui qui m'a accompagné dès le début jusqu'à la fin, il est mon ombre à ma main droite.

À mon père :

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

À ma mère :

Tu m'a donné la vie, la tendresse et le courage pour réussir. Tout ce que je peux t'offrir ne pourra pas exprimer l'amour et la reconnaissance que je te porte. En témoignage, je t'offre ce modeste travail pour te remercier pour tes sacrifices et pour l'affection dont tu m'as toujours entouré.

À ma très chère sœur KHaoula ♥

À mes très chères frères Sid-Ahmed ♥ Tarek ♥ et Nidhal ♥

À toute ma famille : mes grands-parents, mes tantes, mes oncles, mes cousins et mes cousines.

À la personne qui m'a soutenu toute l'année Mon binôme et ma sœur «Kamilia» j'ai partagé tous ces moments de stress, de joie, de panique, de délire et surtout de guigne ensemble.

À tous mes amis : Zabra, Hadjer, Keltouma ♥, Asmaà, Alia, Hadjer, Ryma, Meriem, Adel, Khadidja, Nora et Yasmine merci pour votre soutien morale, ainsi qu'à tous mes camarades de l'équipe CFAO : Radia, Meriem, Latifa, Hafsa, Rim, Fadwa, Amir et ahmed.

À tous les étudiants de la promotion 2018/2019 IL.

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

Asmaà... ♥

DEDICACES

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

Je dédie ce travail à :

Mes chers parents qui je dois toute ma réussite, grâce à leurs soutiens, encouragements et prières, que Dieu me les garde.

À mes chères sœurs : Soumia et Oumayma.

À mes frères : Mounir, Adlen et Aymen.

À tous les membres de l'équipe CFAO.

À notre promoteur Mr. Hamouda Mohamed.

À Mon binôme et ma chère collègue, Saidane Asmaà pour sa foi durant toutes les périodes de réalisation de ce projet.

*À celui qui m'a soutenu tout au long de ce projet
mon fiancé Ramzi.*

À mes amis : Radia, Meriem, Amir, Ahmed, Latifa et Hafsa.

À tous ceux qui m'ont aidé de près ou de loin pour réaliser ce travail.

« Kamilia »

Table des matières

Introduction générale	1
Chapitre I : État de l'art sur l'usinage des pièces complexes en 03-axes.	
<u>Introduction</u>	4
I. Usinage d'une pièce	4
<u>I.1. Processus d'usinage</u>	5
<u>I.2. Usinage 03 axes</u>	5
II. Conception et Fabrication Assistées par Ordinateur « CFAO »	5
<u>II.1. Conception Assistée par Ordinateur « CAO »</u>	6
<u>II.1.1. Méthodes de représentation des surfaces</u>	6
<u>II.1.2. Format d'échange de données</u>	8
<u>II.2. Fabrication Assistée par Ordinateur « FAO »</u>	Erreur ! Signet non défini.1
<u>II.2.1. Fraisage</u>	Erreur ! Signet non défini.1
<u>II.2.2. Définition des fraiseuses</u>	Erreur ! Signet non défini.2
<u>II.2.3. Fraiseuse à 03-axes</u>	Erreur ! Signet non défini.2
<u>II.2.4. Machine-outil à commande numérique</u>	Erreur ! Signet non défini.2
<u>II.2.5. Étapes de production d'une pièce</u>	Erreur ! Signet non défini.4
<u>II.2.6. Stratégies d'usinage en finition</u>	Erreur ! Signet non défini.4
<u>II.2.7. Stratégie d'usinage par « Plans Parallèles »</u>	Erreur ! Signet non défini.5
<u>II.2.8. Outils de coupe</u>	18
<u>II.2.9. Problèmes de l'usinage</u>	Erreur ! Signet non défini.0
<u>Conclusion</u>	Erreur ! Signet non défini.2
Chapitre II : Etude conceptuelle.	
<u>Introduction</u>	Erreur ! Signet non défini.3
I. Solution proposée	Erreur ! Signet non défini.3
<u>I.1. Récupération des paramètres du fichier STL et création du brut</u> ...	Erreur ! Signet non défini.4
<u>I.2. Enrichissement du modèle STL et création des cellules</u>	Erreur ! Signet non défini.4
<u>I.2.1. Enrichissement du modèle STL</u>	Erreur ! Signet non défini.4
<u>I.2.2. Création des cellules</u>	Erreur ! Signet non défini.5
<u>I.2.3. Affectation des points aux cellules</u>	26
<u>I.2.4. Affectation des triangles aux cellules</u>	27

I.3. Faces de la pièce	27
I.4. Détermination des zones visibles et des zones accessibles	28
I.4.1. Détermination des triangles visibles	28
I.4.2. Détermination des sommets accessibles	28
I.4.3. Détermination des triangles accessibles	29
I.5. Création des plans verticaux	29
I.6. Création des contours	Erreur ! Signet non défini.0
I.6.1. Calcul des segments d'intersection	Erreur ! Signet non défini.0
I.6.2. Génération des segments	Erreur ! Signet non défini.2
I.6.3. Chaînage des segments et création des contours	Erreur ! Signet non défini.2
I.7. Choix des outils optimums	Erreur ! Signet non défini.3
I.7.1. Positionnement des outils	Erreur ! Signet non défini.3
I.7.2. Affectation des outils optimums aux sommets du trajet	Erreur ! Signet non défini.4
I.7.3. Affectation des outils optimums au trajet	Erreur ! Signet non défini.5
I.8. Vérification des collisions et des interférences	Erreur ! Signet non défini.5
I.8.1. Vérification des collisions	36
I.8.2. Vérification des interférences	36
I.9. Génération du trajet d'usinage	37
I.9.1. Création des sous contours	37
I.9.2. Insertion de deux points d'engagement et de dégagement d'outil	38
I.9.3. Construction du trajet d'usinage	38
II. Modélisation UML	38
II.1. Diagramme de cas d'utilisation	39
II.2. Diagramme d'activité	Erreur ! Signet non défini.3
II.3. Diagramme de classes	47
Conclusion	Erreur ! Signet non défini.3
Chapitre III : Implémentation et validation.	
Introduction	Erreur ! Signet non défini.4
I. Présentation des langages utilisés	Erreur ! Signet non défini.4
I.1. Présentation du langage C++	Erreur ! Signet non défini.4
I.2. Présentation d'OpenGL	Erreur ! Signet non défini.5
I.3. Présentation de Borland C++ Builder 6	Erreur ! Signet non défini.5
II. Présentation de l'application	56

II.1. Fenêtre principale	56
II.2. Barre du menu principal	56
II.3. Rubrique « Finition pièces complexes en 03-axes à partir d'un fichier STL »	57
III. Présentation des onglets	58
III.1. Lecture du fichier	59
III.2. Enrichissement du modèle et création des cellules	Erreur ! Signet non défini.0
III.3. Visibilité et accessibilité	Erreur ! Signet non défini.0
III.4. Faces du modèle	Erreur ! Signet non défini.1
III.5. Trajet d'usinage	Erreur ! Signet non défini.2
III.6. Test manuel des outils	Erreur ! Signet non défini.3
III.7. Outils optimums	Erreur ! Signet non défini.4
III.8. Trajet des outils	Erreur ! Signet non défini.5
III.9. Simulation des mouvements des outils	66
IV. Test et validation	67
IV.1. Premier modèle STL	67
IV.2. Deuxième modèle STL	Erreur ! Signet non défini.2
Conclusion	78
Conclusion générale	79
Références bibliographiques	81

Liste des figures

Chapitre I : État de l'art sur l'usinage des pièces complexes en 03-axes.

Figure 1. Processus d'usinage.....	5
Figure 2. Localisation d'un point sur une surface paramétrique.....	7
Figure 3. Vecteurs tangents et vecteur normal d'une surface paramétrique.	7
Figure 4. Echange de données entre deux systèmes.....	8
Figure 5. Exemple d'un modèle STL.	9
Figure 6. Etapes de production d'une pièce.	14
Figure 7. Stratégie « Z-Constant ».	15
Figure 8. Stratégie « Isoparamétrique ».	15
Figure 9. Stratégie « Plans Parallèles ».	16
Figure 10. Modes de balayage de l'outil pour la stratégie « Plans Parallèles ».	17
Figure 11. Hauteur de crête.	17
Figure 12. Pas transversal et pas longitudinale.	17
Figure 13. Erreur de flèche.....	18
Figure 14. Outils de finition.	19
Figure 15. Positionnement outil/pièce.....	19
Figure 17. Position d'un outil torique.....	20
Figure 16. Position d'un outil hémisphérique.....	Erreur ! Signet non défini. 0
Figure 18. Position d'un outil cylindrique.	20
Figure 19. Types interférences.	Erreur ! Signet non défini. 1

Chapitre II : Etude conceptuelle.

Figure 1. Organigramme général de la démarche proposée.	Erreur ! Signet non défini. 3
Figure 2. Limites du brut.....	Erreur ! Signet non défini. 4
Figure 3. Enrichissement du modèle STL.	Erreur ! Signet non défini. 5
Figure 4. Création des cellules.	Erreur ! Signet non défini. 6
Figure 5. Affectation des points aux cellules.	Erreur ! Signet non défini. 6
Figure 6 . Affectation des triangles aux cellules.	27
Figure 7. Vérification de l'appartenance d'un point à un triangle.	29
Figure 8 . Plans verticaux en fonction des angles d'orientation....	Erreur ! Signet non défini. 0

Figure 9. Différents cas d'intersection entre un plan et un triangle.	Erreur ! Signet non défini.1
Figure 10. Zones d'interférences et de collisions.....	Erreur ! Signet non défini.6
Figure 11 . Collision avec le corps-outil.	37
Figure 12. Interférence outil cylindrique et outil hémisphérique.....	37
Figure 13. Cas d'utilisation générale.....	39
Figure 14. Cas d'utilisation « Analyser le modèle ».	Erreur ! Signet non défini.0
Figure 15. Cas d'utilisation « calculer la trajectoire ».	Erreur ! Signet non défini.1
Figure 16. Cas d'utilisation « Choisir les outils optimums ».	Erreur ! Signet non défini.2
Figure 17. Cas d'utilisation «Simuler le trajet d'usinage».....	Erreur ! Signet non défini.3
Figure 18. Diagramme d'activité général.....	Erreur ! Signet non défini.4
Figure 19. Diagramme d'activité « Analyser le modèle ».	Erreur ! Signet non défini.5
Figure 20. Diagramme d'activité « Affecter l'outil optimum au point de contact ».....	Erreur ! Signet non défini.5
Figure 21. Diagramme d'activité « Affecter l'outil optimum au contour ».	Erreur ! Signet non défini.6
Figure 22. Diagramme d'activité « Affecter les outils optimums au trajet ».	Erreur ! Signet non défini.6
Figure 23. Diagramme d'activité « Déterminer les 4 outils optimums ».	47
Figure 24. Diagramme d'activité « Calculer la trajectoire ».	47
Figure 25. Diagramme de classe général.....	48
Figure 26. Classe « MODELE_STL ».	48
Figure 27. Classe « CELLULE ».	49
Figure 28. Classe « SOMMET ».	49
Figure 29. Classe « TRIANGLE_STL ».	Erreur ! Signet non défini.0
Figure 30. Classe « OUTIL_STL ».	Erreur ! Signet non défini.0
Figure 31. Classe « BRUT ».	Erreur ! Signet non défini.1
Figure 32. Classe « COULEUR ».	Erreur ! Signet non défini.1
Figure 33. Classe « NORMALE ».	Erreur ! Signet non défini.1
Figure 34. Classe « SEGMENT ».	Erreur ! Signet non défini.2
Figure 35. Classe « CONTOUR ».	Erreur ! Signet non défini.2
Figure 36. Classe « SOUS_CONTOUR ».	Erreur ! Signet non défini.2
Figure 37. Classe « PLAN ».	Erreur ! Signet non défini.3
Figure 38. Classe « TRAJET ».	Erreur ! Signet non défini.3

Chapitre III : Implémentation et validation.

Figure 1. Fenêtre principale.	Erreur ! Signet non défini.	6
Figure 2. Rubrique « Combinaisons outils 05-axes ».		57
Figure 3. Lecture du fichier STL.		58
Figure 4. Rubrique « Finition Pièces Complexes en 03-Axes ».		58
Figure 5. Onglets de l'application logicielle développée.		59
Figure 6. Onglet « Paramètres du modèle ».		59
Figure 7. Onglet « Cellules ».	Erreur ! Signet non défini.	0
Figure 8. Onglet « Visibilité et Accessibilité ».	Erreur ! Signet non défini.	1
Figure 9. Onglet « Faces du Modèle ».	Erreur ! Signet non défini.	2
Figure 10. Onglet « Trajet d'Usinage ».	Erreur ! Signet non défini.	3
Figure 11. Onglet « Test Manuel des Outils ».	Erreur ! Signet non défini.	4
Figure 12. Onglet « Outils Optimum ».	Erreur ! Signet non défini.	5
Figure 13. Onglet « Trajet des Outils ».	Erreur ! Signet non défini.	5
Figure 14. Onglet « Simulation des Outils ».	Erreur ! Signet non défini.	6
Figure 15. Visualisation du premier modèle STL.		67
Figure 16. Normales des triangles.		68
Figure 17. Normales des sommets.		68
Figure 18. Création des cellules et affectation des points.		68
Figure 19. Visibilités des triangles.		69
Figure 20. Sommets accessibles.		69
Figure 21. Accessibilités des triangles.	Erreur ! Signet non défini.	0
Figure 22. Faces de la pièce.	Erreur ! Signet non défini.	0
Figure 23. Création des plans et des contours.	Erreur ! Signet non défini.	1
Figure 24. Minimisation du temps avec la phase d'optimisation.	Erreur ! Signet non défini.	1
Figure 25. Simulation de trajet en 03-axes.	Erreur ! Signet non défini.	2
Figure 26. Visualisation d'outil et coordonnées du point courant.	Erreur ! Signet non défini.	2
Figure 27. Visualisation du deuxième modèle STL.	Erreur ! Signet non défini.	3
Figure 28. Visibilités des triangles.	Erreur ! Signet non défini.	4
Figure 29. Sommets accessibles.	Erreur ! Signet non défini.	4
Figure 30. Accessibilités des triangles.	Erreur ! Signet non défini.	4
Figure 31. Faces de la pièce.	Erreur ! Signet non défini.	5
Figure 32. Création des plans.	Erreur ! Signet non défini.	5

Figure 33. Création des contours.....	Erreur ! Signet non défini.6
Figure 34. Enveloppe de la partie active d'outil et cellules de chevauchement.	Erreur ! Signet non défini.6
Figure 35. Enveloppe de la partie corps d'outil et cellules de chevauchement..	Erreur ! Signet non défini.6
Figure 36. Point de contact avec interférences et collisions.....	77
Figure 37. Simulation du trajet en 03-axes.	77
Figure 38. Visualisation de l'outil et des coordonnées du point courant.	78

Liste des tableaux

Tableau 1. Paramètres du modèle STL.	73
Tableau 2. Visibilité et accessibilité.....	73
Tableau 3. Faces du modèle STL.	75
Tableau 4. Trajet avec et sans optimisation.	77

INTRODUCTION GENERALE

Présentation du sujet :

De nombreux problèmes de nature complexe affrontent les applications à caractère technique, surtout celles touchées par l'évolution technologique. L'introduction de l'outil informatique a permis des progrès scientifiques et technologiques importants dans des domaines aussi variés que l'industrie automobile, l'industrie mécanique, l'industrie des bateaux, l'aéronautique, ...etc.

Parmi les domaines issus de l'essor informatique, le domaine de la Conception et de la Fabrication Assistées par Ordinateur (CFAO) appliqué dans le domaine de la mécanique. La CFAO fait partie des techniques qui concourent à l'automatisation de la production. Avec le développement des outils informatiques et les rénovations des machines-outils à commande numérique, la CFAO est devenue un maillon incontournable de la productique.

Les pièces de formes 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 gauches 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. Avec l'évolution des machines-outils à commande numérique (MOCN), le niveau de qualité de ces pièces a augmenté d'une façon considérable, ce qui impose une attention particulière dans leur mise en production.

Les entreprises de production de pièces mécaniques sont plus que jamais concernées par la réduction du cycle de développement de leurs produits. Cette réduction passe forcément par la réduction des délais de conception et de fabrication. Toutefois, l'amélioration permanente de la qualité des produits, de leurs coûts et de la gestion des connaissances de l'entreprise, restent aussi des impératifs pour une meilleure compétitivité. Cette réduction des délais de fabrication, repose sur une exploitation efficace des méthodes modernes de production, à travers notamment l'automatisation de tâches et la formalisation des connaissances.

Ces dernières décennies ont vu le développement poussé de logiciels informatiques pour la modélisation géométrique de produits (logiciels de CAO) et d'outils pour l'assistance à la fabrication (logiciels de FAO).

Dans la pratique industrielle, un compromis est toujours recherché entre la qualité des pièces usinées et les temps d'usinage pour réduire les coûts. Afin d'optimiser les temps d'usinage et améliorer la qualité des pièces, plusieurs approches développées, considèrent soit les conditions d'usinage soit les stratégies d'usinage soit la sélection des outils optimums. Dans cet ordre d'idées et dans le cadre du programme de recherche piloté par l'équipe « CFAO » « Conception et Fabrication Assistées par Ordinateur » du « CDTA » « Centre de Développement des Technologies Avancées », il nous a été proposé de concevoir une application logicielle qui prend en charge la finition des pièces complexes définies par leurs modèles STL sur des fraiseuses à commande numérique à 03-axes en considérant la stratégie d'usinage par « Plans Parallèles » en utilisant deux modes de balayage (One-Way, ZigZag).

Problématique :

La production des pièces mécaniques complexes avec des surfaces fonctionnelles fait souvent appel à l'enlèvement de matière c'est-à-dire l'usinage. L'usinage de haute précision des surfaces gauches, est l'un des processus clés dans la fabrication moderne. Pour rester compétitifs, les fabricants cherchent constamment à augmenter la qualité, à réduire les coûts d'usinage et les délais de production.

Pour l'opération de finition des surfaces gauches, plusieurs stratégies peuvent être utilisées.

La stratégie la plus utilisée en finition est la stratégie « plans parallèles ». 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 géométrie complexe de ces surfaces rend cette tâche très difficile. Le but recherché est une méthodologie pour optimiser l'opération de finition des surfaces gauches à partir de leurs modèles « STL » sur des fraiseuses numériques à 03-axes par stratégie « plans parallèles ». Autrement dit, étant donné une surface gauche représentée par son modèle « STL », quelles sont les orientations des plans, le pas transversal entre ses plans et la combinaison 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 de modules logiciels pour la production des surfaces de formes complexes initié par l'équipe « Conception et Fabrication Assistées par Ordinateur » « CFAO » de la Division Productique et Robotique du « Centre de Développement des Technologies Avancées » « CDTA ».

Dans ce projet, nous nous intéressons à la finition des pièces complexes, définies par leurs modèles « STL », sur des fraiseuses numériques à 03-axes en considérant la stratégie d'usinage « plans parallèles ». Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle de production des surfaces complexes développé par l'équipe « CFAO » du « CDTA », un module logiciel graphique et interactif permettant la combinaison optimale des deux formes d'outils « Cylindriques et Hémisphériques » lors de la finition des surfaces complexes en évitant les problèmes d'interférences et de collisions. Ce travail permettra d'améliorer la qualité d'usinage et de réduire les temps d'écriture des programmes.

Structuration du mémoire :

Le présent mémoire est composé des parties suivantes :

- Le premier chapitre est consacré à l'étude bibliographique du processus d'usinage des surfaces gauches en 03-axes.
- Le deuxième chapitre est réservé à l'étude conceptuelle de notre application ainsi que la présentation des fonctions et des algorithmes utilisés pendant le développement informatique.
- Le dernier chapitre détaille l'implémentation informatique et les tests réalisés pour valider l'approche proposée.
- Ce mémoire se termine par une conclusion générale et des perspectives.

Chapitre I

État de l'art

Introduction

En mécanique industrielle, la fabrication d'une pièce à partir d'une matière livrée sous forme de produits semi-finis (tôles, barres, etc.) requiert la mise en œuvre d'un ensemble de techniques. L'une d'entre elles est l'usinage, c'est-à-dire un enlèvement de matière par un outil coupant. L'usinage d'une pièce est un besoin permanent que ce soit pour concevoir un moule ou un prototype, reproduire des pièces d'origine ou en produire de nouvelles. La qualité de ces pièces est généralement déterminée par les erreurs de forme et le degré de finition de la surface produite. Une surface usinée est le résultat de la reproduction géométrique et cinématique de la forme et de la trajectoire de la pointe de l'outil. La qualité des états de surface est l'un des aspects les plus pertinents des opérations d'usinage, puisqu'elle représente la phase finale dans le cycle de production pour l'amélioration du degré de finition des surfaces et les propriétés dimensionnelles et géométriques des pièces mécaniques. Donc, il est important de se rendre compte de l'influence des différents facteurs impliqués dans le processus de la coupe afin de choisir les paramètres appropriés permettant d'atteindre la qualité des surfaces désirées et qui dépend de la géométrie de l'outil, des conditions de coupe et enfin la sélection de la cinématique des machines outil à utiliser (1).

Les pièces de formes gauches (libres) sont très répandues dans le domaine de l'industrie manufacturière. Le processus de réalisation de ces surfaces a largement été modifié ces dernières années par l'utilisation de nouvelles techniques comme l'usinage à grande vitesse et la maquette numérique, c'est-à-dire la modélisation complète du produit et de son procédé dans un système de « CFAO » « Conception et Fabrication Assistées par Ordinateur » (2).

Ce chapitre est organisé en trois parties. La première partie est réservée à l'étude du processus d'usinage des surfaces complexes sur des fraiseuses numériques à 03-axes. Dans la deuxième partie, nous avons considéré la conception assistée par ordinateur « CAO », les différents modèles de représentation des surfaces complexes ainsi que le format d'échange de données. La dernière partie est consacrée à la fabrication assistée par ordinateur « FAO ».

I. Usinage d'une pièce

L'usinage d'une pièce par enlèvement de copeaux s'effectue dans le but de donner aux pièces brutes la forme, les dimensions et la précision demandée par le concepteur dans son dessin de définition (3).

I.1. Processus d'usinage

Avec des demandes de plus en plus conséquentes en termes de temps et de quantité pour la réalisation d'un produit, les industriels sont en recherche permanente d'innovation en matière d'usinage. Les principaux types d'usinage actuels sont le tournage et le fraisage. Usiner une surface revient à usiner un ensemble de parties planes plus ou moins petites c'est-à-dire, un maillage plus ou moins fin selon la précision souhaitée. Le processus d'usinage peut être défini en trois parties (Figure 1) (4) :

1. Conception du modèle de la pièce : la pièce est conçue sur ordinateur grâce à un outil de « CAO » en fonction des besoins de l'utilisateur final.

2. Génération des trajectoires d'usinage : selon les outils utilisés, des trajectoires sur toute la pièce sont définies. Ces trajectoires indiquent les chemins que doit suivre les outils pour respecter la précision exigée.

3. Usinage de la pièce : l'usinage est réalisé par la combinaison des mouvements de l'outil et de la pièce.



Figure 1. Processus d'usinage (5).

I.2. Usinage 03 axes

L'usinage 03-axes est l'une des techniques les plus employées dans la fabrication de pièces mécaniques. Il est simple dans son exécution où il fait recours à des machines-outils classiques comme la fraiseuse qui permet un travail de la matière sur 3 axes X, Y et Z (6).

II. Conception et Fabrication Assistées par Ordinateur « CFAO »

La conception et la fabrication assistées par ordinateur « CFAO » permet de concevoir et de réaliser des pièces qui seront ensuite assemblées pour en faire un produit. L'idée générale de ce procédé est d'utiliser les capacités de l'ordinateur afin de concevoir les pièces en 2D ou en 3D pour les fabriquer grâce à une machine connectée directement à l'ordinateur. Les objectifs recherchés par l'intégration du concept de « CFAO » dans le processus de réalisation

des pièces manufacturées, sont d'obtenir une extrême précision, de réaliser un gain de temps et de réduire au maximum l'intervention humaine (7).

II.1. Conception Assistée par Ordinateur « CAO »

La « CAO » comprend l'ensemble des programmes et des techniques de modélisation qui permettent la conception de modèles et de produits. La « CAO » permet aussi de simuler et donc de tester virtuellement les produits avant de les fabriquer. Il est ensuite aisé de transmettre les informations à la « FAO » « Fabrication Assistée par Ordinateur » (8). Pour la représentation des surfaces en « CAO », deux modèles peuvent être utilisés :

➤ **Modèle continu** : la modélisation continue est basée sur les équations mathématiques des différentes entités géométriques.

➤ **Modèle discret** : dans la modélisation discrète, l'objet est décrit par un ensemble d'entités géométriques simples (points, triangles, ...etc.). Dans ce modèle, les paramètres géométriques ne sont pas connus en chaque point et donc ils doivent être estimés.

II.1.1. Méthodes de représentation des surfaces

a. Méthodes de représentation des surfaces du modèle continu

La plupart des surfaces usinées sont des surfaces complexes appelées surfaces gauches. En « CAO », les méthodes de représentation des surfaces sont classées en deux grandes familles : surfaces non paramétriques et surfaces paramétriques.

- **Surfaces paramétriques** : elles sont très utilisées dans la modélisation. Elles sont définies par un ensemble de trois fonctions réelles $X()$, $Y()$ et $Z()$. Ces fonction dépendent de deux paramètres u et v qui n'ont aucune signification physique. Une surface paramétrique est donnée par la formulation suivante :

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

Les paramètres u et v appartiennent à l'intervalle $[0, 1]$. Ainsi, chaque point (u, v) de la surface paramétrique appartient à un carré dans le plan paramétrique défini par les sommets $(0, 0)$, $(0, 1)$, $(1, 0)$ et $(1, 1)$ (Figure 2) (9).

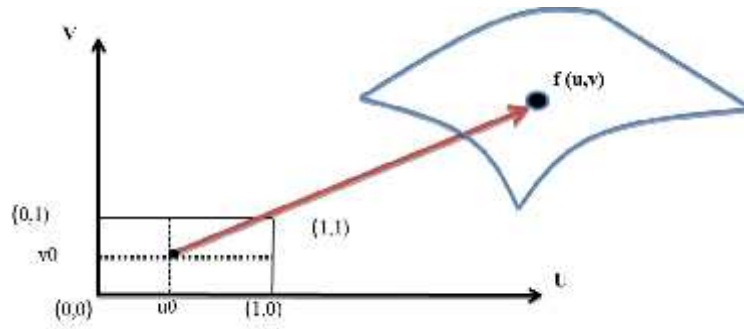


Figure 2. Localisation d'un point sur une surface paramétrique (10).

Propriétés géométriques des surfaces paramétriques : pour l'usinage des surfaces, il est nécessaire de calculer les propriétés géométriques intrinsèques en des points de ces surfaces. Les principales propriétés sont les suivantes :

- **Vecteurs tangents et vecteur normal à la surface :** le vecteur tangent \mathbf{T}_u dans la direction u et le vecteur tangent \mathbf{T}_v dans la direction v à la surface paramétrique au point (u, v) sont donnés respectivement par :

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

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

Ces deux vecteurs définissent le plan tangent à la surface en un point. Le vecteur normal unitaire \vec{n} à la surface au point (u, v) est donné par (Figure 3) :

$$\vec{n} = \frac{\frac{\partial \mathbf{F}}{\partial u} \times \frac{\partial \mathbf{F}}{\partial v}}{\left| \frac{\partial \mathbf{F}}{\partial u} \times \frac{\partial \mathbf{F}}{\partial v} \right|} \quad (4)$$

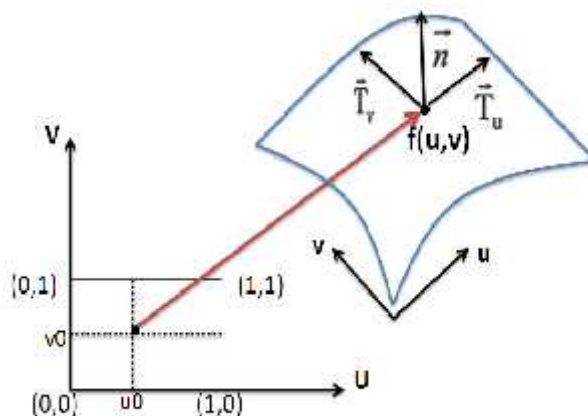


Figure 3. Vecteurs tangents et vecteur normal d'une surface paramétrique (10).

- **Surfaces non paramétriques :** ces surfaces peuvent être représentées sous deux différentes formes :

- **Forme explicite** : une surface explicite est donnée par une équation algébrique de la forme (9) :

$$Z = F(X, Y) \quad (5)$$

Où à chaque valeur de **X** et **Y** correspond une et une seule valeur de **Z**. Donc, cette surface ne peut pas être fermée puisque des valeurs multiples ne sont pas permises.

- **Forme implicite** : une surface implicite est donnée par l'équation (9) :

$$F(X, Y, Z) = 0. \quad (6)$$

Cette représentation nécessite la résolution d'une équation pour obtenir les triplets de coordonnées.

b. Méthodes de représentation des surfaces du modèle discret

Dans le modèle discret, la surface est représentée par des éléments géométriques simples tels que points, triangles, tétraèdre, ...etc.

II.1.2. Format d'échange de données

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 (Figure 4).



Figure 4. Echange de données entre deux systèmes (5).

Parmi ces formats, **SET** (Standard d'Echange et Transfert), **IGES** (Initial Graphical Exchange System), **STL** (Standard Tessellation Language).

- **Format STL (Standard Tessellation Language)** : le format de fichier STL est un format utilisé dans les logiciels de stéréolithographie (STL pour STereo-Lithography). Il a été développé par la société 3D Systems. Il est largement utilisé pour faire du prototypage rapide et pour la fabrication assistée par ordinateur. Le format de fichier STL ne décrit que la géométrie des surfaces d'un objet en 03 dimensions par un ensemble de triangles (Figure 5). Il ne comporte pas d'informations concernant la couleur, la texture ou les autres paramètres habituels d'un modèle de conception assistée par ordinateur.

Le format de fichier STL est le format de fichier le plus utilisé pour l'impression 3D. Utilisé conjointement avec une trancheuse 3D, il permet à un ordinateur de communiquer avec le matériel d'une imprimante 3D.

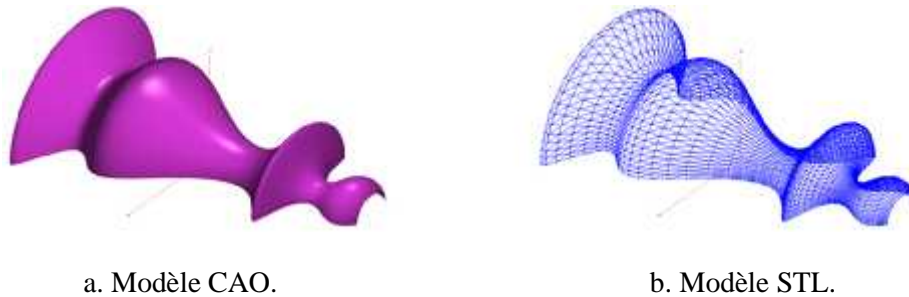
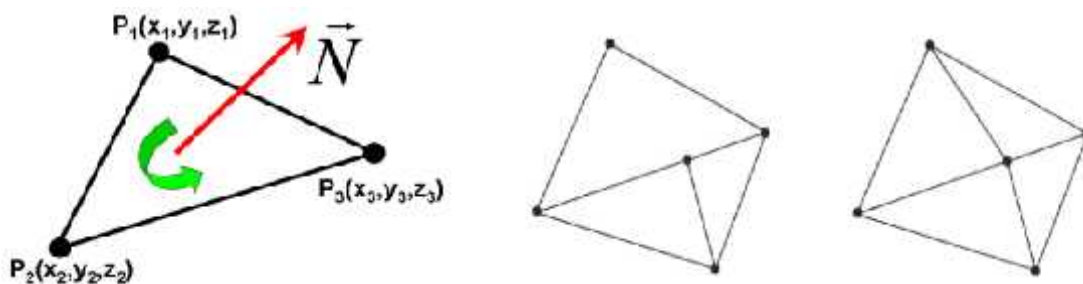


Figure 5. Exemple d'un modèle STL (11).

- **Règles spéciales pour le format STL :** le fichier STL décrit un objet par sa surface externe. Elle est nécessairement fermée et définie par une série de triangles. Chaque triangle est défini par les coordonnées cartésiennes (x, y, z) , de son vecteur normal \vec{n} orienté vers l'extérieur et de ses trois sommets ordonnés dans le sens trigonométrique (Figure 6.a). Chaque triangle doit partager deux (02) sommets avec chacun des triangles voisins (Figure 6.b). Autrement dit, le sommet d'un triangle ne doit pas être sur l'arête d'un autre triangle (12).



a. Coordonnées et l'orientation des sommets. b. Règle de sommet pour les fichiers STL.

Figure 6. Règles du modèle STL (12).

- **Stockage des informations dans un fichier STL :** le format du fichier STL fournit deux manières pour stocker les informations des triangles : codage ASCII et codage binaire. Pour les deux formats, Les coordonnées des sommets et les composantes du vecteur normal unitaire au triangle sont stockées mais de deux manières différentes.

- ❖ **ASCII STL :** Il est spécifié comme un fichier ASCII (caractère imprimable). Il est plus descriptif et lisible mais gros de taille (Figure 7.a). Le fichier ASCII commence par une ligne précédée par le mot réservé « Solid ». Par la suite, pour chaque triangle, sont définis le vecteur et les coordonnées des sommets du triangle. Il se termine par « endsolid » (12).

❖ **STL BINAIRE** : il est spécifié comme un fichier binaire plus commun, moins de taille mais non lisible et trop condensé (Figure 7.b). Son entête se compose de 84 octets, les premiers 80 octets sont utilisés pour obtenir des informations sur le fichier, et les 4 derniers octets représentent le nombre de facettes triangulaires. 50 octets sont utilisés pour représenter les composantes **x**, **y** et **z** de la normale pour chaque facette. 4 octets sont utilisés pour chaque coordonnée **x**, **y** et **z** de chaque sommet du triangle (48 octets par facette). Les deux derniers octets ne sont pas utilisés (12).

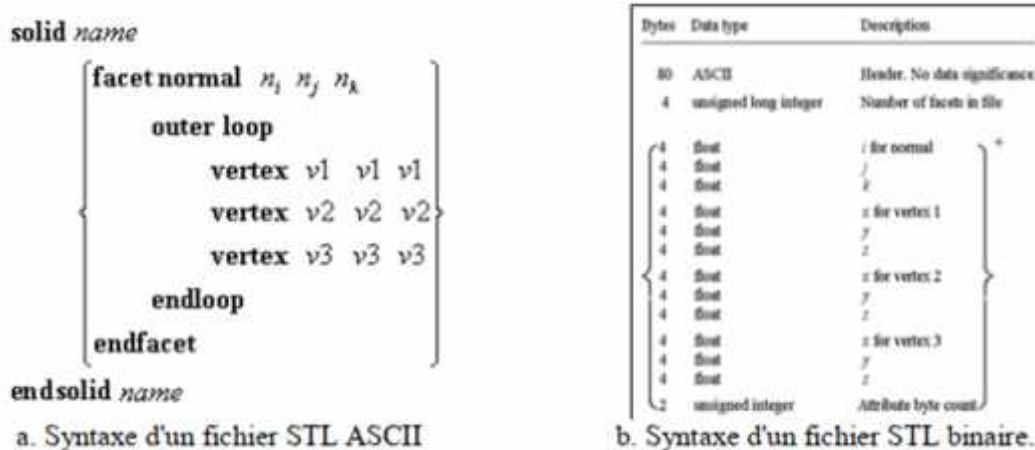


Figure 7. Syntaxe de fichier STL (13).

• **Avantages et inconvénients du modèle STL :**

✓ **Avantage** (12) :

- Compatible avec de nombreux logiciels de modelage 3D et imprimantes 3D.
- Fichier plus simple que les autres formats d'échange de données.
- Simple puisque le triangle est utilisé pour décrire l'objet.

✓ **Inconvénients** (12) :

- Supprime les informations de connectivité.
- Le support des couleurs n'est pas standard et n'est souvent pas possible.
- Le modèle en facettes triangulaires est une approximation du modèle CAO. La fidélité du modèle d'approximation dépend de la précision imposée.
- Le fichier généré est souvent très volumineux.
- Les rayons de courbures très faibles exigent un grand nombre de triangles.

- Redondance de l'information parce que les sommets de chaque triangle sont partagés par plusieurs triangles ce qui augmente considérablement la taille du fichier.

II.2. Fabrication Assistée par Ordinateur « FAO »

Après la validation de l'étape précédente, l'usinage de la pièce proprement dit peut être réalisé manuellement en utilisant différentes machines. Cependant, grâce aux développements de nombreuses machines numériques, il est possible d'automatiser tout ou une partie du processus jusqu'à la matérialisation de la pièce. Cette étape est appelée la fabrication assistée par ordinateur « FAO ». Elle génère en sortie un fichier contenant le programme de pilotage d'une machine-outil à commande numérique. Ce fichier décrit précisément les mouvements que doit exécuter la machine-outil pour réaliser la pièce demandée. Le passage de la « CAO » vers « FAO » nécessite certaines informations contenues dans le modèle conçu. Ce dernier se fait systématiquement dans le cas où la « CAO » et la « FAO » sont intégrées dans le même environnement. Dans le cas contraire, il faut utiliser des formats d'échange de données.

II.2.1. Fraisage

C'est un procédé de fabrication par enlèvement de matière. En fraisage, l'enlèvement de matière « copeaux » résulte de la combinaison de deux mouvements : rotation de l'outil de coupe d'une part (mouvement de coupe « M_c ») et avance rectiligne de la pièce à usiner (mouvement d'avance « M_a ») (Figure 8). Trois modes de fraisage peuvent être distingués : fraisage en bout, fraisage en roulant et fraisage combiné. Le fraisage en bout est le plus répandu car il permet de réaliser des surfaces complexes. Le fraisage en bout peut être réalisé sur des machines-outils à commande numérique (MOCN) 03-axes ou 05-axes (14).

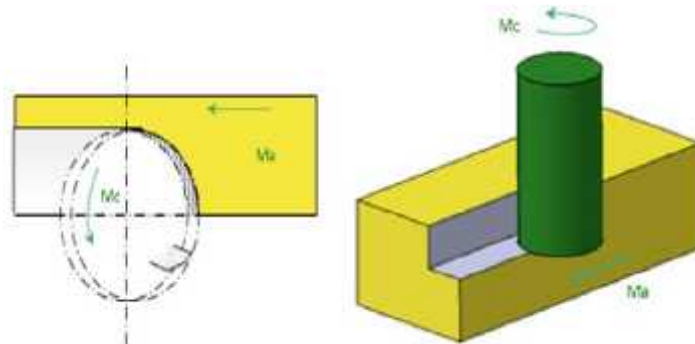


Figure 8. Mouvement de coupe et mouvement d'avance (3).

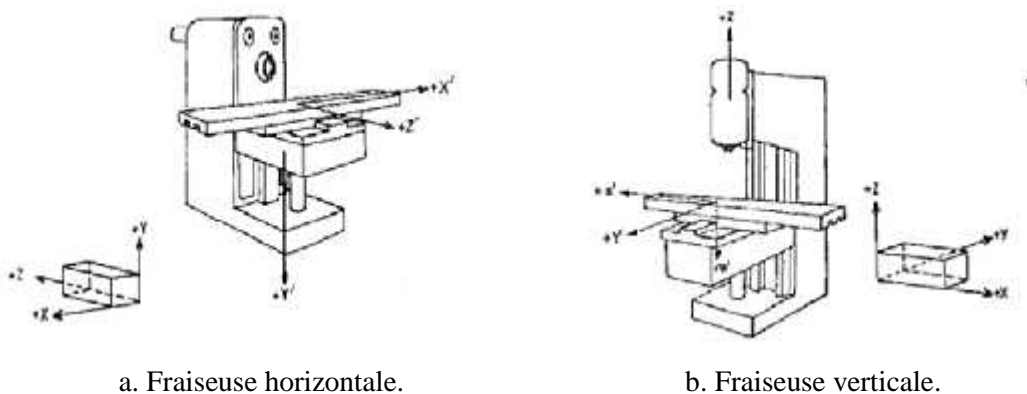
II.2.2. Définition des fraiseuses

Une fraiseuse est une machine-outil utilisée pour usiner des pièces mécaniques. Elle équipée d'un outil coupant appelé une fraise. La fraiseuse est capable d'enlever de la matière sur une pièce initiale afin de la transformer en une autre pièce. Ils existent différents types de fraiseuses selon l'orientation de l'outil. Les fraiseuses peuvent être 03, 04 ou 05-axes (6).

II.2.3. Fraiseuse à 03-axes

Une fraiseuse à 03-axes comporte une broche soit horizontale (Figure 9.a) soit verticale (Figure 9.b) ou universelle, et c'est à la table d'être positionnée en fonction du but recherché sur la pièce. La broche horizontale sert en fait à évacuer plus simplement les copeaux (6). Les axes de la machine sont définis de la manière suivante :

L'axe **Z** est parallèle à la broche principale de la machine, l'axe **X** correspond au plus grand déplacement horizontal tandis que l'axe **Y** forme le trièdre direct avec les axes **X** et **Z**.



a. Fraiseuse horizontale.

b. Fraiseuse verticale.

Figure 9. Fraiseuse 03-axes (15).

II.2.4. Machine-outil à commande numérique

Une Machine-outil à commande numérique « MOCN » est une machine d'usinage à cycle automatique programmable. Le terme « commande numérique » est générique et a été retenu parce que la machine est commandée par des « consignes » numériques fournies par un calculateur. En d'autres termes, les organes mobiles de la machine sont motorisés et un automatisme assure la commande et dans la plupart des cas le contrôle de la position et/ou de la vitesse. Ce type de machine se compose ainsi de deux parties complémentaires :

- **Partie opérative** : c'est la machine-outil. Elle agit directement sur le produit à réaliser.
- **Partie commande** : c'est la commande numérique. Elle permet d'élaborer des ordres en fonction des consignes et des comptes rendus (16).

➤ **Langage de programmation** : la programmation actuelle des machines-outils se fait par l'intermédiaire du langage « G » ou « G-Code » dont les principes sont regroupés dans la norme ISO 6983 et annexes (5).

A l'origine, le code « G » était basé sur un principe de programmation qui remonte à la période des cartes perforées. Cette programmation s'appuie sur des fonctions préparatoires de type G, elle est complétée par des fonctions auxiliaires de type M et technologiques (F, S,...etc.) par l'intermédiaire de ce code. L'utilisateur communique à la machine un ensemble d'instructions explicites. Un programme en « G-Code » est la traduction d'une suite d'actions et de déplacements élémentaires (ligne droite ou arc de cercle par exemple) qui permettent de générer des conditions d'usinage ainsi qu'une trajectoire plus ou moins approximée. Depuis l'établissement de la norme ISO 6983, et avec l'évolution rapide des technologies, de nombreuses extensions ont été ajoutées pour tenir compte des nouveautés et des nouvelles capacités des machines-outils. Ces extensions, n'entrent pas dans la norme et compliquent la tâche des logiciels de FAO, qui doivent créer les lignes de ce langage pour un directeur de commande numérique « DCN » particulier. À côté de l'ISO, de nouveaux codes sont apparus, différents selon le directeur de commande numérique. Ils intègrent de plus en plus, en plus de ISO, des langages propres aux constructeurs de « DCN » (symbolique, C, etc.) ainsi que les interfaces de programmation conversationnelle destinées à simplifier la programmation.

Fonctions préparatoires G : fonctions qui préparent la logique à une action donnée ou à un type de calcul.

Fonctions auxiliaires M : fonctions qui déterminent les mouvements, la sélection de vitesse, l'arrosage, etc.

Le flux de données s'effectue de la conception jusqu'à l'usinage. La maquette CAO est envoyée en FAO sous la forme d'un format standard. Celui-ci peut être STL ou autre format selon les logiciels utilisés, la pièce est ensuite traitée par un logiciel de FAO dont le but est de préparer l'usinage par la génération de trajectoires, d'outils, et de paramètres de coupe. Une fois cette phase terminée, le fichier CL (Cutter location) ainsi obtenu, doit être transféré vers un post-processeur pour le traduire dans un format lisible par la commande numérique. Cette dernière ne lit que le code G qui comporte des fonctions spécifiques à chaque machine (5).

➤ **Post-processeurs** : Si l'on reprend le schéma simplifié du processus d'usinage, le post-processeur sert à transformer un fichier CL « Cutter Location file » en un fichier adapté pour être lu par la commande numérique, c'est-à-dire en code G (Figure 10) (5).



Figure 10. Post-Processeur (5).

II.2.5. Étapes de production d'une pièce

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

- **Ebauchage** : elle permet d'enlever le maximum de matière (excédent) en un minimum de temps (Figure 11.a). La machine doit être puissante et l'outil doit être robuste. Elle permet d'éliminer la croûte superficielle liée au procédé d'obtention du brut (15).
- **Demi-finition** : elle permet d'avoir une bonne approche de la surface finale et d'assurer la régularité du copeau de finition (Figure 11.b). Elle permet de préparer une finition précise en corrigeant les défauts géométriques résultants d'une ébauche et d'assurer une surépaisseur constante et faible en finition (15).
- **Finition** : l'objectif est l'obtention d'une bonne qualité de surface (dimensions, forme et rugosité) dans les tolérances permises par le dessin de définition (Figure 11.c) (15).



a. Ebauchage.

b. Demi-finition.

c. Finition.

Figure 6. Etapes de production d'une pièce (13).

II.2.6. Stratégies d'usinage en finition

Une stratégie d'usinage est une méthodologie utilisée pour générer une série d'opérations, dans le but de réaliser une forme donnée. Elle permet d'associer un processus d'usinage à une entité d'usinage, c'est-à-dire un ensemble d'opérations comprenant, la définition des outils, des conditions de coupe et des trajectoires d'usinage. Les stratégies d'usinage permettent de calculer l'ensemble des trajectoires de l'outil lors de l'usinage d'une

pièce sur machine-outil à commande numérique. Comme il n'y a pas une forme d'outil permettant de générer des surfaces gauches directement, ces dernières sont usinées par balayage d'un outil selon une direction privilégiée caractérisant le mode de balayage. Le choix d'un mode est basé sur un critère d'optimisation du temps d'usinage et/ou de la qualité des surfaces à usiner. Les stratégies les plus utilisées pour la finition des surfaces gauches résultent de l'usinage en 03-axes sont les suivantes :

✓ **Z-Constant** : elle s'appuie sur des trajectoires d'outil résultantes de l'intersection entre la surface à usiner et un ensemble de plans horizontaux parallèles (Figure 12) (13).

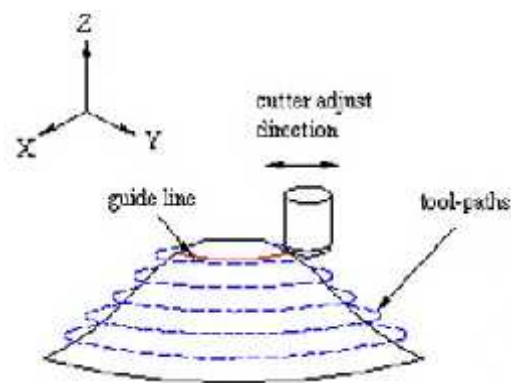


Figure 7. Stratégie « Z-Constant » (15).

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

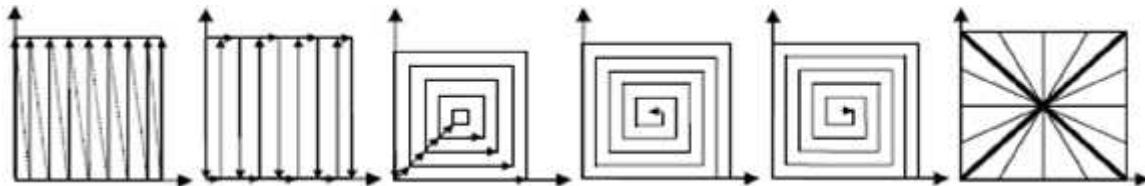


Figure 8. Stratégie « Isoparamétrique » (13).

✓ **Plans Parallèles** : elle s'appuie sur des trajectoires d'outil résultantes de l'intersection entre la surface à usiner et un ensemble de plans verticaux parallèles (Figure 14) (13).

II.2.7. Stratégie d'usinage par « Plans Parallèles »

La stratégie par « Plans Parallèles » est la stratégie la plus utilisée en industrie. Les plans de coupe sont des plans verticaux parallèles entre eux. Ces plans parallèles sont caractérisés par une normale commune dans une direction donnée. Sur chaque point de contact, l'outil doit être tangent à la surface. Pour déterminer une trajectoire d'outil, les points d'intersection entre un plan et la surface à usiner doivent être calculés (17).

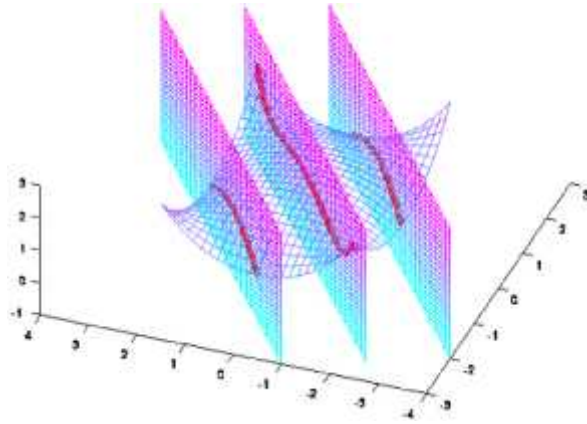


Figure 9. Stratégie « Plans Parallèles » (17).

Les principaux avantages de la stratégie par « Plans Parallèles » sont :

- Elles ne génèrent pas de trajectoires d'outil redondantes ce qui peut générer un gain de temps notable.
- Elles évitent l'apparition de zones non usinées lors de la planification de trajectoires.
- La planification des trajectoires est faite sur l'espace 3D. Très bonne maîtrise de la planification de trajectoire et du pas transversal.
- Pour les surfaces où l'aspect visuel est important, la stratégie par plans parallèles est très intéressante car les trajectoires laissées par l'outil sur la surface sont homogènes.

La stratégie par plans parallèles n'est pas forcément optimale. La mise en œuvre de cette stratégie nécessite la détermination de la distance entre deux passes successives appelée pas transversal **Pt**. Certains modèles ont été développés pour adapter le pas transversal au fur et à mesure de la construction des trajectoires. Les trajectoires définies par les plans parallèles sur la surface peuvent être parcourues suivant deux modes (17) :

✓ **Parcours « One-Way »**: l'outil est toujours sollicité de la même manière d'une passe à l'autre et les trajectoires usinées sont ainsi homogènes. Ce type de parcours améliore l'état de surface par rapport au parcours de type « Zig-Zag » (Figure 15.a) (9).

✓ **Parcours « Zig-Zag »** : l'outil suit une droite jusqu'à sa fin. Ensuite, l'outil se déplace à la fin de la droite suivante sans quitter la surface et reprend l'usinage en sens inverse (Figure 15.b). Ce processus est répété jusqu'à l'usinage de toute la surface. Son avantage est la diminution du temps d'usinage en évitant les retraits hors matière (9).

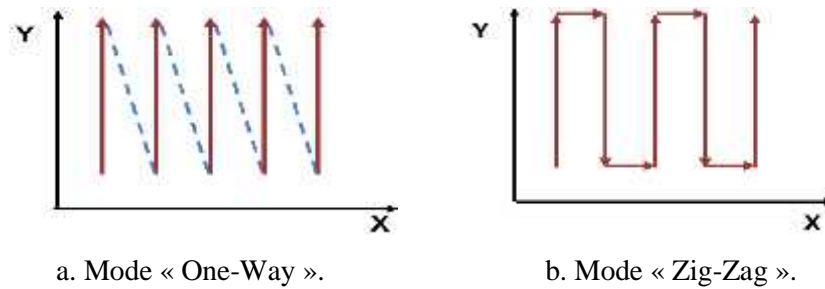


Figure 10. Modes de balayage de l'outil pour la stratégie « Plans Parallèles ».

➤ **Paramètres de la stratégie « Plans parallèles » :** en usinage 03-axes, la construction des trajectoires outils requiert la connaissance d'une direction d'usinage, d'un pas transversal, d'une hauteur de crête, d'un pas longitudinal, d'un outil et d'une erreur de flèche.

- **Direction d'usinage :** est la direction suivie par l'outil lors de l'usinage d'une surface et suivant laquelle les plans parallèles sont orientés (17).
- **Pas transversal « Pt » :** chaque plan est éloigné de son voisin d'une distance notée « Pt » appelée pas transversal (Figure 16) (17).
- **Hauteur de crête « hc » :** correspond à l'épaisseur résiduelle non usinée laissée par l'outil lors de son déplacement suivant deux trajectoires adjacentes (Figure 16). Elle est un critère important à prendre en compte lors de la planification de trajectoires (17).
- **Pas longitudinal :** dépend de l'erreur de corde (erreur de flèche) dans la direction d'usinage et de la courbure de la trajectoire (Figure 17) (9).
- **Erreur de flèche :** les courbes suivies par l'outil sont approchées par interpolation linéaire. Donc, l'outil usine une ligne brisée. Le pas longitudinal d'usinage est calculé par le respect de la tolérance d'usinage « valeur maximale de l'erreur de flèche » (Figure 18) (9).

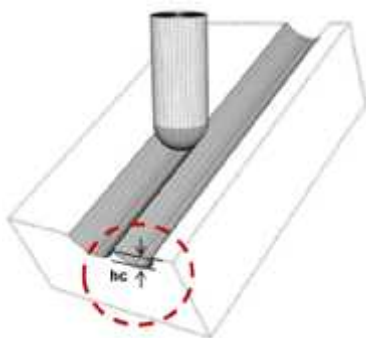


Figure 11. Hauteur de crête (17).

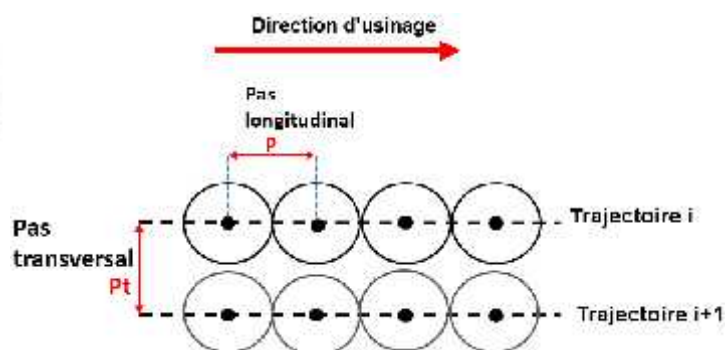


Figure 12. Pas transversal et pas longitudinal.

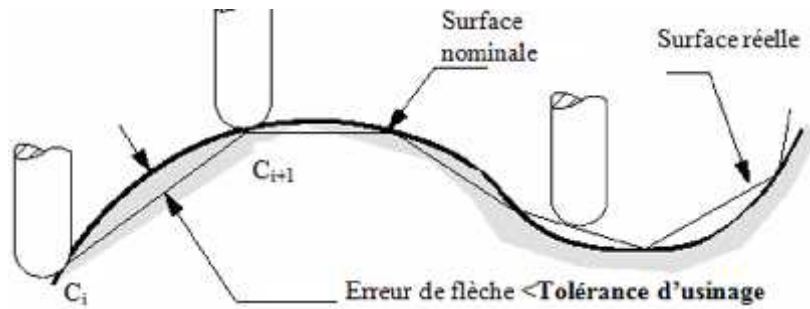


Figure 13. Erreur de flèche (9).

II.2.8. Outils de coupe

L'outil de coupe (fraise) munie de dents est mis en rotation et taille la matière suite à sa rotation et au mouvement relatif de la fraise par rapport à la pièce. La forme de la fraise est variable. Elle peut être cylindrique, torique, hémisphérique ou quelquefois de forme plus complexe. Les caractéristiques physiques de la fraise (vitesse de rotation, avance) dépendent de la matière à usiner, de la profondeur de travail et de la coupe. Le choix du type de la fraise dépend de l'opération à réaliser (14).

➤ **Type d'outils :** l'opération de finition des surfaces gauches peut être réalisée par des outils cylindriques, hémisphériques et toriques et qui peuvent être monoblocs ou à plaquettes.

- **Fraises cylindriques :** si l'arête de coupe arrive jusqu'au centre de l'outil, la vitesse de coupe est faible et l'évacuation des copeaux n'est pas bonne (Figure 19.a) (18).

- **Fraises hémisphériques :** ce sont les outils les plus polyvalents. Ces fraises sont très intéressantes car elles peuvent pivoter autour du point centre outil sans collision entre l'outil et la pièce. Ces fraises coupent de plus en plus mal lorsque le point de contact outil-pièce se rapproche de l'axe d'outil (Figure 19.b) (18).

- **Fraises toriques :** ce sont des outils intermédiaires entre les fraises cylindriques et les fraises hémisphériques. Ils possèdent des avantages considérables dont le plus important est d'avoir une partie active, le « tore » situé sur un rayon éloigné de l'axe de l'outil. Donc, cette partie active possède une vitesse non nulle et la coupe est bonne. Son avantage est la possibilité de pivoter l'outil autour du centre plaquette pour changer l'angle de l'inclinaison tout en conservant le contact outil/pièce (Figure 19.c) (18).



Figure 14. Outils de finition (17).

➤ **Positionnement de l'outil :** le positionnement d'un outil sur une surface gauche peut être décomposé en deux étapes : la définition du point de contact outil/pièce et ensuite l'orientation de l'axe de l'outil (Figure 20).

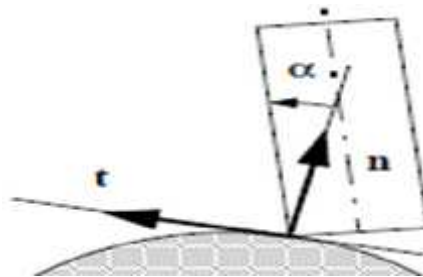


Figure 15. Positionnement outil/pièce (18).

La finition des surfaces gauches en un temps minimum et un bon état de surface, passe impérativement par la détermination des outils (formes et dimensions) et des conditions de coupe (vitesse d'avance, vitesse de coupe). L'outil utilisé doit être constamment tangent à la surface à usiner en chaque point de contact. Le positionnement de l'outil en finition par rapport à la surface en un point de contact dépend du vecteur normal unitaire au point de contact, de l'orientation de l'axe de l'outil, de la forme et des dimensions de l'outil.

- **Position de l'outil torique :** ses positions sont données par (Figure 21) (19):

$$O_E = O_C + r\vec{n} + (R - r) \cdot \vec{v} \quad (7)$$

$$\vec{v} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|} \wedge \vec{u} \quad (8)$$

- **Position de l'outil hémisphérique :** ses positions sont données par (Figure 22) (19):

$$O_E = O_C - r \cdot \vec{n} \quad (9)$$

$$O_L = O_E - r \cdot \vec{u} \quad (10)$$

- **Position de l'outil *cylindrique*** : ses positions sont données par (Figure 23) (19):

$$O_E = O_C - r \cdot \vec{v} \quad (11)$$

$$\vec{v} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|} \wedge \vec{u} \quad (12)$$

Avec :

- ✓ CC : Point de contact entre l'outil et la surface,
- ✓ CE : Point centre de l'outil,
- ✓ CL : Point extrémité de l'outil,
- ✓ \vec{n} : Vecteur unitaire normal à la surface au point de contact,
- ✓ 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.

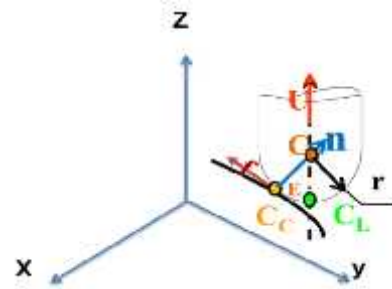
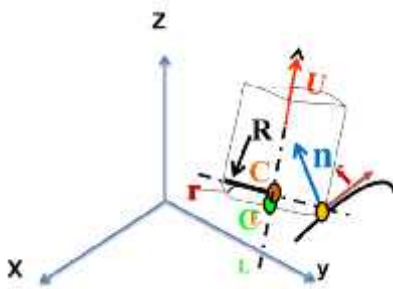


Figure 16. Position d'un outil torique. **Figure 17.** Position d'un outil hémisphérique.

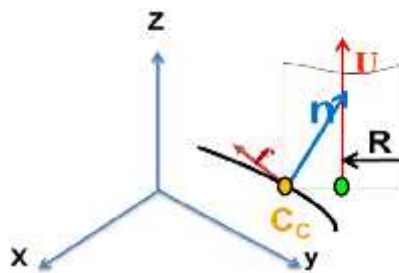


Figure 18. Position d'un outil cylindrique.

Les équations précédentes permettent de calculer soit les coordonnées du centre de l'outil à partir du point de contact, soit le point de contact à partir des coordonnées du centre. Une fois la position de l'extrémité de l'outil déterminée, il reste à définir l'orientation de son axe.

II.2.9. Problèmes de l'usinage

En raison des formes géométriques très complexes des surfaces gauches, un grand risque de problèmes d'interférences et de collisions est généré. Ces problèmes peuvent apparaître au niveau des positionnements d'outils calculés, lors de l'interpolation de la trajectoire ou encore

lors des mouvements hors matière. Pour un point de contact outil pièce donné, toute la difficulté consiste à déterminer l'orientation efficace de l'axe de l'outil. La maîtrise des interférences outil/pièce lors de la génération de trajectoires d'outil est une condition nécessaire pour garantir la conformité de la pièce en termes de géométrie. Les interférences sont en général classées selon trois types (Figure 24)

▮ **Interférences locales** : elles traduisent un enlèvement de matière excessif par la partie active de l'outil sur la surface à usiner. Elles sont principalement dues aux différences de courbures de la partie active de l'outil et de la surface à usiner au point de contact (19).

▮ **Interférences vers l'arrière** : elles sont des pénétrations intempestives de l'arrière de l'outil dans la surface à usiner (19).

▮ **Collisions** : les interférences globales (collisions) représentent de manière générale des collisions entre l'ensemble {corps d'outil, porte outil, broche} et l'ensemble {pièce, porte pièce, éléments de structure...} (19).



Figure 19. Types interférences (13).

➤ **Evitement des problèmes d'usinage** : le problème de la détection des interférences locales ou globales est très largement développé dans le domaine de l'usinage multi-axes. Certaines méthodes sont principalement basées sur la comparaison locale de la surface de l'outil avec la surface à usiner. L'objectif consiste alors à positionner l'outil hors interférence afin d'obtenir un enlèvement de matière maximum. D'autres méthodes sont consacrées à la détection a posteriori des interférences ou collisions en prenant en compte les positionnements outils déjà calculés. Pour corriger les interférences et les collisions, deux solutions sont utilisées :

- Changement d'outil ou bien modification de l'orientation d'outil pour éviter les interférences avec la surface.
- Pour corriger les collisions, il suffit juste de réorienter l'outil quelle que soit sa géométrie (13).

Conclusion

L'usinage des surfaces gauches est un problème complexe nécessitant la prise en compte de plusieurs paramètres tels que la stratégie d'usinage à utiliser, les dimensions des outils, la cinématique de la machine, la direction d'usinage et la qualité de la surface résultante. Les méthodes utilisées généralement pour l'usinage de ces surfaces se basent sur le calcul d'une direction initiale optimale. Suivant la stratégie d'usinage utilisée, les trajectoires secondaires en sont déduites. Cependant, cela ne permet pas d'avoir une direction d'usinage optimale sur l'ensemble de la surface et ne garantit pas la minimisation du temps d'usinage ou de la longueur de la trajectoire d'usinage. En effet, la direction d'usinage varie le long d'une trajectoire d'une surface gauche et elle est difficilement prévisible. Le temps dépend de la vitesse et de la direction d'avance de l'outil. Un choix judicieux d'une direction d'usinage en fonction de la géométrie de la surface permet une réduction du temps d'usinage effectif.

L'opération d'usinage en trois 03-axes étudiée dans ce travail, est le fraisage en bout avec la stratégie « Plans Parallèles » avec des outils à bouts toriques, hémisphériques et cylindriques. Ce choix est justifié par le fait que cette stratégie est la plus utilisée et la mieux maîtrisée en industrie. Il s'agit d'une des rares stratégies qui ne génèrent pas de trajectoires redondantes et évitent l'apparition de zones non usinées lors de la planification de trajectoires.

Chapitre II

ETUDE CONCEPTUELLE

Introduction

En informatique la résolution d'un problème exige sa décomposition en une suite d'étapes successives pour ensuite donner une solution adéquate. L'objectif de notre travail est la conception d'une application logicielle permettant la génération des chemins d'outils pour l'usinage en finition des pièces complexes par la stratégie « Plans Parallèles ». Le trajet ainsi défini, est traduit en un code spécifique interprétable par le directeur de la commande numérique de la machine-outil appelé « G-Code ». Dans ce qui suit, les différents algorithmes proposés ainsi que la conception de notre application sont présentés.

I. Solution proposée

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

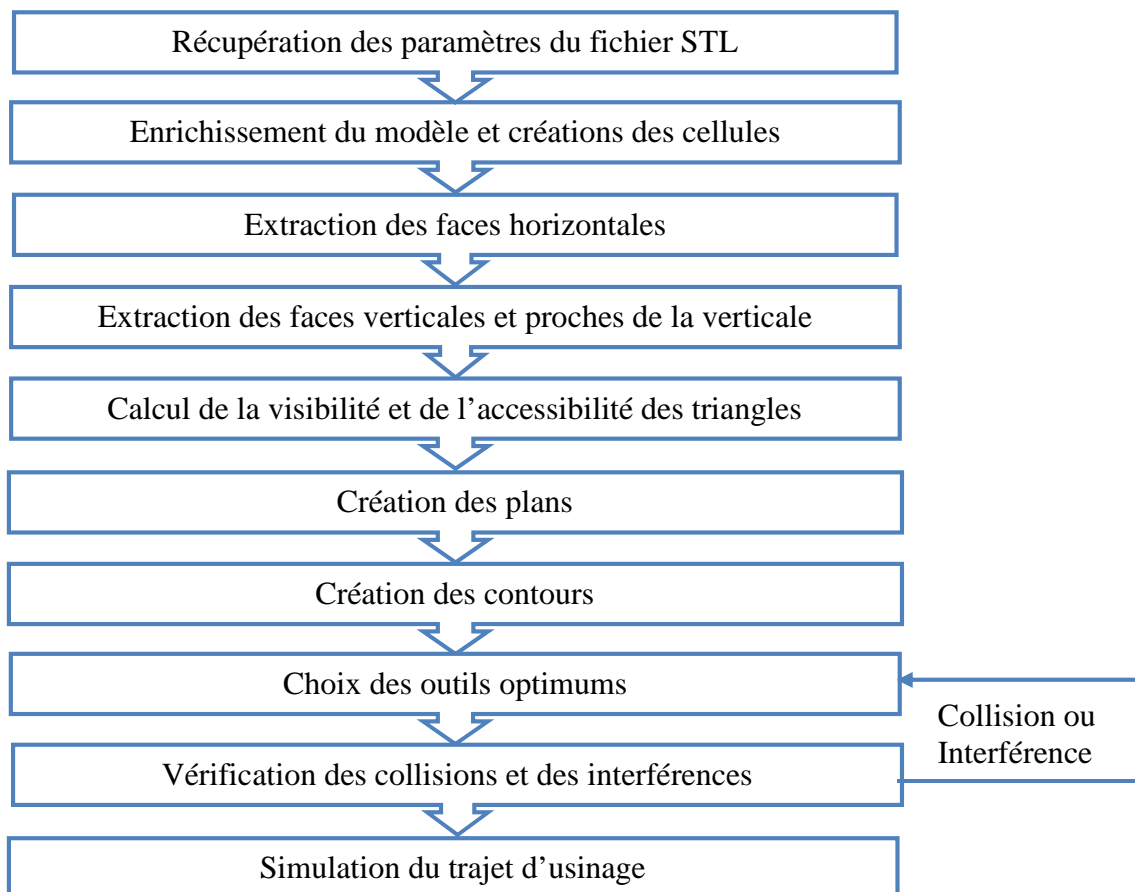


Figure 20. Organigramme général de la démarche proposée.

I.1. Récupération des paramètres du fichier STL et création du brut

Le fichier STL décrivant la pièce à usiner doit être analysé. Cette analyse consiste à vérifier son extension et sa syntaxe. Cette analyse a été faite auparavant, nous avons seulement lu le fichier STL, récupéré les informations des sommets et des triangles et ensuite les stocker dans un fichier texte « .txt ». Par la suite, les dimensions minimales du brut de la pièce sont calculées. La création du brut a pour but d'englober tous les points du modèle STL dans une enveloppe parallélépipédique de faces parallèles aux axes **X**, **Y** et **Z**. Il s'agit de déterminer les coordonnées **Xmin**, **Xmax**, **Ymin**, **Ymax**, **Zmin** et **Zmax** et par suite ses dimensions minimales (longueur, largeur et hauteur) (Figure 2) (13).

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

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

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

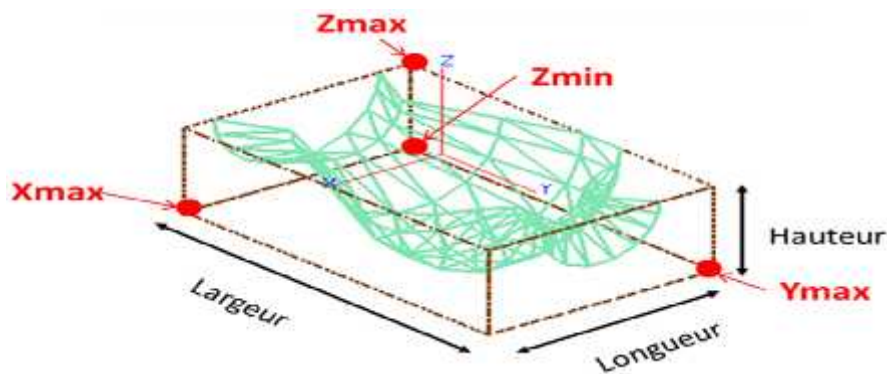


Figure 21. Limites du brut.

I.2. Enrichissement du modèle STL et création des cellules

I.2.1. Enrichissement du modèle STL

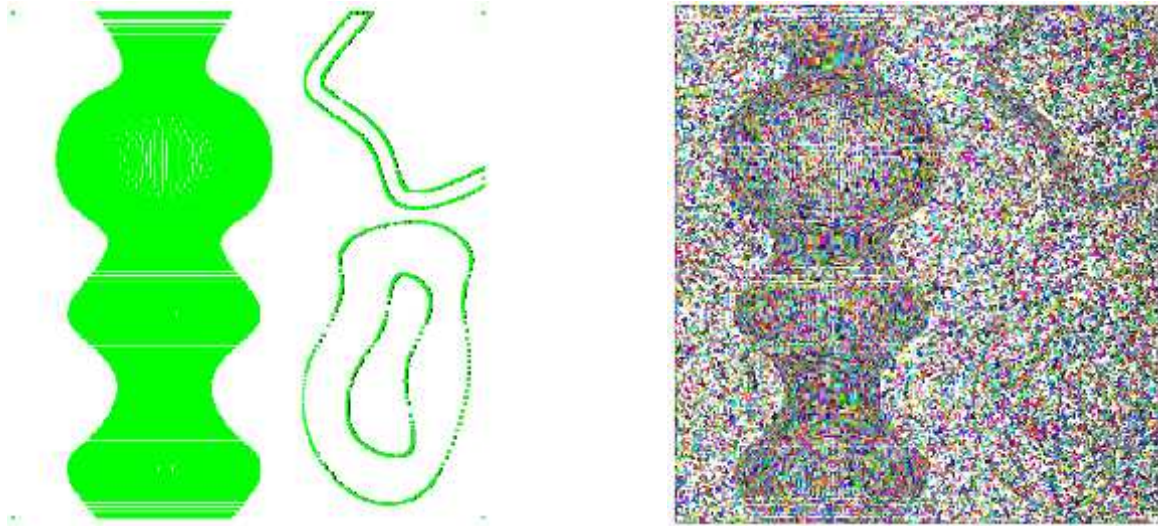
Lorsque le modèle STL est grossier, des problèmes d'interférences et de collisions peuvent exister sans être détectés. Ceci est dû au fait que la densité des points n'est pas importante et l'outil peut pénétrer dans la surface sans être découvert. Afin de remédier à ce problème, la densité des points est augmentée. L'approche utilisée consiste à ajouter de nouveaux points dans les triangles en se basant sur les coordonnées barycentriques. Le nombre de points à ajouter pour chaque triangle est déterminé selon l'aire du triangle et le nombre spécifié par l'utilisateur (Figure 3). Pour chaque triangle du modèle de sommets **P1**, **P2** et **P3**, en utilisant les coordonnées barycentriques, un point **P** quelconque du plan du triangle a pour coordonnées (21) :

$$\vec{0} = * \vec{0}_1 + * \vec{0}_2 + * \vec{0}_3 \quad (4)$$

Avec :

$$= (1-T), \quad = T*(1-S), \quad = (T*S) \quad (5)$$

$$S = r_1 \quad (0, 1), \quad T = \sqrt{r_1} \quad (0, 1) \quad (6)$$



a. Avant enrichissement.

b. Après enrichissement.

Figure 22. Enrichissement du modèle STL.

I.2.2. Création des cellules

La création des cellules consiste à subdiviser le brut en blocs parallélépipédiques de même taille. Cette subdivision nécessite la spécification du nombre de cellules N_x , N_y et N_z suivant les trois axes X , Y et Z . A partir de ces données, les pas et les limites de chaque cellule sont calculés (Figure 4). Les pas suivant les axes X , Y et Z , sont donnés par (13) :

$$\text{Pas}_x = \text{longueur du brut} / N_x \quad (7)$$

$$\text{Pas}_y = \text{largeur du brut} / N_y \quad (8)$$

$$\text{Pas}_z = \text{hauteur du brut} / N_z \quad (9)$$

Les limites de chaque cellule sont calculées comme suit :

Extrémités minimales :

$$\text{Pour } i \text{ de } 0 \text{ à } N_x \quad \mathbf{Xmin} = \mathbf{Brut_Xmin} + i * \text{Pas}_x \quad (10)$$

$$\text{Pour } j \text{ de } 0 \text{ à } N_y \quad \mathbf{Ymin} = \mathbf{Brut_Ymin} + j * \text{Pas}_y \quad (11)$$

$$\text{Pour } k \text{ de } 0 \text{ à } N_z \quad \mathbf{Zmin} = \mathbf{Brut_Zmin} + k * \text{Pas}_z \quad (12)$$

Extrémités maximales :

$$\text{Pour } i \text{ de } 0 \text{ à } N_x \quad \mathbf{X_{max}} = \mathbf{X_{min}} + \mathbf{Pas_x} \quad (13)$$

$$\text{Pour } j \text{ de } 0 \text{ à } N_y \quad \mathbf{Y_{max}} = \mathbf{Y_{min}} + \mathbf{Pas_y} \quad (14)$$

$$\text{Pour } k \text{ de } 0 \text{ à } N_z \quad \mathbf{Z_{max}} = \mathbf{Z_{min}} + \mathbf{Pas_z} \quad (15)$$

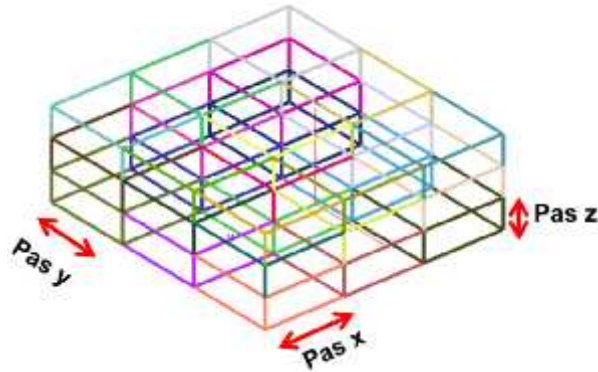


Figure 23. Création des cellules.

I.2.3. Affectation des points aux cellules

Après la création des cellules, chaque sommet des triangles (les sommets originaux et les points ajoutés) du modèle STL doit être affecté à la cellule correspondante. L'affectation des sommets aux cellules consiste à déterminer les indices de la cellule correspondante (13). Ceux-ci sont calculés en se basant sur les coordonnées (X, Y et Z) du point considéré et les coordonnées minimales du brut de la manière suivante (Figure 5) :

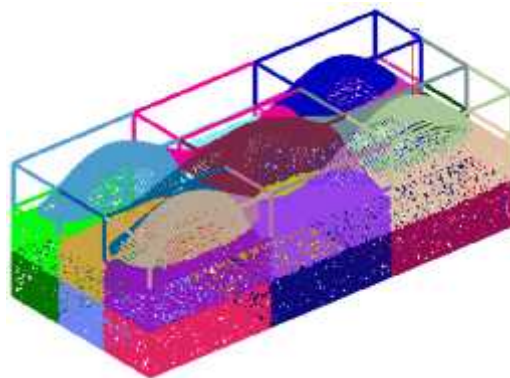


Figure 24. Affectation des points aux cellules.

$$\mathbf{I} = \frac{\mathbf{x} - \mathbf{X_{m_b}}}{\mathbf{P_x}} \quad (16)$$

$$\mathbf{J} = \frac{\mathbf{y} - \mathbf{Y_{m_b}}}{\mathbf{P_y}} \quad (17)$$

$$\mathbf{K} = \frac{\mathbf{z} - \mathbf{Z_{m_b}}}{\mathbf{P_z}} \quad (18)$$

I.2.4. Affectation des triangles aux cellules

Chaque triangle du modèle STL doit être affecté à la cellule correspondante (Figure 6). Ceci permet de limiter la zone de vérification des problèmes de collisions et d'interférences avec les triangles où se positionne l'outil au lieu de vérifier avec tous les triangles du modèle d'une part, et d'autre part pour accélérer le calcul de l'accessibilité (les triangles accessibles à l'usinage). L'affectation des triangles aux cellules consiste à déterminer les indices de la/des cellule(s) correspondante(s). Ceux-ci sont calculés en se basant sur les indices des cellules des trois sommets de chaque triangle.

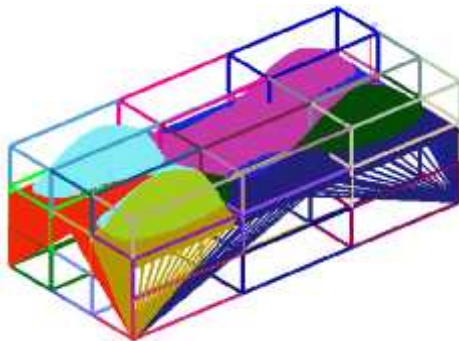


Figure 25 . Affectation des triangles aux cellules.

I.3. Faces de la pièce

Avant de passer à l'étape de génération des points de passage des outils d'usinage, il est nécessaire d'identifier la nature de chaque face triangulaire.

- **Face verticale** : un triangle est vertical si sa normale est de la forme $\vec{N}(n_x, n_y, 0)$.
- **Face proche de la verticale** : d'abord, il faut fixer l'angle θ pour les triangles qui sont considérés comme des triangles proches de la verticale. Ensuite, l'angle θ entre la normale de chaque triangle $\vec{n} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$ et l'axe $\vec{z} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ est calculé avec la formule suivante (22):

$$\theta = \frac{\vec{n} \cdot \vec{z}}{\|\vec{n}\| \cdot \|\vec{z}\|} \quad (19)$$

Avec :

$\|\vec{n}\|$: Norme du vecteur.

$\vec{n} \cdot \vec{z}$: Produit scalaire des deux vecteurs.

Si $\theta < \theta_{max}$: alors le triangle est proche de la verticale.

- **Face horizontale** : le triangle est horizontal si sa normale est de la forme $\vec{N}(0, 0, n_z)$.

I.4. Détermination des zones visibles et des zones accessibles

I.4.1. Détermination des triangles visibles

Chaque triangle du modèle STL est défini par ses trois sommets et sa normale qui est définie par ces coordonnées $\vec{N}(n_x, n_y, n_z)$. Pour déterminer la visibilité géométrique d'un triangle indépendamment des autres triangles, il est nécessaire de vérifier la composante **nz** de sa normale \vec{N} :

\vec{N} Si $n_z \geq 0$, alors le triangle est visible.

\vec{N} Sinon, le triangle est invisible.

I.4.2. Détermination des sommets accessibles

Pour déterminer l'accessibilité d'un sommet, il faut vérifier qu'il n'est pas caché par un autre triangle (l'appartenance à un autre triangle). Pour vérifier l'appartenance d'un point **M** à un triangle **ABC** défini par ses trois côtés **AB**, **AC** et **BC**, il faut vérifier d'abord si le point **M** est localisé sous le triangle et appartient à ses limites. Si cette condition est vérifiée, les paramètres suivants sont calculés selon les deux modes suivants.

Premier mode: (23)

$$= (\vec{A} \otimes \vec{A}) \cdot (\vec{A} \otimes \vec{A}) \quad (20)$$

$$= (\vec{B} \otimes \vec{B}) \cdot (\vec{B} \otimes \vec{B}) \quad (21)$$

$$= (\vec{C} \otimes \vec{C}) \cdot (\vec{C} \otimes \vec{C}) \quad (22)$$

Où « \otimes » désigne le produit vectoriel de deux vecteurs et « \cdot » désigne le produit scalaire.

\vec{N} Si $0, 0$ et 0 , alors le point appartient au triangle.

\vec{N} Sinon, le point n'appartient pas au triangle.

Deuxième mode : (24)

Calculer l'aire du triangle **ABC** et l'aire des sous triangles **AMB**, **AMC** et **BMC**. Par la suite, utiliser la formule de Héron pour vérifier si le point appartient au triangle ou non. Soit **P** le demi-périmètre du triangle **ABC** donné par :

$$P = \frac{1}{2} (AB + AC + BC) \quad (23)$$

Alors l'aire du triangle **ABC** est donnée par :

$$\text{aire} = \sqrt{p \cdot (p - A) \cdot (p - B) \cdot (p - C)} \quad (24)$$

Ñ Si $\text{aire}(\text{AMB}) + \text{aire}(\text{AMC}) + \text{aire}(\text{BMC}) = \text{aire}(\text{ABC})$, alors le point appartient au triangle (Figure 7.a).

Ñ Sinon, le point n'appartient pas au triangle (Figure 7.b).

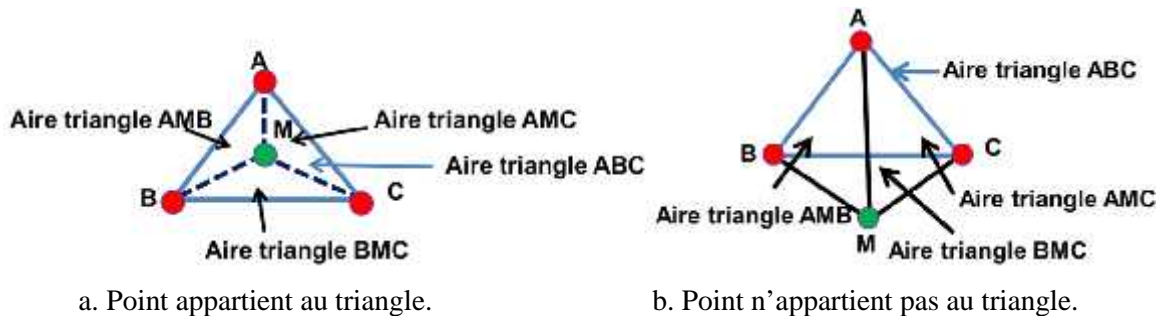


Figure 26. Vérification de l'appartenance d'un point à un triangle.

I.4.3. Détermination des triangles accessibles

Pour déterminer l'accessibilité d'un triangle, il est nécessaire de vérifier que le triangle est visible et que ses trois sommets sont accessibles.

Ñ Si le triangle est visible et ses trois sommets S_1 , S_2 et S_3 sont accessibles, alors le triangle est accessible.

Ñ Sinon, le triangle est inaccessible.

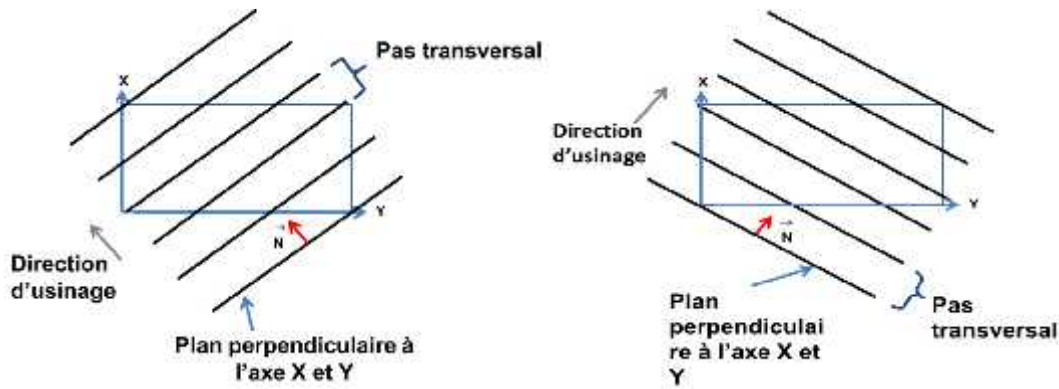
I.5. Création des plans verticaux

L'approche d'usinage utilisée est basée sur la création de plans verticaux. La distance entre ces plans est constante et elle est égale au pas transversal fixé par l'utilisateur en fonction de la précision désirée (erreur de crête). Le nombre de plans dépend du pas transversal, de l'angle d'orientation du plan et des dimensions du brut de la pièce. Pour cela, un vecteur des plans parallèles entre eux et parallèle au plan \mathbf{XY} est créé (Figure 8). Chaque plan créé est défini par son équation déterminée après le choix des paramètres suivants :

Ñ Angle entre l'axe \mathbf{X} et le plan (de 0° à 180°).

Ñ Point de passage du plan.

L'orientation des plans dépend de la géométrie de la pièce. Donc, il est nécessaire de choisir l'orientation qui génère des chemins les plus longs afin de minimiser le nombre d'engagements et de dégagements des outils.



a. Angles dans l'intervalle [0-90].

b. Angles dans l'intervalle [90-180].

Figure 27. Plans verticaux en fonction des angles d'orientation.

I.6. Création des contours

I.6.1. Calcul des segments d'intersection

Pour calculer les segments d'intersection, il faut calculer les points d'intersection entre les plans verticaux et les segments de chaque triangle du modèle.

I.6.1.1. Intersection d'un triangle et un plan

Pour qu'un plan () soit en intersection avec un triangle il faut et il suffit qu'il coupe au moins un segment du triangle. Pour que l'intersection entre un plan et un triangle soit un ensemble vide, il faut qu'il existe deux segments de triangle tels que l'intersection entre chacun d'eux et le plan soit un ensemble vide (Figure 9).

Soit ABC un triangle telle que $A(x_0, y_0, z_0)$, $B(x_1, y_1, z_1)$ et $C(x_2, y_2, z_2)$, () un plan à situer à la position V_0 avec \vec{n} sa normal. Chercher l'intersection entre le triangle ABC et le plan () revient à trouver l'intersection entre le segment $[AB]$ et le plan (), le segment $[AC]$ et le plan (), le segment $[BC]$ et le plan (). Le résultat de l'intersection est soit **un segment** soit **un point** (Figure 9).

➤ Calculer de l'intersection entre un segment et un plan : (25)

Pour obtenir l'intersection d'un segment $[AB]$ avec un plan () de normale \vec{n} situé à la position V , il faut déterminer d'abord la valeur du paramètre « r » suivant :

$$\mathbf{r} = \frac{\mathbf{n} \cdot (V_0 - A)}{\mathbf{n} \cdot (B - A)} \quad (25)$$

Lorsque le dénominateur $\mathbf{n} \cdot (B - A)$ est égale à 0 , le segment $[AB]$ est parallèle au plan (). Dans ce cas, soit le segment ne coupe pas le plan soit il est complètement dans le plan

(chaque fois que \mathbf{A} et \mathbf{B} est dans la plan ()). Sinon, lorsque le dénominateur est non nul et que le paramètre « r » est un nombre réel, le segment $[\mathbf{AB}]$ coupe le plan () uniquement quand $0 \leq r \leq 1$ et l'intersection est un point. Le point d'intersection \mathbf{m} est donné par :

$$\mathbf{m} = \mathbf{A} + r * (\mathbf{B}-\mathbf{A}) \quad (26)$$

Le même principe est appliqué pour calculer l'intersection entre les segments $[\mathbf{AC}]$ et $[\mathbf{BC}]$ avec le plan ().

➤ **Calculer les coordonnées des points d'intersection**

Deux cas possibles sont à distinguer :

- 1^{er} cas : le plan passe par un sommet du triangle.
- 2^{ème} cas : le plan passe par une ou deux arêtes du triangle en un point différent des sommets des triangles.

Pour le premier cas, les coordonnées du sommet sont déjà calculées. Pour vérifier si un sommet appartient à un plan, il suffit juste de remplacer les coordonnées du sommet dans l'équation du plan.

Pour le deuxième cas, l'intersection d'une droite avec un plan est calculée pour déterminer les coordonnées \mathbf{X} , \mathbf{Y} et \mathbf{Z} du point d'intersection \mathbf{m} .

La Figure 11 résume les différents cas de l'intersection d'un plan avec un triangle.

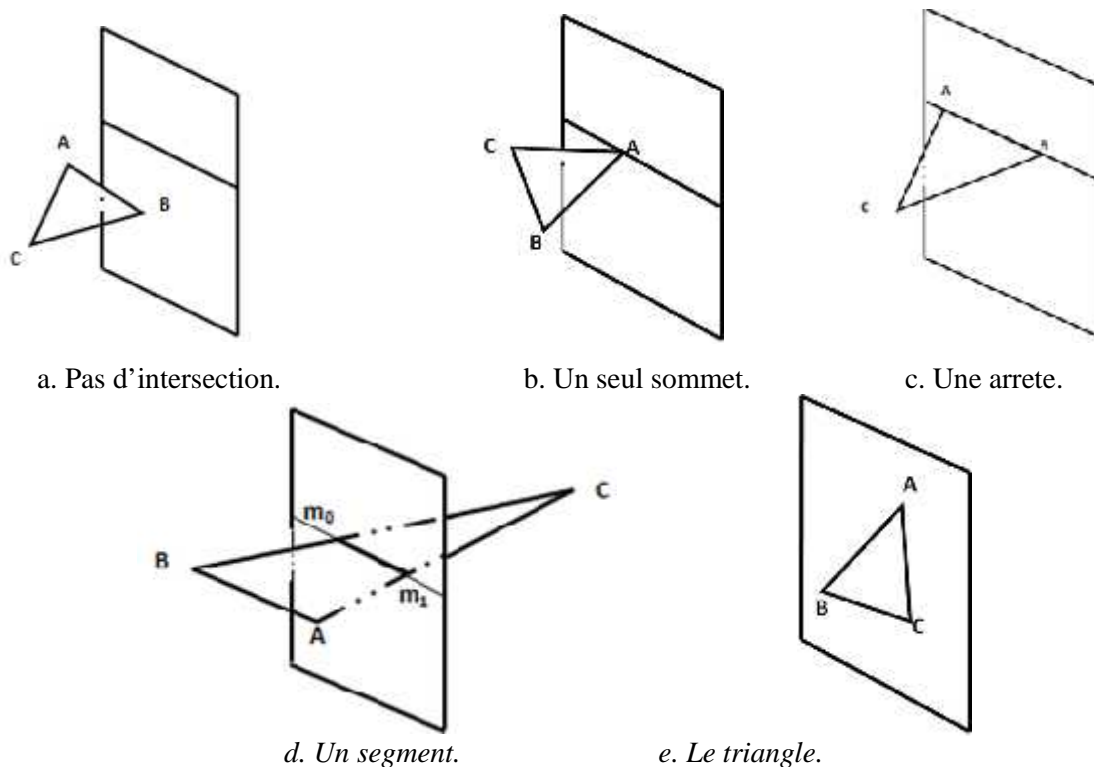


Figure 28. Différents cas d'intersection entre un plan et un triangle.

- **1^{er} cas** : le plan ne passe pas par le triangle. Donc, pas d'intersection (Figure 9.a).
- **2^{ème} cas** : l'intersection du plan avec un triangle est un seul sommet. Cette situation n'est pas considérée puisqu'elle ne génère pas un segment (Figure 9.b).
- **3^{ème} cas** : une arête du triangle appartient au plan, c'est-à-dire deux sommets du triangle appartiennent au plan (Figure 9.c).
- **4^{ème} cas** : le plan passe par deux arêtes. Donc, un seul segment est généré. Il est formé par les deux points d'intersection avec les deux arêtes (Figure 9.d).
- **5^{ème} cas** : les trois arêtes du triangle appartiennent au plan, Donc, tous les segments du triangle sont générés (Figure 9.e).

➤ **Estimation de la normale au point d'intersection :**

Si le point d'intersection est un sommet, alors sa normale est déjà calculée dans la première étape (lecture de fichier STL). Dans le cas où le point d'intersection est un nouveau point calculé, alors la normale est calculée en utilisant la pondération par les distances par rapport aux sommets qui forment l'arête [$P_0 P_1$].

$$\vec{N}_m = \frac{L_0 \vec{N}_{p0} + L_1 \vec{N}_{p1}}{L_0 + L_1} \quad (27)$$

Avec :

L_0 : Distance entre le point m et le point P_0 .

L_1 : Distance entre le point m et le point P_1 .

\vec{N}_{p0} : Normale du point P_0 .

\vec{N}_{p1} : Normale du point P_1 .

I.6.2. Génération des segments

La génération des segments est réalisée au cours du calcul de l'intersection entre l'ensemble des triangles accessibles et les plans verticaux où un segment est formé par deux points d'intersection entre un plan et un triangle.

I.6.3. Chaînage des segments et création des contours

La stratégie utilisée pour l'usinage est « Plans Parallèles ». Elle consiste à générer des contours d'usinage qui forment le trajet primaire. Ce trajet est obtenu par l'intersection des triangles de surface avec des plans parallèles verticaux. La surface étant représentée par le

modèle STL, l'intersection des plans avec les triangles génère des points qui sont reliés entre eux pour former une chaîne fermée ou ouverte appelé « contour ».

Après la génération de la liste globale des segments d'intersection d'un plan, les segments voisins (segments ayant un point commun) sont chaînés pour former un contour. A noter qu'un plan peut contenir plusieurs contours différents et disjoints.

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

- Initialiser la liste des segments chaînés d'un contour par un segment retiré de la liste globale des segments non chaînés.

- Parcourir la liste globale et chercher le segment qui suit le dernier segment inséré dans le contour. Pour vérifier si un segment est le suivant, il faut vérifier si le point de fin du premier segment est le même point de début du deuxième segment. Donc, il suffit de vérifier l'égalité des trois coordonnées **X**, **Y** et **Z** des deux points pour dire que les deux segments sont connectés. Une fois le segment suivant déterminé, il est inséré après le premier segment dans le contour (dans la liste des segments) et il est supprimé de la liste globale des segments pour éviter une boucle infinie. Cette procédure est répétée pour chaque dernier segment suivant trouvé et inséré dans le contour.

- Même chose pour rechercher les segments précédents (en parallèle avec la recherche des segments suivants), le premier segment inséré dans le contour est pris et les précédents de ce segment sont recherchés et insérés avant ce segment avec une logique inversée par rapport à la logique utilisée pour les segments suivants, c'est-à-dire vérification de la coïncidence du point de début du segment sélectionné avec le point de fin de son précédent.

Une fois tous les suivants et tous les précédents sont déterminés, le contour est créé. Ce contour est inséré dans la liste des contours du plan sélectionné.

I.7. Choix des outils optimums

I.7.1. Positionnement des outils

Une fois les points d'intersections sont calculés, ces points sont utilisés pour positionner l'outil par rapport à la surface à usiner. Le calcul de la position (point centre et extrémité) de l'outil par rapport à la surface à usiner repose sur l'ensemble des paramètres suivants :

- Forme géométrique de l'outil (partie active).
- Point de contact outil-surface (coordonnées et normale).
- Rayon d'outil et rayon de coupe respectivement R et r , et l'axe d'orientation d'outil \vec{u} .

Les coordonnées du point centre outil et extrémité pour les trois formes sont données par la formule suivante (19):

$$\mathbf{O}_E = \mathbf{O}_C + r\vec{n} + \mathbf{R}\cdot\vec{v} \quad (28)$$

$$\mathbf{O}_L = \mathbf{O}_E - r\cdot\vec{u} \quad (29)$$

Une fois le point extrémité est calculé, l'extrémité maximale d'outil de la partie active et du porte outil sont calculées comme suit (19):

$$\mathbf{O}_L = \mathbf{O}_L + L\cdot\vec{u} \quad (30)$$

$$\mathbf{O}_L = \mathbf{O}_L + L_A\cdot\vec{u} \quad (31)$$

Avec :

$$\vec{v} = \frac{\vec{u} \cdot \vec{n}}{\|\vec{u} \cdot \vec{n}\|} \vec{u} \quad (32)$$

L : longueur du corps de l'outil.

L_A : longueur de la partie active de l'outil.

Par la suite, les enveloppes qui englobent les trois parties de l'outil sont déterminées.

- **Enveloppe de la partie corps** : les extrémités minimales et maximales de l'enveloppe de la partie corps sont données par (19) :

$$\mathbf{L}_m = \mathbf{C}_L - \mathbf{R} \quad (33)$$

$$\mathbf{L}_m = \mathbf{C}_L + \mathbf{R} \quad (34)$$

- **Enveloppe de la partie active** : les extrémités minimales et maximales de l'enveloppe de la partie active sont données par (19) :

$$\mathbf{L}_m = \mathbf{C}_L - \mathbf{R} \quad (35)$$

$$\mathbf{L}_m = \mathbf{C}_L + \mathbf{R} \quad (36)$$

I.7.2. Affectation des outils optimums aux sommets du trajet

L'objectif de ce travail consiste à optimiser l'opération de finition des surfaces gauches par l'affectation des outils optimums permettant l'évitement des interférences et des collisions. Une des deux formes d'outils cylindriques ou hémisphériques est affectée à chaque point de contact du trajet en considérant les critères suivants.

- Si un point de contact appartient à un segment de forme horizontale, alors le point prend la même forme et un outil cylindrique lui est affecté.

- Si un point de contact appartient à un segment de forme non horizontale, alors le point prend la même forme et un outil hémisphérique lui est affecté.
- Si un point de contact se situe entre un segment de forme horizontale et un autre non horizontale, alors le point est dupliqué et prend les deux formes et un outil hémisphérique et cylindrique lui sont affectés.

Les différentes étapes d'affectation des outils optimums se déroulent comme suit :

- Affecter l'outil du plus grand rayon au point de contact.
- Calculer le positionnement d'outil.
- Déterminer la zone d'interférence et de collision (chevauchement avec le corps ou la partie active de l'outil) (voir gestion des collisions et des interférences).
- Vérifier s'il existe un problème d'interférence et/ou de collision (voir gestion des collisions et des interférences). Si oui, alors retourné à la première étape avec un outil plus petit que le précédent.

Les étapes précédentes sont répétées jusqu'à l'obtention d'un outil optimum.

I.7.3. Affectation des outils optimums au trajet

Après affectation des outils à chaque point de contact, il s'agit de définir le nombre et la forme des outils à utiliser pour tout le trajet. Le nombre d'outils pour tout le trajet est fixé à quatre (04) outils cylindriques et quatre (04) outils hémisphériques dans l'ordre suivant :

- Le plus petit outil optimum.
- Le premier outil optimum le plus utilisé parmi les outils restants.
- Le deuxième outil optimum le plus utilisé parmi la liste des outils restants.
- Le plus petit outil optimum parmi la liste des outils restants.

I.8. Vérification des collisions et des interférences

En raison des géométries très complexes des pièces, des problèmes d'interférences et de collisions peuvent surgir lors du positionnement de l'outil. La détermination des zones d'interférences et de collisions consiste à identifier, pour chaque point de contact, les points des cellules qui se chevauchent avec l'enveloppe de l'outil. Cette façon permet d'éviter le test avec l'ensemble des points des surfaces. La Figure 10 illustre la méthode d'identification des zones d'interférences et de collisions.

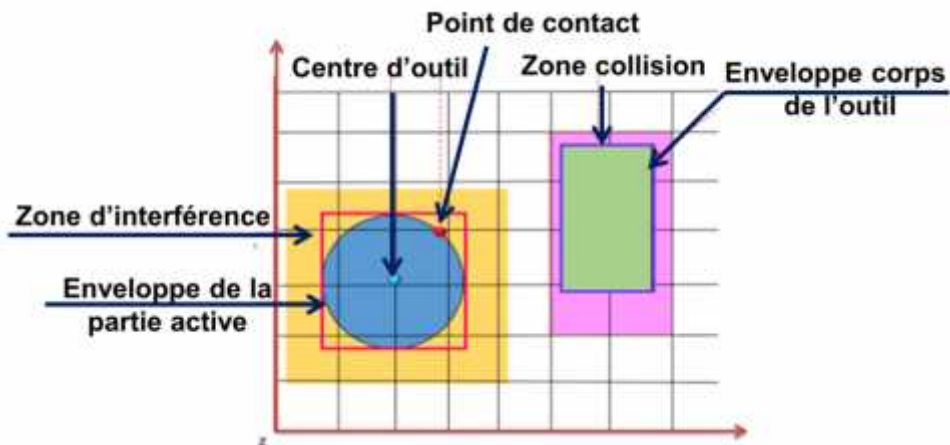


Figure 29. Zones d'interférences et de collisions.

Une fois les zones d'interférences et de collision sont déterminées, les points appartenant à ces dernières zones sont utilisés afin de tester l'existence du problème d'interférences et de collisions pour chaque point de contact.

I.8.1. Vérification des collisions

La collision se produit lorsque des points de la surface à usiner appartiennent au corps de l'outil (partie cylindrique). Comme les types d'outils (cylindrique et hémisphérique) ont la même forme de la partie « corps-outil », alors le traitement de test est le même pour les deux formes d'outils (Figure 11). Ce traitement se fait comme suit :

Pour chaque point **P** de coordonnées **X**, **Y** et **Z** des zones identifiées, la distance **D** entre le point **P** et le point centre d'outil **CE** (ou bien l'axe d'outil) est calculée en utilisant l'équation suivante :

$$D = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2} \quad (37)$$

- Si $D > R$, alors le point ne crée pas de collision.
- Si $D < R$, alors le point crée une collision.

I.8.2. Vérification des interférences

L'interférence se produit lorsque des points de la surface appartiennent à la partie active de l'outil. Le test d'interférence pour l'outil cylindrique consiste aussi à calculer la distance entre le point identifié et l'axe d'outil puis la comparer avec le rayon et vérifier l'appartenance à la partie active (Figure 12). Le test d'interférence pour l'outil hémisphérique consiste à calculer la distance entre le point identifié et le point centre outil puis la comparer avec le rayon et vérifier l'appartenance à la partie active.

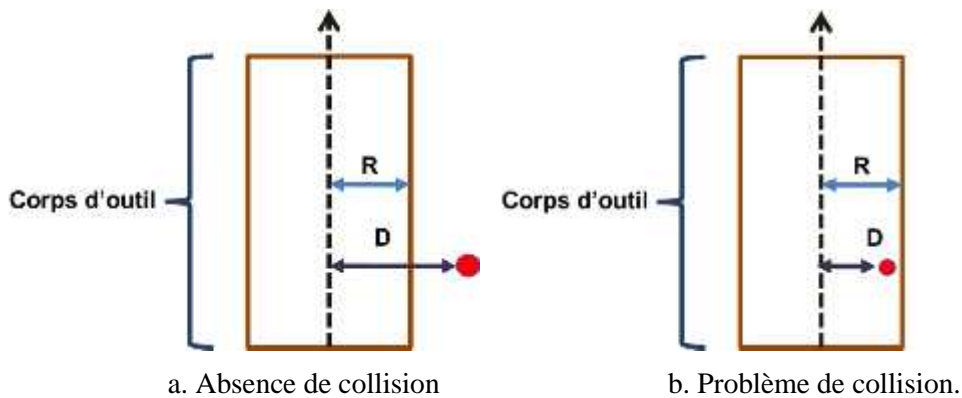


Figure 30 . Collision avec le corps-outil.

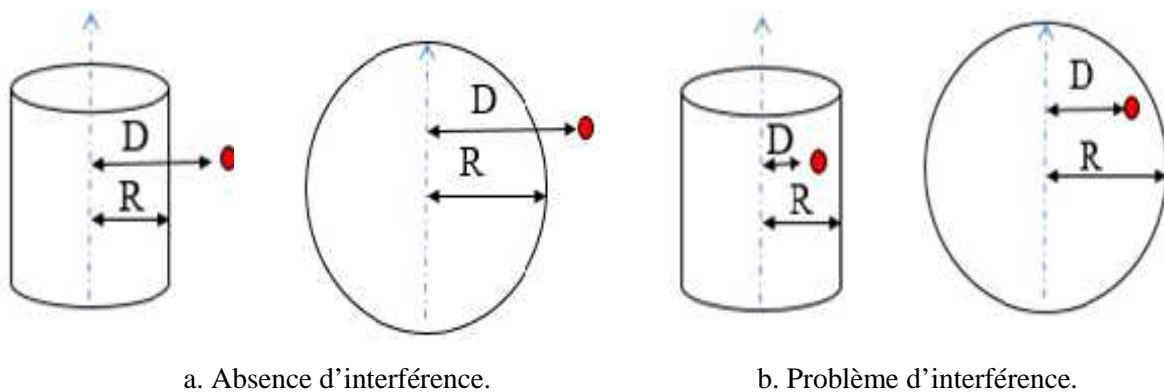


Figure 31. Interférence outil cylindrique et outil hémisphérique.

I.9. Génération du trajet d'usinage

Cette étape permet de générer les mouvements des outils lors de la finition de toutes les surfaces à usiner.

I.9.1. Création des sous contours

La création d'un sous contour d'un contour est déterminé comme suit :

Pour chaque contour donné :

- Parcourir les points de contact et chercher pour chaque point de contact si sa forme d'outil est « cylindrique ».
- Si oui, on l'ajoute dans un sous-contour cylindrique jusqu'à l'obtention d'un point de contact qui a un outil de forme hémisphérique.
- Dans le cas où un point de contact trouvé a un outil de forme hémisphérique, le sous contour précédent est créé et on commence par la création d'un nouveau sous contour hémisphérique qui regroupe les points de contact qui ont des outils hémisphériques par la même procédure précédente.

Donc, on peut trouver dans un seul contour des sous contours cylindriques et des sous contours hémisphériques.

Une fois tous les sous contours cylindriques et tous les sous contours hémisphériques sont déterminés dans un contour donné, ce contour est inséré dans la liste des contours du plan sélectionné.

I.9.2. Insertion de deux points d'engagement et de dégagement d'outil

Pour chaque sous contour, on affecte deux points d'engagement d'outil à son au début et deux points de dégagement d'outil à sa fin.

I.9.3. Construction du trajet d'usinage

➤ **Aller simple (One-Way)** : la construction des passes (trajet) en aller-simple consiste à usiner la surface selon le plan parallèle jusqu'à sa fin et ensuite déplacer l'outil en quittant la surface dans l'autre plan parallèle et refaire le même chemin jusqu'à ce que l'outil balaye toute la surface.

➤ **Aller-retour (Zig-Zag)** : la construction des passes (trajet) en aller-retour consiste à usiner la surface selon le plan parallèle jusqu'à sa fin. Ensuite, déplacer l'outil sans quitter la surface et usiner la surface dans l'autre sens et refaire le même chemin jusqu'à ce que l'outil balaye toute la surface.

II. Modélisation UML

Le langage UML « Unified Modeling Language » est un langage de modélisation orientée objet développé en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) dans le but de définir une notation standard pour la modélisation des applications construites à l'aide d'objets. C'est un « langage visuel » constitué d'un ensemble de « diagrammes ». Chacun donne une vision différente du logiciel à développer pour présenter son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées, ...etc. UML permet de :

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

Dans ce projet, trois diagrammes sont utilisés. Diagramme de cas d'utilisation, diagramme d'activité et diagramme de classe.

II.1. Diagramme de cas d'utilisation

Les cas d'utilisation permettent de recueillir, d'analyser, d'organiser les besoins et de recenser les grandes fonctionnalités du système. Nous commençons par le diagramme de cas d'utilisation général qui donne une vue globale du fonctionnement du système.

Diagramme de cas d'utilisation générale.

Le système est divisé en quatre grandes phases comme illustré par la Figure 13.

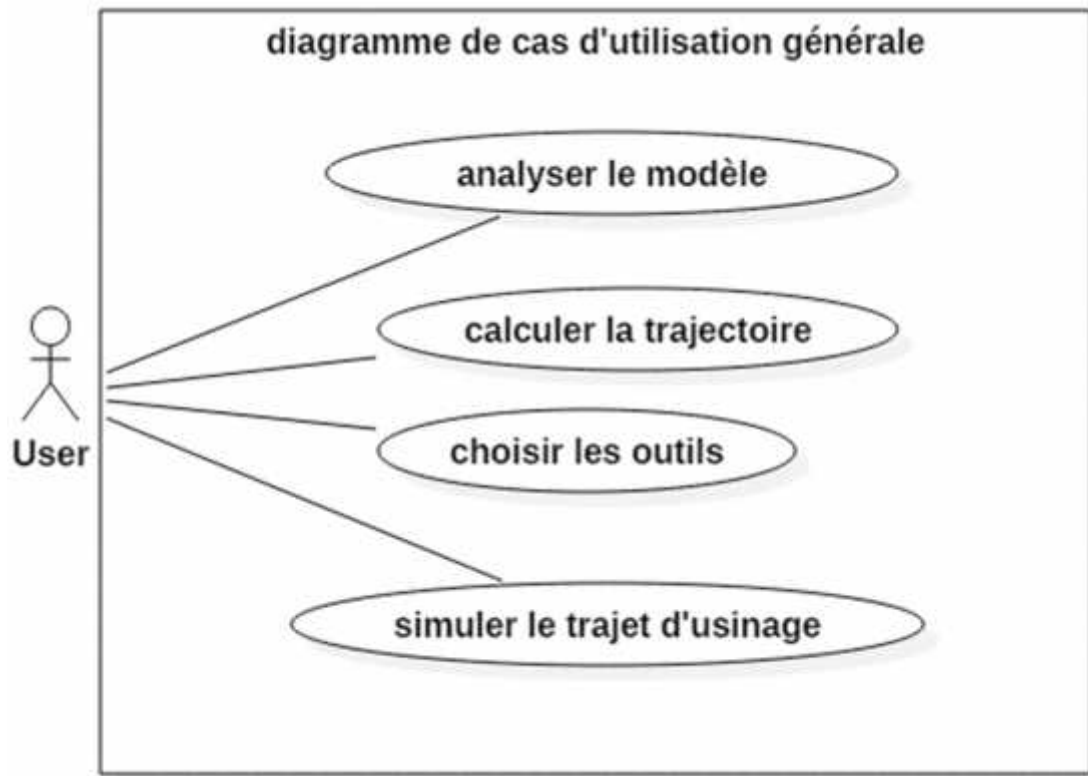


Figure 32. Cas d'utilisation générale.

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

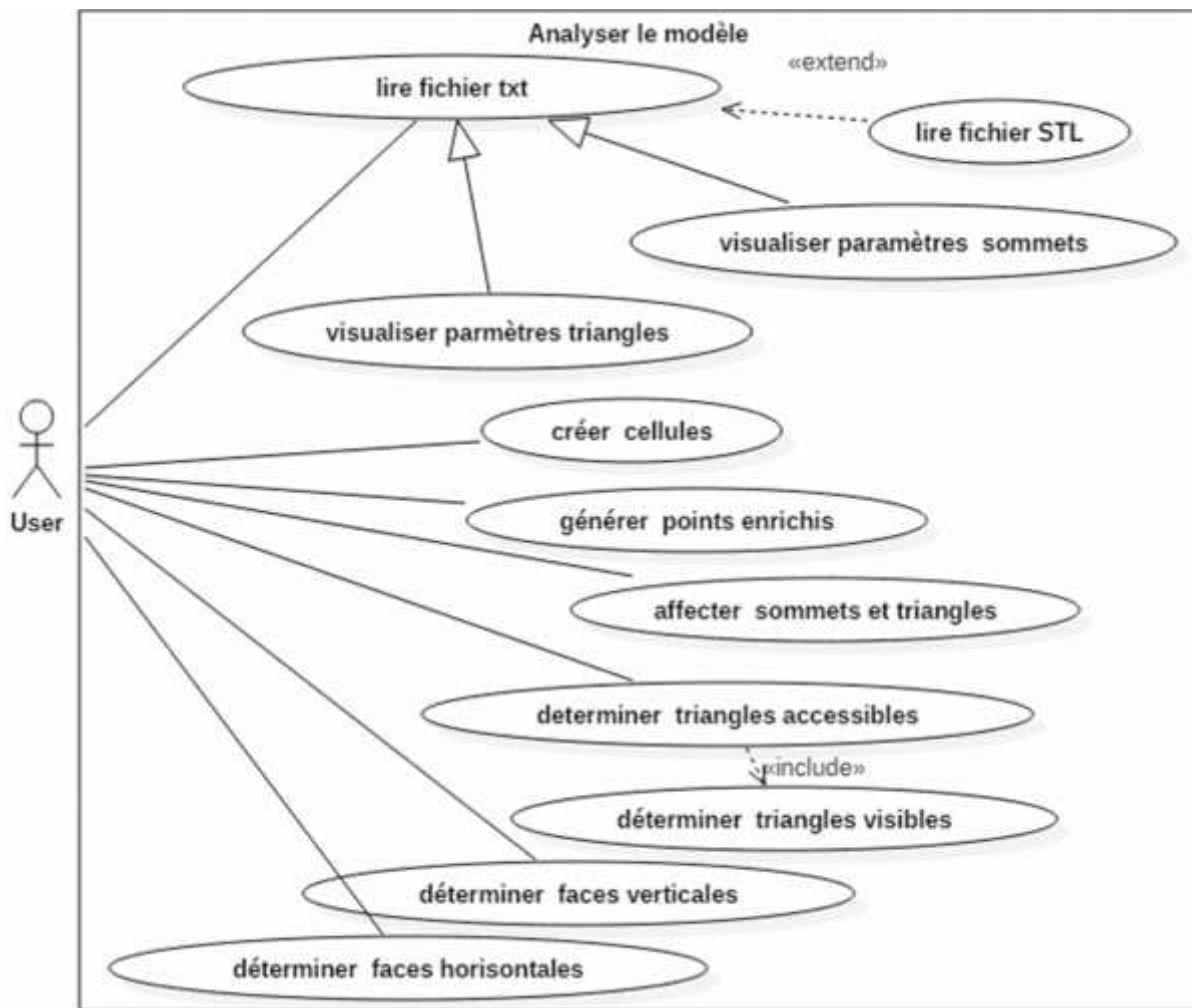


Figure 33. Cas d'utilisation « Analyser le modèle ».

Diagramme de cas d'utilisation « Calculer la trajectoire » :

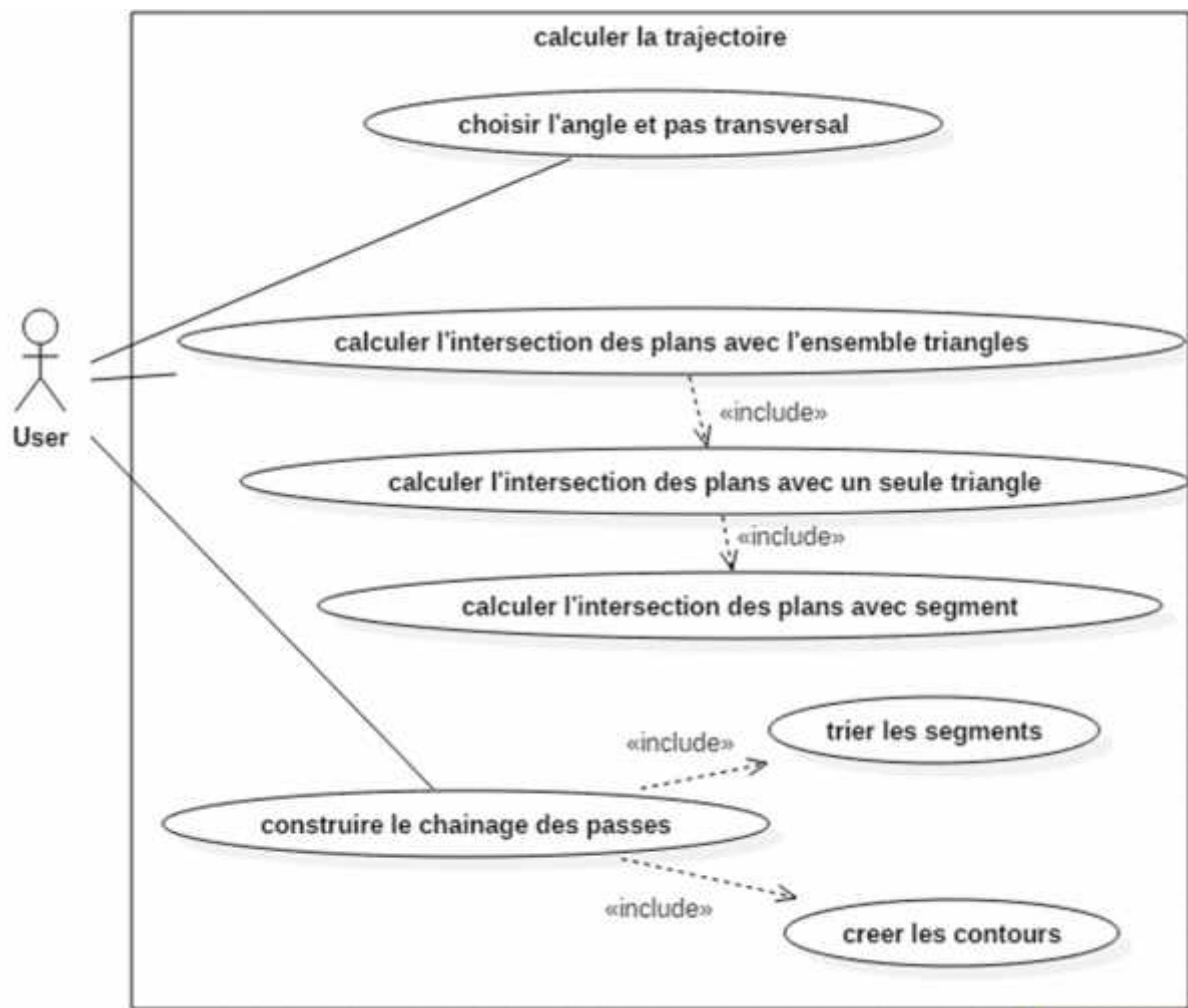


Figure 34. Cas d'utilisation « calculer la trajectoire ».

Diagramme de cas d'utilisation « Choisir les outils optimums »:

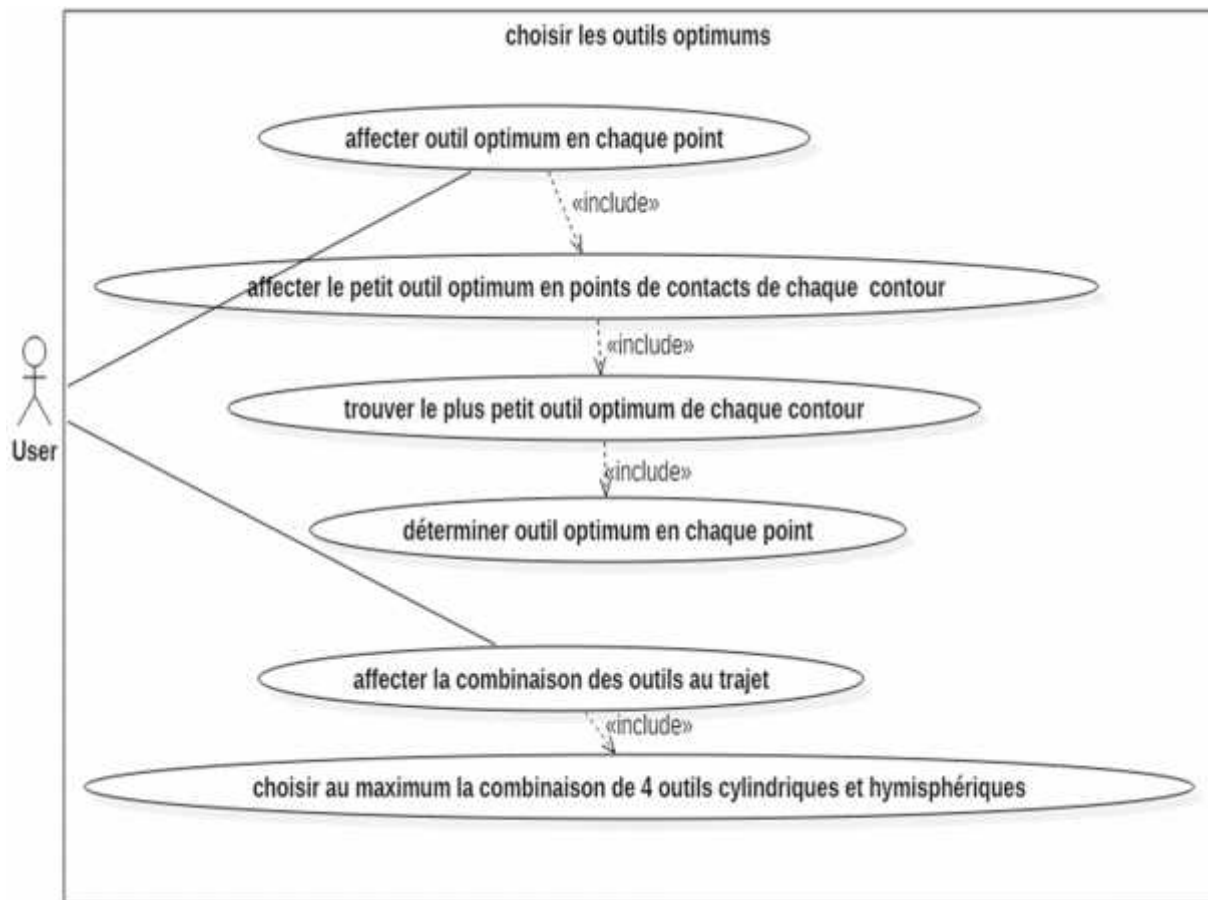


Figure 35. Cas d'utilisation « Choisir les outils optimums ».

Diagramme de cas d'utilisation « Simuler le trajet d'usinage »:

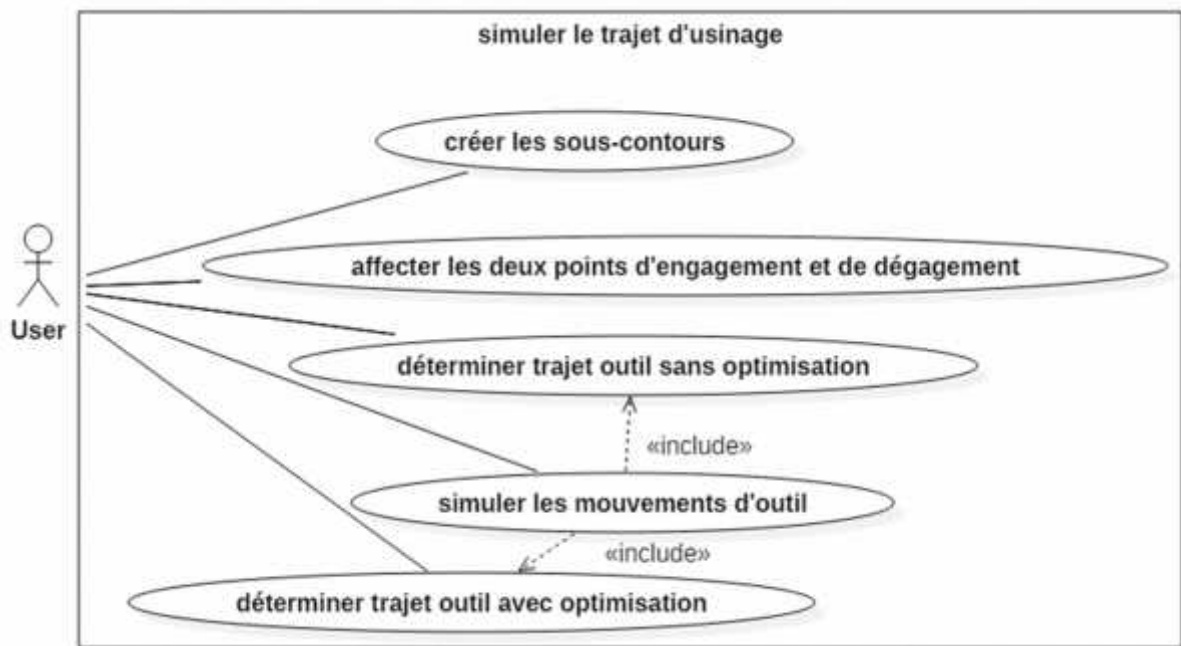


Figure 36. Cas d'utilisation «Simuler le trajet d'usinage».

II.2. Diagramme d'activité

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné. Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de [diagrammes de classes](#) ou de [cas d'utilisation](#), par exemple.

Le diagramme d'activité est une représentation proche de l'organigramme; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Diagramme d'activité général :

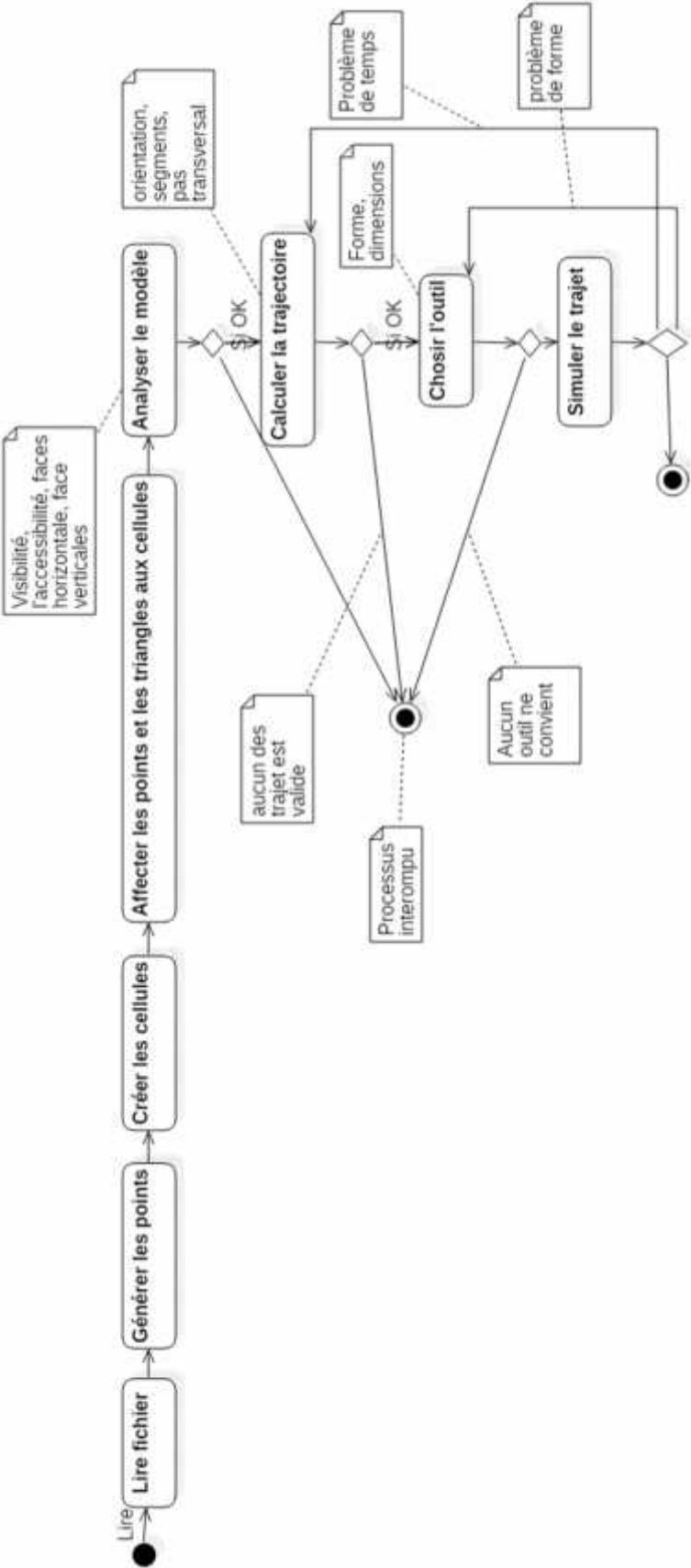


Figure 37. Diagramme d'activité général.

Diagramme d'activité « Analyser le modèle » :

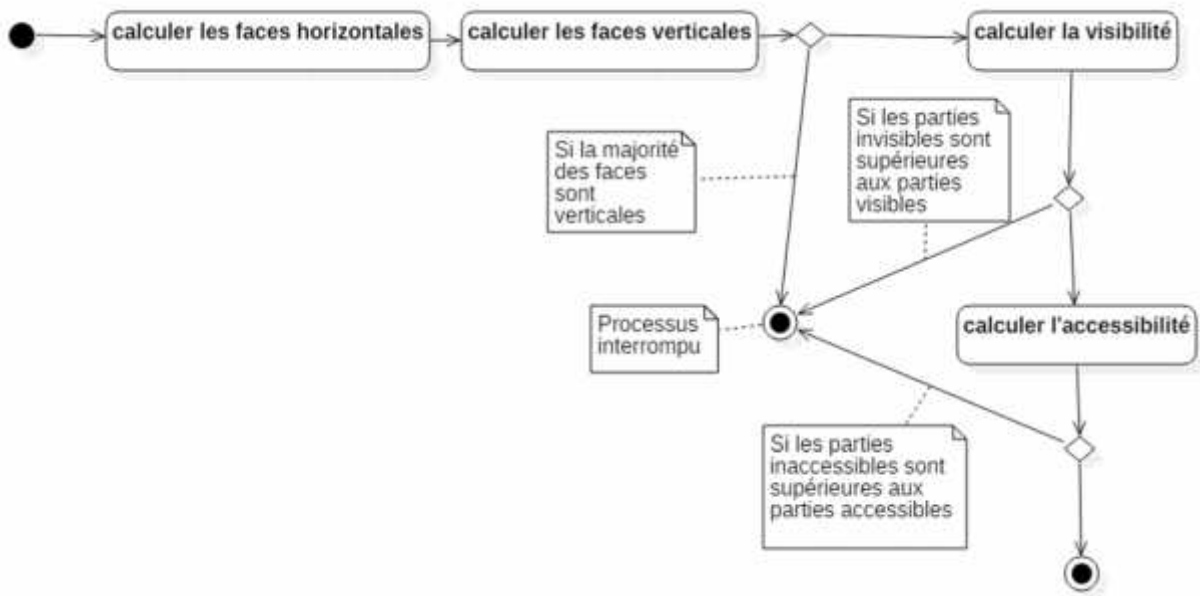


Figure 38. Diagramme d'activité « Analyser le modèle ».

Diagramme d'activité « Affecter l'outil optimum au point de contact » :

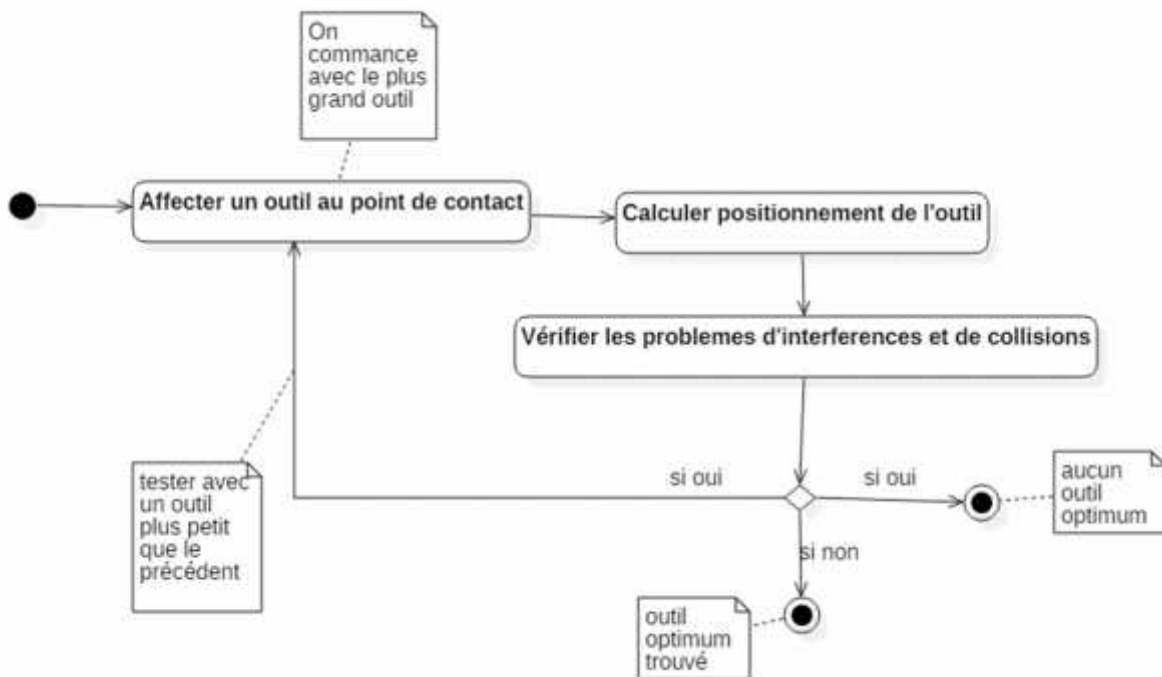


Figure 39. Diagramme d'activité « Affecter l'outil optimum au point de contact ».

Diagramme d'activité « Affecter l'outil optimum au contour » :

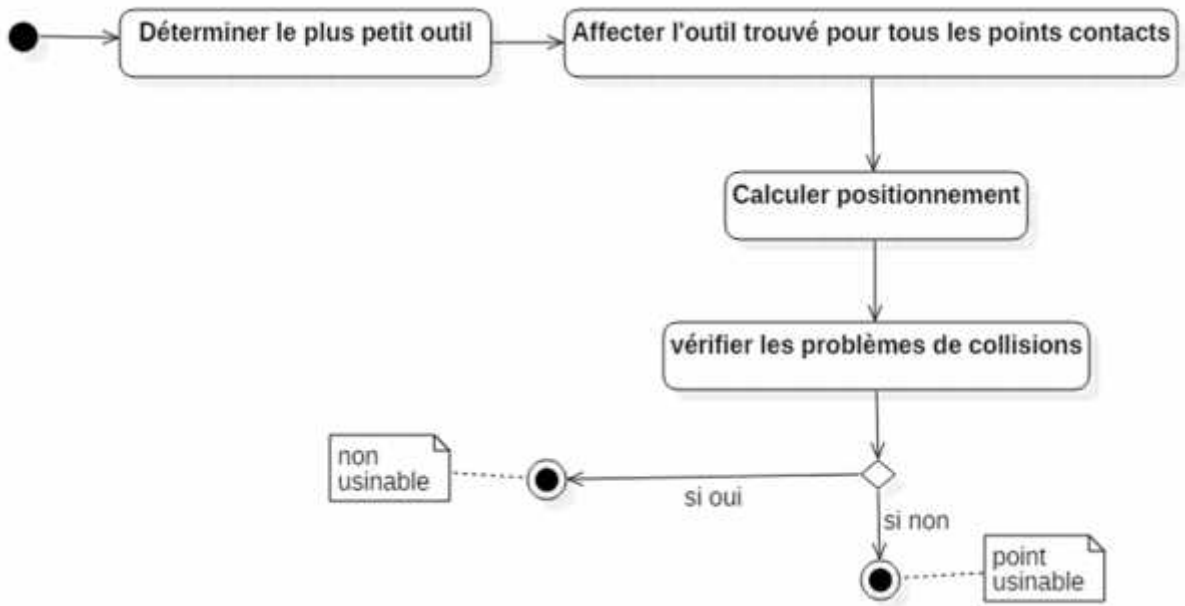


Figure 40. Diagramme d'activité « Affecter l'outil optimum au contour ».

Diagramme d'activité « Affecter les outils optiums de trajet » :

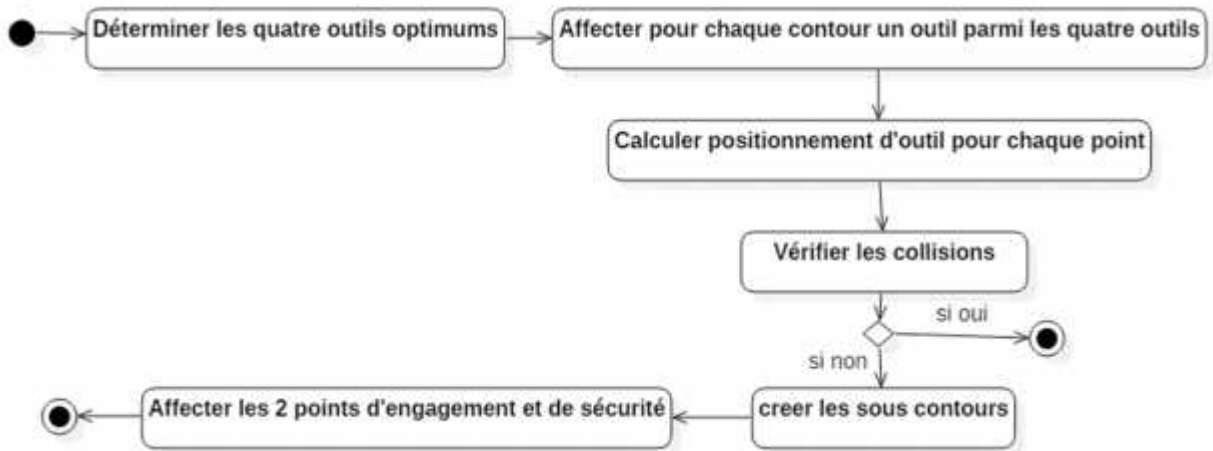


Figure 41. Diagramme d'activité « Affecter les outils optiums au trajet ».

Diagramme d'activité « Déterminer les 4 outils optimaux du trajet » :

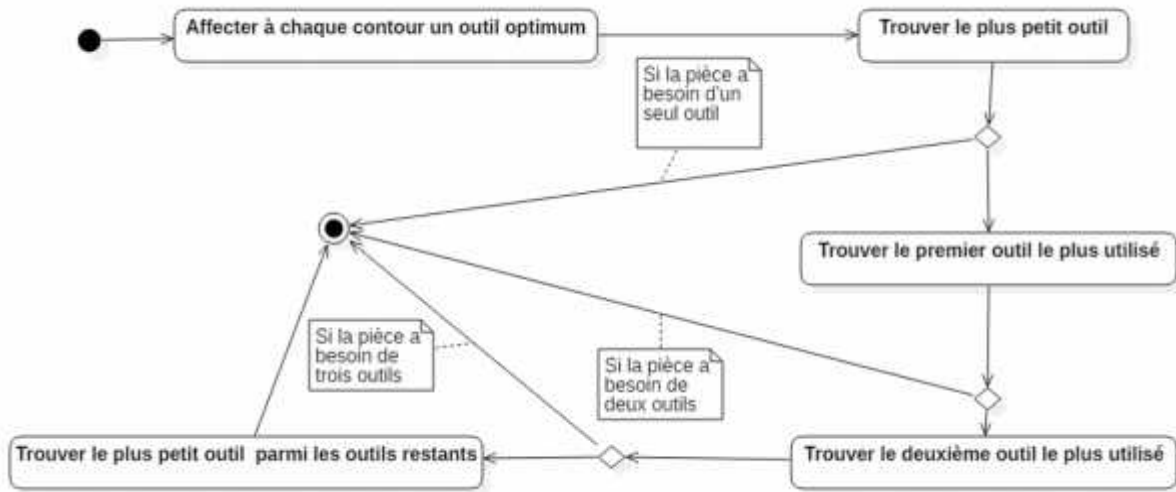


Figure 42. Diagramme d'activité « Déterminer les 4 outils optimaux ».

Diagramme d'activité « calculer la trajectoire » :

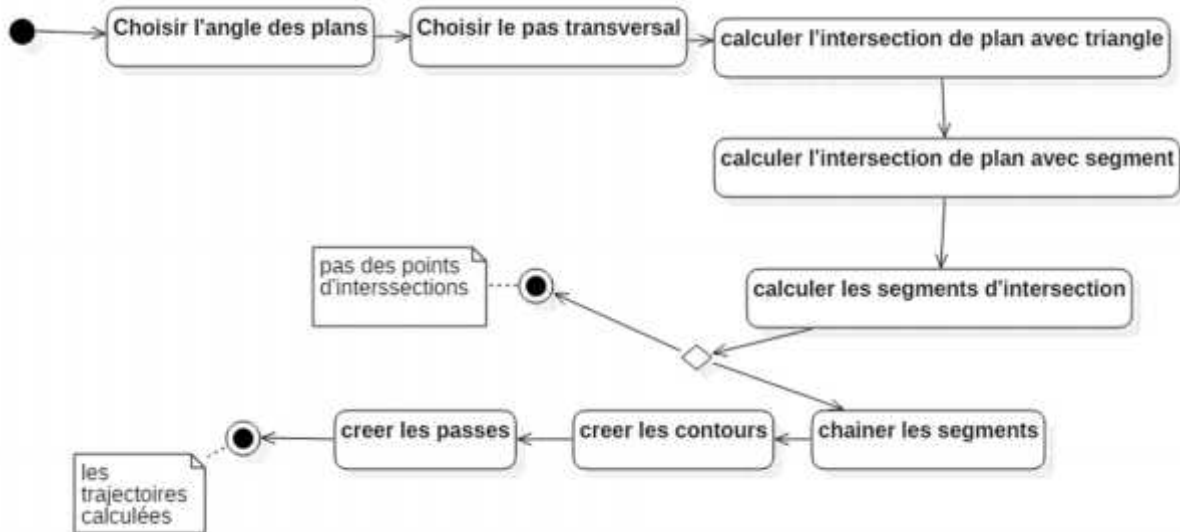


Figure 43. Diagramme d'activité « Calculer la trajectoire ».

II.3. Diagramme de classes

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système et l'ensemble des informations finalisées qui sont gérées par le domaine. Dans ce qui suit, nous allons présenter notre diagramme de classe global en tenant compte des relations entre les classes. Ensuite, nous allons détailler les principales classes utilisées. Le diagramme de classes qui représente notre système est montré par la Figure 25.

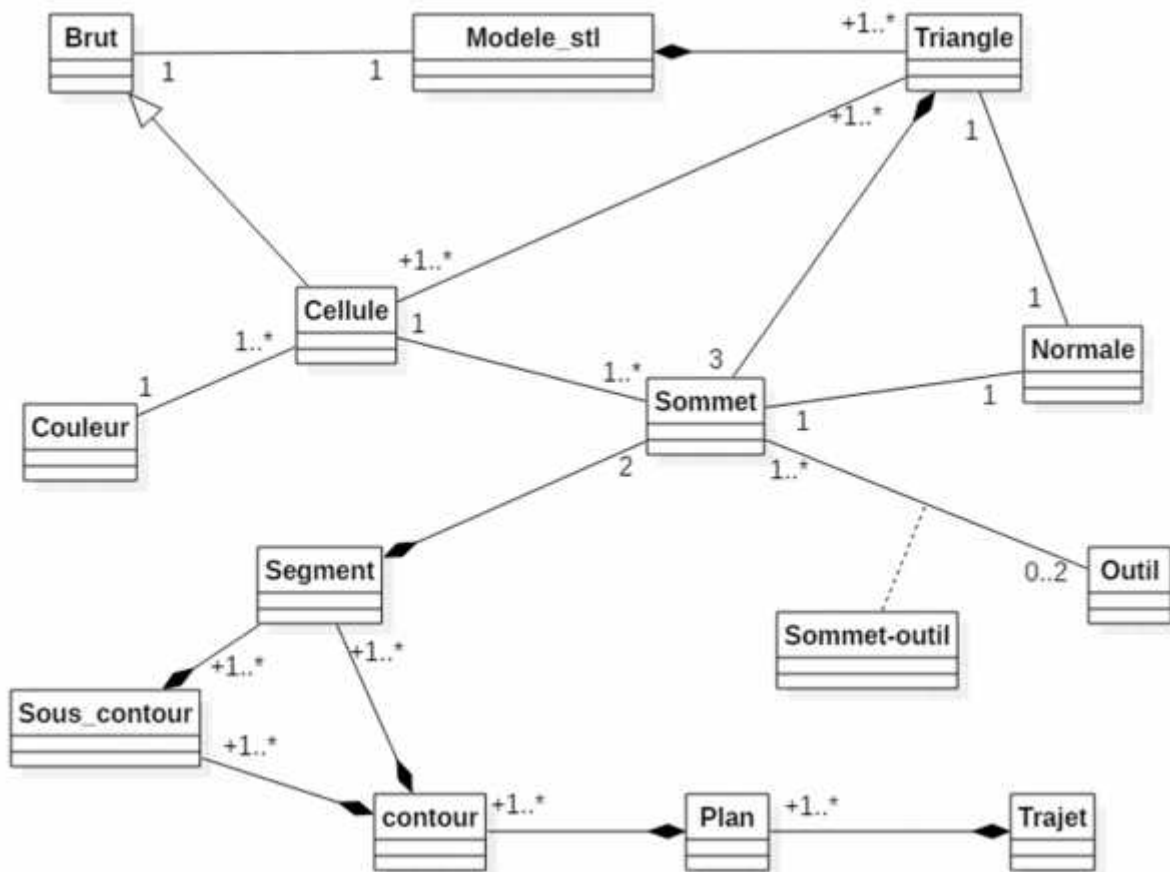


Figure 44. Diagramme de classe général.

Les classes qui définissent notre système sont détaillées comme suit :

➤ **Classe « MODELE_STL »** . c'est la classe principale qui regroupe tous les sommets et triangles du fichier STL ainsi que les cellules créées. Les méthodes de cette classe nous permettent de récupérer les principales informations du fichier STL (Figure 26).

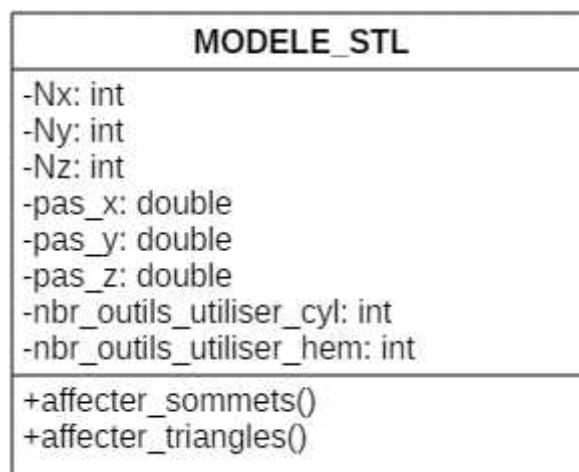


Figure 45. Classe « MODELE_STL ».

➤ **Classe « CELLULE »**. c'est une classe qui représente une cellule par un vecteur d'adresse de sommets et de triangles (Figure 27). Elle hérite de la classe « Brut » comme le montre le diagramme global. C'est-à-dire elle prend tous les attributs et les méthodes de la classe mère.

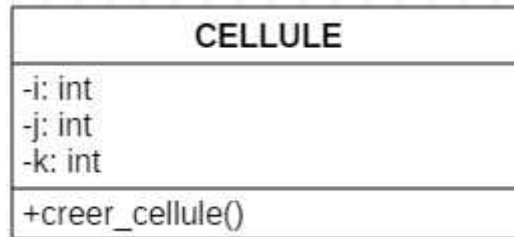


Figure 46. Classe « CELLULE ».

➤ **Classe « SOMMET »**. c'est la classe qui définit un sommet par ses différentes caractéristiques (Figure 28). Les méthodes de cette classe permettent de déterminer les différents paramètres du sommet.

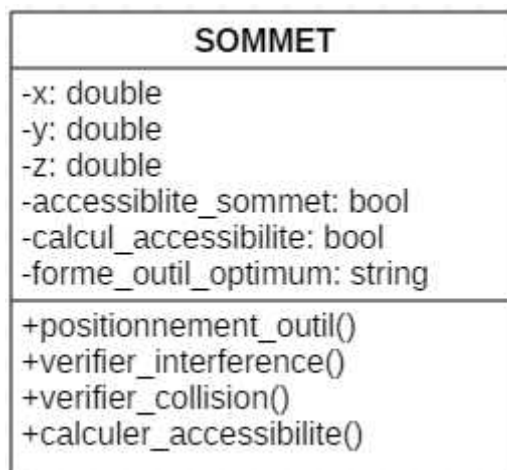


Figure 47. Classe « SOMMET ».

➤ **Classe « TRIANGLE »**. c'est une classe qui englobe les différentes caractéristiques des triangles (Figure 29). Ses fonctions sont données pour calculer les paramètres du triangle.

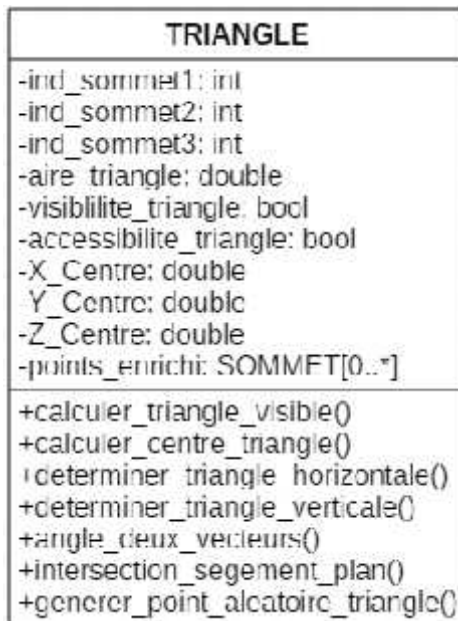


Figure 48. Classe « TRIANGLE_STL ».

➤ **Classe « OUTIL_STL »**. elle représente un outil récupéré depuis une base de données (Figure 30). Les fonctions de cette classe sont implémentées afin de déterminer les différentes caractéristiques de l'outil.



Figure 49. Classe « OUTIL_STL ».

➤ **Classe « BRUT »**. c'est la classe qui détermine les dimensions minimales d'une pièce, d'un triangle ou d'une cellule (Figure 31).

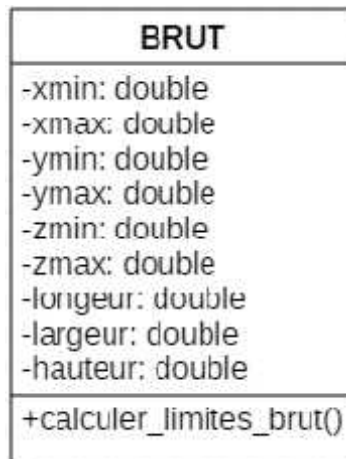


Figure 50. Classe « BRUT ».

➤ **Classe « COULEUR »**. c'est la classe qui permet d'affecter une couleur à un objet géométrique (sommet, triangle, brut, cellule, etc.) (Figure 32).

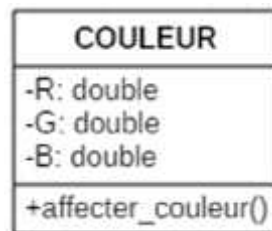


Figure 51. Classe « COULEUR ».

➤ **Classe « NORMALE »**. elle représente la normale d'un sommet, d'un triangle ou d'un plan (Figure 33).

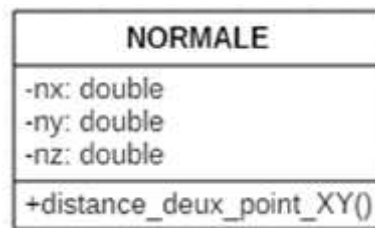


Figure 52. Classe « NORMALE ».

➤ **Classe « SEGMENT »**. c'est une classe pour définir un segment d'intersection qui est représenté par deux points de contact (Figure 34).

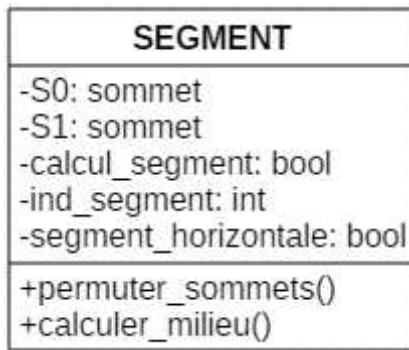


Figure 53. Classe « SEGMENT ».

➤ **Classe « CONTOUR »**. c'est une classe qui contient le tableau des segmets d'intersection et le tableau des points de contact de chaque segment (Figure 35).

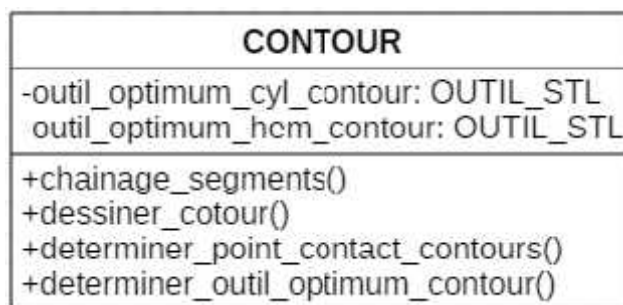


Figure 54. Classe « CONTOUR ».

➤ **Classe « SOUS_CONTOUR »**. elle permet de déterminer un sous contour d'un contour (Figure 36).

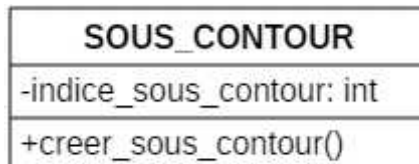


Figure 55. Classe « SOUS_CONTOUR ».

➤ **Classe « PLAN »**. elle contient le tableau des contours d'un plan donné et le tableau des segment de chaque contour (Figure 37).



Figure 56. Classe « PLAN ».

➤ **Classe « TRAJET »**. La classe TRAJET regroupe l'ensemble des plans et le pas transversal entre ses plans (Figure 38).



Figure 57. Classe « TRAJET ».

Conclusion

Dans ce chapitre, nous avons présenté la conception de notre système qui prend en charge la lecture du model STL. Ensuite, l'analyse de notre modèle par la détermination des faces verticales, horizontales, visibles et les faces accessibles à usiner. Par ailleurs, nous avons présenté notre approche qui est basée sur la création des plans parallèles verticaux sur notre modèle pour la sélection des outils optimums hors interférences et collisions tout en délimitant le champ de vérification par la création des zones de test. Les problèmes d'interférences et de collisions ont été bien détaillés pour pouvoir générer un trajet d'outil sain en utilisant des outils cylindriques et hémisphériques. Dans le chapitre suivant, l'implémentation et la mise en œuvre de notre application seront traitées à travers deux exemples de validation.

Chapitre III

Implémentation et validation

Introduction

La réalisation de ce travail s'achève par une présentation de l'application conçue ainsi que l'environnement de développement et les outils utilisés pour sa mise en œuvre. Il s'en suit une présentation de l'interface utilisateur et une validation du système à travers deux exemples de pièces.

Ce chapitre a pour but de donner une vue à la fois intelligible et relativement concrète sur l'application en présentant les informations manipulées à travers les exemples de validation.

I. Présentation des langages utilisés

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

I.1. Présentation du langage C++

Le langage C++, inventé par Bjarne Stroustrup vers 1983, est une évolution orientée objets du langage C de Brian Kernighan et Denis Ritchie. Il s'est enrichi, au cours de la décennie 1980, parallèlement à la stabilisation et la normalisation de C.



Ce langage repose sur les mêmes mécanismes d'écriture et de génération, il apporte notamment la gestion des exceptions, la gestion des références (remplaçant partiellement l'usage quelque peu délicat des pointeurs), la surcharge des opérateurs et les Templates (liste

non exhaustive). Enfin, une rétrocompatibilité a été gardée où les programmes en C se compilent sans difficulté avec un compilateur C++.

Comme tout langage, C++ dispose d'une bibliothèque standard, c'est-à-dire de fonctions et de classes prédéfinies. Elle comporte notamment de nombreux patrons de classes et de fonctions permettant de mettre en œuvre les structures de données les plus importantes (vecteurs dynamiques, listes chaînées, chaînes...) et les algorithmes les plus usuels.

Parmi les environnements de développements nous citons : AnjutaDevStudio, C++ Builder, Code::Blocks (open-source), Dev-C++, Eclipse (open-source), Microsoft Visual C++ etc.

I.2. Présentation d'OpenGL

Open Graphics Library (**OpenGL**) est une API graphique multiplateformes open-source, de bas niveau, dédiée pour les applications générant des images 2D ou 3D. OpenGL est sortie en 1992 par la Silicon Graphics Inc (SGI) et est actuellement gérée par Khronos Group.



OpenGL est conçue pour développer des applications graphiques portables et elle favorise l'innovation et la simplicité d'utilisation en offrant aux développeurs de nombreuses fonctionnalités de visualisation. Etant libre, ouvert et portable, la bibliothèque graphique a pu se vendre dans le milieu scientifique et industrielle, et être utilisée dans de nombreuses applications artistiques ou de traitement d'images.

Du fait de ses performances, l'API OpenGL est aussi présent dans différents domaines jeux-vidéo ludiques, les applications web, les systèmes embarqués et les téléphones portables.

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

I.3. Présentation de Borland C++ Builder 6

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



ressources. Il est compatible avec la version de norme ISO C++ de 2011. Embarcadero C++ Builder 10 Seattle est le moyen le plus rapide de créer et de mettre à jour des applications riches en données, hyper connectées et visuellement engageantes pour Windows 10, Mac, Mobile, IoT et plus encore en utilisant le standard C++.

II. Présentation de l'application

II.1. Fenêtre principale

La fenêtre principale de la plateforme logicielle développée par l'équipe « CFAO » est composée de deux parties (Figure 1) :

➤ **Partie d'affichage** : cette partie est réservée à la visualisation de 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 de manipulation** : cette partie est composée d'une barre de menu et d'un ensemble de boutons de manipulation des différents paramètres des modèles CAO, des courbes et des surfaces (création d'une surface, suppression d'une surface, modification des points de contrôle, ...etc.). Dans cette partie sont lancés tous les modules logiciels intégrés à la plateforme logicielle.

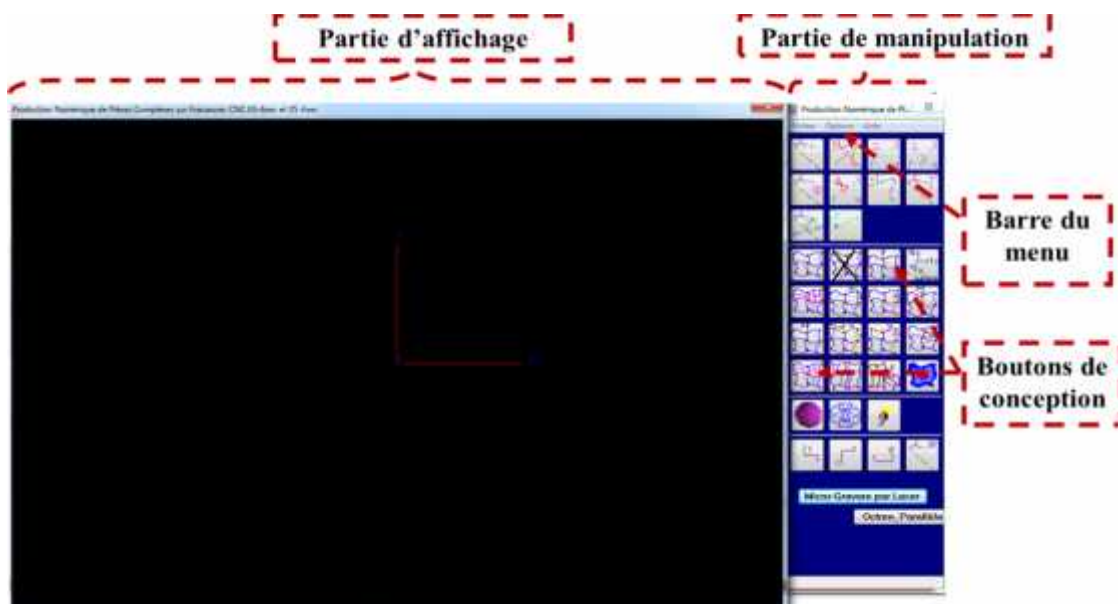


Figure 58. Fenêtre principale.

II.2. Barre du menu principal

La barre du menu principal est composée de trois rubriques :

- Rubrique « Fichier » : comporte toutes les fonctionnalités de manipulation des fichiers (ouverture, création, sauvegarde, ...etc.).
- Rubrique « Option » : permet la modification des différents paramètres des courbes et des surface ainsi que le lancement des différentes opérations d'usinage. Les fenêtres créées dans notre travail, sont lancées à partir de cette rubrique.
- Rubrique « Aide » : pour l'affichage de l'aide.

II.3. Rubrique « Finition pièces complexes en 03-axes à partir d'un fichier STL »

L'exécution de notre application passe par la récupération des informations du fichier STL du modèle des surfaces à usiner. Ces informations sont récupérées à partir des modèles contenus dans des applications développées antérieurement à travers la rubrique « Option, Usinage 05-axes, Combinaisons Outils 05-axes » (Figure 2). La lecture du fichier STL se fait par un simple clic sur le bouton « Lire Fichier STL » et la sauvegarde dans un fichier texte par un simple clic sur le bouton « Sauvegarder » (Figure 3).

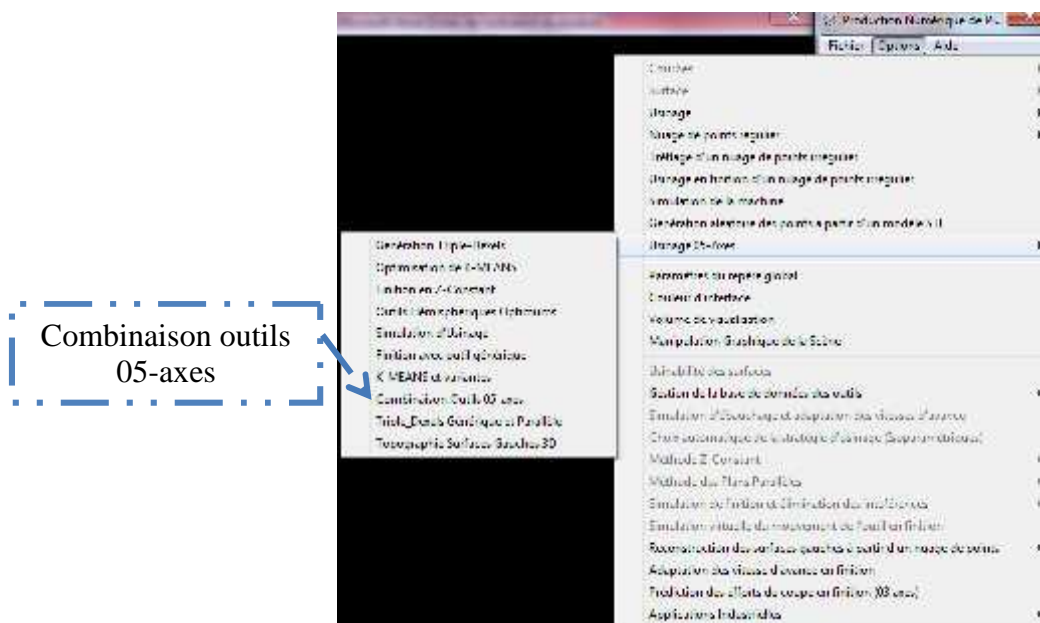


Figure 59. Rubrique « Combinaisons outils 05-axes ».

Ensuite, la partie contenant notre module logiciel développé dans le cadre de ce travail, est aussi lancée à partir du menu au niveau de la rubrique « Option, Applications Industrielles,

Finition Pièces Complexes en 03-Axes » (Figure 4). Le lancement et l'exécution des différentes fonctions de finition 03-axes passe par l'ouverture d'un fichier texte contenant le modèle de la pièce à usiner.

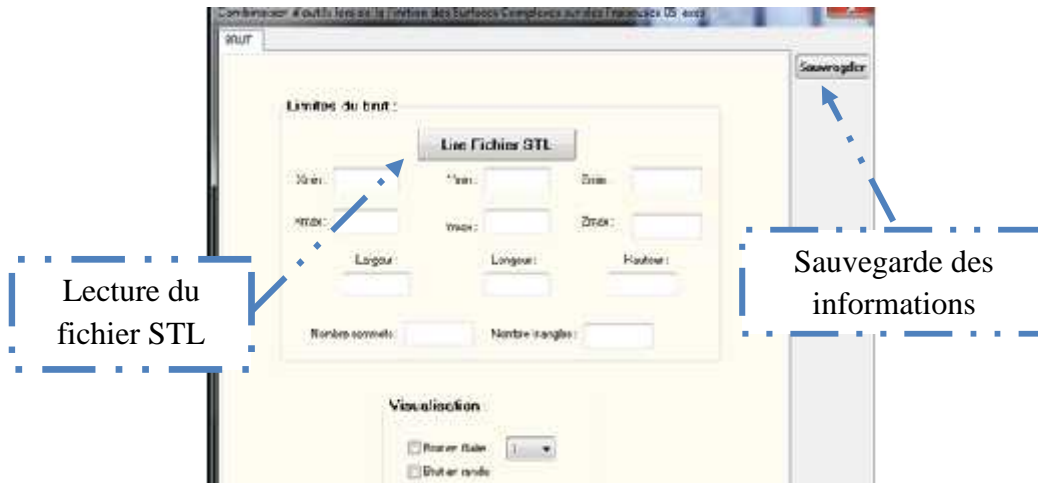


Figure 60. Lecture du fichier STL.



Figure 61. Rubrique « Finition Pièces Complexes en 03-Axes ».

III.Présentation des onglets

Le module logiciel développé est composé de neuf (09) onglets (Figure 5) pour l'activation des fonctionnalités intégrées à la plateforme logicielle.

- Premier onglet : récupération des paramètres du modèle.
- Deuxième onglet : enrichissement du modèle et création des cellules.
- Troisième onglet : détermination des faces visibles et accessibles.
- Quatrième onglet : détermination des faces horizontales, verticales et proches de la verticale.
- Cinquième onglet : création des contours de contact.
- Sixième onglet : tests manuels des différentes formes d'outils.
- Septième onglet : affectation des outils optimums.
- Huitième onglet : création des sous-contours (trajet des outils).
- Neuvième onglet : simulation des mouvements d'outils.



Figure 62. Onglets de l'application logicielle développée.

III.1. Lecture du fichier

Le premier onglet permet la lecture du fichier texte qui regroupe toutes les informations contenues dans la définition des triangles et leurs sommets récupérés à partir du modèle STL (Figure 6). Ceci est réalisé par simple clic sur le bouton « Ouvrir un fichier » suivi par la sélection du fichier à ouvrir.



Figure 63. Onglet «Paramètres du modèle».

Une fois le fichier est lu, il est possible de visualiser les informations suivantes :

- Les paramètres de la pièce brute : **Xmin**, **Ymin**, **Zmin**, **Xmax**, **Ymax**, **Zmax**, **longueur**, **largeur** et **hauteur**.
- Le nombre des sommets, des triangles et l'aire totale.
- Le brut du modèle en deux modes filaire et rendu.
- Les triangles du modèle en deux modes filaire et rendu.
- Les sommets des triangles du modèle.
- Les normales des triangles du modèle.
- Les normales des sommets du modèle.

III.2. Enrichissement du modèle et création des cellules

Dans cet onglet, l'utilisateur doit effectuer trois étapes (Figure 7). La première étape consiste à enrichir le modèle par un clic sur le bouton « Générer points ». Ensuite, diviser le brut en cellules ou le nombre des cellules suivant les trois axes **X**, **Y** et **Z** est spécifié. Une fois les cellules créées, un clic sur le bouton « Affecter sommets et triangles », permet l'affectation des sommets et des triangles aux cellules correspondantes.



Figure 64. Onglet «Cellules».

A ce niveau, les options suivantes sont disponibles :

- Visualisation des cellules en deux modes filaire et rendu.
- Visualisation des sommets et des triangles des cellules.
- Visualisation des paramètres d'une cellule sélectionnée.

III.3. Visibilité et accessibilité

Cet onglet est devisé en deux parties. La première partie sert à déterminer la visibilité des triangles du modèle (Figure 8). A ce stade, il est possible de visualiser :

- Le nombre, l'aire et le pourcentage des triangles visibles.
- Les triangles visibles en mode filaire et rendu.
- Les triangles invisibles en mode filaire et rendu.



Figure 65. Onglet « Visibilité et Accessibilité ».

La deuxième partie sert à déterminer l'accessibilité des triangles visibles (Figure 8). Une fois l'accessibilité des sommets des triangles visibles est traitée, il est possible de déterminer l'accessibilité de chaque triangle visible. Deux modes de calcul sont proposés :

- Aire des triangles.
- Produit vectoriel.

A ce niveau, nous pouvons :

- Récupérer le nombre, l'aire et le pourcentage des triangles accessibles.
- Visualiser les sommets accessibles des triangles du modèle.
- Visualiser les triangles accessibles en mode filaire et rendu.
- Visualiser les triangles inaccessibles en mode filaire et rendu.

III.4. Faces du modèle

Cet onglet est divisé en trois parties. La première partie permet de déterminer les triangles horizontaux (Figure 9). Par un clic sur le bouton « Faces horizontales » on peut :

- Récupérer le nombre, l'aire et le pourcentage des triangles horizontaux.
- Visualiser les triangles horizontaux en mode filaire et rendu.
- Visualiser les triangles horizontaux accessibles en mode filaire et rendu.



Figure 66. Onglet « Faces du Modèle ».

La deuxième partie permet de déterminer les triangles verticaux (Figure 9). Par un clic sur le bouton « Faces verticales », il est possible de :

- Récupérer le nombre, l'aire et le pourcentage des triangles verticaux.
- Visualiser les triangles verticaux en mode filaire et rendu.

La dernière partie permet de déterminer les triangles proches de la verticale (Figure 9). Après le choix de l'angle et par un clic sur le bouton « Faces proches verticales », l'utilisateur peut visualiser :

- Les triangles proches de la verticale en mode filaire et en mode rendu.

III.5. Trajet d'usinage

Cet onglet est subdivisé en deux étapes. La première étape consiste à créer les plans verticaux qui nécessitent l'introduction de deux paramètres à savoir la distance entre deux plans « pas transversal » et l'angle d'orientation des plans dans le plan XY. Par la suite, un clic sur le bouton « Calculer intersection », les segments d'intersection de ces plans avec les triangles du modèle sont calculés (Figure 10). Une fois ces traitements sont terminés, l'utilisateur peut visualiser les informations suivantes :

- Le nombre de plans créés et le nombre des points de contact.

- Les plans verticaux en mode filaire et rendu.
- Les segments d'intersection.
- Les triangles parallèles aux plans.
- Le nombre, l'aire et le pourcentage des triangles parallèles aux plans.

La deuxième étape sert à trier les segments d'intersection et à créer les contours d'usinage de chaque plan (Figure 10). Ceci est réalisé par un simple clic sur le bouton « Chainage ». A ce stade, il est possible de visualiser :

- Les contours créés.
- Les segments horizontaux.
- Les points de contact et les normales des points de contact.

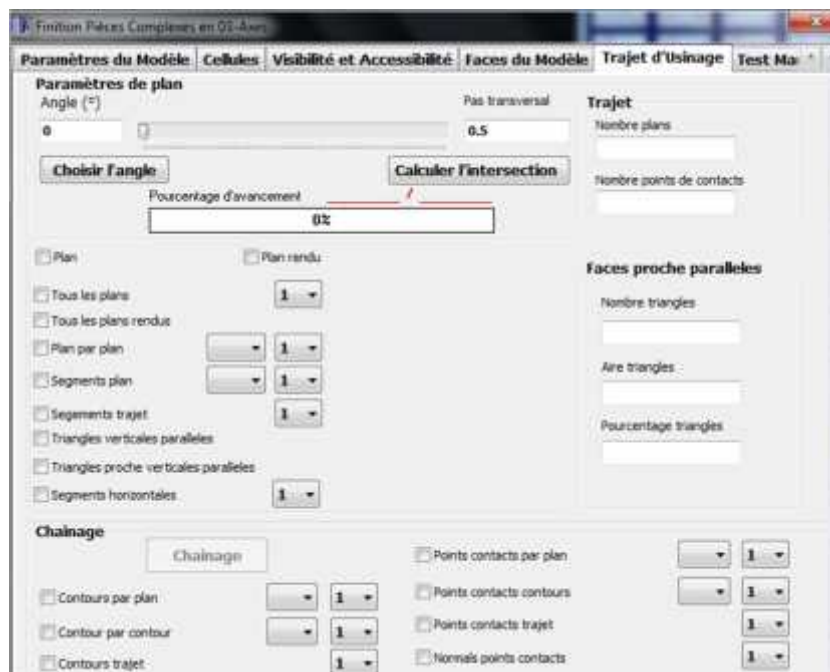


Figure 67. Onglet « Trajet d'Usinage ».

III.6. Test manuel des outils

L'intérêt principal de cet onglet est le test manuel des différentes formes d'outils. L'utilisateur peut choisir l'indice et la forme d'outil souhaités qui sont récupérés à partir d'une base de données d'outils. Ensuite, par un simple clic sur le bouton « Positionnement outil » (Figure 11), en chaque point de contact sélectionné, les paramètres d'outil sont calculés et les enveloppes sont déterminées.



Figure 68. Onglet « Test Manuel des Outils ».

Par la suite les cellules qui se chevauchent avec les enveloppes d'outil sont identifiées ainsi que la détection des interférences et des collisions en fonction de la forme d'outil.

Une fois le positionnement est effectué, Il est possible de visualiser :

- L'outil et le point de contact où se positionne l'outil.
- Enveloppes d'outil.
- Cellules chevauchantes avec les enveloppes d'outil.
- Points en collisions et en interférences avec l'outil.

Et enfin, le bouton « Déterminer outil optimum » (Figure 11), sert à déterminer, pour chaque point de contact sélectionné, l'outil optimum en parcourant la base des données d'outils du plus grand au plus petit jusqu'à trouver l'outil hors collisions et hors interférences.

III.7. Outils optimums

Cet onglet a pour but d'affecter pour chaque point de contact du trajet un outil optimum afin de déterminer la combinaison d'outils optimale pour chaque contour (Figure 12).

Au début, on affecte pour chaque point de contact le premier outil optimum trouvé. Par la suite, l'affectation est effectuée par contour, tel que le plus petit outil optimum trouvé dans un contour donné est affecté à tous les points de contact de ce contour. Enfin, par un clic sur le bouton « choisir 4 outils optimum », le trajet peut prendre au maximum quatre (04) outils cylindriques et quatre (04) outils hémisphériques.



Figure 69. Onglet « Outils Optimum ».

III.8. Trajet des outils

Cet onglet sert à créer les sous-contours et à affecter les deux points d'engagement et les deux points de dégagement pour chaque sous-contour en fonction de la distance d'engagement et la distance de sécurité d'outil choisies par l'utilisateur (Figure 13). Une fois la création est effectuée, on détermine le trajet des outils qui peut être avec ou sans optimisation en fonction du choix de l'utilisateur. A ce niveau, il est possible de visualiser :

- Les points de contact du trajet.
- Les points d'extrémités « CL » des outils.
- Le temps passé pour déterminer le trajet des outils.



Figure 70. Onglet « Trajet des Outils ».

III.9. Simulation des mouvements des outils

Le dernier onglet permet de simuler le travail réalisé en finition 03-axes par le mode de balayage « One-Way » (Figure 14). Des options sont intégrées à cet onglet tel que le lancement, la pause, la continuation du point de départ de la simulation. Il est possible de visualiser :

- La position de l'outil.
- Les points de contact usinés.
- Les points d'extrémité de l'outil « CL ».
- Trajet d'usinage.

Lors de l'exécution de la simulation, les informations suivantes sont affichées :

- Coordonnées du point de contact (X, Y, Z)
- Coordonnées du point « CL » (X, Y, Z).
- Les différents paramètres d'outil : forme, rayon, longueur et vitesse d'usinage.
- Le pourcentage d'avancement.

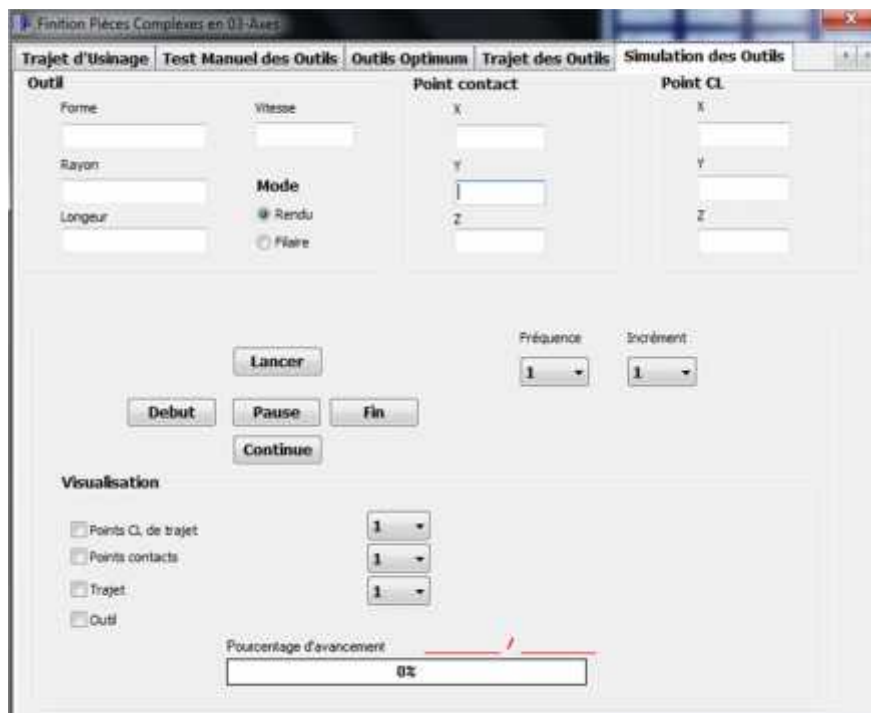


Figure 71. Onglet « Simulation des Outils ».

IV. Test et validation

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

- Windows 7
- Processeur: Intel(R) Pentium(R) CPU B960.
- Mémoire installée (RAM) : 4.00 (3.82 Go utilisable).
- Type de système: système d'exploitation 64 bits.

La validation du travail s'est faite sur huit modèles STL différents générés à partir de leurs modèles CAO et seule deux exemples sont explicités dans la suite de ce travail.

Le premier modèle vise à montrer les différentes étapes de validation et l'usinage avec les outils hémisphériques. Par contre, le deuxième modèle est composé d'un ensemble de faces (verticales, horizontales, complexes, etc.) adapté pour la finition avec combinaison des outils cylindriques et hémisphériques.

IV.1. Premier modèle STL

L'exemple considéré est représenté par la Figure 15. Les résultats des différentes étapes présentés dans les paragraphes suivants sont relatifs à ce modèle STL.

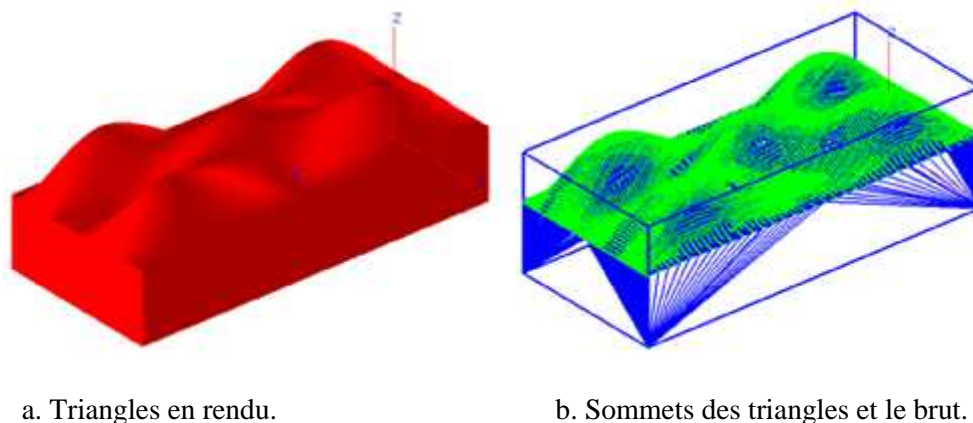


Figure 72. Visualisation du premier modèle STL.

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

Xmin = -27.283 mm	Ymin = -54.134 mm	Zmin = -28.132 mm
Xmax = 150.981 mm	Ymax = 25.867 mm	Zmax = 29.324 mm
Longueur = 178.2639 mm	Largeur = 80 mm	Hauteur = 57.4553 mm

Nombre total des sommets = 27320

Nombre total des triangles = 54636

Aire totale = 50626.643mm².

Les autres paramètres qui sont visualisés sont donnés par la Figure 16 et la Figure 17.

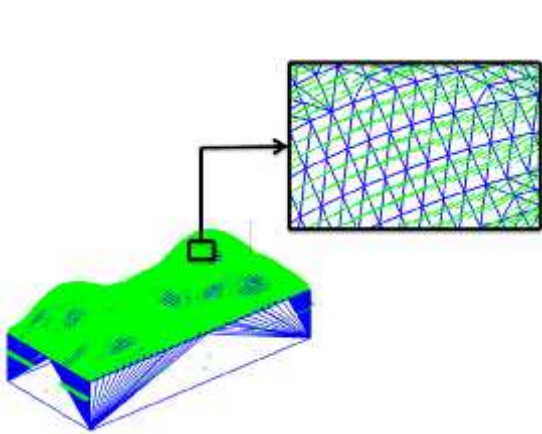


Figure 73. Normales des triangles.

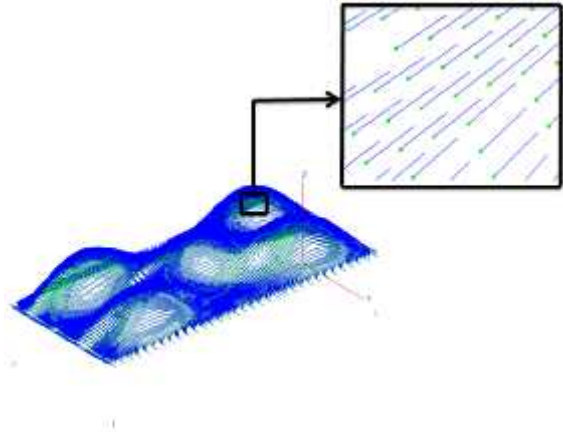


Figure 74. Normales des sommets.

- ❖ **Etape 2 :** subdivision du brut en cellules avec un nombre de 05 cellules suivant les trois axes **X**, **Y** et **Z**. Après l'enrichissement du modèle et l'affectation des sommets et des triangles aux cellules, les résultats obtenus sont donnés par la Figure 18.

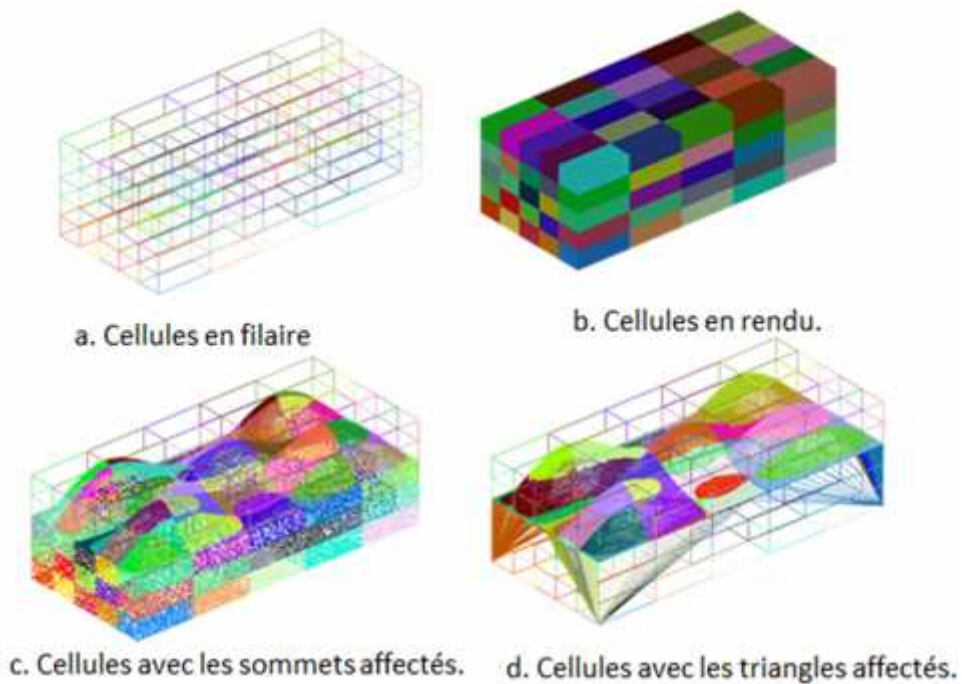


Figure 75. Création des cellules et affectation des points.

- ❖ **Etape 3 :** détermination de la visibilité des triangles (Figure 19) suivie par la détermination de l'accessibilité des sommets (Figure 20) et des triangles par le choix du mode de calcul « pondération par les aires des triangles » (Figure 21).

Les résultats de la visibilité :

Nombre des triangles= 54256 Aire = 18743.8 mm² Pourcentage= 37.02 .

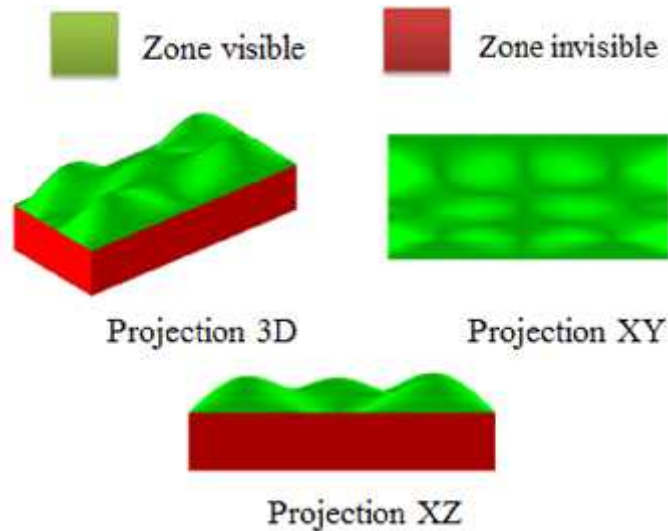


Figure 76. Visibilités des triangles.



Figure 77. Sommets accessibles.

Les résultats de l'accessibilité sont :

Nombre des triangles = 54256 Aire = 18743.79 mm² Pourcentage= 37.02 .

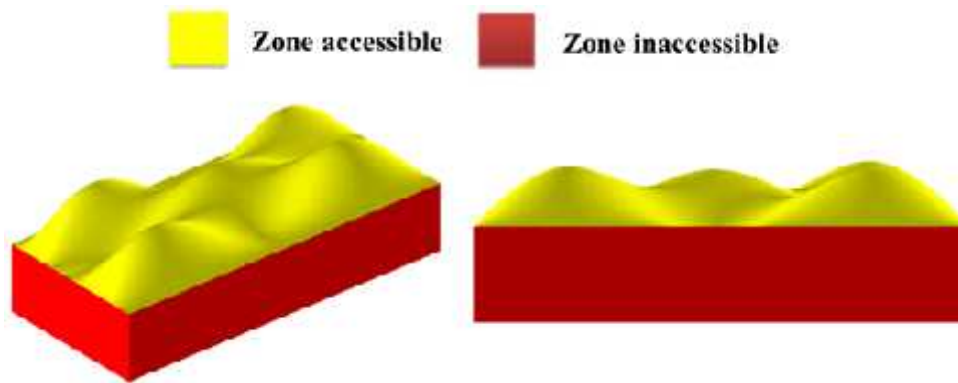


Figure 78. Accessibilités des triangles.

❖ **Etape 4 :** détermination de faces horizontales et verticales (Figure 22).

Les résultats sont :

Nombre des triangles horizontales = 2 Aire face horizontale = 14261.12 mm²

Pourcentage face horizontale = 28.16 .

Nombre des triangles verticales = 378 Aire face verticale = 17621.72 mm²

Pourcentage face verticale = 34.80 .

 **Faces horizontales**  **Faces verticales**

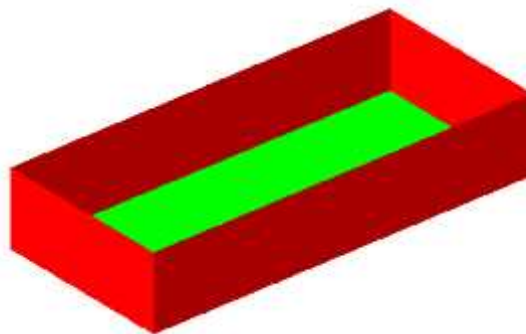


Figure 79. Faces de la pièce.

❖ **Etape 5 :** création des plans verticaux avec un angle de 45° et un pas transversal de 5 mm et calcul des segments d'intersection. Par la suite, le chainage de segments et la création des contours sont effectués (Figure 23).

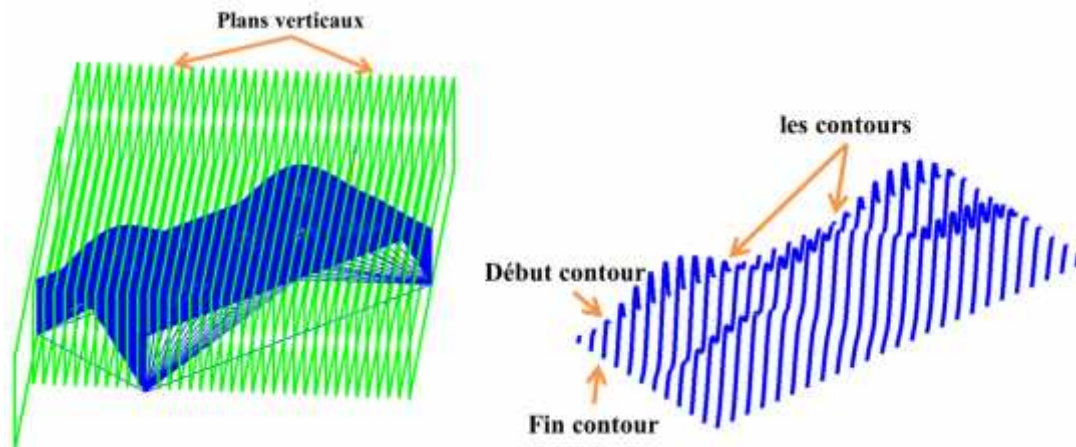


Figure 80. Création des plans et des contours.

- ❖ **Etape 6 :** on identifie le positionnement d'une forme d'outil pour un point de contact donné en fonction du choix de l'utilisateur est identifié (les résultats sont montrés dans le deuxième modèle STL). Par la suite, détermination de l'outil optimum en chaque point de contact sélectionné.
- ❖ **Etape 7 :** cette étape a pour but d'affecter les outils optimums aux points de contact du trajet.
- ❖ **Etape 8 :** création des sous-contours et l'affectation de deux points d'engagement et de deux points de dégagement d'outil est établi, avec une distance d'engagement égale à 10 mm, une distance de sécurité de 50 mm et une vitesse maximale égale à 1000 mm/min. Ainsi que la détermination du trajet des outils avec deux modes (sans optimisation et avec optimisation).

La Figure 24 montre la minimisation du temps de calcul par l'optimisation du trajet des outils.

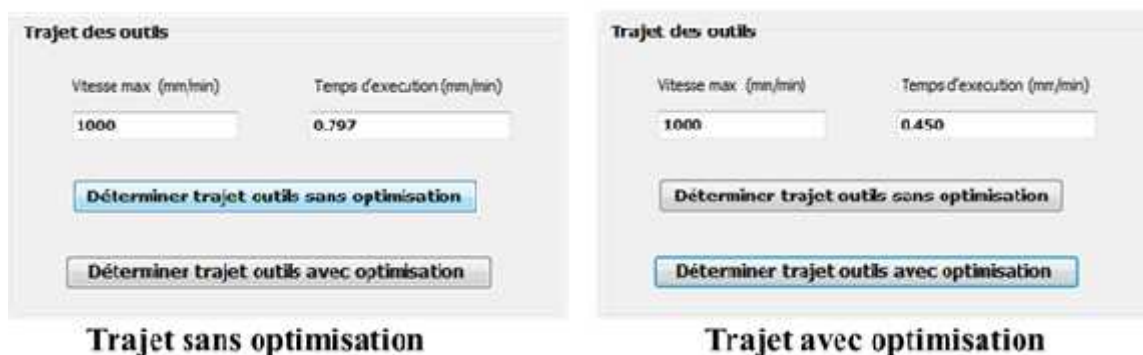


Figure 81. Minimisation du temps avec la phase d'optimisation.

- ❖ **Etape 9 :** simulation du trajet d'usinage pour la finition 03-axes avec le mode « One-Way » (Figure 25).

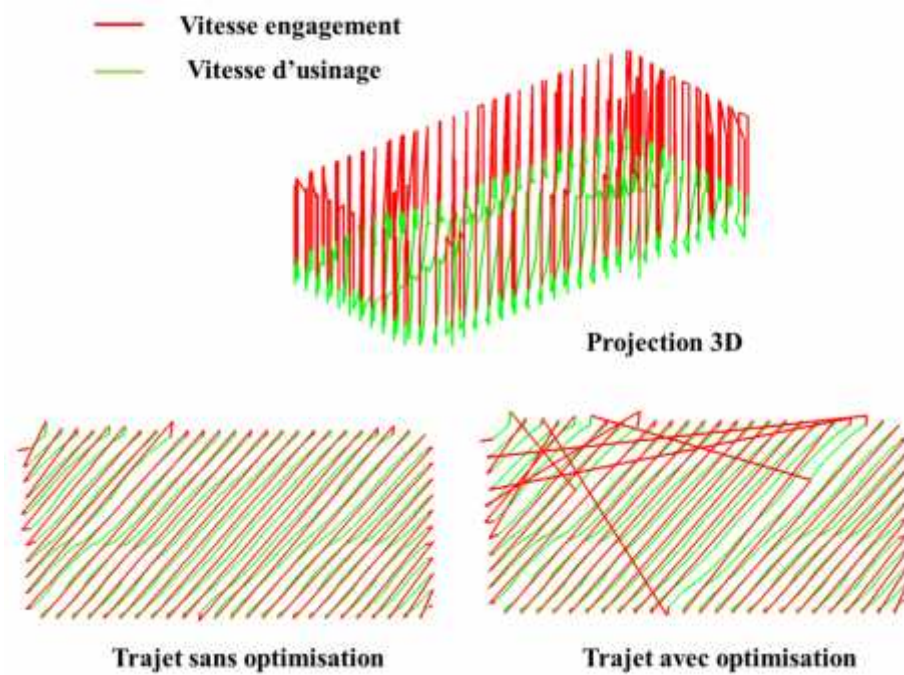


Figure 82. Simulation de trajet en 03-axes.

Lorsque la simulation est en cours de traitement, par un clic sur « pause » à n'importe quel moment, l'outil est visualisé, ses informations ainsi que celles du point de contact où il est positionné et du point « CL » en cours sont récupérées (Figure 26). Dans cet exemple, la forme d'outil est hémisphérique pour tout le trajet d'usinage.

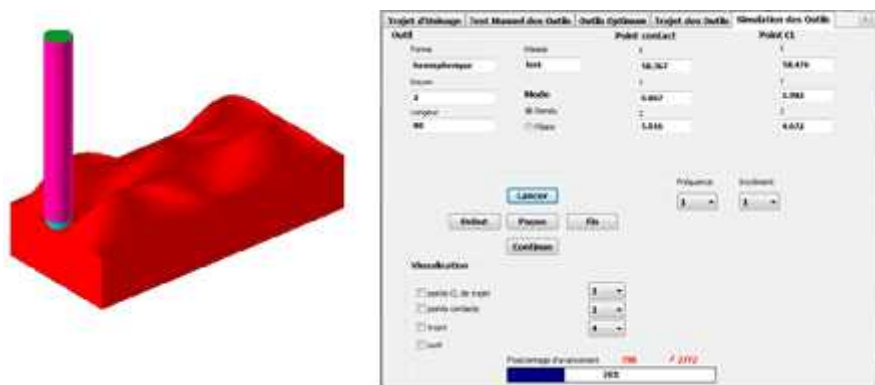
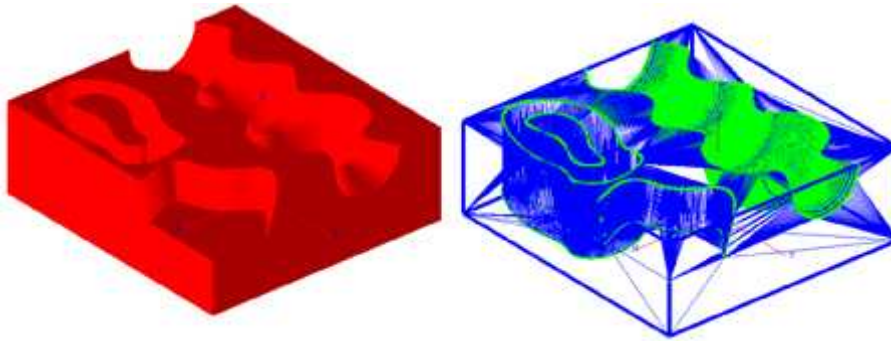


Figure 83. Visualisation d'outil et coordonnées du point courant.

IV.2. Deuxième modèle STL

Les résultats des différentes étapes essentielles présentées dans les paragraphes suivants sont relatifs au deuxième modèle STL (Figure 27).



a. Triangle en rendu.

b. Sommets des triangles et le brut.

Figure 84. Visualisation du deuxième modèle STL.

- ❖ **Etape 1 :** lecture du fichier et calcul des limites du brut. Les résultats sont donnés par le Tableau 1.

Tableau 1. Paramètres du modèle STL.

(Xmin, Xmax)	(Ymin, Ymax)	(Zmin, Zmax)	Longueur (mm)	Largeur (mm)	Hauteur (mm)	Nombre de triangles	Nombre de sommets	Aire total (m ²)
(-49.8 , 90.1)	(-77.9 , 72.09)	(0 , 50)	140	150	50	56962	28483	89634.5

- ❖ **Etape 2 :** subdivision du brut en cellules avec 05 cellules suivant **X**, **Y** et **Z**, l'enrichissement du modèle et l'affectation des points et des triangles aux cellules.
- ❖ **Etape 3 :** détermination de la visibilité et l'accessibilité du modèle (Figure 28, Figure 29 et Figure 30). Les résultats obtenus sont donnés par le Tableau 2.

Tableau 2. Visibilité et accessibilité.

	Nombre de triangles	Aire (m ²)	Pourcentage ()
Visibilité	55790	28518.29	31.8
Accessibilité	55790	28518.29	31.8

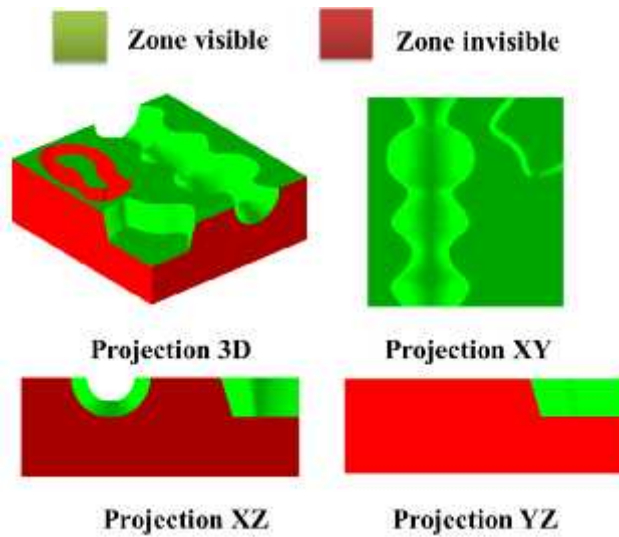


Figure 85. Visibilités des triangles.

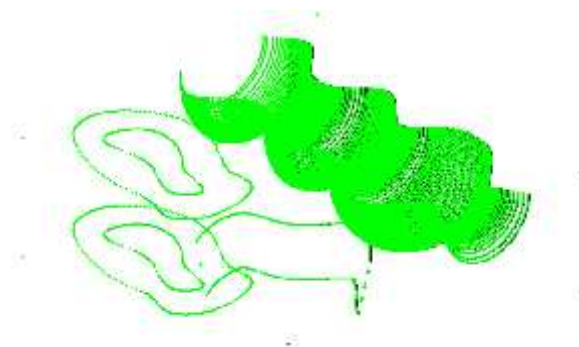


Figure 86. Sommets accessibles.

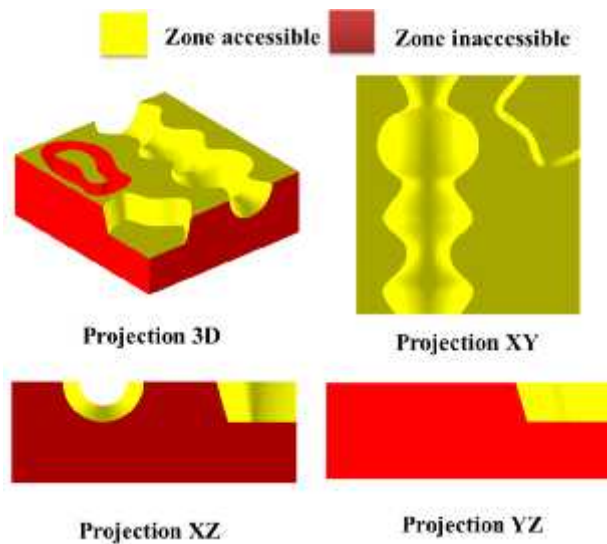


Figure 87. Accessibilités des triangles.

- ❖ **Étape 4** : détermination des faces de la pièce (Figure 31). Les résultats sont dans le Tableau 3.

Tableau 3. Faces du modèle STL.

	Nombre de triangles	Aire (m ²)	Pourcentage (%)
Faces horizontales	2283	34874.04	38.90
Faces verticales	1170	40116.22	44.75

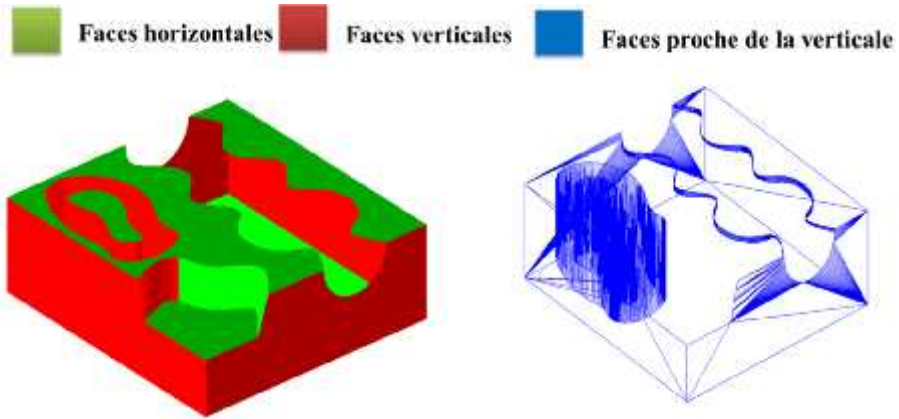


Figure 88. Faces de la pièce.

- ❖ **Etape 5 :** pour un angle de 45° et un pas transversal de 10 mm, les résultats sont montrés par la Figure 32 et la Figure 33.

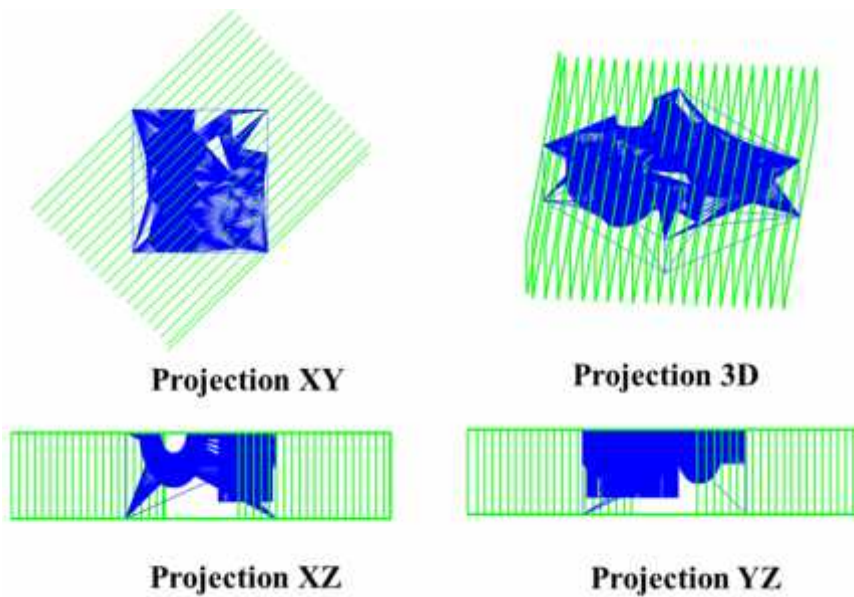


Figure 89. Création des plans.

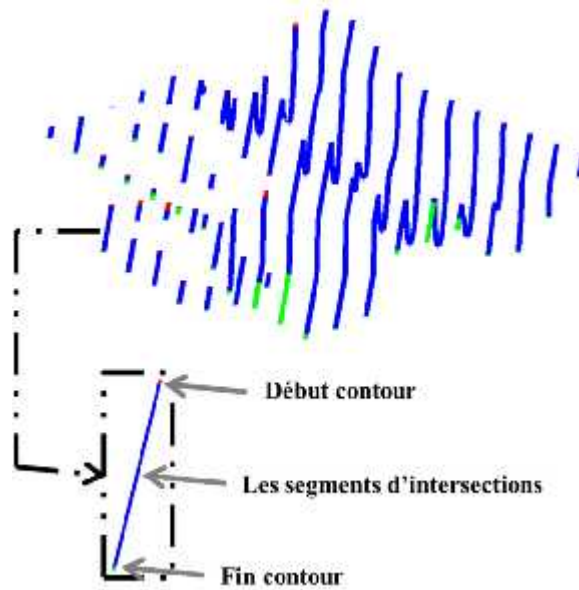


Figure 90. Création des contours.

- ❖ **Étape 6 :** les paramètres et le positionnement d'outil pour un point de contact donné en fonction du choix de l'utilisateur est identifié (Figure 34, Figure 35 et Figure 36). Par la suite, la détermination de l'outil optimum en chaque point de contact est effectuée.

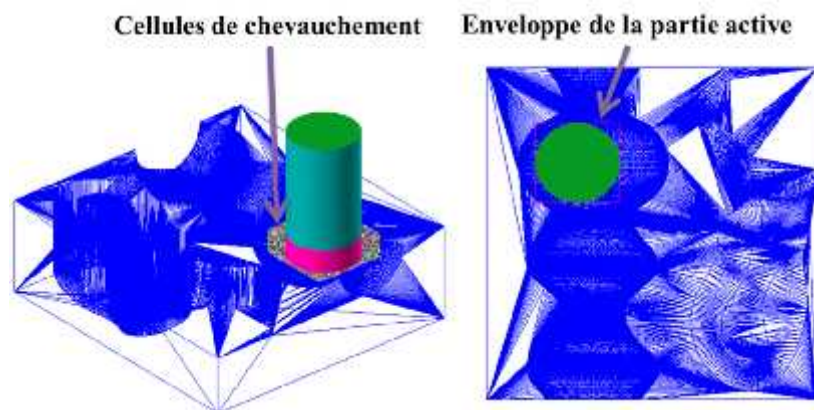


Figure 91. Enveloppe de la partie active d'outil et cellules de chevauchement.

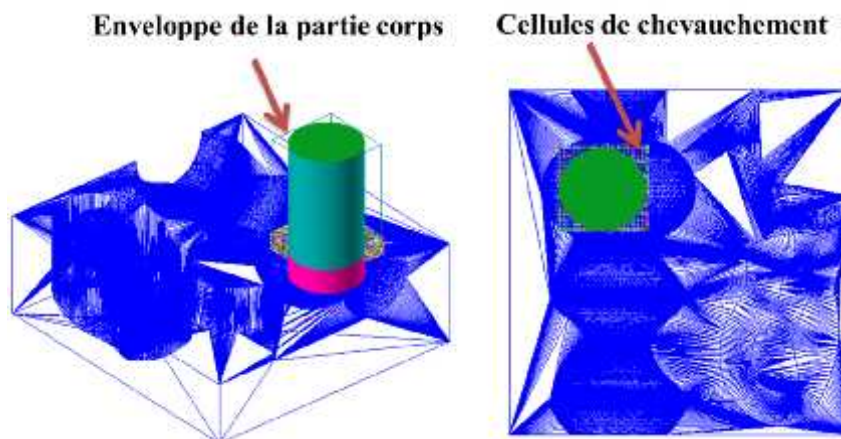


Figure 92. Enveloppe de la partie corps d'outil et cellules de chevauchement.

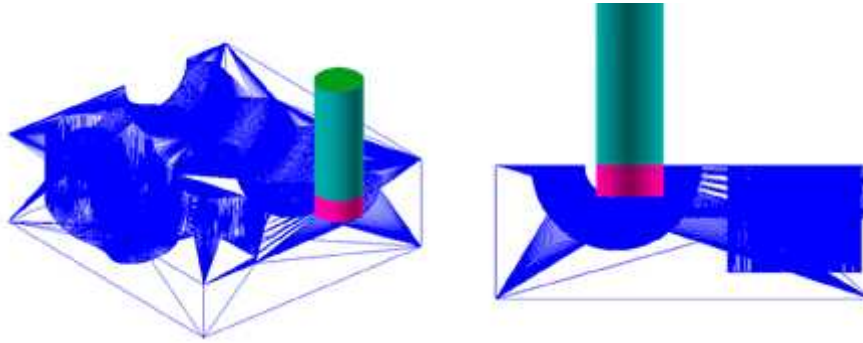


Figure 93. Point de contact avec interférences et collisions.

- ❖ **Etape 7 :** les paramètres des outils optimums aux points de contact sont déterminés.
- ❖ **Etape 8 :** les résultats de cette étape sont donnés dans le Tableau 4.

Tableau 4. Trajet avec et sans optimisation.

	Distance d'engagement (mm)	Distance de dégagement (mm)	Vitesse maximale (mm/min)	Temps d'usinage
Trajet sans optimisation	10	50	1000	3min 39s
Trajet avec optimisation	10	50	1000	1min8s

- ❖ **Etape 9 :** simulation du trajet d'outil avec le mode « One-Way » (Figure 38).

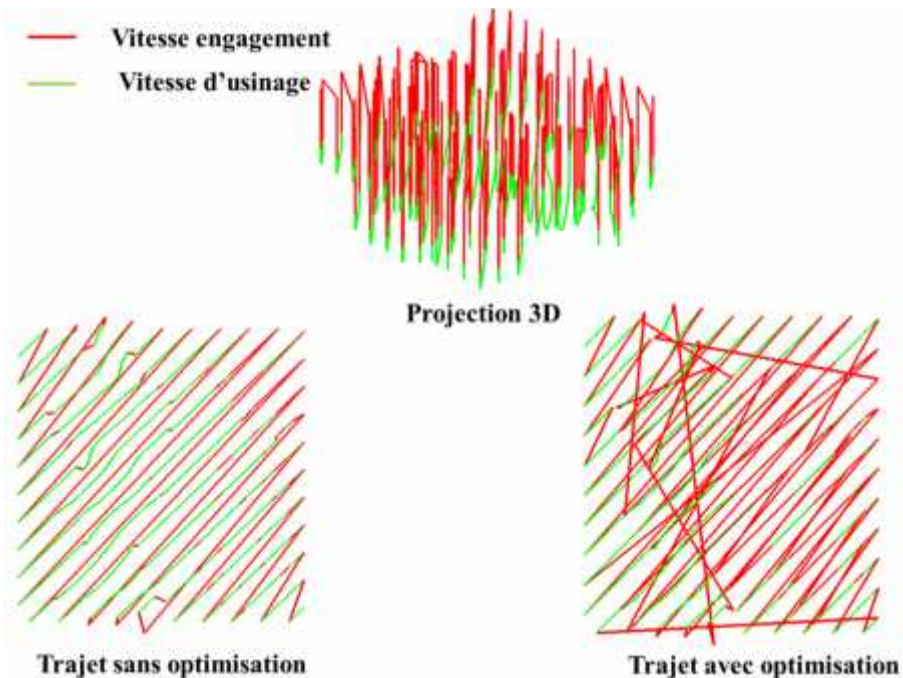


Figure 94. Simulation du trajet en 03-axes.

À n'importe quel moment, l'outil est visualisé et ses informations sont récupérées (Figure 38). Aussi, les informations du point de contact où il est positionné l'outil et du point « CL » en cours sont récupérées.

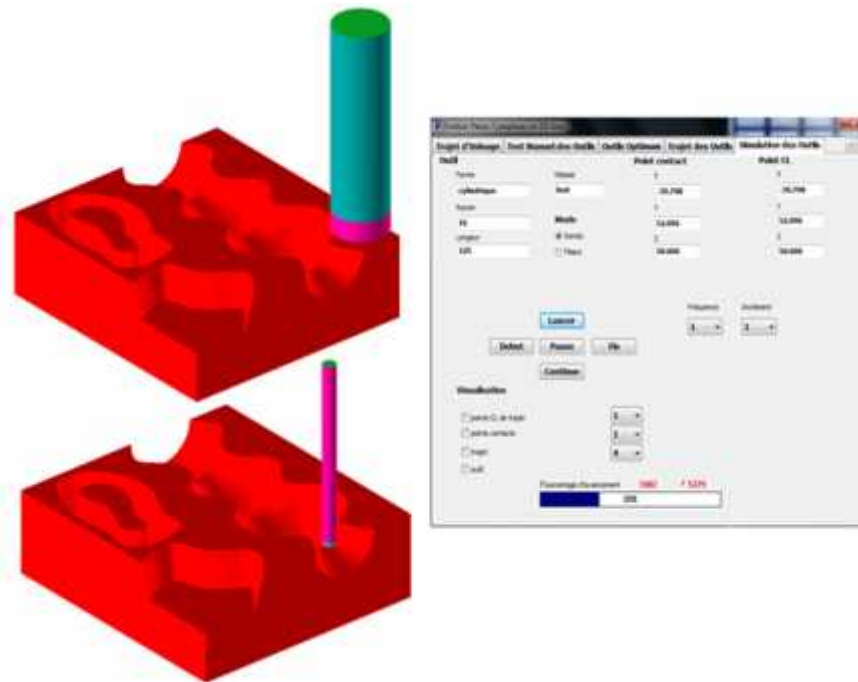


Figure 95. Visualisation de l'outil et des coordonnées du point courant.

Conclusion

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

CONCLUSION GENERALE

Ce mémoire est le résultat de notre projet de fin d'études mené au niveau de l'équipe « CFAO » du « CDTA ». Il consiste à concevoir et à implémenter une application logicielle graphique et interactive pour la finition des pièces complexes, définies par des modèles « STL » en utilisant la stratégie de finition « Plans Parallèles » sur des fraiseuses numériques à 03-axes avec la combinaison des outils cylindriques et hémisphériques. Cette application logicielle permet de générer une trajectoire d'outils sans interférences et sans collisions.

Lors de la réalisation de ce projet, une étude bibliographique a été faite sur les modèles de représentation des surfaces complexes et les différentes caractéristiques et propriétés de ces surfaces ont été analysées toute en focalisant notre analyse sur les surfaces discrètes ainsi que sur les machines-outils numériques et les différentes stratégies d'usinage utilisées dans la phase de fabrication. Cette étude a été complétée par une étude conceptuelle de l'application et une implémentation informatique des solutions proposées aux différentes questions posées durant la réalisation de cette approche sous Windows en utilisant le langage de programmation « C++ », l'environnement de développement « Embarcadero » et la bibliothèque graphique « OpenGL ». A la fin, des tests de validation ont été menés sur deux pièces, l'une est de forme très complexe.

Le principal résultat obtenu, est l'intégration à la plateforme de fabrication de pièces de formes complexes développée par l'équipe « CFAO », d'un module logiciel graphique interactif sous Windows permettant de réaliser les tâches suivantes :

- ✓ Récupération des principaux paramètres du modèle de la pièce.
- ✓ Création des cellules et affectation des sommets et des triangles à ces cellules.
- ✓ Détermination des faces horizontales, des faces verticales et des faces proches de la verticale.
- ✓ Localisation des zones visibles.
- ✓ Identification des zones accessibles de la pièce à usiner.
- ✓ Création des plans d'usinage verticaux.

- ✓ Calcul des points de contact outil/surface entre les plans d'usinage et les triangles.
- ✓ Génération des segments d'intersection et création des contours sur chaque plan.
- ✓ Positionnement des outils hors interférences et hors collisions.
- ✓ Détermination de l'outil optimum en chaque point de contact.
- ✓ Détermination de l'outil optimum pour chaque contour.
- ✓ Détermination de la combinaison optimale composée de quatre (04) outils cylindriques et de quatre (04) outils hémisphériques pour la finition de toute la pièce.
- ✓ Génération du trajet d'usinage sans optimisation.
- ✓ Génération du trajet d'usinage avec optimisation pour minimiser le temps d'usinage.
- ✓ Simulation virtuelle des mouvements des outils et estimation du temps d'usinage.

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

- Combinaison des trois formes d'outils hémisphériques, cylindriques et toriques.
- Calcul des cônes de visibilité et d'accessibilité pour accélérer la détermination des zones visibles et des zones accessibles.
- Développement et implémentation des stratégies d'ébauchage des pièces de formes complexes.
- Développement et implémentation des stratégies de demi-finition des pièces de formes complexes.
- Génération du trajet d'outil en utilisant d'autres stratégies d'usinage (Z-Constant, Isoparamétrique, Iso-Crête, ...etc.).
- Combinaison de deux stratégies d'usinage « Z-Constant » et « Plans-Parallèles » pour tenir compte de la position des faces de la pièce par rapport aux plans horizontal et vertical.
- Génération du trajet d'outil par le mode de balayage de l'outil « Zig-Zag ».
- Génération du programme d'usinage « G-Code » en fonction des différentes configurations des fraiseuses à 03-axes.
- Développer une approche pour la simulation virtuelle de l'opération d'enlèvement de matière en finition afin de contrôler la qualité de la pièce usinée.

REFERENCES BIBLIOGRAPHIQUES

1. B. Bloul, B. Aour, A. Bourdim, R. Harhout. Modélisation de l'état de surface d'une pièce rectifiée en utilisant ANN, 31^{er} Congrès International sur les Ingénieries Civile, Mécanique et Electrique pour l'EnergieCMEEE 2015- Marrakech.
2. C. Tournier. Contribution à la conception des formes complexes :la surface d'usinage en fraisage 5 axes isocrête, These de doctorat, Ecole normale superieure de Cachan, 12 Décembre 2001.
3. L. Hadj abdellah, H. Mohammed Chérif. Présentation d'une gamme d'entretien préventive d'une fraiseuse de type 6 P 13, Mémoire fin d'études, Département de Mécanique, Université Kasdi Merbah-Ouargla, 2011.
4. A. Perles. Optimisation d'usinage de surface gauche par système multi-agent auto-organisateur, Rapport Master 2, Université de Paul Sabtier Toulouse, 05 juillet 2013.
5. R. Laguionie. Programmation avancée des machines outils : implementation de l'usinage de trochoidal et de treilage dans la chaine numerique step_NC, These Master 2 SMA, Ecole Centrale de Nantes, Université de Nantes, 01 septembre 2007.
6. <https://www.usimm.ca/quels-sont-les-differents-types-de-fraiseuses/> [consulté le 20 septembre 2019]
7. <http://junior.universalis.fr> [consulté le 20 septembre 2019]
8. SE. Bendaoudi. Conception Assistée par Ordinateur (Cours & Exercices d'application), Master Académique, département de Génie Mécanique, Université Ahmed Zabana, Relizane.
9. S. Boudjewouad, N. Tafat-Bouزيد. Méthode des plans parallèles pour l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 03-axes, Mémoire fin d'études, Institut d'Informatique, Université Houari Boumediene, Bab Zouar. 2004.
10. A. Driouech, D. Larbi. Tréilage des Surfaces Gauches Discrètes sur des fraiseuses 05-axes. Mémoire fin d'études, département d'Informatique, Université Saad Dahleb, Blida, 2018.

11. S. Boumenir, M. Debeib. Conception et Développement d'une Approche Optimale Pour la Génération du modèle Triple-Dexels de Pièces de Formes Complexes. Mémoire fin d'études, département d'Informatique, Université Saad Dahleb, Blida, 2018.
12. <https://all3dp.com/what-is-stl-file-format-extension-3d-printing> [consulté le 20 septembre 2019]
13. L. Ghemouri, S. Rezkallah. Optimisation de la finition des surfaces complexes sur des fraiseuses 05-axes par combinaison des outils hémisphériques, cylindriques et toriques. Mémoire fin d'études, département d'Informatique, Université Saad Dahleb, Blida, 2018.
14. R. Alia, S. Guemaz. Optimisation des paramètres de coupe pour minimiser la consommation d' énergie pendant le fraisage, Mémoire Master Professionnel, département Génie Mécanique, Université kasdi Merbah, Ouargla 2015.
15. A. Lakrib, M. Boutagga. Conception et réalisation d'une application Pour finition des surfaces complexes sur des fraiseuses à 05-axes, Mémoire fin d'études, Département Mathématique et informatique, Université Saad Dahleb Blida 2015.
16. M. Abdelhak khattab. Programation des MOCN. 2017.
17. S. Djebali. Optimisation globale du processus d'usinage des surfaces gauches. These de doctorat, Université -Paul Sabatier, Toulouse, 13 Novembre 2014.
18. P. Gilles, Positionnement d'outil torique pour l'usinage de surfaces gauches en fraisage 5-axes avec equilibrage de l'effort de coupe transversal. 2009. Thèse de doctorat.
19. M. Bey, H. Bendifallah, M. Dermache, N. Abdelhamid. Finition des surfaces gauches par combinaison de différentes formes et dimensions d'outils. Conference: Huitièmes Journées de Mécanique, JM'08, Centre de Développement des Technologies Avancées, Alger, 10 au 11 AvriLe 2012.
20. B. Ouyaiche, S. Kouihal. Développement d'une application pour la finition des surfaces complexes sur des fraiseuses à 05 axes, Mémoire de fin d'étude Université Saad Dahlab, Blida, 2017.
21. <https://math.stackexchange.com/questions/1785136/generating-randomly-distributed-points-inside-a-given-triangle> [consulté le 20 septembre 2019]
22. <https://fr.wikihow.com/calculer-l'angle-entre-deux-vecteurs> [consulté le 20 septembre 2019]
23. www.les-mathematiques.net/phorum/read.php [consulté le 20 septembre 2019]
24. <https://lexique.netmath.ca/formule-de-heron/> [consulté le 20 septembre 2019]
25. <http://geomalgorithms.com> [consulté le 20 septembre 2019]